LilyPond

El tipografiador de música

Referencia de la notación

El equipo de desarrolladores de LilyPond

Este manual ofrece una referencia para toda la notación musical que se puede producir con LilyPond versión 2.18.2. Da por sentado que el lector está familiarizado con el material que hay en Sección "Manual de aprendizaje" in *Manual de Aprendizaje*.

Para mayor información sobre la forma en que este manual se relaciona con el resto de la documentación, o para leer este manual en otros formatos, consulte Sección "Manuales" in *Información general*.

Si le falta algún manual, encontrará toda la documentación en `http://www.lilypond.org/`.

Copyright © 1999–2012 por los autores.

La traducción de la siguiente nota de copyright se ofrece como cortesía para las personas de habla no inglesa, pero únicamente la nota en inglés tiene validez legal.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

> Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation; sin ninguna de las secciones invariantes. Se incluye una copia de esta licencia dentro de la sección titulada "Licencia de Documentación Libre de GNU".
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Para la versión de LilyPond 2.18.2

Índice General

1 Notación musical 1

1.1 Alturas 1
1.1.1 Escritura de notas 1
Escritura de octava absoluta 1
Escritura de octava relativa 2
Alteraciones accidentales 5
Nombres de las notas en otros idiomas 7
1.1.2 Modificación de varias notas a la vez 9
Comprobación de octava 9
Transposición 10
Inversión 13
Retrogradación 14
Transformaciones modales 14
1.1.3 Imprimir las alturas 16
Clave 16
Armadura de la tonalidad 20
Corchetes de octava 23
Transposición de los instrumentos 24
Alteraciones accidentales automáticas 26
Tesitura 33
1.1.4 Cabeza de las notas 35
Cabezas de nota especiales 35
Cabezas de notas de Notación Fácil 37
Cabezas de notas con formas diversas 39
Improvisación 41
1.2 Duraciones 42
1.2.1 Escritura de las duraciones (valores rítmicos) 43
Duración de las notas 43
Grupos especiales 45
Escalar las duraciones 49
Ligaduras de unión 51
1.2.2 Escritura de silencios 54
Silencios 54
Silencios invisibles 56
Silencios de compás completo 58
1.2.3 Impresión de las duraciones 62
Indicación de compás 62
Indicaciones metronómicas 66
Anacrusas 70
Música sin compasear 71
Notación polimétrica 72
División automática de las notas 75
Mostrar los ritmos de la melodía 76
1.2.4 Barras 79
Barras automáticas 79
Establecer el comportamiento de las barras automáticas 81
Barras manuales 90
Barras progresivas 93

1.2.5 Compases ... 94
Barras de compás ... 94
Numeración de compases ... 101
Comprobación de compás y de número de compás ... 105
Llamadas de ensayo ... 106
1.2.6 Asuntos rítmicos especiales ... 108
Notas de adorno ... 108
Alinear con una cadenza ... 114
Gestión del tiempo ... 114
1.3 Expresiones ... 116
1.3.1 Expresiones adosadas a las notas ... 116
Articulaciones y ornamentos ... 116
Matices dinámicos ... 119
Indicaciones dinámicas contemporáneas ... 125
1.3.2 Expresiones como curvas ... 127
Ligaduras de expresión ... 127
Ligaduras de fraseo ... 130
Marcas de respiración ... 131
Caídas y elevaciones ... 133
1.3.3 Expresiones como líneas ... 134
Glissando ... 134
Arpegio ... 138
Trinos ... 141
1.4 Repeticiones ... 143
1.4.1 Repeticiones largas ... 144
Repeticiones normales ... 144
Marcas de repetición manual ... 152
Repeticiones explícitas ... 154
1.4.2 Repeticiones cortas ... 156
Repeticiones de compás o parte de ellos ... 156
Repeticiones de trémolo ... 159
1.5 Notas simultáneas ... 160
1.5.1 Una voz única ... 161
Notas en acorde ... 161
Repetición de acordes ... 163
Expresiones simultáneas ... 164
Racimos (clusters) ... 166
1.5.2 Varias voces ... 166
Polifonía en un solo pentagrama ... 166
Estilos de voz ... 169
Resolución de las colisiones ... 170
Combinación automática de las partes ... 175
Escribir música en paralelo ... 180
1.6 Notación de los pentagramas ... 182
1.6.1 Impresión de los pentagramas ... 183
Crear instancias de pentagramas nuevos ... 183
Agrupar pentagramas ... 184
Grupos de pentagramas anidados ... 188
Separación de sistemas ... 190
1.6.2 Modificación de pentagramas sueltos ... 191
El símbolo del pentagrama ... 191
Pentagramas de Ossia ... 194
Ocultar pentagramas ... 198
1.6.3 Escritura de las particellas ... 201

Nombres de instrumentos ... 201
Citar otras voces ... 204
Formateo de las notas guía ... 207
1.7 Anotaciones editoriales ... 213
1.7.1 Dentro del pentagrama ... 213
Seleccionar el tamaño de la tipografía para la notación ... 213
Indicaciones de digitación ... 214
Notas ocultas ... 216
Colorear objetos ... 217
Paréntesis ... 219
Plicas ... 219
1.7.2 Fuera del pentagrama ... 220
Globos de ayuda ... 221
Líneas de rejilla ... 222
Corchetes de análisis ... 224
1.8 Texto ... 225
1.8.1 Escritura del texto ... 225
Guiones de texto ... 226
Extensiones de texto ... 227
Indicaciones de texto ... 229
Texto separado ... 231
1.8.2 Formatear el texto ... 233
Introducción al marcado de texto ... 233
Seleccionar la tipografía y su tamaño ... 234
Alineación de texto ... 237
Notación gráfica dentro de elementos de marcado ... 240
Notación musical dentro de elementos de marcado ... 243
Elementos de marcado de varias páginas ... 245
1.8.3 Tipografías ... 246
Explicación de las fuentes tipográficas ... 246
Fuentes de un solo elemento ... 248
Fuentes tipográficas del documento completo ... 248

2 Notación especializada ... 250

2.1 Música vocal ... 250
2.1.1 Notación común para música vocal ... 250
Referencias para música vocal ... 250
Introducir la letra ... 251
Alineación de la letra a una melodía ... 253
Duración automática de las sílabas ... 254
Duración manual de las sílabas ... 257
Varias sílabas sobre una nota ... 258
Varias notas sobre una sílaba ... 259
Líneas de extensión y guiones ... 262
2.1.2 Técnicas específicas para la letra ... 263
Trabajar con letra y variables ... 263
Posicionamiento vertical de la letra ... 264
Colocación horizontal de las sílabas ... 269
Letra y repeticiones ... 271
Letras en divisi ... 279
Polifonía con letras compartidas ... 280
2.1.3 Versos ... 282
Añadir números de verso ... 282
Añadir expresiones dinámicas a los versos ... 283

Añadir el nombre de los cantantes a los versos . . . 283
Versos con ritmos distintos . . . 284
Imprimir los versos al final . . . 287
Imprimir los versos al final en varias columnas . . . 288
2.1.4 Canciones . . . 289
Referencias para canciones . . . 289
Hojas guía de acordes . . . 290
2.1.5 Música coral . . . 290
Referencias para música coral . . . 290
Disposiciones de la partitura para música coral . . . 291
Voces divididas . . . 293
2.1.6 Ópera y musicales . . . 293
Referencias para ópera y musicales . . . 294
Nombres de los personajes . . . 294
Guías musicales . . . 296
Música hablada . . . 300
Diálogos encima de la música . . . 300
2.1.7 Cánticos salmos e himnos . . . 302
Referencias para cánticos y salmos . . . 302
Preparar un cántico . . . 302
Puntuación de un salmo . . . 309
Compases parciales en melodías de himno . . . 312
2.1.8 Música vocal antigua . . . 314
2.2 Teclados y otros instrumentos de varios pentagramas . . . 315
2.2.1 Notación común para instrumentos de teclado . . . 316
Referencias para teclados . . . 316
Cambiar de pentagrama manualmente . . . 317
Cambiar de pentagrama automáticamente . . . 318
Líneas de cambio de pentagrama . . . 320
Plicas de pentagrama cruzado . . . 320
2.2.2 Piano . . . 322
Pedales de piano . . . 322
2.2.3 Acordeón . . . 323
Símbolos de Discanto . . . 323
2.2.4 Arpa . . . 324
Referencias para notación de arpa . . . 325
Pedales de arpa . . . 325
2.3 Instrumentos de cuerda sin trastes . . . 326
2.3.1 Notación común para cuerdas sin trastes . . . 326
Referencias para cuerdas sin trastes . . . 326
Indicaciones de arco . . . 327
Armónicos . . . 327
Snap (Bartók) pizzicato . . . 328
2.4 Instrumentos de cuerda con trastes . . . 329
2.4.1 Notación común para cuerdas con trastes . . . 330
Referencias para cuerdas con trastes . . . 330
Indicación de los números de cuerda . . . 330
Tablaturas predeterminadas . . . 332
Tablaturas personalizadas . . . 345
Marcas de diagramas de trastes . . . 348
Diagramas predefinidos de trastes . . . 357
Diagramas de traste automáticos . . . 367
Digitaciones de la mano derecha . . . 370
2.4.2 Guitarra . . . 372

Indicar la posición y la cejilla . . . 372
Indicar armónicos y notas tapadas . . . 372
Indicación de acordes de potencia o de quinta vacía . . . 374
2.4.3 Banjo . . . 375
Tablaturas de banjo . . . 375
2.5 Percusión . . . 376
2.5.1 Notación común para percusión . . . 376
Referencias para percusión . . . 376
Notación básica de percusión . . . 376
Redobles . . . 377
Percusión afinada . . . 378
Pautas de percusión . . . 378
Pautas de percusión personalizadas . . . 380
Notas fantasma . . . 384
2.6 Instrumentos de viento . . . 385
2.6.1 Notación común para instrumentos de viento . . . 385
Referencias para instrumentos de viento . . . 385
Digitaciones . . . 387
2.6.2 Gaita . . . 389
Definiciones para la gaita . . . 389
Ejemplo de música de gaita . . . 389
2.6.3 Instrumentos de viento madera . . . 390
2.6.3.1 Diagramas de posiciones para viento madera . . . 391
2.7 Notación de acordes . . . 399
2.7.1 Modo de acordes . . . 399
Panorámica del modo de acordes . . . 399
Acordes más usuales . . . 401
Acordes extendidos y alterados . . . 402
2.7.2 Imprimir los acordes . . . 405
Impresión de los nombres de acorde . . . 405
Nombres de acorde personalizados . . . 407
2.7.3 Bajo cifrado . . . 413
Introducción al bajo cifrado . . . 413
Introducir el bajo cifrado . . . 414
Imprimir el bajo cifrado . . . 417
2.8 Música contemporánea . . . 419
2.8.1 Altura y armonía en la música contemporánea . . . 419
Referencias para alturas y armonía en música contemporánea . . . 419
Notación microtonal . . . 420
Armonía y armaduras de tonalidad contemporáneas . . . 420
2.8.2 Enfoques contemporáneos del ritmo . . . 420
Referencias para enfoque contemporáneos del ritmo . . . 420
Grupos de valoración especial en música contemporánea . . . 420
Compases contemporáneos . . . 420
Notación polimétrica extendida . . . 420
Barrado en música contemporánea . . . 420
Líneas divisorias en música contemporánea . . . 420
2.8.3 Notación gráfica . . . 420
2.8.4 Técnicas de partitura contemporáneas . . . 420
2.8.5 Nuevas técnicas instrumentales . . . 420
2.8.6 Lecturas adicionales y partituras de interés . . . 420
Libros y artículos sobre notación musical contemporánea . . . 420
Partituras y ejemplos musicales . . . 420
2.9 Notación antigua . . . 420

2.9.1 Panorámica de los estilos contemplados ... 422
2.9.2 Notación antigua - funcionalidades comunes ... 422
Contextos predefinidos ... 423
Ligaduras ... 423
Custos ... 424
2.9.3 Tipografiar música mensural ... 424
Contextos de la música mensural ... 424
Claves de la música mensural ... 425
Indicaciones de compás de la música mensural ... 426
Cabezas de nota de la música mensural ... 427
Corchetes de la música mensural ... 428
Silencios de la música mensural ... 429
Alteraciones y armaduras de la música mensural ... 430
Alteraciones de anotación (*musica ficta*) ... 430
Ligaduras mensurales blancas ... 431
2.9.4 Tipografiado del canto gregoriano ... 432
Contextos del canto gregoriano ... 433
Claves de canto gregoriano ... 433
Alteraciones y armaduras de canto gregoriano ... 434
Divisiones ... 435
Articulaciones del canto gregoriano ... 435
Puntos de aumentación (*morae*) ... 436
Ligaduras de neumas cuadrados gregorianos ... 437
2.9.5 Tipografiado del canto kievano en notación cuadrada ... 444
Contextos del canto kievano ... 444
Claves del canto kievano ... 444
Notas del canto kievano ... 445
Alteraciones accidentales del canto kievano ... 445
Líneas divisorias del canto kievano ... 446
Melismas del canto kievano ... 446
2.9.6 Trabajar con música antigua - escenarios y soluciones ... 447
Incipits ... 447
Disposición tipo «Mensurstriche» ... 447
Transcripción de canto gregoriano ... 448
Antigua y moderna desde la misma fuente ... 451
Marcado editorial ... 451
2.10 Músicas del mundo ... 451
2.10.1 Notación común para músicas no occidentales ... 451
Extensión de la notación y los sistemas de afinación ... 451
2.10.2 Música árabe ... 452
Referencias para música árabe ... 452
Nombres de nota árabes ... 453
Armaduras de tonalidad árabes ... 454
Indicaciones de compás árabes ... 456
Ejemplo de música árabe ... 456
Lecturas adicionales sobre música árabe ... 457
2.10.3 Música clásica de Turquía ... 457
Referencias para música clásica de Turquía ... 458
Nombres de las notas en turco ... 458

3 Entrada y salida generales ... 459

3.1 Estructura del código de entrada ... 459
3.1.1 Estructura de una partitura ... 459
3.1.2 Varias partituras en un libro ... 460
3.1.3 Varios archivos de salida a partir de uno de entrada ... 461
3.1.4 Nombres de los archivos de salida ... 462
3.1.5 Estructura del archivo ... 463
3.2 Títulos y encabezamientos ... 465
3.2.1 Crear títulos, encabezamientos y pies de página ... 465
Explicación de los títulos ... 465
Presentación predeterminada de los títulos de partes de libro y partitura ... 469
Disposición predeterminada de las cabeceras y pies de página ... 472
3.2.2 Títulos, encabezamientos y pies de página personalizados ... 473
Formateo personalizado del texto de los bloques de título ... 473
Personalización de los títulos ... 473
Disposición personalizada de cabeceras y pies de página ... 476
3.2.3 Crear notas al pie ... 478
Notas al pie dentro de expresiones musicales ... 478
Notas al pie en texto independiente ... 484
3.2.4 Referencia a números de página ... 487
3.2.5 Índice general ... 488
3.3 Trabajar sobre los archivos de entrada ... 490
3.3.1 Inclusión de archivos de LilyPond ... 490
3.3.2 Distintas ediciones a partir de una misma fuente ... 492
Uso de las variables ... 492
Uso de etiquetas ... 494
Uso de ajustes globales ... 497
3.3.3 Caracteres especiales ... 497
Codificación del texto ... 497
Unicode ... 498
Alias de ASCII ... 499
3.4 Controlar la salida ... 500
3.4.1 Extracción de fragmentos de música ... 500
3.4.2 Saltar la música corregida ... 500
3.4.3 Formatos de salida alternativos ... 501
3.4.4 Sustituir la tipografía de la notación ... 501
3.5 Salida MIDI ... 502
3.5.1 Crear archivos MIDI ... 502
3.5.2 Instrumentos MIDI ... 504
3.5.3 ¿Qué sale por el MIDI? ... 505
Contemplado en el MIDI ... 505
No contemplado en el MIDI ... 505
3.5.4 Repeticiones y MIDI ... 505
3.5.5 Control de los matices en el MIDI ... 506
Indicaciones dinámicas ... 506
Volumen maestro MIDI ... 507
Balance entre instrumentos (i) ... 508
Balance entre instrumentos (ii) ... 509
3.5.6 Percusión en MIDI ... 510
3.5.7 El script Articulate ... 511
3.6 Extraer información musical ... 511
3.6.1 Displaying LilyPond notation ... 511
3.6.2 Impresión de las expresiones musicales de Scheme ... 512
3.6.3 Guardar los eventos musicales en un archivo ... 512

4 Problemas de espaciado 513

4.1 Disposición de la página 513
4.1.1 El bloque `\paper` 513
4.1.2 Tamaño del papel y escalado automático 514
Fijar el tamaño del papel 514
Escalado automático al tamaño del papel 515
4.1.3 Variables de espaciado de `\paper` verticales fijas 516
4.1.4 Variables de espaciado de `\paper` verticales flexibles 517
Estructura de las listas-A de espaciado vertical flexible 517
Lista de variables de espaciado de `\paper` verticales flexibles 518
4.1.5 Variables de espaciado de `\paper` horizontales 519
Variables de `\paper` para la anchura y los márgenes 519
Variables de `\paper` para el modo de doble cara 520
Variables de `\paper` para desplazamientos y sangrados 521
4.1.6 Otras variables de `\paper` 521
Variables de `\paper` para los saltos de línea 521
Variables de `\paper` para los saltos de página 522
Variables de `\paper` para la numeración de las páginas 523
Variables de `\paper` diversas 523
4.2 Disposición de la partitura 524
4.2.1 El bloque `\layout` 524
4.2.2 Establecer el tamaño del pentagrama 526
4.3 Saltos 527
4.3.1 Saltos de línea 527
4.3.2 Saltos de página 530
4.3.3 Saltos de página óptimos 531
4.3.4 Paso de página óptimo 531
4.3.5 Saltos de página mínimos 532
4.3.6 Saltos de página de una línea 532
4.3.7 Saltos de línea explícitos 532
4.3.8 Utilizar una voz adicional para los saltos de línea 534
4.4 Espaciado vertical 536
4.4.1 Espaciado vertical flexible dentro de los sistemas 536
Propiedades de espaciado dentro de los sistemas 536
Espaciado de pautas no agrupadas 540
Espaciado de pautas agrupadas 541
Espaciado de las líneas que no son pautas 542
4.4.2 Posicionamiento explícito de los pentagramas y los sistemas 543
4.4.3 Evitar las colisiones verticales 551
4.5 Espaciado horizontal 552
4.5.1 Panorámica del espaciado horizontal 552
4.5.2 Área de espaciado nueva 554
4.5.3 Cambiar el espaciado horizontal 555
4.5.4 Longitud de la línea 557
4.5.5 Notación proporcional 557
4.6 Encajar la música en menos páginas 564
4.6.1 Mostrar el espaciado 564
4.6.2 Cambiar el espaciado 565

5 Cambiar los valores por omisión ... 567

5.1 Contextos de interpretación ... 567
5.1.1 Explicación de los contextos ... 567
Definiciones de salida - estructura de los contextos ... 567
Score. El contexto maestro ... 568
Contextos del nivel superior: contenedores de pentagramas ... 568
Contextos de nivel intermedio. Pentagramas ... 568
Contextos del nivel más bajo. Voces ... 569
5.1.2 Crear y referenciar contextos ... 569
5.1.3 Mantener vivos los contextos ... 573
5.1.4 Modificar los complementos (plug-ins) de contexto ... 575
5.1.5 Cambiar los valores por omisión de los contextos ... 577
Cambiar todos los contextos del mismo tipo ... 578
Cambiar solamente un contexto determinado ... 580
Orden de precedencia ... 582
5.1.6 Definir contextos nuevos ... 582
5.1.7 Orden de disposición de los contextos ... 584
5.2 Explicación del Manual de referencia de funcionamiento interno ... 587
5.2.1 Navegar por la referencia del programa ... 587
5.2.2 Interfaces de la presentación ... 588
5.2.3 Determinar la propiedad del grob ... 589
5.2.4 Convenciones de nombres ... 590
5.3 Modificar las propiedades ... 590
5.3.1 Panorámica de la modificación de las propiedades ... 590
5.3.2 La instrucción `\set` ... 590
5.3.3 La instrucción `\override` ... 593
5.3.4 La instrucción `\tweak` ... 595
5.3.5 `\set` frente a `\override` ... 597
5.3.6 Modificación de las listas-A ... 597
5.4 Conceptos y propiedades útiles ... 599
5.4.1 Modos de entrada ... 599
5.4.2 Dirección y posición ... 601
Indicadores de dirección de las articulaciones ... 601
La propiedad de dirección ... 601
5.4.3 Distancias y medidas ... 602
5.4.4 Propiedades del símbolo del pentagrama ... 603
5.4.5 Objetos de extensión ... 603
Uso del `spanner-interface` ... 603
Uso del `line-spanner-interface` ... 606
5.4.6 Visibilidad de los objetos ... 608
Eliminar el sello ... 608
Hacer transparentes los objetos ... 609
Pintar los objetos de blanco ... 609
Uso de break-visibility (visibilidad en el salto) ... 610
Consideraciones especiales ... 611
5.4.7 Estilos de línea ... 614
5.4.8 Rotación de objetos ... 615
Rotación de objetos de presentación ... 615
Rotación de elementos de marcado ... 616
5.5 Trucos avanzados ... 616
5.5.1 Alineación de objetos ... 616
Establecer directamente `X-offset` y `Y-offset` ... 617
Uso del `side-position-interface` ... 617
Uso del `self-alignment-interface` ... 618

Uso del `break-alignable-interface` 619
5.5.2 Agrupación vertical de objetos gráficos 621
5.5.3 Modificación de los sellos 621
5.5.4 Modificación de las formas 622
Modificación de ligaduras de unión y de expresión 622
5.5.5 Modificación de objetos de extensión divididos 626
Uso de `\alterBroken` 626
5.5.6 Contenedores unpure-pure 628
5.6 Uso de las funciones musicales 629
5.6.1 Sintaxis de las funciones de sustitución 629
5.6.2 Ejemplos de funciones de sustitución 630

Apéndice A Tablas del manual sobre notación 633

A.1 Cuadro de nombres de acordes 633
A.2 Modificadores de acorde más usuales 634
A.3 Afinaciones de cuerdas predefinidas 637
A.4 Diagramas predefinidos de posiciones 639
Diagramas para guitarra 639
Diagramas para ukelele 641
Diagramas para mandolina 642
A.5 Tamaños de página predefinidos 644
A.6 instrumentos MIDI 648
A.7 Lista de colores 648
A.8 La tipografía Feta 650
Glifos de clave 650
Glifos de indicación de compás 650
Glifos de cifras 651
Glifos de alteraciones 651
Glifos de las cabezas de nota predeterminadas 652
Glifos de las cabezas de nota especiales 652
Glifos de las cabezas de nota con formas 653
Glifos de silencios 657
Glifos de corchetes 658
Glifos de puntillos 658
Glifos de matices dinámicos 658
Grifos de inscripciones 659
Glifos de flechas 661
Glifos de puntas de corchete 661
Glifos de pedal 661
Glifos de acordeón 662
Glifos de ligadura 662
Glifos de vaticana 662
Glifos de medicea 663
Glifos de Hufnagel 664
Glifos de mensural 664
Glifos de neomensural 668
Glifos de Petrucci 669
Glifos de Solesmes 670
Glifos de la notación del canto kievano 670
A.9 Estilos de cabezas de nota 671
A.10 Instrucciones de marcado de texto 671
A.10.1 Font 671
A.10.2 Align 680
A.10.3 Graphic 695

A.10.4 Music ... 702
A.10.5 Instrument Specific Markup ... 708
A.10.6 Accordion Registers ... 711
A.10.7 Other ... 716
A.11 Instrucciones de listas de marcado de texto ... 722
A.12 Lista de caracteres especiales ... 723
A.13 Lista de articulaciones ... 725
Indicaciones de articulación ... 725
Indicaciones de adornos ... 725
Indicaciones de calderón ... 725
Indicaciones específicas de ciertos instrumentos ... 726
Indicaciones de repetición ... 726
Indicaciones antiguas ... 726
A.14 Notas de percusión ... 726
A.15 Glosario técnico ... 728
alist (lista-A) ... 728
callback ... 728
closure (cerradura) ... 728
glifo ... 728
grob (objeto gráfico) ... 728
inmutable ... 729
interfaz ... 729
lexer (analizador léxico) ... 729
mutable ... 729
output-def (definición de salida) ... 730
parser (analizador sintáctico) ... 730
variable del analizador sintáctico ... 730
prob (objeto de propiedades) ... 730
cerradura simple ... 730
smob (objeto de Scheme) ... 731
stencil (sello) ... 731
A.16 Todas las propiedades de contexto ... 731
A.17 Propiedades de disposición ... 743
A.18 Funciones musicales disponibles ... 762
A.19 Identificadores de modificación de contextos ... 771
A.20 Predicados de tipo predefinidos ... 771
R5RS primary predicates ... 771
R5RS secondary predicates ... 771
Guile predicates ... 772
LilyPond scheme predicates ... 772
LilyPond exported predicates ... 772
A.21 Funciones de Scheme ... 773

Apéndice B Hoja de referencia rápida ... 797

Apéndice C GNU Free Documentation License ... 801

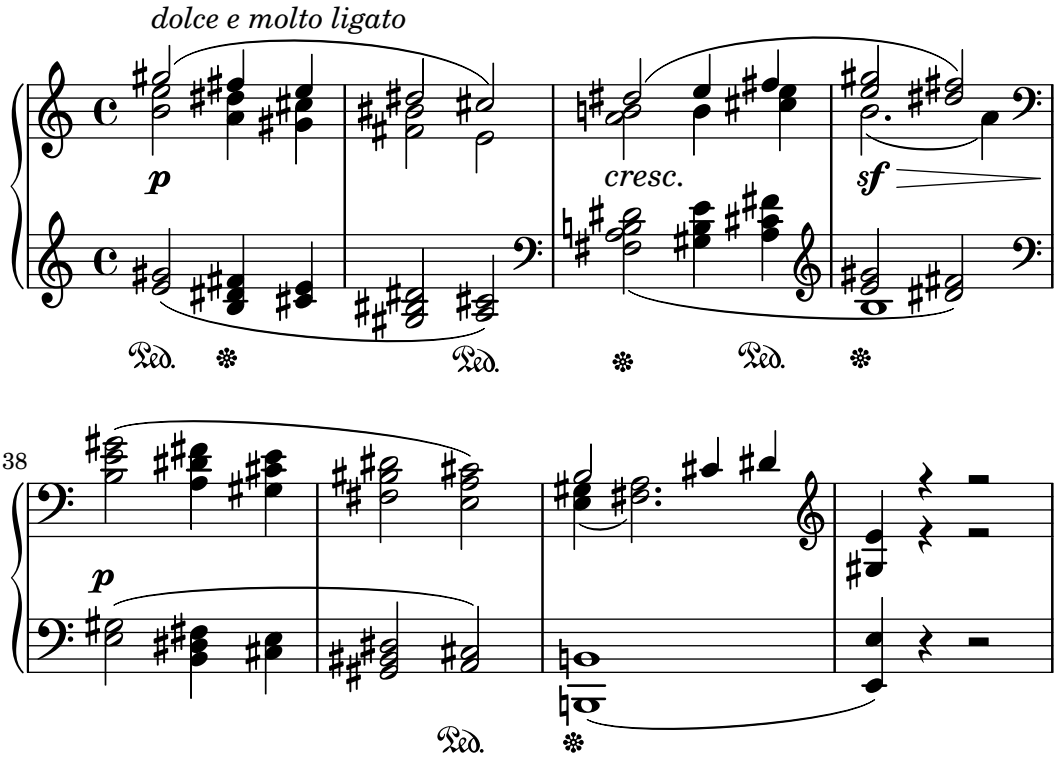
Apéndice D Índice de instrucciones de LilyPond ... 808

Apéndice E Índice de LilyPond ... 818

1 Notación musical

Este capítulo explica cómo crear notación musical.

1.1 Alturas



En esta sección se discute cómo especificar la altura de las notas. Este proceso se compone de tres fases: entrada, modificación y salida.
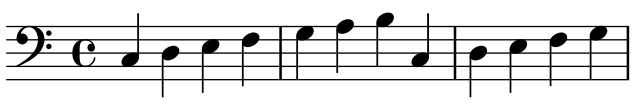
1.1.1 Escritura de notas

En esta sección se describe la manera de introducir la altura de las notas. Existen dos formas distintas de colocar las notas en su octava correspondiente: el modo absoluto y el relativo. En casi todas las ocasiones, será más práctico el modo relativo.

Escritura de octava absoluta
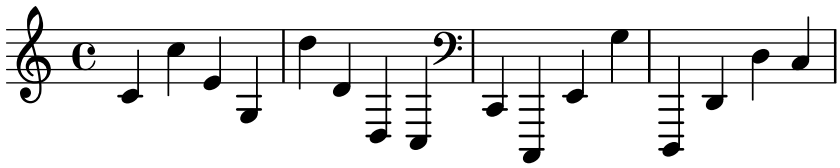
El nombre de una nota se especifica usando las letras minúsculas de la `a` a la `g`. Las notas cuyos nombres van desde `c` hasta `b` se imprimen en la octava inferior al Do central.

```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```



Se pueden especificar otras octavas mediante una comilla simple quote (`'`) o una coma (`,`) . Cada `'` eleva la altura en una octava; cada `,` baja la altura una octava.

```
{
  \clef treble
  c'4 c'' e' g
  d''4 d' d c
  \clef bass
  c,4 c,, e, g
  d,,4 d, d c
}
```



Se puede indicar explícitamente que la música está en notación de octava absoluta si se precede con `\absolute`:

`\absolute` *musicexpr*

se interpreta en el modo de entrada de octava absoluta independientemende de en qué contexto aparece.

Véase también

Glosario musical: Sección “Pitch names” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Escritura de octava relativa

La entrada de octava absoluta requere que se especifique la octava para todas y cada una de las notas. En contraste con ello, el modo de entrada de octava relativa especifica cada octava en relación a la nota anterior: si se cambia la octava de una nota ello afectará a todas las notas siguientes.

El modo relativo de notas se debe introducir de forma explícita usando la instrucción `\relative`:

`\relative` *altura_inicial expresión_musical*

En el modo relativo, se supone que cada nota se encuentra lo más cerca posible de la nota anterior. Esto significa que la octava de una nota que está dentro de *expresión_musical* se calcula como sigue:

- Si no se usa ninguna marca de cambio de octava en una nota, su octava se calcula de forma que el intervalo que forme con la nota anterior sea menor de una quinta. Este intervalo se determina sin considerar las alteraciones.
- Se puede añadir una marca de cambio de octava `'` o `,` para elevar o bajar la altura, respectivamente, en una octava más en relación con la altura calculada sin esta marca.
- Se pueden usar varias marcas de cambio de octava. Por ejemplo, `''` y `,,` alteran la altura en dos octavas.
- La altura de la primera nota es relativa a *altura_inicial*. *altura_inicial* se especifica en modo de octava absoluta. ¿Qué opciones tienen sentido?

  `c` (Do), en cualquier octava

  > La identificación del Do central con `c'` es algo bastante básico, por lo que tiende a ser fácil encontrar octavas de `c`. Si nuestra música comienza con `gis` (un Sol sostenido) por encima de `c'''`, tendríamos que escribir algo como `\relative c''' { gis' ... }`
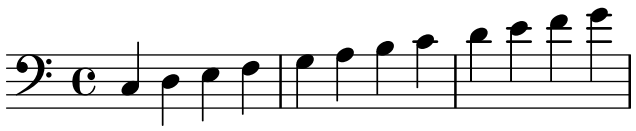
una nota que está una o más octavas de la primera nota de dentro
: Escribir `\relative gis''' { gis ... }` hace que sea fácil determinar la altura absoluta de la primera nota de dentro.

ninguna altura de inicio explícita
: Esta opción (concretamente, escribir `\relative { gis''' ... }`) se puede ver como una versión más compacta de la opción anterior: la primera nota de dentro se escribe ella misma en altura absoluta. Esto resulta ser equivalente a elegir `f` (Fa) como la altura de referencia.

La documentación suele utilizar la primera opción.
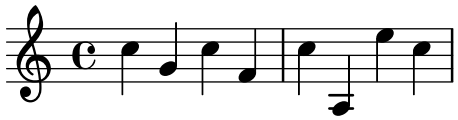
Aquí podemos ver el modo relativo en acción:

```
\relative c {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```
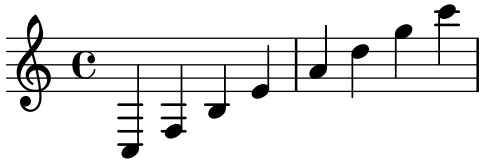


Las marcas de cambio de octava se utilizan para intervalos mayores de la cuarta:

```
\relative c'' {
  c g c f,
  c' a, e'' c
}
```



Una serie de notas sin ninguna marca de octava puede, a pesar de todo, abarcar intervalos muy grandes:
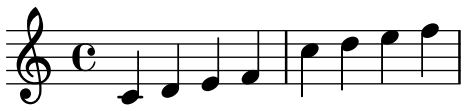
```
\relative c {
  c f b e
  a d g c
}
```



Cuando hay unos bloques `\relative` anidados dentro de otros, es de aplicación el bloque `\relative` más interno.

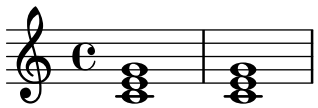
```
\relative c' {
  c d e f
  \relative c'' {
    c d e f
```

```
  }
}
```



`\relative` no tiene efecto sobre los bloques `\chordmode`.
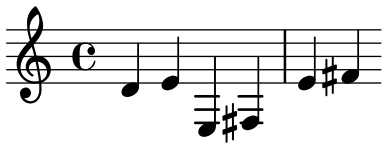
```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```



`\relative` no se permite dentro de los bloques `\chordmode`.

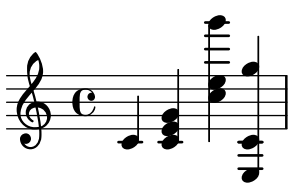
La música que esa dentro de un bloque `\transpose` es absoluta, a no ser que se incluya una instrucción `\relative`.

```
\relative c' {
  d e
  \transpose f g {
    d e
    \relative c' {
      d e
    }
  }
}
```
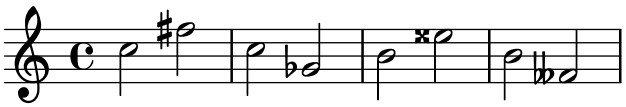


Si el elemento anterior es un acorde, la primera nota del acorde se utiliza para determinar la primera nota del siguiente acorde. Dentro de los acordes, la siguiente nota siempre está en relación a la anterior.

```
\relative c' {
  c
  <c e g>
  <c' e g'>
  <c, e, g''>
}
```



Como se explicó más arriba, la octava de las notas se calcula solamente a partir de sus nombres, sin tener en cuenta alteración alguna. Por tanto, un Mi doble sostenido después de un Si se escribirá más agudo, mientras que un Fa doble sostenido se escribirá más grave. En otras palabras, se considera a la cuarta doble aumentada un intervalo menor que la quinta doble disminuida, independientemente del número de semitonos de cada uno de ellos.

```
\relative c'' {
  c2 fis
  c2 ges
  b2 eisis
  b2 feses
}
```



Es una consecuencia de las reglas expuestas que la primera nota dentro de una expresión musical precedida de `\relative f` se interpreta igual que si estuviera escrita en el modo de altura absoluta.

Véase también

Glosario musical: Sección “fifth” in *Glosario Musical*, Sección “interval” in *Glosario Musical*, Sección “Pitch names” in *Glosario Musical*.

Referencia de la notación: [Comprobación de octava], página 9.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.
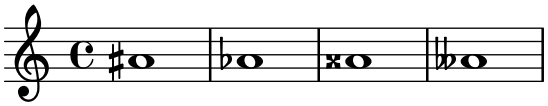
Referencia de funcionamiento interno: Sección “RelativeOctaveMusic” in *Referencia de Funcionamiento Interno*.

Alteraciones accidentales

> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Alteraciones accidentales y armaduras” in *Manual de Aprendizaje*.
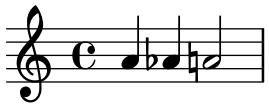
Se escribe una nota *sostenida* añadiendo `is` al nombre de la nota, y un *bemol* añadiendo `es`. Como es de esperar, un *dobles sostenido* y un *doble bemol* se obtiene añadiendo `isis` o `eses`. Esta sintaxis deriva de los nombres de las notas en holandés. Para utilizar otros nombres para las alteraciones, consulte [Nombres de las notas en otros idiomas], página 7.

```
ais1 aes aisis aeses
```



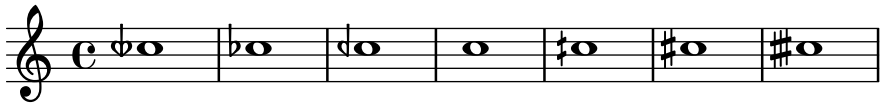
Un becuadro cancelará el efecto de una alteración accidental o de la armadura. Sin embargo, los becuadros no se codifican dentro de la sintaxis del nombre de la nota con un sufijo; una nota becuadro se muestra como un simple nombre de nota natural:

```
a4 aes a2
```
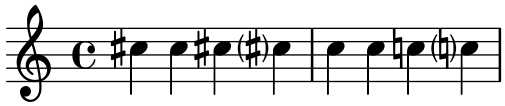


Se pueden escribir medios bemoles y los medios sostenidos; a continuación presentamos una serie de DOs cada vez más agudos:

```
ceseh1 ces ceh c cih cis cisih
```



Normalmente las alteraciones accidentales se imprimen automáticamente, pero también puede imprimirlas manualmente. Un alteración recordatoria se puede forzar añadiendo un signo de admiración `!` después de la altura de la nota. Se puede obtener una alteración de precaución (o sea, una alteración entre paréntesis) añadiendo el signo de interrogación `?` después del nombre de la nota. Estas alteraciones adicionales se pueden usar también para producir notas con becuadro.
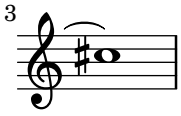
```
cis cis cis! cis? c c c! c?
```



Las alteraciones sobre notas unidas por ligadura sólo se imprimen al comienzo de un sistema:

```
cis1~ cis~
\break
cis
```



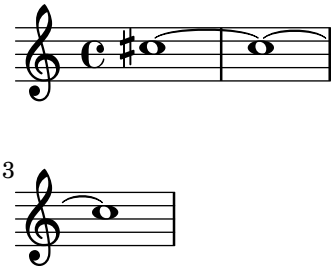


Fragmentos de código seleccionados

Ocultar las alteraciones sobre notas ligadas al principio de un sistema nuevo

Aquí se muestra la manera de ocultar las alteraciones de las notas ligadas al comienzo de un sistema nuevo.
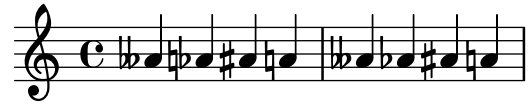
```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```



Evitar que se añadan becuadros adicionales automáticamente

Según las reglas tradicionales de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar un doble sostenido o un doble bemol anterior en la misma nota. Para modificar este comportamiento a la práctica actual, establezca el valor de la propiedad `extraNatural` a `##f` (falso) dentro del contexto de `Staff`.

```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Véase también

Glosario musical: Sección “sharp” in *Glosario Musical*, Sección “flat” in *Glosario Musical*, Sección “double sharp” in *Glosario Musical*, Sección “double flat” in *Glosario Musical*, Sección “Pitch names” in *Glosario Musical*, Sección “quarter tone” in *Glosario Musical*.

Manual de aprendizaje: Sección “Alteraciones accidentales y armaduras” in *Manual de Aprendizaje*.

Referencia de la notación: [Alteraciones accidentales automáticas], página 26, [Alteraciones de anotación (musica ficta)], página 430, [Nombres de las notas en otros idiomas], página 7.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental_engraver” in *Referencia de Funcionamiento Interno*, Sección “Accidental” in *Referencia de Funcionamiento Interno*, Sección “AccidentalCautionary” in *Referencia de Funcionamiento Interno*, Sección “accidental-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos
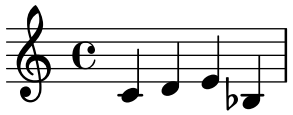
No existen estándares reconocidos ampliamente para denotar los bemoles de tres cuartos, de manera que los símbolos de LilyPond no se ajustan a ningún estándar.

Nombres de las notas en otros idiomas

Existen conjuntos predefinidos de nombres de notas y sus alteraciones para algunos idiomas aparte del inglés. La selección del idioma de los nombres de las notas se suele hacer al principio del archivo; el ejemlo siguiente está escrito utilizando los nombres italianos de las notas:

```
\language "italiano"

\relative do' {
  do re mi sib
}
```



Los idiomas disponibles y los nombres de las notas que definen son los siguientes:
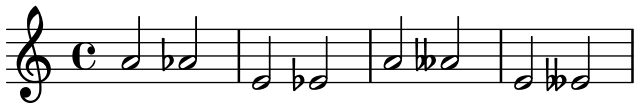
Idioma	**Nombre de las notas**
`nederlands`	c d e f g a bes b
`catalan`	do re mi fa sol la sib si
`deutsch`	c d e f g a b h
`english`	c d e f g a bf b
`espanol` o `español`	do re mi fa sol la sib si
`italiano` o `français`	do re mi fa sol la sib si
`norsk`	c d e f g a b h
`portugues`	do re mi fa sol la sib si
`suomi`	c d e f g a b h
`svenska`	c d e f g a b h
`vlaams`	do re mi fa sol la sib si

Además de los nombres de las notas, los sufijos de las alteraciones pueden también variar dependiendo del idioma:

Idioma	**sostenido**	**bemol**	**doble sostenido**	**doble bemol**
`nederlands`	-is	-es	-isis	-eses
`catalan`	-d/-s	-b	-dd/-ss	-bb
`deutsch`	-is	-es	-isis	-eses
`english`	-s/-sharp	-f/-flat	-ss/-x/-sharpsharp	-ff/-flatflat
`espanol` o `español`	-s	-b	-ss/-x	-bb
`italiano` o `français`	-d	-b	-dd	-bb
`norsk`	-iss/-is	-ess/-es	-ississ/-isis	-essess/-eses
`portugues`	-s	-b	-ss	-bb
`suomi`	-is	-es	-isis	-eses
`svenska`	-iss	-ess	-ississ	-essess
`vlaams`	-k	-b	-kk	-bb

En holandés, `aes` se contrae como `as`, pero las dos formas se aceptan en LilyPond. De forma similar, se aceptan tanto `es` como `ees`. Esto se aplica también a `aeses` / `ases` y a `eeses` / `eses`. A veces se definen solamente estos nombres contraídos en los archivos de idioma correspondientes.

```
a2 as e es a ases e eses
```



Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol 'normales'. La tabla siguiente relacione los nombres de las notas para las alteraciones de un cuarto de tono en distintos idiomas; aquí, los prefijos *semi-* y *sesqui-* respectivamente significan 'medio' y 'uno y medio'. Los idiomas que no aparecen en esta tabla no aportan aún nombres especiales para las notas.

Idioma	semi-sostenido	semi-bemol	sesqui-sostenido	sesqui-bemol
`nederlands`	-ih	-eh	-isih	-eseh
`deutsch`	-ih	-eh	-isih	-eseh
`english`	-qs	-qf	-tqs	-tqf
`espanol`	-cs	-cb	-tcs	-tcb
`italiano`	-sd	-sb	-dsd	-bsb
`portugues`	-sqt	-bqt	-stqt	-btqt

Casi todos los idiomas presentados aquí están asociados comúnmente con la música clásica occidental, también conocida como *Período de la práctica común*. Sin embargo, también están contempladas las alturas y los sistemas de afinación alternativos: véase Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

Véase también

Glosario musical: Sección “Pitch names” in *Glosario Musical*, Sección “Common Practice Period” in *Glosario Musical*.

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

Archivos instalados: ‘`scm/define-note-names.scm`’.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

1.1.2 Modificación de varias notas a la vez
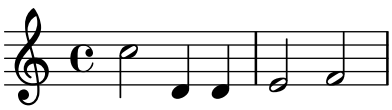
Esta sección trata de la manera de modificar las alturas.

Comprobación de octava

En el modo relativo, es fácil olvidar una marca de cambio de octava. Las comprobaciones de octava hacen más fácil encontrar estos errores, mediante la presentación de una advertencia y corrigiendo la octava si la nota se encuentra en una octava distinta de lo esperado.

Para comprobar la octava de una nota, especifique la octava absoluta después del símbolo `=`. Este ejemplo genera un mensaje de advertencia (y corrige la altura) porque la segunda nota es la octava absoluta `d''` en lugar de `d'` como indica la corrección de octava.
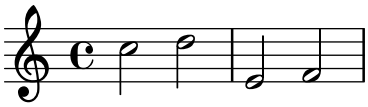
```
\relative c'' {
  c2 d='4 d
  e2 f
}
```



La octava de las notas se puede comprobar también con la instrucción `\octaveCheck` *`altura_de_control`*. *`altura_de_control`* se especifica en modo absoluto. Esto comprueba que el intervalo entre la nota anterior y la *`altura_de_control`* se encuentra dentro de una cuarta (es decir, el cálculo normal para el modo relativo). Si esta comprobación fracasa, se imprime un mensaje de advertencia, pero la nota previa no se modifica. Las notas posteriores están en relación a la *`altura_de_control`*.

```
\relative c'' {
  c2 d
  \octaveCheck c'
  e2 f
```
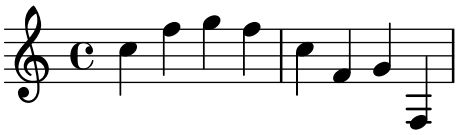
```
}
```



Compare los dos compases siguientes. La primera y tercera comprobaciones de `\octaveCheck` fracasan, pero la segunda es correcta.

```
\relative c'' {
  c4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveCheck” in *Referencia de Funcionamiento Interno*.

Transposición

Una expresión musical se puede transportar mediante `\transpose`. La sintaxis es
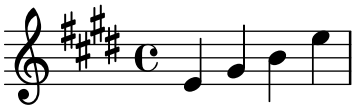
```
\transpose nota_origen nota_destino expresión_musical
```

Esto significa que la *`expresión_musical`* se transporta el intervalo que hay entre las notas *`nota_origen`* y *`nota_destino`*: cualquier nota con la altura de *`nota_origen`* se cambia por *`nota_destino`* y cualquier otra nota se transporta el mismo intervalo. Las dos notas se introducen en modo absoluto.

> **Nota:** La música que está dentro de un bloque `\transpose` es absoluta, a no ser que se incluya una instrucción `\relative` dentro del bloque.
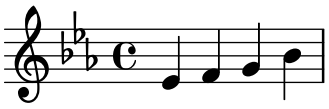
Por ejemplo, tomemos una pieza escrita en la tonalidad de Re mayor. Se puede transportar hacia arriba a Mi mayor; observe que también la armadura de la tonalidad se transporta automáticamente.

```
\transpose d e {
  \relative c' {
    \key d \major
    d4 fis a d
  }
}
```



Si una particella escrita en Do (*afinación de concierto* normal) se debe tocar con un clarinete en La (para el que un La se escribe como un Do, y que suena una tercera menor por debajo de lo que está escrito), la particella correspondiente se produce mediante:
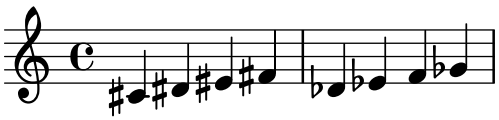
```
\transpose a c' {
  \relative c' {
    \key c \major
    c4 d e g
  }
}
```



Observe que especificamos `\key c \major` de forma explícita. Si no especificamos ninguna tonalidad, las notas se transportan pero no se imprime la armadura.

`\transpose` distingue entre notas enarmónicas: tanto `\transpose c cis` como `\transpose c des` transportan un semitono hacia arriba. La primera versión imprime sostenidos y las notas no se mueven de su lugar en la escala, en cambio la segunda imprime bemoles de la nota siguiente.

```
music = \relative c' { c d e f }
\new Staff {
  \transpose c cis { \music }
  \transpose c des { \music }
}
```



`\transpose` también se puede usar para introducir notas escritas para un instrumento transpositor. Los ejemplos anteriores muestran cómo escribir alturas en Do (o en *afinación de concierto*) y tipografiarlas para un instrumento transpositor, pero también es posible el caso contrario si, por ejemplo, tenemos un conjunto de partes instrumentales y quiere hacer un guión en Do para el director. Por ejemplo, al introducir música para trompeta en Si bemol que comienza por un Mi en la partitura (Re de concierto), se puede escribir:

```
musicaEnSiBemol = { e4 ... }
\transpose c bes, \musicaEnSiBemol
```

Para imprimir esta música en Fa (por ejemplo, al arreglarla para trompa) puede envolver la música existente con otro `\transpose`:

```
musicaEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musicaEnSiBemol }
```

Para ver más información sobre instrumentos transpositores, consulte [Transposición de los instrumentos], página 24.

Fragmentos de código seleccionados

Transportar música con el menor número de alteraciones

Este ejemplo utiliza código de Scheme para forzar las modificaciones enarmónicas de las notas, y así tener el menor número de alteraciones accidentales. En este caso se aplican las siguientes reglas:

- Se quitan las dobles alteraciones
- Si sostenido -> Do
- Mi sistenido -> Fa
- Do bemol -> Si
- Fa bemol -> Mi

De esta forma se selecciona el mayor número de notas enarmónicas naturales.
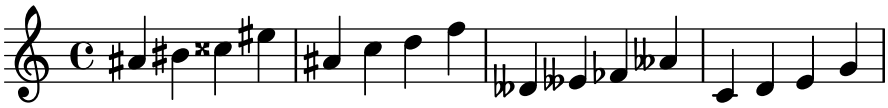
```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         (a (* 4 (ly:pitch-alteration p)))
         ;; alteration, a, in quarter tone steps,
         ;; for historical reasons
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1) (or (eq? n 6) (eq? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eq? n 0) (eq? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
      ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
     (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
     (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
     (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (if (pair? es)
         (ly:music-set-property!
          music 'elements
          (map (lambda (x) (naturalize x)) es)))
     (if (ly:music? e)
         (ly:music-set-property!
          music 'element
          (naturalize e)))
     (if (ly:pitch? p)
         (begin
           (set! p (naturalize-pitch p))
           (ly:music-set-property! music 'pitch p)))
     music))

naturalizeMusic =
#(define-music-function (parser location m)
   (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }
```

```
\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
}
```



Véase también

Referencia de la notación: [Transposición de los instrumentos], página 24, [Inversión], página 13, [Transformaciones modales], página 14, [Escritura de octava relativa], página 2, [Retrogradación], página 14.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TransposedMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` ni `\relative` dentro de su argumento. Para usar el modo relativo dentro de música transportada, se debe colocar otro `\relative` dentro de `\transpose`.

No se imprimen alteraciones accidentales triples cuando se usa `\transpose`. En lugar de ello, se usará una nota ‘equivalente enarmónicamente’ (p.ej., Re bemol en lugar de Mi triple bemol).
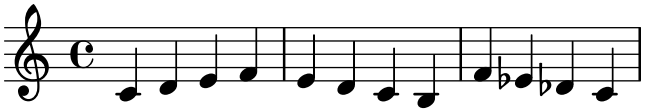
Inversión

Una expresión musical se puede invertir y transportar en una sola operación con:

```
\inversion nota-pivote nota-destino expresión_musical
```

La *`expresión_musical`* se invierte intervalo a intervalo alrededor de *`nota-pivote`*, y después se transporta de manera que *`nota-pivote`* se hace corresponder con *`nota-destino`*.

```
music = \relative c' { c d e f }
\new Staff {
  \music
  \inversion d' d' \music
  \inversion d' ees' \music
}
```



Nota: Los motivos a invertir deberían venir expresados en modo absoluto o convertirse previamente a la forma absoluta encerrándolos en un bloque `\relative`.

Véase también
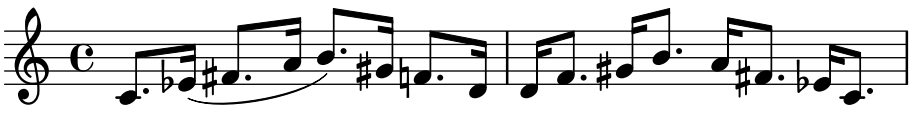
Referencia de la notación: [Transformaciones modales], página 14, [Retrogradación], página 14, [Transposición], página 10.

Retrogradación

Se puede revertir una expresión musical para producir su retrogradación:

```
music = \relative c' { c8. ees16( fis8. a16 b8.) gis16 f8. d16 }

\new Staff {
  \music
  \retrograde \music
}
```



Advertencias y problemas conocidos

Las ligaduras de unión manuales dentro de `\retrograde` se reomperán y generarán advertencias. Se pueden generar algunas ligaduras automáticamente activando [División automática de las notas], página 75.

Véase también

Referencia de la notación: [Inversión], página 13, [Transformaciones modales], página 14, [Transposición], página 10.

Transformaciones modales

En una composición musical que está basada en una escala, con frecuencia se transforman los motivos de diversas formas. Un motivo se puede *transportar* para que comience en distintos lugares de la escala o puede *invertirse* alrededor de una nota pivote de la escala. También se puede revertir para producir una *retrogradación*, véase [Retrogradación], página 14.

> **Nota:** Cualquier nota que no entre en la escada dada, se dejará sin transformar.

Transposición modal
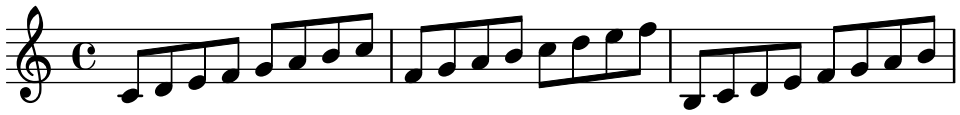
Se puede transportar un motivo dentro de una escala dada con:

```
\modalTranspose nota-origen nota-destino escala motivo
```

Las notas del *motivo* se desplazan dentro de la *escala* el número de grados de la escala dados por el intervalo entre *nota-destino* y *nota-origen*:

```
diatonicScale = \relative c' { c d e f g a b }
motif = \relative c' { c8 d e f g a b c }
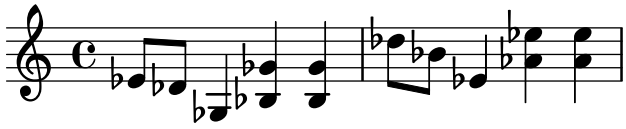
\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \diatonicScale \motif
}
```



Se pueden especificar escalas ascendentes de cualquier longitud y con cualesquiera intervalos:

```
pentatonicScale = \relative c' { ges aes bes des ees }
motif = \relative c' { ees8 des ges,4 <ges' bes,> <ges bes,> }
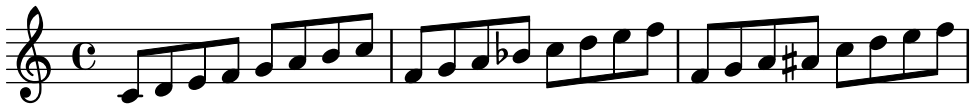
\new Staff {
  \motif
  \modalTranspose ges ees' \pentatonicScale \motif
}
```



Cuando se utiliza con una escala cromática, `\modalTranspose` tiene un efecto similar a `\transpose`, pero con la posibilidad de especificar los nombres de las notas que se quieren usar:

```
chromaticScale = \relative c' { c cis d dis e f fis g gis a ais b }
motif = \relative c' { c8 d e f g a b c }

\new Staff {
  \motif
  \transpose c f \motif
  \modalTranspose c f \chromaticScale \motif
}
```



Inversión modal

Se puede invertir un motivo dentro de una escala dada alrededor de una nota pivote dada y transportada al mismo tiempo en una única operación, con:
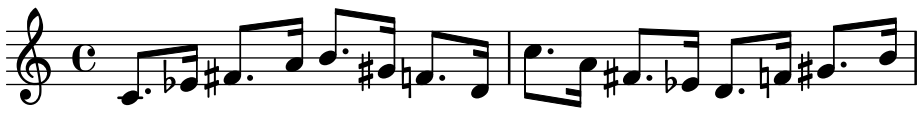
`\modalInversion` *`nota-pivote nota-destino escala motivo`*

Las notas del *motivo* se colocan al mismo número de grados de distancia de la escala a partir de la *nota-pivote* dentro de la *escala*, pero en la dirección opuestoa, y el resultado se desplaza después dentro de la *escala* el número de grados de la escala dados por el intervalo entre la *nota-destino* y la *nota-pivote*.

Así pues, para invertir sencillamente alrededor de una nota de la escala, utilice el mismo valor para *nota-pivote* y *nota-destino*:

```
octatonicScale = \relative c' { ees f fis gis a b c d }
motif = \relative c' { c8. ees16 fis8. a16 b8. gis16 f8. d16 }
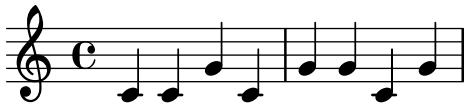
\new Staff {
  \motif
  \modalInversion fis' fis' \octatonicScale \motif
}
```



Para invertir alrededor de una nota pivote entre dos notas de la escala, invierta alrededor de una de las notas y después transporte en un grado de la escala. Las dos notas especificadas se pueden interpretar como que horquillan entre ellas a la nota pivote:

```
scale = \relative c' { c g' }
motive = \relative c' { c c g' c, }
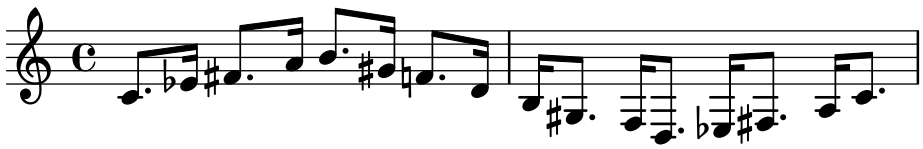
\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}
```



La operación combinada de inversión y retrogradación produce la inversión retrógrada:

```
octatonicScale = \relative c' { ees f fis gis a b c d }
motif = \relative c' { c8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \retrograde \modalInversion c' c' \octatonicScale \motif
}
```



Véase también

Referencia de la notación: [Inversión], página 13, [Retrogradación], página 14, [Transposición], página 10.
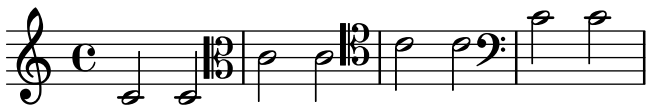
1.1.3 Imprimir las alturas

Esta sección trata de cómo alterar la presentación de la altura de las notas.

Clave

Es posible cambiar la clave. En todos los ejemplos se muestra el Do central. Estos nombres de clave pueden (aunque no necesariamente) encerrarse entre comillas.

```
\clef treble
c2 c
\clef alto
c2 c
\clef tenor
c2 c
\clef bass
c2 c
```



Otras claves son las siguientes:

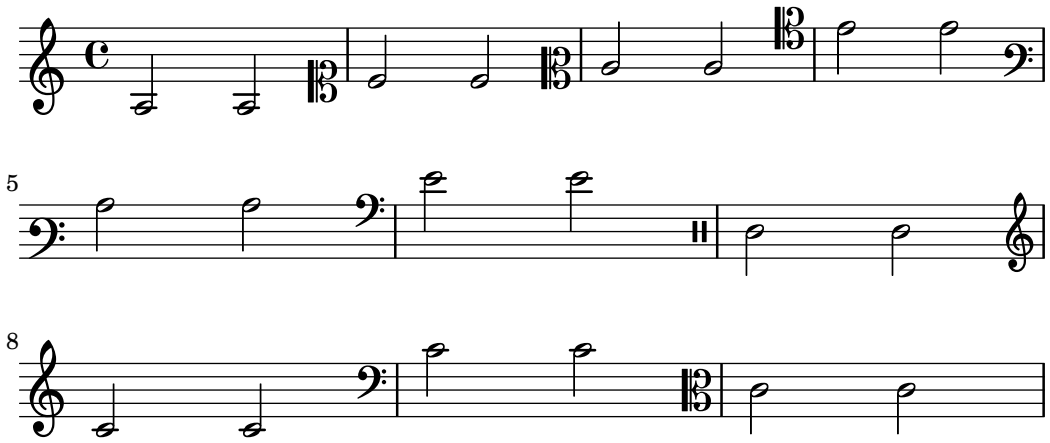
```
\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c

\break

\clef varbaritone
c2 c
\clef subbass
c2 c
\clef percussion
c2 c

\break
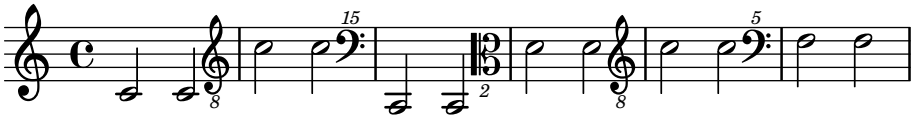
\clef G   % synonym for treble
c2 c
\clef F   % synonym for bass
c2 c
\clef C   % synonym for alto
c2 c
```



Al añadir `_8` o `^8` al nombre de la clave, la clave se transpone una octava hacia abajo o hacia arriba, respectivamente, y `_15` y `^15` la transpone dos octavas. Si es necesario se pueden usar otros números enteros. El argumento *nombre_de_clave* se debe encerrar entre comillas si contiene caracteres no alfabéticos:
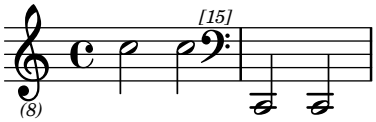
```
\clef treble
c2 c
\clef "treble_8"
c2 c
\clef "bass^15"
c2 c
```

```
\clef "alto_2"
c2 c
\clef "G_8"
c2 c
\clef "F^5"
c2 c
```



Se puede obtener una indicación de octavación opcional encerrando el argumento numérico entre paréntesis o corchetes rectos:

```
\clef "treble_(8)"
c2 c
\clef "bass^[15]"
c2 c
```



Las alturas se imprimen como si el argumento numérico no tuviera los paréntesis o corchetes.

Ciertas claves para usos especiales se describen en [Claves de la música mensural], página 425, [Claves de canto gregoriano], página 433, [Tablaturas predeterminadas], página 332 y [Tablaturas personalizadas], página 345. Para mezclar distintas claves al usar notas guía dentro de una partitura, consulte las funciones `\cueClef` y `\cueDuringWithClef` en la sección [Formateo de las notas guía], página 207.

Fragmentos de código seleccionados

Trucaje de las propiedades de clave

La instrucción `\clef "treble_8"` equivale a un ajuste de `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

Observe que la modificación del glifo, la posición de la clave o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama: para hacer esto también se debe especificar la posición del Do central. In order to get key signatures on the correct staff lines, `middleCClefPosition` must also be set. Los parámetros posicionales están en relación con la tercera línea del pentagrama, los números positivos desplazan hacia arriba, contando una unidad por cada línea y espacio. El valor de `clefTransposition` se establecería normalmente a 7, -7, 15 or -15, pero son válidos otros valores.

Cuando se produce un cambio de clave en el salto de línea se imprime la clave nueva tanto al final de la línea anterior como al principio de la nueva, de forma predeterminada. Si no se necesita la clave de advertencia al final de la línea anterior, se puede quitar estableciendo el valor de la propiedad `explicitClefVisibility` de `Staff`, a `end-of-line-invisible`. El comportamiento predeterminado se puede recuperar con `\unset Staff.explicitClefVisibility`.

Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

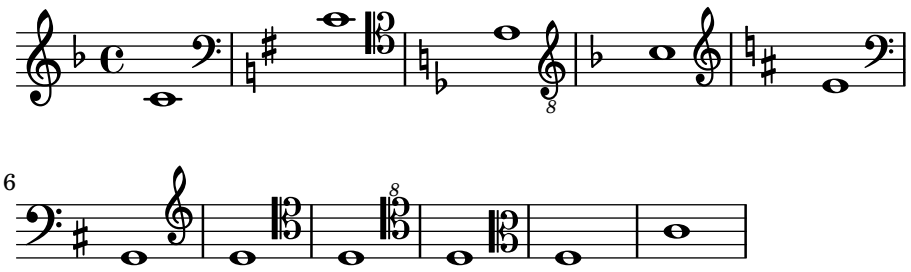
```
\layout { ragged-right = ##t }
{
  % The default treble clef
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  \set Staff.middleCClefPosition = #6
  \key g \major
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  \set Staff.middleCClefPosition = #4
  \key f \major
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefTransposition = #-7
  \set Staff.middleCPosition = #1
  \set Staff.middleCClefPosition = #1
  \key f \major
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefTransposition = #0
  \set Staff.middleCPosition = #-4
  \set Staff.middleCClefPosition = #-4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefTransposition = #7
  c'1
  \set Staff.clefTransposition = #0
  \set Staff.clefPosition = #0
  c'1
```

```
  % Return to the normal clef:

  \set Staff.middleCPosition = #0
  c'1
}
```



Véase también

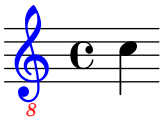
Referencia de la notación: [Claves de la música mensural], página 425, [Claves de canto gregoriano], página 433, [Tablaturas predeterminadas], página 332 [Tablaturas personalizadas], página 345, [Formateo de las notas guía], página 207.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Clef_engraver” in *Referencia de Funcionamiento Interno*, Sección “Clef” in *Referencia de Funcionamiento Interno*, Sección “ClefModifier” in *Referencia de Funcionamiento Interno*, Sección “clef-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los números de octavación adjuntos a las claves se tratan como grobs distintos. Así pues, cualquier `\override` (sobreescritura) efectuada al objeto *Clef* deberá aplicarse también al grob *ClefModifier* como un `\override` diferente.



Armadura de la tonalidad

> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Alteraciones accidentales y armaduras” in *Manual de Aprendizaje*.
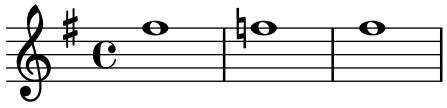
La armadura indica la tonalidad en que se toca una pieza. Está denotada por un conjunto de alteraciones (bemoles o sostenidos) al comienzo del pentagrama. El establecimiento o modificación de la armadura se hace con la instrucción `\key`:

```
\key nota modo
```

Aquí, *`modo`* debe ser `\major` o `\minor` para obtener la tonalidad *`nota`* mayor o *`nota`* menor, respectivamente. También puede usar los nombres estándar de modo (también conocidos como

modos eclesiásticos): `\ionian` (jónico), `\dorian` (dórico), `\phrygian` (frigio), `\lydian` (lidio), `\mixolydian` (mixolidio), `\aeolian` (eolio) y `\locrian` (locrio).
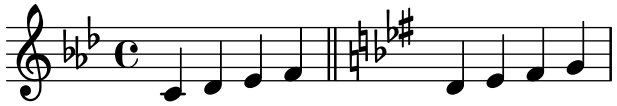
```
\key g \major
fis1
f
fis
```



Se pueden definir modos adicionales, escribiendo una lista con la alteración que lleva cada nota de la escala cuando el modo comienza en Do.

```
freygish = #`((0 . ,NATURAL) (1 . ,FLAT) (2 . ,NATURAL)
    (3 . ,NATURAL) (4 . ,NATURAL) (5 . ,FLAT) (6 . ,FLAT))
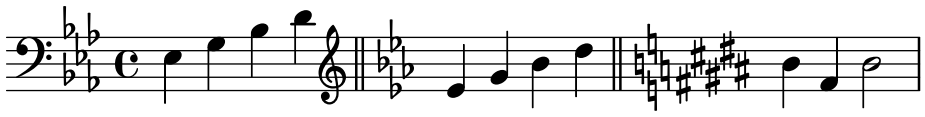
\relative c' {
  \key c \freygish c4 des e f
  \bar "||" \key d \freygish d es fis g
}
```



Las alteraciones de la armadura de tonalidad se pueden imprimir en distinta octava que en sus posiciones tradicionales, o en más de una octava, usando las propiedades `flat-positions` y `sharp-positions` de `KeySignature`. Las entradas en estas propiedades especifican el rango de posiciones de pentagrama en que se imprimirán las alteraciones. Si en una entrada se especifica una sola posición, las alteraciones se colocan dentro de la octava que termina en dicha posición del pentagrama.

```
\override Staff.KeySignature.flat-positions = #'((-5 . 5))
\override Staff.KeyCancellation.flat-positions = #'((-5 . 5))
\clef bass \key es \major es g bes d
\clef treble \bar "||" \key es \major es g bes d

\override Staff.KeySignature.sharp-positions = #'(2)
\bar "||" \key b \major b fis b2
```
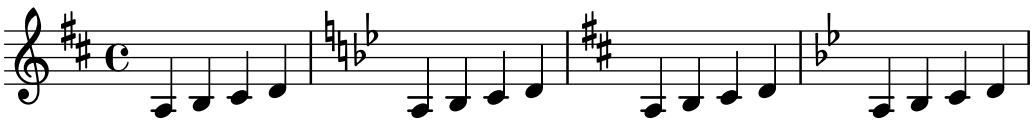


Fragmentos de código seleccionados

Evitar que se impriman becuadros cuando cambia la armadura

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo al valor "falso" la propiedad `printKeyCancellation` del contexto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keySignature` property, dentro del contexto `Staff`.
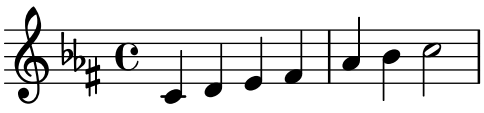
Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

`\set Staff.keySignature = #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)` donde, para cada elemento dentro de la lista, `octava` especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), `paso` especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y `alteración` es `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (observe la coma precedente.)

De forma alternativa, para cada elemento de la lista el uso del formato más conciso `(paso . alteración)` especifica que la misma alteración debe estar en todas las octavas.

He aquí un ejemplo de una posible armadura para generar una escala exátona:

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                              ((0 . 5) . ,FLAT)
                              ((0 . 3) . ,SHARP))
  c4 d e fis
  aes4 bes c2
}
```



Véase también

Glosario musical: Sección “church mode” in *Glosario Musical*, Sección “scordatura” in *Glosario Musical*.

Manual de aprendizaje: Sección “Alteraciones accidentales y armaduras” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

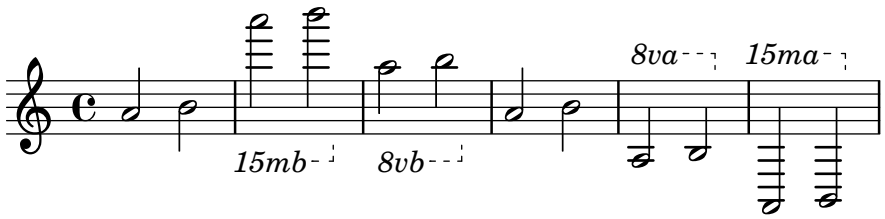
Referencia de funcionamiento interno: Sección “KeyChangeEvent” in *Referencia de Funcionamiento Interno*, Sección “Key_engraver” in *Referencia de Funcionamiento Interno*, Sección

"Key_performer" in *Referencia de Funcionamiento Interno*, Sección "KeyCancellation" in *Referencia de Funcionamiento Interno*, Sección "KeySignature" in *Referencia de Funcionamiento Interno*, Sección "key-signature-interface" in *Referencia de Funcionamiento Interno*.

Corchetes de octava

Los *corchetes de Ottava* introducen un trasporte adicional de una octava para el pentagrama.

```
a2 b
\ottava #-2
a2 b
\ottava #-1
a2 b
\ottava #0
a2 b
\ottava #1
a2 b
\ottava #2
a2 b
```

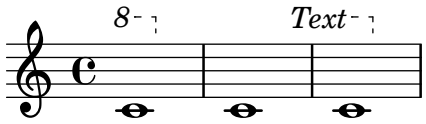


Fragmentos de código seleccionados

Texto de octava alta y baja

Internamente, `\ottava` establece las propiedades `ottavation` (por ejemplo, a `8va` o a `8vb`) y `middleCPosition`. Para sobreescribir el texto del corchete, ajuste `ottavation` después de invocar la instrucción `\ottava`.

```
{
  \ottava #1
  \set Staff.ottavation = #"8"
  c''1
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```

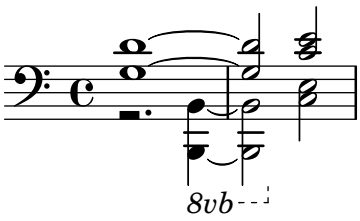


Añadir una indicación de octava alta a una sola voz

Si tiene más de una voz en el mismo pentagrama, el cambio de octavación de una voz transportará la posición de las notas en todas las voces mientras dure el corchete de octava. Si la octavación se quiere aplicar a una voz solamente, se deben ajustar explícitamente la middleCPosition y el corchete de octava. En este fragmento de código, el valor de middleCPosition para la

clave de Fa es normalmente 6, seis posiciones por encima de la línea del Do central, de manera que en la porción de 8va el valor de middleCPosition es aún 7 posiciones (una octava) más alta.

```
{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
      r2.
      \set Staff.ottavation = #"8vb"
      \once \override Staff.OttavaBracket.direction = #DOWN
      \set Voice.middleCPosition = #(+ 6 7)
      <b,,, b,,>4 ~ |
      q2
      \unset Staff.ottavation
      \unset Voice.middleCPosition
      <c e>2
    }
  >>
}
```



Véase también

Glosario musical: Sección “octavation” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ottava_spanner_engraver” in *Referencia de Funcionamiento Interno*, Sección “OttavaBracket” in *Referencia de Funcionamiento Interno*, Sección “ottava-bracket-interface” in *Referencia de Funcionamiento Interno*.

Transposición de los instrumentos

Al tipografiar partituras donde participan instrumentos transpositores, ciertas partes se pueden tipografiar en un tono distinto del *tono de concierto*. En estos casos, se debe especificar la tonalidad del *instrumento transpositor*; de otro modo, la salida MIDI y las notas guía en otras partes producirían alturas incorrectas. Para ver más información sobre partes citadas como guía, consulte [Citar otras voces], página 204.

`\transposition` *pitch*
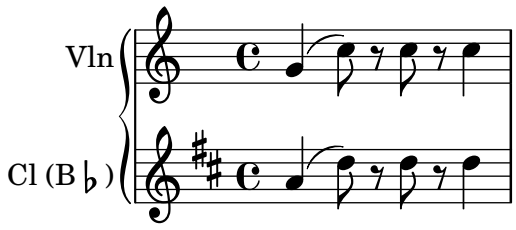
El tono usado para `\transposition` debe corresponderse con el sonido real que se oye cuando el instrumento transpositor interpreta un Do central `c'` escrito en el pentagrama. Esta nota se escribe en altura absoluta, por tanto un instrumento que produce un sonido real un tono más agudo que la música impresa (un instrumento en Re) debe usar `\transposition d'`. La instrucción `\transposition` se debe usar *solamente* si las notas *no* se van a escribir en afinación de concierto.

A continuación pueden verse algunas notas para violín y para clarinete en Si bemol, donde las partes se han introducido usando las notas y la armadura tal y como aparecen en la partitura del director. Lo que tocan los dos instrumentos está sonando al unísono.

```
\new GrandStaff <<
  \new Staff = "violin" {
    \relative c'' {
      \set Staff.instrumentName = #"Vln"
      \set Staff.midiInstrument = #"violin"
      % not strictly necessary, but a good reminder
      \transposition c'

      \key c \major
      g4( c8) r c r c4
    }
  }
  \new Staff = "clarinet" {
    \relative c'' {
      \set Staff.instrumentName = \markup { Cl (B\flat) }
      \set Staff.midiInstrument = #"clarinet"
      \transposition bes

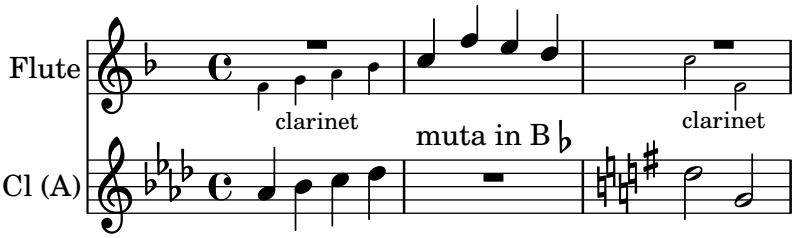
      \key d \major
      a4( d8) r d r d4
    }
  }
>>
```



La `\transposition` se puede cambiar durante la pieza. Por ejemplo, un clarinetista puede cambiar del clarinete en La al clarinete en Si bemol.

```
flute = \relative c'' {
  \key f \major
  \cueDuring #"clarinet" #DOWN {
    R1 _\markup\tiny "clarinet"
    c4 f e d
    R1 _\markup\tiny "clarinet"
  }
}
clarinet = \relative c'' {
  \key aes \major
  \transposition a
  aes4 bes c des
  R1^\markup { muta in B\flat }
  \key g \major
  \transposition bes
  d2 g,
}
\addQuote "clarinet" \clarinet
```

```
<<
  \new Staff \with { instrumentName = #"Flute" }
    \flute
  \new Staff \with { instrumentName = #"Cl (A)" }
    \clarinet
>>
```



Véase también

Glosario musical: Sección “concert pitch” in *Glosario Musical*, Sección “transposing instrument” in *Glosario Musical*.

Referencia de la notación: [Citar otras voces], página 204, [Transposición], página 10.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Alteraciones accidentales automáticas

Existen muchas convenciones distintas sobre la forma de tipografiar las alteraciones. LilyPond proporciona una función para especificar qué estilo de alteraciones usar. Esta función se invoca como sigue:

```
\new Staff <<
  \accidentalStyle voice
  { ... }
>>
```

El estilo de alteraciones se aplica al `Staff` en curso de forma predeterminada (con la excepción de los estilos `piano` y `piano-cautionary`, que se explican más adelante). Opcionalmente, la función puede tomar un segundo argumento que determina en qué ámbito se debe cambiar el estilo. Por ejemplo, para usar el mismo estilo en todos los pentagramas del `StaffGroup` en curso, use

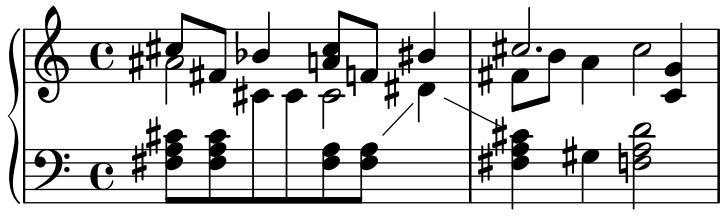
```
\accidentalStyle StaffGroup.voice
```

Están contemplados los siguientes estilos de alteración. Para dar una muestra de cada uno de los estilos, utilizamos el ejemplo siguiente:

```
musicA = {
  <<
    \relative c' {
      cis'8 fis, bes4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
    \\
    \relative c' {
      ais'2 cis, |
      fis8 b a4 cis2 |
    }
  >>
}
```

```
musicB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative c' {
      <fis, a cis>8[ <fis a cis>
      \change Staff = up
      cis' cis
      \change Staff = down
      <fis, a> <fis a>]
      \showStaffSwitch
      \change Staff = up
      dis'4 |
      \change Staff = down
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}

\new PianoStaff {
  <<
    \context Staff = "up" {
      \accidentalStyle default
      \musicA
    }
    \context Staff = "down" {
      \accidentalStyle default
      \musicB
    }
  >>
}
```



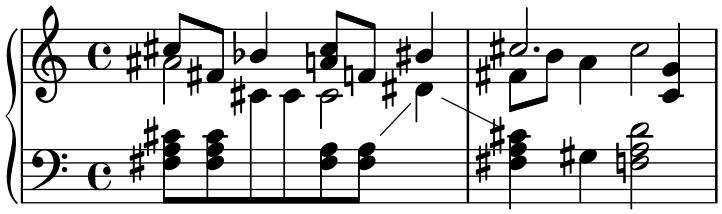
Observe que las últimas líneas de este ejemplo se pueden sustituir por las siguientes, siempre y cuando queramos usar el mismo estilo en los dos pentagramas.

```
\new PianoStaff {
  <<
    \context Staff = "up" {
      %%% cambie la línea siguiente como desee:
      \accidentalStyle Score.default
      \musicA
    }
    \context Staff = "down" {
      \musicB
    }
  >>
```

```
}
```

`default (predeterminado)`

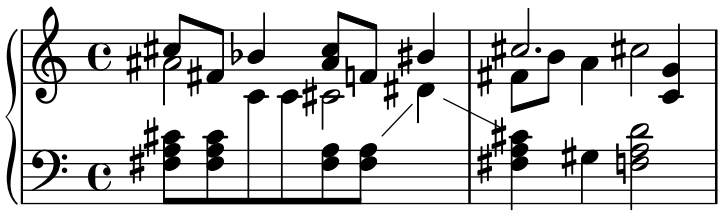
Es el comportamiento de composición tipográfica predeterminado. Corresponde a la práctica común del s. XVIII: las alteraciones accidentales se recuerdan hasta el final del compás en el que aparecen y sólo en la misma octava. Así, en el ejemplo siguiente, no se imprimen becuadros antes del Si natural en el segundo compás ni en el último Do:



`voice (voz)`

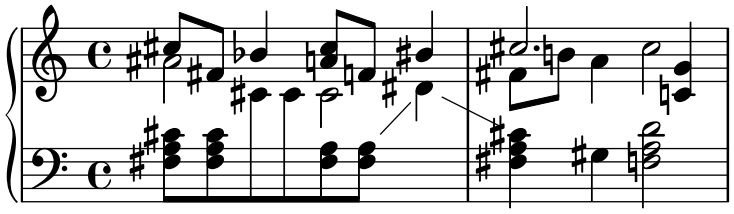
El comportamiento normal es recordar las alteraciones accidentales al nivel de `Staff`. En este estilo, sin embargo, se tipografían las alteraciones individualmente para cada voz. Aparte de esto, la regla es similar a `default`.

Como resultado, las alteraciones de una voz no se cancelan en las otras voces, lo que con frecuencia lleva a un resultado no deseado: en el ejemplo siguiente, es difícil determinar si el segundo La se debe tocar natural o sostenido. Por tanto, la opción `voice` se debe usar sólo si las voces se van a leer individualmente por músicos distintos. Si el pentagrama va a utilizarse por parte de un solo músico (p.ej., un director, o en una partitura de piano), entonces se deben usar en su lugar los estilos `modern` o `modern-cautionary`.



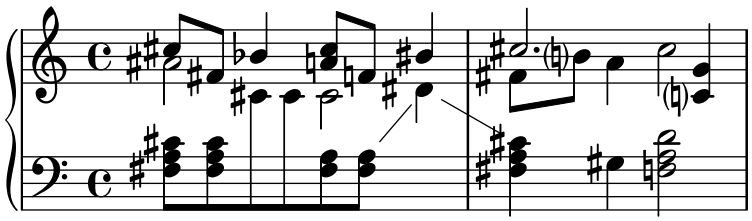
`modern (moderno)`

Esta regla corresponde a la práctica común del s. XX. Omite algunos becuadros adicionales, que tradicionalmente se imprimían precediendo a un sostenido que sigue a un doble sostenido, o a un bemol que sigue a un doble bemol. La regla `modern` imprime las mismas alteraciones que el estilo `default`, con dos adiciones que sirven para evitar la ambigüedad: después de alteraciones temporales se imprimen indicaciones de cancelación también en el compás siguiente (para notas en la misma octava) y, en el mismo compás, para notas en octavas distintas. De aquí los becuadros antes del Si natural y del Do en el segundo compás del pentagrama superior:



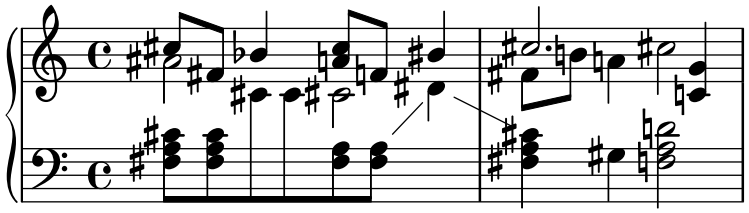
`modern-cautionary (moderno de precaución)`

Esta regla es similar a `modern`, pero las alteraciones 'añadidas' (las que no se imprimen en el estilo `default`) se imprimen como alteraciones de precaución. Se imprimen de forma predeterminada con paréntesis, pero también se pueden imprimir en tamaño reducido definiendo la propiedad `cautionary-style` de `AccidentalSuggestion`.



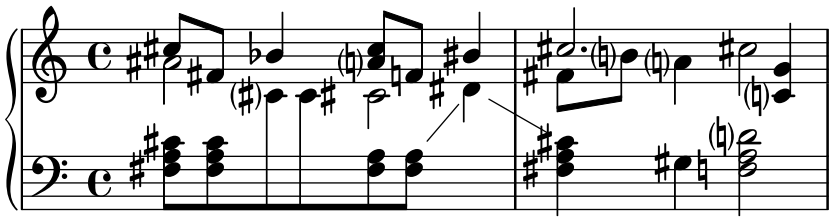
`modern-voice (moderno, para voces)`

Esta regla se usa para que puedan leer las alteraciones en varias voces, tanto músicos que tocan una voz como músicos que tocan todas las voces. Se imprimen las alteraciones para cada voz, pero *se cancelan* entre voces dentro del mismo `Staff`. Por tanto, el La en el último compás se cancela porque la cancelación anterior estaba en una voz distinta, y el Re en el pentagrama inferior se cancela a causa de la alteración en otra voz en el compás previo:



`modern-voice-cautionary (moderno, voz, de precaución)`

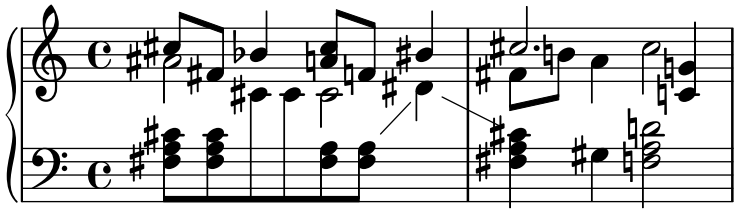
Esta regla es la misma que `modern-voice`, pero con las alteraciones añadidas (las que el estilo `voice` no imprime) compuestas como de precaución. Incluso aunque todas las alteraciones impresas por el estilo `default` *son* impresas con esta regla, algunas de ellas se tipografían como de precaución.



`piano`

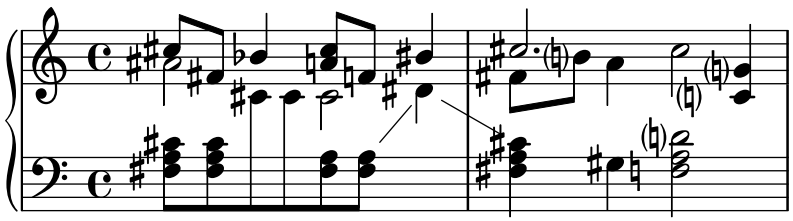
Esta regla refleja la práctica del s.XX para la notación de piano. Su comportamiento es muy similar al estilo `modern`, pero aquí las alteraciones también se cancelan entre distintos pentagramas del mismo grupo `GrandStaff` o `PianoStaff`, de ahí todas las cancelaciones de las últimas notas.

Este estilo de alteración se aplica de manera predeterminada al grupo `GrandStaff` o `PianoStaff` en curso.



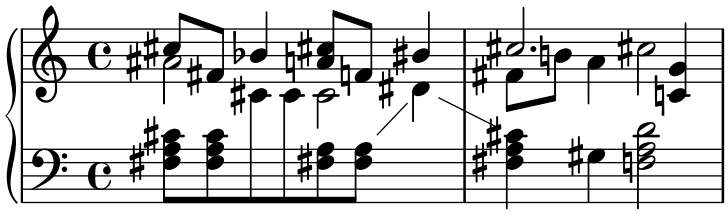
`piano-cautionary (piano, de precaución)`

Igual que `\accidentalStyle piano` pero con las alteraciones añadidas compuestas como de precaución.



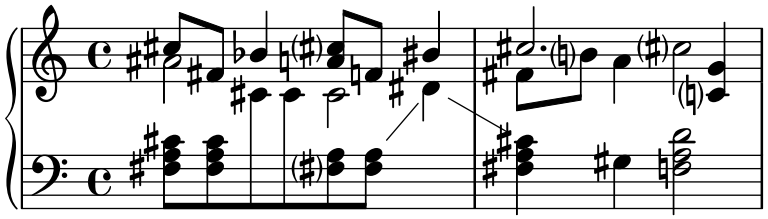
`neo-modern`

Esta regla reproduce una práctica común en la música contemporánea: las alteraciones accidentales se imprimen como en `modern`, pero se vuelven a imprimir si aparece la misma nota otra vez en el mismo compás (excepto si la nota se repite inmediatamente).



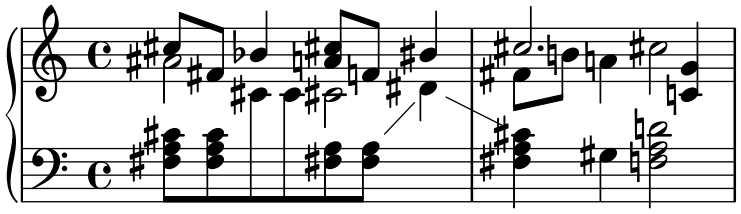
`neo-modern-cautionary`

Esta regla es similar a `neo-modern`, pero las alteraciones adicionales se imprimen como alteraciones de precaución.



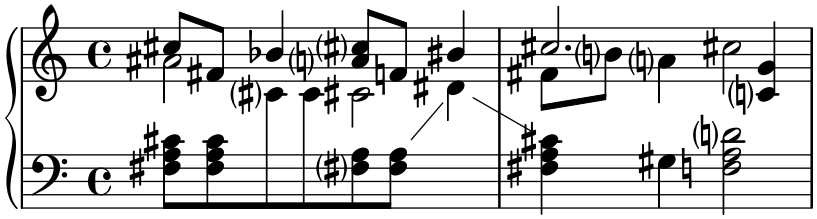
`neo-modern-voice`

Esta regla se usa para alteraciones accidentales sobre varias voces que se han de leer por parte de músicos que tocan una voz, así como por músicos que tocan todas las voces. Las alteraciones se imprimen para cada voz como con `neo-modern`, pero se cancelan para otras voces que están en el mismo pentagrama `Staff`.



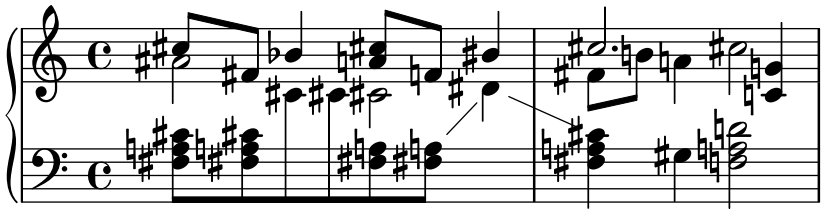
`neo-modern-voice-cautionary`

Esta regla es similar a `neo-modern-voice`, pero las alteraciones adicionales se imprimen como alteraciones de precaución.



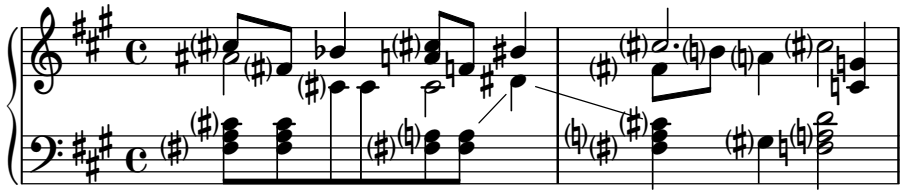
`dodecaphonic`

Esta regla refleja una práctica introducida por los compositores de principios del s.XX, en un intento de abolir la jerarquía entre notas naturales y alteradas. Con este estilo, *todas* las notas llevan alteración, incluso becuadros.



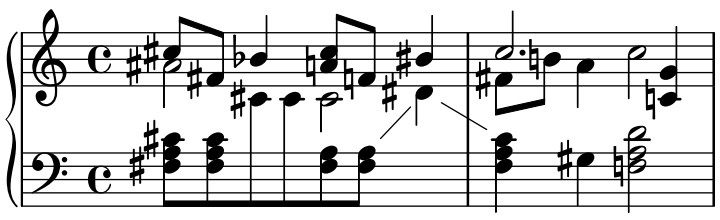
`teaching (enseñanza)`

Esta regla está pensada para estudiantes, y hace más sencillo crear hojas de escalas con alteraciones de precaución creadas automáticamente. Las alteraciones se imprimen como en el estilo `modern`, pero se añaden alteraciones de precaución para todas las notas sostenidas o bemoles especificadas por la armadura, excepto si la nota se repite inmediatamente.



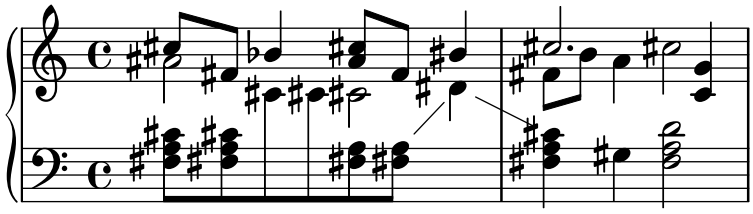
`no-reset (no restablecer)`

Es el mismo que `default` pero con alteraciones que duran 'para siempre' y no sólo dentro del mismo compás:



`forget (olvidar)`

Es lo opuesto a `no-reset`: Las alteraciones no se recuerdan en absoluto: de aquí que todas las alteraciones se compongan tipográficamente en relación a la armadura de la tonalidad, sin que estén afectadas por lo que viene antes.



Véase también

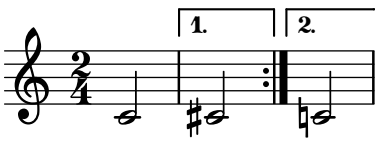
Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental” in *Referencia de Funcionamiento Interno*, Sección “Accidental_engraver” in *Referencia de Funcionamiento Interno*, Sección “GrandStaff” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” in *Referencia de Funcionamiento Interno*, Sección “AccidentalPlacement” in *Referencia de Funcionamiento Interno*, Sección “accidental-suggestion-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

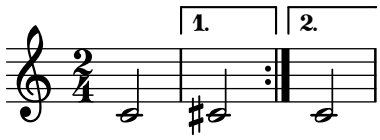
Las notas simultáneas no se tienen en cuenta para la determinación automática de las alteraciones accidentales; sólo se consideran las notas anteriores y la armadura de la tonalidad. Puede ser necesario forzar las alteraciones accidentales con with `!` o `?` cuando la misma nota, con el mismo nombre, ocurre simultáneamente con distintas alteraciones, como en ‘`<f! fis!>`’.

La cancelación de precaución de alteraciones se hace mirando el compás previo. Sin embargo, en el bloque `\alternative` que sigue a una sección de repetición de primera y segunda vez `\repeat volta N`, se esperaría que la cancelación se calculase utilizando el compás previo *que se ha tocado*, no el compás previo *que se ha impreso*. En el ejemplo siguiente, el Do natural de la casilla de segunda vez no necesita becuadro:



Se puede usar el siguiente rodeo del problema: definir una función que cambie localmente el estilo de alteraciones a `forget`:

```
forget = #(define-music-function (parser location music) (ly:music?) #{
  \accidentalStyle forget
  #music
  \accidentalStyle modern
#})
{
  \accidentalStyle modern
  \time 2/4
  \repeat volta 2 {
    c'2
  }
  \alternative {
     cis'
     \forget c'
  }
}
```
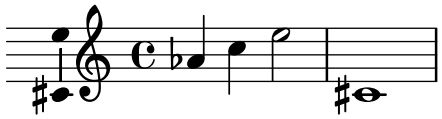


Tesitura

El término *ambitus* o ámbito, denota el rango de notas que abarca una voz dada en una parte musical. También puede denotar el margen de notas que es capaz de tocar un determinado instrumento musical. Los ámbitos se imprimen en las partes vocales de tal manera que los intérpretes puedan determinar con facilidad si cumplen con sus propias posibilidades.

Los ámbitos se presentan al comienzo de la pieza junto a la clave inicial. El rango se especifica gráficamente mediante dos cabezas de nota que representan a las notas inferior y superior. Sólo se imprimen alteraciones si no forman parte de la armadura de la tonalidad.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\relative c'' {
  aes c e2
  cis,1
}
```
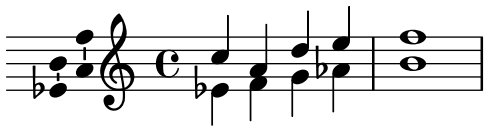


Fragmentos de código seleccionados

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.
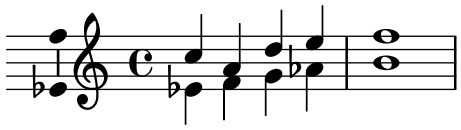
```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Ámbitos con varias voces

La adición del grabador `Ambitus_engraver` al contexto de `Staff` crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Modificación de la separación en las indicaciones de tesitura

Es posible ajustar la separación predeterminada entre las notas de la indicación de tesitura y la línea que las une.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

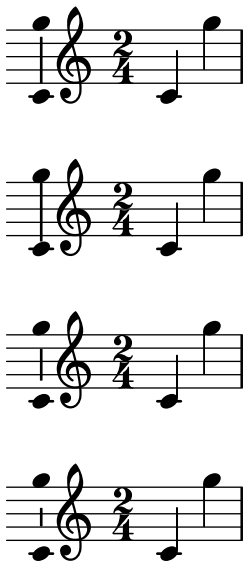
\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1
  c'4 g''
```

```
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1.5
  c'4 g''
}
```



Véase también

Glosario musical: Sección “ambitus” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ambitus_engraver” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “Ambitus” in *Referencia de Funcionamiento Interno*, Sección “AmbitusAccidental” in *Referencia de Funcionamiento Interno*, Sección “AmbitusLine” in *Referencia de Funcionamiento Interno*, Sección “AmbitusNoteHead” in *Referencia de Funcionamiento Interno*, Sección “ambitus-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se efectúa un tratamiento de las posibles colisiones en caso de varias indicaciones de ámbito simultáneas en distintas voces.

1.1.4 Cabeza de las notas
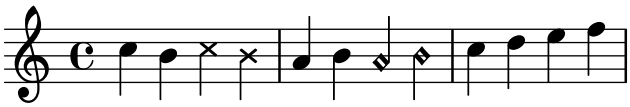
Esta sección propone formas de alterar las cabezas de las figuras.

Cabezas de nota especiales

Se puede modificar la apariencia de la cabeza de las notas:

```
c4 b
\override NoteHead.style = #'cross
c4 b
\revert NoteHead.style
a b
```

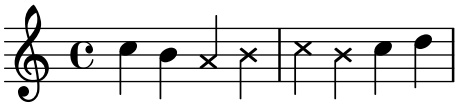
```
\override NoteHead.style = #'harmonic
a b
\revert NoteHead.style
c4 d e f
```



Para ver todos los estilos de cabeza de las notas, consulte Sección A.9 [Estilos de cabezas de nota], página 671.

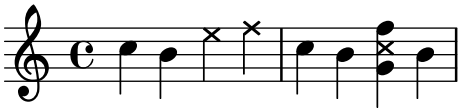
El estilo `cross` (aspas) se usa para representar una amplia variedad de intenciones musicales. Las siguientes instrucciones genéricas predefinidas modifican la forma de la cabeza de las figuras tanto en el contexto de pentagrama normal como en el de tablatura, y se pueden usar para representar cualquier significado musical:

```
c4 b
\xNotesOn
 a b c4 b
\xNotesOff
c4 d
```



La forma de función musical de esta instrucción predefinida se puede usar dentro y fuera de los acordes para generar cabezas de nota en aspa, tanto en el contexto de pentagrama normal como en el de tablatura:
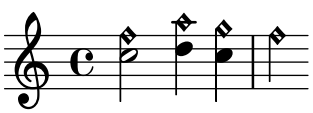
```
c4 b
\xNote { e f }
c b < g \xNote c f > b
```



Como sinónimos de `\xNote`, `\xNotesOn` y `\xNotesOff`, se pueden usar `\deadNote`, `\deadNotesOn` y `\deadNotesOff`. El término *dead note* (nota muerta) se utiliza corrientemente por parte de los guitarristas.

También existe una abreviatura similar para las formas en rombo:

```
<c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic> f\harmonic
```



Instrucciones predefinidas

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

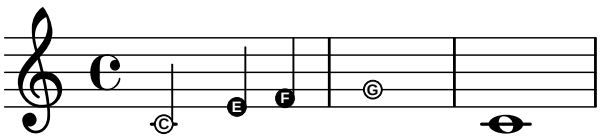
Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 671, [Notas en acorde], página 161, [Indicar armónicos y notas tapadas], página 372.

Referencia de funcionamiento interno: Sección “note-event” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Ledger_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteHead” in *Referencia de Funcionamiento Interno*, Sección “LedgerLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “note-head-interface” in *Referencia de Funcionamiento Interno*, Sección “ledger-line-spanner-interface” in *Referencia de Funcionamiento Interno*.

Cabezas de notas de Notación Fácil

Las notas con cabeza de ‘notación facilitada’ tienen el nombre de la nota (en inglés) dentro de la cabeza. Se usan en la música para principiantes. Para que las letras sean legibles, se deben imprimir en un tamaño grande de fuente tipográfica. Para imprimir con una fuente más grande, véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.

```
#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}
```



Instrucciones predefinidas

`\easyHeadsOn`, `\easyHeadsOff`.

Fragmentos de código seleccionados

Números como notas de notación fácil

Las cabezas de nota de notación fácil utilizan la propiedad `note-names` del objeto `NoteHead` para determinar lo que aparece dentro de la cabeza. Mediante la sobreescritura de esta propiedad, es posible imprimir números que representen el grado de la escala.

Se puede crear un grabador simple que haga esto para la cabeza de cada nota que ve.

```
#(define Ez_numbers_engraver
   (make-engraver
    (acknowledgers
     ((note-head-interface engraver grob source-engraver)
      (let* ((context (ly:translator-context engraver))
             (tonic-pitch (ly:context-property context 'tonic))
             (tonic-name (ly:pitch-notename tonic-pitch))
             (grob-pitch
              (ly:event-property (event-cause grob) 'pitch))
             (grob-name (ly:pitch-notename grob-pitch))
```

```
              (delta (modulo (- grob-name tonic-name) 7))
              (note-names
               (make-vector 7 (number->string (1+ delta)))))
          (ly:grob-set-property! grob 'note-names note-names))))))

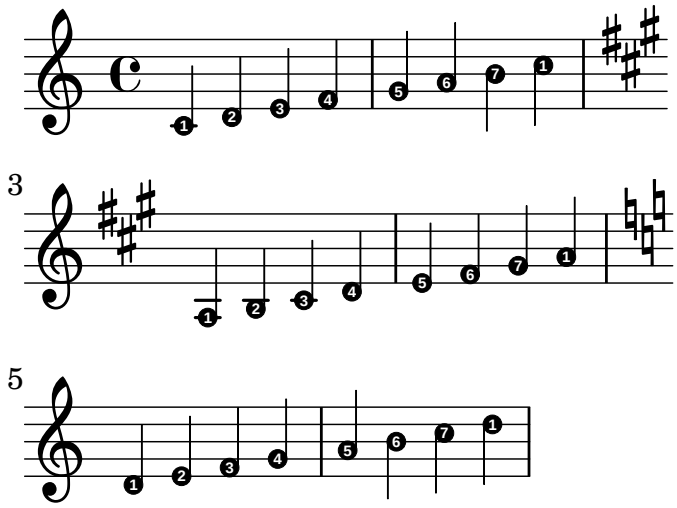
#(set-global-staff-size 26)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```



Véase también

Referencia de la notación: Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.
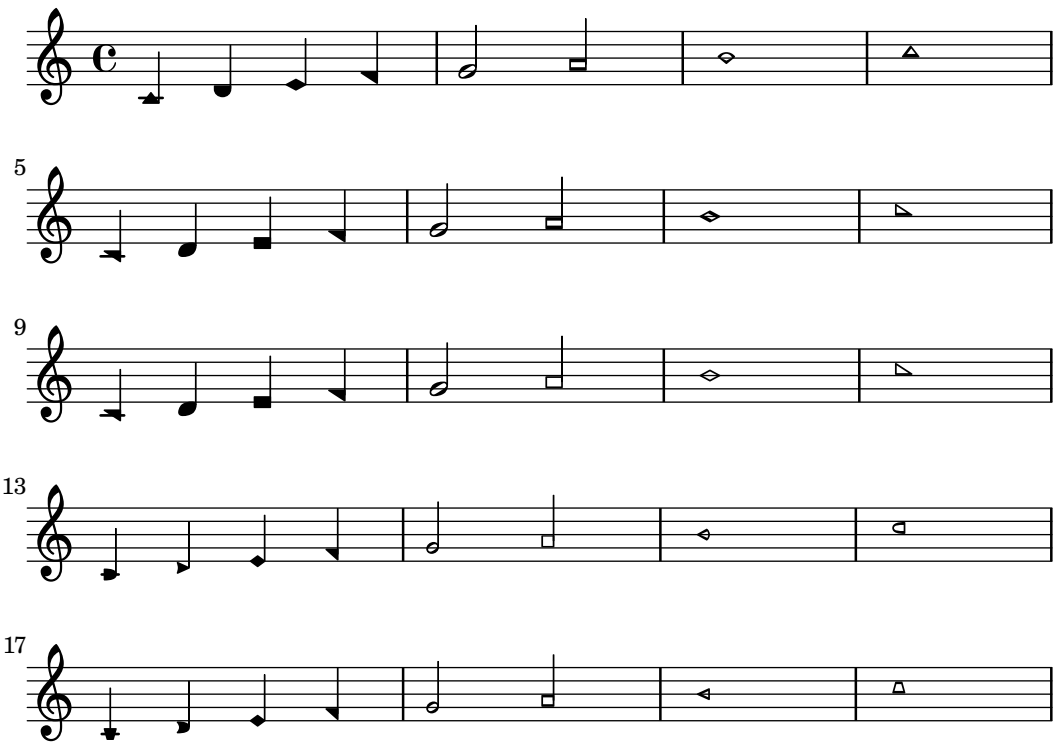
Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “note-event” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteHead” in *Referencia de Funcionamiento Interno*, Sección “note-head-interface” in *Referencia de Funcionamiento Interno*.

Cabezas de notas con formas diversas

En la notación de cabezas con forma, la forma de la cabeza corresponde a la función armónica de una nota dentro de la escala. Esta notación se hizo popular en los libros americanos de canciones durante el s.XIX. Las cabezas de nota con formas se pueden producir en los estilos 'Sacred Harp', 'Southern Harmony', Funk (Harmonica Sacra), Walker y Aiken '(Christian Harmony)':
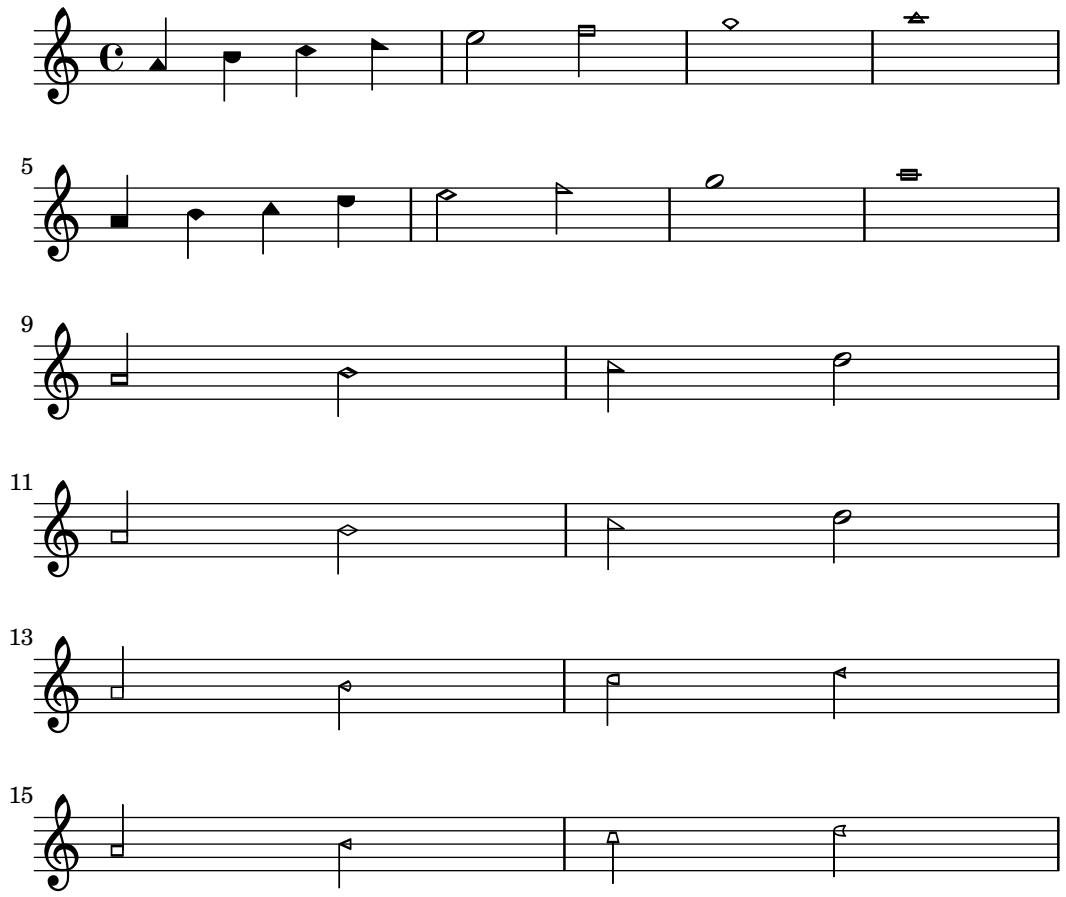
```
\aikenHeads
c, d e f g2 a b1 c \break
\sacredHarpHeads
c,4 d e f g2 a b1 c \break
\southernHarmonyHeads
c,4 d e f g2 a b1 c \break
\funkHeads
c,4 d e f g2 a b1 c \break
\walkerHeads
c,4 d e f g2 a b1 c \break
```



Las formas se determinan en función del grado de la escala, donde la tónica está determinada por la instrucción `\key` Cuando se escribe en un tono menor, la nota de la escala se puede determinar a partir del relativo mayor:

```
\key a \minor
\aikenHeads
a b c d e2 f g1 a \break
\aikenHeadsMinor
a,4 b c d e2 f g1 a \break
\sacredHarpHeadsMinor
a,2 b c d \break
\southernHarmonyHeadsMinor
a2 b c d \break
\funkHeadsMinor
```

```
a2 b c d \break
\walkerHeadsMinor
a2 b c d \break
```



Instrucciones predefinidas

`\aikenHeads`, `\aikenHeadsMinor`, `\funkHeads`, `\funkHeadsMinor`, `\sacredHarpHeads`, `\sacredHarpHeadsMinor`, `\southernHarmonyHeads`, `\southernHarmonyHeadsMinor`, `\walkerHeads`, `\walkerHeadsMinor`.

Fragmentos de código seleccionados

Aplicar estilos de cabeza según la nota de la escala

La propiedad `shapeNoteStyles` se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad `tonic`). Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como `triangle`, triángulo, `cross`, aspas, y `xcircle`, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos trambién se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como `\aikenHeads` o `\sacredHarpHeads`.

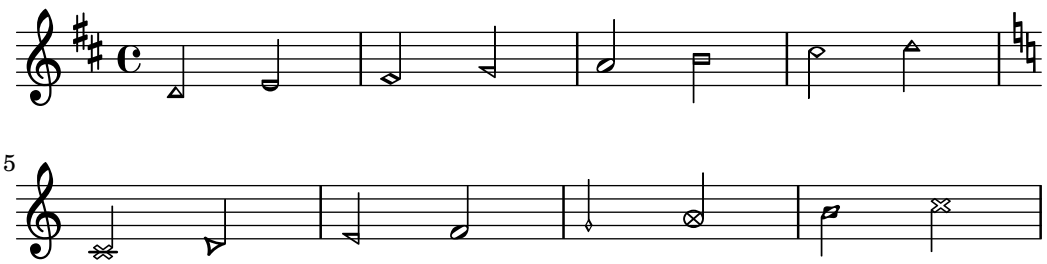
Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```



Para ver todos los estilos de formas de cabeza de nota, consulte Sección A.9 [Estilos de cabezas de nota], página 671.

Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 671.

Referencia de funcionamiento interno: Sección “note-event” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteHead” in *Referencia de Funcionamiento Interno*, Sección “note-head-interface” in *Referencia de Funcionamiento Interno*.

Improvisación

La improvisación se denota a veces mediante cabezas de nota en forma de barra inclinada, donde el ejecutante puede elegir cualquier nota pero con el ritmo especificado. Estas cabezas de nota se crean así:

```
\new Voice \with {
  \consists "Pitch_squash_engraver"
} {
```

```
  e8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  e2 ~ e8 f4 f8 ~
  f2
  \improvisationOff
  a16( bes) a8 g e
}
```



Instrucciones predefinidas
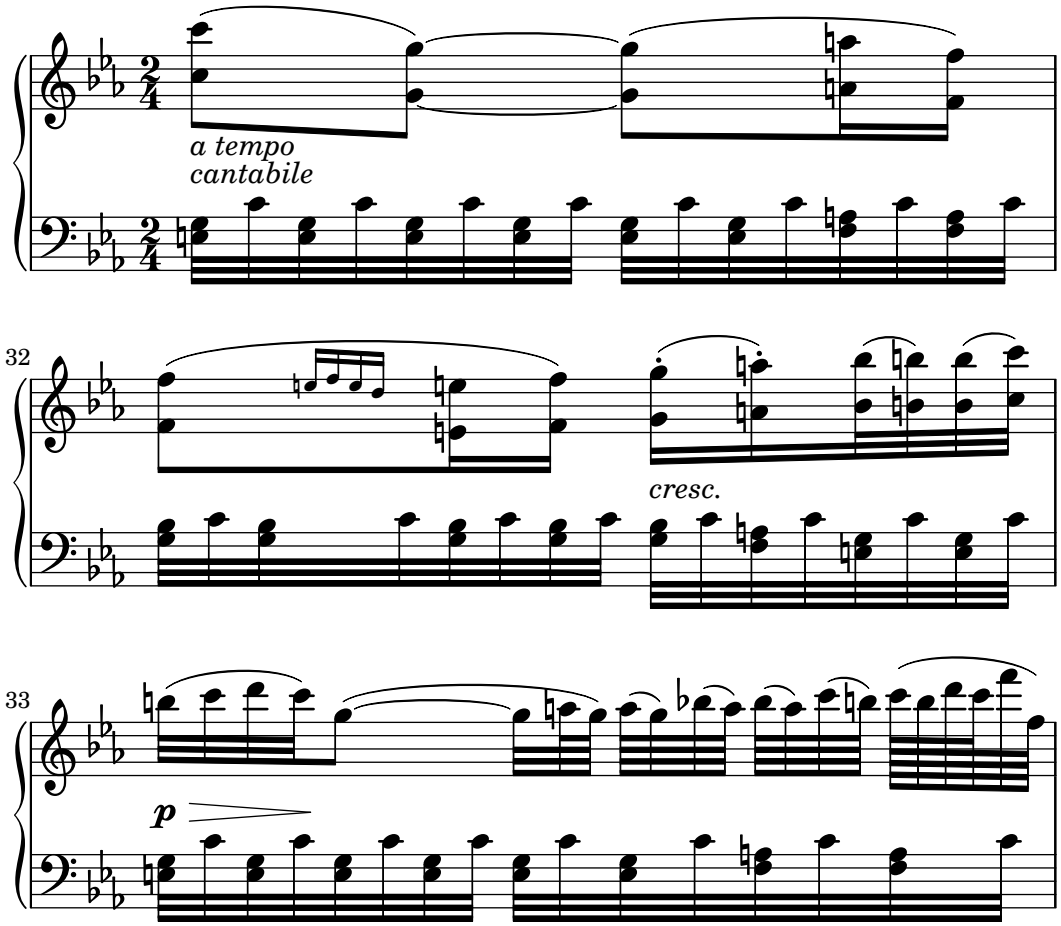
`\improvisationOn`, `\improvisationOff`.

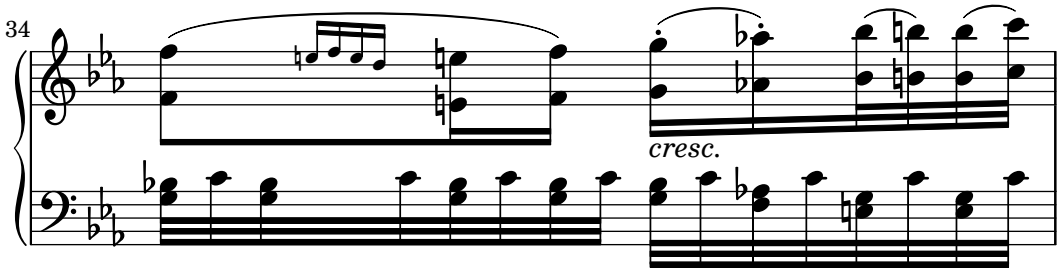
Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*.

1.2 Duraciones
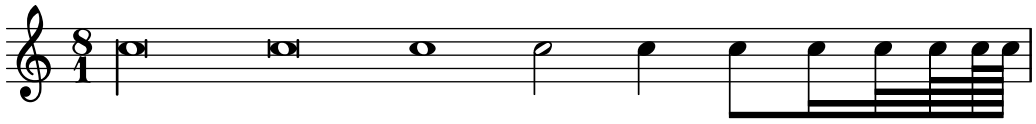




Esta sección trata de los ritmos, los silencios, las duraciones, las barras y los compases.

1.2.1 Escritura de las duraciones (valores rítmicos)

Duración de las notas
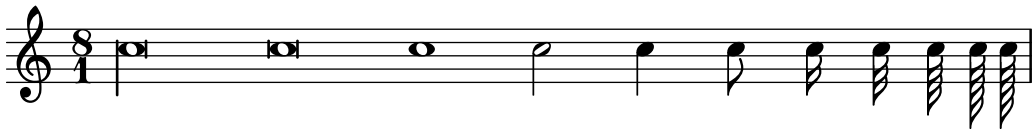
Las duraciones se designan mediante números y puntos. Las duraciones se introducen como sus valores recíprocos respecto de la redonda. Por ejemplo, una negra se escribe usando un `4` (puesto que es 1/4 de redonda), mientras que una blanca se escribe con un `2` (por ser 1/2 de redonda). Para notas mayores de la redonda se deben usar las instrucciones `\longa` (que es una breve doble) y `\breve`. Se pueden especificar duraciones tan cortas como la garrapatea (con el número 128). Son posibles valores más cortos, pero sólo como notas unidas por una barra.

```
\time 8/1
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Aquí se pueden ver las mismas duraciones con el barrado automático desactivado.
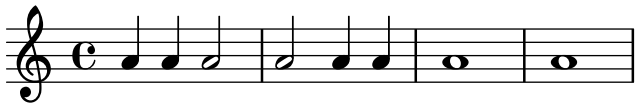
```
\time 8/1
\autoBeamOff
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Se puede escribir una nota con la duración de una cuádruple breve mediante `\maxima`, pero esto está contemplado solamente dentro de la notación musical antigua. Para ver más detalles, consulte Sección 2.9 [Notación antigua], página 420.
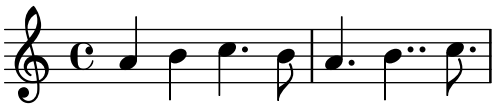
Si la duración se omite, su valor será el de la duración de la nota anterior. Por omisión, el valor de la primera nota es el de una negra.

```
a a a2 a a4 a a1 a
```



Para obtener duraciones de notas con puntillo, escriba un punto (`.`) después del número. Las notas con doble puntillo se especifican escribiendo dos puntos, y así sucesivamente.

```
a4 b c4. b8 a4. b4.. c8.
```



Ciertas duraciones no se pueden representar sólo con duraciones binarias y puntillos; la única forma de representarlas es ligando dos o más notas. Para ver más detalles, consulte [Ligaduras de unión], página 51.

Para ver de qué manera se pueden especificar las duraciones de las sílabas de la letra y cómo alinear la letra a las notas, consulte Sección 2.1 [Música vocal], página 250.

Opcionalmente se pueden espaciar las notas de forma estrictamente proporcional a sus duraciones. Para ver más detalles sobre este y otros ajustes que controlan la notación proporcional, consulte Sección 4.5.5 [Notación proporcional], página 557.

Normalmente los puntillos se desplazan hacia arriba para evitar las líneas del pentagrama, excepto en situaciones de polifonía. Se pueden situar los puntillos manualmente encima o debajo de las líneas de la pauta; véase Sección 5.4.2 [Dirección y posición], página 601.

Instrucciones predefinidas

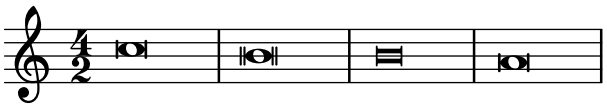
`\autoBeamOn`, `\autoBeamOff`, `\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Fragmentos de código seleccionados

Formas alternativas de la figura breve

Las figuras de breve también están disponibles con dos líneas verticales a los lados de la cabeza en lugar de una sola línea y la forma en estilo barroco.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```
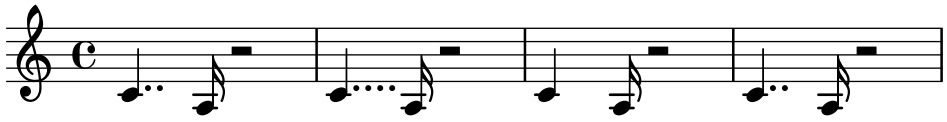


Modificar el número de puntillos de una nota

La cantidad de puntillos de una nota se puede modificar independientemente de los puntillos que se escriben después de la nota.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = #4
  c4.. a16 r2 |
  \override Dots.dot-count = #0
  c4.. a16 r2 |
  \revert Dots.dot-count
```

```
  c4.. a16 r2 |
}
```



Véase también

Glosario musical: Sección “breve” in *Glosario Musical*, Sección “longa” in *Glosario Musical*, Sección “maxima” in *Glosario Musical*, Sección “note value” in *Glosario Musical*, Sección “Duration names notes and rests” in *Glosario Musical*.

Referencia de la notación: [Barras automáticas], página 79, [Ligaduras de unión], página 51, [Plicas], página 219, Sección 1.2.1 [Escritura de las duraciones (valores rítmicos)], página 43, Sección 1.2.2 [Escritura de silencios], página 54, Sección 2.1 [Música vocal], página 250, Sección 2.9 [Notación antigua], página 420, Sección 4.5.5 [Notación proporcional], página 557.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Dots” in *Referencia de Funcionamiento Interno*, Sección “DotColumn” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe un límite fundamental para las duraciones de los silencios (tanto para el más largo como para el más corto), pero el numero de glifos es limitado: sólo se pueden imprimir desde el silencio de garrapatea (128) hasta el de máxima (8 redondas).
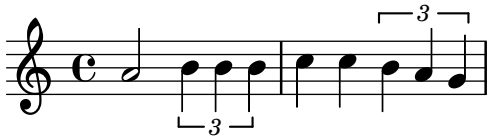
Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical con la instrucción `\tuplet`, multiplicando la rapidez de la expresión musical por una fracción:

```
\tuplet fracción { música }
```
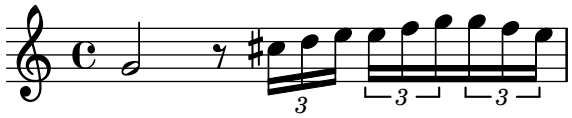
El nomerador de la fracción se imprime encima o debajo de las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor que normalmente tienen 2:

```
a2 \tuplet 3/2 { b4 b b }
c4 c \tuplet 3/2 { b4 a g }
```



Si se están escribiendo pasajes con muchos grupos de valoración especial, resulta fastidioso tener que escribir una instrucción `\tuplet` distinta para cada grupo. Es posible especificar la duración de un conjunto de grupos directamente antes de la música para que así se agrupen automáticamente:
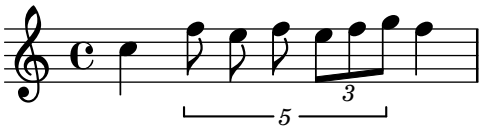
```
g2 r8 \tuplet 3/2 8 { cis16 d e e f g g f e }
```



Se pueden colocar manualmente los corchetes de tresillo encima o debajo de la pauta; véase Sección 5.4.2 [Dirección y posición], página 601.

Los grupos pueden anidarse unos dentro de otros:

```
\autoBeamOff
c4 \tuplet 5/4 { f8 e f \tuplet 3/2 { e[ f g] } } f4
```



La modificación de los grupos especiales anidados que comienzan en el mismo instante musical se debe hacer con `\tweak`.

Para modificar la duración de las notas sin imprimir un corchete de grupo especial, véase [Escalar las duraciones], página 49.

Instrucciones predefinidas

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

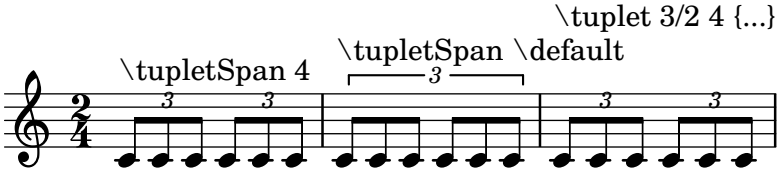
Fragmentos de código seleccionados

Escribir varios grupos especiales usando una sola instrucción \tuplet

La propiedad `tupletSpannerDuration` establece cuánto debe durar cada grupo de valoración especial contenido dentro del corchete que aparece después de `\tuplet`. Así, se pueden escribir muchos tresillos seguidos dentro de una sola expresión `\tuplet`, ahorrando trabajo de teclado.

Existen varias maneras de fijar el valor de `tupletSpannerDuration`. La instrucción `\tupletSpan` lo establece a una duración dada, y lo borra cuando se especifica `\default` en lugar de una duración. Otra forma es utilizar un argumento opcional con `\tuplet`.

```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\\tupletSpan \\default" c c c c c }
  \tuplet 3/2 4 { c8^"\\tuplet 3/2 4 {...}" c c c c c }
}
```

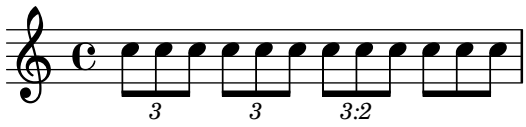


Cambiar el número del grupo especial

De forma predeterminada sólo se imprime sobre el corchete de grupo el numerador del grupo especial. De forma alternativa, se puede imprimr un quebrado en la forma numerador:denominador del número del grupo, o eliminar el número.

```
\relative c'' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
```

```
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```
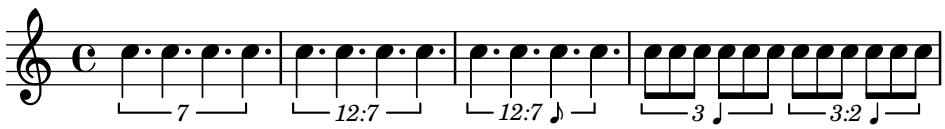


Números de agrupación especial distintos a los predeterminados
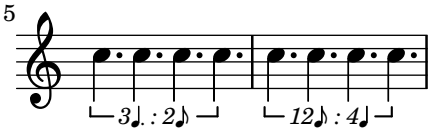
LilyPond también proporciona funciones de formato para imprimir números de grupo especial diferentes a la propia fracción, así como para añadir una figura al número o a la fracción de la agrupación.

```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7) "8")
  \tuplet 3/2  { c4. c4. c4. c4. }

  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text "4")
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text "4")
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }

  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes "4." "8")
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12 "8" 4 "4")
  \tuplet 3/2  { c4. c4. c4. c4. }
}
```



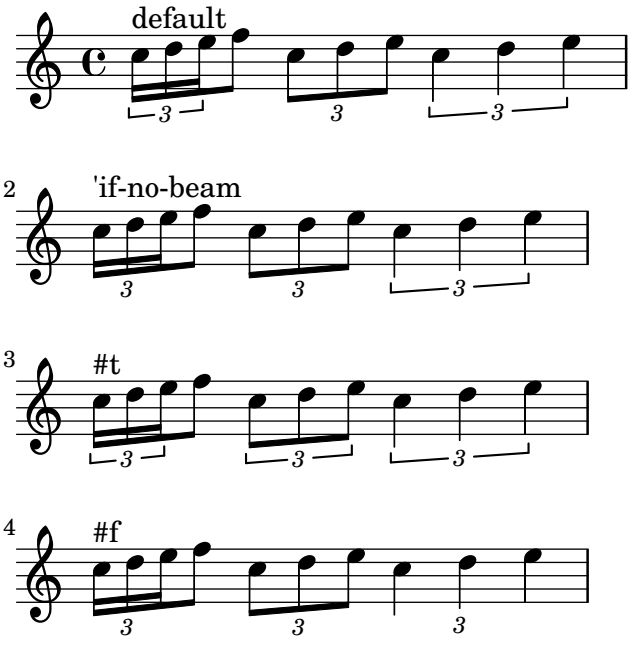


Controlar la visibilidad de los corchetes de grupo especial

El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial. Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `'bracket-visibility` a `#t` (imprimir el corchete siempre), `#f` (no imprimirlo nunca) o `#'if-no-beam` (imprimir el corchete solamente si no hay barra).

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    << \music s4^"default" >>
    \override TupletBracket.bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket.bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket.bracket-visibility = ##f
    << \music s4^"#f" >>
  }
}
```

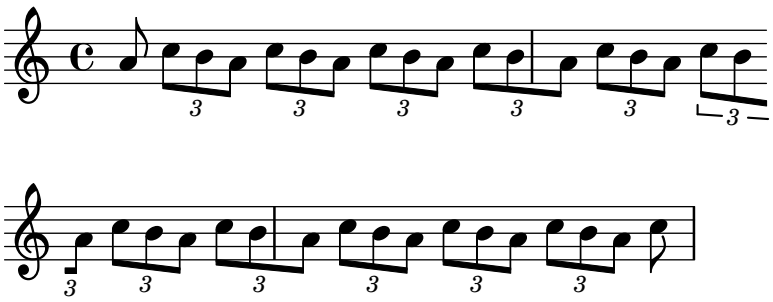


Permitir saltos de línea dentro de grupos especiales con barra

Este ejemplo artificial muestra cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra. Observe que estos grupos sincopados se deben barrar manualmente.

```
\layout {
  \context {
    \Voice
    % Permit line breaks within tuplets
```

```
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks
    \override Beam.breakable = ##t
  }
}
\relative c'' {
  a8
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  % Insert a manual line break within a tuplet
  \tuplet 3/2 { c[ b \bar "" \break a] }
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  c8
}
```



Véase también

Glosario musical: Sección “triplet” in *Glosario Musical*, Sección “tuplet” in *Glosario Musical*, Sección “polymetric” in *Glosario Musical*.

Manual de aprendizaje: Sección “Métodos de trucaje” in *Manual de Aprendizaje*.

Referencia de la notación: [Gestión del tiempo], página 114, [Escalar las duraciones], página 49, Sección 5.3.4 [La instrucción tweak], página 595, [Notación polimétrica], página 72.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TupletBracket” in *Referencia de Funcionamiento Interno*, Sección “TupletNumber” in *Referencia de Funcionamiento Interno*, Sección “TimeScaledMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las notas de adorno se pueden escribir dentro de los corchetes de grupo especial, *excepto* cuando un pentagrama comienza por una nota de adorno seguida de un grupo de valoración especial. En este caso concreto, se debe poner la nota de adorno antes de la instrucción `\tuplet` para evitar errores.

Cuando se utiliza un grupo de valoración especial al comienzo de una pieza con una indicación de `\tempo`, la música se debe escribir explícitamente dentro de un bloque `\new Voice`, tal y como se explica en Sección “Las voces contienen música” in *Manual de Aprendizaje*.
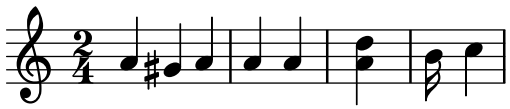
Escalar las duraciones

La duración de las figuras, silencios o acordes se puede multiplicar por un factor `N/M` añadiendo `*N/M` (o `*N` si `M` es 1) a la duración. Esto no afectará a la apariencia de las notas o silencios que se producen, pero la duración alterada se usará para calcular la posición dentro del compás y para establecer la duración en la salida MIDI. Los factores de multiplicación se pueden combinar en la forma `*L*M/N`. Los factores son parte de la duración: si no especificamos una duración para

las notas siguientes, la duración por omisión que se toma de la nota anterior incluirá cualquier factor de escala que se haya aplicado.

En el siguiente ejemplo las tres primeras notas duran exactamente dos partes, pero no se imprime ningún corchete de tresillo.

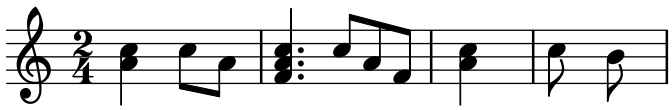
```
\time 2/4
% Alter durations to triplets
a4*2/3 gis a
% Normal durations
a4 a
% Double the duration of chord
<a d>4*2
% Duration of quarter, appears like sixteenth
b16*4 c4
```



La duración de los silencios espaciadores también se puede modificar mediante un multiplicador. Esto es útil para saltar muchos compases, como por ejemplo `s1*23`.

De la misma forma, se pueden comprimir por una fracción trozos de música más largos, como si cada nota, acorde o silencio tuviera la fracción como multiplicador. Esto dejará intacta la apariencia de la *música*, pero la duración interna de las notas se multiplicará por la fracción *numerador*/*denominador*. He aquí un ejemplo que muestra cómo se puede comprimir y expandir la música:

```
\time 2/4
% Normal durations
<c a>4 c8 a
% Scale music by *2/3
\scaleDurations 2/3 {
  <c a f>4. c8 a f
}
% Scale music by *2
\scaleDurations 2/1 {
  <c' a>4 c8 b
}
```



Una aplicación de esta instrucción se encuentra en la notación polimétrica, véase [Notación polimétrica], página 72.

Véase también

Referencia de la notación: [Grupos especiales], página 45, [Silencios invisibles], página 56, [Notación polimétrica], página 72.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Advertencias y problemas conocidos

El cálculo de la posición dentro de un compás sdebe tener en cuenta todos los factores de escalado que se han aplicado a las notas dentro de ese compás, y cualquier acarreo fraccionado desde los compases anteriores. Este cálculo se efectúa utilizando números racionales. Si un numerador o denominador intermedio en dicho proceso de cálculo sobrepasa la cantidad de 2^30, la ejecución y el tipografiado se detendrán en ese punto sin indicar ningún error.
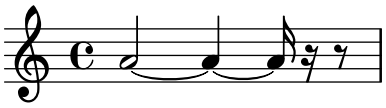
Ligaduras de unión

Una ligadura de unión conecta dos notas adyacentes de la misma altura. La ligadura en efecto extiende la longitud de una nota.

> **Nota:** No deben confundirse las ligaduras de unión con las *ligaduras de expresión*, que indican articulación, ni con las *ligaduras de fraseo*, que indican el fraseo musical. Una ligadura de unión es tan sólo una manera de extender la duración de una nota, algo parecido a lo que hace el puntillo.
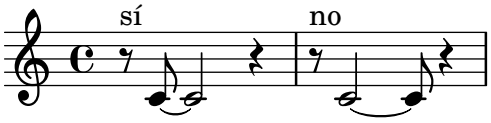
Se introduce una ligadura de unión escribiendo el símbolo de la tilde curva (~) después de la primera de cada pareja de notas que se quieren unir. Esto indica que la nota se une a la siguiente, que debe tener la misma altura.

```
a2~ a4~ a16 r r8
```



Se usan ligaduras de unión bien cuando la nota atraviesa la barra de compás o bien cuando no se pueden usar puntillos para denotar el ritmo. También se deben usar ligaduras cuando las notas atraviesan subdivisiones del compás de mayor duración:

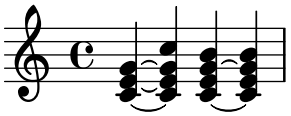
```
\relative c' {
  r8^"sí" c~ c2 r4 |
  r8^"no" c2~ c8 r4
}
```



Si necesitamos ligar muchas notas a través de las líneas divisorias, nos podría resultar más fácil utilizar la división automática de las notas, véase [División automática de las notas], página 75. Este mecanismo divide automáticamente las notas largas y las liga a través de las barras de compás.

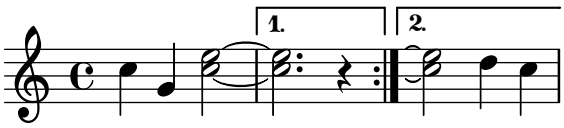
Cuando se aplica una ligadura de unión a un acorde, se conectan todas las cabezas de las notas cuyas alturas coinciden. Si no coincide ningún par de cabezas, no se crea ninguna ligadura. Los acordes se pueden ligar parcialmente colocando las ligaduras dentro del acorde.
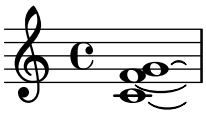
```
<c e g>~ <c e g c>
<c~ e g~ b> <c e g b>
```



Cuando la segunda vez de una repetición comienza con una nota ligada, es necesario especificar la ligadura repetida como sigue:

```
\repeat volta 2 { c g <c e>2~ }
\alternative {
  % First alternative: following note is tied normally
  { <c e>2. r4 }
  % Second alternative: following note has a repeated tie
  { <c e>2\repeatTie d4 c } }
```



Las ligaduras *L.v.* (*laissez vibrer*, dejar vibrar) indican que las notas no se deben apagar al final. Se usan en la notación para piano, arpa y otros instrumentos de cuerda y percusión. Se pueden introducir de la siguiente manera:
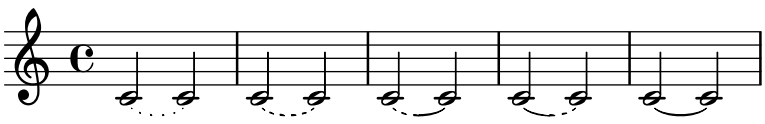
```
<c f g>1\laissezVibrer
```



Es posible hacer manualmente que las ligaduras de unión se curven hacia arriba o hacia abajo; véase Sección 5.4.2 [Dirección y posición], página 601.
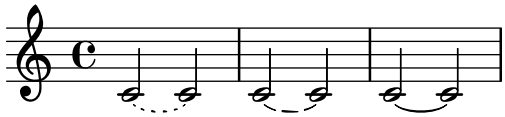
Se puede hacer que las ligaduras de expresión sean discontinuas, punteadas o una combinación de continuas y discontinuas.

```
\tieDotted
c2~ c
\tieDashed
c2~ c
\tieHalfDashed
c2~ c
\tieHalfSolid
c2~ c
\tieSolid
c2~ c
```



Se pueden especificar patrones de discontinuidad personalizados:
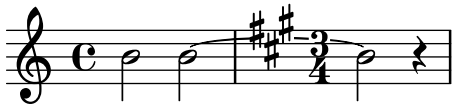
```
\tieDashPattern #0.3 #0.75
c2~ c
\tieDashPattern #0.7 #1.5
c2~ c
\tieSolid
c2~ c
```



Las definiciones de patrones de discontinuidad para las ligaduras de unión tienen la misma estructura que las definiciones de patrones de discontinuidad para las ligaduras de expresión. Para ver más información acerca de los patrones de discontinuidad complejos, consulte los fragmentos de código bajo [Ligaduras de expresión], página 127.

Sobreescriba las propiedades de disposición *whiteout* y *layer* para las ligaduras de unión que colisionen con otros objetos del pentagrama.

```
\override Tie.layer = #-2
\override Staff.TimeSignature.layer = #-1
\override Staff.KeySignature.layer = #-1
\override Staff.TimeSignature.whiteout = ##t
\override Staff.KeySignature.whiteout = ##t
b2 b~
\time 3/4
\key a \major
b r4
```



Instrucciones predefinidas
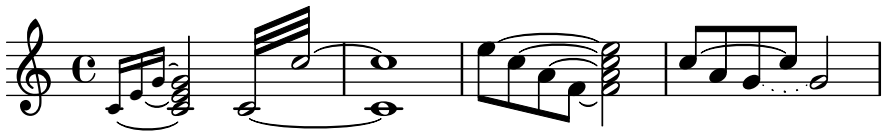
`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`, `\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Fragmentos de código seleccionados

Uso de ligaduras en los arpegios

En ocasiones se usan ligaduras de unión para escribir los arpegios. En este caso, las dos notas ligadas no tienen que ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` al valor `#t`. La misma funcionalidad es de utilidad, por ejemplo, para ligar un trémolo a un acorde, pero en principio también se puede usar para notas normales consecutivas.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Grabado manual de las ligaduras

Se pueden grabar a mano las ligaduras modificando la propiedad `tie-configuration` del objeto `TieColumn`. El primer número indica la distancia a partir de la tercera línea del pentagrama en espacios de pentagrama, y el segundo número indica la dirección (1 = hacia arriba, -1 = hacia abajo).

```
\relative c' {
  <c e g>2~ <c e g>
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>2~ <c e g>
}
```



Véase también

Glosario musical: Sección “tie” in *Glosario Musical*, Sección “laissez vibrer” in *Glosario Musical*.

Referencia de la notación: [Ligaduras de expresión], página 127, [División automática de las notas], página 75.

Fragmentos de código: Sección “slurs” in *Fragmentos de código*, Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LaissezVibrerTie” in *Referencia de Funcionamiento Interno*, Sección “LaissezVibrerTieColumn” in *Referencia de Funcionamiento Interno*, Sección “TieColumn” in *Referencia de Funcionamiento Interno*, Sección “Tie” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Un cambio de pentagrama cuando hay una ligadura activa no producirá una ligadura inclinada.

Los cambios de clave o de octava durante una ligadura de unión no están bien definidos realmente. En estos casos puede ser preferible una ligadura de expresión.

1.2.2 Escritura de silencios

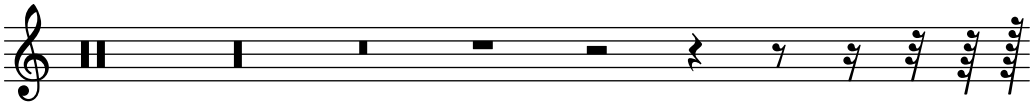
Los silencios se escriben como parte de la música dentro de las expresiones musicales.

Silencios

Los silencios se introducen como notas con el nombre `r`. Las duraciones mayores que la redonda utilizan las instrucciones predefinidas que se muestran aquí:

```
\new Staff {
  % These two lines are just to prettify this example
  \time 16/1
  \omit Staff.TimeSignature
  % Print a maxima rest, equal to four breves
  r\maxima
  % Print a longa rest, equal to two breves
  r\longa
  % Print a breve rest
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
```

```
}
```



Los silencios de un compás, centrados en medio del compás, se deben hacer con silencios multicompás. Se pueden usar para un solo compás así como para muchos compases, y se tratan en [Silencios de compás completo], página 58.

Para especificar explícitamente la posición vertical de un silencio, escriba una nota seguida de `\rest`. Se colocará un silencio en la posición en que debería aparecer la nota. Esto posibilita la aplicación manual precisa de formato a la música polifónica, ya que el formateador automático de colisiones de silencios no mueve estos silencios.

```
a4\rest d4\rest
```



Fragmentos de código seleccionados

Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
\layout {
  indent = 0
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}

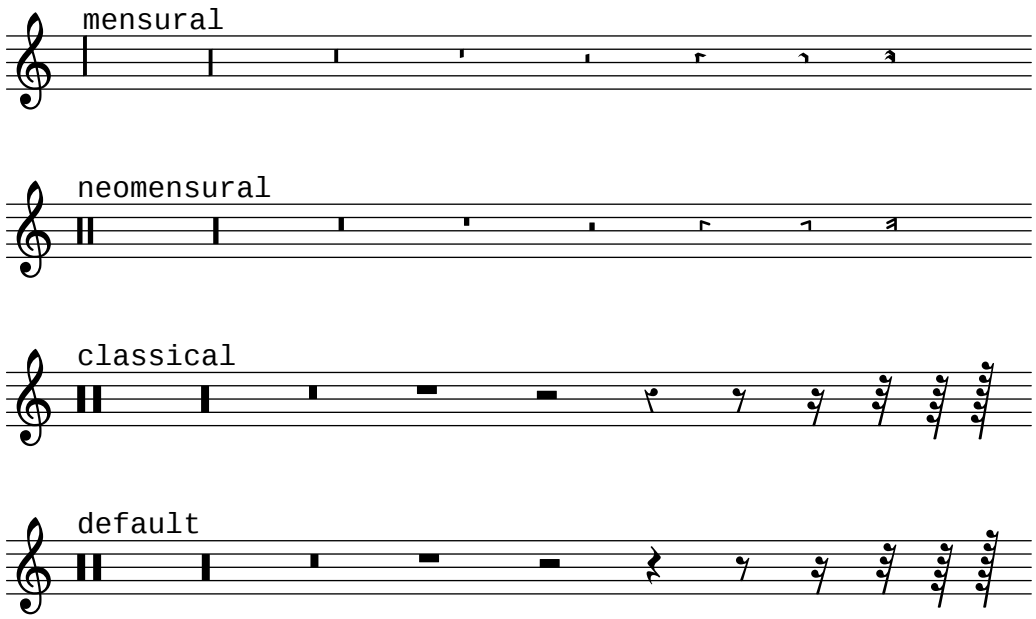
\new Staff \relative c {
  \cadenzaOn
  \override Staff.Rest.style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest.style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest.style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""

  \override Staff.Rest.style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
```

```
}
```



Véase también

Glosario musical: Sección “breve” in *Glosario Musical*, Sección “longa” in *Glosario Musical*, Sección “maxima” in *Glosario Musical*.

Referencia de la notación: [Silencios de compás completo], página 58.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Rest” in *Referencia de Funcionamiento Interno*.
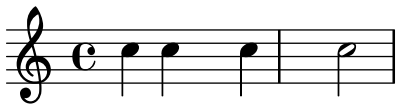
Advertencias y problemas conocidos

No existe un límite fundamental respecto de las duraciones de los silencios (tanto para el más corto como para el más largo), pero el número de glifos es limitado: hay silencios desde la garrapatea (128) hasta la máxima (8 redondas).

Silencios invisibles
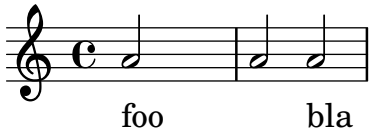
Un silencio invisible (también conocido como ‘skip’ o desplazamiento) se puede introducir como una nota con el nombre `s`:

```
c4 c s c
s2 c
```



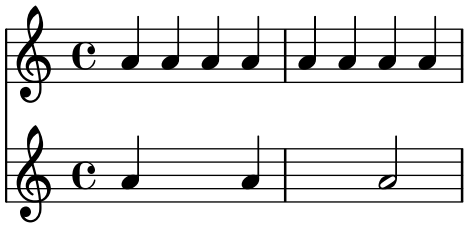
Los silencios de separación sólo están disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, se usa la instrucción `\skip` para producir un desplazamiento de una cierta magnitud temporal. `\skip` requiere una duración explícita, pero se ignora si las sílabas de la letra toman sus duraciones de las notas de una melodía asociada, a través de `\addlyrics` o de `\lyricsto`.

```
<<
  {
    a2 \skip2 a2 a2
  }
  \new Lyrics {
    \lyricmode {
      foo2 \skip 1 bla2
    }
  }
>>
```



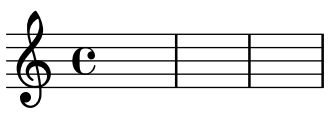
Dado que `\skip` es una instrucción, no afecta a las duraciones por omisión de las notas siguientes, a diferencia de `s`.

```
<<
  {
    \repeat unfold 8 { a4 }
  }
  {
    a4 \skip 2 a |
    s2 a
  }
>>
```



Un silencio de separación produce implícitamente contextos `Staff` y `Voice` si no existe ninguno, igual que las notas y los silencios normales:

```
s1 s s
```



`\skip` tan sólo desplaza un tiempo musical; no produce ninguna salida, de ninguna clase.

```
% This is valid input, but does nothing
\skip 1 \skip1 \skip 1
```

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*.

Referencia de la notación: [Notas ocultas], página 216, Sección 5.4.6 [Visibilidad de los objetos], página 608.

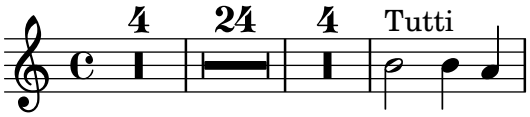
Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SkipMusic” in *Referencia de Funcionamiento Interno*.

Silencios de compás completo

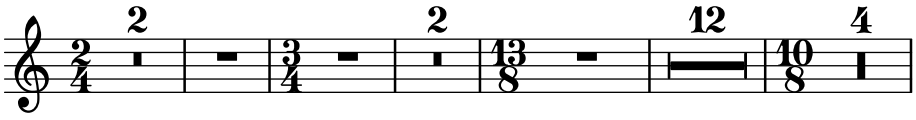
Los silencios de uno o más compases completos se introducen como notas con el nombre `R` en mayúscula:

```
% Rest measures contracted to single measure
\compressFullBarRests
R1*4
R1*24
R1*4
b2^"Tutti" b4 a4
```
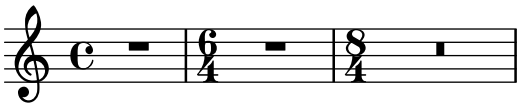


La duración de los silencios de compás completo es idéntica a la notación de la duración que se usa para las notas. La duración de un silencio multi-compás debe ser siempre un número entero de compases, por lo que con frecuencia han de utilizarse puntillos o fracciones:

```
\compressFullBarRests
\time 2/4
R1 | R2 |
\time 3/4
R2. | R2.*2 |
\time 13/8
R1*13/8 | R1*13/8*12 |
\time 10/8
R4*5*4 |
```



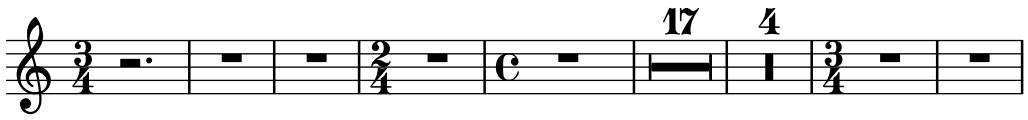
Un silencio de un compás completo se imprime como un silencio de redonda o de breve, centrado en el compás, según el tipo de compás vigente.

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```
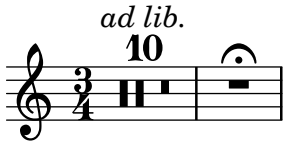


De forma predeterminada un silencio multicompás se expande en la partitura impresa para mostrar explícitamente todos los compases de silencio. De forma alternativa, un silencio multicompás se puede presentar como un solo compás que contiene un símbolo de silencio multicompás, con el número de compases impreso encima del compás:

```
% Default behavior
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Rest measures contracted to single measure
\compressFullBarRests
r1 | R1*17 | R1*4 |
% Rest measures expanded
\expandFullBarRests
\time 3/4
R2.*2 |
```



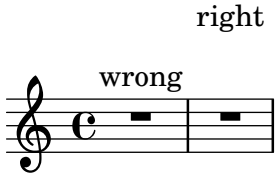
Se pueden añadir elementos de marcado a los silencios multicompás. Se proporciona la instrucción predefinida `\fermataMarkup` para añadir calderones.

```
\compressFullBarRests
\time 3/4
R2.*10^\markup { \italic "ad lib." }
R2.^\fermataMarkup
```



> **Nota:** Los elementos de marcado que se añaden a un silencio multicompás son objetos del tipo `MultiMeasureRestText`, no `TextScript`. Las sobreescrituras de propiedades deben ir dirigidas hacia el objeto correcto, o se ignorarán. Véase el ejemplo siguiente:

```
% This fails, as the wrong object name is specified
\override TextScript.padding = #5
R1^"wrong"
% This is the correct object name to be specified
\override MultiMeasureRestText.padding = #5
R1^"right"
```



Cuando un silencio multicompás sigue inmediatamente al establecimiento de un compás parcial con `\partial`, es posible que no se emitan las advertencias correspondientes de comprobación de compás.

Instrucciones predefinidas

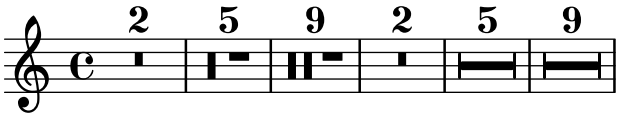
`\textLengthOn`, `\textLengthOff`, `\fermataMarkup`, `\compressFullBarRests`, `\expandFullBarRests`.

Fragmentos de código seleccionados

Cambiar la forma de los silencios multicompás

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como "Kirchenpausen", «silencios eclesiásticos»); en caso contrario se muestra una barra normal. Este número predeterminado de diez se puede cambiar sobreescribiendo la propiedad `expand-limit`:

```
\relative c'' {
  \compressFullBarRests
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest.expand-limit = #3
  R1*2 | R1*5 | R1*9
}
```



Posicionar los silencios multicompás

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predeterminada de un símbolo de silencio multicompás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicompás de las voces de numeración par e impar están separados verticalmente. La colocación de los silencios multicompás se puede controlar como se ve a continuación:

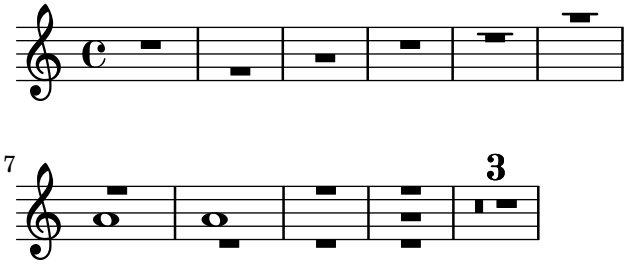
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line
  R1
  % They can be moved using an override
  \override MultiMeasureRest.staff-position = #-2
  R1
  \override MultiMeasureRest.staff-position = #0
  R1
  \override MultiMeasureRest.staff-position = #2
  R1
  \override MultiMeasureRest.staff-position = #3
  R1
  \override MultiMeasureRest.staff-position = #6
  R1
  \revert MultiMeasureRest.staff-position
  \break

  % In two Voices, odd-numbered voices are under the top line
  << { R1 } \\ { a1 } >>
```

```
  % Even-numbered voices are under the bottom line
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override
  << { R1 } \\ { R1 } \\
     \once \override MultiMeasureRest.staff-position = #0
     { R1 }
  >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed
  \compressFullBarRests
  <<
   \revert MultiMeasureRest.direction
    { R1*3 }
    \\
   \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```

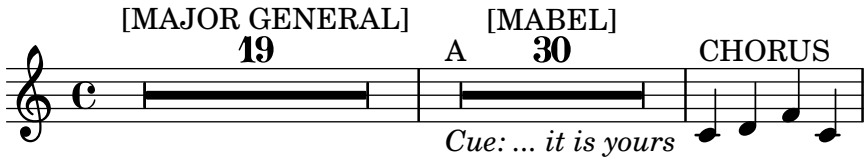


Marcado de silencios multicompás

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un acorde vacío con un marcado aplicado antes del silencio multicompás.

El texto aplicado a un siencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.

```
\relative c' {
  \compressFullBarRests
  \textLengthOn
  <>^\markup { [MAJOR GENERAL] }
  R1*19
  <>_\markup { \italic { Cue: ... it is yours } }
  <>^\markup { A }
  R1*30^\markup { [MABEL] }
  \textLengthOff
  c4^\markup { CHORUS } d f c
}
```



Véase también

Glosario musical: Sección “multi-measure rest” in *Glosario Musical*.

Referencia de la notación: Sección 1.2 [Duraciones], página 42, Sección 1.8 [Texto], página 225, Sección 1.8.2 [Formatear el texto], página 233, [Guiones de texto], página 226.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MultiMeasureRest” in *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestNumber” in *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestText” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

En las digitaciones encima de silencios multicompás (p. ej., `R1*10-4`), la cifra de la digitación puede chocar con el número del compás.

No hay ninguna forma de condensar automáticamente muchos silencios en un solo silencio multicompás.
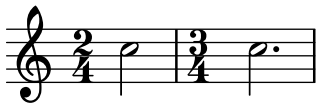
Los silencios multicompás no toman parte en las colisiones de silencios.

1.2.3 Impresión de las duraciones

Indicación de compás
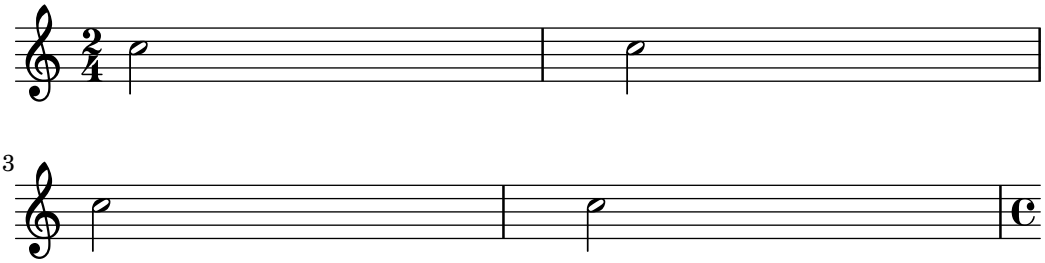
La indicación de compás se establece como sigue:
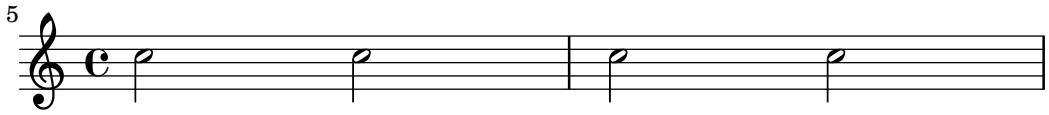
```
\time 2/4 c2
\time 3/4 c2.
```



La indicación de compás se imprime al comienzo de una pieza y siempre que hay un cambio de compás. Si se produce un cambio al final de una línea, se imprime una indicación de advertencia en dicho lugar. Se puede modificar este comportamiento predeterminado, véase Sección 5.4.6 [Visibilidad de los objetos], página 608.
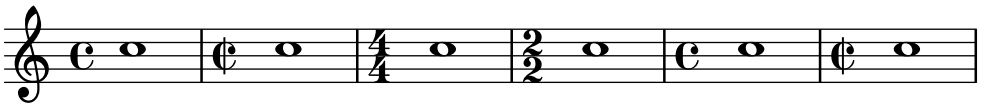
```
\time 2/4
c2 c
\break
c c
\break
\time 4/4
c c c c
```





El símbolo de compás que se usa en 2/2 y 4/4 se puede cambiar a un estilo numérico:

```
% Default style
\time 4/4 c1
\time 2/2 c1
% Change to numeric style
\numericTimeSignature
\time 4/4 c1
\time 2/2 c1
% Revert to default style
\defaultTimeSignature
\time 4/4 c1
\time 2/2 c1
```
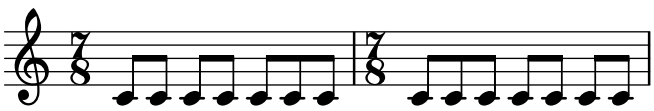


Las indicaciones de compás de la música mensural se tratan en [Indicaciones de compás de la música mensural], página 426.

Además de ajustar la indicación de compás que se imprime, la instrucción `\time` establece también los valores de las propiedades basadas en el tipo de compás `baseMoment`, `beatStructure` y `beamExceptions`. Los valores predefinidos de estas propiedades están en '`scm/time-signature-settings.scm`'.

El valor predeterminado de `beatStructure` puede sobreescribirse dentro de la propia instrucción `\time` escribiéndolo como primer argumento opcional:
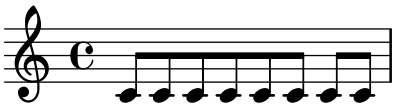
```
\score {
  \new Staff {
    \relative c' {
      \time #'(2 2 3) 7/8
      \repeat unfold 7 { c8 } |
      \time #'(3 2 2) 7/8
      \repeat unfold 7 { c8 } |
    }
  }
}
```



De forma alternativa, los valores predeterminados de todoas estas variables basadas en la indicación de compás, incluidas `baseMoment` y `beamExceptions`, se pueden establecer juntas. Los valores se pueden fijar independientemente para varios tipos de compás distintos. Los nuevos valores tienen efecto cuando se ejecuta una instrucción `\time` posterior con el mismo valor del tipo de compás:

```
\score {
  \new Staff {
    \relative c' {
      \overrideTimeSignatureSettings
```

```
        4/4         % timeSignatureFraction
        1/4         % baseMomentFraction
        #'(3 1)     % beatStructure
        #'()        % beamExceptions
      \time 4/4
      \repeat unfold 8 { c8 } |
    }
  }
}
```
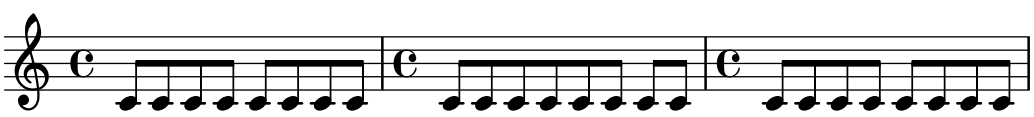


`\overrideTimeSignatureSettings` toma cuatro argumentos:

1. *`timeSignatureFraction`*, una fracción que describe el tipo de compás al que se aplican estos valores.
2. *`baseMomentFraction`*, una fracción que contiene el numerador y denominador de la unidad de medida básica de ese tipo de compás.
3. *`beatStructure`*, una lista de Scheme que indica la estructura de los pulsos del compás, en unidades del momento base.
4. *`beamExceptions`*, una lista-A que contiene cualesquiera reglas de barrado para el tipo de compás que vayan más allá de terminar en cada pulso, como se describe en [Establecer el comportamiento de las barras automáticas], página 81.

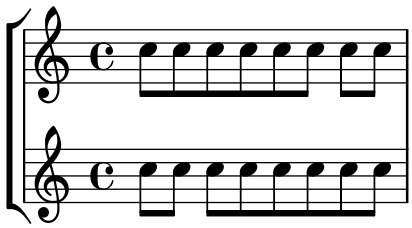
Los valores modificados de las propiedades predeterminadas del tipo de compás se pueden restaurar a los valores originales:

```
\score{
  \relative c' {
    \repeat unfold 8 { c8 } |
    \overrideTimeSignatureSettings
      4/4         % timeSignatureFraction
      1/4         % baseMomentFraction
      #'(3 1)     % beatStructure
      #'()        % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings 4/4
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}
```



Se pueden establecer diferentes valores de las propiedades predeterminadas del tipo de compás para los distintos pentagramas moviendo el `Timing_translator` y el `Default_bar_line_engraver` del contexto `Score` al contexto `Staff`.

```
\score {
  \new StaffGroup <<
     \new Staff {
        \overrideTimeSignatureSettings
          4/4        % timeSignatureFraction
          1/4        % baseMomentFraction
          #'(3 1)    % beatStructure
          #'()       % beamExceptions
        \time 4/4
        \repeat unfold 8 {c''8}
     }
     \new Staff {
        \overrideTimeSignatureSettings
          4/4        % timeSignatureFraction
          1/4        % baseMomentFraction
          #'(1 3)    % beatStructure
          #'()       % beamExceptions
        \time 4/4
        \repeat unfold 8 {c''8}
     }
  >>
  \layout {
    \context {
      \Score
      \remove "Timing_translator"
      \remove "Default_bar_line_engraver"
    }
    \context {
      \Staff
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    }
  }
}
```



Otro método para cambiar estas variables relacionadas con el tipo de compás, que evita la reimpresión de la indicación de compás en el momento del cambio, se muestra en [Establecer el comportamiento de las barras automáticas], página 81.

Instrucciones predefinidas
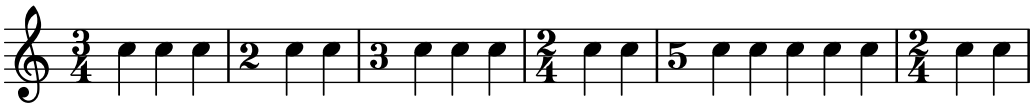
`\numericTimeSignature`, `\defaultTimeSignature`.

Fragmentos de código seleccionados

Indicación de compás imprimiendo sólo el numerador (en lugar de la fracción)

A veces, la indicación de compás no debe imprimir la fracción completa (p.ej. 7/4), sino sólo el numerador (7 en este caso). Esto se puede hacer fácilmente utilizando `\override Staff.TimeSignature.style = #'single-digit` para cambiar el estilo permanentemente. Usando `\revert Staff.TimeSignature.style`, se puede revertir el cambio. Para aplicar el estilo de un dígito único a una sola indicación de compás, utilice la instrucción `\override` y anteponga la instrucción `\once`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-digit
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-digit style only for the next time signature
  \once \override Staff.TimeSignature.style = #'single-digit
  \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Véase también

Glosario musical: Sección “time signature” in *Glosario Musical*

Referencia de la notación: [Indicaciones de compás de la música mensural], página 426, [Establecer el comportamiento de las barras automáticas], página 81, [Gestión del tiempo], página 114.

Archivos instalados: ‘`scm/time-signature-settings.scm`’.
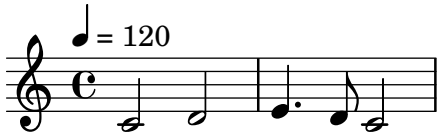
Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” in *Referencia de Funcionamiento Interno*, Sección “Timing_translator” in *Referencia de Funcionamiento Interno*.

Indicaciones metronómicas
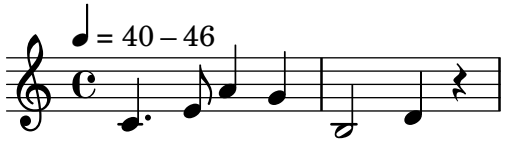
Es muy sencillo escribir una indicación metronómica básica:

```
\tempo 4 = 120
c2 d
e4. d8 c2
```
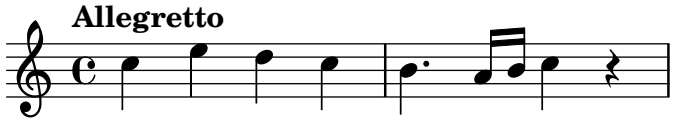


También se pueden imprimir indicaciones metronómicas como un intervalo entre dos números:

```
\tempo 4 = 40 - 46
c4. e8 a4 g
b,2 d4 r
```
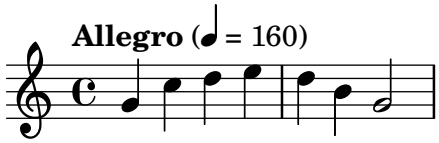


Se pueden usar indicaciones de tempo con texto:

```
\tempo "Allegretto"
c4 e d c
b4. a16 b c4 r4
```
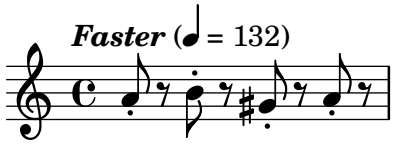


La combinación de una indicación metronómica y un texto hace que la marca de metrónomo se encierre entre paréntesis automáticamente:

```
\tempo "Allegro" 4 = 160
g4 c d e
d4 b g2
```
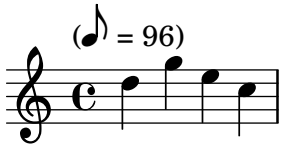


En general, el texto puede ser cualquier objeto de marcado:

```
\tempo \markup { \italic Faster } 4 = 132
a8-. r8 b-. r gis-. r a-. r
```
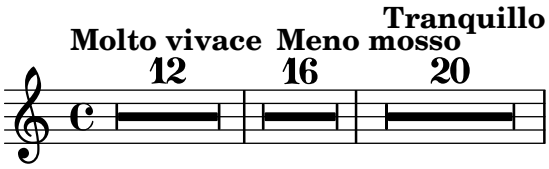


Se puede escribir una indicación metronómica entre paréntesis sin ninguna indicación textual, escribiendo una cadena vacía en la entrada:

```
\tempo "" 8 = 96
d4 g e c
```



En una particella de un instrumento con períodos de silencio largos, en ocasiones se suceden muy cerca distintas indicaciones de tempo. La instrucción `\markLengthOn` aporta un espacio horizontal adicional para evitar que las indicaciones de tempo se superpongan, y `\markLengthOff` restablece el comportamiento predeterminado de ignorar las indicaciones de tempo para el espaciado horizontal.

```
\compressFullBarRests
\markLengthOn
\tempo "Molto vivace"
R1*12
\tempo "Meno mosso"
R1*16
\markLengthOff
\tempo "Tranquillo"
R1*20
```



Fragmentos de código seleccionados

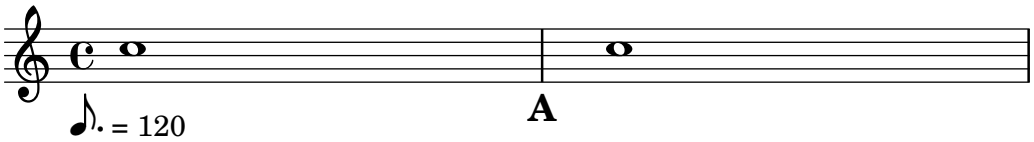
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout { ragged-right = ##f }

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```
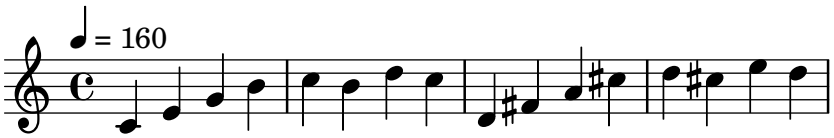


Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
```
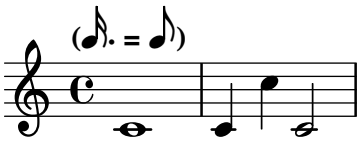
```
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Crear indicaciones metronómicas en modo de marcado

Se pueden crear indicaciones metronómicas nuevas en modo de marcado, pero no cambian el tempo en la salida MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note #"16." #1
      " = "
      \smaller \general-align #Y #DOWN \note #"8" #1
      )
    }
  }
  c1
  c4 c' c,2
}
```



Para ver más detalles, consulte Sección 1.8.2 [Formatear el texto], página 233.

Véase también

Glosario musical: Sección “metronome” in *Glosario Musical*, Sección “metronomic indication” in *Glosario Musical*, Sección “tempo indication” in *Glosario Musical*, Sección “metronome mark” in *Glosario Musical*.

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 233, Sección 3.5 [Salida MIDI], página 502.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno:: Sección “MetronomeMark” in *Referencia de Funcionamiento Interno*.
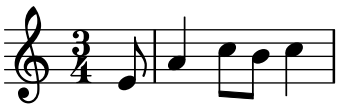
Anacrusas

Los compases parciales como las *anacrusas* o partes *al alzar* se escriben usando la instrucción `\partial`:

```
\partial duración
```
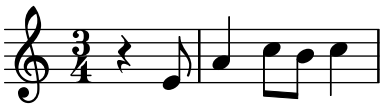
donde *`duración`* es la longitud *restante* del compás parcial *antes* del comienzo del siguiente compás completo.

```
\time 3/4
\partial 8
e8 | a4 c8 b c4 |
```



La *duración* puede ser cualquier valor menor de un compás completo:

```
\time 3/4
\partial 4.
r4 e8 | a4 c8 b c4 |
```
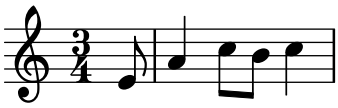


El `\partial` *`duración`* se puede también escribir como:

```
\set Timing.measurePosition -duración
```

Así que el primer ejemplo podría haberse escrito así:

```
\time 3/4
\set Timing.measurePosition = #(ly:make-moment -1/8)
e8 | a4 c8 b c4 |
```



La propiedad `measurePosition` contiene un número racional, que suele ser positivo y que indica qué porción del compás ha transcurrido hasta ahora en este punto. La instrucción `\partial` *`duración`* lo establece a un número negativo, en cuyo caso tiene un significado distinto: en esta ocasión indica que el compás actual (el primero) será *precedido* por un compás cero (el compás parcial) con una duración dada por el valor *duración*.

Véase también

Glosario musical: Sección “anacrusis” in *Glosario Musical*.

Referencia de la notación: [Notas de adorno], página 108.
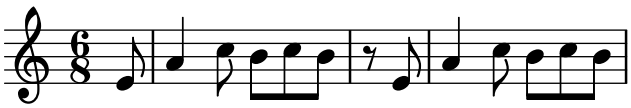
Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing_translator” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La instrucción `\partial` debe usarse solamente en el comienzo de una pieza. Si se emplea después del comienzo se obtendrán varias advertencias o efectos desagradables, en este caso utilice en su lugar `\set Timing.measurePosition`.
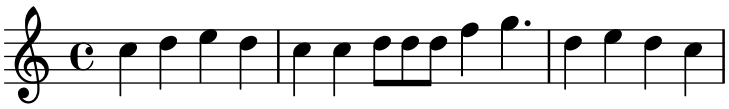
```
\time 6/8
\partial 8
e8 | a4 c8 b[ c b] |
\set Timing.measurePosition = #(ly:make-moment -1/4)
r8 e,8 | a4 c8 b[ c b] |
```



Música sin compasear
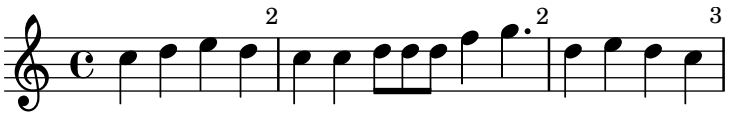
En la música medida se insertan líneas divisorias y se calculan los números de compás automáticamente. En música sin metro (es decir, cadencias), esto no es deseable y se puede 'desactivar' usando la instrucción `\cadenzaOn`, para despúes 'reactivarlo' en el lugar adecuado usando `\cadenzaOff`.

```
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```
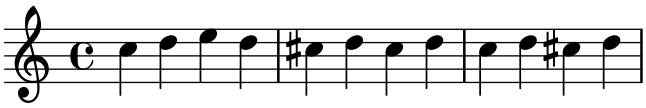


La numeración de compases se continúa al final de la cadencia.

```
% Show all bar numbers
\override Score.BarNumber.break-visibility = #all-visible
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```
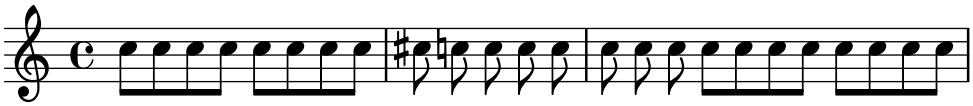


Al insertar una instrucción `\bar` dentro de una cadencia no se inicia un compás nuevo, incluso aunque se imprima una línea divisoria. Así pues, las alteraciones accidentales (cuyo efecto se suele suponer que permanece hasta el final del compás) serán válidas aún después de la línea divisoria que se imprime por parte de `\bar`. Si se quieren imprimir alteraciones accidentales posteriores tendrán que insertarse manualmente alteraciones forzadas o de cortesía, véase [Alteraciones accidentales], página 5.

```
c4 d e d
\cadenzaOn
cis4 d cis d
\bar "|"
% First cis is printed without alteration even if it's after a \bar
cis4 d cis! d
\cadenzaOff
\bar "|"
```



El barrado automático se desactiva mediante `\cadenzaOn`. Por tanto, todo el barrado en las cadencias se debe introducir manualmente. Véase [Barras manuales], página 90.

```
\repeat unfold 8 { c8 }
\cadenzaOn
cis8 c c c c
\bar"|"
c8 c c
\cadenzaOff
\repeat unfold 8 { c8 }
```



Estas instrucciones predefinidas afectan a todas las pautas de la partitura, aunque aparezcan en un solo contexto `Voice`. Para cambiar esto, traslade el `Timing_translator` del contexto `Score` al contexto `Staff`, véase [Notación polimétrica], página 72.

Instrucciones predefinidas

`\cadenzaOn`, `\cadenzaOff`.

Véase también

Glosario musical: Sección “cadenza” in *Glosario Musical*.

Referencia de la notación: Sección 5.4.6 [Visibilidad de los objetos], página 608, [Notación polimétrica], página 72, [Barras manuales], página 90, [Alteraciones accidentales], página 5.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Advertencias y problemas conocidos

Sólo se insertan saltos de línea y de página automáticamente en las líneas divisorias, por lo que es necesario insertar manualmente líneas divisorias ‘invisibles’ en fragmentos largos de música sin medir para permitir los saltos:

```
\bar ""
```

Notación polimétrica

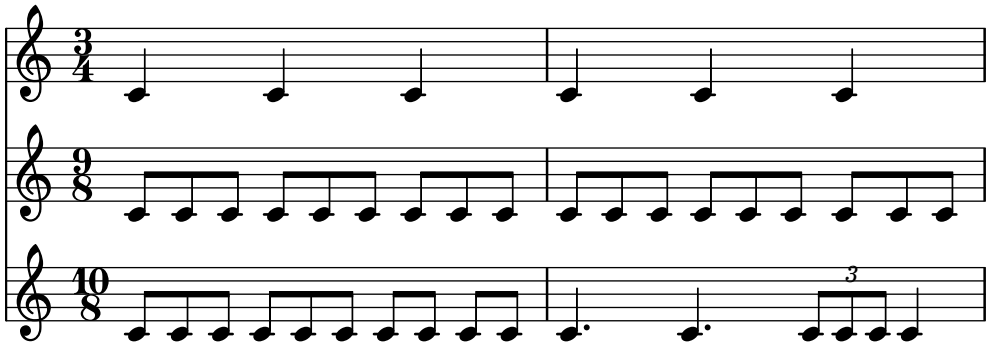
Está contemplada la notación polimétrica, ya sea explícitamente o mediante la modificación manual de la indicación visible del compás y/o el escalado de la duración de las notas.

Pentagramas con distintas indicaciones de compás y compases de igual longitud

Establezca una indicación de compás común para cada pentagrama, y fije `timeSignatureFraction` a la fracción deseada. Luego use la función `\scaleDurations` para escalar las duraciones de las notas en cada pauta a la indicación de compás común.

En el siguiente ejemplo, se usa en paralelo música con compases de 3/4, 9/8 y 10/8. En el segundo pentagrama, las duraciones mostradas se multiplican por 2/3 (pues 2/3 * 9/8 = 3/4), y en el tercer pentagrama, las duraciones que se muestran están multiplicadas por 3/5 (pues 3/5 * 10/8 = 3/4). Con frecuencia habrá que insertar las barras de forma manual, pues el escalado de las duraciones afecta a las reglas de barrado automático.

```
\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c4 c c |
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = 9/8
    \scaleDurations 2/3
      \repeat unfold 6 { c8[ c c] }
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = 10/8
    \scaleDurations 3/5 {
      \repeat unfold 2 { c8[ c c] }
      \repeat unfold 2 { c8[ c] } |
      c4. c \tuplet 3/2 { c8[ c c] } c4
    }
  }
>>
```



Pentagramas con distintas indicaciones de compás y longitudes de compás distintas

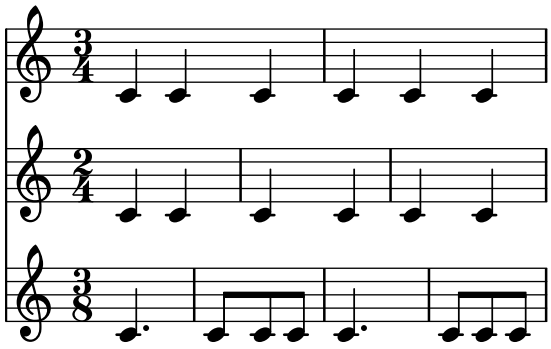
Se puede dar a cada pentagrama su propia indicación de compás independiente trasladando los grabadores `Timing_translator` y `Default_bar_line_engraver` al contexto de `Staff`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
```

```
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

% Now each staff has its own time signature.

\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c4 c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c4 c |
    c4 c |
  }
  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
>>
```



Indicaciones de compás compuesto
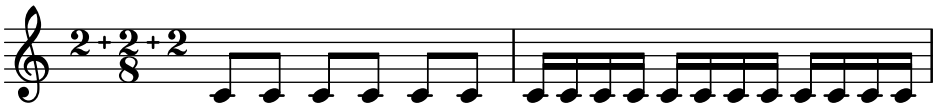
Se crean usando la función `\compoundMeter`. La sintaxis es:

```
\compoundMeter #'(lista de listas)
```

La construcción más simple es una lista única, en la que el *último* número indica el denominador de la indicación de compás y los anteriores son los numeradores.

```
\relative c' {
  \compoundMeter #'((2 2 2 8))
  \repeat unfold 6 c8 \repeat unfold 12 c16
```
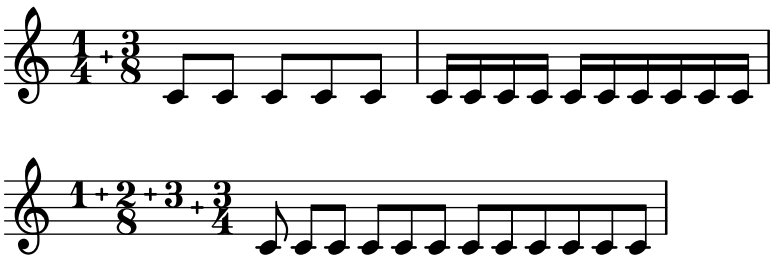
```
}
```



Se pueden construir compases más complejos utilizando listas adicionales. Asimismo, los ajustes de barrado se ajustarán dependiendo de los valores.

```
\relative c' {
  \compoundMeter #'((1 4) (3 8))
  \repeat unfold 5 c8 \repeat unfold 10 c16
}

\relative c' {
  \compoundMeter #'((1 2 3 8) (3 4))
  \repeat unfold 12 c8
}
```



Véase también

Glosario musical: Sección “polymetric” in *Glosario Musical*, Sección “polymetric time signature” in *Glosario Musical*, Sección “meter” in *Glosario Musical*.

Referencia de la notación: [Barras automáticas], página 79, [Barras manuales], página 90, [Indicación de compás], página 62, [Escalar las duraciones], página 49.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” in *Referencia de Funcionamiento Interno*, Sección “Timing_translator” in *Referencia de Funcionamiento Interno*, Sección “Default_bar_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos
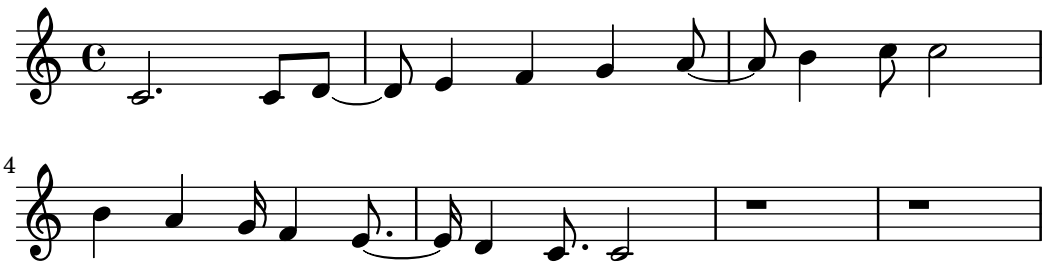
Al usar distintos compases en paralelo, las notas que están en el mismo instante de tiempo se colocan en la misma posición horizontal. Sin embargo, las barras de compás en los distintos pentagramas harán que el espaciado de notas sea menos regular en cada pentagrama individual de lo que sería normal sin las distintas indicaciones de compás.

División automática de las notas

Las notas largas se pueden convertir automáticamente en notas ligadas. Se hace mediante la sustitución del `Note_heads_engraver` por el `Completion_heads_engraver`. De forma similar, los silencios largos que sobrepasan líneas de compás se dividen automáticamente sustituyendo el grabador `Rest_engraver` con el grabador `Completion_rest_engraver`. En el ejemplo siguiente, las notas y los silencios que atraviesan la barra de compás se dividen, y además las notas se unen mediante una ligadura.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
  \remove "Rest_engraver"
  \consists "Completion_rest_engraver"
}

{ c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2 }
```



Estos grabadores dividen todas las notas y silencios largos en la barra de compás, e inserta ligaduras en las notas. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

Véase también

Glosario musical: Sección “tie” in *Glosario Musical*

Manual de aprendizaje: Sección “Explicación de los grabadores” in *Manual de Aprendizaje*, Sección “Añadir y eliminar grabadores” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Completion_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Rest_engraver” in *Referencia de Funcionamiento Interno*, Sección “Completion_rest_engraver” in *Referencia de Funcionamiento Interno*, Sección “Forbid_line_break_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No todas las duraciones (sobre todo las que contienen grupos especiales) se pueden representar exactamente con notas normales y puntillos, pero el grabador `Completion_heads_engraver` nunca insertará grupos especiales.

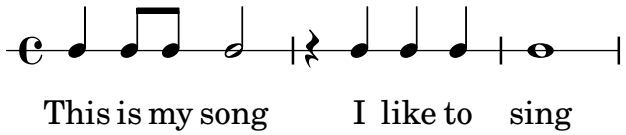
`Completion_heads_engraver` afecta sólo a las notas; no divide silencios.

Mostrar los ritmos de la melodía

A veces podemos querer mostrar solamente el ritmo de una melodía. Esto se puede hacer con un pentagrama de ritmo. Todas las alturas de las notas se convierten en barras inclinadas, y el pentagrama tiene una sola línea:

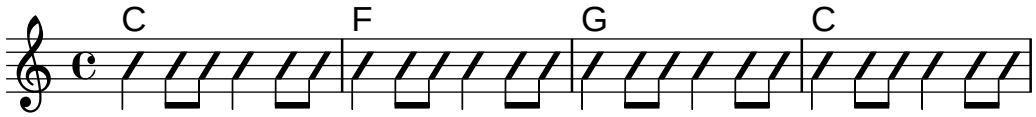
```
<<
  \new RhythmicStaff {
    \new Voice = "myRhythm" {
      \time 4/4
      c4 e8 f g2
      r4 g g f
      g1
    }
```

```
  }
  \new Lyrics {
    \lyricsto "myRhythm" {
      This is my song
      I like to sing
    }
  }
>>
```



Las tablas de acordes de guitarra ofrecen a menudo los ritmos de rasgueado. Esto se puede hacer con el grabador `Pitch_squash_engraver` y `\improvisationOn`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 f g c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
>>
```



Instrucciones predefinidas

`\improvisationOn`, `\improvisationOff`.

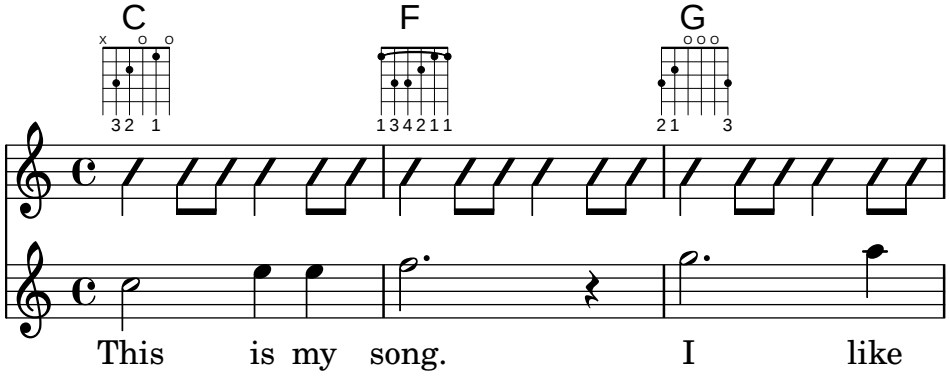
Fragmentos de código seleccionados

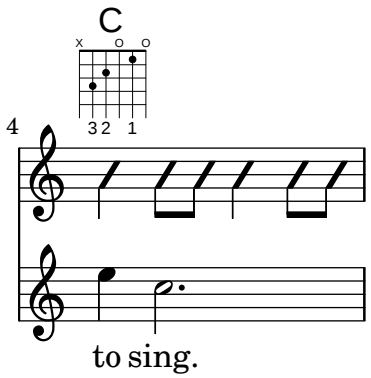
Ritmos rasgueados de guitarra

Para la música de guitarra, es posible mostrar los ritmos de rasgueo, además de las notas de la melodía, acordes y diagramas de posiciones.

```
\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
    \chordmode {
      c1 | f | g | c
    }
  }
```

```
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
    }
  }
  \new Voice = "melody" {
    \relative c'' {
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      This is my song.
      I like to sing.
    }
  }
>>
```





Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

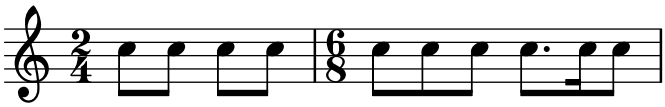
Referencia de funcionamiento interno: Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*, Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*.

1.2.4 Barras

Barras automáticas

De manera predeterminada, las barras de corchea se insertan automáticamente:
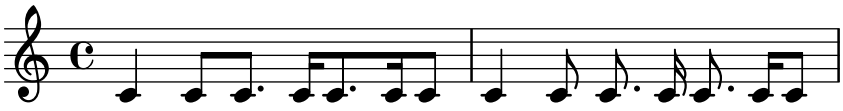
```
\time 2/4 c8 c c c
\time 6/8 c8 c c c8. c16 c8
```



Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita; véase [Barras manuales], página 90. Las barras *se deben* introducir manualmente si se quieren extender por encima de los silencios.

Si no se necesita el barrado automático, se puede desactivar con `\autoBeamOff` y activarse con `\autoBeamOn`:

```
c4 c8 c8. c16 c8. c16 c8
\autoBeamOff
c4 c8 c8. c16 c8.
\autoBeamOn
c16 c8
```



Nota: Si se usan barras para indicar los melismas de las canciones, entonces se debe desactivar el barrado automático con `\autoBeamOff` e indicar las barras manualmente. La utilización de `\partcombine` con `\autoBeamOff` puede producir resultados no deseados. Véanse los fragmentos de código para mayor información.

Se pueden crear patrones de barrado que difieran de los valores automáticos predeterminados; véase [Establecer el comportamiento de las barras automáticas], página 81.

Instrucciones predefinidas
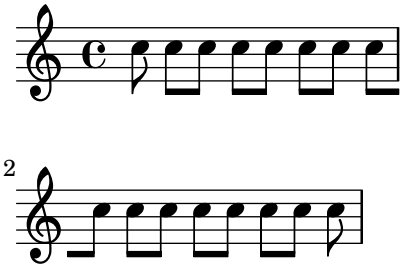
`\autoBeamOff`, `\autoBeamOn`.

Fragmentos de código seleccionados

Barras que atraviesan saltos de línea

Normalmente están prohibidos los saltos de línea si las barras atraviesan las líneas divisorias. Se puede cambiar este comportamiento como se muestra aquí:
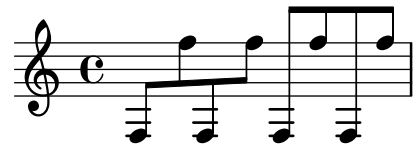
```
\relative c'' {
  \override Beam.breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}
```



Cambiar el salto de las barras en ángulo

Se insertan automáticamente barras en ángulo cuando se detecta un intervalo muy grande entre las notas. Se puede hacer un ajuste fino de este comportamiento a través de la propiedad `auto-knee-gap`. Se traza una barra doblada si el salto es mayor que el valor de `auto-knee-gap` más el ancho del objeto barra (que depende de la duración de las notas y de la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido a 5.5 espacios de pentagrama.

```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Partcombine y autoBeamOff

Puede ser difícil comprender la función de `\autoBeamOff` cuando se usa con `\partcombine`.

Puede ser preferible usar

`\set Staff.autoBeaming = ##f`

en su lugar, para asegurarse de que el barrado automático se desactiva para todo el pentagrama.
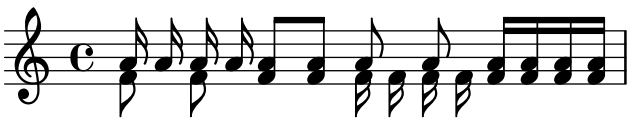
`\partcombine` funciona aparentemente con tres voces: plica arriba cuando hay una sola voz, plica abajo con la otra, y plica arriba cuando está combinada.

Una llamada a `\autoBeamOff` en el primer argumento de partcombine se aplica a la voz que está activa en el momento en que se procesa la llamada, ya sea voz única con la plica hacia

arriba, hacia abajo o combinadas. Una llamada a `\autoBeamOff` en el segundo argumento se aplica a la voz que está sola con la plica abajo.

Para pode usar `\autoBeamOff` con el objeto de detener todo el barrado automático cuando se usa con `\partcombine`, es necesario hacer tres llamadas a `\autoBeamOff`.

```
{
  %\set Staff.autoBeaming = ##f % turns off all autobeaming
  \partcombine
  {
    \autoBeamOff % applies to split up stems
    \repeat unfold 4 a'16
    %\autoBeamOff % applies to combined up stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
  }
  {
    \autoBeamOff % applies to down stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
  }
}
```



Véase también

Referencia de la notación: [Barras manuales], página 90, [Establecer el comportamiento de las barras automáticas], página 81.

Archivos instalados: ‘`scm/auto-beam.scm`’.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Auto_beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “Beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “Beam” in *Referencia de Funcionamiento Interno*, Sección “BeamEvent” in *Referencia de Funcionamiento Interno*, Sección “BeamForbidEvent” in *Referencia de Funcionamiento Interno*, Sección “beam-interface” in *Referencia de Funcionamiento Interno*, Sección “unbreakable-spanner-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las propiedades de una barra vienen determinadas al *comienzo* de su construcción y cualquier cambio adicional en las propiedades de la barra que se produzca antes de que la barra se ha completado no tendrá efecto hasta que inicie la *siguiente* barra nueva.

Establecer el comportamiento de las barras automáticas

Cuando está habilitado el barrado automático, la colocación de las barras automáticas viene determinada por tres propiedades de contexto: `baseMoment`, `beatStructure` y `beamExceptions`. Los valores predeterminados de estas variables se pueden sobreescribir como se describe más abajo, o de forma alternativa los propios valores predeterminados se pueden cambiar como se explica en [Indicación de compás], página 62.

Si hay definida una regla de `beamExceptions` para el compás en curso, se usa dicha regla para determinar la colocación de las barras; se ignoran los valores de `baseMoment` y `beatStructure`.

Si no está definida ninguna regla de `beamExceptions` para el tipo de compás en curso, la colocación de las barras está determinada por los valores de `baseMoment` y `beatStructure`.
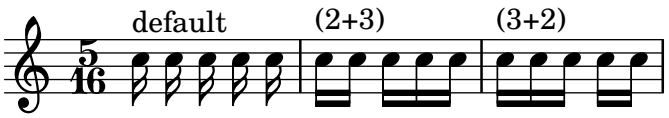
Barrado basado en `baseMoment` *y* `beatStructure`

De forma predeterminada, las reglas de `beamExceptions` están definidas para los compases más comunes, y así las reglas de `beamExceptions` se deben desactivar si pretendemos que el barrado automático esté basado en `baseMoment` y `beatStructure`. Las reglas de `beamExceptions` se desactivan mediante
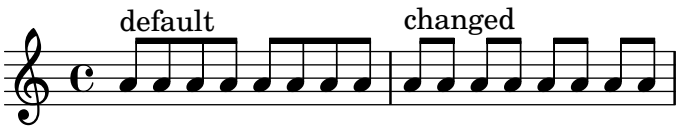
```
\set Timing.beamExceptions = #'()
```

Cuando el valor de `beamExceptions` se ha fijado a `#'()`, ya sea debido a un ajuste explícito o a causa de que no hay ninguna regla de `beamExceptions` definida internamente para el compás actual, los puntos finales de las barras están en los pulsos según viene determinado por las propiedades de contexto `baseMoment` y `beatStructure`. `beatStructure` es una lista de Scheme que define la longitud de cada pulso dentro del compás en unidades de `baseMoment`. De forma predeterminada, `baseMoment` es una unidad más que el denominador del compás. De forma predeterminada también, cada unidad de longitud `baseMoment` es un único pulso.

```
\time 5/16
c16^"default" c c c c |
% beamExceptions are unlikely to be defined for 5/16 time,
% but let's disable them anyway to be sure
\set Timing.beamExceptions = #'()
\set Timing.beatStructure = #'(2 3)
c16^"(2+3)" c c c c |
\set Timing.beatStructure = #'(3 2)
c16^"(3+2)" c c c c |
```



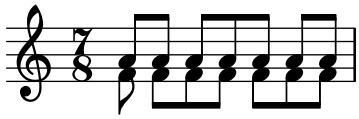

```
\time 4/4
a8^"default" a a a a a a a
% Disable beamExceptions because they are definitely
% defined for 4/4 time
\set Timing.beamExceptions = #'()
\set Timing.baseMoment = #(ly:make-moment 1/4)
\set Timing.beatStructure = #'(1 1 1 1)
a8^"changed" a a a a a a a
```



Los cambios en los ajustes de barrado se pueden limitar a contextos específicos. Si no se incluye ningún ajuste en un contexto de nivel más bajo, se aplican los ajustes del contexto circundante.

```
\new Staff {
  \time 7/8
  % No need to disable beamExceptions
  % as they are not defined for 7/8 time
```
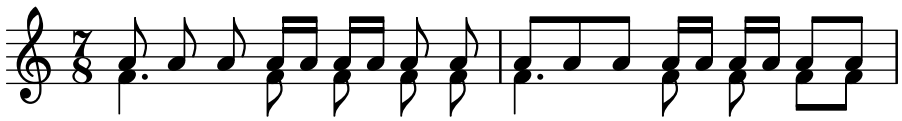
```
  \set Staff.beatStructure = #'(2 3 2)
  <<
    \new Voice = one {
      \relative c'' {
        a8 a a a a a a
      }
    }
    \new Voice = two {
      \relative c' {
        \voiceTwo
        \set Voice.beatStructure = #'(1 3 3)
        f8 f f f f f f
      }
    }
  >>
}
```



En caso de usar varias voces, se debe especificar el contexto `Staff` si queremos aplicar el barrado a todas las voces del pentagrama:
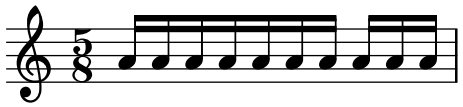
```
\time 7/8
% rhythm 3-1-1-2
% Change applied to Voice by default --  does not work correctly
% Because of autogenerated voices, all beating will
% be at baseMoment (1 . 8)
\set beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>
```



El valor de `baseMoment` se puede ajustar para cambiar el comportamiento de las barras, si se desea. Cuando se hace, el valor de `beatStructure` se debe fijar de manera que sea compatible con el nuevo valor de `baseMoment`.

```
\time 5/8
% No need to disable beamExceptions
% as they are not defined for 5/8 time
\set Timing.baseMoment = #(ly:make-moment 1/16)
\set Timing.beatStructure = #'(7 3)
\repeat unfold 10 { a16 }
```
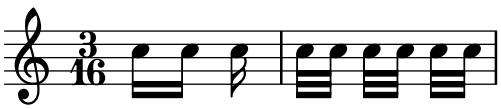


`baseMoment` es un *momento*, una unidad de duración musical. Se crea una cantidad del tipo *momento* por medio de la función de Scheme `ly:make-moment`. Para ver más información acerca de esta función, consulte [Gestión del tiempo], página 114.

De forma predeterminada `baseMoment` está fijado a una unidad más que el denominador del compás. Todas las excepciones a este valor predeterminado están en '`scm/time-signature-settings.scm`'.

Barrado basado en `beamExceptions`

Las reglas de autobarrado especiales (distintas de terminar una barra sobre un pulso) están definidas en la propiedad `beamExceptions`.

```
\time 3/16
\set Timing.beatStructure = #'(2 1)
\set Timing.beamExceptions =
  #'(                          ;start of alist
     (end .                    ;entry for end of beams
      (                        ;start of alist of end points
       ((1 . 32) . (2 2 2))    ;rule for 1/32 beams -- end each 1/16
      )))                      %close all entries
c16 c c |
\repeat unfold 6 { c32 } |
```



`beamExceptions` es una lista-A con una clave de rule-type y un valor de beaming-rules.

En este momento el único valor disponible de rule-type es `'end` para la terminación de las barras.

Beaming-rules es una lista-A de Scheme (o una lista de parejas) que indica el tipo de barra y la agrupación que se aplica a las barras que contienen notas con una duración más breve de ese tipo de barra.

```
#'((tipo-barra1 . agrupación1)
   (tipo-barra2 . agrupación2)
   (tipo-barra3 . agrupación3))
```

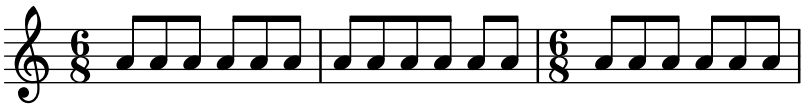
Tipo de barra es una dupla de Scheme que indica la duración de la barra, p.ej. `(1 . 16)`.

Agrupación es una lista de Scheme que indica la agrupación que aplicar a la barra. El valor de agrupación está en unidades del tipo de barra.

> **Nota:** Un valor de `beamExceptions` debe ser una lista de excepciones *completa*. Esto es, toda excepción que se tenga que aplicar debe estar incluida en este ajuste. No es posible añadir, eliminar o cambiar sólo una de las excepciones. Aunque esto puede parecer engorroso, significa que no es necesario conocer los ajustes de barrado actuales para poder especificar un patrón de barrado nuevo.

Cuando cambia el compás, se fijan los valores predeterminados de `Timing.baseMoment`, `Timing.beatStructure` y `Timing.beamExceptions`. Un ajuste en el tipo de compás da como resultado un reinicio de los ajustes de barrado automático para el contexto `Timing` al comportamiento predeterminado.
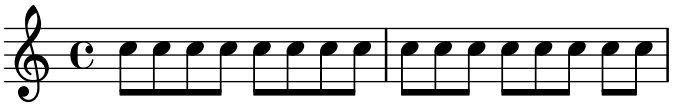
```
\time 6/8
\repeat unfold 6 { a8 }
% group (4 + 2)
\set Timing.beatStructure = #'(4 2)
\repeat unfold 6 { a8 }
% go back to default behavior
\time 6/8
\repeat unfold 6 { a8 }
```



Los ajustes de barrado automático predeterminados para un tipo de compás están determinados en el archivo ‘`scm/time-signature-settings.scm`’. La forma de cambiar los ajustes predeterminados de barrado automático para un tipo de compás se describe en [Indicación de compás], página 62.

Muchos ajustes de barrado automáticos para un tipo de compás contienen una entrada para `beamExceptions`. Por ejemplo, el compás de 4/4 trata de unir el compás en dos partes si solo hay corcheas. La regla `beamExceptions` puede sobreescribir el ajuste `beatStructure` si no se reinicia `beamExceptions`.

```
\time 4/4
\set Timing.baseMoment = #(ly:make-moment 1/8)
\set Timing.beatStructure = #'(3 3 2)
% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c8} |
% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c8}
```



De forma similar, las corcheas en compás de 3/4 se unen mediante una sola barra para todo el compás, de forma predeterminada. Para unir las corcheas en 3/4 mediante una barra en cada parte, reinicie `beamExceptions`.

```
\time 3/4
% by default we beam in (6) due to beamExceptions
\repeat unfold 6 {a8} |
% This will beam (1 1 1) due to default baseMoment and beatStructure
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a8}
```



En la música tipografiada de los períodos clásico y romántico, con frecuencia las barras comienzan a mitad de un compás en 3/4, pero la práctica moderna es evitar la falsa impresión de 6/8 (véase Gould, pág. 153). Se producen situaciones similares en el compás de 3/8. Este comportamiento viene controlado mediante la propiedad de contexto `beamHalfMeasure`, que tiene efecto solamente sobre indicaciones de compás que tienen la cifra 3 en el numerador:

```
\time 3/4
r4. a8 a a |
\set Timing.beamHalfMeasure = ##f
r4. a8 a a |
```



Cómo funciona el barrado automático

Cuando se habilita el barrado automático, la colocación de las barras de corchea automáticas viene determinada por las propiedades de contexto `baseMoment`, `beatStructure` y `beamExceptions`.

Son de aplicación las siguientes reglas, en orden de prioridad, cuando se determina el aspecto de las barras:

- Si está especificada una barra manual con `[...]`, fijar la barra tal y como se ha especificado; en caso contrario,
- si está definida en `beamExceptions` una regla de barrado para este tipo de barra, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,
- si está definida en `beamExceptions` una regla de final de barra para un tipo de barra más largo, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,
- usar los valores de `baseMoment` y de `beatStructure` para determinar los finales de los pulsos dentro del compás, y terminar las barras al final de los pulsos.

En las reglas anteriores, el tipo de barra *beam-type* es la duración de la nota más breve dentro del grupo unido por una barra.

Las reglas de barrado predefinidas están en el archivo '`scm/time-signature-settings.scm`'.

Fragmentos de código seleccionados
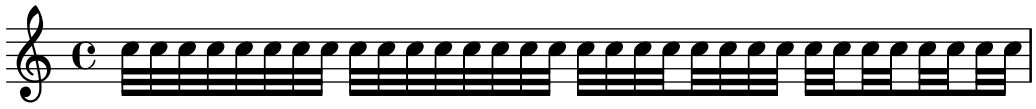
Subdivisión de barras

Las barras de semicorchea, o de figuras más breves, no se subdividen de forma predeterminada. Esto es, las tres (o más) barras se amplían sin romperse sobre grupos completos de notas. Este comportamiento se puede modificar para subdividir las barras en subgrupos mediante el establecimiento de la propiedad `subdivideBeams`. Cuando está activada, las barras se subdividen a intervalos definidos por el valor actual de `baseMoment` mediante la reducción de las barras repetidas a una sola entre los subgrupos. Observe que el valor predeterminado de `baseMoment` es uno más que el denominador del tipo de compás actual, si no se fija explícitamente. Se debe ajustar a una fracción que da la duración del subgrupo de barras utilizando la función `ly:make-moment`, como se ve en este fragmento de código. Asimismo, cuando se modifica `baseMoment`, se debería cambiar también `beatStructure` para que corresponda al `baseMoment` nuevo:

```
\relative c'' {
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c c]

  % Set beam sub-group length to an eighth note
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = #'(2 2 2 2)
  c32[ c c c c c c c]
```
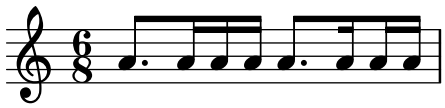
```
  % Set beam sub-group length to a sixteenth note
  \set baseMoment = #(ly:make-moment 1/16)
  \set beatStructure = #'(4 4 4 4)
  c32[ c c c c c c c]
}
```



Barras que se atienen al pulso estrictamente

Se puede hacer que las barras secundarias apunten en la dirección del pulso o fracción a que pertenecen. La primera barra evita los corchetes sueltos (que es el comportamiento predeterminado); la segunda barra sigue el pulso o fracción estrictamente.

```
\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



Símbolos de dirección y símbolos de agrupación de compás

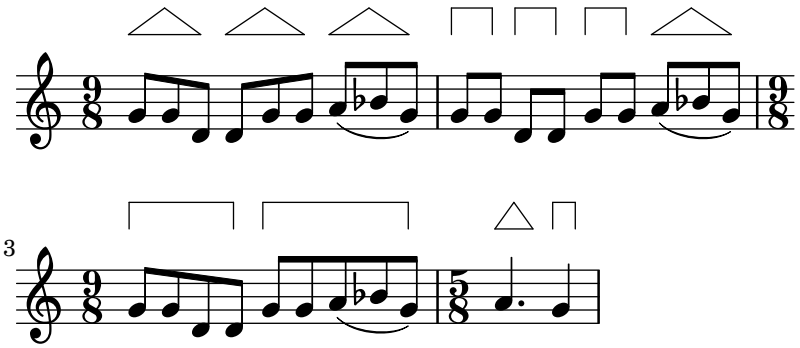
La agrupación de pulsos dentro de un compás está controlada por la propiedad de contexto `beatStructure`. Hay establecidos valores de `beatStructure` para muchos tipos de compases en '`scm/time-signature-settings.scm`'. Los valores de `beatStructure` se pueden cambiar o establecer con `\set`. Como alternativa, se puede usar `\time` para establecer tanto el compás como la estructura de pulsos. Para ello, especificamos la agrupación interna de los pulsos del compás como una lista de números (en la sintaxis de Scheme) antes de la indicación de compás.

`\time` se aplica al contexto `Timing`, por lo que no restablece los valores de `beatStructure` ni de `baseMoment` que se establezcan en otros contextos de nivel inferior, como `Voice`.

Si el grabador `Measure_grouping_engraver` está incluido en uno de los contextos de presentación, se imprimirán signos de agrupación de pulsos. Estos símbolos facilitan la lectura de música moderna rítmicamente compleja. En este ejemplo, el compás de 9/8 se agrupa según dos patrones distintos utilizando los dos métodos, mientras que el compás de 5/8 se agrupa de acuerdo con el ajuste predeterminado que está en '`scm/time-signature-settings.scm`':

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = #'(2 2 2 3)
    g8 g d d g g a( bes g) |
    \time #'(4 5) 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
  }
```

```
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```
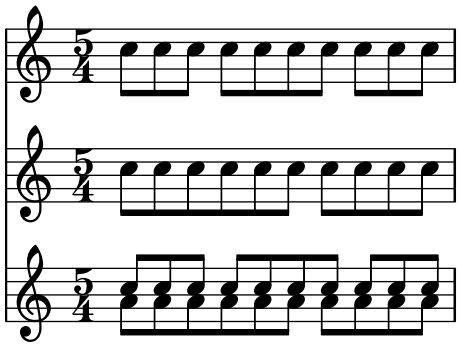


Finales de barra en el contexto Score

Las reglas de final de barra especificadas en el contexto `Score` se aplican a todos los pentagramas, pero se pueden modificar tanto en los niveles de `Staff` como de `Voice`:

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1/8)
  \set Score.beatStructure = #'(3 4 3)
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = #'(6 4)
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = #'(6 4)
          a8 a a a a a a a a
        }
      >>
    }
  >>
```

```
}
```



Véase también

Referencia de la notación: [Indicación de compás], página 62.

Archivos de inicio: ‘`scm/time-signature-settings.scm`’.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

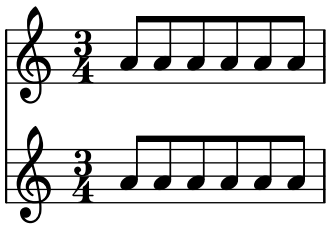
Referencia de funcionamiento interno: Sección “Auto_beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “Beam” in *Referencia de Funcionamiento Interno*, Sección “BeamForbidEvent” in *Referencia de Funcionamiento Interno*, Sección “beam-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Si una partitura termina mientras una barra de corchea automática no ha terminado y aún acepta notas, esta última barra no se imprime en absoluto. Lo mismo sirve para las voces polifónicas introducidas con `<< ... \\ ... >>`. Si una voz polifónica termina mientras una barra de corchea automática aún admite notas, no se imprime. El rodeo para estos problemas es aplicar el barrado manual a la última barra de la voz o partitura.

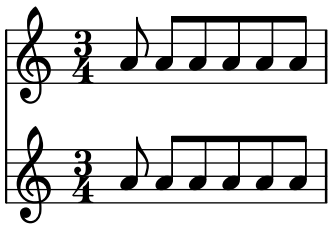
De forma predeterminada, el traductor `Timing` recibe el nombre del contexto `Score` como alias. Esto significa que el establecimiento del compás en ua pauta afectará al barrado de las otras pautas también. Así, un ajuste en el compás en un pentagrama tardío reiniciará el barrado personalizado que se había ajustado en un pentagrama más temprano. Una forma de evitar este problema es ajustar la indicación de compás en un pentagrama solamente.

```
<<
  \new Staff {
    \time 3/4
    \set Timing.baseMoment = #(ly:make-moment 1/8)
    \set Timing.beatStructure = #'(1 5)
    \repeat unfold 6 { a8 }
  }
  \new Staff {
    \repeat unfold 6 { a8 }
  }
>>
```



Los ajustes de barrado predeterminados para dicho compás también se pueden cambiar, de forma que siempre se utilice siempre el barrado deseado. La manera de efectuar cambios en los ajustes de barrado automático para una indicación de compás se describe en [Indicación de compás], página 62.
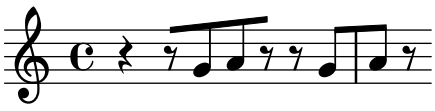
```
<<
  \new Staff {
    \overrideTimeSignatureSettings
      3/4               % timeSignatureFraction
      1/8               % baseMomentFraction
      #'(1 5)           % beatStructure
      #'()              % beamExceptions
    \time 3/4
    \repeat unfold 6 { a8 }
  }
  \new Staff {
    \time 3/4
    \repeat unfold 6 { a8 }
  }
>>
```



Barras manuales

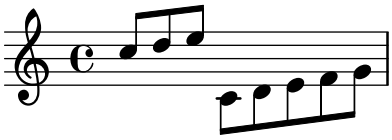
En ciertos casos puede ser preciso sobreescribir el algoritmo de barrado automático. Por ejemplo, el barrador automático no escribe barras por encima de los silencios o las líneas divisorias, y en las partituras corales el barrado se ajusta con frecuencia para que siga la medida de la letra en vez de la de las notas. Tales barras se especifican manualmente marcando los puntos de comienzo y final con `[` y `]`

```
r4 r8[ g' a r] r g[ | a] r
```
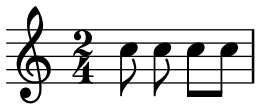


La dirección de las barras se puede establecer manualmente utilizando indicadores de dirección:

```
c8^[ d e] c,_[ d e f g]
```



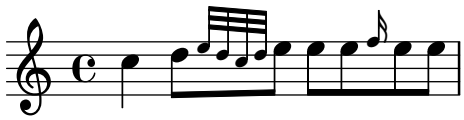
Se pueden marcar notas individuales con `\noBeam` para evitar que resulten unidas por una barra:

```
\time 2/4
c8 c\noBeam c c
```
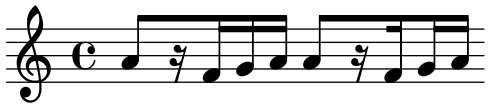


Se pueden producir al mismo tiempo barras de notas de adorno y barras normales. Las notas de adorno sin barra no se colocan dentro de las barras de notas normales.

```
c4 d8[
\grace { e32 d c d }
e8] e[ e
\grace { f16 }
e8 e]
```



Se puede conseguir un control incluso más estricto sobre las barras estableciendo las propiedades `stemLeftBeamCount` y `stemRightBeamCount`. Especifican el número de barras que se dibujarán en los lados izquierdo y derecho, respectivamente, de la nota siguiente. Si cualquiera de estas dos propiedades está ajustada a un valor, dicho valor se usará una sola vez, y luego se borrará. En este ejemplo, el último Fa se imprime con sólo una barra en el lado izquierdo, es decir, la barra de corchea del grupo como un todo.

```
a8[ r16 f g a]
a8[ r16
\set stemLeftBeamCount = #2
\set stemRightBeamCount = #1
f16
\set stemLeftBeamCount = #1
g16 a]
```



Instrucciones predefinidas

`\noBeam`.

Fragmentos de código seleccionados

Corchetes rectos y extremos de barra sueltos

Son posibles tanto los corchetes rectos sobre notas sueltas como extremos de barra sueltos en figuras unidas, con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[]` emparejados.

Para corchetes rectos que apunten a la derecha sobre notas sueltas, use indicadores de barra emparejados `[]` y establezca `stemLeftBeamCount` a cero (véase el ejemplo 1).

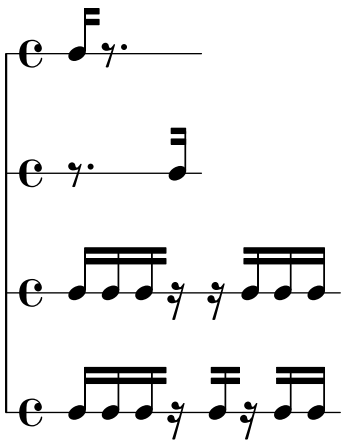
Para corchetes rectos que apunten a la izquierda, establezca en su lugar `stemRightBeamCount` (ejemplo 2).

Para extremos sueltos que apunten a la derecha al final de un conjunto de notas unidas, establezca `stemRightBeamCount` a un valor positivo. Y para extremos sueltos que apunten a la izquierda al principio de un conjunto de notas unidas, establezca `stemLeftBeamCount` en su lugar (ejemplo 3).

A veces, para una nota suelta rodeada de silencios tiene sentido que lleve los dos extremos sueltos del corchete plano, apuntando a derecha e izquierda. Hágalo solamente con indicadores de barra emparejados `[ ]` (ejemplo 4).

(Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes de la cantidad de barras no se recuerdan, y por ello el par de corchetes planos aplicados a la nota Do semicorchea `c'16[]` del último ejemplo no tiene nada que ver con el `\set` de dos notas por detrás.)

```
\score {
  <<
    % Example 1
    \new RhythmicStaff {
      \set stemLeftBeamCount = #0
      c16[]
      r8.
    }
    % Example 2
    \new RhythmicStaff {
      r8.
      \set stemRightBeamCount = #0
      c16[]
    }
    % Example 3
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r r
      \set stemLeftBeamCount = #2
      c16 c c
    }
    % Example 4
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r
      c16[]
      r16
      \set stemLeftBeamCount = #2
      c16 c
    }
  >>
}
```



Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601, [Notas de adorno], página 108.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” in *Referencia de Funcionamiento Interno*, Sección “BeamEvent” in *Referencia de Funcionamiento Interno*, Sección “Beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “beam-interface” in *Referencia de Funcionamiento Interno*, Sección “Stem_engraver” in *Referencia de Funcionamiento Interno*.

Barras progresivas
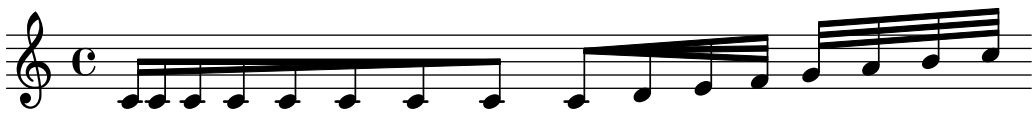
Las barras progresivas se usan para indicar que un pequeño grupo de notas se debe tocar a una velocidad creciente (o decreciente), sin cambiar el tempo general de la pieza. El ámbito de la barra progresiva se debe indicar manualmente usando `[` y `]`, el efecto de ángulo de la barra se inicia especificando una dirección en la propiedad `grow-direction` del objeto `Beam`.

Si queremos que la colocación de las notas y el sonido de la salida MIDI refleje el *ritardando* o *accelerando* indicado por la barra progresiva, las notas se deben agrupar como una expresión musical delimitada por llaves y precedida de una instrucción `featheredDurations` que especifica la razón entre las duraciones de la primera y la última notas dentro del grupo.

Los corchetes rectos muestran el ámbito de la barra y las llaves muestran qué notas han de modificar sus duraciones. Normalmente delimitarían el mismo grupo de notas, pero no es un requisito: las dos instrucciones son independientes.

En el ejemplo siguiente las ocho semicorcheas ocupan exactamente el mismo tiempo que una blanca, pero la primera nota tiene la mitad de duración que la última, con las notas intermedias alargándose gradualmente. Las cuatro primeras fusas se aceleran gradualmente, mientras que las últimas cuatro fusas están a un tempo constante.

```
\override Beam.grow-direction = #LEFT
\featherDurations #(ly:make-moment 2/1)
{ c16[ c c c c c c c] }
\override Beam.grow-direction = #RIGHT
\featherDurations #(ly:make-moment 2/3)
{ c32[ d e f] }
% revert to non-feathered beams
\override Beam.grow-direction = #'()
{ g32[ a b c] }
```



El espaciado en la salida impresa representa las duraciones de las notas de una forma sólo aproximada, pero la salida MIDI es exacta.

Instrucciones predefinidas

`\featherDurations`.

Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Advertencias y problemas conocidos

La instrucción `\featherDurations` sólo funciona con fragmentos musicales muy breves, y cuando los números de la fracción son pequeños.
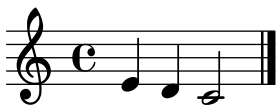
1.2.5 Compases

Barras de compás

Las líneas divisorias delimitan a los compases pero también se pueden usar para indicar las repeticiones. Normalmente, las líneas divisorias normales se insertan de manera automática en la salida impresa en lugares que están basados en el compás actual.

Las barras de compás sencillas insertadas automáticamente se pueden cambiar por otros tipos con la instrucción `\bar`. Por ejemplo, se suele poner una doble barra de cierre al final de la pieza:

```
e4 d c2 \bar "|."
```



No deja de ser válida la última nota de un compás si no termina sobre la línea divisoria automática: se supone que la nota se prolonga sobre el compás siguiente. Pero una secuencia larga de dichos compases prolongados puede hacer que la música aparezca comprimida o incluso que se salga de la página. Esto es a causa de que los saltos de línea automáticos solamente se producen al final de compases completos, es decir, cuando todas las notas han finalizado antes de que el compás termine.

> **Nota:** Una duración incorrecta puede hacer que se trate de evitar la producción de saltos de línea, dando como resultado una línea de música con una compresión exagerada o música que se sale de la página.

También se permiten saltos de línea en las barras insertadas manualmente incluso dentro de compases incompletos. Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice lo siguiente:

```
\bar ""
```

De esta manera se insertará una barra invisible de compás y se hará posible el salto de línea en este punto, sin forzarlo. No se incrementa el contador de los números de compás. Para forzar un salto de línea, consulte Sección 4.3.1 [Saltos de línea], página 527.
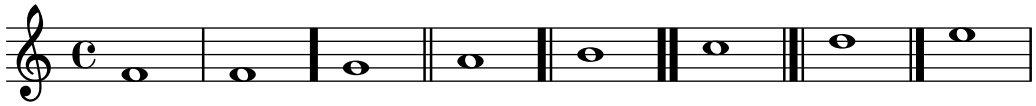
Esta y otras líneas divisorias especiales se pueden insertar manualmente en cualquier punto. Cuando coinciden con el final de un compás, sustituyen a la línea divisoria simple que se habría insertado automáticamente en dicho lugar. Cuando no coinciden con el final de un compás, se inserta la línea especificada en ese punto en la salida impresa.

Observe que las líneas divisorias manuales son puramente visuales. No afectan a ninguna de las propiedades que una barra normal afectaría, como a los números de compás, alteraciones, saltos

de línea, etc. No afectan tampoco al cálculo y colocación de las líneas divisorias subsiguientes. Cuando una divisoria se coloca manualmente donde ya existe una divisoria normal, los efectos de la línea original no se alteran.

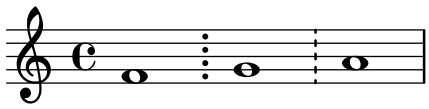
Están disponibles para su inserción manual dos tipos de líneas divisorias simples y cinco tipos de doble barra:

```
f1 \bar "|"
f1 \bar "."
g1 \bar "||"
a1 \bar ".|"
b1 \bar ".."
c1 \bar "|.|"
d1 \bar "|."
e1
```



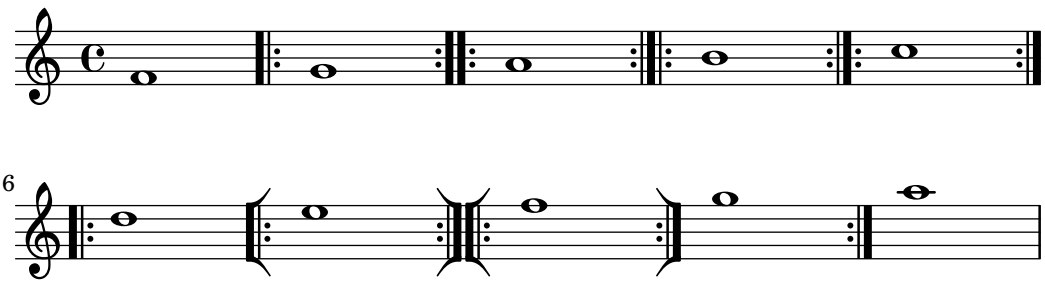
así como la barra de puntos y la discontinua:

```
f1 \bar ";"
g1 \bar "!"
a1
```



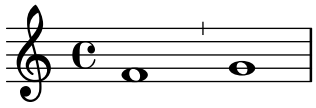
y nueve tipos de barra de repetición:

```
f1 \bar ".|:"
g1 \bar ":..:"
a1 \bar ":|.|:"
b1 \bar ":|.:"
c1 \bar ":.|.:"
d1 \bar "[|:"
e1 \bar ":|][|:"
f1 \bar ":|]"
g1 \bar ":|."
a1
```



Además se puede imprimir una línea divisoria como una marca corta:

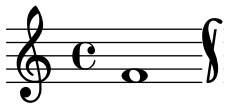
```
f1 \bar "'" g1
```



Sin embargo, dado que las mencionadas marcas se utilizan usualmente en el canto gregoriano, es preferible en este caso utilizar `\divisioMinima`, que se describe en la sección [Divisiones], página 435 dentro de Canto gregoriano.

LilyPond contempla la notación del canto kievano y ofrece una línea divisoria especial kievana:
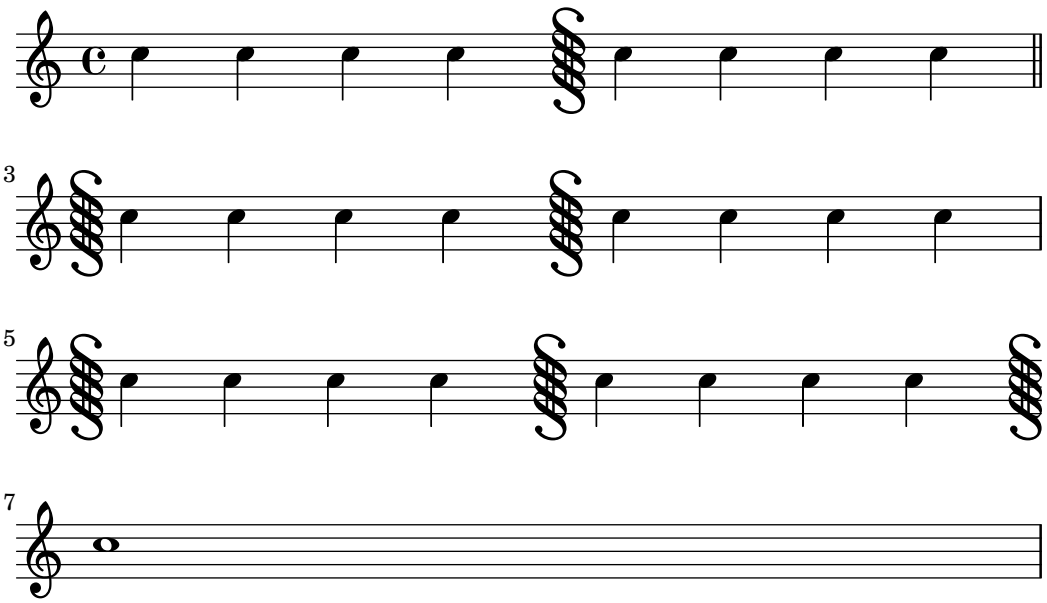
```
f1 \bar "k"
```



Pueden verse más detalles de esta notación explicados en Sección 2.9.5 [Tipografiado del canto kievano en notación cuadrada], página 444.

Para los símbolos de segno en línea, existen tres tipos de barras de compás que se diferencian en su comportamiento en los saltos de línea:

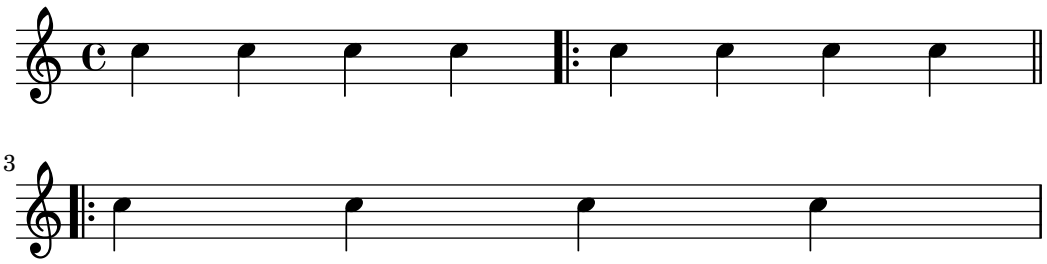
```
c4 c c c
\bar "S"
c4 c c c \break
\bar "S"
c4 c c c
\bar "S-|"
c4 c c c \break
\bar "S-|"
c4 c c c
\bar "S-S"
c4 c c c \break
\bar "S-S"
c1
```



Aunque se pueden insertar manualmente barras de compás con significado de repeticiones, no se reconocen como repeticiones por parte de LilyPond. Las secciones repetidas se introducen mejor utilizando las diversas instrucciones de repetición (véase Sección 1.4 [Repeticiones], página 143), que imprimen automáticamente las barras correspondientes.

Además se puede especificar `".|:-||"`, que equivale a `".|:"` excepto en los saltos de línea, en que produce una doble barra al final de la línea y una repetición izquierda (de comienzo) al principio de la línea siguiente.

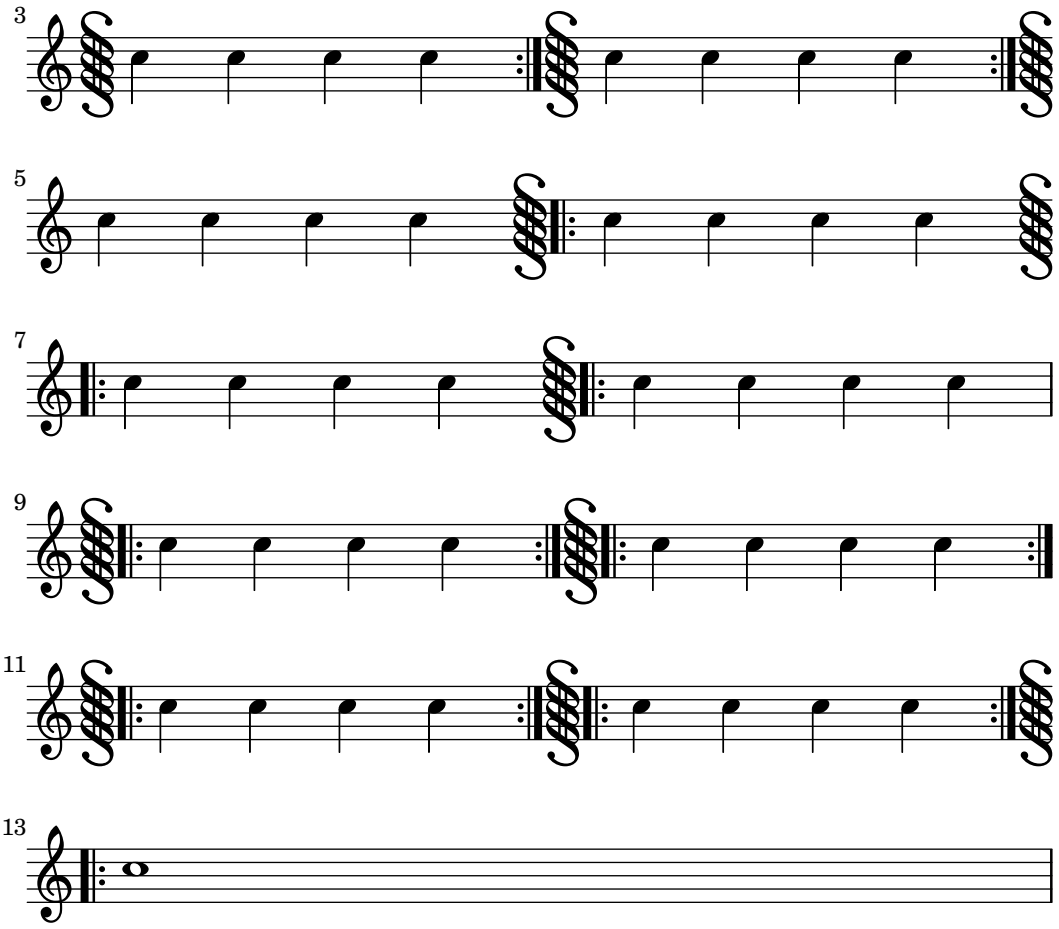
```
c4 c c c
\bar ".|:-||"
c4 c c c \break
\bar ".|:-||"
c4 c c c
```



Para combinaciones de repeticiones con el símbolo de segno, existen seis variantes diferentes:

```
c4 c c c
\bar ":|.S"
c4 c c c \break
\bar ":|.S"
c4 c c c
\bar ":|.S-S"
c4 c c c \break
\bar ":|.S-S"
c4 c c c
\bar "S.|:-S"
c4 c c c \break
\bar "S.|:-S"
c4 c c c
\bar "S.|:"
c4 c c c \break
\bar "S.|:"
c4 c c c
\bar ":|.S.|:"
c4 c c c \break
\bar ":|.S.|:"
c4 c c c
\bar ":|.S.|:-S"
c4 c c c \break
\bar ":|.S.|:-S"
c1
```





Además, existe una instrucción `\inStaffSegno` que crea una barra de compás con símbolo de segno, situada en conjunción con una línea de repetición adecuada si se utiliza con una instrucción `\repeat volta`, véase [Repeticiones normales], página 144.

Se pueden definir tipos nuevos de líneas divisorias con `\defineBarLine`:

`\defineBarLine` *tipo_de_barra* `#'(`*final comienzo extensión*`)`

Las variables de `\defineBarline` pueden incluir la cadena de caracteres 'vacía' `""`, que equivale a imprimir una línea divisoria invisible. También se puede establecer al valor falso `#f` que no imprime línea divisoria alguna.

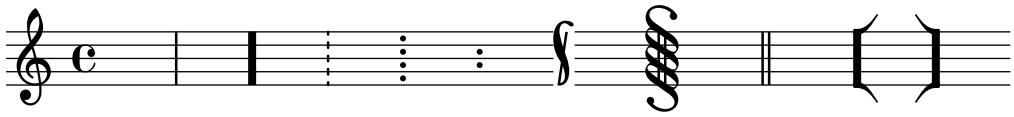
Después de la definición, la nueva línea divisoria se puede utilizar mediante `\bar` *tipo_de_barra*.

Actualmente están disponibles diez elementos de barra de compás:

```
\defineBarLine ":" #'("" ":" "")
\defineBarLine "=" #'("=" "" "")
\defineBarLine "[" #'("" "[" "")
\defineBarLine "]" #'("]" "" "")

\new Staff {
  s1 \bar "|"
  s1 \bar "."
  s1 \bar "!"
  s1 \bar ";"
  s1 \bar ":"
  s1 \bar "k"
  s1 \bar "S"
  s1 \bar "="
  s1 \bar "["
```

```
  s1 \bar "]"
  s1 \bar ""
}
```

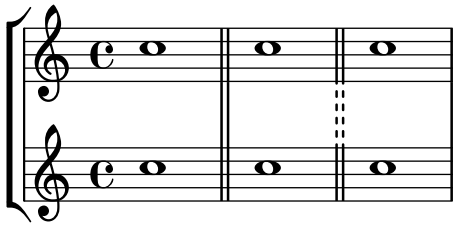


La línea divisoria `"="` proporciona la barra de doble extensión, usada en combinación con el símbolo de segno. No lo use como una doble línea divisoria fina aislada; para ello es preferible `\bar "||"`.

El signo `"-"` inicia anotaciones a las barras de compás, que son útiles para distinguir entre aquellas que tienen idéntica apariencia pero distinto comportamiento en los saltos de línea y/o diferentes barras de extensión. La parque que sigue al signo `"-"` no se usa para construir la barra de compás.

```
\defineBarLine "||-dashedSpan" #'("||" "" "!!")

\new StaffGroup <<
  \new Staff {
    c1 \bar "||"
    c1 \bar "||-dashedSpan"
    c1
  }
  \new Staff {
    c1
    c1
    c1
  }
>>
```

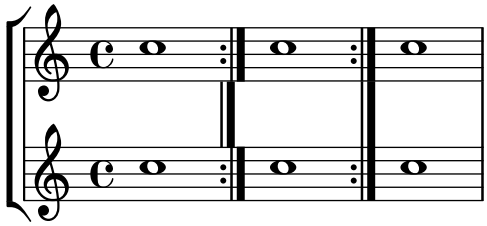


Además, el carácter del espacio en blanco `" "` sirve como contenedor para definir barras de extensión correctamente alineadas a las barras principales:

```
\defineBarLine ":|.-wrong" #'(":|." "" "|.")
\defineBarLine ":|.-right" #'(":|." "" " |.")

\new StaffGroup <<
  \new Staff {
    c1 \bar ":|.-wrong"
    c1 \bar ":|.-right"
    c1
  }
  \new Staff {
    c1
    c1
```
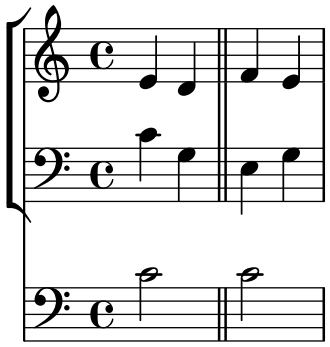
```
    c1
  }
>>
```



Si se necesitan elementos adicionales, LilyPond provee una forma sencilla de definirlos. Para ver más información sobre cómo modificar o añadir barras de compás, consulte el archivo '`scm/bar-line.scm`'.

En las partituras con muchos pentagramas, una instrucción `\bar` en uno de ellos se aplica automáticamente a todos los demás. Las líneas resultantes se conectan entre los distintos pentagramas de un `StaffGroup`, `PianoStaff` o `GrandStaff`.

```
<<
  \new StaffGroup <<
    \new Staff {
      e4 d
      \bar "||"
      f4 e
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



La instrucción '`\bar` *`tipo de barra`*' es una forma corta de hacer '`\set Timing.whichBar =` *`tipo de barra`*'. Cuando `whichBar` se establece con el valor de una cadena, se crea una línea divisoria de ese tipo.

El tipo de barra predeterminado que se usa para las líneas divisorias insertadas automáticamente es `"|"`. Se puede cambiar en cualquier momento con '`\set Timing.defaultBarType =` *`tipo de barra`*'.

Véase también

Referencia de la notación: Sección 4.3.1 [Saltos de línea], página 527, Sección 1.4 [Repeticiones], página 143, [Agrupar pentagramas], página 184.

Archivos instalados: '`scm/bar-line.scm`'.

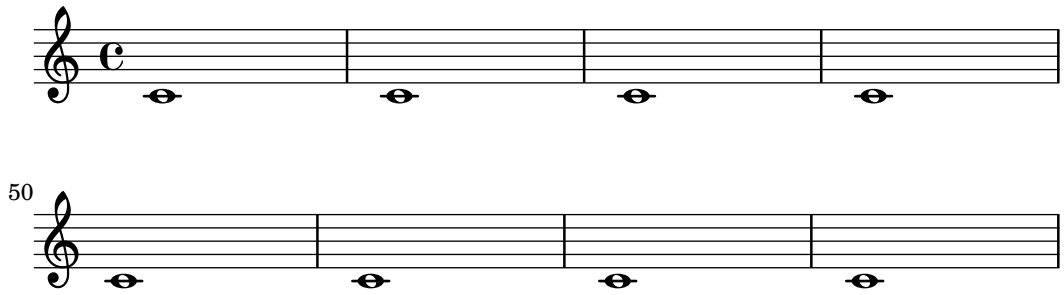
Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarLine” in *Referencia de Funcionamiento Interno* (creada al nivel de `Staff` (pentagrama)), Sección “SpanBar” in *Referencia de Funcionamiento Interno* (a través de los pentagramas), Sección “Timing_translator” in *Referencia de Funcionamiento Interno* (para las propiedades del contador de tiempo Timing).

Numeración de compases

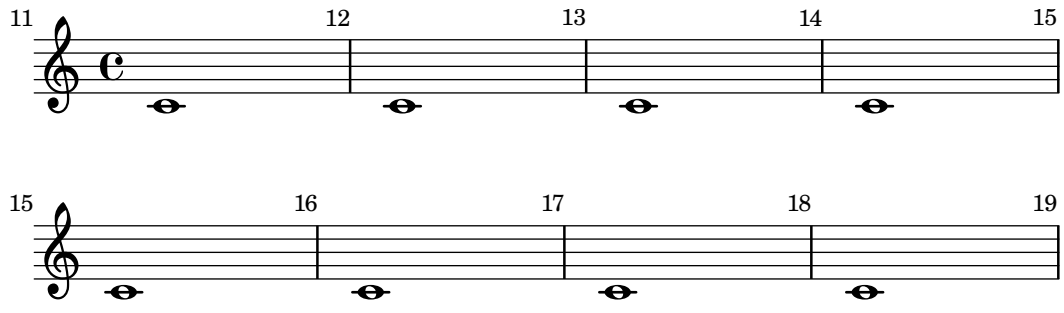
Por defecto, los números de compás se imprimen al principio de la línea, excepto la primera. El número propiamente dicho se almacena en la propiedad `currentBarNumber`, que normalmente se actualiza automáticamente para cada compás. También se puede establecer manualmente:

```
c1 c c c
\break
\set Score.currentBarNumber = #50
c1 c c c
```



Los números de compás se pueden tipografiar a intervalos regulares en vez de al principio de cada línea. Para hacerlo se debe sobreescribir el comportamiento predeterminado de forma que se puedan imprimir en otros lugares. Esto viene controlado por la propiedad `break-visibility` de `BarNumber`. Toma tres valores que se pueden fijar al valor `#t` o `#f` para especificar si el número de compás correspondiente es visible o no. El orden de los tres valores es `visible al final de la línea`, `visible en mitad de la línea`, `visible al principio de la línea`. En el ejemplo siguiente se imprimen los números de compás en todos los sitios posibles:

```
\override Score.BarNumber.break-visibility = #'#(#t #t #t)
\set Score.currentBarNumber = #11
% Permit first bar number to be printed
\bar ""
c1 | c | c | c
\break
c1 | c | c | c
```

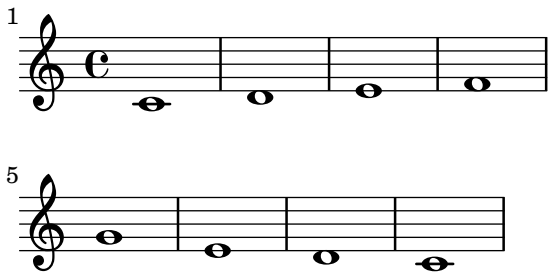


Fragmentos de código seleccionados

Imprimir el número de compás en el primer compás

De forma predeterminada se suprime el número del primer compás de una partitura si es menor o igual a '1'. Al establecer el valor de `barNumberVisibility` a `all-bar-numbers-visible`, se puede imprimir cualquier número de compás para el primer compás y todos los siguientes. Observe que, para que esto funcione, se debe insertar una líinea divisoria vacía antes de la primera nota.

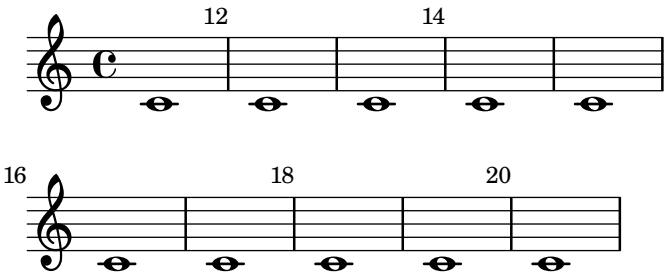
```
\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Imprimir números de compás a intervalos regulares

Se pueden imprimir los números de compás a intervalos regulares mediante el establecimiento de la propiedad `barNumberVisibility`. Aquí los números de compás se imprimen a cada dos compases excepto al final de la línea.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



Imprimir números de compás dentro de rectángulos o circunferencias

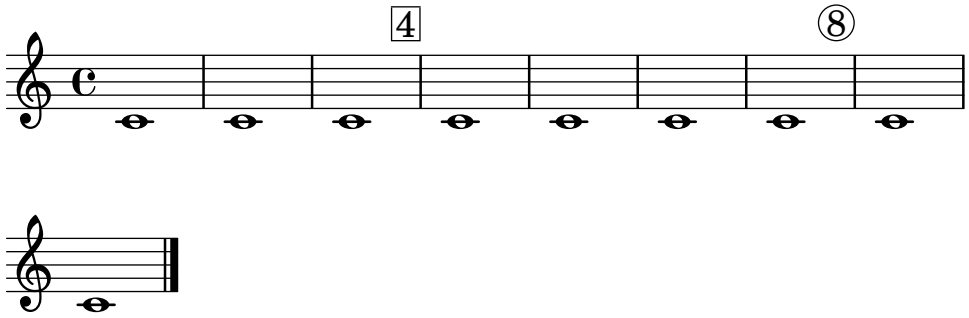
Los números de compás también se pueden imprimir dentro de rectángulos o de circunferencias.

```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }
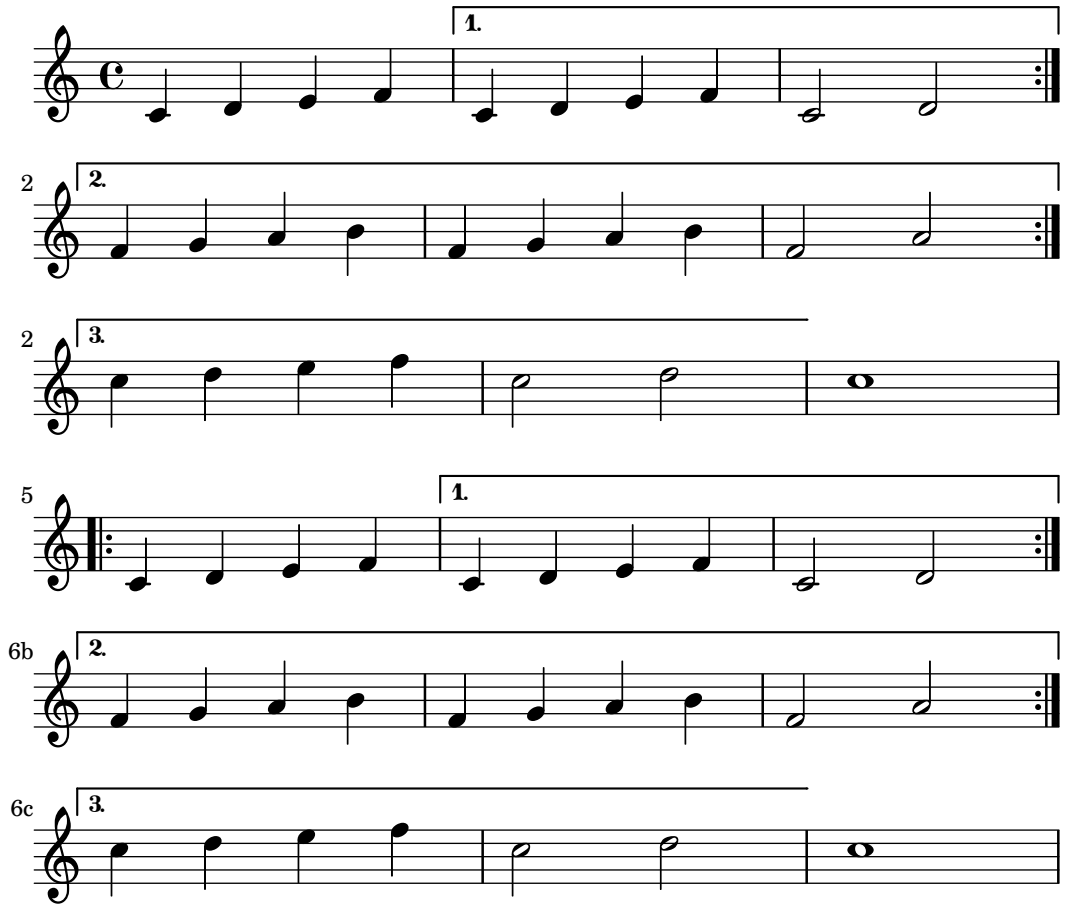
  % Draw a circle round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```



Numeración de compases alternativa

Se pueden seleccionar dos métodos alternativos para la numeración de compases, especiales para cuando hay repeticiones.

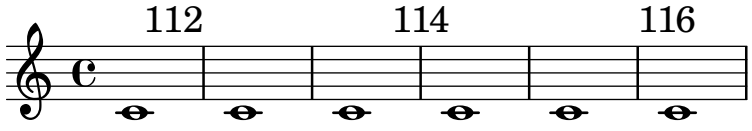
```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1
}
```



Alineación de los números de compás

Los números de compás se alinean de forma predeterminada por la derecha con su objeto padre. Éste es por lo general el borde izquierdo de una línea o, si los números se imprimen dentro de la línea, el lado izquierdo de una línea divisoria. Los números también se pueden situar directamente sobre la barra de compás o alineados por la izquierda con ella.

```
\relative c' {
  \set Score.currentBarNumber = #111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber.self-alignment-X = #CENTER
  c1 | c1
  % Left-align bar numbers
  \override Score.BarNumber.self-alignment-X = #LEFT
  c1 | c1
}
```
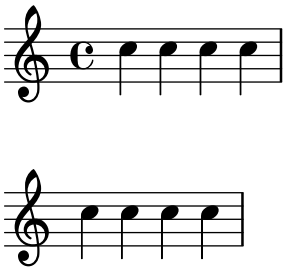


Suprimir los números de compás de toda la partitura

Se pueden eliminar completamente los números de compás quitando el grabador `Bar_number_engraver` del contexto de `Score`.

```
\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```



Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarNumber” in *Referencia de Funcionamiento Interno*, Sección “Bar_number_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los números de compás pueden colisionar con el corchete `StaffGroup`, si hay uno en la parte de arriba. Para solucionarlo, se puede usar la propiedad de relleno `padding` de `BarNumber` para colocar el número correctamente. Consulte Sección “StaffGroup” in *Referencia de Funcionamiento Interno* y Sección “BarNumber” in *Referencia de Funcionamiento Interno* para ver más información.

Comprobación de compás y de número de compás

Las comprobaciones de compás ayudan a detectar errores en las duraciones. Una comprobación de compás se escribe usando el símbolo de la barra vertical, `|`, en cualquier lugar donde se espera que caiga una línea divisoria. Si se encuentran líneas de comprobación de compás en otros lugares, se imprime una lista de advertencias en el archivo log de registro, mostrando los números de línea y columna en que han fallado las comprobaciones de compás. En el siguiente ejemplo, la segunda comprobación de compás avisará de un error

```
\time 3/4 c2 e4 | g2 |
```

Las comprobaciones de compás también se pueden usar en la letra de las canciones:
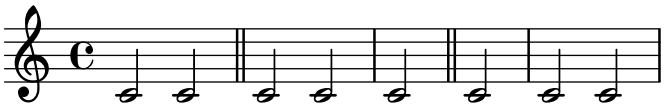
```
\lyricmode {
  \time 2/4
  Bri -- lla, | bri -- lla, |
}
```

Las duraciones incorrectas a menudo desbaratan la partitura completamente, sobre todo si la música es polifónica, por ello el mejor sitio para empezar a corregir la entrada es hacer un barrido en busca de comprobaciones de compás fallidas y duraciones incorrectas.

Si varias comprobaciones de compás corresponden al mismo intervalo musical, sólo se imprime el primer mensaje de advertencia. Esto permite que la advertencia se centre en la fuente del error de cuenta de tiempo.

También es posible redefinir la acción que se lleva a cabo cuando se encuentra una barra vertical o símbolo de comprobación de compás, `|`, en el código de entrada, de forma que haga algo distinto a una comprobación de compás. Se hace asignando una expresión musical a `"|"`, el símbolo de barra vertical. En el ejemplo siguiente se establece `|` de forma que inserte una doble línea divisoria cuando aparece en el código de entrada, en vez de comprobar el final de un compás.

```
"|" = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
}
```



Al copiar piezas musicales grandes puede servir de ayuda comprobar que el número de compás de LilyPond corresponde al original desde el que está copiando. Esto se puede comprobar con `\barNumberCheck`, por ejemplo:

```
\barNumberCheck #123
```

imprime una advertencia si el `currentBarNumber` (número del compás actual) no es el 123 en el momento de la compilación.

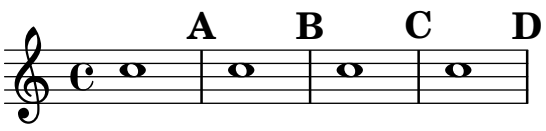
Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Llamadas de ensayo

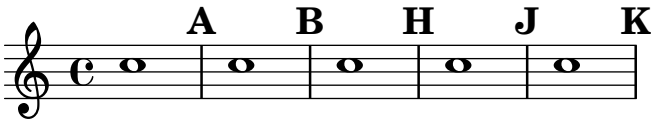
Para imprimir una letra de ensayo, utilice la orden `\mark`.

```
c1 \mark \default
c1 \mark \default
c1 \mark \default
c1 \mark \default
```



La letra de ensayo se incrementa automáticamente si usa `\mark \default`, pero también puede utilizar un número entero como argumento para establecer la indicación manualmente. El valor que se utilizará se almacena dentro de la propiedad `rehearsalMark`.

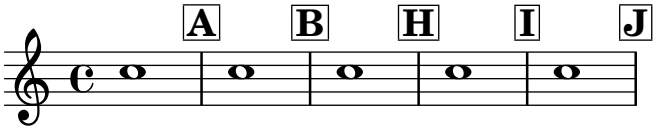
```
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



La letra 'I' se salta de acuerdo con las tradiciones de grabado. Si quiere incluir la letra 'I', use una de las instrucciones siguientes, según el estilo de marcas de ensayo que desee (sólo letras, letras dentro de un rectángulo o letras dentro de un círculo).

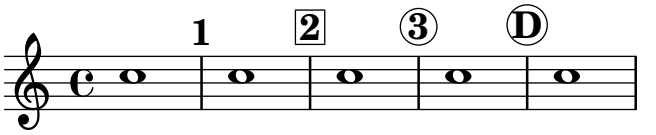
```
\set Score.markFormatter = #format-mark-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
\set Score.markFormatter = #format-mark-circle-alphabet
```

```
\set Score.markFormatter = #format-mark-box-alphabet
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



El estilo se define por medio de la propiedad `markFormatter`. Es una función que toma como argumentos la marca en curso (un número entero) y el contexto actual. Debe devolver un objeto de marcado. En el ejemplo siguiente, se establece `markFormatter` con el valor de un procedimiento enlatado. Después de algunos compases, se establece con el valor de una función que produce un número encerrado en una caja.

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark \default
c1 \mark \default
\set Score.markFormatter = #format-mark-box-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-letters
c1
```



El archivo '`scm/translation-functions.scm`' contiene las definiciones de `format-mark-numbers` (el formato por omisión), `format-mark-box-numbers`, `format-mark-letters` y `format-mark-box-letters`. Se pueden usar éstos como inspiración para otras funciones de formateo.

Podemos utilizar `format-mark-barnumbers`, `format-mark-box-barnumbers` y `format-mark-circle-barnumbers` para obtener números de compás en lugar de números o letras secuenciales.

Se pueden especificar otros estilos de letra de ensayo de forma manual:

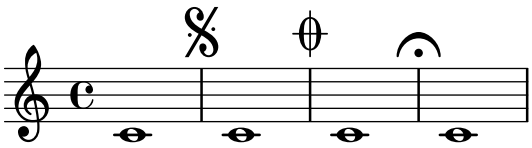
```
\mark "A1"
```

Observe que `Score.markFormatter` no afecta a las marcas que se especifican de esta forma. Sin embargo, es posible aplicar un elemento `\markup` a la cadena.

```
\mark \markup{ \box A1 }
```

Los glifos musicales (como el segno) se pueden imprimir dentro de un elemento `\mark`

```
c1 \mark \markup { \musicglyph #"scripts.segno" }
c1 \mark \markup { \musicglyph #"scripts.coda" }
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1
```



Consulte Sección A.8 [La tipografía Feta], página 650 para ver una lista de los símbolos que se pueden imprimir con `\musicglyph`.

Para ver formas comunes de trucar la colocación de las letras de ensayo, consulte Sección 1.8.2 [Formatear el texto], página 233. Para un control más preciso, consulte `break-alignable-interface` en la sección Sección 5.5.1 [Alineación de objetos], página 616.

El archivo '`scm/translation-functions.scm`' contiene las definiciones de `format-mark-numbers` y de `format-mark-letters`. Se pueden utilizar como inspiración para escribir otras funciones de formateo.

Véase también

Referencia de la notación: Sección A.8 [La tipografía Feta], página 650, Sección 1.8.2 [Formatear el texto], página 233, Sección 5.5.1 [Alineación de objetos], página 616.

Archivos de inicio: '`scm/translation-functions.scm`'.

Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.
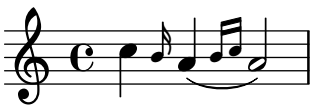
Referencia de funcionamiento interno: Sección "MarkEvent" in *Referencia de Funcionamiento Interno*, Sección "Mark_engraver" in *Referencia de Funcionamiento Interno*, Sección "RehearsalMark" in *Referencia de Funcionamiento Interno*.

1.2.6 Asuntos rítmicos especiales

Notas de adorno
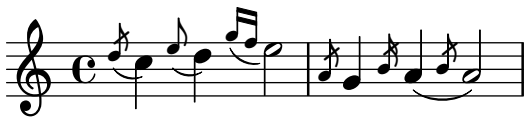
Los mordentes y notas de adorno son ornamentos musicales, impresos en un tipo más pequeño y no ocupan ningún tiempo lógico adicional en el compás.

```
c4 \grace b16 a4(
\grace { b16 c16 } a2)
```
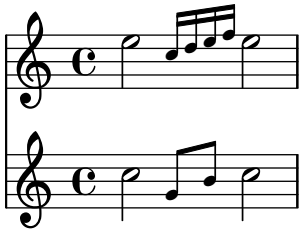


Hay otros tres tipos posibles de notas de adorno; la *acciaccatura* o mordente de una nota (una nota de adorno sin medida que se indica mediante una nota ligada y con el corchete tachado) y la *appoggiatura*, que toma una fracción fija de la nota principal a la que se adjunta, y que se imprime sin tachar. Es posible escribir una nota de adorno con la plica tachada, como la *acciaccatura* pero sin la ligadura, como para colocarla entre notas que están ligadas entre sí, utilizando la función `\slashedGrace`.

```
\acciaccatura d8 c4
\appoggiatura e8 d4
\acciaccatura { g16 f } e2
\slashedGrace a,8 g4
\slashedGrace b16 a4(
\slashedGrace b8 a2)
```



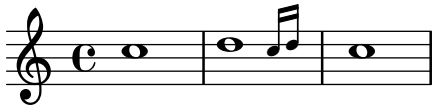
La colocación de notas de adorno se sincroniza entre los distintos pentagramas. En el siguiente ejemplo, hay dos semicorcheas de adorno por cada corchea de adorno:

```
<<
  \new Staff { e2 \grace { c16 d e f } e2 }
  \new Staff { c2 \grace { g8 b } c2 }
>>
```



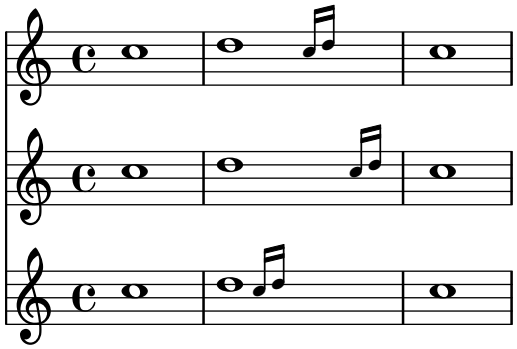
Si queremos terminar una nota con un adorno, usamos la instrucción `\afterGrace`. Toma dos argumentos: la nota principal, y las notas de adorno que siguen a la nota principal.

```
c1 \afterGrace d1 { c16[ d] } c1
```
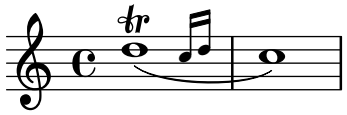


Esto pone las notas de adorno después de un espacio que dura 3/4 de la longitud de la nota principal. La fracción predeterminada de 3/4 se puede cambiar estableciendo `afterGraceFraction`. El ejemplo siguiente muestra el resultado de establecer el espacio en su valor predeterminado, en 15/16, y por último en 1/2 de la nota principal.

```
<<
  \new Staff {
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 15 16))
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 1 2))
    c1 \afterGrace d1 { c16[ d] } c1
  }
>>
```
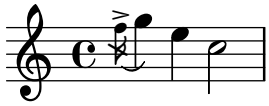


El espacio entre la nota principal y la de adorno también se puede especificar usando espaciadores. El ejemplo siguiente sitúa la nota de adorno después de un espacio que dura 7/8 de la nota principal.

```
\new Voice {
  <<
    { d1^\trill_( }
    { s2 s4. \grace { c16 d } }
  >>
  c1)
}
```



Una expresión musical `\grace` introduce ajustes de tipografía especiales, por ejemplo para producir un tipo más pequeño y para fijar las direcciones. Por ello, cuando se introducen trucos para la presentación, deben ir dentro de la expresión de adorno. Las sobreescrituras se deben también revertir dentro de la expresión de adorno. Aquí, la dirección predeterminada de la plica de la nota de adorno se sobreescribe y luego se revierte.

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4 e c2
}
```
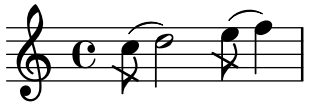


Fragmentos de código seleccionados

Utilizar la barra que tacha las notas de adorno con notas normales

Es posible aplicar la barrita que cruza la barra de las acciaccaturas, en otras situaciones.
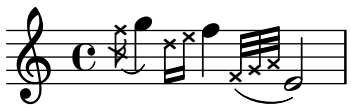
```
\relative c'' {
  \override Flag.stroke-style = #"grace"
  c8( d2) e8( f4)
}
```



Trucar la disposición de las notas de adorno dentro de la música

La disposición de las expresiones de adorno se puede cambiar a lo largo de toda la música usando las funciones `add-grace-property` y `remove-grace-property`. El ejemplo siguiente borra la definición de la dirección de la plica para esta nota de adorno, de manera que las plicas no siemmpre apuntan hacia arriba, y cambia la forma predeterminada de las cabezas a aspas.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```
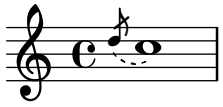


Redefinición de los valores globales predeterminados para notas de adorno

Los valores predeterminados para las notas de adorno están almacenados en los identificadores `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` y `stopAppoggiaturaMusic`, que están definidos en el archivo '`ly/grace-init.ly`'. Redefiniéndolos se pueden obtener otros efectos.

```
startAcciaccaturaMusic = {
  <>(
  \override Flag.stroke-style = #"grace"
  \slurDashed
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>)
}

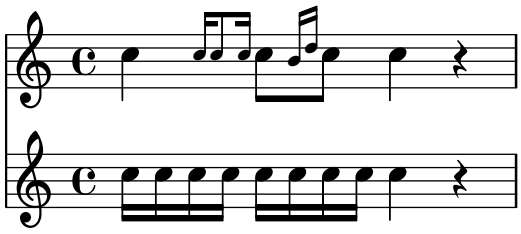
\relative c'' {
  \acciaccatura d8 c1
}
```



Posicionamiento de las notas de adorno con espacio flotante

Al establecer la propiedad `'strict-grace-spacing` hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

```
\relative c'' {
  <<
    \override Score.SpacingSpanner.strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16] }
      c8[ \grace { b16 d } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```



Véase también

Glosario musical: Sección “grace notes” in *Glosario Musical*, Sección “acciaccatura” in *Glosario Musical*, Sección “appoggiatura” in *Glosario Musical*.

Referencia de la notación: [Escalar las duraciones], página 49, [Barras manuales], página 90.

Archivos de inicio: ‘`ly/grace-init.ly`’.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.
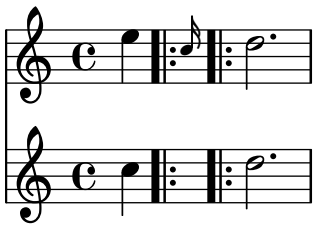
Referencia de funcionamiento interno: Sección “GraceMusic” in *Referencia de Funcionamiento Interno*, Sección “Grace_beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “Grace_auto_beam_engraver” in *Referencia de Funcionamiento Interno*, Sección “Grace_engraver” in *Referencia de Funcionamiento Interno*, Sección “Grace_spacing_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Una *acciaccatura* de varias notas con una barra se imprime sin tachar, y tiene exactamente la misma apariencia que una *appoggiatura* de varias notas con barra.
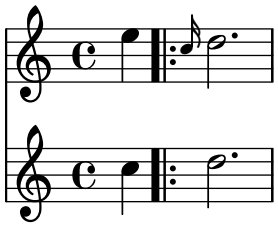
La sincronización de las notas de adorno también puede acarrear sorpresas. La notación de pentagramas, como armaduras, líneas divisorias, etc., se sincronizan también. Ponga cuidado cuando mezcle pentagramas con adornos y sin adornos, por ejemplo

```
<<
  \new Staff { e4 \bar ".|:" \grace c16 d2. }
  \new Staff { c4 \bar ".|:" d2. }
>>
```



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

```
<<
  \new Staff { e4 \bar ".|:" \grace c16 d2. }
  \new Staff { c4 \bar ".|:" \grace s16 d2. }
>>
```



El uso de notas de adorno dentro de los contextos de voz tiende a confundir la manera en que se tipografía la voz. Esto se puede soslayar mediante la inserción de un silencio o nota entre la instrucción de voz y la nota de adorno.

```
accMusic = {
  \acciaccatura { f8 } e8 r8 \acciaccatura { f8 } e8 r4
}

\new Staff {
  <<
    \new Voice {
      \relative c'' {
        r8 r8 \voiceOne \accMusic \oneVoice r8 |
        r8 \voiceOne r8 \accMusic \oneVoice r8 |
      }
    }
    \new Voice {
      \relative c' {
        s8 s8 \voiceTwo \accMusic \oneVoice s8 |
        s8 \voiceTwo r8 \accMusic \oneVoice s8 |
      }
    }
  >>
}
```



Las secciones de adorno sólo se deben usar dentro de expresiones de música secuenciales. No están contemplados el anidado ni la yuxtaposición de secciones de adorno, y podría producir caídas u otros errores.

Cada nota de adorno en la salida MIDI tiene una longitud que es 1/4 de su duración real. Si la duración combinada de las notas de adorno es mayor que la longitud de la nota precedente, se genera un error "`Retrocediendo en el tiempo MIDI`". Tiene dos opciones: en primer lugar, puede hacer más corta la duración de las notas de adorno, por ejemplo:

```
c'8 \acciaccatura { c'8[ d' e' f' g'] }
```

se convierte en:

```
c'8 \acciaccatura { c'16[ d' e' f' g'] }
```

Otra opción es cambiar explícitamente la duración musical:

```
c'8 \acciaccatura { \scaleDurations 1/2 { c'8[ d' e' f' g'] } }
```

Véase [Escalar las duraciones], página 49.

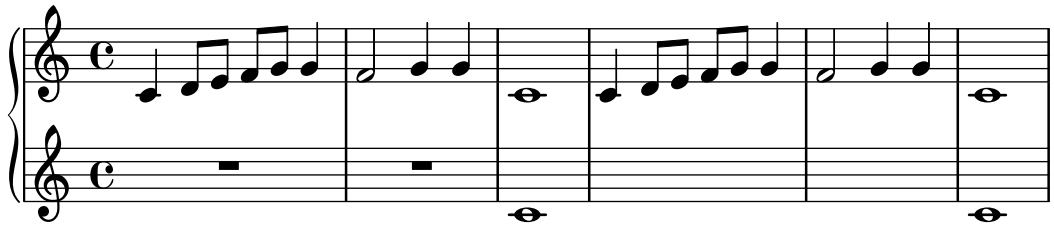
Alinear con una cadenza

En un contexto orquestal, las cadenzas presentan un problema especial: al construir una partitura que tiene una cadenza, todos los demás instrumentos deben saltar tantas notas como la longitud de la cadenza, pues en caso contrario empezarán demasiado pronto o demasiado tarde.

Una solución a este problema son las funciones `mmrest-of-length` y `skip-of-length`. Estas funciones de Scheme toman un fragmento de música como argumento y generan un `\skip` o silencio multicompás, de la longitud exacta del fragmento.

```
MyCadenza = \relative c' {
  c4 d8 e f g g4
  f2 g4 g
}

\new GrandStaff <<
  \new Staff {
    \MyCadenza c'1
    \MyCadenza c'1
  }
  \new Staff {
    #(mmrest-of-length MyCadenza)
    c'1
    #(skip-of-length MyCadenza)
    c'1
  }
>>
```



Véase también

Glosario musical: Sección “cadenza” in *Glosario Musical*.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Gestión del tiempo

El tiempo está administrado por el `Timing_translator`, que de forma predeterminada vive en el contexto de `Score`. Se añade un alias, `Timing`, al contexto en que se coloca el `Timing_translator`. Para asegurar que está disponible el alias `Timing`, quizá tenga que crear explícitamente una instancia del contexto contenedor (como `Voice` o `Staff`).

Se usan las siguientes propiedades de `Timing` para seguir la pista del tiempo dentro de la partitura.

`currentBarNumber`
: El número de compás en curso. Para ver un ejemplo que muestra el uso de esta propiedad, consulte [Numeración de compases], página 101.
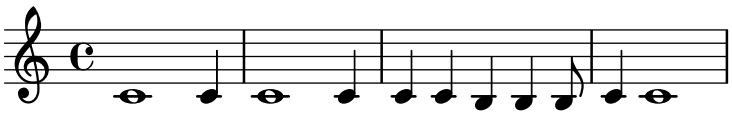
`measureLength`
: La longitud de los compases dentro de la indicación actual de compás. Para un 4/4 esto es 1, y para el 6/8 es 3/4. Su valor determinad cuándo se insertan las líneas divisorias y cómo se generan las barras automáticas.

`measurePosition`
: El punto en que nos encontramos dentro del compás. Esta cantidad se reinicia sustrayendo `measureLength` cada vez que se alcanza o se excede `measureLength`. Cuando eso ocurre, se incrementa `currentBarNumber`.

`timing`
: Si tiene un valor verdadero, las variables anteriores se actualizan a cada paso de tiempo. Cuando tiene un valor falso, el grabador se queda en el compás actual indefinidamente.

La cuenta del tiempo se puede cambiar estableciendo el valor de cualquiera de estas variables explícitamente. En el siguiente ejemplo, se imprime la indicación de compás predeterminada 4/4, pero `measureLength` tiene está ajustado a 5/4. En los 4/8 hasta el tercer compás, la posición `measurePosition` se adelanta en 1/8 hasta 5/8, acortando ese compás en 1/8. Entonces, la siguiente línea divisoria cae en 9/8 en vez de hacerlo en 5/4.

```
\new Voice \relative c' {
  \set Timing.measureLength = #(ly:make-moment 5/4)
  c1 c4 |
  c1 c4 |
  c4 c
  \set Timing.measurePosition = #(ly:make-moment 5/8)
  b4 b b8 |
  c4 c1 |
}
```



Como lo ilustra el ejemplo, `ly:make-moment n m` construye una duración de la fracción n/m de una redonda. Por ejemplo, `ly:make-moment 1 8` es una duración de una corchea y `ly:make-moment 7 16` es la duración de siete semicorcheas.
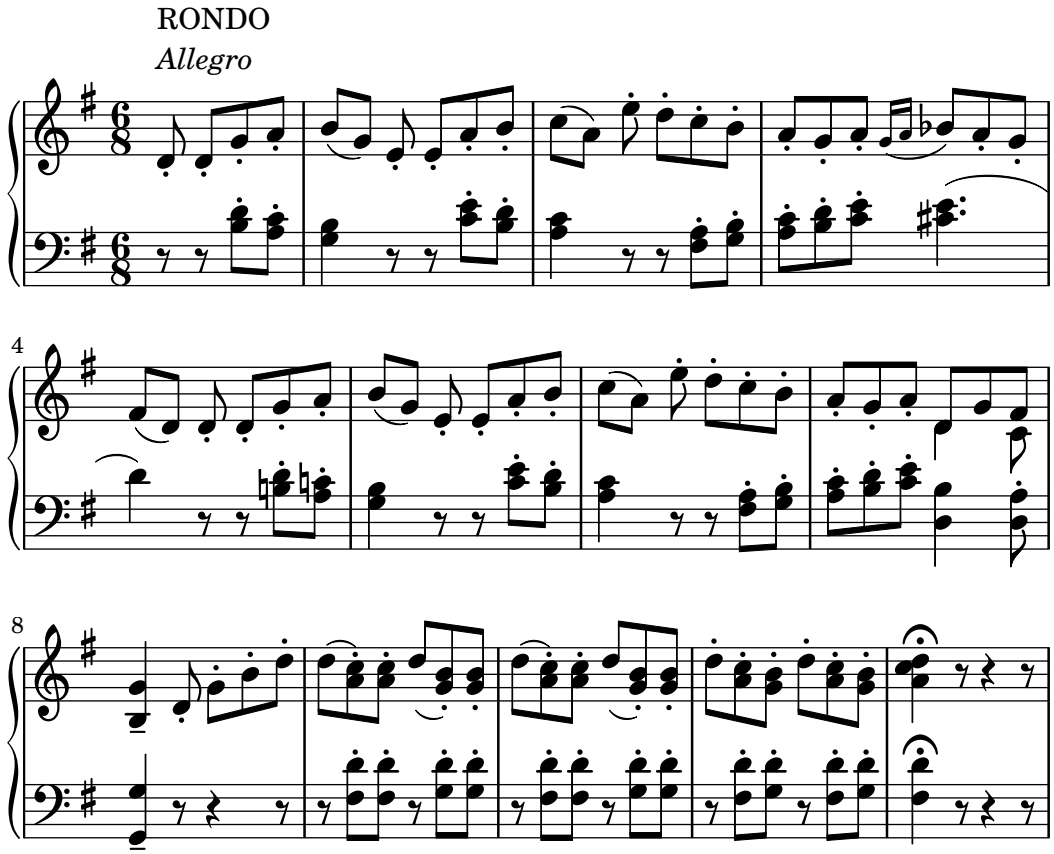
Véase también

Referencia de la notación: [Numeración de compases], página 101, [Música sin compasear], página 71.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing_translator” in *Referencia de Funcionamiento Interno*, Sección “Score” in *Referencia de Funcionamiento Interno*.

1.3 Expresiones



Esta sección relaciona diversas marcas de expresión que se pueden crear en un a partitura.

1.3.1 Expresiones adosadas a las notas

Esta sección explica cómo crear marcas expresivas que están aplicadas a notas: articulaciones, ornamentos y matices. También se tratan los métodos para crear las nuevas marcas dinámicas.
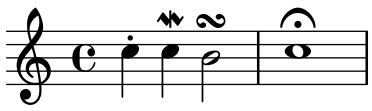
Articulaciones y ornamentos

Se pueden adjuntar a las notas un amplio abanico de símbolos para denotar articulaciones, adornos y otras indicaciones de ejecución, utilizando la sintaxis siguiente:

```
nota\nombre
```
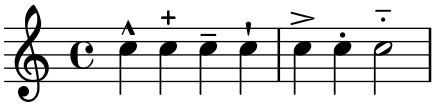
La lista de los posibles valores de `nombre` está en Sección A.13 [Lista de articulaciones], página 725. Por ejemplo:

```
c4\staccato c\mordent b2\turn
c1\fermata
```



Algunas de estas articulaciones tienen abreviaturas que facilitan su escritura. Las abreviaturas se escriben detrás del nombre de la nota, y su sintaxis consiste en un guión `-` seguido de un símbolo que especifica la articulación. Existen abreviaturas predefinidas para el *marcato*, *stopped* (nota apagada), *tenuto*, *staccatissimo*, *accent* (acento), *staccato* (picado), y *portato*. La salida correspondiente a estas articulaciones aparece de la siguiente manera:

```
c4-^ c-+ c-- c-!
c4-> c-. c2-_
```
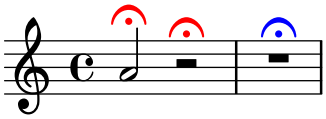


Las reglas para la colocación predeterminada de las articulaciones se encuentran definidas en el archivo '`scm/script.scm`'. Las articulaciones y ornamentos se pueden colocar manualmente por encima o por debajo del pentagrama; véase Sección 5.4.2 [Dirección y posición], página 601.

Las articulaciones son objetos `Script`. Sus propiedades se describen de forma más completa en Sección "Script" in *Referencia de Funcionamiento Interno*.

Las articulaciones se pueden adjuntar a los silencios así como a las notas, pero no se pueden adjuntar a silencions multicompás. Está disponible una instrucción predefinida especial, `\fermataMarkup`, para adosar un calderón a un silencio multicompás (y sólo a un silencio multicompás). La instrucción crea un objeto `MultiMeasureRestText`.

```
\override Script.color = #red
\override MultiMeasureRestText.color = #blue
a2\fermata r\fermata
R1\fermataMarkup
```



Además de articulaciones, se pueden adjuntar textos y elementos de marcado a las notas. Véase [Guiones de texto], página 226.

Para ver más información acerca de la ordenación de los elementos Script y TextScript que se adjuntan a las notas, consulte Sección "Colocación de los objetos" in *Manual de Aprendizaje*.

Fragmentos de código seleccionados

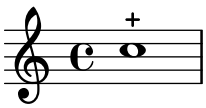
Modificar los valores predeterminados para la notación abreviada de las articulaciones
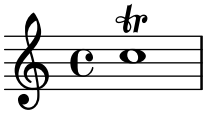
Las abreviaturas se encuentran definidas dentro del archivo '`ly/script-init.ly`', donde las variables `dashHat`, `dashPlus`, `dashDash`, `dashBar`, `dashLarger`, `dashDot` y `dashUnderscore` reciben valores predeterminados. Se pueden modificar estos valores predeterminados para las abreviaturas. Por ejemplo, para asociar la abreviatura `-+` (`dashPlus`) con el símbolo del semitrino en lugar del símbolo predeterminado `+`, asigne el valor `trill` a la variable `dashPlus`:

```
\relative c'' { c1-+ }

dashPlus = "trill"

\relative c'' { c1-+ }
```
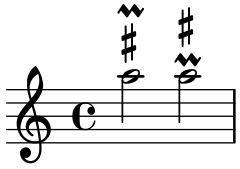




Controlar la ordenación vertical de las inscripciones

El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `'script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el sostenido) tiene primero la prioridad más baja, por lo que se sitúa en la posición más baja en el primer ejemplo. En el segundo, el semitrino (el `Script`) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

```
\relative c''' {
  \once \override TextScript.script-priority = #-100
  a2^\prall^\markup { \sharp }
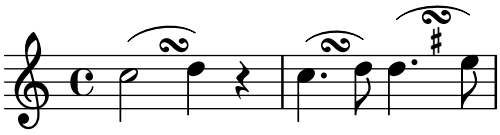
  \once \override Script.script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur property`. Cambiando las fracciones `2/3` y `1/3` puede ajustarse la posición horizontal.

```
\relative c'' {
  c2*2/3 ( s2*1/3\turn d4) r
  <<
    { c4.( d8) }
    { s4 s\turn }
  >>
  \transpose c d \relative c'' <<
    { c4.( d8) }
    {
      s4
      \once \set suggestAccidentals = ##t
      \once \override AccidentalSuggestion #'outside-staff-priority = ##f
      \once \override AccidentalSuggestion #'avoid-slur = #'inside
      \once \override AccidentalSuggestion #'font-size = #-3
      \once \override AccidentalSuggestion #'script-priority = #-1
      \single \hideNotes
      b8-\turn \noBeam
      s8
    }
  >>
}
```



Véase también

Glosario musical: Sección “tenuto” in *Glosario Musical*, Sección “accent” in *Glosario Musical*, Sección “staccato” in *Glosario Musical*, Sección “portato” in *Glosario Musical*.

Manual de aprendizaje: Sección “Colocación de los objetos” in *Manual de Aprendizaje*.

Referencia de la notación: [Guiones de texto], página 226, Sección 5.4.2 [Dirección y posición], página 601, Sección A.13 [Lista de articulaciones], página 725, [Trinos], página 141.

Archivos instalados: ‘`scm/script.scm`’.

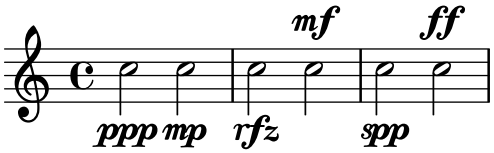
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Script” in *Referencia de Funcionamiento Interno*, Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Matices dinámicos

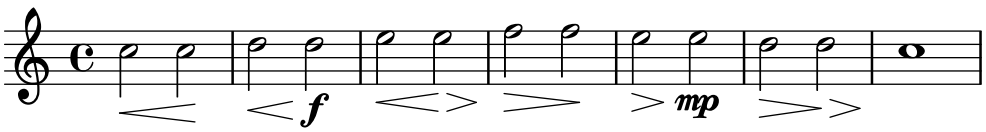
Las marcas dinámicas de matiz absoluto se especifican usando una instrucción después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz` y `\rfz`. Las indicaciones dinámicas se pueden colocar manualmente por encima o por debajo del pentagrama; para ver más detalles, consulte Sección 5.4.2 [Dirección y posición], página 601.

```
c2\ppp c\mp
c2\rfz c^\mf
c2_\spp c^\ff
```
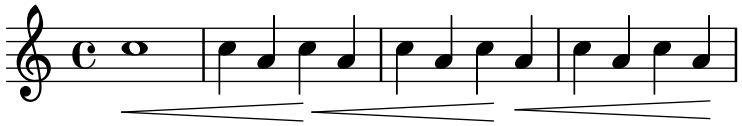


Una indicación de *crescendo* se comienza con `\<` y se termina con `\!`, con un matiz absoluto o con otra indicación de crescendo o de decrescendo. Una indicación de *decrescendo* comienza con `\>` y se termina también con `\!`, con un matiz dinámico absoluto o con otra indicación de crescendo o de decrescendo. Se pueden usar `\cr` y `\decr` en lugar de `\<` y `\>`. De forma predeterminada, se tipografían reguladores en ángulo cuando se utiliza esta notación.

```
c2\< c\!
d2\< d\f
e2\< e\>
f2\> f\!
e2\> e\mp
d2\> d\>
c1\!
```
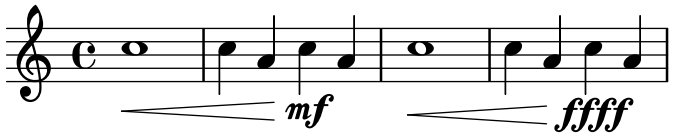


Un regulador terminado mediante `\!` finaliza en el borde derecho de la nota que lleva el `\!` adosado. En el caso en que esté terminado con el comienzo de otra indicación *crescendo* o *decrescendo*, terminará en el centro de la nota que tiene adosada la siguiente indicación `\<` ó `\>`. El regulador siguiente empezará entonces en el borde derecho de la misma nota en lugar del borde izquierdo, como sería usual si hubiese terminado con `\!` previamente.

```
c1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
```



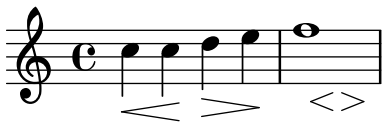
Los reguladores que terminan con indicaciones dinámicas absolutas en lugar de con `\!` también se tipografían de manera parecida. Sin embargo, la longitud de la propia indicación dinámica puede alterar el punto en que finaliza el regulador anterior.

```
c1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
```
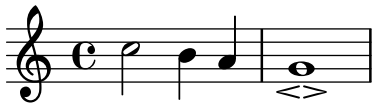


Se requieren silencios espaciadores para tipografiar varias indicaciones dinámicas sobre una sola nota. Esto es útil especialmente para añadir un *crescendo* y un *decrescendo* a la misma nota:

```
c4\< c\! d\> e\!
<< f1 { s4 s4\< s4\> s4\! } >>
```
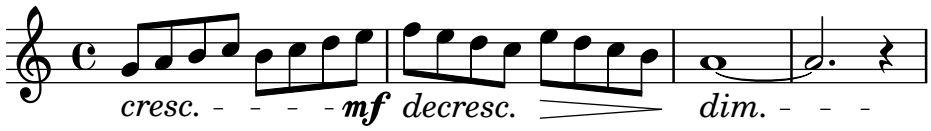


Se puede usar la instrucción `\espressivo` para indicar un crescendo y un decrescendo sobre la misma nota. Sin embargo, advierta que esta indicación está implementada como una articulación, no como una expresión dinámica.

```
c2 b4 a
g1\espressivo
```
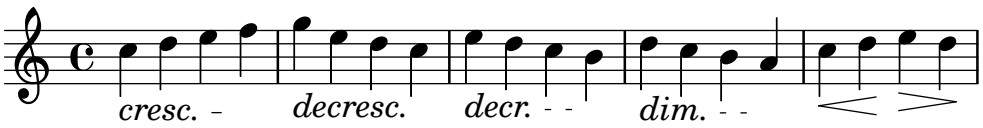


Las indicaciones de crescendo textuales se inician con `\cresc`. Los decrescendos textuales se inician con `\decresc` o con `\dim`. Se trazan líneas extensoras cuando es necesario.

```
g8\cresc a b c b c d e\mf |
f8\decresc e d c e\> d c b |
a1\dim ~ |
a2. r4\! |
```



También pueden sustituirse los reguladores gráficos por cambios de dinámica textuales:

```
\crescTextCresc
c4\< d e f\! |
\dimTextDecresc
g4\> e d c\! |
\dimTextDecr
e4\> d c b\! |
\dimTextDim
d4\> c b a\! |
\crescHairpin
\dimHairpin
c4\< d\! e\> d\! |
```
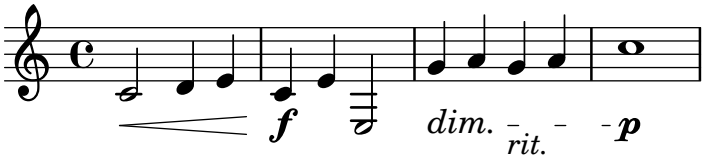


Para crear nuevas indicaciones de matiz absoluto o de texto que deba alinearse con los matices; véase [Indicaciones dinámicas contemporáneas], página 125.

La colocación vertical de las indicaciones de dinámica se maneja por parte de Sección “DynamicLineSpanner” in *Referencia de Funcionamiento Interno*.

Está disponible un contexto `Dynamics` para tipografiar las indicaciones de matiz dinámico en su propia línea horizontal. Utilice silencios de separación para indicar los tiempos (las notas que están dentro de un contexto `Dynamics` también ocupan tiempo musical, pero no se imprimen). El contexto `Dynamics` puede contener otros elementos como inscripciones textuales, objetos extensores de texto e indicaciones de pedal de piano.

```
<<
  \new Staff \relative c' {
    c2 d4 e |
    c4 e e,2 |
    g'4 a g a |
    c1 |
  }
  \new Dynamics {
    s1\< |
    s1\f |
    s2\dim s2-"rit." |
    s1\p |
  }
>>
```



Instrucciones predefinidas

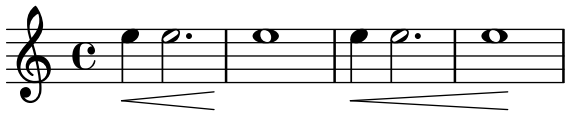
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`, `\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

Fragmentos de código seleccionados

Establecer el comportamiento de los reguladores en las barras de compás

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobreescribiendo la propiedad `'to-barline`.
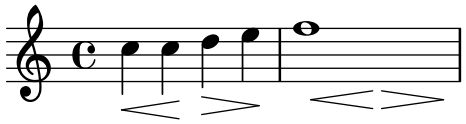
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustar la longitud mínima de los reguladores

Si los reguladores son demasiado cortos, se pueden alargar modificando la propiedad `minimum-length` del objeto `Hairpin`.
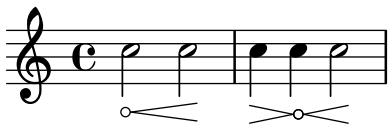
```
\relative c'' {
  c4\< c\! d\> e\!
  \override Hairpin.minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



Impresión de reguladores utilizando la notación «al niente»

Se pueden imprimir reguladores con un círculo en la punta (notación «al niente») estableciendo la propiedad `circled-tip` del objeto `Hairpin` al valor `#t`.

```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```
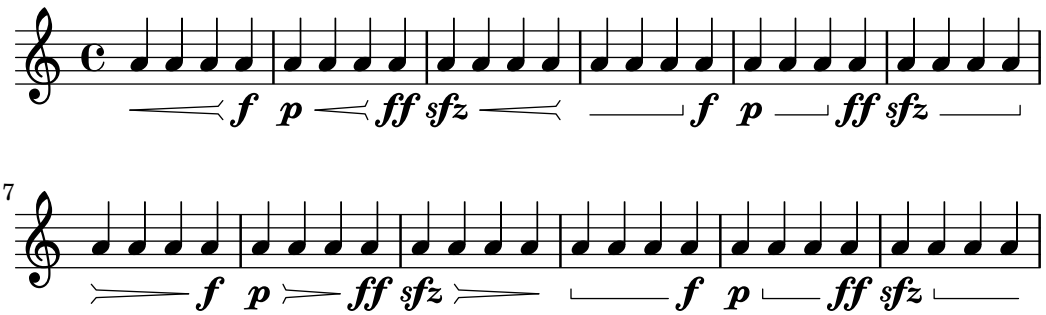


Imprimir reguladores en diversos estilos

Los reguladores se pueden crear en una amplia variedad de estilos.

```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
```

```
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
}
```



Indicaciones dinámicas y textuales alineadas verticalmente

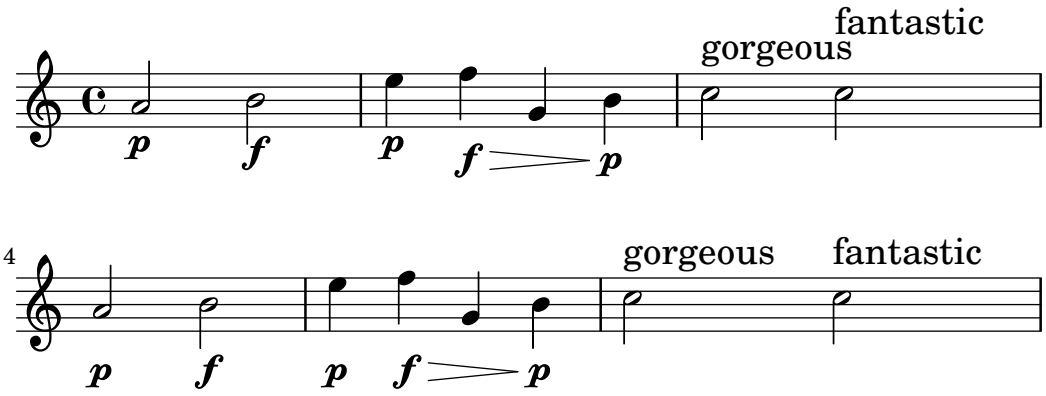
Todos los objetos `DynamicLineSpanner` (reguladores e indicaciones dinámicas de texto) se sitúan con su línea de referencia a una distancia de al menos `'staff-padding` del pentagrama, a no ser que otros elementos de notación los fuerce a colocarse a más distancia. Si se ajusta `'staff-padding` a un valor suficientemente grande, las indicaciones dinámicas quedarán alineadas.

Se usa una idea similar junto a \textLengthOn para alinear las inscripciones de texto a lo largo de su línea de base.

```
\markup \vspace #1 %avoid LSR-bug

music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

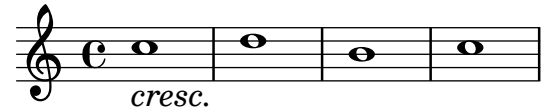
{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = #3
  \textLengthOn
  \override TextScript.staff-padding = #1
  \music
}
```



Ocultar la línea de extensión de las expresiones textuales de dinámica

Los cambios de dinámica con estilo de texto (como cresc. y dim.) se imprimen con una línea intermitente que muestra su alcance. Esta línea se puede suprimir de la siguiente manera:

```
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```
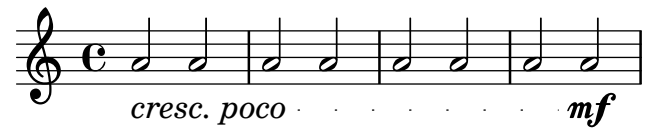


Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales

Se puede modificar el texto empleado para los crescendos y decrescendos modificando las propiedades de contexto `crescendoText` y `decrescendoText`.

El estilo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `'dashed-line` (línea discontinua), y entre otros valores posibles se encuentran `'line` (línea), `'dotted-line` (línea de puntos) y `'none` (nada):

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Véase también

Glosario musical: Sección “al niente” in *Glosario Musical*, Sección “crescendo” in *Glosario Musical*, Sección “decrescendo” in *Glosario Musical*, Sección “hairpin” in *Glosario Musical*.

Manual de aprendizaje: Sección “Articulaciones y matices dinámicos” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601, [Indicaciones dinámicas contemporáneas], página 125, Sección 3.5.3 [¿Qué sale por el MIDI?], página 505, Sección 3.5.5 [Control de los matices en el MIDI], página 506.

Fragmentos de código: Sección "Expressive marks" in *Fragmentos de código*.
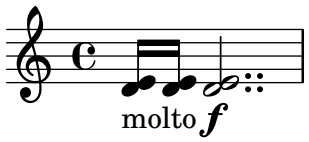
Referencia de funcionamiento interno: Sección "DynamicText" in *Referencia de Funcionamiento Interno*, Sección "Hairpin" in *Referencia de Funcionamiento Interno*, Sección "DynamicLineSpanner" in *Referencia de Funcionamiento Interno*, Sección "Dynamics" in *Referencia de Funcionamiento Interno*.

Indicaciones dinámicas contemporáneas

La manera más fácil de crear indicaciones dinámicas es usar objetos de marcado (`\markup`).

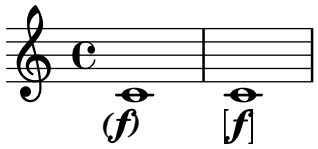
```
moltoF = \markup { molto \dynamic f }

\relative c' {
  <d e>16_\moltoF <d e>
  <d e>2..
}
```
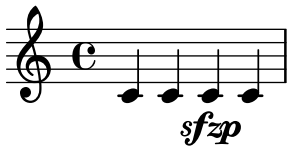


En el modo de marcado se pueden crear indicaciones dinámicas editoriales (entre paréntesis o corchetes). La sintaxis del modo de marcado se describe en Sección 1.8.2 [Formatear el texto], página 233.

```
roundF = \markup {
    \center-align \concat { \bold { \italic ( }
           \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative c' {
  c1_\roundF
  c1_\boxF
}
```



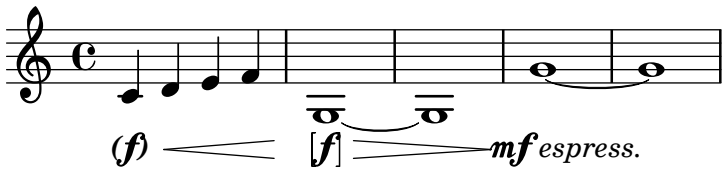
Las indicaciones dinámicas sencillas y centradas se crean fácilmente con la función `make-dynamic-script`.

```
sfzp = #(make-dynamic-script "sfzp")
\relative c' {
  c4 c c\sfzp c
}
```
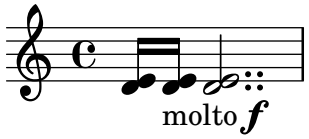


En general `make-dynamic-script` toma cualquier objeto de marcado como argumento. La fuente tipográfica de matices sólo contiene los caracteres `f`, `m`, `p`, `r`, `s` y `z`, por lo que si se desea obtener una indicación dinámica que incluya texto normal o signos de puntuación, es necesario utilizar instrucciones de marcado que devuelvan los ajustes de la familia de fuente tipográfica y su codificación a las del texto normal, por ejemplo `\normal-text`. El interés de la utilización de `make-dynamic-script` en lugar de un elemento de marcado corriente está en asegurar la alineación vertical de los objetos de marcado y reguladores que se aplican a la misma cabeza de nota.

```
roundF = \markup { \center-align \concat {
           \normal-text { \bold { \italic ( } }
           \dynamic f
           \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
              \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative c' {
  c4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g1
  g'1~\mfEspressDynamic
  g1
}
```



Se puede utilizar en su lugar la forma Scheme del modo de marcado. Su sintaxis se explica en Sección “Construcción del marcado en Scheme” in *Extender*.
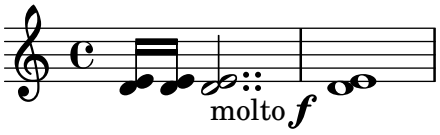
```
moltoF = #(make-dynamic-script
            (markup #:normal-text "molto"
                    #:dynamic "f"))
\relative c' {
  <d e>16 <d e>
  <d e>2..\moltoF
}
```



Para alinear el texto del matiz dinámico a la izquierda en lugar de centrarlo sobre una nota, utilice un `\tweak`:

```
moltoF = \tweak DynamicText.self-alignment-X #LEFT
         #(make-dynamic-script
            (markup #:normal-text "molto"
```

```
                    #:dynamic "f"))
\relative c' {
  <d e>16 <d e>
  <d e>2..\moltoF <d e>1
}
```



Los ajustes para las fuentes tipográficas en el modo de marcado se describen en [Seleccionar la tipografía y su tamaño], página 234.

Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 233, [Seleccionar la tipografía y su tamaño], página 234, Sección 3.5.3 [¿Qué sale por el MIDI?], página 505, Sección 3.5.5 [Control de los matices en el MIDI], página 506.

Extender LilyPond: Sección “Construcción del marcado en Scheme” in *Extender*.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

1.3.2 Expresiones como curvas

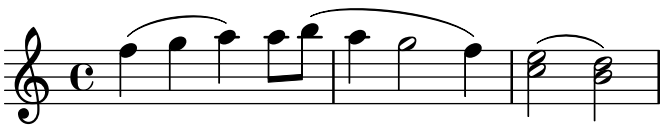
Esta sección explica cómo crear varias marcas expresivas de forma curva: ligaduras de expresión y de fraseo, respiraciones, caídas y elevaciones de tono.

Ligaduras de expresión

Las *Ligaduras de expresión* se introducen utilizando paréntesis:

> **Nota:** En música polifónica, las ligaduras de expresión deben terminar en la misma voz en la que empiezan.

```
f4( g a) a8 b(
a4 g2 f4)
<c e>2( <b d>2)
```
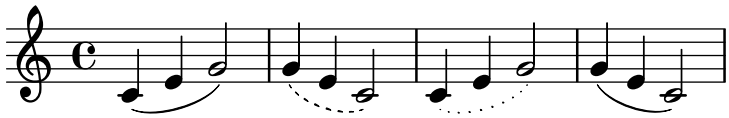


Las ligaduras de expresión se pueden colocar manualmente por encima o por debajo de las notas; véase Sección 5.4.2 [Dirección y posición], página 601.

No están permitidas las ligaduras de expresión simultáneas o solapadas, pero una ligadura de fraseo se puede solapar con una ligadura de expresión. Esto hace posible imprimir dos ligaduras distintas al mismo tiempo. Para ver más detalles, consulte [Ligaduras de fraseo], página 130.

Las ligaduras de expresión pueden ser continuas, discontinuas o de puntos. El estilo predeterminado de las ligaduras de expresión es el continuo:
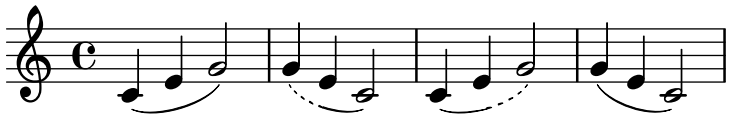
```
c4( e g2)
\slurDashed
g4( e c2)
\slurDotted
```

```
c4( e g2)
\slurSolid
g4( e c2)
```
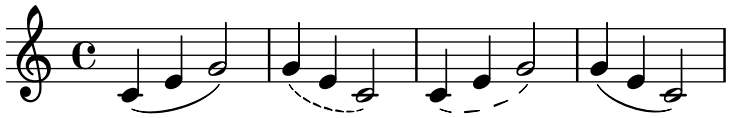


También se puede hacer que las ligaduras de expresión sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda discontinua):

```
c4( e g2)
\slurHalfDashed
g4( e c2)
\slurHalfSolid
c4( e g2)
\slurSolid
g4( e c2)
```



Se pueden definir patrones de discontinuidad para las ligaduras de expresión:

```
c4( e g2)
\slurDashPattern #0.7 #0.75
g4( e c2)
\slurDashPattern #0.5 #2.0
c4( e g2)
\slurSolid
g4( e c2)
```



Instrucciones predefinidas

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurHalfDashed`, `\slurHalfSolid`, `\slurDashPattern`, `\slurSolid`.

Fragmentos de código seleccionados

Utilizar ligaduras dobles para acordes legato

Algunos compositores escriben dos ligaduras cuando quieren acordes legato. Esto se puede conseguir estableciendo `doubleSlurs`.

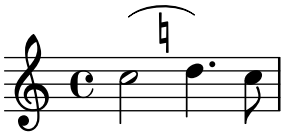
```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Situar los elementos de marcado de texto por dentro de las ligaduras

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida al valor falso para que se impriman por dentro de las ligaduras de expresión.

```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```
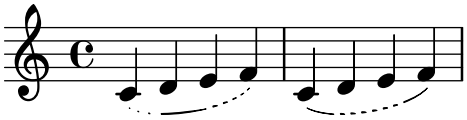


Hacer ligaduras de expresión con estructura compleja de discontinuidad

Las ligaduras de expresión se pueden construir con patrones de discontinuidad complejos mediante la definición de la propiedad `dash-definition`. `dash-definition` es una lista de `elementos de discontinuidad`. Un `elemento de discontinuidad` es una lista de parámetros que definen el comportamiento de discontinuidad de un segmento de la ligadura de expresión.

La ligadura se define en términos del parámetro de bezier t cuyo rango va de 0 a la izquierda de la ligadura hasta 1 en el extremo derecho de la ligadura. Cada `elemento de discontinuidad` es una lista `(t-inicio t-final fracción-discontinuidad período-discontinuidad)`. La región de la ligadura desde `t-inicio` hasta `t-final` tendrá una fracción `fracción-discontinuidad` de cada `período-discontinuidad` de color negro. `período-discontinuidad` se define en términos de espacios de pentagrama. `fracción-discontinuidad` se establece al valor de 1 para una ligadura continua.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'((0 0.3 0.1 0.75)
                              (0.3 0.6 1 1)
                              (0.65 1.0 0.4 0.75))
  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1)
                              (0.3 0.7 0.4 0.75)
                              (0.75 1.0 1 1))
  c4( d e f)
}
```



Véase también

Glosario musical: Sección “slur” in *Glosario Musical*.

Manual de aprendizaje: Sección “Acerca de la no anidabilidad de llaves y ligaduras” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601, [Ligaduras de fraseo], página 130.
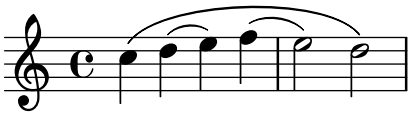
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Slur” in *Referencia de Funcionamiento Interno*.

Ligaduras de fraseo

Las *ligaduras de fraseo* (o marcas de fraseo) que indican una frase musical se escriben usando las instrucciones `\(` y `\)` respectivamente:

```
c4\( d( e) f(
e2) d\)
```
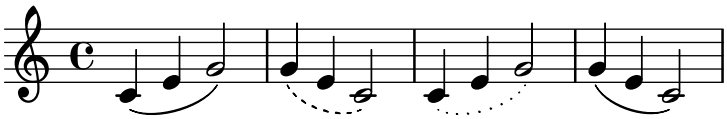


Tipográficamente, la ligadura de fraseo se comporta casi exactamente igual que una ligadura de expresión normal. Sin embargo, se tratan como objetos diferentes. Una `\slurUp` no tendrá ningún efecto sobre una ligadura de fraseo. El fraseo se puede situar manualmente por encima o por debajo de las notas; véase Sección 5.4.2 [Dirección y posición], página 601.

No se pueden tener varias ligaduras de fraseo simultáneas o solapadas.
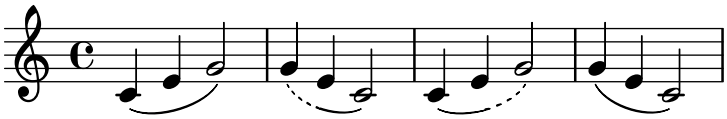
Las ligaduras de fraseo pueden ser continuas, de puntos o de rayas. El estilo predeterminado para las ligaduras de fraseo es el continuo:

```
c4\( e g2\)
\phrasingSlurDashed
g4\( e c2\)
\phrasingSlurDotted
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```
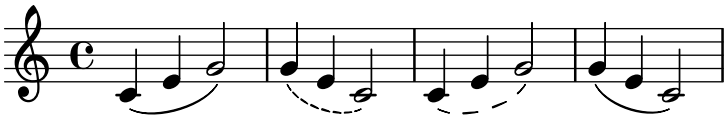


Se puede hacer también que las ligaduras de fraseo sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda mitad discontinua):

```
c4\( e g2\)
\phrasingSlurHalfDashed
g4\( e c2\)
\phrasingSlurHalfSolid
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



Se pueden definir patrones de discontinuidad para las ligaduras de fraseo:

```
c4\( e g2\)
\phrasingSlurDashPattern #0.7 #0.75
g4\( e c2\)
\phrasingSlurDashPattern #0.5 #2.0
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



Las definiciones de patrones de discontinuidad tienen la misma estructura que las definiciones de patrones de discontinuidad de las ligaduras de expresión. Para ver más información acerca de patrones de discontinuidad complejos, consulte los fragmentos de código bajo [Ligaduras de expresión], página 127.

Instrucciones predefinidas

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`, `\phrasingSlurDashed`, `\phrasingSlurDotted`, `\phrasingSlurHalfDashed`, `\phrasingSlurHalfSolid`, `\phrasingSlurDashPattern`, `\phrasingSlurSolid`.

Véase también

Manual de aprendizaje: Sección “Acerca de la no anidabilidad de llaves y ligaduras” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601, [Ligaduras de expresión], página 127.
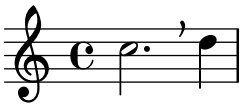
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PhrasingSlur” in *Referencia de Funcionamiento Interno*.

Marcas de respiración
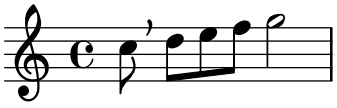
Las respiraciones se introducen utilizando `\breathe`:

```
c2. \breathe d4
```



Las marcas de respiración dan por terminadas las barras automáticas; para sobreescribir este comportamiento, consulte [Barras manuales], página 90.

```
c8 \breathe d e f g2
```



Están contemplados los indicadores musicales para las marcas de respiración en la notación antigua, o divisiones. Para ver más detalles, consulte [Divisiones], página 435.

Fragmentos de código seleccionados

Cambiar el símbolo de la marca de respiración

El glifo de la marca de respiración se puede ajustar sobreescribiendo la propiedad de texto del objeto de presentación `BreathingSign`, con cualquier otro texto de marcado.
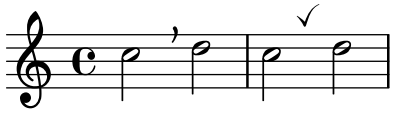
```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph #"scripts.rvarcomma" }
  \breathe
  d2
}
```



Usar una raya corta como símbolo de respiración

La música vocal y de viento usa con frecuencia una raya corta como signo de respiración. Esto indica una repiración que quita un poco de tiempo a la nota anterior en lugar de producir una corta pausa, lo que se indica con la marca de respiración en forma de coma. La marca se puede mover ligeramente hacia arriba para alejarla del pentagrama.

```
\relative c'' {
  c2
  \breathe
  d2
  \override BreathingSign.Y-offset = #2.6
  \override BreathingSign.text =
    \markup { \musicglyph #"scripts.tickmark" }
  c2
  \breathe
  d2
}
```
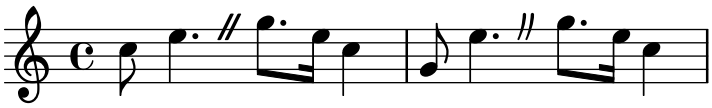


Insertar una cesura

Las marcas de cesura se pueden crear sobreescribiendo la propiedad `'text` del objeto `BreathingSign`. También está disponible una marca de cesura curva.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph #"scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph #"scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
```

```
}
```



Véase también

Glosario musical: Sección “caesura” in *Glosario Musical*.

Referencia de la notación: [Divisiones], página 435.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BreathingEvent” in *Referencia de Funcionamiento Interno*, Sección “BreathingSign” in *Referencia de Funcionamiento Interno*, Sección “Breathing_sign_engraver” in *Referencia de Funcionamiento Interno*.

Caídas y elevaciones

Se pueden expresar *caídas* y *subidas* de tono (falls y doits) añadidas a las notas mediante la instrucción `\bendAfter`. La dirección de la caída o elevación se indica con un signo más o menos (arriba o abajo). El número indica el intervalo de alturas sobre el que se extiende la caída o elevación *partiendo de* la nota principal.

```
c2\bendAfter #+4
c2\bendAfter #-4
c2\bendAfter #+6.5
c2\bendAfter #-6.5
c2\bendAfter #+8
c2\bendAfter #-8
```



Fragmentos de código seleccionados

Ajustar la forma de las subidas y caídas de tono

Puede ser necesario trucar la propiedad `shortest-duration-space` para poder ajustar el tamaño de las caídas y subidas de tono («falls» y «doits»).

```
\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = #4.0
  c2-\bendAfter #5
  c2-\bendAfter #-4.75
  c2-\bendAfter #8.5
  c2-\bendAfter #-6
}
```



Véase también

Glosario musical: Sección “fall” in *Glosario Musical*, Sección “doit” in *Glosario Musical*.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

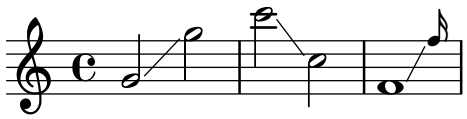
1.3.3 Expresiones como líneas

Esta sección explica cómo crear diversas marcas de expresión que siguen un camino lineal: glissandi, arpegios y trinos.

Glissando

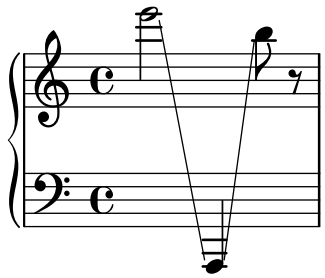
Se crea un *glissando* añadiendo `\glissando` después de la nota:

```
g2\glissando g'
c2\glissando c,
\afterGrace f,1\glissando f'16
```



Un glissando puede conectar notas de un pentagrama a otro:

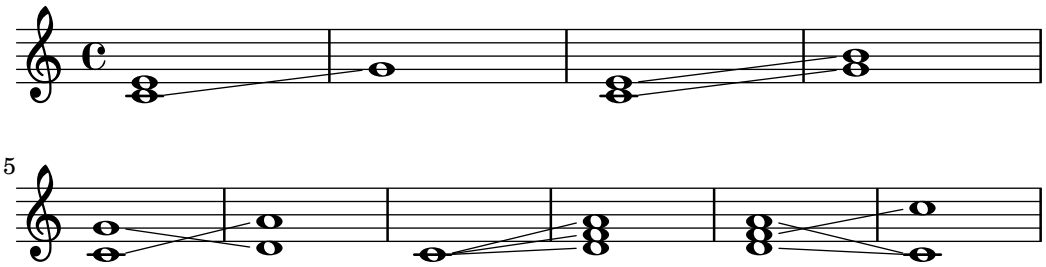
```
\new PianoStaff <<
  \new Staff = "right" {
    e'''2\glissando
    \change Staff = "left"
    a,,4\glissando
    \change Staff = "right"
    b''8 r |
  }
  \new Staff = "left" {
    \clef bass
    s1
  }
>>
```



Un glissando puede conectar las notas dentro de un acorde. Si se necesita hacer algo que no sea una conexión de una nota con otra entre los dos acordes, las conexiones entre las notas vienen definidas por medio de `\glissandoMap`, donde las notas de un acorde se suponen numeradas, empezando en cero, en el orden en que aparecen en el código de entrada del archivo '`.ly`'.

```
<c, e>1\glissando g' |
<c, e>1\glissando |
<g' b> |
\break
\set glissandoMap = #'((0 . 1) (1 . 0))
<c, g'>1\glissando |
<d a'> |
\set glissandoMap = #'((0 . 0) (0 . 1) (0 . 2))
c1\glissando |
```

```
<d f a> |
\set glissandoMap = #'((2 . 0) (1 . 0) (0 . 1))
<f d a'>1\glissando |
<c c'> |
```
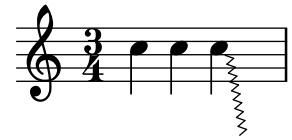


Se pueden crear distintos tipos de glissando. Para ver más detalles, consulte Sección 5.4.7 [Estilos de línea], página 614.

Fragmentos de código seleccionados

Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Añadir marcas de tiempo a glissandos largos

Los pulsos que se saltan en glissandos muy largos se indican a veces mediante marcas de tiempo, que a menudo consisten en figuras sin cabeza. Estas plicas se pueden usar también para albergar indicaciones expresivas intermedias.

Si las plicas no quedan bien alineadas con el glissando, podría ser necesario recolocarlas ligeramente.
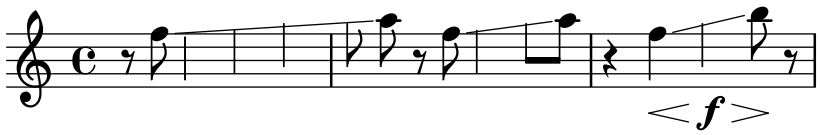
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
```

```
glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8

  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
  a8 |

  r4 f\glissando \<
  \glissandoSkipOn
  a4\f \>
  \glissandoSkipOff
  b8\! r |
}
```

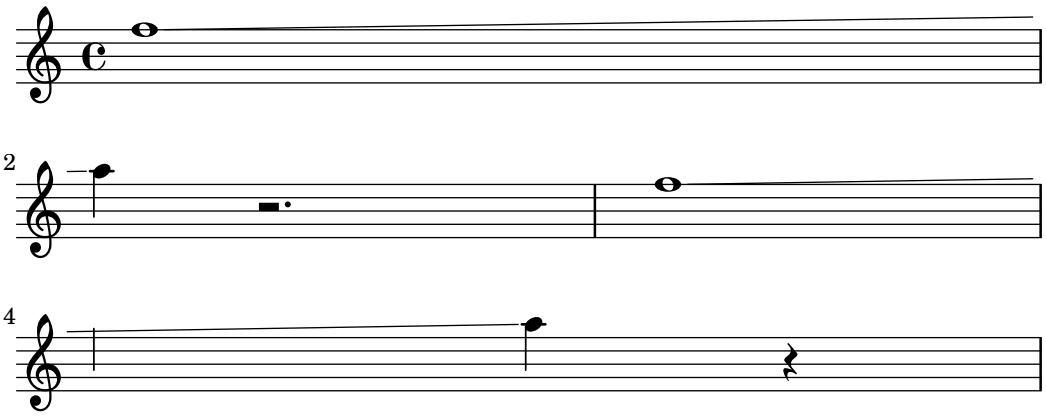


Hacer que los glissandos se puedan dividir en el salto de línea

Si se ajusta la propiedad `breakable` al valor `#t` en combinación con `after-line-breaking`, podemos hacer que un glissando se divida en el salto de línea:

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

\relative c'' {
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  f1\glissando |
  \break
  a4 r2. |
  f1\glissando
  \once \glissandoSkipOn
  \break
  a2 a4 r4 |
}
```



Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

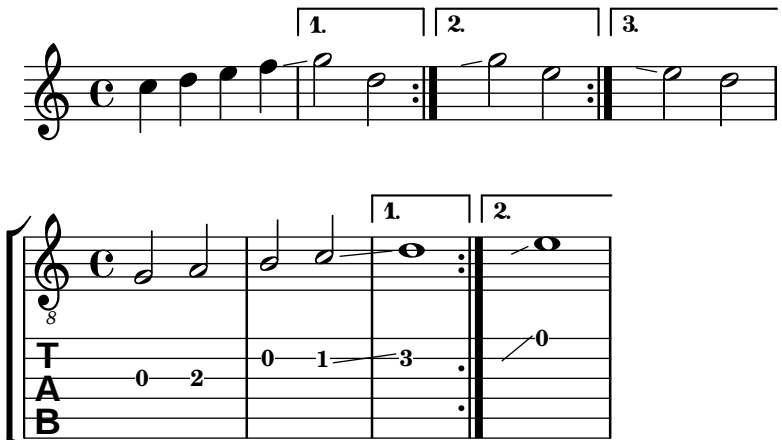
Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

```
repeatGliss = #(define-music-function (parser location grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = #3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c e1 }
  }
}
```

```
\score {
  \new StaffGroup <<
    \new Staff <<
      \context Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \context TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```



Véase también

Glosario musical: Sección “glissando” in *Glosario Musical*.

Referencia de la notación: Sección 5.4.7 [Estilos de línea], página 614.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Glissando” in *Referencia de Funcionamiento Interno*.
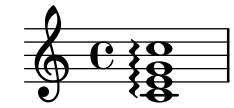
Advertencias y problemas conocidos

La impresión de texto sobre la línea (como *gliss.*) no está contemplada.

Arpegio

Un signo de *acorde arpegiado* (conocido también como acorde quebrado) sobre un acorde, se denota adjuntando `\arpeggio` al acorde:

```
<c e g c>1\arpeggio
```



Se pueden escribir distintos tipos de acordes arpegiados. `\arpeggioNormal` produce la vuelta al arpegio normal:

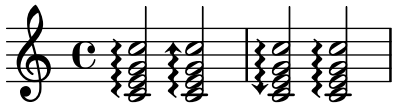
```
<c e g c>2\arpeggio

\arpeggioArrowUp
```

```
<c e g c>2\arpeggio

\arpeggioArrowDown
<c e g c>2\arpeggio

\arpeggioNormal
<c e g c>2\arpeggio
```



Se pueden crear símbolos especiales de arpegio *con corchete*:
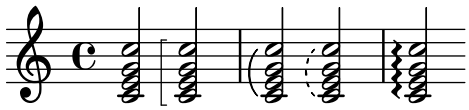
```
<c e g c>2

\arpeggioBracket
<c e g c>2\arpeggio

\arpeggioParenthesis
<c e g c>2\arpeggio

\arpeggioParenthesisDashed
<c e g c>2\arpeggio

\arpeggioNormal
<c e g c>2\arpeggio
```



Las propiedades de discontinuidad del paréntesis del arpegio se controlan mediante la propiedad `'dash-details`, que se describe en [Ligaduras de expresión], página 127.

Los acordes arpegiados se pueden desarrollar explícitamente utilizando *ligaduras de unión*. Véase [Ligaduras de unión], página 51.

Instrucciones predefinidas

`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`, `\arpeggioParenthesis` `\arpeggioParenthesisDashed`.

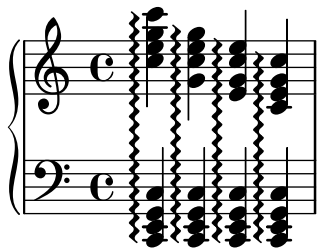
Fragmentos de código seleccionados

Crear arpegios que se cruzan entre pentagramas dentro de un sistema de piano

Dentro de un `PianoStaff`, es posible hacer que un arpegio cruce entre los pentagramas ajustando la propiedad `PianoStaff.connectArpeggios`.

```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
```

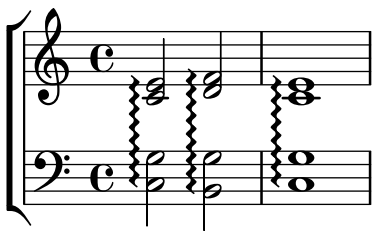
```
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
>>
```



Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos

Se pueden crear arpegios que se cruzan entre pentagramas dentro de contextos distintos a `GrandStaff`, `PianoStaff` y `StaffGroup` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

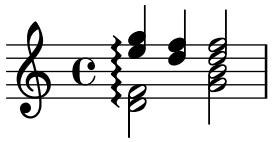
```
\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
    >>
  }
  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}
```



Crear arpegios entre notas de voces distintas

Se puede trazar un símbolo de arpegio entre notas de distintas voces que están sobre el mismo pentagrama si el grabador `Span_arpeggio_engraver` se traslada al contexto de `Staff` context:

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Véase también

Glosario musical: Sección “arpeggio” in *Glosario Musical*.

Referencia de la notación: [Ligaduras de expresión], página 127, [Ligaduras de unión], página 51.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Arpeggio” in *Referencia de Funcionamiento Interno*, Sección “Slur” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No es posible mezclar arpegios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

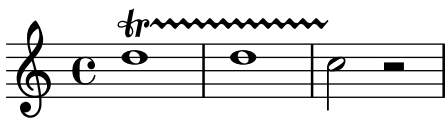
La forma sencilla de especificar el corchete en los arpegios de paréntesis no funciona para los arpegios de pentagrama cruzado; see [Plicas de pentagrama cruzado], página 320.

Trinos

Los trinos cortos sin línea extensora se imprimen con `\trill`; véase [Articulaciones y ornamentos], página 116.
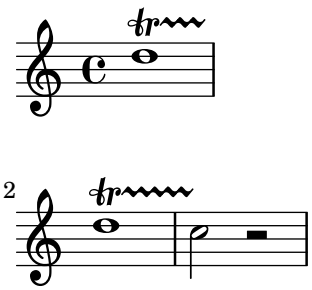
Los *trinos* largos mantenidos, con línea de extensión, se hacen con `\startTrillSpan` y `\stopTrillSpan`:

```
d1\startTrillSpan
d1
c2\stopTrillSpan
r2
```



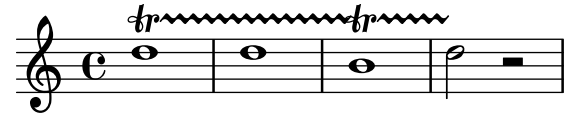
Un trino extendido que atraviesa un salto de línea recomienza exactamente encima de la primera nota de la nueva línea.

```
d1\startTrillSpan
\break
d1
c2\stopTrillSpan
r2
```
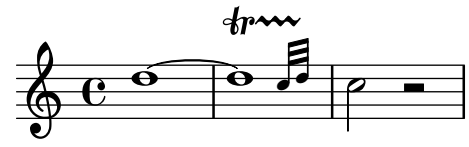


Los trinos extendidos consecutivos funcionan sin necesidad de instrucciones `\stopTrillSpan` explícitas, porque cada trino se convertirá automáticamente en el borde derecho del trino anterior.

```
d1\startTrillSpan
d1
b1\startTrillSpan
d2\stopTrillSpan
r2
```



También se pueden combinar los trinos con notas de adorno. La sintaxis de esta construcción y el método para colocar las notas de adorno con precisión están descritos en [Notas de adorno], página 108.

```
d1~\afterGrace
d1\startTrillSpan { c32[ d]\stopTrillSpan }
c2 r2
```
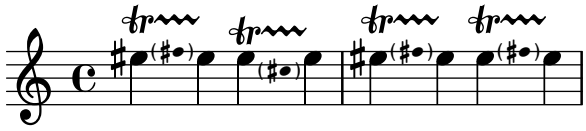


Los trinos que se tienen que ejecutar sobre notas auxiliares explícitas se pueden tipografiar con la instrucción `pitchedTrill`. El primer argumento es la nota principal, y el segundo es la nota *trinada*, que se imprime como una cabeza de nota, sin plica y entre paréntesis.

```
\pitchedTrill
d2\startTrillSpan fis
d2
c2\stopTrillSpan
r2
```



Es necesario añadir manualmente las alteraciones subsiguientes de la misma nota dentro del mismo compás. Sólo se imprime la alteración del primer trino con nota, dentro de un compás.

```
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
```



Instrucciones predefinidas

`\startTrillSpan`, `\stopTrillSpan`.

Véase también

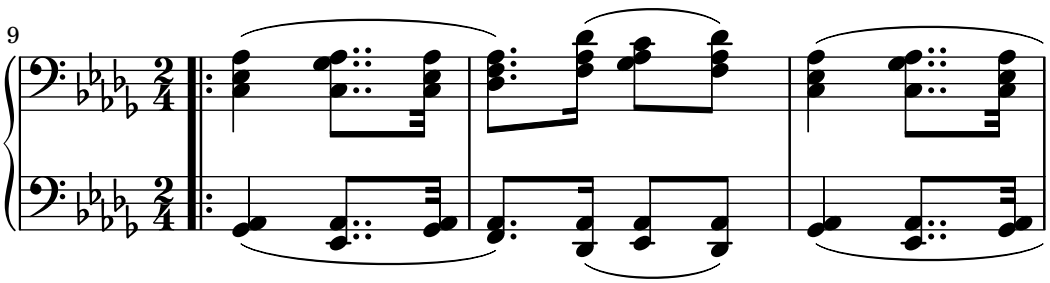
Glosario musical: Sección “trill” in *Glosario Musical*.
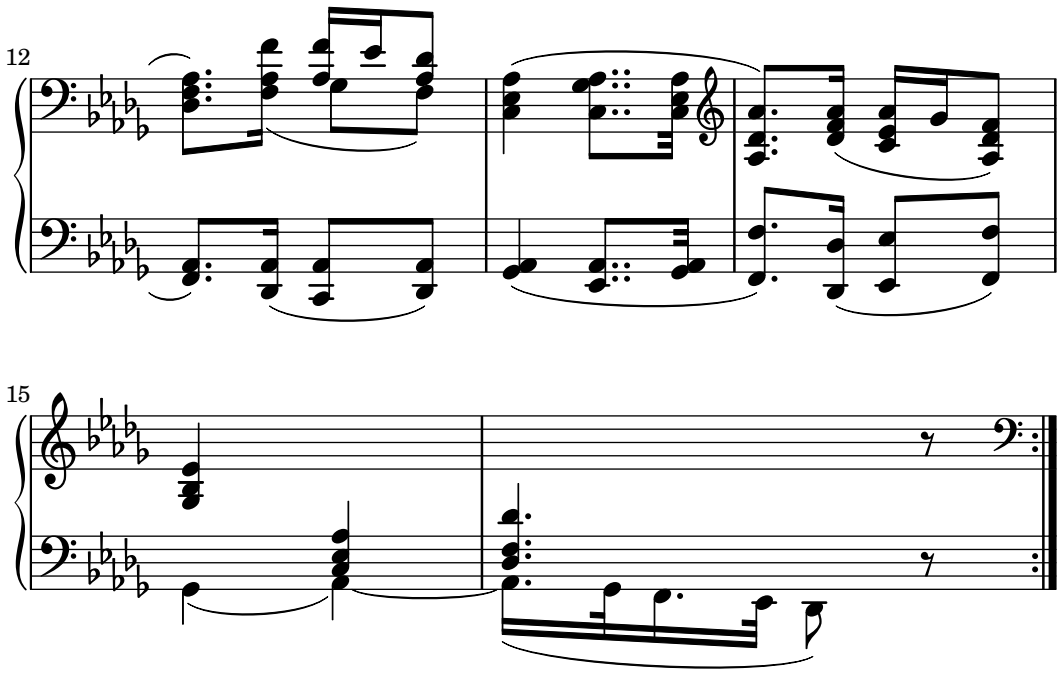
Referencia de la notación: [Articulaciones y ornamentos], página 116, [Notas de adorno], página 108.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TrillSpanner” in *Referencia de Funcionamiento Interno*.

1.4 Repeticiones





La repetición es un concepto fundamental en música, y existen varios tipos de notación para las repeticiones. LilyPond contempla los siguientes tipos de repetición:

`volta (primera y segunda vez)`
: La repetición de la música no se imprime de forma desarrollada, sino que se indica encerrándola entre barras de repetición. Si el salto de la repetición se encuentra al comienzo de una pieza, la barra de repetición sólo se imprime al final del fragmento. Se imprimen una serie de finales alternativos (volte) de izquierda a derecha indicados mediante corchetes. Ésta es la notación estándar para las repeticiones con finales alternativos.

`unfold (desplegada)`
: Las música repetida se escribe y se interpreta completamente tantas veces como especifique el valor *`número_de_repeticiones`*. Es útil cuando se está escribiendo música repetitiva.

`percent (porcentaje)`
: Hacer repeticiones de compases o parte de ellos. Tienen un aspecto semejante a un signo de porcentaje. Las repeticiones de porcentaje se deben declarar dentro de un contexto `Voice`.

`tremolo (trémolo)`
: Hacer barras de trémolo.

1.4.1 Repeticiones largas

Esta sección trata sobre la forma de introducir repeticiones largas, normalmente de varios compases. Las repeticiones adoptan dos formas: repeticiones encerradas entre signos de repetición, o repeticiones explícitas, que se usan para escribir música repetitiva. También se pueden controlar manualmente los signos de repetición.

Repeticiones normales
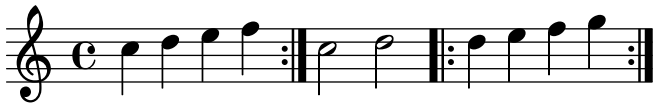
La sintaxis de una repetición normal es

```
\repeat volta número_de_repeticiones expresión_musical
```

donde *`expresión_musical`* es una expresión musical.

Una repetición simple sin finales alternativos:

```
\repeat volta 2 { c4 d e f }
c2 d
\repeat volta 2 { d4 e f g }
```



Los filanes alternativos (casillas de primera y segunda vez) se pueden generar utilizando `\alternative`. Cada grupo de alternativas debe, a su vez, estar encerrado entre llaves curvas.
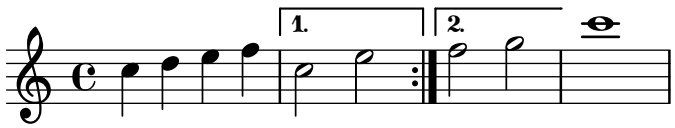
```
\repeat volta número_de_repeticiones expresión_musical
\alternative {
  { expresión_musical }
}
```

donde *`expresión_musical`* es una expresión musical.

Si existen más repeticiones que finales alternativos, se asigna el primer final alternativo a las repeticiones más antiguas.
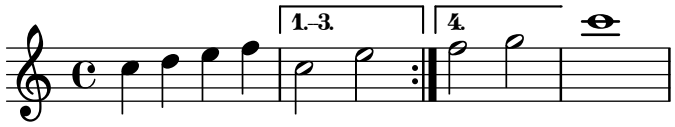
Repetición única con primera y segunda vez:

```
\repeat volta 2 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
}
c1
```



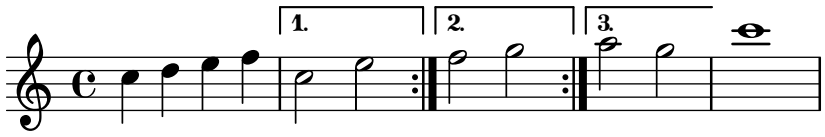
Varias repeticiones con primera y segunda vez:

```
\repeat volta 4 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
}
c1
```



Más de una repetición con más de un final alternativo:

```
\repeat volta 3 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
  { a2 g | }
}
c1
```
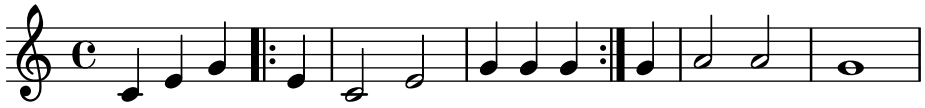


Nota: Si hay dos o más alternativas, no debe aparecer nada entre la llave de cierre de una y la de apertura de la otra dentro de un bloque `\alternative`, pues en caso contrario no obtendremos el número de finales esperado.

Nota: Si incluimos `\relative` dentro de un `\repeat` sin instanciar el contexto `Voice` explícitamente, aparecerán pentagramas adicionales no deseados. Véase Sección "Aparece un pentagrama de más" in *Utilización del Programa*.
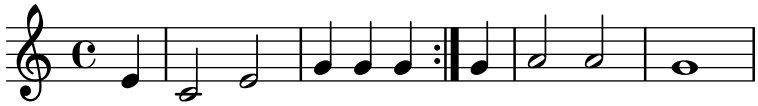
Si una repetición comienza en medio de un compás y no tiene finales alternativos (primera y segunda vez), normalmente el final de la repetición también caerá en el medio de un compás, de forma que el comienzo y el final formen un compás completo. En tales casos, los signos de repetición no son verdaderas líneas divisorias. No utilice instrucciones `\partial` o comprobaciones de compás en los lugares en que se imprimen estos símbolos de repetición:

```
% no \partial here
c4 e g  % no bar check here
% no \partial here
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g  % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```



De forma similar, si una repetición compienza con el compás de anacrusa inicial de una partitura y no tiene finales alternativos, se dan las mismas condiciones que en el ejemplo anterior, excepto que en este caso se necesita la instrucción `\partial` al principio de la partitura:
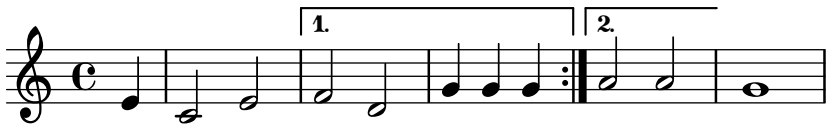
```
\partial 4  % required
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g  % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```



Cuando se añaden primera y segunda vez a una repetición que comienza con un compás incompleto, se hace necesario establecer la propiedad de contexto `Timing.measureLength` manualmente, en los siguiente lugares específicos:

- en el comienzo de los complases incompletos del bloque `\alternative`, que normalmente se producen al final de cada final alternativo, excepto (en la mayoría de los casos) el último de ellos.
- en el comienzo de cada uno de los finales alternativos, excepto el primero.

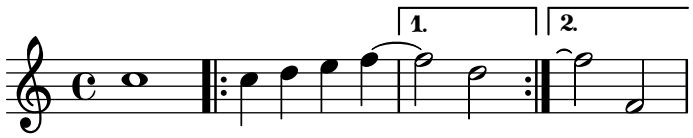
```
\partial 4
\repeat volta 2 { e4 | c2 e | }
\alternative {
  {
    f2 d |
    \set Timing.measureLength = #(ly:make-moment 3/4)
    g4 g g  % optional bar check is allowed here
  }
  {
    \set Timing.measureLength = #(ly:make-moment 4/4)
    a2 a |
  }
}
g1 |
```



La propiedad `measureLength` se describe en [Gestión del tiempo], página 114.

Se pueden añadir ligaduras de unión al segundo final:

```
c1
\repeat volta 2 { c4 d e f~ }
\alternative {
  { f2 d }
  { f2\repeatTie f, }
}
```
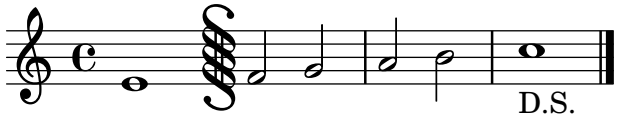


La instrucción `\inStaffSegno` se puede utilizar para generar una barra de compás combinada que incorpora el símbolo del segno junto con la barra de repetición, cuando se usa con la instrucción `\repeat volta`. Se selecciona automáticamente el tipo correcto de línea divisoria de repetición, ya sea de comienzo o fin de repeticion, o de doble repetición. Observe que la indicación “D.S.” correspondiente se debe añadir manualmente.

Salida de una repetición:
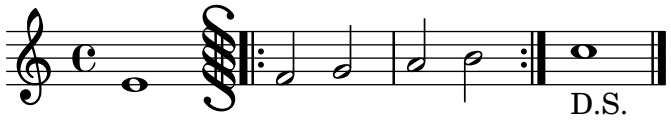
```
e1
\inStaffSegno
f2 g a b
```

```
c1_"D.S." \bar "|."
```



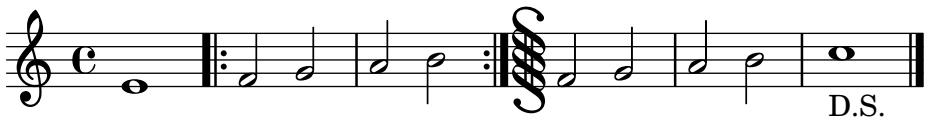
Al comienzo de una repetición:

```
e1
\repeat volta 2 {
  \inStaffSegno  % start repeat
  f2 g a b
}
c1_"D.S." \bar "|."
```



Al final de una repetición:

```
e1
\repeat volta 2 {
  f2 g a b
  \inStaffSegno  % end repeat
}
f2 g a b
c1_"D.S." \bar "|."
```



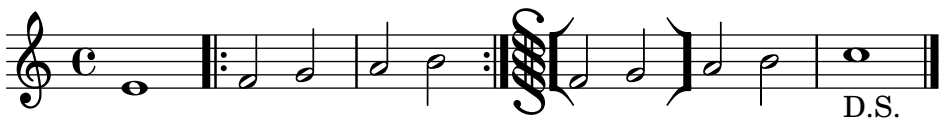
Entre dos repeticiones:

```
e1
\repeat volta 2 {
  f2 g a b
}
\inStaffSegno  % double repeat
\repeat volta 2 {
  f2 g a b
}
c1_"D.S." \bar "|."
```



Se pueden obtener símbolos de línea divisoria alternativos mediante el establecimiento (en el contexto Score) de las propiedades `segnoType`, `startRepeatSegnoType`, `endRepeatSegnoType` o `doubleRepeatSegnoType` al tipo de línea requerido. Los tipos de línea divisoria alternativos deben seleccionarse a partir de los tipos predefinidos o de tipos definidos anteriormente con la instrucción `\defineBarLine` (véase [Barras de compás], página 94).

```
\defineBarLine ":|.S[" #'(":|." "S[" "")
\defineBarLine "]" #'("]" "" "")
e1
\repeat volta 2 {
  f2 g a b
  \once \set Score.endRepeatSegnoType = ":|.S["
  \inStaffSegno
}
f2 g \bar "]" a b
c1_"D.S." \bar "|."
```
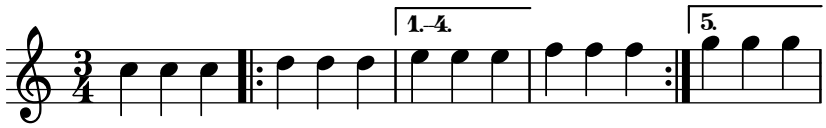


Fragmentos de código seleccionados

Acortar los corchetes de primera y segunda vez

De forma predeterminada, los corchetes de primera y segunda vez se trazan encima de los finales alternativos completos, pero es posible acortartlos estableciendo un valor cierto para `voltaSpannerDuration`. En el ejemplo siguiente, el corchete sólo dura un compás, que corresponde a una duración de 3/4.

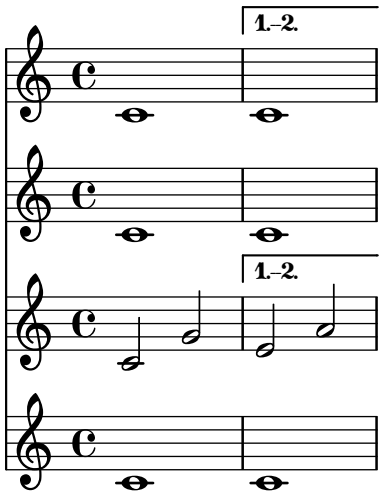
```
\relative c'' {
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3/4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



Añadir corchetes de primera y segunda vez a más pentagramas

El grabador `Volta_engraver` reside de forma predeterminada dentro del contexto de `Score`, y los corchetes de la repetición se imprimen así normalmente sólo encima del pentagrama superior. Esto se puede ajustar añadiendo el grabador `Volta_engraver` al contexto de `Staff` en que deban aparecer los corchetes; véase también el fragmento de código "Volta multi staff".
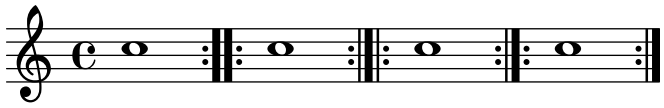
```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



Establecer el tipo de repetición doble predeterminado para la primera y segunda vez

Existen tres estilos distintos de repetiones dobles para la primera y segunda vez, que se pueden ajustar utilizando `doubleRepeatType`.
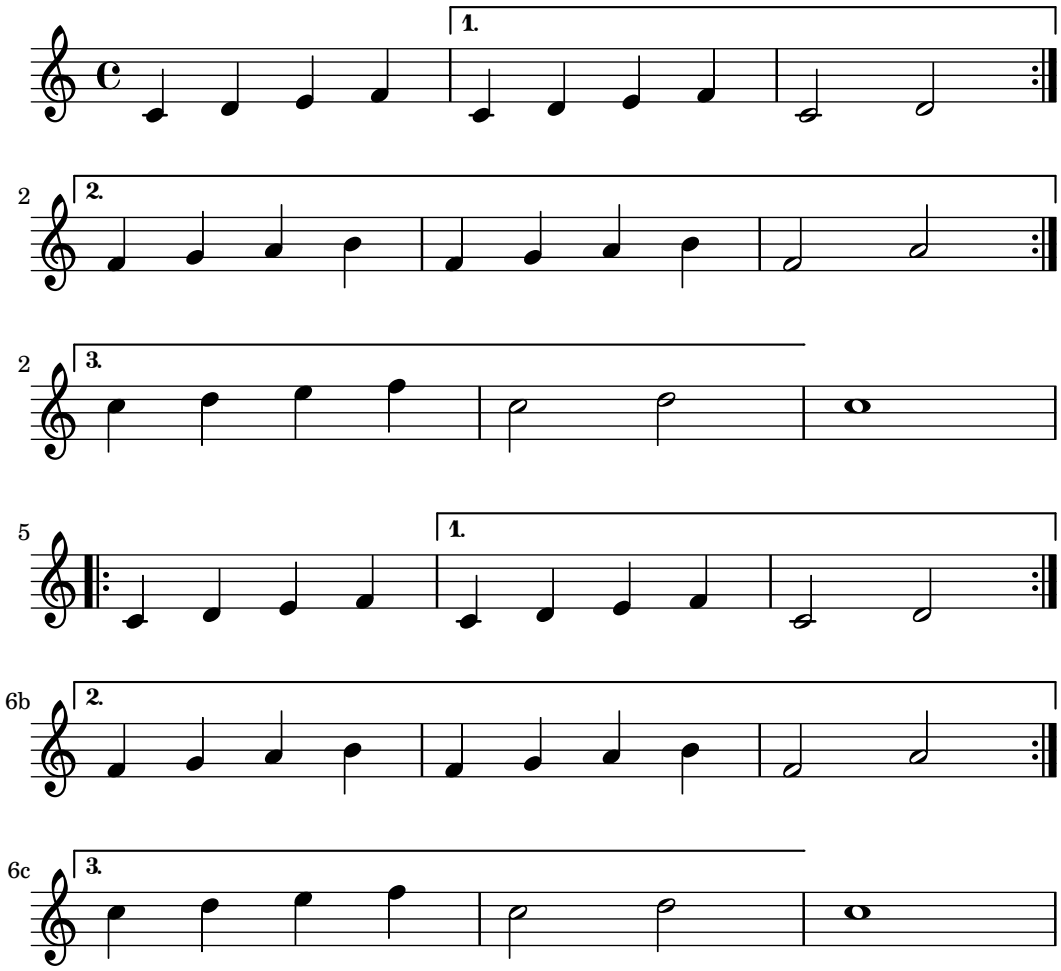
```
\relative c'' {
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":..:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.:"
  \repeat volta 1 { c1 }
}
```



Numeración de compases alternativa

Se pueden seleccionar dos métodos alternativos para la numeración de compases, especiales para cuando hay repeticiones.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1
}
```



Véase también

Glosario musical: Sección “repeat” in *Glosario Musical*, Sección “volta” in *Glosario Musical*.

Referencia de la notación: [Barras de compás], página 94, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 575, [Modificación de ligaduras de unión y de expresión], página 622, [Gestión del tiempo], página 114.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” in *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las ligaduras de expresión que abarcan desde un bloque `\repeat` hasta un bloque `\alternative` sólo funcionan para la casilla de primera vez. La apariencia visual de una ligadura de expresión que se continúa introduciéndose en otras casillas de repetición puede simularse con `\repeatTie` si la ligadura se extiende hasta una sola nota dentro de la casilla de repetición, aunque este método no funciona en `TabStaff`. Otros métodos que pueden venir bien para indicar ligaduras que se continúan por encima de varias notas en las casillas de repetición, y que también funcionan en contextos de tablatura `TabStaff`, están explicados en [Modificación de ligaduras de unión y de expresión], página 622.

Asimismo, no es posible hacer que las ligaduras de expresión se replieguen desde el final de una alternativa hasta el comienzo de la repetición.

Los glissandos que se extienden desde un bloque `\repeat` hasta un bloque `\alternative` solamente funcionan para la casilla de primera vez. Se puede indicar la apariencia visual de un

glissando que se continúa hasta el interior de otras casillas de repetición mediante la codificación de un glissando que empieza en una nota de adorno oculta. Para ver un ejemplo, consulte "Extender los glissandos sobre las repeticiones" bajo el epígrafe Fragmentos de código seleccionados, en [Glissando], página 134.

Si una repetición que comienza con un compás incompleto tiene un bloque `\alternative` que contiene modificaciones a la propiedad `measureLength`, la utilización de `\unfoldRepeats` dará lugar a líneas divisorias erróneamente colocadas y advertencias de comprobación de compás.

Una repetición anidada como

```
\repeat ...
\repeat ...
\alternative
```

es ambigua porque no está claro a qué `\repeat` pertenece la `\alternative`. Esta ambigüedad se resuelve haciendo que la `\alternative` pertenezca siempre a la `\repeat` más interna. Para más claridad, se recomienda usar llaves en tales situaciones.

Marcas de repetición manual
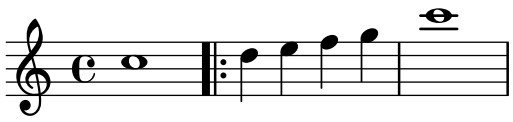
> **Nota:** Estos métodos sólo se utilizan para realizar construcciones de repetición poco usuales, y pueden tener un comportamiento distinto al esperado. En casi todas las situaciones, se deben crear las repeticiones utilizando la instrucción estándar `\repeat` o imprimiendo las barras de compás correspondientes. Para ver más información, consulte [Barras de compás], página 94.

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme de instrucciones de repetición.

`start-repeat`

Imprimir una barra de compás `.|:`

```
c1
\set Score.repeatCommands = #'(start-repeat)
d4 e f g
c1
```
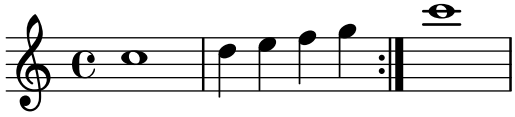


En la práctica habitual del grabado no se imprimen signos de repetición al principio de la pieza.

`end-repeat`

Imprimir una barra de compás `:|.`
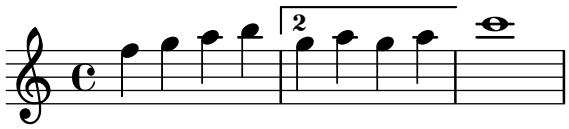
```
c1
d4 e f g
\set Score.repeatCommands = #'(end-repeat)
c1
```



`(volta` *`number`*`) ... (volta #f)`

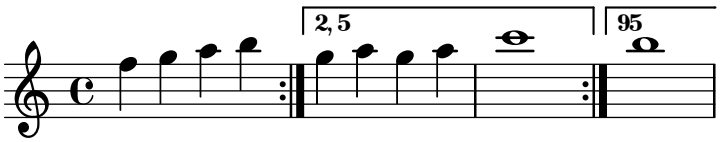
Crear una nueva casilla de repetición con el número que se especifica. El corchete de vez se debe terminar de forma explícita, pues en caso contrario no se imprime.

```
f4 g a b
\set Score.repeatCommands = #'((volta "2"))
g4 a g a
\set Score.repeatCommands = #'((volta #f))
c1
```
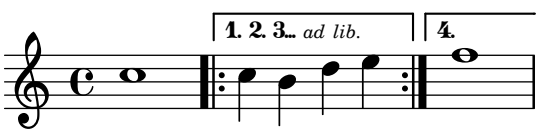


Se pueden producir varias instrucciones de repetición en el mismo punto:

```
f4 g a b
\set Score.repeatCommands = #'((volta "2, 5") end-repeat)
g4 a g a
c1
\set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
b1
\set Score.repeatCommands = #'((volta #f))
```



Se puede incluir texto dentro de la casilla de primera y segunda vez. El texto puede ser un número o números, o un elemento de marcado, véase Sección 1.8.2 [Formatear el texto], página 233. La forma más fácil de usar texto de marcado es definir el marcado previamente, y luego incluirlo dentro de una lista de Scheme.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative c'' {
  c1
  \set Score.repeatCommands =
    #(list(list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```

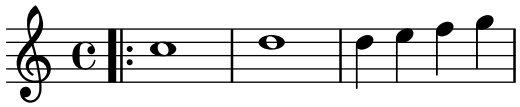


Fragmentos de código seleccionados

Imprimir puntos de repetición al prinicpio de la pieza

Se puede imprimir una línea divisoria de la forma `.|:` al principio de la pieza, sobreescribiendo la propiedad correspondiente:

```
\relative c'' {
  \once \override Score.BreakAlignment.break-align-orders =
    #(make-vector 3 '(instrument-name
                      left-edge
                      ambitus
                      breathing-sign
                      clef
                      key-signature
                      time-signature
                      staff-bar
                      custos))
  \once \override Staff.TimeSignature.space-alist =
    #'((first-note . (fixed-space . 2.0))
       (right-edge . (extra-space . 0.5))
       ;; free up some space between time signature
       ;; and repeat bar line
       (staff-bar . (extra-space . 1)))
  \bar ".|:"
  c1
  d1
  d4 e f g
}
```



Véase también

Referencia de la notación: [Barras de compás], página 94, Sección 1.8.2 [Formatear el texto], página 233.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” in *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” in *Referencia de Funcionamiento Interno*.
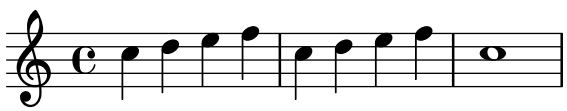
Repeticiones explícitas

Mediante la utilización de la instrucción `unfold` se pueden usar las repeticiones para simplificar la escritura desplegada de música repetitiva. La sintaxis es:

```
\repeat unfold número_de_repeticiones expresión_musical
```

donde *`expresión_musical`* es una expresión musical y *`número_de_repeticiones`* es el número de veces que *`expresión_musical`* se repite.

```
\repeat unfold 2 { c4 d e f }
c1
```



En ciertos casos, especialmente dentro de un contexto `\relative`, la función `\repeat unfold` no es exactamente igual que escribir la expresión musical varias veces. P. ej.:
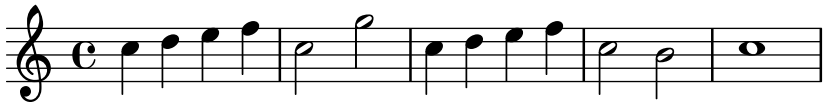
```
\repeat unfold 2 { a'4 b c }
```

no equivale a

```
a'4 b c | a'4 b c
```
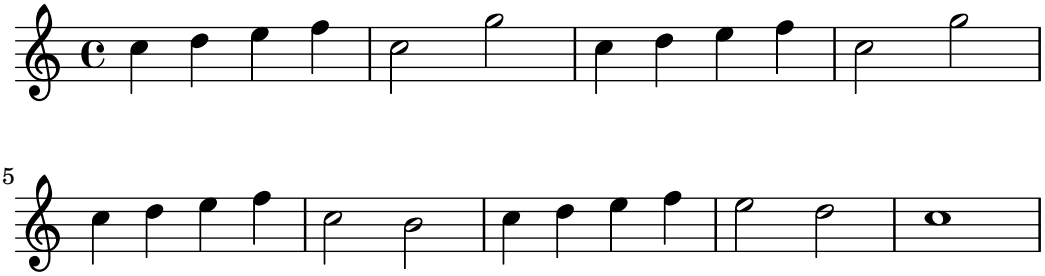
Se pueden hacer repeticiones desplegadas con finales alternativos.

```
\repeat unfold 2 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
}
c1
```
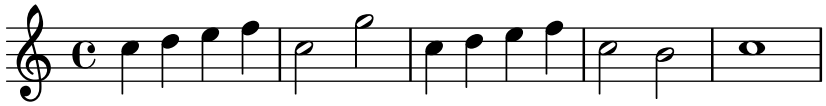


Si hay más repeticiones que finales alternativos, el primer final alternativo se aplica las veces necesarias hasta que las alternativas restantes completan el número total de repeticiones.

```
\repeat unfold 4 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
 }
c1
```



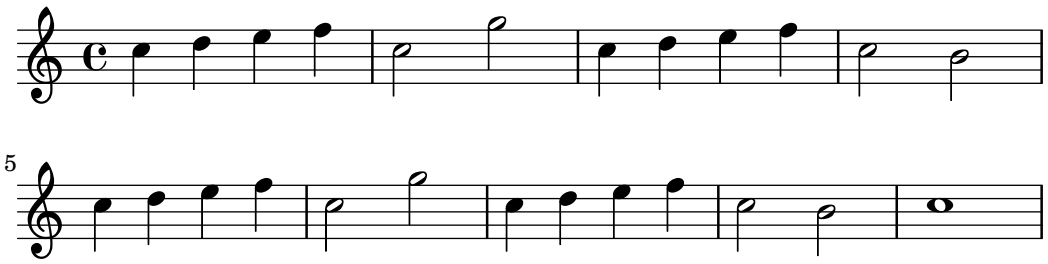
Si existen más finales alternativos que repeticiones, se aplican solo los primeros finales alternativos. Las alternativas restantes se ignoran y no se imprimen.

```
\repeat unfold 2 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
}
c1
```



Es posible también anidar varias funciones `unfold` (con finales alternativos o sin ellos).

```
\repeat unfold 2 {
  \repeat unfold 2 { c4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
  }
}
c1
```



Las construcciones de acorde se pueden repetir mediante el símbolo de repetición de acordes `q`. Véase [Repetición de acordes], página 163.

> **Nota:** Si pone `\relative` dentro de un bloque `\repeat` sin instanciar explícitamente el contexto `Voice`, aparecerán pentagramas no deseados. Véase Sección "Aparece un pentagrama de más" in *Utilización del Programa*.

Véase también

Referencia de la notación: [Repetición de acordes], página 163.

Snippets: Fragmentos de código: Sección "Repeats" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "RepeatedMusic" in *Referencia de Funcionamiento Interno*, Sección "UnfoldedRepeatedMusic" in *Referencia de Funcionamiento Interno*.

1.4.2 Repeticiones cortas

Esta sección trata de cómo introducir repeticiones cortas. Las repeticiones cortas pueden adoptar dos formas: barras inclinadas o signos de porcentaje si estamos representando repeticiones de una sola nota, un compás o dos compases, y trémolos en caso contrario.

Repeticiones de compás o parte de ellos

Los patrones cortos que se repiten se imprimen una sola vez, y el patrón repetido se sustituye por un símbolo especial.

La sintaxis es:

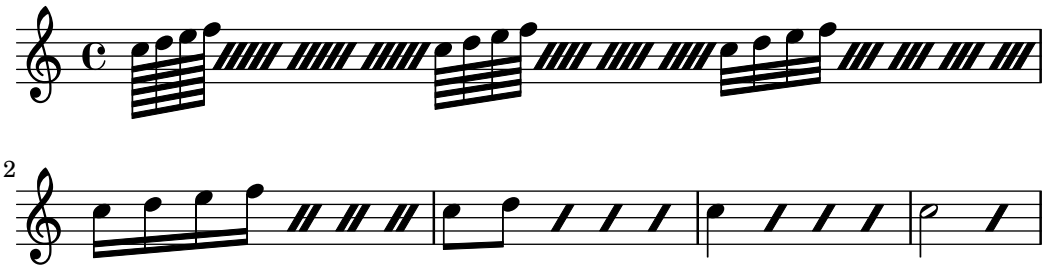
```
\repeat percent número expresión_musical
```

donde *`expresión_musical`* es una expresión musical.

Los patrones más cortos que un compás se sustituyen por barras inclinadas.
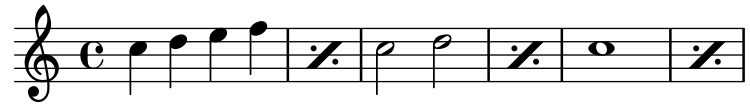
```
\repeat percent 4 { c128 d e f }
\repeat percent 4 { c64 d e f }
\repeat percent 5 { c32 d e f }
\repeat percent 4 { c16 d e f }
\repeat percent 4 { c8 d }
```

```
\repeat percent 4 { c4 }
\repeat percent 2 { c2 }
```
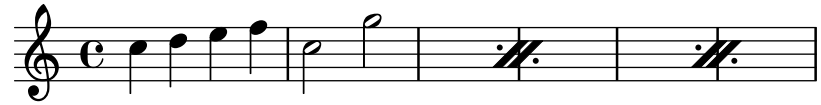


Los patrones de repetición de uno o dos compases se sustituyen por signos parecidos al símbolo de porcentaje.

```
\repeat percent 2 { c4 d e f }
\repeat percent 2 { c2 d }
\repeat percent 2 { c1 }
```

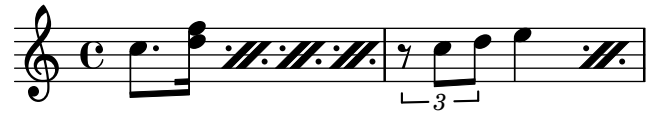


```
\repeat percent 3 { c4 d e f | c2 g' }
```



Los patrones más cortos que un compás pero que contienen duraciones mezcladas utilizan un símbolo de porcentaje doble.

```
\repeat percent 4 { c8. <d f>16 }
\repeat percent 2 { \tuplet 3/2 { r8 c d } e4 }
```
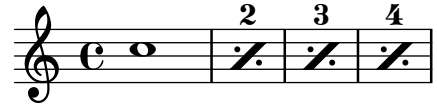


Fragmentos de código seleccionados

Contador de repeticiones de tipo porcentaje

Las repeticiones de compases completos de más de dos repeticiones pueden llevar un contador si se activa la propiedad adecuada, como se ve en este ejemplo:

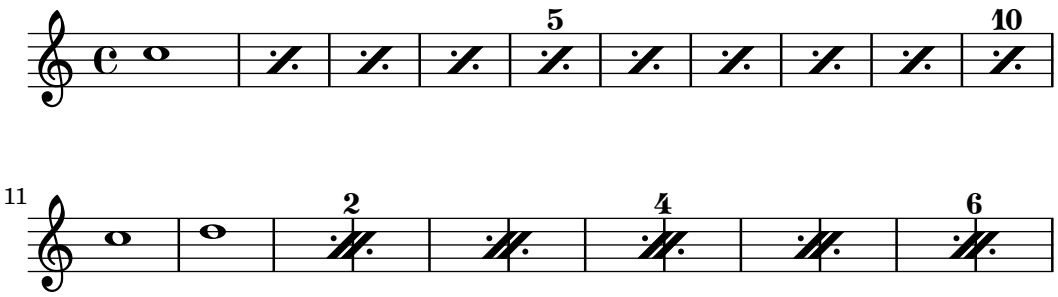
```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Visibilidad del contador de repeticiones de tipo porcentaje

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto `repeatCountVisibility`.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```
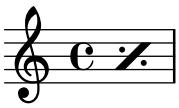


Símbolos de porcentaje sueltos

También se pueden imprimir símbolos de porcentaje sueltos.

```
makePercent =
#(define-music-function (parser location note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Véase también

Glosario musical: Sección “percent repeat” in *Glosario Musical*, Sección “simile” in *Glosario Musical*.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

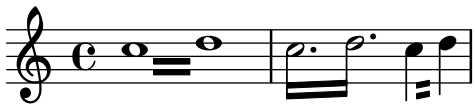
Referencia de funcionamiento interno: Sección “RepeatSlash” in *Referencia de Funcionamiento Interno*, Sección “RepeatSlashEvent” in *Referencia de Funcionamiento Interno*, Sección “DoubleRepeatSlash” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeat” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeatCounter” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “Percent_repeat_engraver” in *Referencia de Funcionamiento Interno*, Sección “DoublePercentEvent” in *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeat” in *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeatCounter” in *Referencia de Funcionamiento Interno*, Sección “Double_percent_repeat_engraver” in *Referencia de Funcionamiento Interno*, Sección “Slash_repeat_engraver” in *Referencia de Funcionamiento Interno*.

Repeticiones de trémolo

Los trémolos pueden adoptar dos formas: alternancia entre dos acordes o dos notas, y repetición rápida de una sola nota o acorde. Los trémolos que consisten en una alternancia se indican por medio de la adición de barras entre las notas o acordes que se alternan, mientras que los trémolos que consisten en la repetición rápida de una sola nota se indican mediante la adición de barras cruzadas a una nota única.

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo` (trémolo):

```
\repeat tremolo 8 { c16 d }
\repeat tremolo 6 { c16 d }
\repeat tremolo 2 { c16 d }
```
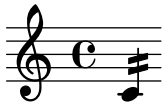


La sintaxis de `\repeat tremolo` espera que se escriban exactamente dos notas dentro de las llaves, y el número de repeticiones debe corresponderse con un valor que se pueda expresar con figuras normales o con puntillo. Así, `\repeat tremolo 7` es válido y produce una nota con doble puntillo, pero `\repeat tremolo 9` no es válido.

La duración del trémolo es igual a la duración de la expresión entre llaves, multiplicada por el número de repeticiones: `\repeat tremolo 8 { c16 d16 }` da como resultado un trémolo de redonda, escrito como dos redondas unidas por barras de trémolo.
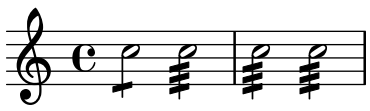
Existen dos maneras de colocar marcas de trémolo sobre una única nota. Incluso aquí se puede utilizar la sintaxis `\repeat tremolo`, en cuyo caso la nota no debe ir encerrada entre llaves:

```
\repeat tremolo 4 c'16
```



El mismo resultado se puede obtener escribiendo `:`*`N`* después de la nota, donde *`N`* indica la duración de la subdivisión (debe ser 8 como mínimo). Si *`N`* es 8, se añade una barra de corchea a la plica de la nota. Si *`N`* se omite, se utiliza el último valor (almacenado en `tremoloFlags`):

```
c2:8 c:32
c: c:
```
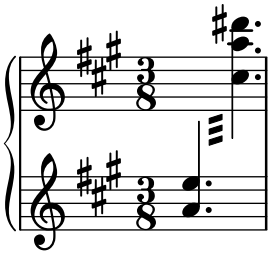


Fragmentos de código seleccionados

Trémolos de pentagrama cruzado

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
```

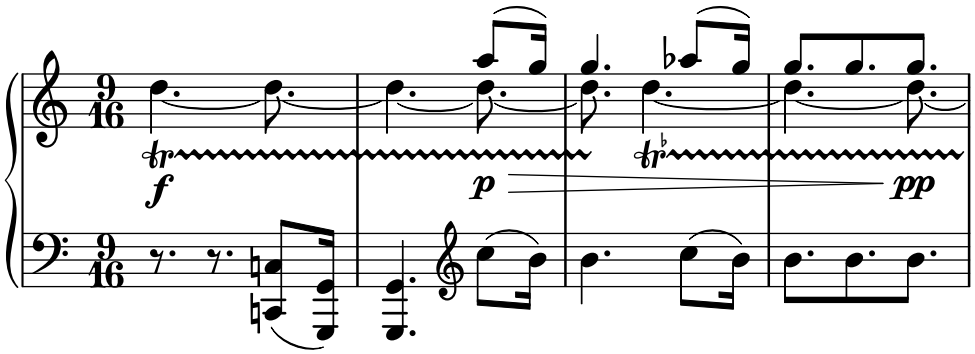
```
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```
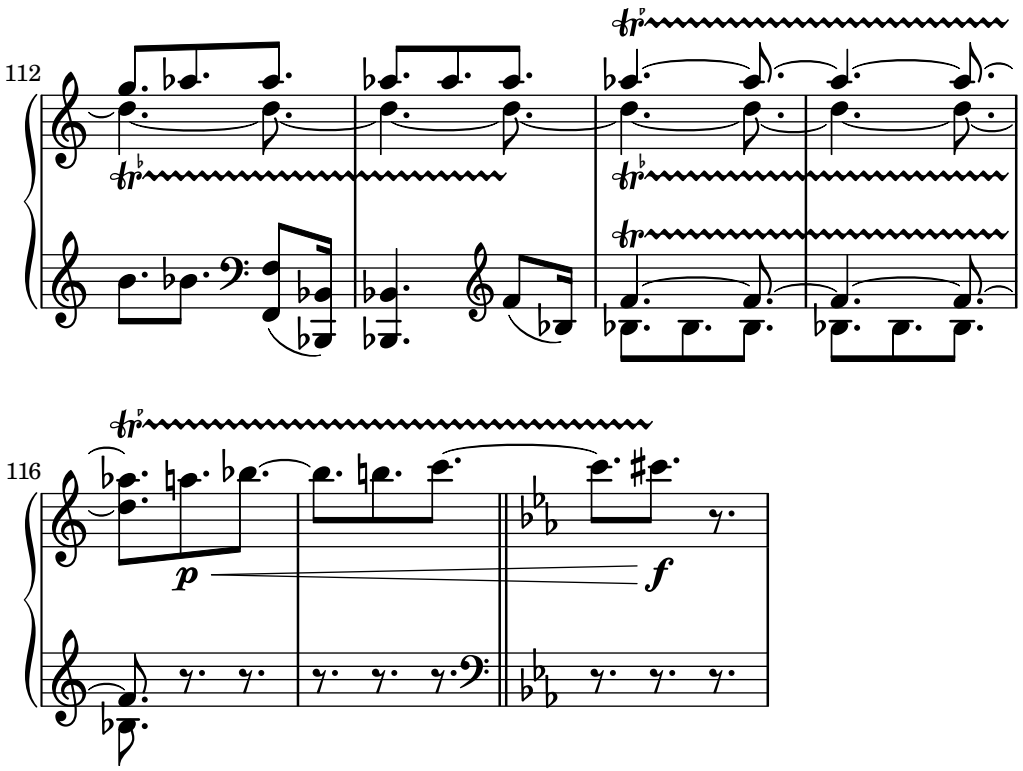


Véase también

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

1.5 Notas simultáneas





La polifonía en música hace referencia a tener más de una voz en una pieza cualquiera de música. En LilyPond la polifonía hace referencia a tener más de una voz en el mismo pentagrama.

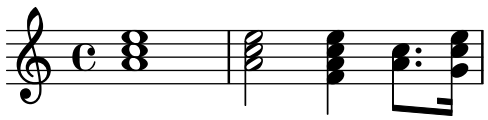
1.5.1 Una voz única

Esta sección trata de la notas simultáneas dentro de la misma voz.

Notas en acorde

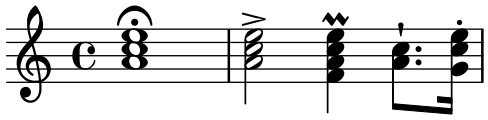
Un acorde se forma encerrando un conjunto de notas entre < y >. Un acorde puede ir seguido de una duración, como si fueran simples notas.

```
<a c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
```



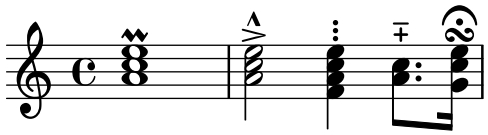
Los acordes también pueden ir seguidos de articulaciones, de nuevo como si fueran simples notas.

```
<a c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^! <g c e>16-.
```



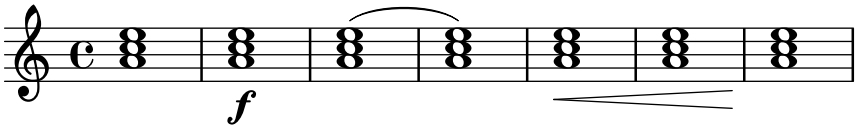
Las notas dentro del propio acorde también pueden ir seguidas de articulaciones y ornamentos.

```
<a c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4
<a-+ c-->8. <g\fermata c e\turn>16
```
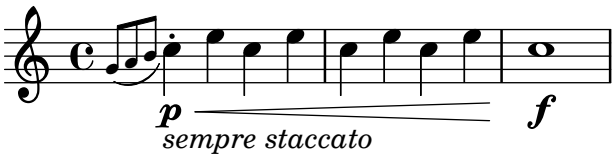


Sin embargo, algunos elementos de notación tales como las expresiones de matices dinámicos, los reguladores y las ligaduras de expresión, se deben unir al acorde y no a las notas que integran el mismo, pues en caso contrario no se imprimirán.

```
<a\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>)
<a c e>\< <a c e> <a c e>\!
```



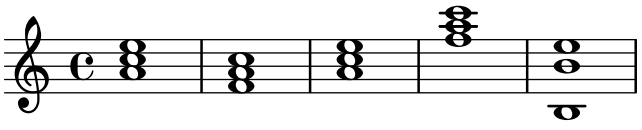
Un acorde funciona como un mero contenedor para las notas que lo componen y sus articulaciones y otros elementos adjuntos. Como consecuencia, un acorde sin ninguna nota en su interior no tiene realmente ninguna duración. Cualqier articulación adjunta se producirá en el mismo momento musical que la nota o acorde siguiente y se puede combinar con ellos (para ver posibilidades más complejas de combinaciones de estos elementos, consulte [Expresiones simultáneas], página 164):

```
\grace { g8( a b }
<> ) \p \< -. -\markup \italic "sempre staccato"
\repeat unfold 4 { c4 e }  c1\f
```



Se puede usar el modo relativo para la altura de las notas de los acordes. La primera nota del acorde siempre es relativa a la primera nota del acorde anterior, o en caso de que el elemento precedente no sea un acorde, la altura de la última nota que vino antes del acorde. El resto de las notas del acorde son relativas a la nota anterior *dentro del mismo acorde*.

```
<a c e>1 <f a c> <a c e> <f' a c> <b, e b,>
```



Para ver más información sobre los acordes, consulte Sección 2.7 [Notación de acordes], página 399.

Véase también

Glosario musical: Sección “chord” in *Glosario Musical*.

Manual de aprendizaje: Sección “Combinar notas para formar acordes” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.7 [Notación de acordes], página 399, [Articulaciones y ornamentos], página 116, [Escritura de octava relativa], página 2, Sección 1.5.2 [Varias voces], página 166.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Advertencias y problemas conocidos

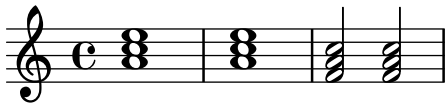
Los acordes que contienen más de dos alturas en el mismo espacio de pentagrama, como '`<e f! fis!>`', crean notas cuyas cabezas se superponen. Dependiendo de la situación, una mejor representación puede requerir:

- el uso temporal de varias voces, véase Sección 1.5.2 [Varias voces], página 166, '`<< f! \\ <e fis!> >>`',
- la transcripción enarmónica de una o más notas, '`<e f ges>`', or
- Culsters o racimos; véase [Racimos (clusters)], página 166.

Repetición de acordes

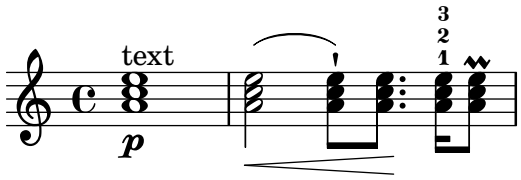
Para reducir el tecleo, se puede usar una abreviatura de repetición del acorde anterior. El símbolo de repetición de acordes es `q`:

```
<a c e>1 q <f a c>2 q
```



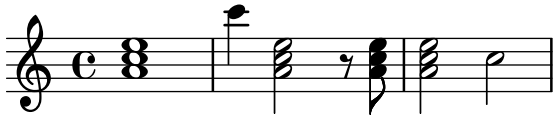
Como en los acordes normales, el símbolo de repetición de acordes puede usarse con duraciones, articulaciones, elementos de marcado, ligaduras de expresión, barras, etc., pues sólo se duplican las notas del acorde precedente.

```
<a c e>1\p^"text" q2\<( q8)[-! q8.]\! q16-1-2-3 q8\prall
```
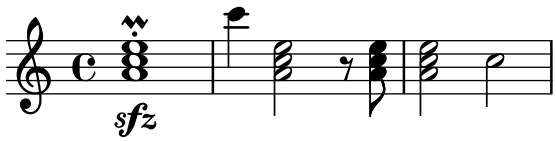


El símbolo de repetición de acordes siempre recuerda la última ocurrencia de un acorde, por lo que es posible repetir el acorde más reciente incluso si se han escrito en medio otras notas que no están en un acorde, o silencios.

```
<a c e>1 c'4 q2 r8 q8 |
q2 c, |
```



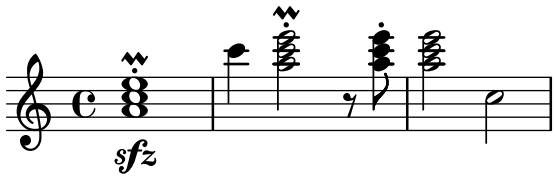
Sin embargo, el símbolo de repetición de acordes no retiene los matices dinámicos, las articulaciones ni los ornamentos que contiene o que están adosados al acorde anterior.

```
<a-. c\prall e>1\sfz c'4 q2 r8 q8 |
q2 c, |
```



Para poder retener algunos de ellos, se puede llamar explícitamente a la función `\chordRepeats` con un argumento adicional que especifique una lista de *tipos de evento* que conservar, a no ser que los eventos de ese tipo ya estén presentes en el propio acorde `q`.
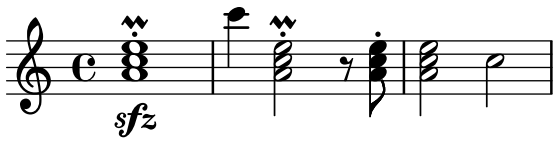
```
\relative c'' {
  \chordRepeats #'(articulation-event)
  { <a-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c, |
}
```



Aquí, el uso de `\chordRepeats` dentro de una construcción `\relative` produce un resultado inseperado: una vez se han expandido los eventos de acorde, no pueden distinguirse de aquellos introducidos como acordes normales, haciendo que `\relative` asigne una octava basada en su contexto actual.

Dado que las instancias anidadas de `\relative` no se afectan mutuamente, se puede usar otra instrucción `\relative` dentro de `\chordRepeats` para establecer las relaciones de octava antes de expandir los acordes repetidos. En tal caso, todo el contenido de la instrucción `\relative` interior no afecta a la exterior; de aquí la distinta escritura de octava en la última nota de este ejemplo.

```
\new Voice
\relative c'' {
  \chordRepeats #'(articulation-event)
  \relative c''
  { <a-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c |
}
```



Las interacciones con `\relative` se producen solamente con las llamadas explícitas de `\chordRepeats`: la expansión implícita al comienzo del proceso de tipografiado se hace en un momento en que todas las instancias de `\relative` ya se han procesado.

Véase también

Referencia de la notación: Sección 2.7 [Notación de acordes], página 399, [Articulaciones y ornamentos], página 116.

Archivos de inicio: ‘`ly/chord-repetition-init.ly`’.
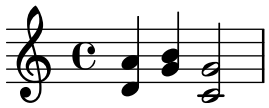
Expresiones simultáneas

Una o más expresiones musicales encerradas entre ángulos dobles se entienden como simultáneas. Si la primera expresión comienza con una sola nota o si toda la expresión simultánea aparece explícitamente dentro de una sola voz, la expresión completa se sitúa sobre un solo pentagrama; en caso contrario los elementos de la expresión simultánea se sitúan en pentagramas distintos.
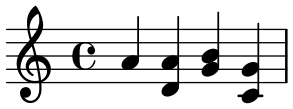
Los ejemplos siguientes muestran expresiones simultáneas sobre un solo pentagrama:

```
\new Voice {  % explicit single voice
  << { a4 b g2 } { d4 g c,2 } >>
```
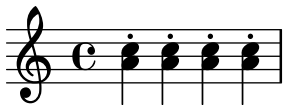
```
}
```



```
% single first note
a << { a4 b g } { d4 g c, } >>
```



Esto puede ser de utilidad si las secciones simultáneas tienen idénticas duraciones, pero se producirán errores si se intentan poner notas de distinta duración sobre la misma plica. Las notas, articulaciones y cambios de propiedades que están dentro de *un solo* contexto '`Voice`' se recolectan y se representan en el orden musical:
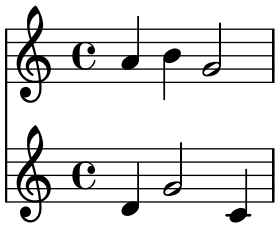
```
<a c>4-.  <>-. << c a >>  << { c-. <c a> } { a s-. } >>
```



Tratar de poner más de una plica o barra de corchea, o distintas duraciones o propiedades en el mismo momento musical, requiere el uso de más de una voz.

El ejemplo siguiente muestra cómo las expresiones simultáneas pueden generar varios pentagramas de forma implícita:

```
% no single first note
<< { a4 b g2 } { d4 g2 c,4 } >>
```



Aquí no hay problema en tener distintas duraciones porque se interpretan en voces distintas.

Advertencias y problemas conocidos

Si hay notas de dos o más voces en la misma posición del pentagrama, con las plicas en la misma dirección, y no tienen desplazamiento (o tienen especificado el mismo desplazamiento), aparecerá el mensaje

```
advertencia: demasiadas columnas de notas que chocan entre sí. Se ignoran
```

durante la compilación. Este mensaje se puede suprimir mediante

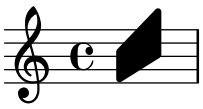
```
\override NoteColumn.ignore-collision = ##t
```

Sin embargo, esto no solamente suprime la advertencia sino que puede evitar cualquier resolución de colisiones y puede tener otros efectos no deseados (consulte también *Problemas conocidos* en [Resolución de las colisiones], página 170).

Racimos (clusters)

Un «cluster» o racimo indica que se deben tocar simultáneamente un conjunto de notas consecutivas. Se escriben aplicando la función `\makeClusters` a una secuencia de acordes, p.ej.:

```
\makeClusters { <g b>2 <c g'> }
```



Se pueden mezclar en el mismo pentagrama notas normales y clusters, incluso al mismo tiempo. En tal caso, no se hace ningún intento de evitar automáticamente las colisiones entre clusters y notas normales.

Véase también

Glosario musical: Sección "cluster" in *Glosario Musical*.

Fragmentos de código: Sección "Simultaneous notes" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "ClusterSpanner" in *Referencia de Funcionamiento Interno*, Sección "ClusterSpannerBeacon" in *Referencia de Funcionamiento Interno*, Sección "Cluster_spanner_engraver" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los clusters sólo tienen un buen aspecto cuando abarcan un mínimo de dos acordes. En caso contrario aparecerán excesivamente estrechos.

Los clusters no llevan plica y por sí mismos no pueden indicar las duraciones, pero la longitud del cluster que se imprime viene determinada por la duración de los acordes que lo definen. Los racimos separados necesitan silencios de separación entre ellos.

Los clusters no producen ninguna salida MIDI.

1.5.2 Varias voces
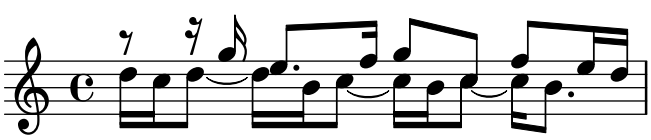
Esta sección trata las notas simultáneas en varias voces o varios pentagramas.

Polifonía en un solo pentagrama

Instanciar las voces explícitamente

La estructura básica necesaria para obtener varias voces independientes en un mismo pentagrama se ilustra en el ejemplo siguiente:

```
\new Staff <<
  \new Voice = "first"
    { \voiceOne r8 r16 g e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    { \voiceTwo d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Aquí se crean explícitamente instancias de voces, cada una de las cuales recibe un nombre. Las instrucciones `\voiceOne` ... `\voiceFour` (voz uno hasta voz cuatro) preparan las voces de

manera que la primera y segunda voces llevan las plicas hacia arriba, las voces segunda y cuarta llevan las plicas hacia abajo, las cabezas de las notas en las voces tercera y cuarta se desplazan horizontalmente, y los silencios de las voces respectivas se desplazan también automáticamente para evitar las colisiones. La instrucción `\oneVoice` (una voz) devuelve todos los ajustes de las voces al estado neutro predeterminado.
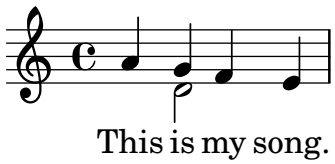
Pasajes polifónicos temporales

Se puede crear un pasaje polifónico temporal con la construcción siguiente:

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

Aquí, la primera expresión dentro de un pasaje polifónico temporal se coloca en el contexto `Voice` que estaba en uso inmediatamente antes del pasaje polifónico, y ese mismo contexto `Voice` continua después de la sección temporal. Otras expresiones dentro de los ángulos se asignan a distintas voces temporales. Esto permite asignar la letra de forma continua a una voz antes, durante y después de la sección polifónica:

```
<<
  \new Voice = "melody" {
    a4
    <<
      {
        \voiceOne
        g f
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melody" {
  This is my song.
  }
>>
```
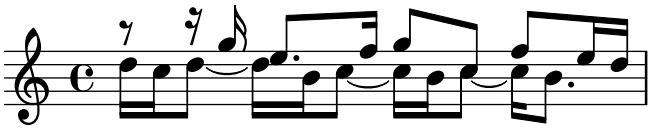


Aquí, las instrucciones `\voiceOne` y `\voiceTwo` son necesarias para definir los ajustes de cada voz.

La construcción de la doble barra invertida

La construcción `<< {...} \\ {...} >>`, en que las dos expresiones (o más) están separadas por doble barra invertida, se comporta de forma distinta a la construcción similar sin las dobles barras: *todas* las expresiones dentro de esta construcción se asignan a contextos `Voice` nuevos. Estos contextos `Voice` nuevos se crean implícitamente y reciben los nombres fijos `"1"`, `"2"`, etc.

El primer ejemplo podría haberse tipografiado de la manera siguiente:
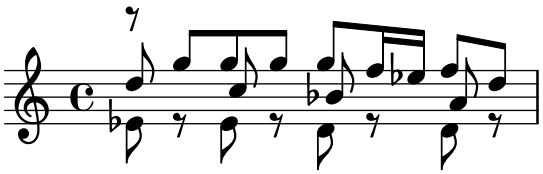
```
<<
  { r8 r16 g e8. f16 g8[ c,] f e16 d }
  \\
  { d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Esta sintaxis se puede usar siempre que no nos importe que se creen nuevas voces temporales que después serán descartadas. Estas voces creadas implícitamente reciben ajustes equivalentes al efecto de las instrucciones `\voiceOne` ... `\voiceFour`, en el orden en que aparecen en el código.

En el siguiente ejemplo, la voz intermedia lleva las plicas hacia arriba, de manera que la introducimos en tercer lugar para que pase a ser la voz tres, que tiene las plicas hacia arriba tal y como queremos. Se usan silencios de separación para evitar la aparición de silencios duplicados.

```
<<
  { r8 g g  g g f16 ees f8 d }
  \\
  { ees,8 r ees r d r d r }
  \\
  { d'8 s c s bes s a s }
>>
```



En todas las partituras excepto las más simples, se recomienda crear contextos `Voice` explícitos como aparece explicado en Sección “Contextos y grabadores” in *Manual de Aprendizaje* y en Sección “Voces explícitas” in *Manual de Aprendizaje*.

Orden de las voces
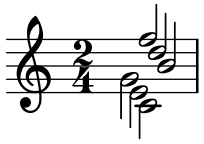
Al escribir varias voces en el archivo de entrada, utilice el orden siguiente:

```
Voz 1: las más aguda
Voz 2: la más grave
Voz 3: la segunda más aguda
Voz 4: la segunda más grave
Voz 5: la tercera más aguda
Voz 6: la tercera más grave
etc.
```

A pesar de que esto puede parecer contrario a la intuición, simplifica el proceso de disposición automática de las figuras. Observe que las voces de numeración impar reciben plicas hacia arriba, y las de numeración par reciben plicas hacia abajo:

```
\new Staff <<
  \time 2/4
  { f''2 }  % 1: highest
  \\
```

```
  { c'2  }  % 2: lowest
  \\
  { d''2 }  % 3: second-highest
  \\
  { e'2  }  % 4: second-lowest
  \\
  { b'2  }  % 5: third-highest
  \\
  { g'2  }  % 6: third-lowest
>>
```
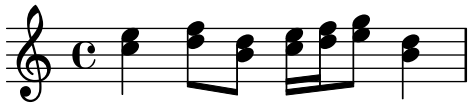


Nota: No se pueden crear letras ni elementos de extensión (como ligaduras, reguladores, etc.) que se crucen de una voz a otra.

Duraciones idénticas

En el caso especial en que queremos tipografiar fragmentos de música que discurre en paralelo y con las mismas duraciones, se pueden combinar en un solo contexto de voz, formando así acordes. Para conseguirlo, las incorporamos dentro de una construcción de música simultánea, dentro de una voz creada explícitamente:

```
\new Voice <<
  { e4 f8 d e16 f g8 d4 }
  { c4 d8 b c16 d e8 b4 }
>>
```



Este método conduce a barrados extraños y advertencias si los fragmentos de música no tienen las mismas duraciones exactas.

Instrucciones predefinidas

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

Véase también

Manual de aprendizaje: Sección “Las voces contienen música” in *Manual de Aprendizaje*, Sección “Voces explícitas” in *Manual de Aprendizaje*.

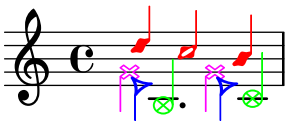
Referencia de la notación: [Pautas de percusión], página 378, [Silencios invisibles], página 56, [Plicas], página 219.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Estilos de voz

Se pueden aplicar colores y formas distintos a las voces para permitir identificarlas fácilmente:

```
<<
  { \voiceOneStyle d4 c2 b4 }
  \\
  { \voiceTwoStyle e,2 e }
  \\
  { \voiceThreeStyle b2. c4 }
  \\
  { \voiceFourStyle g'2 g }
>>
```



Para recuperar la presentación normal se utiliza la instrucción `\voiceNeutralStyle`.

Instrucciones predefinidas

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

Véase también
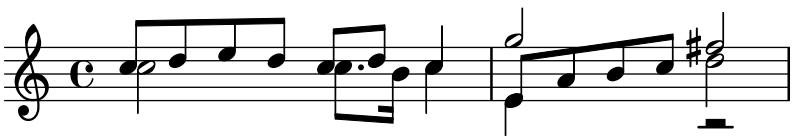
Manual de aprendizaje: Sección “Oigo voces” in *Manual de Aprendizaje*, Sección “Otras fuentes de información” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Resolución de las colisiones
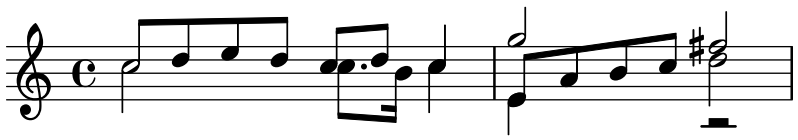
Las cabezas de notas que están en diferentes voces y tienen la misma altura, la misma forma de cabeza, y dirección opuesta de la plica, se combinan automáticamente, pero las que tienen cabezas distintas o la misma dirección de la plica no se combinan. Los silencios que se encuentran en el lado opuesto de una plica en otra voz se desplazan verticalmente. El ejemplo siguiente muestra tres circunstancias distintas, sobre los pulsos 1 y 3 en el primer compás y sobre el pulso 1 del segundo compás, donde la combinación automática falla.

```
<<
  {
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Las cabezas de notas diferentes se pueden combinar como se muestra más abajo. En este ejemplo las cabezas del pulso 1 del primer compás sí se combinan:
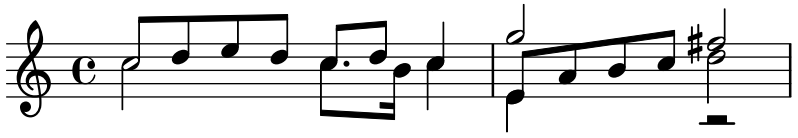
```
<<
  {
    \mergeDifferentlyHeadedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Las figuras negras y blancas no se combinan de esta manera, pues sería difícil poder distinguirlas.

También se pueden combinar cabezas con puntillos diferentes, como se muestra en el tercer pulso del primer compás:
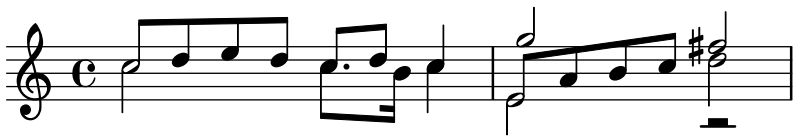
```
<<
  {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



La blanca y la corchea en el comienzo del segundo compás no están correctamente combinadas porque la combinación automática no es capaz de completarse satisfactoriamente cuando se encuentran tres o más notas alineadas en la misma columna de notas, y en este caso la cabeza combinada es incorrecta. Para conseguir que la combinación seleccione la cabeza correcta, se debe aplicar un `\shift` o desplazamiento a la nota que no se debe combinar. Aquí, se aplica

\shiftOn para mover el sol agudo fuera de la columna, y entonces \mergeDifferentlyHeadedOn hace su trabajo correctamente.

```
<<
  {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    \shiftOn
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



La instrucción \shiftOn permite (aunque no fuerza) que las notas de una voz se puedan desplazar. Cuando\shiftOn se aplica a una voz, una nota o acorde de esta voz se desplaza solamente si su plica chocase en caso contrario con una plica de otra voz, y sólo si las plicas en colisión apuntan en la misma dirección. La instrucción \shiftOff evita la posibilidad de este tipo de desplazamiento.
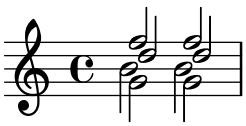
De forma predeterminada, las voces externas (normalmente las voces uno y dos) tienen \shiftOff especificado (desplazamiento desactivado), mientras que las voces interiores (tres y siguientes) tienen especificado \shiftOn (desplazamiento activado). Cuando se aplica un desplazamiento, las voces con plicas hacia arriba (voces de numeración impar) se desplazan hacia la derecha, y las voces con las plicas hacia abajo (voces con numeración par) se desplazarn a la izquierda.

He aquí un ejemplo que le ayudará a visualizar la forma en que se expanidría internamente una expresión polifónica.

> **Nota:** Observe que con tres o más voces, el orden vertical de las mismas dentro de su archivo de entrada ¡no sería el mismo que el orden vertical de las voces en el pentagrama!

```
\new Staff \relative c'' {
  %% abbreviated entry
  <<
    { f2  }  % 1: highest
    \\
    { g,2 }  % 2: lowest
    \\
    { d'2 }  % 3: upper middle
    \\
```

```
    { b2  }  % 4: lower middle
  >>
  %% internal expansion of the above
  <<
    \new Voice = "1" { \voiceOne   \shiftOff f'2 }
    \new Voice = "2" { \voiceTwo   \shiftOff g,2 }
    \new Voice = "3" { \voiceThree \shiftOn  d'2 } % shifts right
    \new Voice = "4" { \voiceFour  \shiftOn  b2  } % shifts left
  >>
}
```



Dos instrucciones adicionales, `\shiftOnn` y `\shiftOnnn`, ofrecen niveles de desplazamiento mayores que se pueden especificar temporalmente para resolver colisiones en situaciones complejas; véase Sección "Ejemplos reales de música" in *Manual de Aprendizaje*.

Sólo se combinan las notas si tienen la plica en direcciones opuestas (como la tienen, por ejemplo, en las voces uno y dos de forma predeterminada o cuando las plicas se establecen explícitamente en direcciones opuestas).

Instrucciones predefinidas

`\mergeDifferentlyDottedOn`, `\mergeDifferentlyDottedOff`, `\mergeDifferentlyHeadedOn`, `\mergeDifferentlyHeadedOff`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`.

Fragmentos de código seleccionados
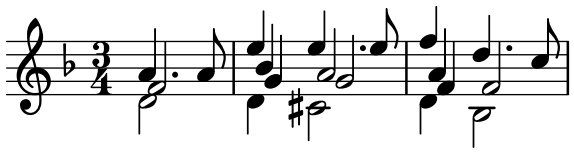
Voces adicionales para evitar colisiones

En ciertos casos de polifonía compleja, se necesitan voces adicionales para evitar colisiones entre las notas. Si se necesitan más de cuatro voces paralelas, las voces adicionales se añaden definiendo una variable que utiliza la función de Scheme `context-spec-music`.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice  {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
```

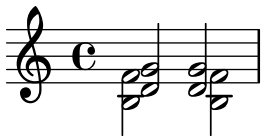
```
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
}
```



Forzar el desplazamiento horizontal de las notas

Cuando el motor de tipografiado no es capaz de todo, se puede usar la sintaxis siguiente para sobreescribir las decisiones de tipografía. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = #1.7
    <b f'>2
  }
>>
```



Véase también

Glosario musical: Sección “polyphony” in *Glosario Musical*.

Manual de aprendizaje: Sección “Varias notas a la vez” in *Manual de Aprendizaje*, Sección “Las voces contienen música” in *Manual de Aprendizaje*, Sección “Ejemplos reales de música” in *Manual de Aprendizaje*.

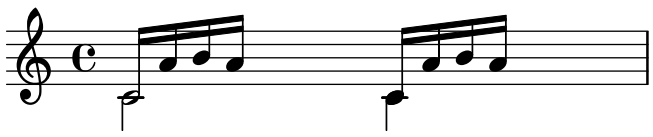
Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “NoteColumn” in *Referencia de Funcionamiento Interno*, Sección “NoteCollision” in *Referencia de Funcionamiento Interno*, Sección “RestCollision” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El uso de `\override NoteColumn.ignore-collision = ##t` hace que las notas con distinta cabeza en voces diferentes se mezclen incorrectamente.

```
\mergeDifferentlyHeadedOn
<< { c16 a' b a } \\ { c,2 } >>
\override NoteColumn.ignore-collision = ##t
<< { c16 a' b a } \\ { c,2 } >>
```



Combinación automática de las partes

La combinación automática de particellas se usa para mezclar dos partes musicales distintas sobre un pentagrama. Esto puede ser de gran ayuda especialmente al tipografiar partituras orquestales. Se imprime una sola voz cuando la música de las dos voces es la misma, pero en aquellos lugares en que difieren, se imprime una segunda voz. Las direcciones de las plicas se establecen hacia arriba o hacia abajo según proceda, al tiempo que se identifican y quedan marcadas las partes de solista y de dúo.

La sintaxis para la combinación automática de las partes es:
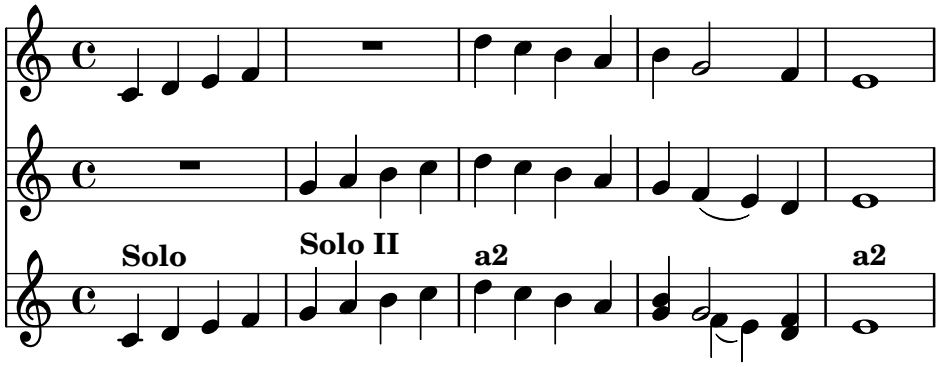
`\partcombine` *expresión_musical_1 expresión_musical_2*

El ejemplo siguiente ejemplifica la funcionalidad básica, poniendo las partes en un solo pentagrama en forma polifónica, y estableciendo las direcciones de las plicas de forma adecuada. Se utilizan las mismas variables para las partes independientes y el pentagrama combinado.

```
instrumentOne = \relative c' {
  c4 d e f |
  R1 |
  d'4 c b a |
  b4 g2 f4 |
  e1 |
}

instrumentTwo = \relative g' {
  R1 |
  g4 a b c |
  d4 c b a |
  g4 f( e) d |
  e1 |
}

<<
  \new Staff \instrumentOne
  \new Staff \instrumentTwo
  \new Staff \partcombine \instrumentOne \instrumentTwo
>>
```



Las dos partes tienen notas idénticas en el tercer compás, por lo que aparecen solamente una vez. Las direcciones de las plicas y ligaduras se establecen de forma automática, según se trate de un solo o de un unísono. Cuando se necesita en situaciones de polifonía, la primera parte (que recibe el nombre de contexto `one`) recibe las plicas hacia arriba, mientras que la segunda (llamada `two`) siempre recibe las plicas hacia abajo. En los fragmentos de solo, las partes se marcan con "Solo" y "Solo II", respectivamente. Las partes en unísono (*a due*) se marcan con el texto "a2".

Los dos argumentos de `\partcombine` se interpretan como contextos de Sección "Voice" in *Referencia de Funcionamiento Interno* separados, por lo que si la música está escrita en modo relativo, entonces *las dos* partes han de incluir la función `\relative`, es decir:

```
\partcombine
  \relative ... expresión_musical_1
  \relative ... expresión_musical_2
```

Una sección `\relative` que se encuentra fuera de `\partcombine` no tiene ningún efecto sobre las notas de *`expresión_musical_1`* y *`expresión_musical_2`*.

En las partituras profesionales, las voces con frecuencia se mantienen separadas entre sí durante pasajes prolongados incluso si algunas notas coninciden entre ambas, y podrían escribirse fácilmente como un unísono. Por ello, la combinación de las notas en un acorde, o mostrar una voz como solo, no son soluciones ideales porque la función `\partcombine` considera cada nota de forma individual. En este caso, el resultado de la función `\partcombine` se puede alterar o corregir con las instrucciones siguientes:

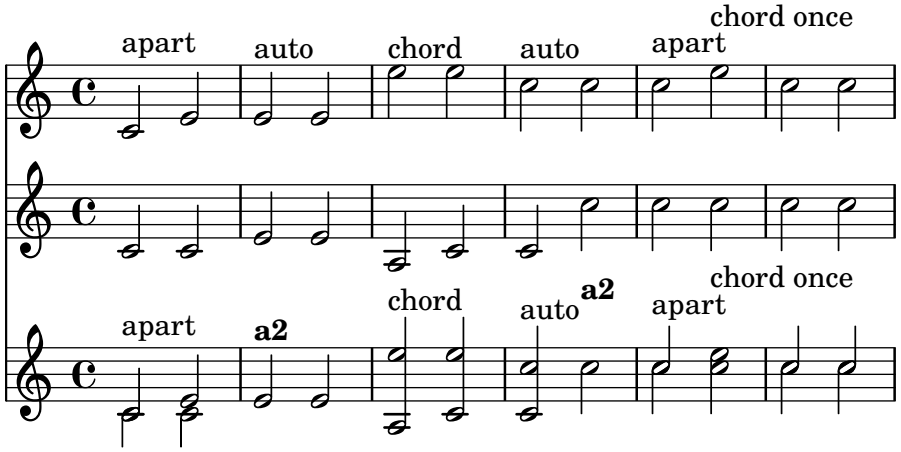
Las instrucciones que finalizan en `...Once` se aplican exclusivamente a la nota siguiente dentro de la expresión musical.

- `\partcombineApart` y `\partcombineApartOnce` mantienen las notas como dos voces separadas incluso si se pueden combinar en un acorde o unísono.
- `\partcombineChords` y `\partcombineChordsOnce` combinan las notas en un acorde.
- `\partcombineUnisono` y `\partcombineUnisonoOnce` combinan las dos voces como unísono y marcan el resultado como "unison".
- `\partcombineSoloI` y `\partcombineSoloIOnce` muestran solo la voz uno y la marcan como "Solo".
- `\partcombineSoloII` o `\partcombineSoloIIOnce` imprimen solo la voz dos y la marcan como "Solo".
- `\partcombineAutomatic` y `\partcombineAutomaticOnce` terminan el efecto de las instrucciones anteriores y retornan a la funcionalidad estándar de `\partcombine`.

```
instrumentOne = \relative c' {
  \partcombineApart c2^"apart" e |
  \partcombineAutomatic e2^"auto" e |
  \partcombineChords e'2^"chord" e |
  \partcombineAutomatic c2^"auto" c |
  \partcombineApart c2^"apart" \partcombineChordsOnce e^"chord once" |
  c2 c |
```

```
}
instrumentTwo = \relative c' {
  c2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}

<<
  \new Staff { \instrumentOne }
  \new Staff { \instrumentTwo }
  \new Staff { \partcombine \instrumentOne \instrumentTwo }
>>
```



Uso de \partcombine con letra

La instrucción `\partcombine` no está diseñada para funcionar con la letra de las canciones; si una de las voces recibe un nombre explícito para poder adjuntar una letra, el combinador de partes no funciona. Sin embargo, se puede conseguir este efecto usando un contexto `NullVoice`. Véase [Polifonía con letras compartidas], página 280.

Fragmentos de código seleccionados

Combinar dos partes sobre el mismo pentagrama

La herramienta de combinación de partes (instrucción `\partcombine`) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como "solo" o "a2" se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad `printPartCombineTexts` al valor "falso". Para partituras vocales (como himnos), no hay necesidad de añadir los textos "solo" o "a2", por lo que se deben desactivar. Sin embargo, podría ser mejor no usarlo si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.

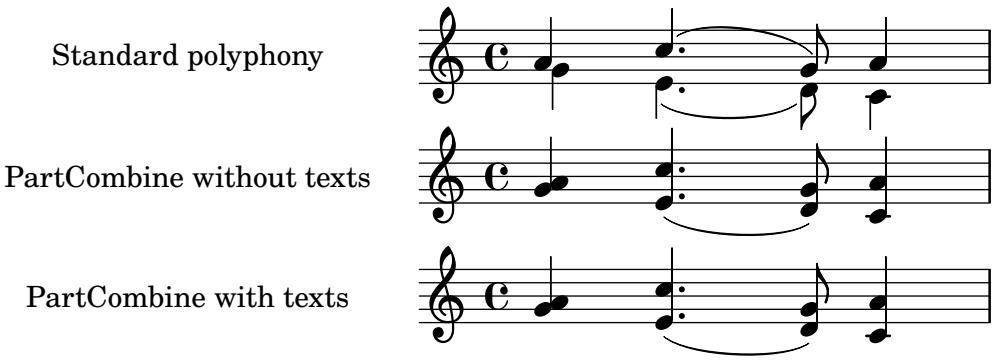
Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, `\partcombine` sin textos, y `\partcombine` con textos.
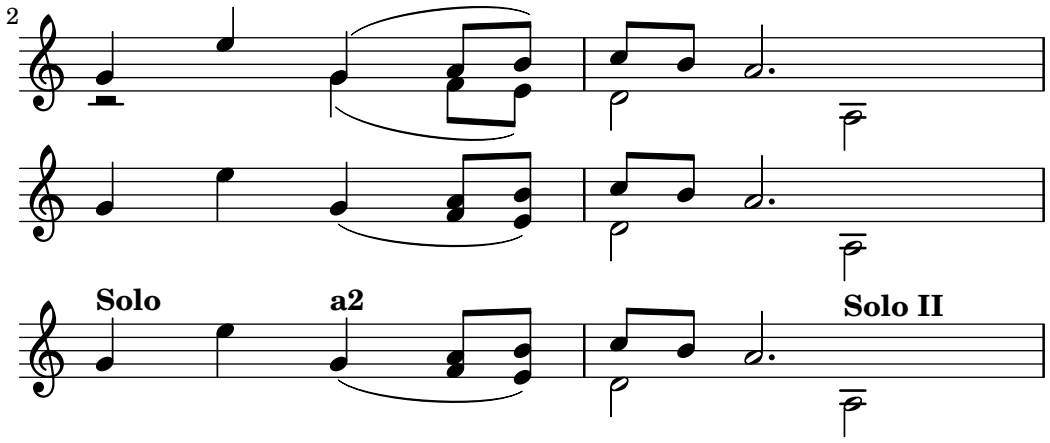
```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
```

```
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    <<
    \new Staff {
      \set Staff.instrumentName = #"Standard polyphony"
      << \musicUp \\ \musicDown >>
    }
    \new Staff \with { printPartCombineTexts = ##f } {
      \set Staff.instrumentName = #"PartCombine without texts"
      \partcombine \musicUp \musicDown
    }
    \new Staff {
      \set Staff.instrumentName = #"PartCombine with texts"
      \partcombine \musicUp \musicDown
    }
    >>
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar.collapse-height = #30
    }
  }
}
```
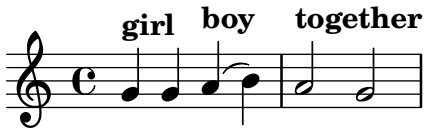




Cambiar los textos de partcombine

Al utilizar la posibilidad de combinación automática de partes, se puede modificar el texto que se imprime para las secciones de solo y de unísono:

```
\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partcombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
>>
```



Véase también

Glosario musical: Sección “a due” in *Glosario Musical*, Sección “part” in *Glosario Musical*.

Referencia de la notación: Sección 1.6.3 [Escritura de las particellas], página 201.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PartCombineMusic” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Todas las funciones `\partcombine...` admiten exclusivamente dos voces.

Las funciones `\partcombine...` no se pueden escribir dentro de los bloques `\tuplet` ni `\relative`.

Si `printPartCombineTexts` está establecido y las dos voces tocan y terminan las mismas notas en el mismo compás, el combinador de partes puede tipografiar `a2` más de una vez en ese compás.

`\partcombine` solo tiene en cuenta el momento de inicio de las notas dentro de una voz `Voice`; por ejemplo, no puede recordar si una nota dentro de una voz ya ha comenzado cuando

las notas que se combinan se han iniciado justo ahora en la otra voz. Esto puede conducir a cierto número de problemas inesperados entre los que se incluye la impresión incorrecta de marcas de "Solo" o de "Unison".

`\partcombine` conserva todos los elementos extensos (ligaduras, reguladores, etc.) dentro del mismo contexto `Voice` de forma que si uno cualquiera de estos elementos extensos inicia o termina en un contexto `Voice` diferente, puede no imprimirse o hacerlo incorrectamente.

Si la función `\partcombine` no puede combinar las dos expresiones musicales (es decir, cuando las dos voces tienen duraciones distintas), otorgará internamente sus propios nombres a las voces: `one` y `two` respectivamente. Esto significa que si se produce un cambio a un contexto `Voice` que tenga un nombre distinto, se ignorarán los eventos dentro del contexto `Voice` que tiene el nombre distinto.

Consulte también el apartado *Advertencias y problemas conocidos* en la sección [Tablaturas predeterminadas], página 332 al utilizar `\partcombine` con tablaturas, y la *Nota* de [Barras automáticas], página 79 al utilizar barrado automático.

Escribir música en paralelo
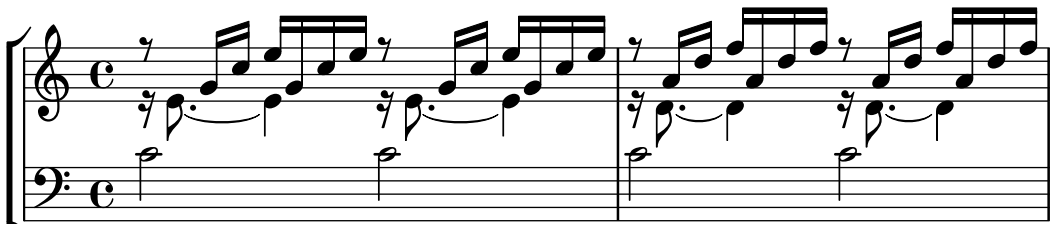
La música para varias partes se puede intercalar dentro del código de entrada. La función `\parallelMusic` admite una lista que contiene los nombres de las variables que se van a crear, y una expresión musical. El contenido de los compases alternativos extraídos de la expresión se convierten en el valor de las variables respectivas, de manera que podemos utilizarlas más tarde para imprimir la música.

> **Nota:** Es obligatorio utilizar comprobaciones de compás `|`, y los compases deben tener la misma longitud.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~   e'4           r16 e'8.~   e'4           |
  c'2                       c'2                       |

  % Bar 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~   d'4           r16 d'8.~   d'4           |
  c'2                       c'2                       |

}
\new StaffGroup <<
  \new Staff << \voiceA \\ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```



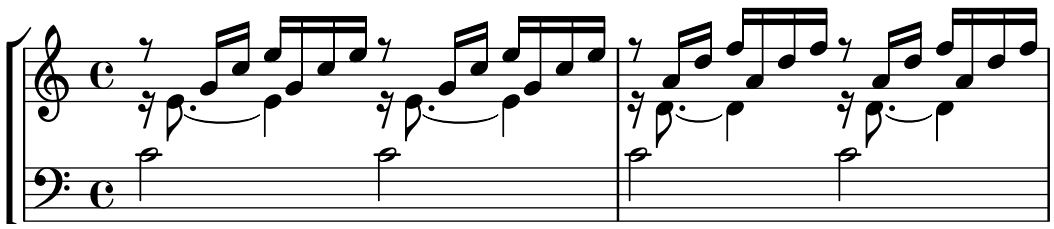
Se puede usar el modo relativo. Observe que la instrucción `\relative` no se utiliza dentro del propio bloque `\parallelMusic`. Las notas guardan relación con la nota anterior en la misma voz,

no con la nota anterior dentro del código de entrada (dicho de otra manera, las notas relativas de `vozA` ignoran a las notas que hay en `vozB`.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g16 c e g, c e r8 g,16 c e g, c e  |
  r16 e8.~ e4       r16 e8.~  e4        |
  c2                c                   |

  % Bar 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~  d4      r16 d8.~  d4       |
  c2                c                  |

 }
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>
```



Esto funciona aceptablemente bien para música de piano. El siguiente ejemplo asigna cada cuatro compases consecutivos a cuatro variables:

```
global = {
  \key g \major
  \time 2/4
}
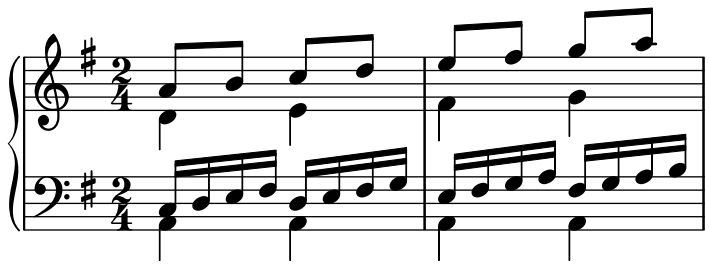
\parallelMusic #'(voiceA voiceB voiceC voiceD) {
  % Bar 1
  a8    b     c   d     |
  d4          e         |
  c16 d e fis d e fis g |
  a4          a         |

  % Bar 2
  e8      fis  g     a   |
  fis4         g         |
  e16 fis g  a fis g a b |
  a4           a         |

  % Bar 3 ...
}

\score {
  \new PianoStaff <<
```

```
      \new Staff {
        \global
        <<
          \relative c'' \voiceA
          \\
          \relative c'  \voiceB
        >>
      }
      \new Staff {
        \global \clef bass
        <<
          \relative c \voiceC
          \\
          \relative c \voiceD
        >>
      }
    >>
}
```

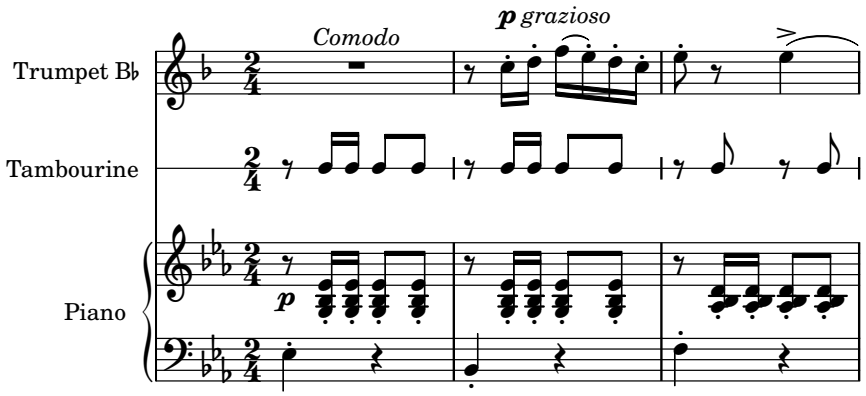


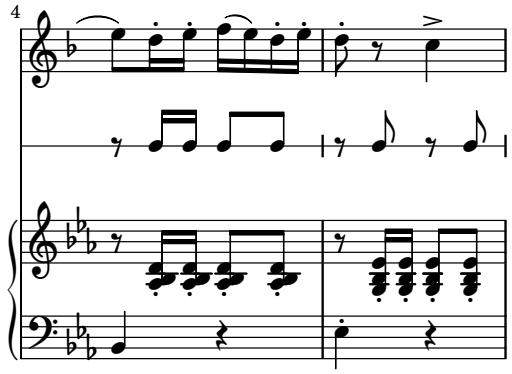
Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” in *Manual de Aprendizaje*

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

1.6 Notación de los pentagramas





Esta sección explica cómo influir sobre la apariencia de los pentagramas, cómo imprimir partituras con más de un pentagrama y cómo añadir indicaciones de tempo y notas guía a los pentagramas.

1.6.1 Impresión de los pentagramas
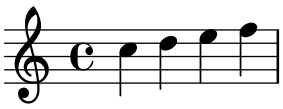
Esta sección describe los distintos métodos de creación de pentagramas y grupos de ellos.

Crear instancias de pentagramas nuevos

Las *pautas* y los *pentagramas* o pautas de cinco líneas se crean con las instrucciones `\new` o `\context`. Para ver más detalles, consulte Sección 5.1.2 [Crear y referenciar contextos], página 569.

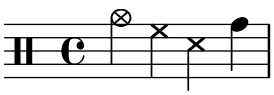
El contexto básico de pentagrama es `Staff`:

```
\new Staff { c4 d e f }
```



El contexto `DrumStaff` crea una pauta de cinco líneas preparada para un conjunto de batería típico. Cada instrumento se presenta con un símbolo distinto. Los instrumentos se escriben en el modo de percusión que sigue a una instrucción `\drummode`, con cada instrumento identificado por un nombre. Para ver más detalles, consulte [Pautas de percusión], página 378.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



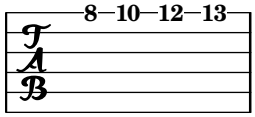
`RhythmicStaff` crea una pauta de una sola línea que sólo muestra las duraciones de la entrada. Se preservan las duraciones reales. Para ver más detalles, consulte [Mostrar los ritmos de la melodía], página 76.

```
\new RhythmicStaff { c4 d e f }
```



`TabStaff` crea una tablatura con seis cuerdas en la afinación estándar de guitarra. Para ver más detalles, consulte [Tablaturas predeterminadas], página 332.
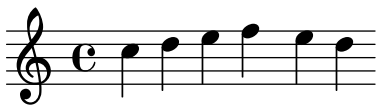
```
\new TabStaff { c4 d e f }
```



Existen dos contextos de pauta específicos para la notación de música antigua: `MensuralStaff` y `VaticanaStaff`. Se describen en [Contextos predefinidos], página 423.

`GregorianTranscriptionStaff` crea una pauta para la notación moderna de canto gregoriano. No muestra líneas divisorias.

```
\new GregorianTranscriptionStaff { c4 d e f e d }
```



Se pueden definir contextos nuevos de pentagrama único. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 582.

Véase también

Glosario musical: Sección “staff” in *Glosario Musical*, Sección “staves” in *Glosario Musical*.

Referencia de la notación: Sección 5.1.2 [Crear y referenciar contextos], página 569, [Pautas de percusión], página 378, [Mostrar los ritmos de la melodía], página 76, [Tablaturas predeterminadas], página 332, [Contextos predefinidos], página 423, [El símbolo del pentagrama], página 191, [Contextos del canto gregoriano], página 433, [Contextos de la música mensural], página 424, Sección 5.1.6 [Definir contextos nuevos], página 582.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

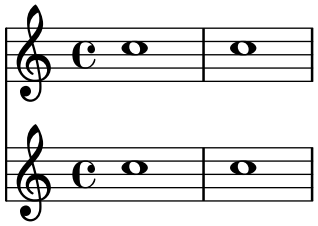
Referencia de funcionamiento interno: Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “DrumStaff” in *Referencia de Funcionamiento Interno*, Sección “GregorianTranscriptionStaff” in *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*, Sección “TabStaff” in *Referencia de Funcionamiento Interno*, Sección “MensuralStaff” in *Referencia de Funcionamiento Interno*, Sección “VaticanaStaff” in *Referencia de Funcionamiento Interno*, Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*.

Agrupar pentagramas

Existen varios contextos para agrupar pentagramas individuales formando sistemas. Cada contexto de agrupación establece el estilo del delimitador de comienzo del sistema y el comportamiento de las barras de compás.

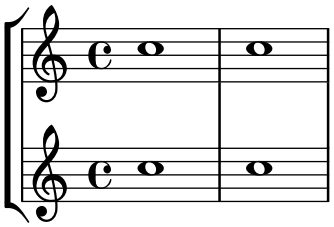
Si no se especifica ningún contexto, se usan las propiedades predeterminadas: el grupo comienza con una línea vertical y las barras de compás no están conectadas.

```
<<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



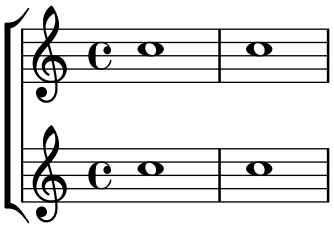
En el contexto `StaffGroup`, el grupo se inicia con un corchete y las barras de compás se dibujan atravesando todos los pentagramas.

```
\new StaffGroup <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



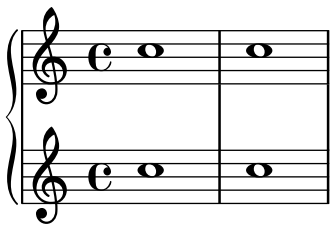
En un `ChoirStaff` (sistema de coro), el grupo se inicia con un corchete, pero las barras de compás no están conectadas.

```
\new ChoirStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



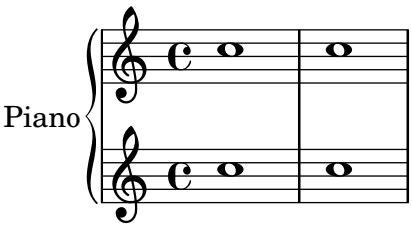
En un `GrandStaff` (sistema de piano), el grupo se inicia con una llave y las barras de compás se conectan entre los pentagramas.

```
\new GrandStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



El `PianoStaff` (sistema de piano) es idéntico a `GrandStaff`, excepto que contempla directamente la impresión del nombre del instrumento. Para ver más detalles, consulte [Nombres de instrumentos], página 201.

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = #"Piano"
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Cada contexto de grupo de pentagramas fija la propiedad del delimitador de inicio `systemStartDelimiter` a uno de los siguientes valores: `SystemStartBar` (línea), `SystemStartBrace` (llave) o `SystemStartBracket` (corchete). También está disponible un cuarto delimitador, `SystemStartSquare` (corchete en ángulo recto), pero se debe especificar explícitamente.

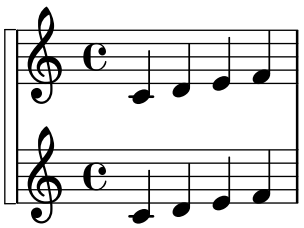
Se pueden definir contextos nuevos de grupo de pentagramas. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 582.

Fragmentos de código seleccionados

Uso del corchete recto al comienzo de un grupo de pentagramas

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



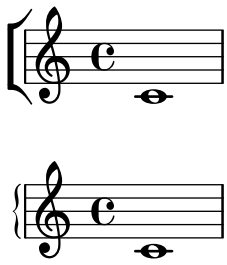
Mostrar corchete o llave en grupos de un solo pentagrama

Si hay un solo pentagrama en un de los tipos de sistema `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar sobreescribiendo `collapse-height` para fijar su valor de manera que sea menor que el número de líneas en la pauta.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\score {
  \new StaffGroup <<
```

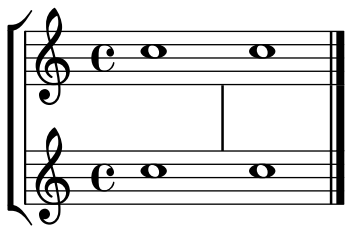
```
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
```



Disposición Mensurstriche (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra estableciendo la propiedad `transparent`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Véase también

Glosario musical: Sección “brace” in *Glosario Musical*, Sección “bracket” in *Glosario Musical*, Sección “grand staff” in *Glosario Musical*.

Referencia de la notación: [Nombres de instrumentos], página 201, Sección 5.1.6 [Definir contextos nuevos], página 582.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “StaffGroup” in *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” in *Referencia de Funcionamiento Interno*, Sección “GrandStaff” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” in *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” in *Referencia de Funcionamiento Interno*.
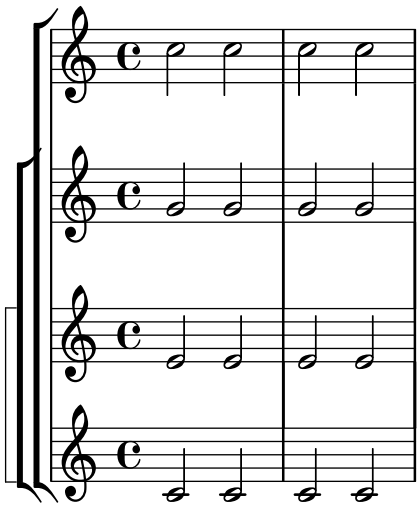
Advertencias y problemas conocidos

`PianoStaff` no acepta `ChordNames` de forma predeterminada.

Grupos de pentagramas anidados

Los contextos de grupos de pentagramas se pueden anidar hasta una profundidad arbitraria. En este caso, cada contexto descendiente crea un corchete nuevo adyacente al corchete de su grupo padre.

```
\new StaffGroup <<
  \new Staff { c2 c | c2 c }
  \new StaffGroup <<
    \new Staff { g2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
    <<
      \new Staff { e2 e | e2 e }
      \new Staff { c2 c | c2 c }
    >>
  >>
>>
```



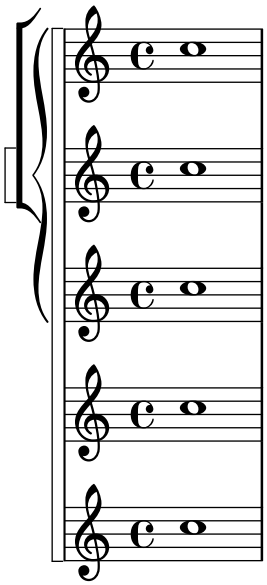
Se pueden definir nuevos contextos de grupos de pentagramas anidados. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 582.

Fragmentos de código seleccionados

Anidado de grupos de pentagramas

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La instrucción `\set StaffGroup.systemStartDelimiterHierarchy` toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` y `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                              (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Véase también

Referencia de la notación: [Agrupar pentagramas], página 184, [Nombres de instrumentos], página 201, Sección 5.1.6 [Definir contextos nuevos], página 582.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

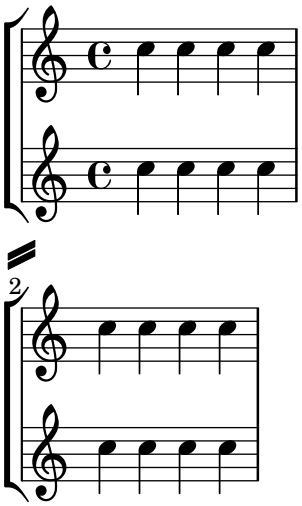
Referencia de funcionamiento interno: Sección “StaffGroup” in *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” in *Referencia de*

Funcionamiento Interno, Sección “SystemStartBracket” in *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” in *Referencia de Funcionamiento Interno*.

Separación de sistemas

Si el número de sistemas por página varía de una página a otra, es costumbre separar los sistemas colocando una marca separadora entre ellos. De forma predeterminada, el separador de sistemas es nulo, pero se puede activar con una opción de `\paper`.

```
\book {
  \score {
    \new StaffGroup <<
      \new Staff {
        \relative c'' {
          c4 c c c
          \break
          c4 c c c
        }
      }
      \new Staff {
        \relative c'' {
          c4 c c c
          \break
          c4 c c c
        }
      }
    >>
  }
  \paper {
    system-separator-markup = \slashSeparator
    % following commands are needed only to format this documentation
    paper-width = 100\mm
    paper-height = 100\mm
    tagline = ##f
  }
}
```



Véase también

Referencia de la notación: Sección 4.1 [Disposición de la página], página 513.

Fragmentos de código: Sección “Notación de los pentagramas” in *Fragmentos de código*.
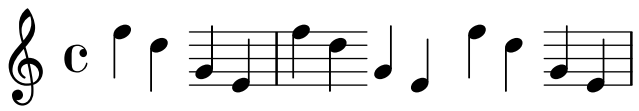
1.6.2 Modificación de pentagramas sueltos

Esta sección explica cómo cambiar los atributos específicos de un pentagrama: por ejemplo, cambiar el número de líneas o el tamaño de la pauta. También se describen los métodos para comenzar y terminar los pentagramas, y establecer secciones de ossia.

El símbolo del pentagrama

Se pueden usar las instrucciones `\stopStaff` y `\startStaff` para detener y (re)iniciar, respectivamente, la impresión de las líneas de la pauta en cualquier punto de la partitura.

```
\stopStaff f4 d \startStaff g, e
f'4 d \stopStaff g, e
f'4 d \startStaff g, e
```



Instrucciones predefinidas
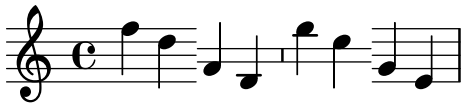
`\startStaff`, `\stopStaff`.

Las líneas de la pauta pertenecen al grob `StaffSymbol` (incluidas las líneas adicionales) y se pueden modificar usando las propiedades de `StaffSymbol`, pero esas modificaciones deben hacerse antes de que el pentagrama se (re)inicie.

Se puede cambiar el número de líneas de la pauta:
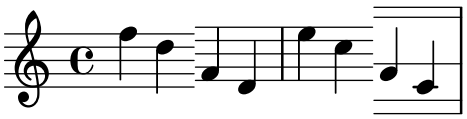
```
f4 d \stopStaff
\override Staff.StaffSymbol.line-count = #2
\startStaff g, e |

f'4 d \stopStaff
\revert Staff.StaffSymbol.line-count
\startStaff g, e |
```



La posición de cada una de las líneas de la pauta también puede cambiarse. Una lista de números fija la posición de cada línea. `0` corresponde a la línea central normal, y las posiciones normales de las líneas son `(-4 -2 0 2 4)`. Se imprime una sola línea de pauta por cada valor que se introduce, de manera que el número de líneas así como su posición en la pauta pueden cambiarse con una sola operación de sobreescritura.
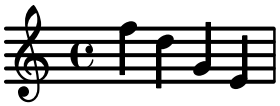
```
f4 d \stopStaff
\override Staff.StaffSymbol.line-positions = #'(1 3 5 -1 -3)
\startStaff g, e |
f'4 d \stopStaff
\override Staff.StaffSymbol.line-positions = #'(8 6.5 -6 -8 -0.5)
\startStaff g, e
```



Para preservar las direcciones típicas de las plicas (hacia arriba para la mitad inferior del pentagra y hacia abajo para la mitad superior), debemos alinear la línea (o espacio) central de la pauta personalizada con la posición de la línea central normal (0). Puede ser necesario un ajuste de la posición de la clave y del Do central para que se correspondan a las líneas nuevas. Véase [Clave], página 16.
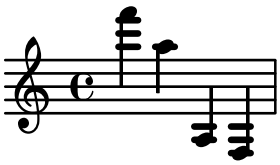
Puede cambiarse el grosor de las líneas de la pauta. También resultan afectadas, de forma predeterminada, las líneas adicionales y no la plica de las figuras.

```
\new Staff \with {
  \override StaffSymbol.thickness = #3
}
{ f4 d g, e }
```



También es posible fijar el grosor de las líneas adicionales de forma independiente del de las líneas de la pauta.

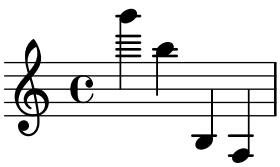
```
\new Staff \with {
  \override StaffSymbol.thickness = #2
  \override StaffSymbol.ledger-line-thickness = #'(0.5 . 0.4)
}
{ f'4 a, a,, f }
```



El primer valor se multiplica por el grosor de las líneas del pentagrama, el segundo por el ancho de un espacio del pentagrama, y después los dos valores se suman para obtener el grosor de las líneas adicionales.
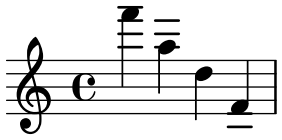
Se pueden alterar las posiciones de las líneas adicionales:

```
\new Staff \with {
  \override StaffSymbol.ledger-positions = #'(-3 -2 -1 2 5 6)
}
{ f'4 a, a,, f }
```
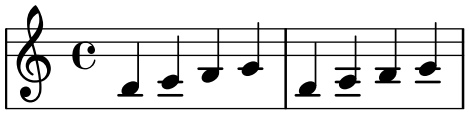


Puede hacerse que las líneas adicionales añadidas aparezcan por encima o por debajo de la cabeza de las figuras, dependiendo de la posición actual relativa a otras cabezas de figura que tienen asimismo sus propias líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol.ledger-extra = #4
}
{ f'4 a, d, f, }
```



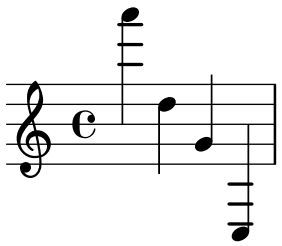
También puede hacerse que las líneas adicionales aparezcan dentro del pentagrama allí donde se requieren líneas de pauta personalizadas. El ejemplo muestra la posición predeterminada de las líneas adicionales cuando el valor de posición explícito `ledger-position` se ha fijado o no. La instrucción `\stopStaff` se necesita en el ejemplo para revertir la instrucción de sobreescritura `\override` para todo el `StaffSymbol` (el símbolo de pauta).

```
\override Staff.StaffSymbol.line-positions = #'(-8 0 2 4)
d4 e f g
\stopStaff
\startStaff
\override Staff.StaffSymbol.ledger-positions = #'(-8 -6 (-4 -2) 0)
d4 e f g
```



Puede alterarse la distancia entre líneas de la pauta. Esto afecta también al espaciado de las líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol.staff-space = #1.5
}
{ f'4 d, g, e, }
```
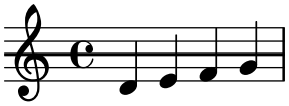


Fragmentos de código seleccionados

Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
```

```
}
```



Véase también

Glosario musical: Sección “line” in *Glosario Musical*, Sección “ledger line” in *Glosario Musical*, Sección “staff” in *Glosario Musical*.

Referencia de la notación: [Clave], página 16.
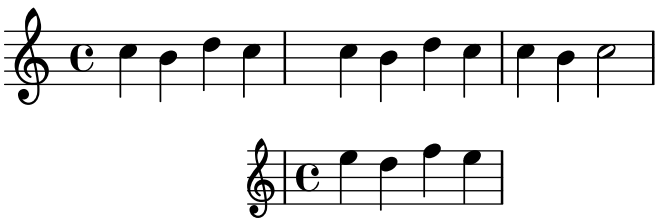
Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*, Sección “staff-symbol-interface” in *Referencia de Funcionamiento Interno*.

Pentagramas de Ossia

Los pentagramas *Ossia* se pueden preparar mediante la creación de un pentagrama simultáneo nuevo en la posición adecuada:

```
\new Staff \relative c'' {
  c4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```



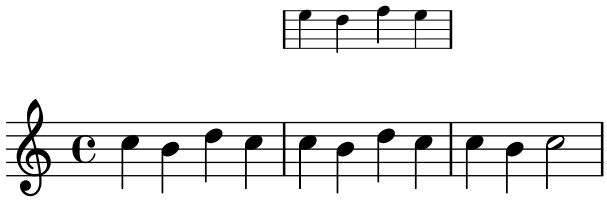
Sin embargo, el ejemplo anterior no es lo que normalmente se desea. Para crear pentagramas de ossia que estén encima del pentagrama original, que no tengan compás ni clave, y que tengan un tamaño menor de tipografía se deben usar varios trucos. El Manual de aprendizaje describe una técnica específica para llegar a este objetivo, empezando por Sección “Anidado de expresiones musicales” in *Manual de Aprendizaje*.

El ejemplo siguiente utiliza la propiedad `alignAboveContext` para alinear el pentagrama de ossia. Este método es muy conveniente cuando se necesitan sólo algunos pentagramas de ossia.

```
\new Staff = "main" \relative c'' {
  c4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove "Time_signature_engraver"
      alignAboveContext = #"main"
      fontSize = #-3
      \override StaffSymbol.staff-space = #(magstep -3)
```
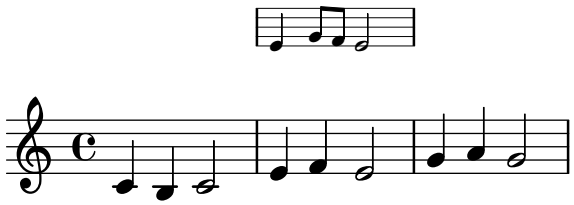
```
      \override StaffSymbol.thickness = #(magstep -3)
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
}
```
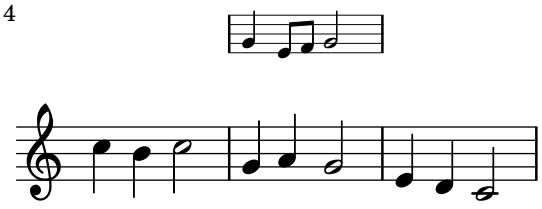


Si se requieren muchos pentagramas de ossia aislados, puede ser más conveniente la creación de un contexto `Staff` vacío con un *identificador de contexto* específico; después se pueden crear los pentagramas de ossia *llamando* a este contexto y usando `\startStaff` y `\stopStaff` en los puntos deseados. Las ventajas de este método son más patentes si la pieza es más larga que en ejemplo siguiente.

```
<<
  \new Staff = "ossia" \with {
    \remove "Time_signature_engraver"
    \hide Clef
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
    \override StaffSymbol.thickness = #(magstep -3)
  }
  { \stopStaff s1*6 }

  \new Staff \relative c' {
    c4 b c2
    <<
      { e4 f e2 }
      \context Staff = "ossia" {
        \startStaff e4 g8 f e2 \stopStaff
      }
    >>
    g4 a g2 \break
    c4 b c2
    <<
      { g4 a g2 }
      \context Staff = "ossia" {
        \startStaff g4 e8 f g2 \stopStaff
      }
    >>
    e4 d c2
  }
>>
```
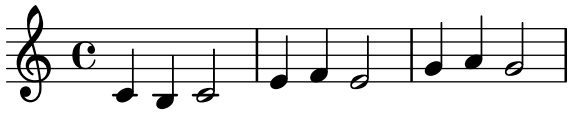


4
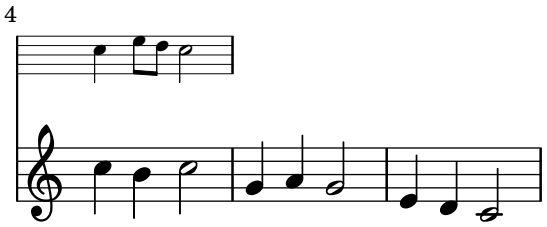


Como alternativa, se puede usar la instrucción `\RemoveEmptyStaves` para crear pentagramas de ossia. Este método es muy conveniente cuando los pentagramas de ossia aparecen inmediatamente después de un salto de línea. Para ver más información sobre `\RemoveEmptyStaves`, consulte [Ocultar pentagramas], página 198.

```
<<
  \new Staff = "ossia" \with {
    \remove "Time_signature_engraver"
    \hide Clef
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
    \override StaffSymbol.thickness = #(magstep -3)
  } \relative c'' {
    R1*3
    c4 e8 d c2
  }
  \new Staff \relative c' {
    c4 b c2
    e4 f e2
    g4 a g2 \break
    c4 b c2
    g4 a g2
    e4 d c2
  }
>>

\layout {
  \context {
    \Staff \RemoveEmptyStaves
    \override VerticalAxisGroup.remove-first = ##t
  }
}
```





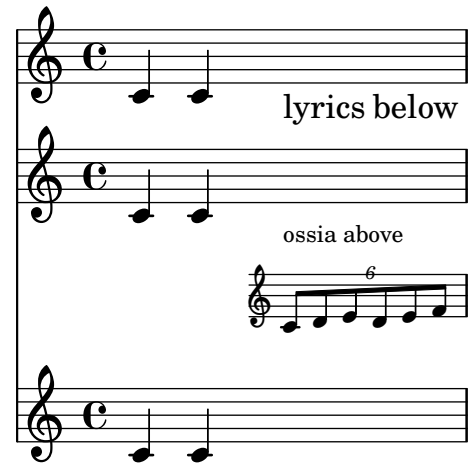
Fragmentos de código seleccionados

Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \tuplet 6/4 {
          \override TextScript.padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Véase también

Glosario musical: Sección “ossia” in *Glosario Musical*, Sección “staff” in *Glosario Musical*, Sección “Frenched staff” in *Glosario Musical*.

Manual de aprendizaje: Sección “Anidado de expresiones musicales” in *Manual de Aprendizaje*, Sección “Tamaño de los objetos” in *Manual de Aprendizaje*, Sección “Longitud y grosor de los objetos” in *Manual de Aprendizaje*.

Referencia de la notación: [Ocultar pentagramas], página 198.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*.

Ocultar pentagramas

Se pueden ocultar las líneas del pentagrama quitando el grabador `Staff_symbol_engraver` del contexto de `Staff`. Como alternativa se puede utilizar la instrucción `\stopStaff`.

```
\new Staff \with {
  \remove "Staff_symbol_engraver"
}
\relative c''' { a8 f e16 d c b a2 }
```

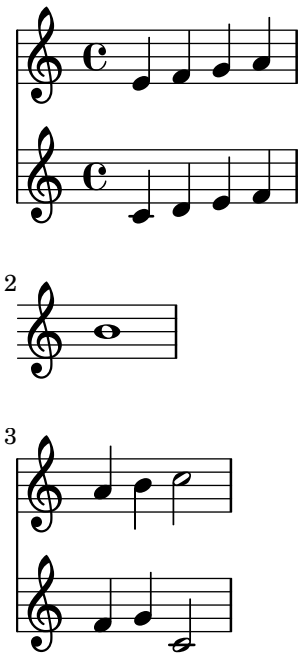


Se pueden ocultar los pentagramas que están vacíos mediante la instrucción `\RemoveEmptyStaves` dentro del bloque `\layout` para el contexto `\Staff`. En las partituras orquestales, este estilo se conoce como ‘partitura a la francesa’. De forma predeterminada, esta instrucción oculta y elimina todos los pentagramas vacíos de la partitura excepto los del primer sistema.

> **Nota:** Un pentagrama se considera vacío cuando contiene solamente silencios, silencios multicompás, desplazamientos, silencios de separación o una combinación de estos elementos.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
  }
}

\relative c' <<
  \new Staff {
    e4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
>>
```



También se puede usar `\RemoveEmptyStaves` en el contexto `\Staff` para crear secciones de ossia para un pentagrama. Para ver más detalles, consulte [Pentagramas de Ossia], página 194.

Se puede usar la instrucción `\RemoveEmptyStaves` para ocultar los pentagramas vacíos en contextos de música antigua como `\VaticanaStaff`. De forma similar, se puede usar `\RhythmicStaff \RemoveEmptyStaves` para ocultar los contextos `RhythmicStaff` vacíos.

Instrucciones predefinidas

`\Staff \RemoveEmptyStaves`, `\VaticanaStaff \RemoveEmptyStaves`, `\RhythmicStaff \RemoveEmptyStaves`.

Fragmentos de código seleccionados

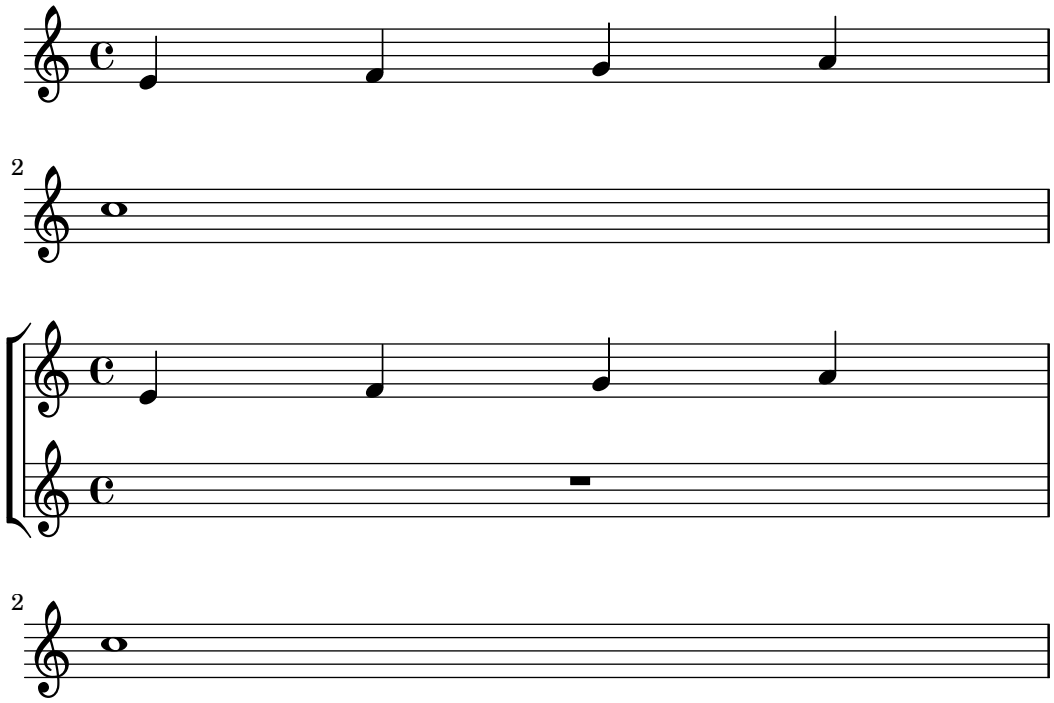
Quitar la primera línea vacía

El primer pentagrama vacío también se puede suprimir de la partitura estableciendo la propiedad `remove-first` de `VerticalAxisGroup`. Esto se puede hacer globalmente dentro del bloque `\layout`, o localmente dentro del pentagrama concreto que se quiere suprimir. En este último caso, tenemos que especificar el contexto (`Staff` se aplica sólo al pentagrama actual) delante de la propiedad.

El pentagrama inferior del segundo grupo no se elimina, porque el ajuste sólo se aplica al pentagrama concreto dentro del que se escribe.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
```

```
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```



Véase también

Glosario musical: Sección “Frenched staff” in *Glosario Musical*.

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 577, [El símbolo del pentagrama], página 191, [Pentagramas de Ossia], página 194, [Notas ocultas], página 216, [Silencios invisibles], página 56, Sección 5.4.6 [Visibilidad de los objetos], página 608.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*, Sección “Lyrics” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*, Sección “Staff_symbol_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La eliminación del grabador `Staff_symbol_engraver` también oculta las barras de compás. Si se fuerza la visibilidad de la barra de compás, pueden ocurrir errores de formato visual. En este caso, utilice las siguientes sobreescrituras de valores en vez de quitar el grabador:

```
\omit StaffSymbol
\override NoteHead.no-ledgers = ##t
```

Para ver los fallos y problemas conocidos, así como las advertencias asociadas con `\RemoveEmptyStaves`, consulte Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 577.
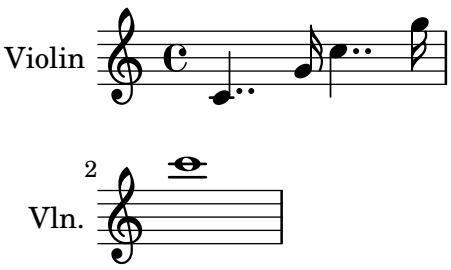
1.6.3 Escritura de las particellas

Esta sección explica cómo insertar indicaciones de tempo y nombres de instrumento en una partitura. También se describen métodos para citar otras voces y dar formato a las notas guía.

Nombres de instrumentos
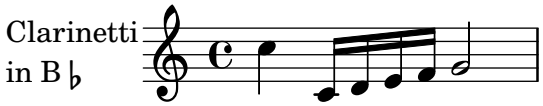
Se pueden imprimir los nombres de los instrumentos en el lado izquierdo de los pentagramas dentro de los contextos `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` y `ChoirStaff`. El valor de `instrumentName` se usa para el primer pentagrama, y el valor de `shortInstrumentName` se usa para todos los pentagramas siguientes.

```
\new Staff \with {
  instrumentName = #"Violin "
  shortInstrumentName = #"Vln. "
}
{ c4.. g'16 c4.. g'16 \break | c1 }
```
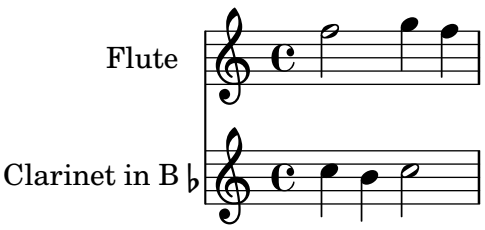


También podemos usar `\markup` para construir nombres de instrumento más complicados:

```
\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
}
{ c4 c,16 d e f g2 }
```



Cuando se agrupan dos o más contextos de pentagrama, los nombres de instrumento y los nombres cortos aparecen centrados de forma predeterminada. Para centrar nombres de instrumento de varias líneas, se debe utilizar `\center-column`:

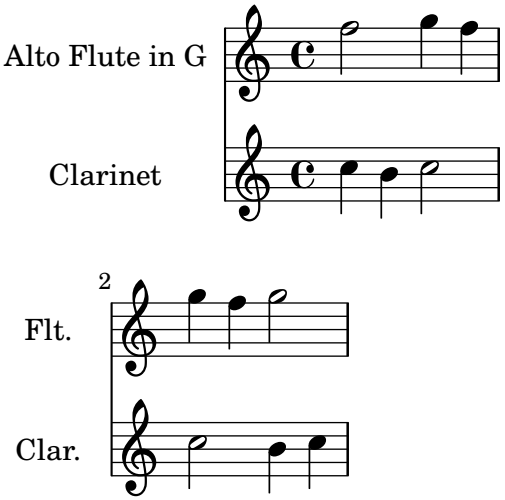
```
<<
  \new Staff \with {
    instrumentName = #"Flute"
  }
  { f2 g4 f }
  \new Staff \with {
    instrumentName = \markup {
      \center-column { "Clarinet" }
      \line { "in B" \smaller \flat }
    }
  }
  { c4 b c2 }
>>
```



Sin embargo, si los nombres de instrumento son más largos, éstos no se centran para un grupo de pentagramas a no ser que se aumenten los valores del sangrado, `indent`, y del sangrado corto, `short-indent`. Para ver más detalles sobre estos ajustes, consulte [Variables de \paper para desplazamientos y sangrados], página 521.

```
\relative c'' {
  <<
    \new Staff \with {
      instrumentName = #"Alto Flute in G"
      shortInstrumentName = #"Flt."
    }
    {
      f2 g4 f \break
      g4 f g2
    }
    \new Staff \with {
      instrumentName = #"Clarinet"
      shortInstrumentName = #"Clar."
    }
    {
      c,4 b c2 \break
      c2 b4 c
    }
  >>
}

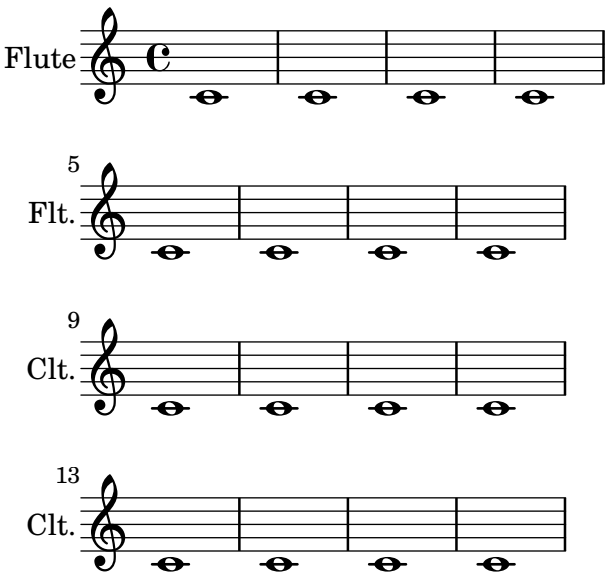
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}
```



Para añadir nombres de instrumento a otros contextos (como `ChordNames` o `FiguredBass`), debemos añadir el grabador `Instrument_name_engraver` a dicho contexto. Para ver más detalles, consulte Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 575.

El nombre de instrumento corto `shortInstrumentName` se puede cambiar en mitad de una pieza; sin embargo, sólo el primer `instrumentName` se imprime y el resto de ellos se ignoran:

```
\new Staff \with {
  instrumentName = #"Flute"
  shortInstrumentName = #"Flt."
}
{
  c1 c c c \break
  c1 c c c \break
  \set Staff.instrumentName = #"Clarinet"
  \set Staff.shortInstrumentName = #"Clt."
  c1 c c c \break
  c1 c c c \break
}
```

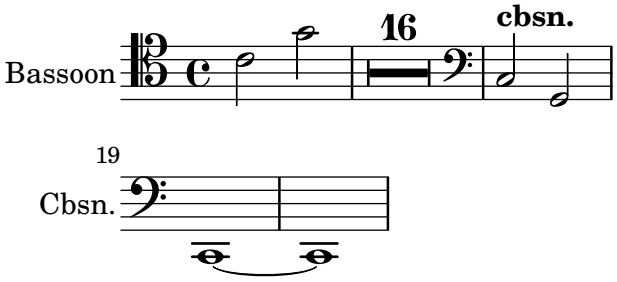


Si se necesita un *cambio* de instrumento, se debe usar `\addInstrumentDefinition` en combinación con `\instrumentSwitch` para crear una lista detallada de los cambios necesarios para el intercambio. La instrucción `\addInstrumentDefinition` tiene dos argumentos: una cadena identificativa, y una lista asociativa de propiedades de contexto y los valores que el instrumento

va a usar. Se debe situar en el ámbito del nivel más alto. `\instrumentSwitch` se usa en la expresión musical para declarar el cambio de instrumento:

```
\addInstrumentDefinition #"contrabassoon"
  #`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
     (shortInstrumentName . "Cbsn.")
     (clefGlyph . "clefs.F")
     (middleCPosition . 6)
     (clefPosition . 2)
     (instrumentCueName . ,(make-bold-markup "cbsn."))
     (midiInstrument . "bassoon"))

\new Staff \with {
  instrumentName = #"Bassoon"
}
\relative c' {
  \clef tenor
  \compressFullBarRests
  c2 g'
  R1*16
  \instrumentSwitch "contrabassoon"
  c,,2 g \break
  c,1 ~ | c1
}
```



Véase también

Referencia de la notación: [Variables de `\paper` para desplazamientos y sangrados], página 521, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 575.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “InstrumentName” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

Citar otras voces

Es muy frecuente que una voz use las mismas notas que otra voz. Por ejemplo, los violines primero y segundo tocando la misma frase durante un determinado pasaje musical. Esto se hace dejando que una voz *cite* a la otra, sin tener que volver a introducir la música para la segunda voz.

La instrucción `\addQuote`, utilizada en el ámbito del nivel sintáctico superior, define un flujo de música desde el que es posible citar fragmentos.
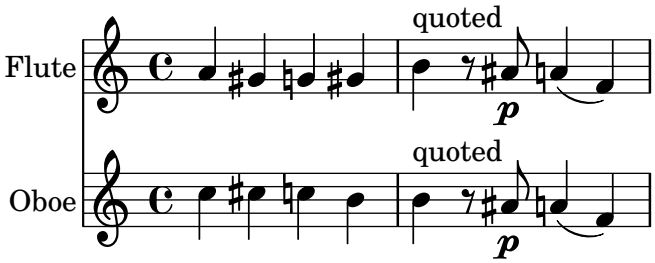
La instrucción `\quoteDuring` se usa para indicar el punto en que comienza la cita. Va seguida por dos argumentos: el nombre de la voz citada, tal y como se definió con `\addQuote`, y una expresión musical que indica la duración de la cita.

```
fluteNotes = \relative c'' {
  a4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```
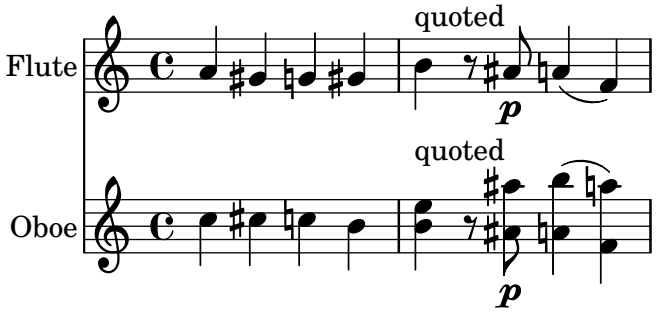


Si la expresión musical que se usa para `\quoteDuring` contiene cualquier cosa que no sea un silencio de separación o un silencio multicompás, se produce una situación de polifonía, lo que no suele ser deseable:

```
fluteNotes = \relative c'' {
  a4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"flute" { e4 r8 ais b4 a }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```
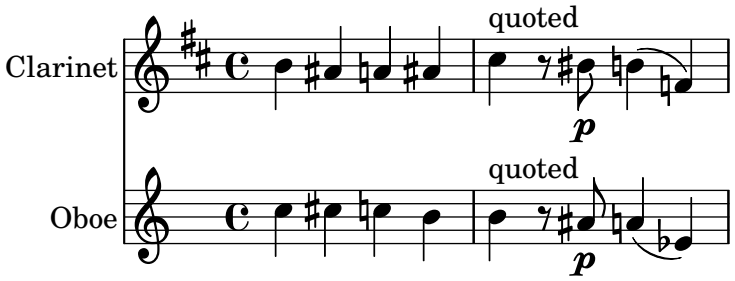


La instrucción `\quoteDuring` usa los ajustes de `\transposition` tanto de la parte citada como de la que cita, para producir notas que tienen la misma altura de sonido en la parte que cita y en las de la parte citada.

```
clarinetNotes = \relative c'' {
  \transposition bes
  \key d \major
  b4 ais a ais | cis4^"quoted" r8 bis\p b4( f)
}

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"clarinet" { s1 }
}

\addQuote "clarinet" { \clarinetNotes }


\score {
  <<
    \new Staff \with { instrumentName = "Clarinet" } \clarinetNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```
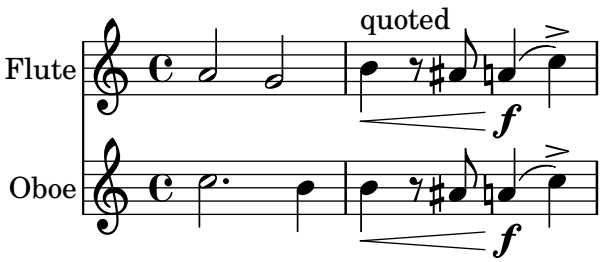


De forma predeterminada, la música citada incluye todas las articulaciones, matices dinámicos, elementos de marcado, etc. de la expresión citada. Es posible elegir cuáles de estos objetos de la música que se cita se imprimen, mediante la propiedad de contexto `quotedEventTypes`.

```
fluteNotes = \relative c'' {
  a2 g2 |
  b4\<^"quoted" r8 ais a4\f( c->)
 }

oboeNotes = \relative c'' {
  c2. b4 |
  \quoteDuring #"flute" { s1 }
}
```

```
\addQuote "flute" { \fluteNotes }

\score {
  <<
    \set Score.quotedEventTypes = #'(note-event articulation-event
                                     crescendo-event rest-event
                                     slur-event dynamic-event)
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
 }
```



Las citas también se pueden etiquetar, véase [Uso de etiquetas], página 494.

Véase también

Referencia de la notación: [Transposición de los instrumentos], página 24, [Uso de etiquetas], página 494.

Archivos de inicio: ‘`scm/define-event-classes.scm`’.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Music classes” in *Referencia de Funcionamiento Interno*, Sección “QuoteMusic” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.
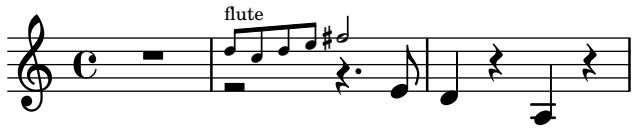
Advertencias y problemas conocidos

Sólo el contenido de la primera voz de una instrucción `\addQuote` se tiene en cuenta para la cita, de manera que si la expresión musical contiene enunciados `\new` o `\context Voice`, sus contenidos no serían citados. La cita de notas de adorno no está contemplada y puede hacer que LilyPond termine de forma abrupta; la cita de unos tresillos dentro de otros puede dar como resultado una notación de pobre calidad.

Formateo de las notas guía
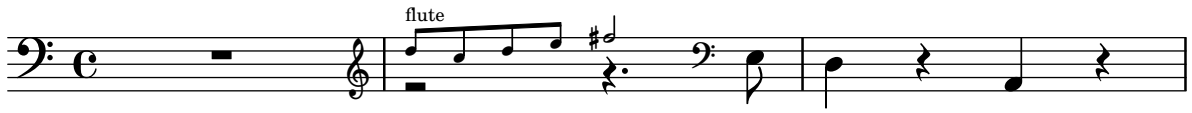
La manera más sencilla de dar formato a las notas guía es crear explícitamente un contexto `CueVoice` dentro de la parte.

```
R1
<<
  { e2\rest r4. e8 }
  \new CueVoice {
    \stemUp d'8^"flute" c d e fis2
  }
>>
d,4 r a r
```
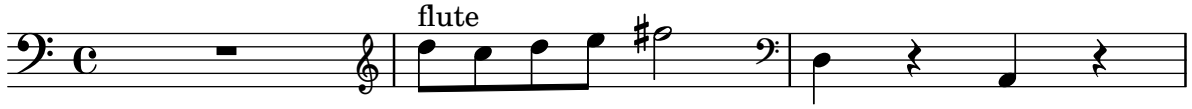


La instrucción `\cueClef` puede usarse también con un contexto `CueVoice` explícito si se requiere un cambio de clave, e imprime una clave del tamaño adecuado para las notas guía. Después puede utilizarse la instrucción `\cueClefUnset` para volver a la clave original, de nuevo con un signo de clave del tamaño adecuado.

```
\clef "bass"
R1
<<
  { e2\rest r4. \cueClefUnset e,8 }
  \new CueVoice {
    \cueClef "treble" \stemUp d''8^"flute" c d e fis2
  }
>>
d,,4 r a r
```



Las instrucciones `\cueClef` y `\cueClefUnset` se pueden usar también sin un `CueVoice` si es necesario.

```
\clef "bass"
R1
\cueClef "treble"
d'8^"flute" c d e fis2
\cueClefUnset
d,,4 r a r
```



En casos de colocación más compleja de notas guía, por ejemplo con transposición o insertando notas guía procedentes de más de una fuente, pueden usarse las instrucciones `\cueDuring` o `\cueDuringWithClef`. Son una forma más especializada de `\quoteDuring`, véase [Citar otras voces], página 204 en la sección anterior.

La sintaxis es:

`\cueDuring #nombre_de_la_cita #dirección #música`

y

`\cueDuringWithClef #nombre_de_la_cita #dirección #clave #música`
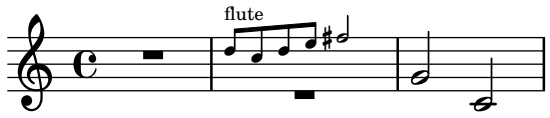
La música procedente de los compases correspondientes del elemento *`nombre_de_la_cita`* se añade como un contexto `CueVoice` y ocurre simultáneamente con la *`música`*, lo que produce una situación polifónica. La *`dirección`* toma un argumento `UP` (arriba) o `DOWN` (abajo), y corresponde a las voces primera y segunda, respectivamente, determinando cómo se imprimen las notas guía en relación a la otra voz.

```
fluteNotes = \relative c'' {
  r2. c4 | d8 c d e fis2 | g2 d |
}
```

```
oboeNotes = \relative c'' {
  R1
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuring #"flute" #UP { R1 }
  g2 c,
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \oboeNotes
}
```
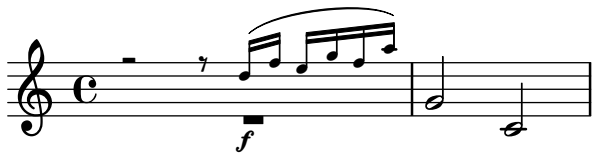


Es posible ajustar qué aspectos de la música se citan con `\cueDuring` estableciendo el valor de la propiedad `quotedCueEventTypes`. Su valor por omisión es `'(note-event rest-event tie-event beam-event tuplet-span-event)`, lo que significa que solamente las notas, silencios, ligaduras de unión, barras y grupos especiales se citan, pero no las articulaciones, marcas dinámicas, elementos de marcado, etc.

Nota: Cuando un contexto `Voice` da comienzo con `\cueDuring`, como en el ejemplo siguiente, el contexto `Voice` se debe declarar explícitamente, pues en caso contrario toda la expresión musical pertenecería al contexto `CueVoice`.

```
oboeNotes = \relative c'' {
  r2 r8 d16(\f f e g f a)
  g8 g16 g g2.
}
\addQuote "oboe" { \oboeNotes }

\new Voice \relative c'' {
  \set Score.quotedCueEventTypes = #'(note-event rest-event tie-event
                                      beam-event tuplet-span-event
                                      dynamic-event slur-event)
  \cueDuring #"oboe" #UP { R1 }
  g2 c,
}
```
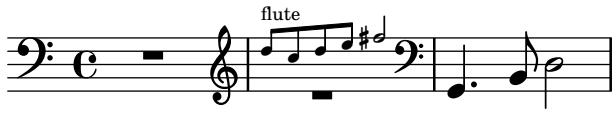


El nombre del instrumento que está tocando se puede imprimir si se ajusta la propiedad `instrumentCueName` dentro de un contexto `CueVoice` provisional. La colocación y estilo del `instrumentCueName` viene controlado a través del objeto `InstrumentSwitch`, véase [Nombres de instrumentos], página 201. Si las notas guía requieren un cambio de clave, puede hacerse manualmente pero la clave original se debe restaurar al final de las notas guía.

```
fluteNotes = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \clef treble
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuring #"flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}
```



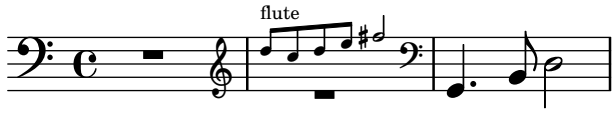
De forma alternativa, puede usarse la función `\cueDuringWithClef` en su lugar. Esta instrucción admite un argumento adicional para especificar el cambio de clabe que se necesita imprimir para las notas guía, pero después imprime automáticamente la clave original una vez que ha finalizado la serie de notas guía.

```
fluteNotes = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuringWithClef #"flute" #UP #"treble" { R1 }
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}
```



Como `\quoteDuring`, `\cueDuring` tiene en cuenta las transposiciones instrumentales. Las notas guía se producen en las alturas en que se escribirían para el instrumento que recibe la cita, para así producir las alturas de sonido del instrumento original.

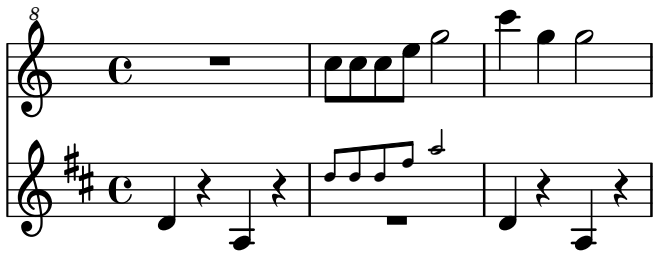
Para transportar las notas guía de forma diferente, use `\transposedCueDuring`. Esta instrucción acepta un argumento adicional para especificar (en modo absoluto) la altura impresa con que queremos representar el sonido de un Do central de concierto. Esto es útil para toma citas de un instrumento que está en un registro completamente diferente.

```
piccoloNotes = \relative c''' {
  \clef "treble^8"
  R1
  c8 c c e g2
  c4 g g2
}

bassClarinetNotes = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring #"piccolo" #UP d { R1 }
  d4 r a r
}

\addQuote "piccolo" { \piccoloNotes }

<<
  \new Staff \piccoloNotes
  \new Staff \bassClarinetNotes
>>
```



La instrucción `\killCues` elimina las notas guía de una expresión musical, de forma que la misma expresión musical pueda utilizarse para producir la particella instrumental (con notas guía) y la partitura de conjunto. La instrucción `\killCues` elimina solamente las notas y eventos que se han citado mediante `\cueDuring`. Otros elementos de marcado asociados con las guías, como los cambios de clave o una etiqueta identificativa del instrumento fuente, pueden marcarse para su inclusión selectiva dentro de la partitura; véase [Uso de etiquetas], página 494.

```
fluteNotes = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}

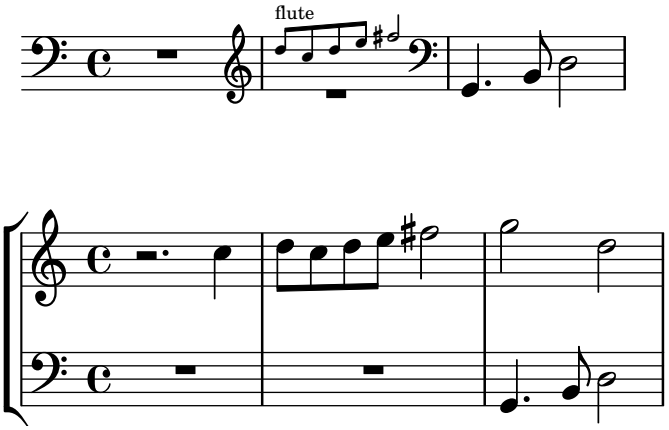
bassoonNotes = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    \new CueVoice { \set instrumentCueName = "flute" }
  }
  \cueDuring #"flute" #UP { R1 }
```

```
  \tag #'part \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

\new StaffGroup <<
  \new Staff {
    \fluteNotes
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoonNotes } }
  }
>>
```



Como alternativa, los cambios de clave y los letreros de instrumento se pueden recolectar dentro de una definición de instrumento para su uso repetido, utilizando `\addInstrumentDefinition` que se descibe en [Nombres de instrumentos], página 201.

Véase también

Referencia de la notación: [Citar otras voces], página 204, [Transposición de los instrumentos], página 24, [Nombres de instrumentos], página 201, [Clave], página 16, [Guías musicales], página 296, [Uso de etiquetas], página 494.

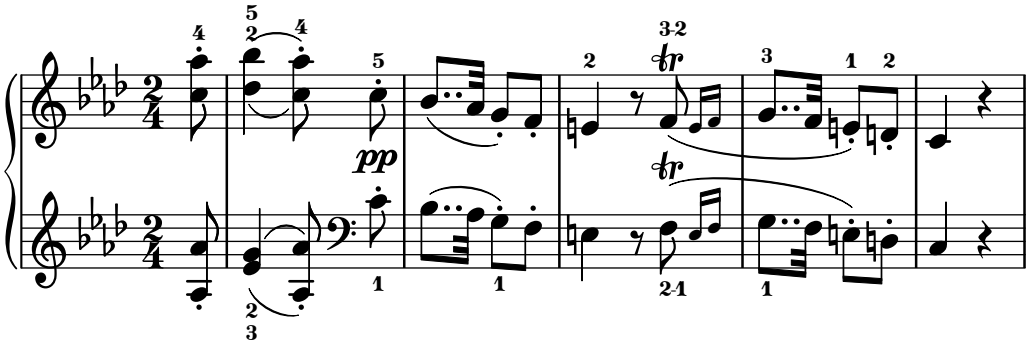
Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CueVoice” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Pueden ocurrir colisiones con los silencios si se usa `\cueDuring`, entre los contextos de `Voice` y de `CueVoice`. Al usar `\cueDuringWithClef` o `\transposedCueDuring` el argumento adicional requerido para cada caso debe ir después de la cita y de la dirección.

1.7 Anotaciones editoriales



Esta sección trata de las diversas maneras de modificar el aspecto de las notas y de aplicar énfasis analítico o educativo.

1.7.1 Dentro del pentagrama

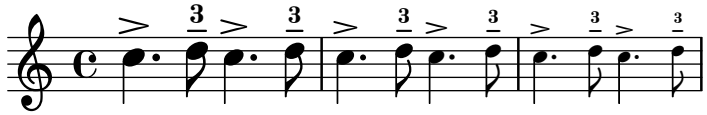
Esta sección trata sobre cómo aplicar énfasis a los elementos situados dentro del pentagrama.

Seleccionar el tamaño de la tipografía para la notación

Se puede alterar el tamaño de la fuente tipográfica de los elementos de notación. Esto no cambia el tamaño de los símbolos variables, como las barras de corchea o las ligaduras.
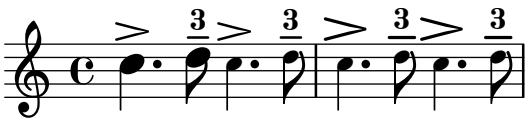
> **Nota:** Para los tamaños de tipografía del texto, consulte [Seleccionar la tipografía y su tamaño], página 234.

```
\huge
c4.-> d8---3
\large
c4.-> d8---3
\normalsize
c4.-> d8---3
\small
c4.-> d8---3
\tiny
c4.-> d8---3
\teeny
c4.-> d8---3
```



Internamente, esto establece un valor para la propiedad `fontSize`. Esto, a su vez, hace que se establezca el valor de la propiedad `font-size` de todos los objetos de presentación. El valor de `font-size` es un número que indica el tamaño con relación al tamaño estándar para la altura vigente del pentagrama. Cada unidad hacia arriba es un incremento de un 12% del tamaño de la fuente, aproximadamente. Seis unidades son exactamente un factor de dos. La función de Scheme `magstep` convierte un número de tamaño `font-size` a un factor de escalado. La propiedad `font-size` también se puede establecer directamente, de forma que queden afectados solamente determinados objetos gráficos.

```
\set fontSize = #3
c4.-> d8---3
\override NoteHead.font-size = #-4
c4.-> d8---3
\override Script.font-size = #2
c4.-> d8---3
\override Stem.font-size = #-5
c4.-> d8---3
```



Los cambios en el tamaño de la fuente se obtienen por medio del escalado del tamaño del diseño que se encuentra más cerca del tamaño deseado. El tamaño estándar para la tipografía (para `font-size = #0`), depende de la altura estándar del pentagrama. Para un pentagrama de 20 puntos, se selecciona una fuente de 10 puntos.

La propiedad `font-size` sólo se puede establecer en objetos gráficos que utilicen fuentes tipográficas. Éstos son los que contemplan el interfaz de presentación `font-interface`.

Instrucciones predefinidas

`\teeny` (enano), `\tiny` (muy pequeño), `\small` (pequeño), `\normalsize` (normal), `\large` (grande), `\huge` (enorme).

Véase también
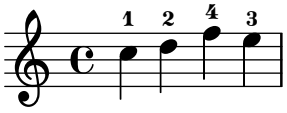
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “font-interface” in *Referencia de Funcionamiento Interno*.

Indicaciones de digitación
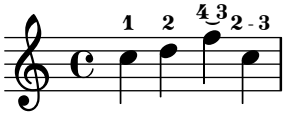
Las instrucciones de digitación se pueden introducir usando ‘*nota-dígito*’:

```
c4-1 d-2 f-4 e-3
```
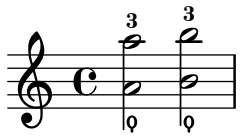


Para los cambios de dedo se pueden usar elementos de marcado de texto o de cadenas de caracteres.

```
c4-1 d-2 f\finger \markup \tied-lyric #"4~3" c\finger "2 - 3"
```
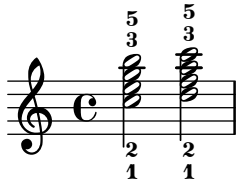


Puede usar la articulación de pulgar para indicar que una nota se debe tocar con el pulgar (p.ej. en música de violoncello).

```
<a_\thumb a'-3>2 <b_\thumb b'-3>
```



Las digitaciones para los acordes también se pueden añadir a las notas individuales escribiéndolas después de las alturas.

```
<c-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
```



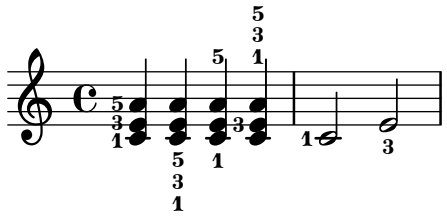
Las indicaciones de digitación se pueden situar manualmente encima o debajo del pentagrama, véase Sección 5.4.2 [Dirección y posición], página 601.

Fragmentos de código seleccionados

Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación. Para que se tenga en cuenta la orientación de las digitaciones, se debe utilizar una construcción de acorde <> aunque sea una sola nota.

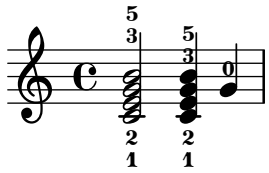
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Nota: se debe usar una construcción de acorde <>, aunque sea una sola nota.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



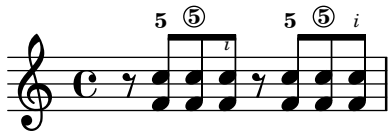
Evitar colisiones con digitaciones de acordes

Las digitaciones y números de cuerda que se aplican a las notas individuales evitan automáticamente las barras y las plicas de las figuras, pero esto no es cierto de forma predeterminada para las digitaciones y números de cuerda que se aplican sobre notas individuales de acordes. El ejemplo siguiente muestra cómo se puede sobreescribir este comportamiento predeterminado.

```
\relative c' {
  \set fingeringOrientations = #'(up)
  \set stringNumberOrientations = #'(up)
  \set strokeFingerOrientations = #'(up)

  % Default behavior
  r8
  <f c'-5>8
  <f c'\5>8
  <f c'-\rightHandFinger #2 >8

  % Corrected to avoid collisions
  r8
  \override Fingering.add-stem-support = ##t
  <f c'-5>8
  \override StringNumber.add-stem-support = ##t
  <f c'\5>8
  \override StrokeFinger.add-stem-support = ##t
  <f c'-\rightHandFinger #2 >8
}
```



Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601.

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.
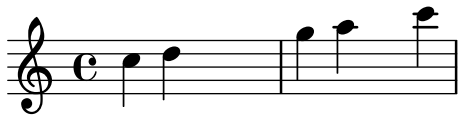
Referencia de funcionamiento interno: Sección “FingeringEvent” in *Referencia de Funcionamiento Interno*, Sección “fingering-event” in *Referencia de Funcionamiento Interno*, Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*, Sección “New_fingering_engraver” in *Referencia de Funcionamiento Interno*, Sección “Fingering” in *Referencia de Funcionamiento Interno*.

Notas ocultas

Las notas ocultas (o invisibles, o transparentes) pueden ser de utilidad en la preparación de ejercicios de teoría o de composición.
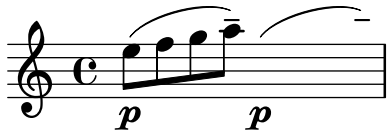
```
c4 d
\hideNotes
e4 f
```

```
\unHideNotes
g a
\hideNotes
b
\unHideNotes
c
```



La cabeza, la plica y el corchete de las figuras, así como los silencios, son invisibles. Las barras son invisibles si comienzan en una nota oculta. Los objetos de notación que están anclados a notas invisibles, son a pesar de ello visibles.

```
e8(\p f g a)--
\hideNotes
e8(\p f g a)--
```



Instrucciones predefinidas

`\hideNotes`, `\unHideNotes`.

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*.

Notation Reference: [Silencios invisibles], página 56, Sección 5.4.6 [Visibilidad de los objetos], página 608, [Ocultar pentagramas], página 198.

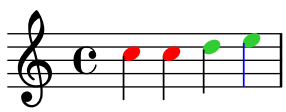
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_spacing_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” in *Referencia de Funcionamiento Interno*.

Colorear objetos

Se pueden asignar colores a los objetos individuales. Los nombres de color válidos se encuentran relacionados en la Sección A.7 [Lista de colores], página 648.

```
\override NoteHead.color = #red
c4 c
\override NoteHead.color = #(x11-color 'LimeGreen)
d
\override Stem.color = #blue
e
```
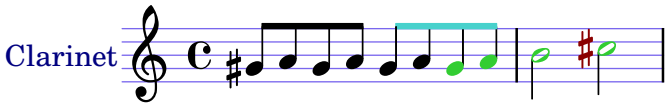


Se puede acceder al espectro completo de colores definido para X11 usando la función de Scheme x11-color. Esta función acepta un único argumento; éste puede ser un símbolo de la

forma *'FulanoMengano* o una cadena de la forma *"FulanoMengano"*. La primera forma es más rápida de escribir y también más eficiente. Sin embargo, al usar la segunda forma es posible acceder a los colores de X11 por medio de la forma del nombre que tiene varias palabras.

Si `x11-color` no entiende el parámetro, el color predeterminado que se devuelve es el negro.

```
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
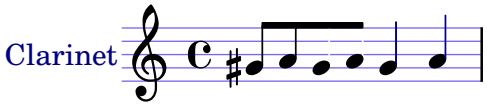
gis8 a
\override Beam.color = #(x11-color "medium turquoise")
gis a
\override Accidental.color = #(x11-color 'DarkRed)
gis a
\override NoteHead.color = #(x11-color "LimeGreen")
gis a
% this is deliberate nonsense; note that the stems remain black
\override Stem.color = #(x11-color 'Boggle)
b2 cis
```



Se pueden especificar colores RGB exactos utilizando la función de Scheme `rgb-color`.

```
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}

\override Stem.color = #(rgb-color 0 0 0)
gis8 a
\override Stem.color = #(rgb-color 1 1 1)
gis8 a
\override Stem.color = #(rgb-color 0 0 0.5)
gis4 a
```



Véase también

Referencia de la notación: Sección A.7 [Lista de colores], página 648, Sección 5.3.4 [La instrucción tweak], página 595.

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Advertencias y problemas conocidos

Un color de X11 no es necesariamente de la misma tonalidad exacta que un color normal de nombre similar.
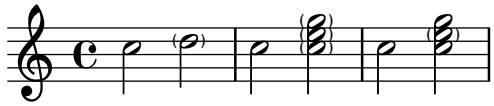
No todos los colores de X11 se distinguen entre sí en un navegador web, es decir, un navegador de web podría no mostrar ninguna diferencia entre `LimeGreen` (verde lima) y `ForestGreen` (verde bosque). Para la web se recomiendan los colores normales (o sea: `blue`, azul, `green`, verde, `red`, rojo).

Las notas de un acorde no se pueden colorear con `\override`; en su lugar utilice `\tweak`. Consulte Sección 5.3.4 [La instrucción tweak], página 595 para ver más detalles.

Paréntesis
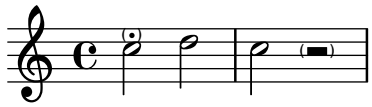
Los objetos se pueden encerrar entre paréntesis anteponiendo `\parenthesize` al evento musical. Si se aplica a un acorde, encierra cada nota dentro de un par de paréntesis. También se pueden poner entre paréntesis las notas individuales de un acorde.

```
c2 \parenthesize d
c2 \parenthesize <c e g>
c2 <c \parenthesize e g>
```



Los objetos que no son notas también se pueden poner entre paréntesis. Para las articulaciones se necesita un guión antes de la instrucción `\parenthesize`.

```
c2-\parenthesize -. d
c2 \parenthesize r
```



Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Parenthesis_engraver” in *Referencia de Funcionamiento Interno*, Sección “ParenthesesItem” in *Referencia de Funcionamiento Interno*, Sección “parentheses-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Al poner un acorde entre paréntesis, se encierra cada una de las notas individuales entre paréntesis, en vez de un solo paréntesis grande rodeando al acorde completo.

Plicas

Cuando se encuentra con una nota, se crea automáticamente un objeto `Stem` (plica). Para las redondas y los silencios, también se crean pero se hacen invisibles.

Se puede hacer manualmente que las plicas apunten hacia arriba o hacia abajo; véase Sección 5.4.2 [Dirección y posición], página 601.

Instrucciones predefinidas
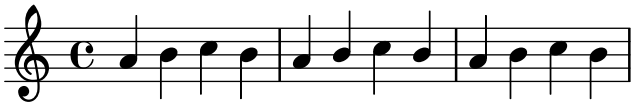
`\stemUp`, `\stemDown`, `\stemNeutral`.

Fragmentos de código seleccionados

Dirección predeterminada de las plicas sobre la tercera línea del pentagrama

La dirección predeterminada de las plicas sobre la tercera línea del pentagrama está determinada por la propiedad `neutral-direction` del objeto `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```
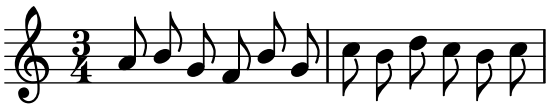


Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía

LilyPond puede alterar la dirección de la plica de las notas que van en la tercera línea de un pentagrama de forma que siga la melodía, mediante la adición del grabador `Melody_engraver` al contexto Voice y sobreescribiendo el valor de `neutral-direction` para el objeto Stem (plica).

```
\relative c'' {
  \time 3/4
  \autoBeamOff
  a8 b g f b g |
  c b d c b c
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \override Stem.neutral-direction = #'()
  }
}
```



Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 601.

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem_engraver” in *Referencia de Funcionamiento Interno*, Sección “Stem” in *Referencia de Funcionamiento Interno*, Sección “stem-interface” in *Referencia de Funcionamiento Interno*.
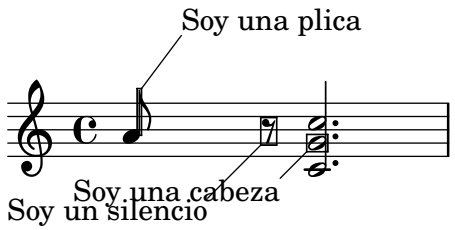
1.7.2 Fuera del pentagrama

Esta sección trata sobre cómo enfatizar elementos que están dentro del pentagrama, desde fuera del pentagrama.

Globos de ayuda

Los elementos de notación se pueden marcar y nombrar con la ayuda de un cartel o globo de ayuda rectangular. El propósito principal de esta funcionalidad es la explicación de la notación.
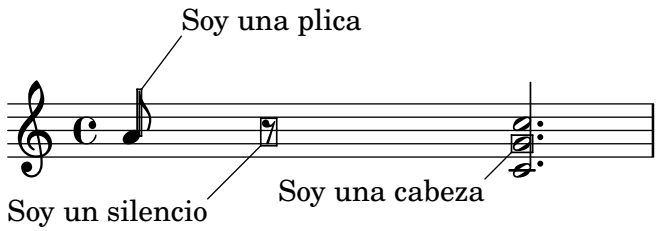
```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```



Existen dos funciones musicales, `balloonGrobText` y `balloonText`; el primero se usa como `\once\override` para adjuntar un texto a cualquier grob, y el último se usa como `\tweak`, normalmente dentro de acordes, para adjuntar un texto a una nota individual.

Los textos de globo no influyen en el espaciado de las notas, pero esto puede cambiarse:

```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```



Instrucciones predefinidas

`\balloonLengthOn`, `\balloonLengthOff`.

Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Balloon_engraver” in *Referencia de Funcionamiento Interno*, Sección “BalloonTextItem” in *Referencia de Funcionamiento Interno*, Sección “balloon-interface” in *Referencia de Funcionamiento Interno*.

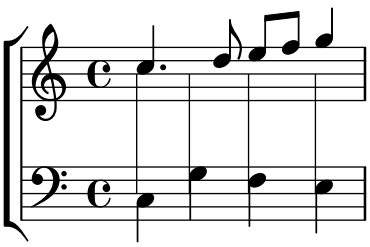
Líneas de rejilla

Se pueden dibujar líneas verticales entre los pentagramas sincronizadas con las notas.

Se debe usar el grabador `Grid_point_engraver` para crear los puntos extremos de las líneas, mientras que el grabador `Grid_line_span_engraver` se debe utilizar para trazar efectivamente las líneas. De forma predeterminada, esto centra las líneas de rejilla horizontalmente debajo y al lado izquierdo de la cabeza de las notas. Las líneas de rejilla se extienden a partir de línea media de los pentagramas. El intervalo `gridInterval` debe especificar la duración entre las líneas de rejilla.

```
\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #(ly:make-moment 1/4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative c'' {
      \stemUp
      c4. d8 e8 f g4
    }
    \new Staff \relative c {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```
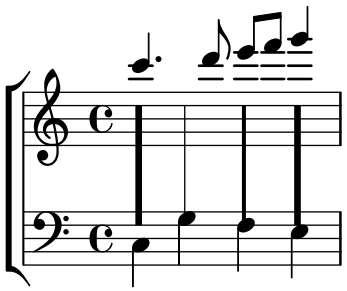


Fragmentos de código seleccionados

Líneas de rejilla: modificar su aspecto

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
```

```
        \stemUp
        c'4. d8 e8 f g4
      }
    }
    \new Staff {
      \relative c {
        % this moves them up one staff space from the default position
        \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
        \stemDown
        \clef bass
        \once \override Score.GridLine.thickness = #5.0
        c4
        \once \override Score.GridLine.thickness = #1.0
        g'4
        \once \override Score.GridLine.thickness = #3.0
        f4
        \once \override Score.GridLine.thickness = #5.0
        e4
      }
    }
  >>
  \layout {
    \context {
      \Staff
      % set up grids
      \consists "Grid_point_engraver"
      % set the grid interval to one quarter note
      gridInterval = #(ly:make-moment 1/4)
    }
    \context {
      \Score
      \consists "Grid_line_span_engraver"
      % this moves them to the right half a staff space
      \override NoteColumn.X-offset = #-0.5
    }
  }
}
```



Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

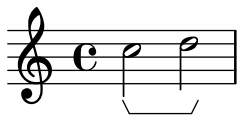
Referencia de funcionamiento interno: Sección “Grid_line_span_engraver” in *Referencia de Funcionamiento Interno*, Sección “Grid_point_engraver” in *Referencia de Funcionamiento Inter-*

no, Sección “GridLine” in *Referencia de Funcionamiento Interno*, Sección “GridPoint” in *Referencia de Funcionamiento Interno*, Sección “grid-line-interface” in *Referencia de Funcionamiento Interno*, Sección “grid-point-interface” in *Referencia de Funcionamiento Interno*.

Corchetes de análisis

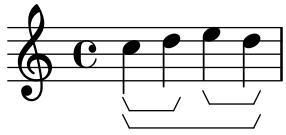
Los corchetes se usan en análisis musical para indicar la estructura de las piezas musicales. Están contemplados los corchetes horizontales simples.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c2\startGroup
  d\stopGroup
}
```



Los corchetes de análisis se pueden anidar.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```
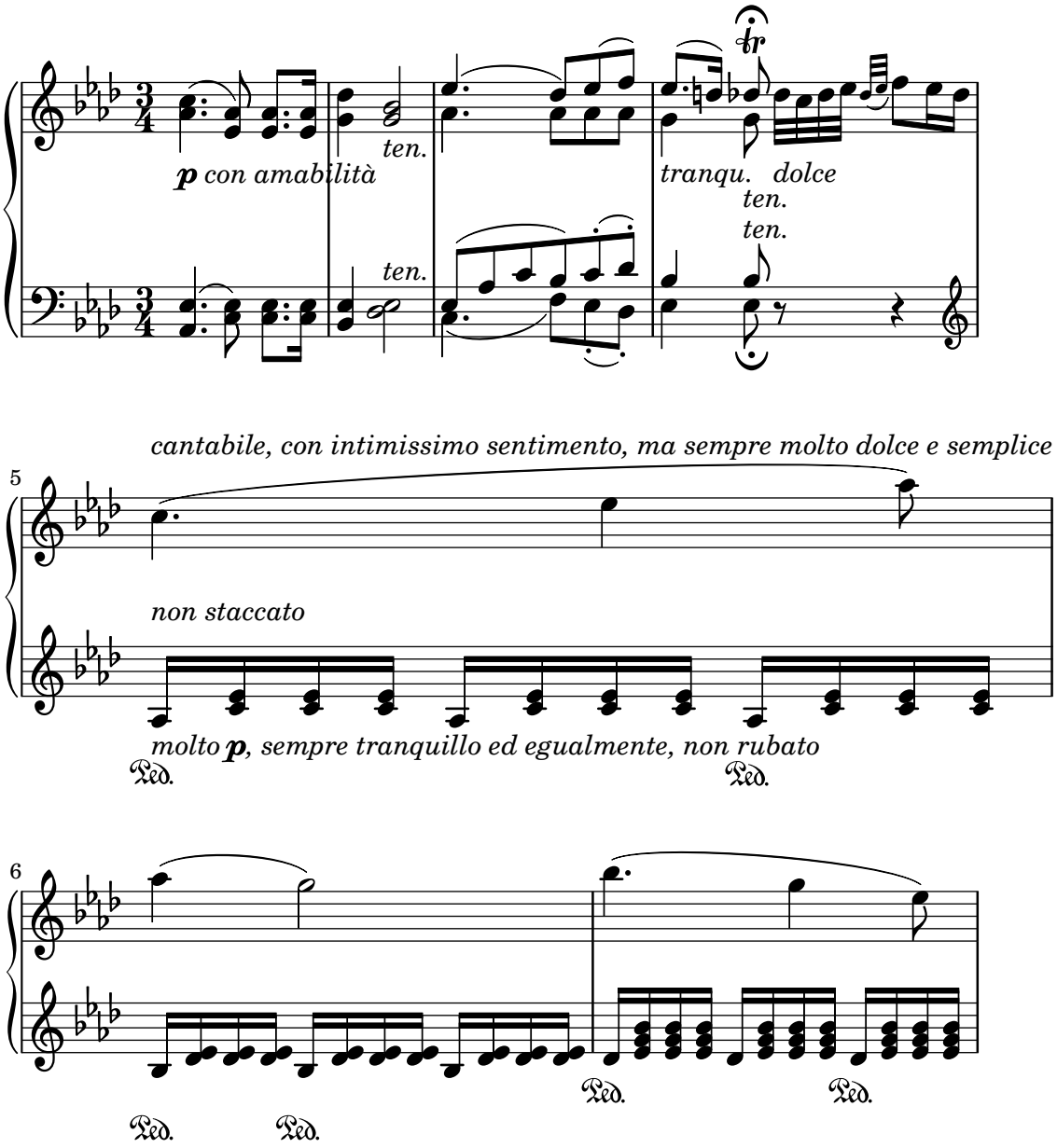


Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Horizontal_bracket_engraver” in *Referencia de Funcionamiento Interno*, Sección “HorizontalBracket” in *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-interface” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

1.8 Texto



Esta sección explica cómo incluir texto (con diversos estilos de formateo) en las partituras.

Ciertos elementos de texto de los que no nos ocupamos aquí se tratan en otras secciones específicas: Sección 2.1 [Música vocal], página 250, Sección 3.2 [Títulos y encabezamientos], página 465.

1.8.1 Escritura del texto
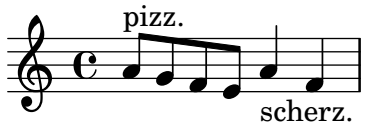
Esta sección presenta las distintas formas de añadir texto a una partitura.

Nota: Para escribir texto con caracteres acentuados y especiales (como los de idiomas distintos del inglés), sencillamente inserte los caracteres directamente en el archivo de LilyPond. El archivo se debe guardar como UTF-8. Para ver más información, consulte [Codificación del texto], página 497.

Guiones de texto
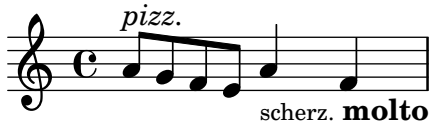
Es posible añadir indicaciones de "texto entre comillas" a una partitura, como se muestra en el ejemplo siguiente. Estas indicaciones se pueden colocar manualmente por encima o por debajo del pentagrama, utilizando la sintaxis que se describe en Sección 5.4.2 [Dirección y posición], página 601.

```
a8^"pizz." g f e a4-"scherz." f
```
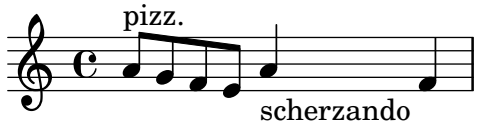


Esta sintaxis es en realidad una abreviatura; se puede añadir explícitamente a una nota un formateado de texto más complejo utilizando un bloque `\markup`, como se describe bajo Sección 1.8.2 [Formatear el texto], página 233.

```
a8^\markup { \italic pizz. } g f e
a4_\markup { \tiny scherz. \bold molto } f
```



De forma predeterminada, las indicaciones de texto no afectan al espaciado de las notas. Sin embargo, su anchura sí podría tenerse en cuenta: en el siguiente ejemplo, la primera cadena de texto no afecta al espaciado, pero la segunda sí afecta.

```
a8^"pizz." g f e
\textLengthOn
a4_"scherzando" f
```



Se pueden adjuntar articulaciones a las notas, además de inscripciones de texto. Para ver más información, consulte [Articulaciones y ornamentos], página 116.

Para ver más información sobre el orden relativo de las inscripciones de texto y las articulaciones, consulte Sección "Colocación de los objetos" in *Manual de Aprendizaje*.

Instrucciones predefinidas

`\textLengthOn`, `\textLengthOff`.

Véase también

Manual de aprendizaje: Sección "Colocación de los objetos" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 233, Sección 5.4.2 [Dirección y posición], página 601, [Articulaciones y ornamentos], página 116.

Fragmentos de código: Sección "Text" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextScript" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

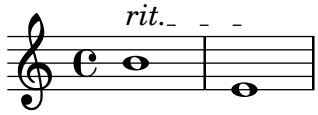
La comprobación necesaria para asegurar que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, requiere cálculos adicionales. En caso de que desee un proceso ligeramente más rápido, puede utilizar

```
\override Score.PaperColumn.keep-inside-line = ##f
```

Extensiones de texto

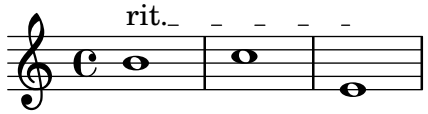
Ciertas indicaciones de ejecución, p.ej., *rallentando* o *accelerando*, se escriben como texto y se extienden sobre muchos compases mediante líneas de puntos. Estos objetos, que reciben el nombre de 'spanners' u objetos de extensión, se pueden crear desde una nota hasta otra usando la siguiente sintaxis:

```
\override TextSpanner.bound-details.left.text = "rit."
b1\startTextSpan
e,\stopTextSpan
```



La cadena de texto que imprimir se establece a través de propiedades de objeto. De forma predeterminada se imprime en estilo cursiva, pero se pueden conseguir distintos efectos de formato utilizando bloques `\markup`, como se describe en Sección 1.8.2 [Formatear el texto], página 233.

```
\override TextSpanner.bound-details.left.text =
  \markup { \upright "rit." }
b1\startTextSpan c
e,\stopTextSpan
```



El estilo de la línea, así como la cadena de texto, se pueden definir como una propiedad de objeto. Esta sintaxis se describe en Sección 5.4.7 [Estilos de línea], página 614.

Instrucciones predefinidas

`\textSpannerUp`, `\textSpannerDown`, `\textSpannerNeutral`.

Advertencias y problemas conocidos

LilyPond sólo puede manejar un objeto de extensión de texto por cada voz.

Fragmentos de código seleccionados
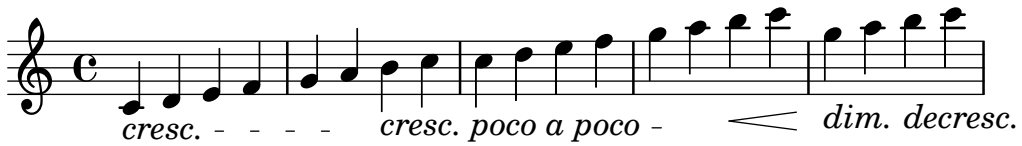
Objetos extensores de texto postfijos para dinámica

Los objetos de extensión \cresc, \dim y \decresc ahora se pueden redefinir como operadores postfijos y producir un solo objeto de extensión de texto. La definición de extensores personalizados también es fácil. Se pueden mezclar con facilidad los crescendi textuales y en forma de reguladores. `\<` y `\>` producen reguladores gráficos de forma predeterminada, \cresc etc. producen elementos extensores de texto de forma predeterminada.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
```

```
            'span-direction START
            'span-type 'text
            'span-text "cresc. poco a poco")

\relative c' {
  c4\cresc d4 e4 f4 |
  g4 a4\! b4\crpoco c4 |
  c4 d4 e4 f4 |
  g4 a4\! b4\< c4 |
  g4\dim a4 b4\decresc c4\!
}
```

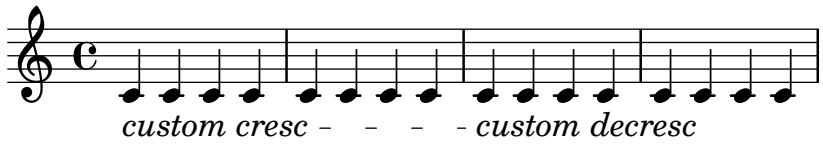


Objeto personalizado de extensión de texto de matices dinámicos, postfijo

Funciones postfijas para la creación de objetos de extensión de texto personalizados. Los objetos de extensión deben comenzar en la primera nota del compás. Hay que utilizar -\mycresc, en caso contrario el comienzo del eobjeto de extensión se asignará a la nota siguiente.

```
% Two functions for (de)crescendo spanners where you can explicitly give the
% spanner text.
mycresc =
#(define-music-function (parser location mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (parser location mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
}
```



Véase también

Referencia de la notación: Sección 5.4.7 [Estilos de línea], página 614, [Matices dinámicos], página 119, Sección 1.8.2 [Formatear el texto], página 233.
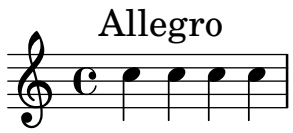
Fragmentos de código: Sección “Text” in *Fragmentos de código*, Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextSpanner” in *Referencia de Funcionamiento Interno*.

Indicaciones de texto
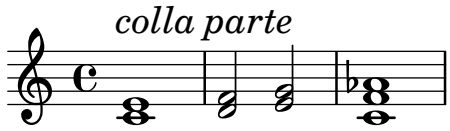
Se pueden añadir a la partitura varios elementos de texto con la sintaxis que se describe en [Llamadas de ensayo], página 106:

```
c4
\mark "Allegro"
c c c
```
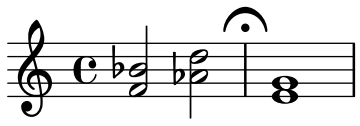


Esta sintaxis posibilita colocar cualquier texto sobre la línea divisoria; se pueden incorporar formatos más complejos para el texto usando un bloque `\markup`, como está explicado en Sección 1.8.2 [Formatear el texto], página 233:

```
<c e>1
\mark \markup { \italic { colla parte } }
<d f>2 <e g>
<c f aes>1
```



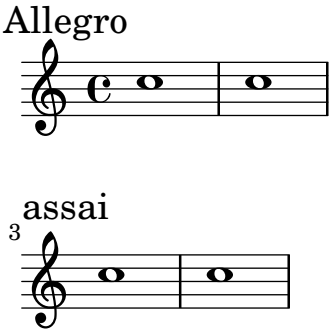
Esta sintaxis también permite imprimir símbolos especiales como llamadas, segno o calderones, especificando el nombre del símbolo correspondiente como se explica en [Notación musical dentro de elementos de marcado], página 243:

```
<bes f>2 <aes d>
\mark \markup { \musicglyph #"scripts.ufermata" }
<e g>1
```



Estos objetos se tipografían solamente sobre el pentagrama superior de la partitura; dependiendo de si están especificados al final o en medio de un compás, se colocarán sobre la línea divisoria o entre las notas. Si se especifican en un salto de línea, las llamadas se imprimen al principio de la línea siguiente.

```
\mark "Allegro"
c1 c
\mark "assai" \break
c  c
```



Instrucciones predefinidas
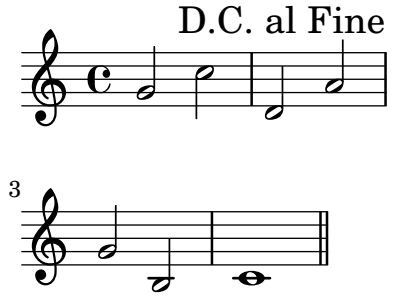
`\markLengthOn`, `\markLengthOff`.

Fragmentos de código seleccionados

Imprimir marcas al final de una línea

Se pueden imprimir marcas al final de la línea actual, en vez de al principio de la línea siguiente. En estos casos, puede ser preferible alinear el borde derecho de la marca con la línea divisoria.

```
\relative c'' {
  g2 c
  d,2 a'
  \once \override Score.RehearsalMark.break-visibility = #end-of-line-visible
  \once \override Score.RehearsalMark.self-alignment-X = #RIGHT
  \mark "D.C. al Fine"
  \break
  g2 b,
  c1 \bar "||"
}
```

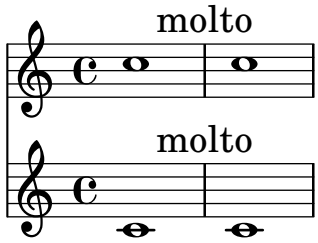


Imprimir marcas de ensayo en cualquier pentagrama

Aunque normalmente las marcas de ensayo textuales sólo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
```

```
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```



Véase también

Referencia de la notación: [Llamadas de ensayo], página 106, Sección 1.8.2 [Formatear el texto], página 233, [Notación musical dentro de elementos de marcado], página 243, Sección A.8 [La tipografía Feta], página 650.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MarkEvent” in *Referencia de Funcionamiento Interno*, Sección “Mark_engraver” in *Referencia de Funcionamiento Interno*, Sección “RehearsalMark” in *Referencia de Funcionamiento Interno*.

Texto separado

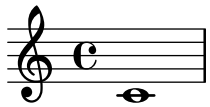
Un bloque `\markup` puede existir de forma independiente, fuera de cualquier bloque `\score`, como una “expresión de nivel superior”. Esta sintaxis se describe en Sección 3.1.5 [Estructura del archivo], página 463.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

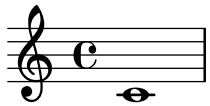
Tomorrow, and tomorrow, and tomorrow...

Esto hace posible imprimir texto separado de la música, lo que es especialmente útil cuando el archivo de entrada contiene varias piezas, tal y como se describe en Sección 3.1.2 [Varias partituras en un libro], página 460.

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Los bloques de texto independientes pueden abarcar varias páginas, posibilitando la impresión de documentos de texto o libros íntegramente desde LilyPond. Esta funcionalidad y la sintaxis que requiere se describen en [Elementos de marcado de varias páginas], página 245.

Instrucciones predefinidas

`\markup`, `\markuplist`.

Fragmentos de código seleccionados

Elemento de marcado de texto independiente en dos columnas

Los textos independientes se pueden disponer en varias columnas utilizando instrucciones `\markup`:

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
    \hspace #2
    \column \italic {
      \line { O sacred feast }
      \line { in which Christ is received, }
      \line { the memory of His Passion is renewed, }
      \line { the mind is filled with grace, }
      \line { and a pledge of future glory is given to us. }
      \line { Amen. }
    }
    \hspace #1
  }
}
```

O sacrum convivium
in quo Christus sumitur,
recolitur memoria passionis ejus,
mens impletur gratia,
futurae gloriae nobis pignus datur.
Amen.

O sacred feast
in which Christ is received,
the memory of His Passion is renewed,
the mind is filled with grace,
and a pledge of future glory is given to us.
Amen.

Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 233, Sección 3.1.5 [Estructura del archivo], página 463, Sección 3.1.2 [Varias partituras en un libro], página 460, [Elementos de marcado de varias páginas], página 245.

Fragmentos de código: Sección "Text" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextScript" in *Referencia de Funcionamiento Interno*.

1.8.2 Formatear el texto

Esta sección presenta los formateados básico y avanzado de texto, usando la sintaxis específica del modo de `\markup`.

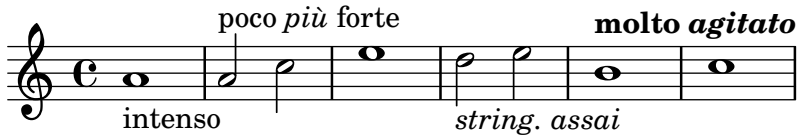
Introducción al marcado de texto

Se usa un bloque `\markup` para tipografiar texto con una sintaxis ampliable que se denomina "modo de marcado".

La sintaxis del marcado es similar a la sintaxis usual de LilyPond: una expresión `\markup` se encierra entre llaves `{ ... }`. Una sola palabra se considera como una expresión mínima, y por tanto no necesita estar encerrada entre llaves.

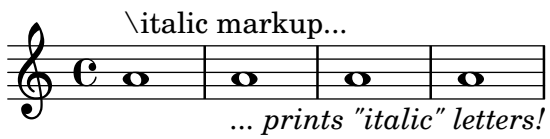
A diferencia de las indicaciones de "texto entrecomillado" simples, los bloques `\markup` pueden contener expresiones anidadas o instrucciones de marcado, que se introducen utilizando el carácter de barra invertida `\` . Estas instrucciones sólo afectan a la expresión que sigue inmediatamente.

```
a1-\markup intenso
a2^\markup { poco \italic più forte  }
c e1
d2_\markup { \italic "string. assai" }
e
b1^\markup { \bold { molto \italic  agitato } }
c
```



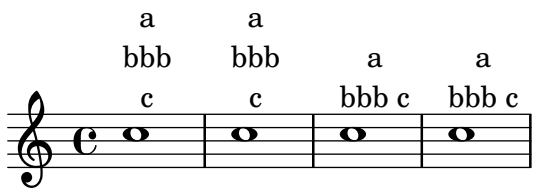
Un bloque `\markup` puede contener también cadenas de texto entre comillas. Dichas cadenas se tratan como expresiones de texto mínimas, y por tanto cualquier instrucción de marcado o carácter especial (como `\` y `#`) se imprimen literalmente sin afectar al formateo del texto. Se puede hacer que se impriman las propias comillas si se les antepone una barra invertida.

```
a1^"\italic markup..."
a_\markup { \italic "... prints \"italic\" letters!" }
a a
```



Para que se traten como una expresión distinta, es necesario que las listas de palabras estén encerradas entre comillas o precedidas de una instrucción. La forma en que están definidas las expresiones de marcado afecta a cómo se apilan, se centran y se alinean estas expresiones; en el ejemplo siguiente, la segunda expresión `\markup` se trata igual que la primera:
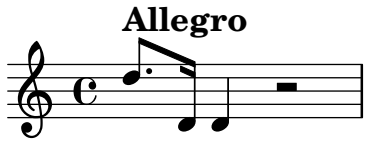
```
c1^\markup { \center-column { a bbb c } }
c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { bbb c } } }
c1^\markup { \center-column { a "bbb c" } }
```



Los marcados se pueden almacenar dentro de variables. Estas variables se pueden adjuntar directamente a las notas:

```
allegro = \markup { \bold \large Allegro }

{
  d''8.^\allegro
  d'16 d'4 r2
}
```



Se puede encontrar una lista exhaustiva de las instrucciones específicas de `\markup` en ⟨undefined⟩ [Text markup commands], página ⟨undefined⟩.

Véase también

Referencia de la notación: ⟨undefined⟩ [Text markup commands], página ⟨undefined⟩.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Archivos de inicio: ‘`scm/markup.scm`’.
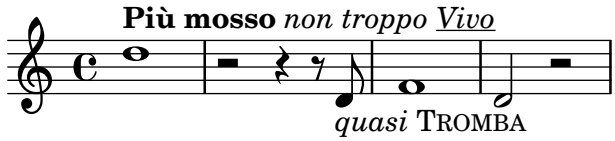
Advertencias y problemas conocidos

Los errores de sintaxis para el modo de marcado a menudo producen confusión.

Seleccionar la tipografía y su tamaño

Está contemplado de forma básica el cambio de la fuente tipográfica en el modo de marcado:

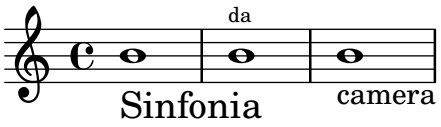
```
d1^\markup {
  \bold { Più mosso }
  \italic { non troppo \underline Vivo }
}
r2 r4 r8
d,_\markup { \italic quasi \smallCaps Tromba }
f1 d2 r
```



El tamaño de la fuente tipográfica se puede alterar en relación al tamaño global del pentagrama, de una serie de formas como se ve a continuación.

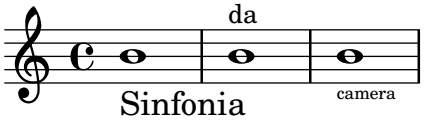
Se puede fijar a un tamaño predefinido,

```
b1_\markup { \huge Sinfonia }
b1^\markup { \teeny da }
b1-\markup { \normalsize camera }
```



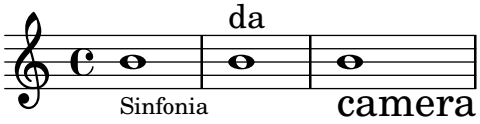
Se puede establecer a un tamaño relativo al valor anterior,

```
b1_\markup { \larger Sinfonia }
b1^\markup { \smaller da }
b1-\markup { \magnify #0.6  camera }
```



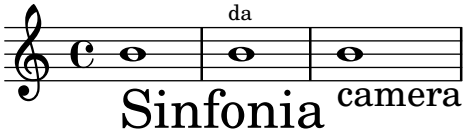
Se puede aumentar o disminuir de forma relativa al valor fijado por el tamaño global del pentagrama,

```
b1_\markup { \fontsize #-2 Sinfonia }
b1^\markup { \fontsize #1 da }
b1-\markup { \fontsize #3 camera }
```



También se puede establecer a un tamaño de puntos fijo, independientemente del tamaño global del pentagrama,

```
b1_\markup { \abs-fontsize #20 Sinfonia }
b1^\markup { \abs-fontsize #8 da }
b1-\markup { \abs-fontsize #14 camera }
```



El texto se puede imprimir como subíndice o como superíndice. De forma predeterminada se imprimen en un tamaño menor, pero también se puede usar un tamaño normal:

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement
      \sub { (part two) }  }
  }
}
```

1st movement
1st movement (part two)

El modo de marcado ofrece una manera fácil de elegir familias de tipografía alternativas. A no ser que se especifique de otro modo, se selecciona automáticamente la fuente predeterminada con serifa, de tipo romano: en la última línea del ejemplo siguiente, no hay diferencia entre la primera y la segunda palabra.

```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \roman Valentine and Proteus. }
  }
}
```

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Algunas de estas familias, usadas para elementos específicos como números o matices, no ofrecen todos los caracteres, como se explica en [Indicaciones dinámicas contemporáneas], página 125 y [Marcas de repetición manual], página 152.

Si se usan dentro de una palabra, algunas instrucciones de cambio de tipografía o de formateo pueden producir un espacio vacío no deseado. Esto se puede solucionar concatenando en uno solo los distintos elementos de texto:

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```

1st movement
p, *con dolce espressione*

Se puede encontrar una lista exhaustiva de instrucciones de cambio y utilización personalizada de las fuentes tipográficas en Sección A.10.1 [Font], página 671.

También es posible definir conjuntos personalizados de fuentes, tal y como se explica en Sección 1.8.3 [Tipografías], página 246.

Instrucciones predefinidas

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

Véase también

Referencia de la notación: Sección A.10.1 [Font], página 671, [Indicaciones dinámicas contemporáneas], página 125, [Marcas de repetición manual], página 152, Sección 1.8.3 [Tipografías], página 246.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

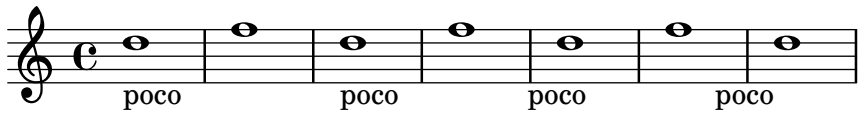
La utilización de las instrucciones de tamaño de texto `\teeny`, `\tiny`, `\small`, `\normalsize`, `\large` y `\huge` conducen a un espaciado de las líneas inconsistente comparado con el uso de `\fontsize`.

Alineación de texto

Esta subsección trata sobre cómo colocar texto en el modo de marcado. Los objetos de marcado también se pueden mover como un todo, usando la sintaxis que se describe en Sección “Mover objetos” in *Manual de Aprendizaje*.

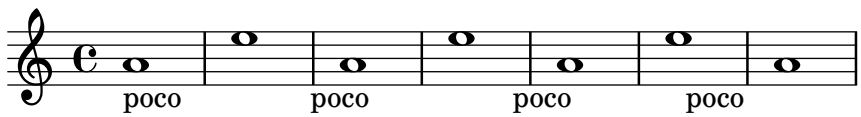
Los objetos de marcado se pueden alinear de distintas maneras. De forma predeterminada, una indicación de texto se alinea sobre el borde izquierdo: en el ejemplo siguiente, no existe diferencia entre los marcados primero y segundo.

```
d1-\markup { poco }
f
d-\markup { \left-align poco }
f
d-\markup { \center-align { poco } }
f
d-\markup { \right-align poco }
```



Se puede realizar un ajuste fino de la alineación horizontal usando un valor numérico:

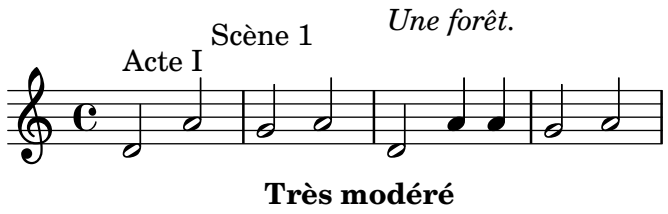
```
a1-\markup { \halign #-1 poco }
e'
a,-\markup { \halign #0 poco }
e'
a,-\markup { \halign #0.5 poco }
e'
a,-\markup { \halign #2 poco }
```



Ciertos objetos pueden poseer sus propios procedimientos de alineación, y por tanto no resultan afectados por estas instrucciones. Es posible mover estos objetos de marcado como un todo, como se muestra por ejemplo en [Indicaciones de texto], página 229.

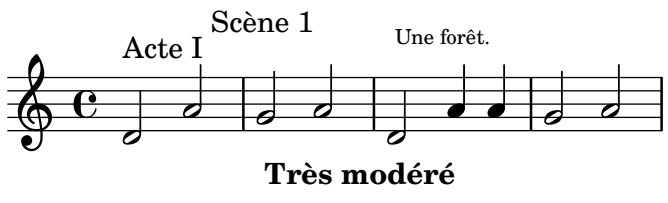
La alineación vertical es un poco más compleja. Como se ha dicho más arriba, los objetos de marcado se pueden mover como un todo; sin embargo, también es posible mover elementos específicos dentro de un bloque de marcado. En este caso, el elemento a mover se debe preceder de un *punto de anclaje*, que puede ser otro elemento de marcado o un objeto invisible. El ejemplo siguiente ejemplifica estas dos posibilidades; el último marcado en este ejemplo no tiene punto de anclaje y por ello no resulta movido.

```
d2^\markup {
  Acte I
  \raise #2 { Scène 1 }
}
a'
g_\markup {
  \null
  \lower #4 \bold { Très modéré }
}
a
d,^\markup {
  \raise #4 \italic { Une forêt. }
}
a'4 a g2 a
```



Algunas instrucciones pueden afectar tanto a la alineación horizontal como a la vertical de los objetos de texto en el modo de marcado. Cualquier objeto afectado por estas instrucciones se deben preceder de un punto de anclaje:

```
d2^\markup {
  Acte I
  \translate #'(-1 . 2) "Scène 1"
}
a'
g_\markup {
  \null
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,^\markup {
  \null
  \translate-scaled #'(-1 . 2) \teeny "Une forêt."
}
a'4 a g2 a
```



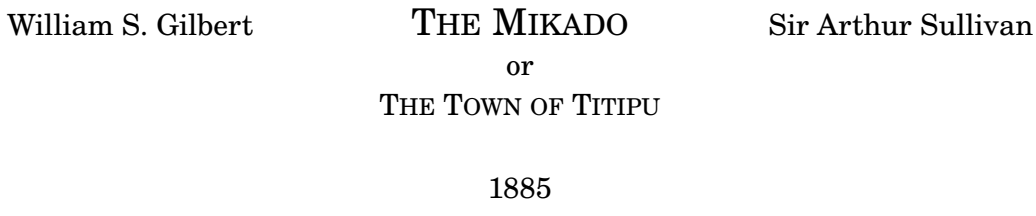
Un objeto de marcado puede incluir varias líneas de texto. En el ejemplo siguiente, cada elemento o expresión se sitúa en su propia línea, ya sea alineada por la izquierda o centrada:

```
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}
```



De forma similar, una lista de elementos o expresiones se puede repartir de forma que rellene por completo el ancho de la línea horizontal (si hay un solo elemento, se centra en el papel). A su vez, estas expresiones pueden incluir texto de varias líneas o cualquier otra expresión de marcado:

```
\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
\markup {
  \fill-line { 1885 }
}
```



Las indicaciones de texto largas se pueden también ajustar automáticamente según un ancho de línea dado. Estarán alineados por la izquierda o justificados, como se muestra en el ejemplo siguiente.

```
\markup {
  \column {
    \line  \smallCaps { La vida breve }
```

```
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada.  Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada.  Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}
```

La vida breve

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Hay una lista exhaustiva de instrucciones de alineación de texto en Sección A.10.2 [Align], página 680.

Véase también

Manual de aprendizaje: Sección “Mover objetos” in *Manual de Aprendizaje*.

Referencia de la notación: Sección A.10.2 [Align], página 680, [Indicaciones de texto], página 229.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

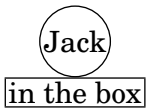
Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Notación gráfica dentro de elementos de marcado

Se puede añadir diversos objetos gráficos a una partitura, utilizando instrucciones de marcado.

Ciertas instrucciones de marcado permiten la decoración de elementos de texto con gráficos, como se muestra en el ejemplo siguiente.

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```



Erik Satie [1866 - 1925]

Prelude

Algunas instrucciones pueden requerir un aumento del relleno alrededor del texto: esto se puede conseguir con algunas instrucciones de marcado descritas exhaustivamente en Sección A.10.2 [Align], página 680.

```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
      \bold "Largo to Presto"
  }
  \pad-around #3
      "String quartet keeps very even time,
Flute quartet keeps very uneven time."
}
```

Charles Ives (1874 - 1954)

THE UNANSWERED QUESTION

A Cosmic Landscape

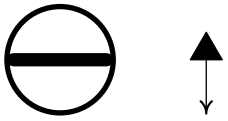
Largo to Presto

String quartet keeps very even time, Flute quartet keeps very uneven time.

Se pueden imprimir otros símbolos o elementos gráficos sin que se requiera ningún otro texto. De igual manera que en el caso de las expresiones de marcado, dichos objetos se pueden combinar:

```
\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

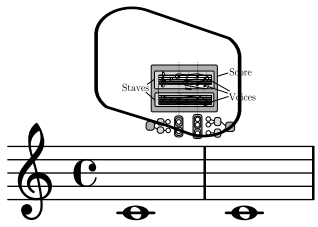
  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}
```



Entre las funcionalidades gráficas avanzadas se encuentran la posibilidad de incluir archivos de imagen externos convertidos al formato de PostScript Encapsulado (*eps*), y la inclusión directa de gráficos dentro del archivo de entrada, usando código PostScript nativo. En tal caso puede ser de utilidad la especificación explícita del tamaño del dibujo, como se ejemplifica a continuación:

```
c1^\markup {
  \combine
    \epsfile #X #10 #"./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript #"
      -2 3 translate
      2.7 2 scale
      newpath
      2 -1 moveto
      4 -2 4 1 1 arct
      4 2 3 3 1 arct
      0 4 0 3 1 arct
      0 0 1 -1 1 arct
      closepath
      stroke"
  }
```

```
c
```



Hay una lista exhaustiva de instrucciones específicas de gráficos en Sección A.10.3 [Graphic], página 695.

Véase también

Referencia de la notación: Sección A.10.3 [Graphic], página 695, Sección 1.7 [Anotaciones editoriales], página 213.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’, ‘`scm/stencil.scm`’.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

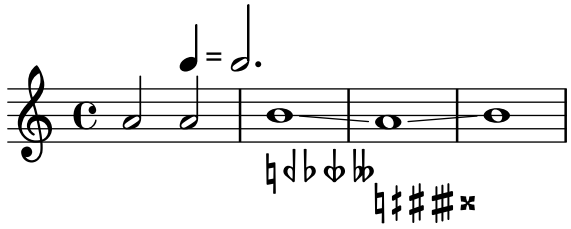
Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Notación musical dentro de elementos de marcado

Se pueden añadir a la partitura diversos elementos de notación musical, dentro de un objeto de marcado.

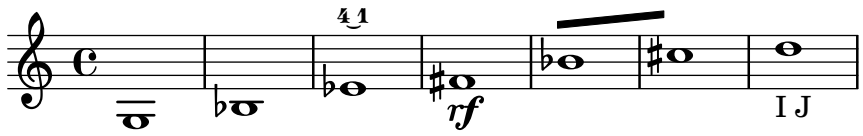
Las notas y las alteraciones se pueden escribir utilizando instrucciones de marcado:

```
a2 a^\markup {
  \note #"4" #1
  =
  \note-by-number #1 #1 #1.5
}
b1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b
```



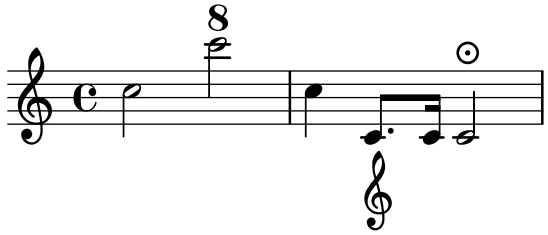
Otros objetos de notación se pueden también imprimir en el modo de marcado:

```
g1 bes
ees\finger \markup \tied-lyric #"4~1"
fis_\markup { \dynamic rf }
bes^\markup {
  \beam #8 #0.1 #0.5
}
cis
d-\markup {
  \markalphabet #8
  \markletter #8
}
```



De forma más general, cualquier símbolo musical disponible se puede incluir por separado dentro de un objeto de marcado, como se ejemplifica a continuación; hay una lista exhaustiva de estos símbolos y sus nombres en Sección A.8 [La tipografía Feta], página 650.

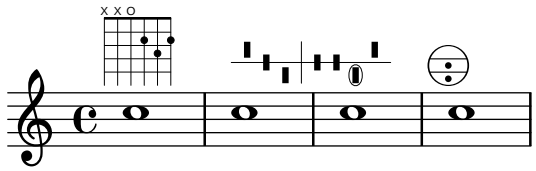
```
c2
c'^\markup { \musicglyph #"eight" }
c,4
c,8._\markup { \musicglyph #"clefs.G_change" }
c16
c2^\markup { \musicglyph #"timesig.neomensural94" }
```



Otra forma de imprimir glifos que no son de texto se encuentra descrita en [Explicación de las fuentes tipográficas], página 246. Tiene la utilidad de imprimir llaves de distintos tamaños.

El modo de marcado también contempla diagramas para instrumentos específicos:
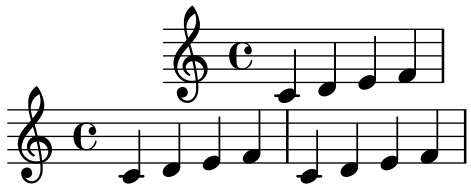
```
c1^\markup {
  \fret-diagram-terse #"x;x;o;2;3;2;"
}
c^\markup {
  \harp-pedal #"^-v|--ov^"
}
c
c^\markup {
  \combine
    \musicglyph #"accordion.discant"
    \combine
      \raise #0.5 \musicglyph #"accordion.dot"
      \raise #1.5 \musicglyph #"accordion.dot"
}
```



Dichos diagramas se encuentran documentados en Sección A.10.5 [Instrument Specific Markup], página 708.

Incluso una partitura completa se puede incluir dentro de un objeto de marcado. En tal caso, el bloque anidado `\score` debe contener un bloque `\layout`, como se muestra aquí:

```
c4 d^\markup {
  \score {
    \relative c' { c4 d e f }
    \layout { }
  }
}
e f |
c d e f
```



Hay una lista exhaustiva de instrucciones relacionadas con la notación musical en Sección A.10.4 [Music], página 702.

Véase también

Referencia de la notación: Sección A.10.4 [Music], página 702, Sección A.8 [La tipografía Feta], página 650, [Explicación de las fuentes tipográficas], página 246.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’, ‘`scm/fret-diagrams.scm`’, ‘`scm/harp-pedals.scm`’.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Elementos de marcado de varias páginas

Aunque los objetos de marcado estándar no se pueden dividir, una sintaxis específica hace posible la introducción de líneas de texto que pueden abarcar varias páginas:

```
\markuplist {
  \justified-lines {
    Un texto muy largo de líneas justificadas.
    ...
  }
  \wordwrap-lines {
    Otro párrafo muy largo.
    ...
  }
  ...
}
```

Un texto muy largo de líneas justificadas. ...

Otro párrafo muy largo. ...

...

Esta sintaxis acepta una lista de elementos de marcado, que pueden ser

- el resultado de una instrucción de lista de marcado,
- una lista de marcados,
- o una lista de listas de marcado.

Hay una lista exhaustiva de las instrucciones de lista de marcado en ⟨undefined⟩ [Text markup list commands], página ⟨undefined⟩.

Véase también

Referencia de la notación: ⟨undefined⟩ [Text markup list commands], página ⟨undefined⟩.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Manual de extensión: Sección “New markup list command definition” in *Extender*.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Instrucciones predefinidas

`\markuplist`.

1.8.3 Tipografías

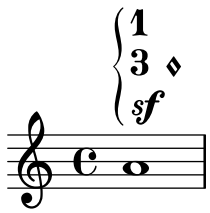
Esta sección presenta la forma en que se manejan las fuentes tipográficas, y cómo se pueden cambiar en las partituras.

Explicación de las fuentes tipográficas

Las fuentes tipográficas se manejan a través de distintas bibliotecas. FontConfig se utiliza para detectar las tipografías disponibles en el sistema; las tipografías seleccionadas se trazan utilizando Pango.

Las tipografías de notación musical se pueden describir como un conjunto de glifos específicos, ordenados en varias familias. La siguiente sintaxis permite usar directamente varias tipografías `feta` de LilyPond distintas a las de texto en el modo de marcado:

```
a1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup #"brace120"
    \override #'(font-encoding . fetaText)
    \column { 1 3 sf }
    \override #'(font-encoding . fetaMusic)
    \lookup #"noteheads.s0petrucci"
  }
}
```



Sin embargo, todos estos glifos excepto las llaves de varios tamaños que están contenidas en la fuente tipográfica `fetaBraces` están disponibles utilizando la sintaxis más sencilla que se describe en [Notación musical dentro de elementos de marcado], página 243.
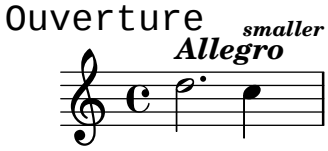
Cuando se usan los glifos de la fuente de llaves `fetaBraces`, el tamaño de la llave se especifica por la parte numérica del nombre del glifo, en unidades arbitrarias. Se puede especificar cualquier entero de `0` a `575` inclusive, siendo `0` el que produce la llave más pequeña. El valor óptimo se debe determinar por ensayo y error. Todos estos glifos son llaves izquierdas; las llaves derechas se pueden obtener mediante rotación, véase Sección 5.4.8 [Rotación de objetos], página 615.

Están disponibles tres familias de tipografías de texto: la tipografía *romana* (con serifa), que de forma predeterminada es New Century Schoolbook, la fuente *sans* y la fuente monoespaciada *de máquina de escribir*. Estas dos últimas familias vienen determinadas por la instalación de Pango.

Nota: No existen fuentes tipográficas predeterminadas asociadas con las familias de fuente *sans* y *typewriter*. Un archivo de entrada que especifica alguna de éstas puede dar lugar a salidas diferentes en distintos sistemas. Para asegurar un resultado consistente entre distintas plataformas, se deben especificar las tipografías por su nombre, y dichas fuentes deben estar disponibles en cualquier sistema que vaya a procesar el documento. Véase [Fuentes de un solo elemento], página 248 y [Fuentes tipográficas del documento completo], página 248.

Cada familia puede incluir distintas formas y series. El ejemplo siguiente muestra la posibilidad de seleccionar familias, formas, series y tamaños alternativos. El valor que se da a `font-size` es el cambio requerido a partir del tamaño predeterminado.

```
\override Score.RehearsalMark.font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript.font-shape = #'italic
\override Voice.TextScript.font-series = #'bold
d2.^\markup "Allegro"
\override Voice.TextScript.font-size = #-3
c4^smaller
```



Se puede usar una sintaxis similar en el modo de marcado, aunque en este caso es preferible usar la sintaxis sencilla que se explica en [Seleccionar la tipografía y su tamaño], página 234:

```
\markup {
  \column {
    \line {
      \override #'(font-shape . italic)
      \override #'(font-size . 4)
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter)
      {
        \override #'(font-series . bold)
        re
        di
```

```
      }
      \override #'(font-family . sans)
      Creta
    }
  }
}
```

Idomeneo,
re di Creta

Aunque es fácil cambiar entre las fuentes tipográficas preconfiguradas, también es posible usar otras fuentes, como se explica en las siguientes secciones: [Fuentes de un solo elemento], página 248 y [Fuentes tipográficas del documento completo], página 248.

Véase también
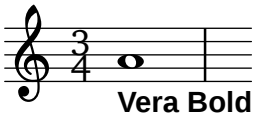
Referencia de la notación: Sección A.8 [La tipografía Feta], página 650, [Notación musical dentro de elementos de marcado], página 243, Sección 5.4.8 [Rotación de objetos], página 615, [Seleccionar la tipografía y su tamaño], página 234, Sección A.10.1 [Font], página 671.

Fuentes de un solo elemento

Se puede usar en una partitura cualquier fuente tipográfica que esté instalada en el sistema operativo y reconocida por parte de FontConfig, usando la siguiente sintaxis:

```
\override Staff.TimeSignature.font-name = #"Bitstream Charter"
\override Staff.TimeSignature.font-size = #2
\time 3/4

a1_\markup {
  \override #'(font-name . "Vera Bold")
    { Vera Bold }
}
```



La instrucción siguiente presenta una lista de todas las tipografías disponibles en el sistema operativo:

```
lilypond -dshow-available-fonts x
```

Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 246, [Fuentes tipográficas del documento completo], página 248.

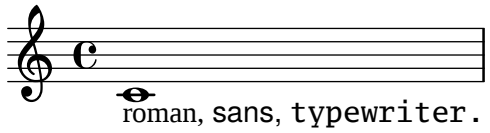
Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Fuentes tipográficas del documento completo

Se pueden cambiar las fuentes tipográficas que se usan como tipografías predeterminadas en las familias *romana*, *sans-serif* y *monoespaciada*, especificándolas en este orden como se ve en el ejemplo de abajo, que escala las tipografías automáticamente con el valor fijado para el tamaño global del pentagrama. Para leer más explicaciones sobre las fuentes tipográficas, consulte [Explicación de las fuentes tipográficas], página 246.

```
\paper  {
  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                          (/ staff-height pt 20)))
}

\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



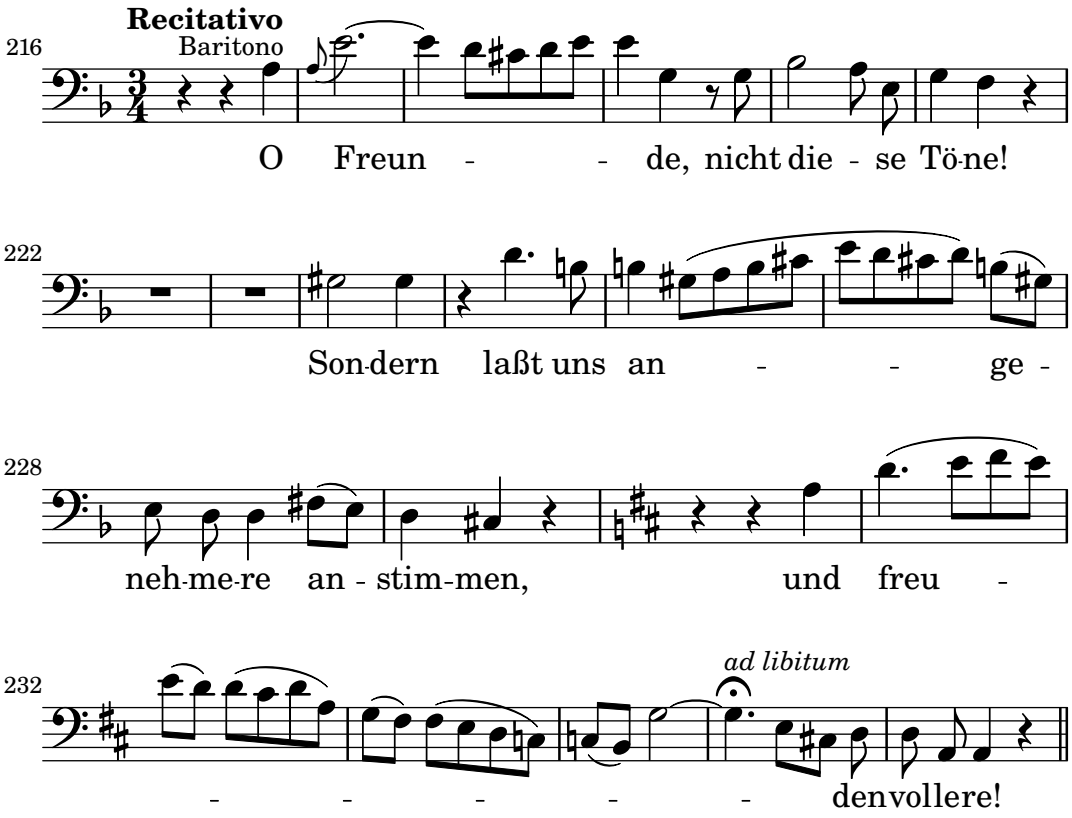
Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 246, [Fuentes de un solo elemento], página 248, [Seleccionar la tipografía y su tamaño], página 234, Sección A.10.1 [Font], página 671.

2 Notación especializada

Este capítulo explica cómo crear notación musical para instrumentos de un tipo específico o dentro de estilos específicos.

2.1 Música vocal



Esta sección explica cómo tipografiar música vocal, y cómo asegurarse de que la letra se alinea con las notas de su melodía correspondiente.

2.1.1 Notación común para música vocal

Esta sección se ocupa de las cuestiones comunes a la mayoría de los tipos de música vocal.

Referencias para música vocal

Esta sección indica dónde encontrar los detalles de las cuestiones de notación que pueden surgir en cualquier tipo de música vocal.

- Casi todos los estilos de música vocal utilizan texto escrito como letra. Hay una introducción a esta notación en Sección “Elaborar canciones sencillas” in *Manual de Aprendizaje*.
- La música vocal probablemente requiere el uso del modo de marcado o `markup`, ya sea para la letra o para otros elementos de texto (nombres de los personajes, etc.). Esta sintaxis está descrita en [Introducción al marcado de texto], página 233.
- Los *ambitus* o indicaciones de tesitura vocal se pueden añadir al principio de los pentagramas vocales, como se explica en [Tesitura], página 33.
- De forma predeterminada, las indicaciones dinámicas se sitúan debajo del pentagrama, pero en la música coral se suelen disponer por encima del pentagrama para evitar la letra, como se explica en [Disposiciones de la partitura para música coral], página 291.

Véase también

Glosario musical: Sección “ambitus” in *Glosario Musical*.

Manual de aprendizaje: Sección “Elaborar canciones sencillas” in *Manual de Aprendizaje*.

Referencia de la notación: [Introducción al marcado de texto], página 233, [Tesitura], página 33, [Disposiciones de la partitura para música coral], página 291.

Fragmentos de código: Sección “Vocal music” in *Fragmentos de código*.

Introducir la letra

La letra de las canciones se introduce en un modo de entrada especial que se inicia mediante la palabra clave `\lyricmode`, o bien mediante `\addlyrics` ó `\lyricsto`. En este modo especial de entrada, el carácter `d` no se analiza como una nota, sino más bien como una sílaba de una sola letra. Dicho de otra forma, las sílabas se introducen como las notas, pero la altura de las notas se sustituye por texto.

Por ejemplo:

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas: mediante la indicación de la duración de cada sílaba explícitamente, como en el ejemplo anterior, o dejando que la letra se alinee automáticamente con una melodía u otra voz de música, utilizando `\addlyrics` o `\lyricsto`. El primer método se describe a continuación bajo el epígrafe [Duración manual de las sílabas], página 257. El segundo método se describe en [Duración automática de las sílabas], página 254.

Una palabra o sílaba de la letra comienza con un carácter alfabético (y algunos otros caracteres, véase más abajo) y termina con un espacio o un dígito. El resto de los caracteres dentro de la sílaba pueden ser cualesquier caracteres que no sean dígitos o espacios.

A causa de que cualquier carácter que no es un dígito o un espacio se considera parte de la sílaba, una palabra es válida incluso si termina con `}`, lo que a menudo lleva al siguiente fallo:
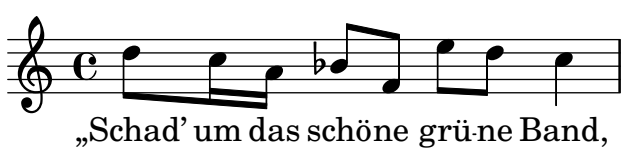
```
\lyricmode { la la la}
```

En este ejemplo, el símbolo `}` se encuentra incluido dentro de la sílaba final, por tanto la llave de apertura no se compensa con la correspondiente llave de cierre y el archivo de entrada probablemente no se podrá procesar. En vez de ello, las llaves siempre deben ir rodeadas por espacios:

```
\lyricmode { la la la }
```
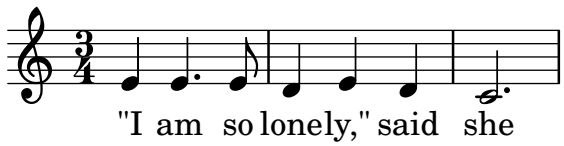
Los signos de puntuación, las letras con caracteres acentuados o de cualquier idioma distinto del inglés, o los caracteres especiales (como el símbolo del corazón o las comillas inclinadas), se pueden insertar directamente en el archivo de entrada, siempre que éste se grabe con la codificación UTF-8. Para más información, consulte Sección 3.3.3 [Caracteres especiales], página 497.

```
\relative c'' { d8 c16 a bes8 f e' d c4 }
\addlyrics { „Schad’ um das schö -- ne grü -- ne Band, }
```



Se pueden usar las comillas normales dentro de la letra, pero deben ir precedidas por el carácter de barra invertida y la sílaba en su conjunto se debe encerrar en otro par de comillas. Por ejemplo,

```
\relative c' { \time 3/4 e4 e4. e8 d4 e d c2. }
\addlyrics { "\"I" am so lone -- "ly,\"" said she }
```



La definición completa del comienzo de una palabra en el modo Lyrics (letra) es algo más compleja. Una palabra en el modo Lyrics comienza por: un carácter alfabético, `_`, `?`, `!`, `:`, `'`, los caracteres de control desde `^A` hasta `^F`, desde `^Q` hasta `^W`, `^Y`, `^^`, cualquier carácter de 8 bits con código ASCII por encima del 127, o una combinación de dos caracteres consistente en la combinación de una barra invertida seguida por `` ` ``, `'`, `"` ó `^`.

Un enorme control sobre la apariencia de la letra proviene del uso de elementos de marcado `\markup` dentro de la propia letra. Para ver una explicación de muchas de las opciones, consulte Sección 1.8.2 [Formatear el texto], página 233.

Fragmentos de código seleccionados

Dar formato a sílabas de la letra

Es posible usar el modo de marcado para dar formato a sílabas individuales dentro de la letra.

```
mel = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

<<
  \new Voice = melody \mel
  \new Lyrics \lyricsto melody \lyr
>>
```



Véase también

Manual de aprendizaje: Sección “Canciones” in *Manual de Aprendizaje*.

Referencia de la notación: [Duración automática de las sílabas], página 254, Sección 1.8.3 [Tipografías], página 246, Sección 1.8.2 [Formatear el texto], página 233, Sección 5.4.1 [Modos de entrada], página 599, [Duración manual de las sílabas], página 257, Sección 3.3.3 [Caracteres especiales], página 497.

Referencia de funcionamiento interno: Sección “LyricText” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Alineación de la letra a una melodía

La letra se imprime mediante su interpretación dentro del contexto llamado `Lyrics`, véase Sección 5.1.1 [Explicación de los contextos], página 567.

`\new Lyrics \lyricmode { ... }`

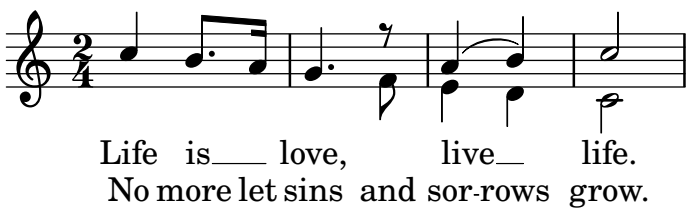
La letra se puede alinear con la melodía de dos maneras principales:

- Se puede alinear la letra automáticamente, tomándose las duraciones de las sílabas de otra voz de música, o (en circunstancias especiales) una melodía asociada, usando `\addlyrics`, `\lyricsto`, o estableciendo el valor de la propiedad `associatedVoice`. Para ver más detalles, consulte [Duración automática de las sílabas], página 254.

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative c'' {
      \voiceOne
      c4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative c' {
      \voiceTwo
       s2 s4. f8 e4 d c2
    }
  >>

% takes durations and alignment from notes in "one"
  \new Lyrics \lyricsto "one" {
    Life is __ _ love, live __ life.
  }

% takes durations and alignment from notes in "one" initially
% then switches to "two"
  \new Lyrics \lyricsto "one" {
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



La primera línea de letra muestra la forma normal de introducir la letra.
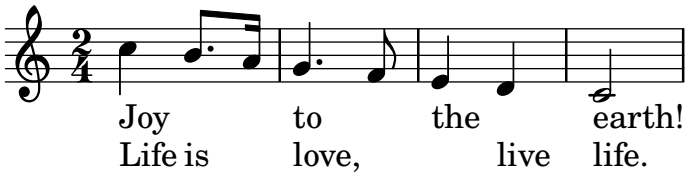
La segunda línea de texto muestra cómo se puede cambiar la voz de la que se toman las duraciones para las sílabas. Esto es útil si el texto de las distintas estrofas corresponde a las mismas notas de formas diferentes, y todas las duraciones están disponibles dentro de contextos de voz. Para ver más detalles, consulte Sección 2.1.3 [Versos], página 282.

- El texto se puede alinear independientemente de la duración de cualquier nota si las duraciones de las sílabas se especifican explícitamente, y se escriben con `\lyricmode`.

```
<<
  \new Voice = "one" \relative c'' {
    \time 2/4
    c4 b8. a16 g4. f8 e4 d c2
  }

% uses previous explicit duration of 2;
  \new Lyrics \lyricmode {
    Joy to the earth!
  }

% explicit durations, set to a different rhythm
  \new Lyrics \lyricmode {
    Life4 is love,2. live4 life.2
  }
>>
```



El primer verso no está alineado con las notas porque no se especificaron las duraciones, y se utiliza el valor previo de 2 (blanca) para todas las sílabas.

El segundo verso muestra cómo las palabras se pueden alinear de forma independiente de las notas. Esto es útil si el texto de los distintos versos se corresponde con las notas de maneras diferentes, pero las duraciones requeridas no están disponibles en un contexto de música. Para ver más detalles, consulte [Duración manual de las sílabas], página 257. Esta técnica también es útil cuando se quiere preparar un diálogo encima de la música; para ver ejemplos que lo muestran, consulte [Diálogos encima de la música], página 300.

Cuando se escriben de esta forma, las sílabas del texto se alinean por la izquierda con las notas, de forma predeterminada, pero se pueden centrar sobre las notas de una melodía especificando una voz asociada, si existe. Para ver más detalles, consulte [Duración manual de las sílabas], página 257.

Véase también

Manual de aprendizaje: Sección “Alineación de la letra a una melodía” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.1 [Explicación de los contextos], página 567, [Duración automática de las sílabas], página 254. Sección 2.1.3 [Versos], página 282, [Duración manual de las sílabas], página 257, [Diálogos encima de la música], página 300, [Duración manual de las sílabas], página 257.

Referencia de funcionamiento interno: Sección “Lyrics” in *Referencia de Funcionamiento Interno*.

Duración automática de las sílabas

Las sílabas de la letra se pueden alinear automáticamente con las notas de una melodía de tres formas:

- especificando por su nombre el contexto Voice que contiene la melodía, con `\lyricsto`,

- introduciendo la letra con `\addlyrics` y colocándola inmediatamente después del contexto de voz que contiene la melodía,
- estableciendo un valor para la propiedad `associatedVoice`, la alineación de la letra se puede mover a un contexto de voz distinto, especificado por su nombre, en cualquier momento musical.

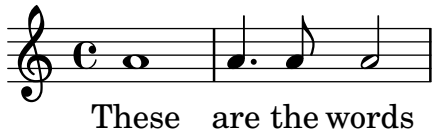
En los tres métodos se pueden trazar guiones de separación entre las sílabas de una palabra y líneas extensoras después del final de una palabra. Para ver más detalles, véase [Líneas de extensión y guiones], página 262.

El contexto `Voice` que contiene la melodía con que se está alineando la letra, no debe haber "muerto", o se perderá la letra a partir de este punto. Esto puede ocurrir si existen períodos en que dicha voz no tiene nada que hacer. Para ver métodos para mantener vivos los contextos, consulte Sección 5.1.3 [Mantener vivos los contextos], página 573.

Uso de `\lyricsto`

Las sílabas de la letra se pueden alinear automáticamente bajo las notas de una melodía especificando por su nombre el contexto de voz que contiene la melodía, con `\lyricsto`:

```
<<
  \new Voice = "melody" {
    a1 a4. a8 a2
  }
  \new Lyrics \lyricsto "melody" {
    These are the words
  }
>>
```



Esto alinea las sílabas con las notas del contexto `Voice` nombrado, que debe existir previamente. Por tanto, normalmente se especifica primero el contexto `Voice` seguido del contexto `Lyrics`. La letra en sí sigue a la instrucción `\lyricsto`. La instrucción `\lyricsto` invoca automáticamente el modo de letra, por lo que la palabra clave `\lyricmode` se puede omitir. De forma predeterminada, la letra se coloca por debajo de las notas. Para otras colocaciones, consulte [Posicionamiento vertical de la letra], página 264.

Uso de `\addlyrics`

La instrucción `\addlyrics` es realmente tan sólo una forma cómoda de escribir una estructura de LilyPond más complicada que establece la letra.

```
{ MÚSICA }
\addlyrics { LETRA }
```
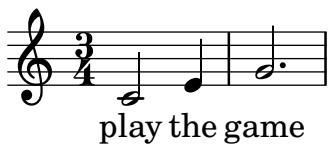
es lo mismo que

```
\new Voice = "blabla" { MÚSICA }
\new Lyrics \lyricsto "blabla" { LETRA }
```

He aquí un ejemplo:
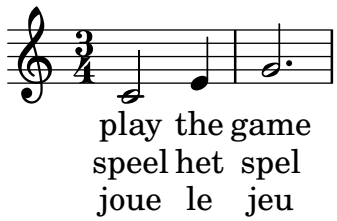
```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
```

```
}
```



Se pueden añadir más versos poniendo más secciones `\addlyrics`:

```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```



La instrucción `\addlyrics` no es capaz de manejar situaciones de polifonía. Asimismo, no puede usarse para asociar letra a un contexto de tablatura `TabVoice`. Para estos casos debería usar `\lyricsto`.

Uso de `associatedVoice`

Se puede cambiar la melodía a la que se alinea la letra mediante el establecimiento de la propiedad `associatedVoice`,
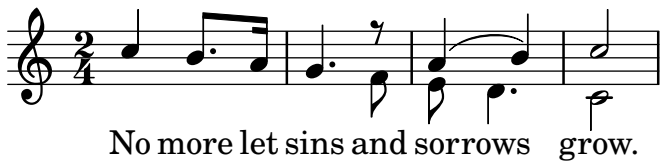
```
\set associatedVoice = #"lala"
```

El valor de la propiedad (aquí: `"lala"`) debe ser el nombre de un contexto `Voice`. Por razones técnicas, la instrucción `\set` se debe escribir una sílaba antes de aquella a la que se quiere aplicar el cambio de voz.

He aquí un ejemplo que muestra su uso:

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative c'' {
      \voiceOne
      c4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative c' {
      \voiceTwo
       s2 s4. f8 e8 d4. c2
    }
  >>
% takes durations and alignment from notes in "one" initially
% then switches to "two"
  \new Lyrics \lyricsto "one" {
```

```
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



Véase también

Referencia de la notación: [Líneas de extensión y guiones], página 262, Sección 5.1.3 [Mantener vivos los contextos], página 573, [Posicionamiento vertical de la letra], página 264.

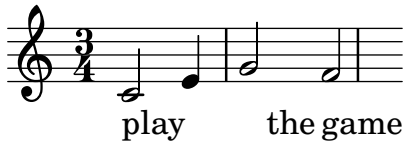
Duración manual de las sílabas

En ciertas músicas vocales complejas puede ser deseable colocar la letra de forma completamente independiente de las notas. En este caso, no use `\lyricsto` ni `\addlyrics` y no establezca un valor para `associatedVoice`. Las sílabas se escriben como notas – pero sustituyendo los nombres de las notas por texto – y la duración de cada sílaba se escribe explícitamente después de la sílaba.

De forma predeterminada, las sílabas se alinean por la izquierda con el momento musical correspondiente. Se pueden trazar líneas separadores entre las sílabas, como es usual, pero no se pueden trazar líneas extensoras cuando no hay una voz asociada.

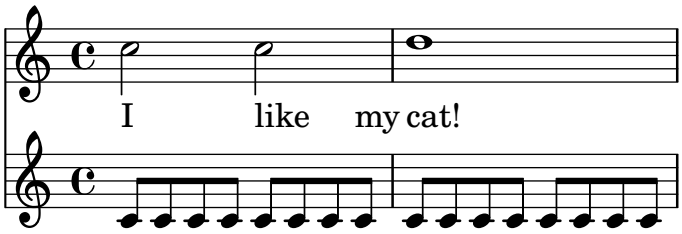
He aquí dos ejemplos:

```
<<
  \new Voice = "melody" {
    \time 3/4
    c2 e4 g2 f
  }
  \new Lyrics \lyricmode {
    play1 the4 game4
  }
>>
```




```
<<
  \new Staff {
    \relative c'' {
      c2 c2
      d1
    }
  }
  \new Lyrics {
    \lyricmode {
      I2 like4. my8 cat!1
```
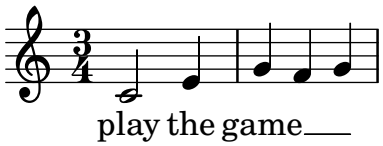
```
    }
  }
  \new Staff {
    \relative c' {
      c8 c c c c c c c
      c8 c c c c c c c
    }
  }
>>
```



Esta técnica es muy útil cuando se escriben diálogos encima de la música, véase [Diálogos encima de la música], página 300.

Para centrar las sílabas sobre las notas en sus momentos musicales correspondientes, ajuste `associatedVoice` al nombre del contexto de voz que contiene dichas notas. Cuando está establecido el valor de `associatedVoice`, se pueden usar tanto dobles guiones como dobles barras bajas para trazar correctamente guiones separadores y líneas extensoras bajos los melismas.

```
<<
  \new Voice = "melody" {
    \time 3/4
    c2 e4 g f g
  }
  \new Lyrics \lyricmode {
    \set associatedVoice = #"melody"
    play2 the4 game2. __
  }
>>
```



Véase también

Referencia de la notación: Sección 5.1.3 [Mantener vivos los contextos], página 573.

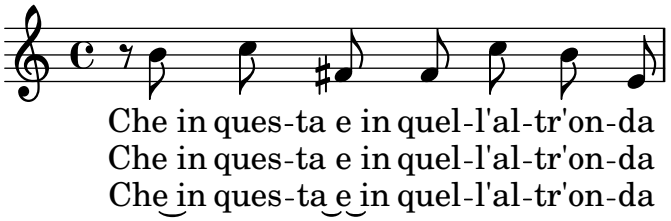
Referencia de funcionamiento interno: Sección “Lyrics” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Varias sílabas sobre una nota

Para asignar más de una sílaba a una única nota con espacios entre las sílabas, podemos encerrar la frase entre comillas o utilizar un carácter de guión bajo `_`. De forma alternativa, podemos usar el símbolo de tilde curva (`~`) para obtener una ligadura de texto.

```
{
  { \autoBeamOff
```

```
    r8 b c fis, fis c' b e,
  }
  \addlyrics
  {
    \override LyricHyphen.minimum-distance = #1.0 % Ensure hyphens are visible
    Che_in ques -- ta_e_in quel -- l'al -- tr'on -- da
  }
  \addlyrics { "Che in" ques -- "ta e in" quel -- l'al -- tr'on -- da }
  \addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da }
}
```



Véase también

Referencia de funcionamiento interno: Sección "LyricCombineMusic" in *Referencia de Funcionamiento Interno*.

Varias notas sobre una sílaba

A veces, y sobre todo en la música medieval y del Barroco, varias notas se cantan sobre una sílaba única; tales vocalizaciones reciben el nombre de melismas, o melismata (véase Sección "melisma" in *Glosario Musical*). La sílaba de un melisma se suele alinear por la izquierda con la primera nota del melisma.
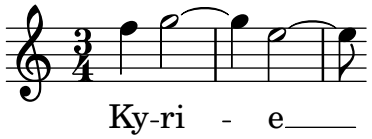
Cuando se produce un melisma sobre una sílaba distinta de la última de una palabra, dicha sílaba se suele unir a la siguiente con un guión separador. Esto se indica en el código de entrada escribiendo un doble guión, `--`, inmediatamente después de la sílaba.

De forma alternativa, cuando se produce un melisma sobre la última o sobre la única sílaba de una palabra, se suele trazar una línea extensora desde el final de la sílaba hasta la última nota del melisma. Esto se indica en el código de entrada escribiendo una doble barra baja o carácter de subrayado, `__`, inmediatamente después de la palabra.

Existen cinco formas de indicar los melismas:
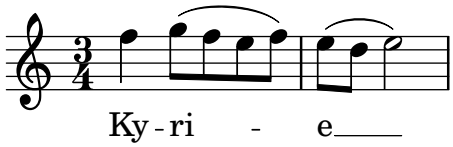
- Se crean melismas automáticamente sobre las notas unidas mediante ligadura de unión:

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g2 ~ |
    g4 e2 ~ |
    e8
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e __
  }
>>
```



- Se pueden crear melismas automáticamente a partir de la música escribiendo ligaduras de expresión sobre las notas de cada melisma. Ésta es la forma usual de escribir la letra:
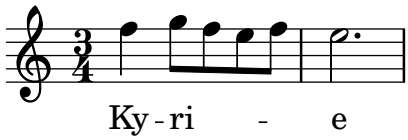
```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8 ( f e f )
    e8 ( d e2 )
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e __
  }
>>
```



Observe que las ligaduras de fraseo no afectan a la creación de melismas.

- Se considera que las notas forman un melisma cuando se unen manualmente mediante barra, siempre y cuando el barrado automático esté desactivado. Véase [Establecer el comportamiento de las barras automáticas], página 81.

```
<<
  \new Voice = "melody" {
    \time 3/4
    \autoBeamOff
    f4 g8[ f e f]
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```
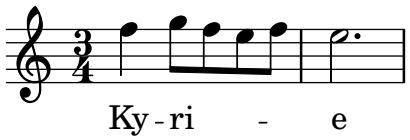


Claramente, esto no es adecuado para los melismas formados por notas de duración más larga que la corchea.

- Un grupo de notas sin ligadura de expresión se trata como un melisma si están comprendidas entre `\melisma` y `\melismaEnd`.
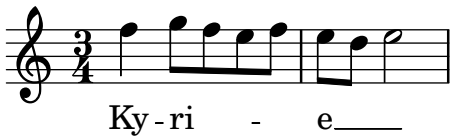
```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8
    \melisma
    f e f
```

```
    \melismaEnd
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```
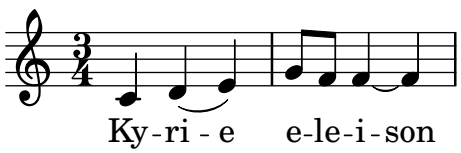


- Se puede definir un melisma enteramente dentro de la letra escribiendo un carácter de barra baja suelto, _, por cada nota adicional que se quiere añadir al melisma.

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8 f e f
    e8 d e2
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ _ _ e __ _ _
  }
>>
```



Es posible tener ligaduras de unión o de expresión y barras manuales en la melodía sin que indiquen melisma. Para hacerlo, ajuste el valor de `melismaBusyProperties`:

```
<<
  \new Voice = "melody" {
    \time 3/4
    \set melismaBusyProperties = #'()
    c4 d ( e )
    g8 [ f ] f4 ~ f
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e e -- le -- i -- son
  }
>>
```



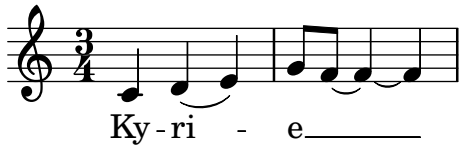
Se pueden usar otros valores para `melismaBusyProperties` si queremos incluir o excluir selectivamente las ligaduras de unión, ligaduras de expresión o barras de la detección automática

de los melismas; véase `melismaBusyProperties` en el apartado Sección “Tunable context properties” in *Referencia de Funcionamiento Interno*.

Como alternativa, si todas las indicaciones de melismas se van a ignorar, se puede establecer `ignoreMelismata` al valor verdadero; véase [Versos con ritmos distintos], página 284.

Si se requiere un melisma en el transcurso de un pasaje en el que `melismaBusyProperties` está activo, puede indicarse colocando una sola barra baja en la letra por cada nota que se debe incluir dentro del melisma:

```
<<
  \new Voice = "melody" {
    \time 3/4
    \set melismaBusyProperties = #'()
    c4 d ( e )
    g8 [ f ] ~ f4 ~ f
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ e __ _ _ _
  }
>>
```



Instrucciones predefinidas

`\autoBeamOff`, `\autoBeamOn`, `\melisma`, `\melismaEnd`.

Véase también

Glosario musical: Sección “melisma” in *Glosario Musical*.

Manual de aprendizaje: Sección “Alineación de la letra a una melodía” in *Manual de Aprendizaje*.

Referencia de la notación: [Alineación de la letra a una melodía], página 253, [Duración automática de las sílabas], página 254, [Establecer el comportamiento de las barras automáticas], página 81, [Versos con ritmos distintos], página 284.

Referencia de funcionamiento interno: Sección “Tunable context properties” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se crean automáticamente líneas extensoras bajo los melismas; se deben insertar manualmente con un doble guión bajo.

Líneas de extensión y guiones

Los melismas se indican mediante una línea horizontal centrada entre una sílaba y la siguiente. Esta línea recibe el nombre de línea de extensión, y se escribe como ‘ `__` ’ (fíjese en los espacios antes y después de los dos guiones bajos).

Nota: Los melismas se indican en la partitura con líneas extensoras, que se escriben con un doble guión bajo; pero también se pueden introducir melismas cortos mediante saltos de notas individuales, que se escriben como caracteres de guión bajo sueltos; de forma predeterminada éstos no producen la impresión de una línea de extensión.

Los guiones centrados se escriben como ' -- ' entre sílabas de una misma palabra (fíjese en los espacios antes y después de los dos guiones). El guión quedará centrado entre las sílabas, y su longitud se ajustará en función del espacio que exista entre ellas.

En la música grabada de modo muy apretado, se pueden quitar los guiones. Se puede controlar si esto ocurre o no, mediante las propiedades `minimum-distance` (distancia mínima entre las dos sílabas) y la `minimum-length` (umbral por debajo del cual se suprimen los guiones) de `LyricHyphen`.

Véase también
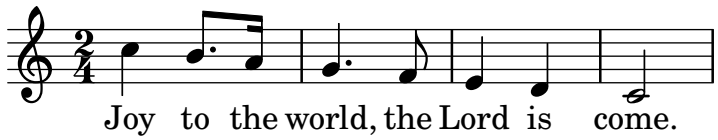
Referencia de funcionamiento interno: Sección "LyricExtender" in *Referencia de Funcionamiento Interno*, Sección "LyricHyphen" in *Referencia de Funcionamiento Interno*.

2.1.2 Técnicas específicas para la letra

Trabajar con letra y variables

Se pueden crear variables que contienen letra, pero la letra se debe introducir en el modo de letra:

```
musicOne = \relative c'' {
  c4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
  <<
    \new Voice = "one" {
      \time 2/4
      \musicOne
    }
    \new Lyrics \lyricsto "one" {
      \verseOne
    }
  >>
}
```
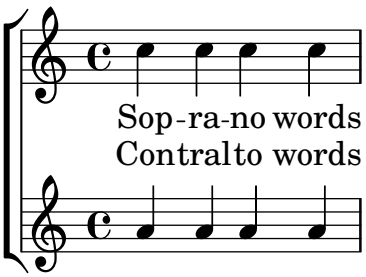


No es necesario escribir las duraciones si la variables se va a invocar con `\addlyrics` o con `\lyricsto`.

Para disposiciones distintas o más complejas, la mejor forma es definir en primer lugar las variables de la música y de la letra, después preparar una jerarquía de pentagramas y letras, omitiendo la letra en sí, y después poner la letra utilizando `\context` por debajo. Esto asegura que las voces referenciadas por `\lyricsto` siempre han sido definidas previamente. Por ejemplo:

```
sopranoMusic = \relative c'' { c4 c c c }
```

```
contraltoMusic = \relative c'' { a4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        \sopranoWords
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        \contraltoWords
      }
    }
  >>
}
```



Véase también

Referencia de la notación: [Posicionamiento vertical de la letra], página 264.

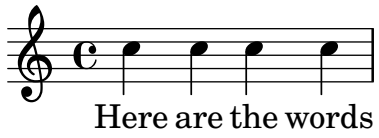
Referencia de funcionamiento interno: Sección “LyricCombineMusic” in *Referencia de Funcionamiento Interno*, Sección “Lyrics” in *Referencia de Funcionamiento Interno*.

Posicionamiento vertical de la letra

Dependiendo del tipo de música, la letra puede colocarse sobre el pentagrama, debajo del pentagrama, o entre dos pentagramas. Lo más fácil es situar la letra debajo de la pauta asociada, y se puede conseguir simplemente definiendo el contexto Lyrics por debajo del contexto Staff:

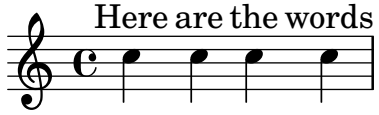
```
\score {
  <<
```

```
    \new Staff {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Se puede colocar la letra sobre el pentagrama usando uno de los dos métodos disponibles: el más sencillo (y preferible) es usar la misma sintaxis que se ha visto más arriba y especificar explícitamente la posición de la letra:

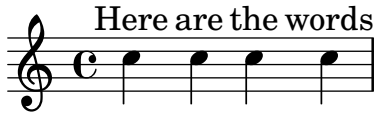
```
\score {
  <<
    \new Staff = "staff" {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics \with { alignAboveContext = "staff" } {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Como alternativa, se puede usar un proceso en dos pasos. En primer lugar se declara el contexto Lyrics (sin ningún contenido) antes de los contextos Staff y Voice, y después se escribe la instrucción `\lyricsto` después de la declaración de Voice a que hace referencia, usando `\context`, como se ve a continuación:

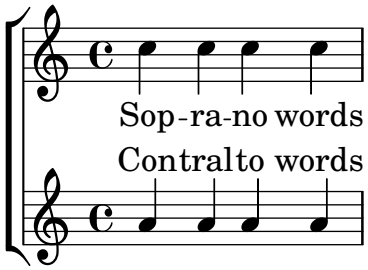
```
\score {
  <<
    \new Lyrics = "lyrics" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "melody" {
```

```
        \relative c'' { c4 c c c }
      }
    }
    \context Lyrics = "lyrics" {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Cuando hay dos voces sobre pentagramas separados, la letra se puede colocar entre ellos usando cualquiera de estos métodos. He aquí un ejemplo del segundo método:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "contraltos" {
        \relative c'' { a4 a a a }
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        Sop -- ra -- no words
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        Con -- tral -- to words
      }
    }
  >>
}
```



Es posible generar otras combinaciones de letra y pautas mediante la elaboración de los ejemplos anteriores, o examinando las plantillas del Manual de Aprendizaje, véase Sección “Plantillas de conjuntos vocales” in *Manual de Aprendizaje*.

Fragmentos de código seleccionados

*Conseguir el espaciado de la letra de la versión 2.12*en versiones más recientes

El motor de espaciado vertical cambió en la versión 2.14. Esto puede hacer que se altere el espaciado de la letra de las canciones. Es posible fijar propiedades para los contextos `Lyric` y `Staff` de forma que el motor de espaciado se comporte como lo hacía en la versión 2.12.

```
global = {
  \key d \major
  \time 3/4
}

sopMusic = \relative c' {
  % VERSE ONE
  fis4 fis fis | \break
  fis4. e8 e4
}

altoMusic = \relative c' {
  % VERSE ONE
  d4 d d |
  d4. b8 b4 |
}
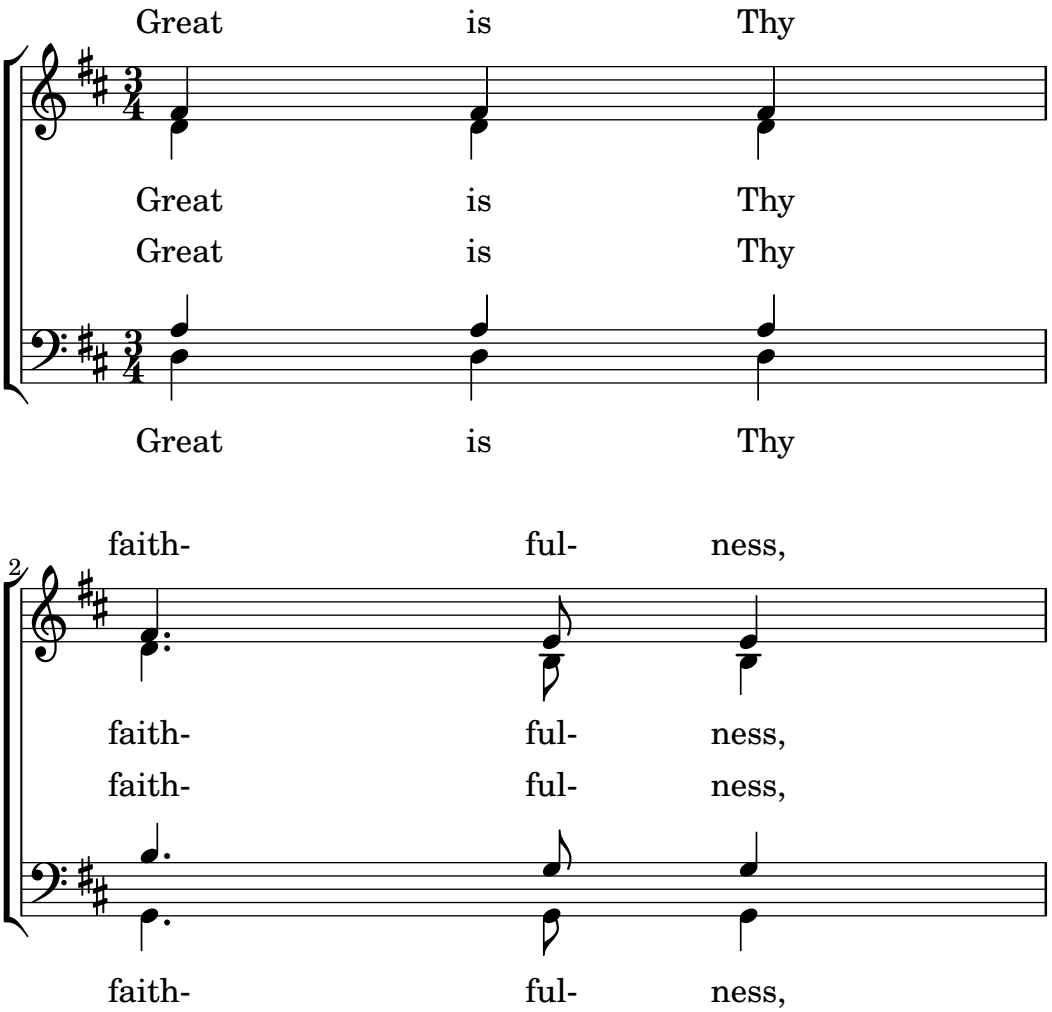
tenorMusic = \relative c' {
  a4 a a |
  b4. g8 g4 |
}

bassMusic = \relative c {
  d4 d d |
  g,4. g8 g4 |
}

words = \lyricmode {
  Great is Thy faith- ful- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
```

```
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
      \new Voice = "basses" {
        \voiceTwo  \global \bassMusic
      }
    >>
    \new Lyrics = basses
    \context Lyrics = sopranos \lyricsto sopranos \words
    \context Lyrics = altos \lyricsto altos \words
    \context Lyrics = tenors \lyricsto tenors \words
    \context Lyrics = basses \lyricsto basses \words
  >>
  \layout {
    \context {
      \Lyrics
      \override VerticalAxisGroup.staff-affinity = ##f
      \override VerticalAxisGroup.staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
    \context {
      \Staff
      \override VerticalAxisGroup.staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
  }
}
```



Véase también
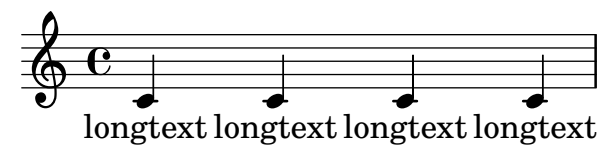
Manual de aprendizaje: Sección “Plantillas de conjuntos vocales” in *Manual de Aprendizaje*.
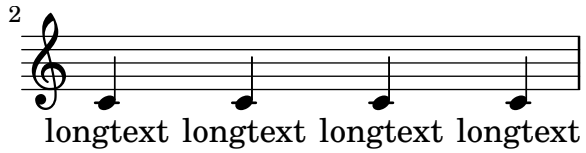
Referencia de la notación: Sección 5.1.7 [Orden de disposición de los contextos], página 584, Sección 5.1.2 [Crear y referenciar contextos], página 569.

Colocación horizontal de las sílabas

Para aumentar la separación entre las líneas de texto, establezca la propiedad `minimum-distance` de `LyricSpace`.
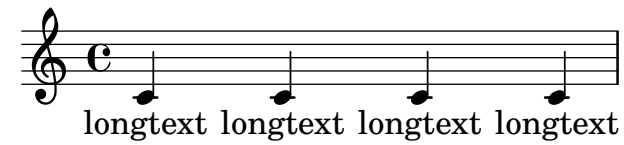
```
{
  c c c c
  \override Lyrics.LyricSpace.minimum-distance = #1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```
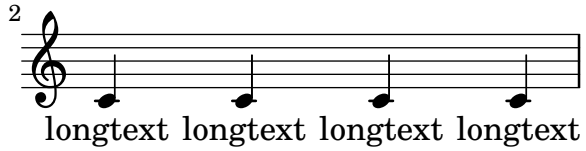




Para efectuar este cambio en todas las letras de la partitura, ajuste la propiedad dentro del bloque `\layout`.

```
\score {
  \relative c' {
  c c c c
  c c c c
  }
  \addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace.minimum-distance = #1.0
    }
  }
}
```





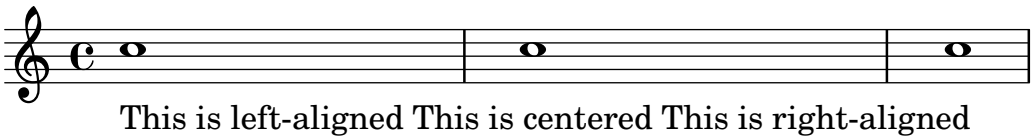
Fragmentos de código seleccionados

Alineación de la letra

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. `#-1` es izquierda, `#0` es centrado y `#1` es derecha; sin embargo, podemos usar también `#LEFT`, `#CENTER` y `#RIGHT`.

```
\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText.self-alignment-X = #1
  "This is right-aligned"
```

```
}
```



La comprobación necesaria para asegurar que las inscripciones de texto están dentro de los márgenes, requiere cálculos adicionales. Para acelerar ligeramente el procesado, se puede desactivar esta funcionalidad:

```
\override Score.PaperColumn.keep-inside-line = ##f
```

Para hacer que el texto evite también las líneas divisorias, use

```
\layout {
  \context {
    \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
      \hide BarLine
  }
}
```
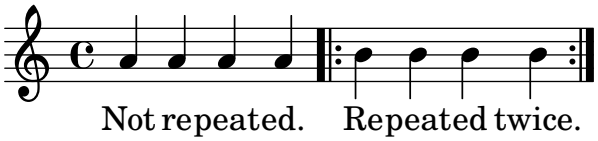
Letra y repeticiones

Repeticiones sencillas

Las repeticiones *en la música* se describen extensamente en alguna otra parte de este manual; véase Sección 1.4 [Repeticiones], página 143. Esta sección explica cómo añadir letra a secciones de música que se repiten.
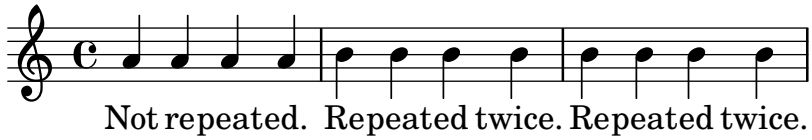
La letra que va con una sección de música que se repite, debiera estar encerrada en la misma construcción de repeticiones exacta que la música, si el texto no cambia.

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}
```



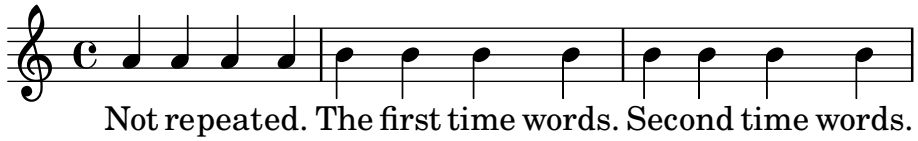
El texto se expandirá correctamente de esta manera si las repeticiones se despliegan.

```
\score {
  \unfoldRepeats {
    <<
      \new Staff {
        \new Voice = "melody" {
          \relative c'' {
            a4 a a a
            \repeat volta 2 { b4 b b b }
          }
        }
      }
      \new Lyrics {
        \lyricsto "melody" {
          Not re -- peat -- ed.
          \repeat volta 2 { Re -- peat -- ed twice. }
        }
      }
    >>
  }
}
```
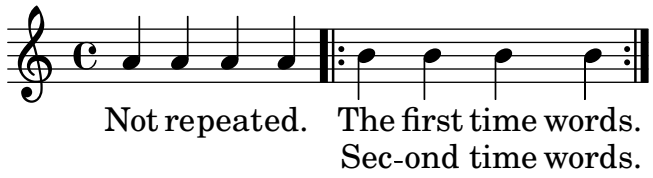


Si la sección repetida ha de desplegarse y tiene distinta letra, escriba simplemente todo el texto:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat unfold 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        The first time words.
        Sec -- ond time words.
      }
    }
  >>
}
```



Cuando el texto de las secciones de primera o segunda vez es distinto, la letra de cada repetición se debe escribir dentro de contextos `Lyrics` diferentes, anidados correctamente en secciones paralelas:

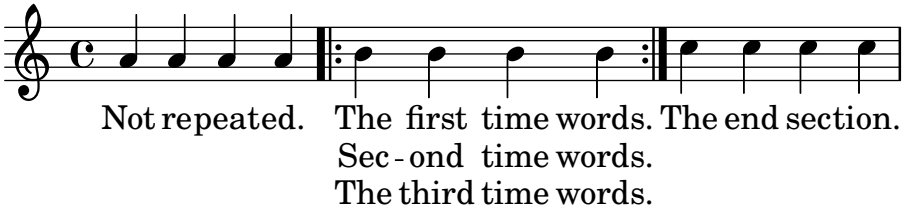
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
      >>
    }
  >>
}
```



Se pueden añadir más estrofas de una manera similar:

```
\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative c'' {
          a4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics \lyricsto "singleVoice" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
```

```
        \set associatedVoice = "singleVoice"
        Sec -- ond time words.
      }
      \new Lyrics {
        \set associatedVoice = "singleVoice"
        The third time words.
      }
    >>
    The end sec -- tion.
  }
>>
}
```

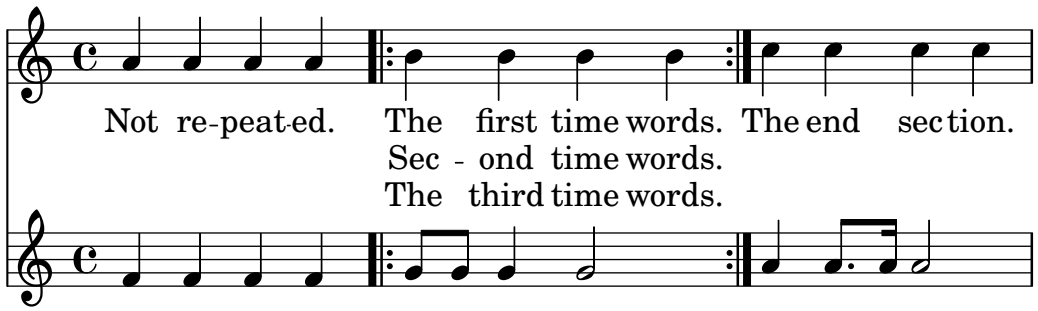


Sin embargo, si esta construcción se inserta dentro de un contexto de varios pentagramas tal como un `ChoirStaff`, la letra de los versos segundo y tercero aparecerán debajo del pentagrama inferior.

Para posicionarlos correctamente utilice `alignBelowContext`:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics = "firstVerse" \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics = "secondVerse"
        \with { alignBelowContext = #"firstVerse" } {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
        \new Lyrics = "thirdVerse"
        \with { alignBelowContext = #"secondVerse" } {
          \set associatedVoice = "melody"
          The third time words.
        }
      >>
      The end sec -- tion.
    }
```
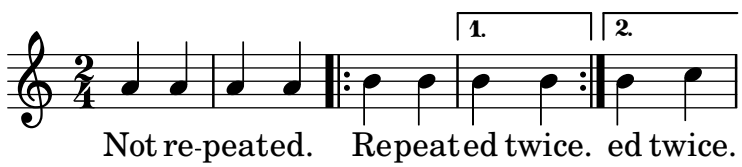
```
    \new Voice = "harmony" {
      \relative c' {
        f4 f f f \repeat volta 2 { g8 g g4 g2 } a4 a8. a16 a2
      }
    }
  >>
}
```



Repeticiones con finales alternativos (primera y segunda vez)

Si el texto de la sección repetida es el mismo, se puede usar exactamente la misma estructura tanto para la letra como para la música.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- }
        \alternative { { ed twice. } { ed twice. } }
      }
    }
  >>
}
```

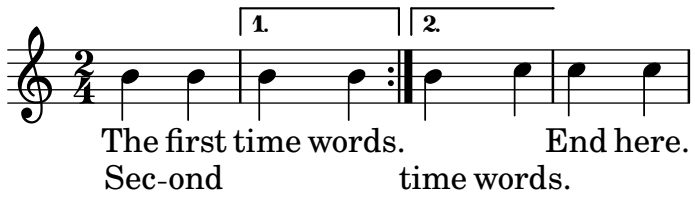


Pero cuando la sección repetida tiene un texto distinto, no se puede usar una construcción de repetición para el texto y es necesario insertar manualmente instrucciones `\skip` para saltar sobre las notas en las secciones alternativas que no correspondan.

Nota: no utilice un guión bajo, _, para saltar notas: el guión bajo indica un melisma, y hace que la sílaba anterior resulte alineada por la izquierda.

Nota: La instrucción `\skip` debe ir seguida por un número, pero este número se ignora dentro de la letra, que deriva las duraciones de las notas de una melodía asociada a través de `\addlyrics` o `\lyricsto`. Cada `\skip` salta una sola nota de cualquier valor, independientemente del valor del número que se escribe a continuación.

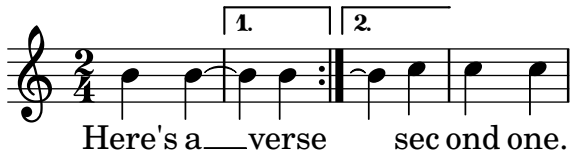
```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        The first time words.
        \repeat unfold 2 { \skip 1 }
        End here.
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Sec -- ond
        \repeat unfold 2 { \skip 1 }
        time words.
      }
    }
  >>
}
```



Cuando una nota está ligada sobre dos o más finales alternativos (sobre la primera nota de una casilla de primera y segunda vez), se usa una ligadura de unión para arrastrar la nota dentro del primer final alternativo y se usa una instrucción `\repeatTie` de ligadura de repetición en las casillas de repetición siguientes. Esta estructura produce problemas de alineación difíciles cuando hay letra de por medio; aumentar la longitud de las secciones alternativas de forma que las notas ligadas se vean contenidas completamente dentro de ellas, puede ofrecer un resultado más aceptable.

La ligadura de unión crea un melisma dentro de la casilla de primera vez pero no en la segunda vez y siguientes; por tanto, para alinear la letra correctamente es necesario inhabilitar la creación automática de melismas sobre la sección de finales alternativos e insertar saltos manuales.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        \repeat volta 2 { Here's a __ }
        \alternative {
          { \skip 1 verse }
          { \skip 1 sec }
        }
        ond one.
      }
    }
  >>
}
```

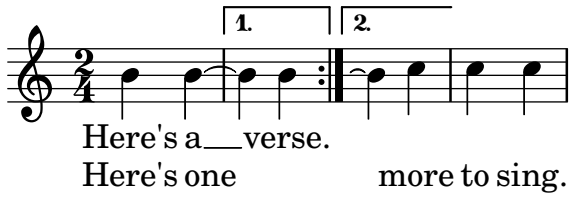


Observe que si se usa `\unfoldRepeats` sobre una sección que contiene alguna `\repeatTie`, la `\repeatTie` se debe eliminar para evitar que se impriman al mismo tiempo los dos tipos de ligadura.

Cuando la sección repetida tiene textos distintos, no se puede usar `\repeat` para la letra y se deben insertar instrucciones `\skip` manualmente, como se vio antes.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
```

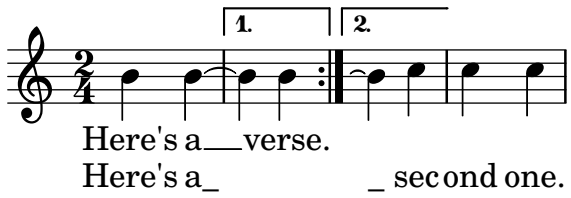
```
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's one
        \repeat unfold 2 { \skip 1 }
        more to sing.
      }
    }
  >>
}
```



Si queremos imprimir líneas extensoras y guiones que entran o salen de las casillas de primera y segunda vez, hemos de insertarlos manualmente.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's "a_"
        \skip 1
        "_" sec -- ond one.
      }
    }
  >>
```

```
}
```



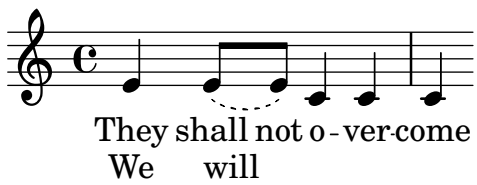
Véase también

Referencia de la notación: Sección 5.1.3 [Mantener vivos los contextos], página 573, Sección 1.4 [Repeticiones], página 143.

Letras en divisi

Cuando solamente difieren las palabras y las duraciones de las dos partes mientras que la altura de las notas es la misma, el método adecuado puede ser la desactivación temporal de la detección automática de los melismas e indicar el melisma dentro de la letra:
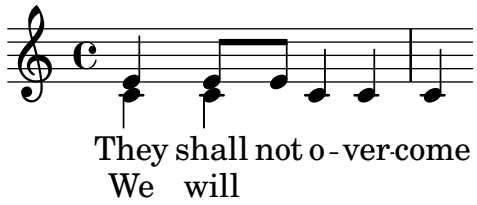
```
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDown
        \slurDashed
        e4 e8 ( e ) c4 c |
        \unset melismaBusyProperties
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "melody" {
      We will _
    }
  >>
}
```



Cuando difieren tanto la música como las palabras, puede ser mejor imprimir la música y la letra que difieren mediante el nombrado de contextos de voz y adjuntando la letra a estos contextos específicos:

```
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
```

```
        <<
          {
            \voiceOne
            e4 e8 e
          }
          \new Voice = "splitpart" {
            \voiceTwo
            c4 c
          }
        >>
        \oneVoice
        c4 c |
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "splitpart" {
      We will
    }
  >>
}
```



En la música coral es frecuente que una parte vocal se divida en dos durante varios compases. La construcción `<< {...} \\ {...} >>`, en la que dos (o más) expresiones musicales están separadas por dos barras invertidas, puede en principio parecer la forma más adecuada de realizar las voces divididas. Sin embargo, esta construcción asigna **todas** las expresiones que están dentro de ella a **contextos de voz NUEVOS** lo que hace que no se asigne a ellas *ninguna letra* porque la letra se asigna al contexto de voz original, que no es lo que pretendemos, en general. La construcción correcta que debemos usar es el pasaje polifónico temporal, véase la sección *Pasajes polifónicos temporales* dentro de [Polifonía en un solo pentagrama], página 166.

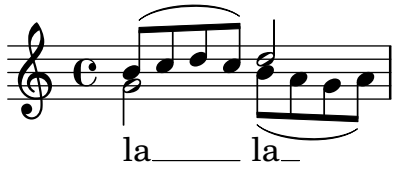
Polifonía con letras compartidas

Cuando dos voces cuyas notas tienen distintas duraciones comparten la misma letra, la alineación de ésta a una de las voces puede dar lugar a problemas en la otra voz. Por ejemplo, la segunda línea extensora que se ve debajo es demasiado corta, porque la letra está alineada solamente a la voz superior:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice = "sopranoVoice" { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
```

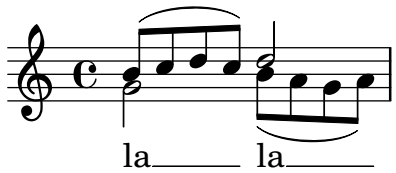
```
  \new Lyrics \lyricsto "sopranoVoice" \words
>>
```



Para obtener el resultado deseado, alinee la letra con un contexto `NullVoice` ('Voz Nula') nuevo que contenga una combinación adecuada de las dos voces. Las notas del contexto `NullVoice` no aparecen en la página impresa, pero pueden utilizarse para alinear la letra adecuadamente:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```

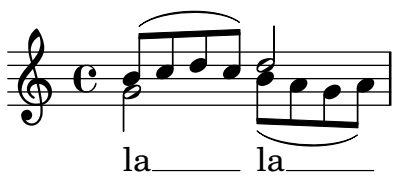


El contexto de voz nula `NullVoice` debe estar situado dentro de un contexto `Staff` y contener notas que ya se están mostrando en dicho pentagrama, y en la misma octava. En caso contrario, la `NullVoice` podría interactuar con las voces impresas de forma inesperada. Por ejemplo, notas arbitrarias en la `NullVoice` pueden hacer que aparezcan (o desaparezcan) alteraciones accidentales en el pentragrama.

Este método se puede usar trambién con la función `\partcombine`, que en sí no permite incluir letra:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice \partcombine \soprano \alto
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```



Advertencias y problemas conocidos

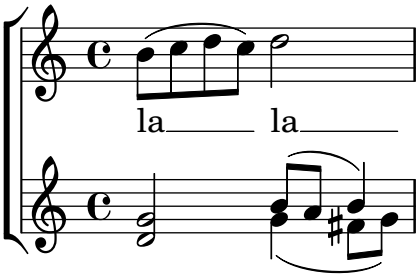
La función `\addLyrics` solo funciona con letras de `Voice`, por lo que no puede utilizarse con `NullVoice`.

La función `\partcombine` se describe en [Combinación automática de las partes], página 175.

Por último, este método se puede usar incluso si las voces están en distintos pentagramas, y no se limita a dos voces únicamente:

```
soprano = \relative { b'8( c d c) d2 }
altoOne = \relative { g'2 b8( a b4) }
altoTwo = \relative { d'2 g4( fis8 g) }
aligner = \relative { b'8( c d c) d( d d d) }
words = \lyricmode { la __ la __ }

\new ChoirStaff <<
  \new Staff <<
    \soprano
    \new NullVoice = "aligner" \aligner
  >>
  \new Lyrics \lyricsto "aligner" \words
  \new Staff \partcombine \altoOne \altoTwo
>>
```



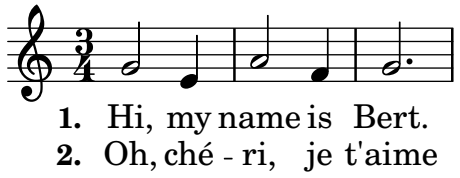
Observe, sin embargo, que en la segunda mitad del compás que se ve arriba, las notas del contexto `NullVoice` reflejan las duraciones de las notas del pentagrama inferior, pero no se desvian de la altura única que se muestra en el pentagrama al que pertenece la `NullVoice`. Aunque no es obligatorio en este ejemplo en particular, generalmente es buena idea introducir las notas de esta forma.

2.1.3 Versos

Añadir números de verso

Los números de los versos se pueden añadir estableciendo `stanza`, p.ej.,

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = #"2. "
  Oh, ché -- ri, je t'aime
}
```



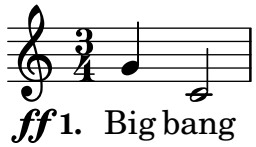
Estos números aparecerán inmediatamente antes de la primera sílaba.

Añadir expresiones dinámicas a los versos

Los versos que difieren en su sonoridad se pueden especificar escribiendo una indicación dinámica antes de cada verso. En LilyPond, todo lo que aparece delante de un verso está dentro del objeto `StanzaNumber`; las indicaciones dinámicas no son diferentes. Por razones técnicas, tendrá que establecer el valor de la sección stanza (verso) fuera de `\lyricmode`:

```
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}

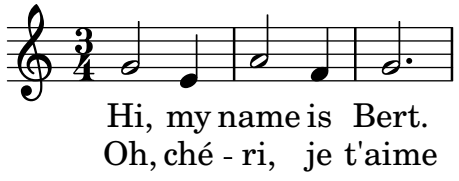
<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \text
>>
```



Añadir el nombre de los cantantes a los versos

También se pueden poner los nombres de los cantantes. Se imprimen al comienzo de la línea, igual que los nombres de instrumento. Se crean estableciendo un valor para `vocalName`. Se puede definir una versión abreviada como `shortVocalName`.

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = #"Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = #"Ernie "
  Oh, ché -- ri, je t'aime
}
```



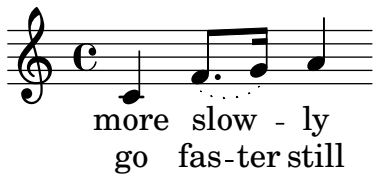
Versos con ritmos distintos

Con frecuencia, los distintos versos de una canción encajan de formas ligeramente diferentes con la misma melodía. Estas variaciones se pueden capturar con el uso de `\lyricsto`.

Ignorar los melismas

Existe la posibilidad de que el texto tenga un melisma en un verso, pero varias sílabas en otro. Una solución es hacer que la voz más rápida ignore el melisma. Esto se consigue estableciendo `ignoreMelismata` en el contexto Lyrics.

```
<<
  \relative c' \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c4
    \slurDotted
    f8.[( g16])
    a4
  }
  \new Lyrics \lyricsto "lahlah" {
    more slow -- ly
  }
  \new Lyrics \lyricsto "lahlah" {
    go
    \set ignoreMelismata = ##t
    fas -- ter
    \unset ignoreMelismata
    still
  }
>>
```



Advertencias y problemas conocidos
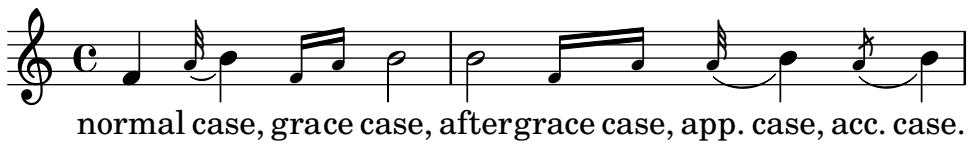
A diferencia de casi todas las instrucciones `\set`, `\set ignoreMelismata` no funciona si va precedido de `\once`. Es necesario utilizar `\set` y `\unset` para delimitar la letra en que se quieren ignorar los melismas.

Aplicar sílabas a notas de adorno

De forma predeterminada, las notas de adorno (p.ej. insertadas por medio de `\grace`) no pueden recibirla asignación de sílabas si se utiliza `\lyricsto`, pero este comportamiento puede cambiarse:

```
<<
  \new Voice = melody \relative c' {
    f4 \appoggiatura a32 b4
    \grace { f16 a16 } b2
    \afterGrace b2 { f16[ a16] }
    \appoggiatura a32 b4
    \acciaccatura a8 b4
```
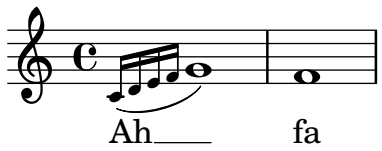
```
  }
  \new Lyrics
  \lyricsto melody {
    normal
    \set includeGraceNotes = ##t
    case,
    gra -- ce case,
    after -- grace case,
    \set ignoreMelismata = ##t
    app. case,
    acc. case.
  }
>>
```



Advertencias y problemas conocidos

Como ocurre con `associatedVoice`, `includeGraceNotes` se tiene que establecer al menos una sílaba antes de la que se va a colocar bajo una nota de adorno. Para el caso de una nota de adorno al comienzo mismo de una pieza, considere la posibilidad de usar un bloque `\with` o `\context`:

```
<<
  \new Voice = melody \relative c' {
    \grace { c16( d e f }
    g1) f
  }
  \new Lyrics \with { includeGraceNotes = ##t }
  \lyricsto melody {
    Ah __ fa
  }
>>
```

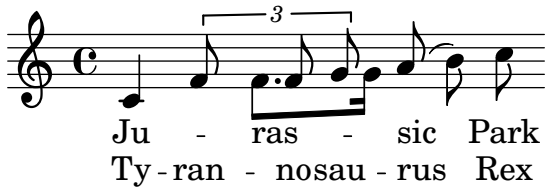


Cambio a una melodía alternativa

Son posibles variaciones más complejas en la coordinación de la letra y la música. La melodía a la que se está alineando la letra puede cambiarse desde dentro de la letra mediante el establecimiento de la propiedad `associatedVoice`:

```
<<
  \relative c' \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c4
    <<
      \new Voice = "alternative" {
```

```
        \voiceOne
        \tuplet 3/2 {
          % show associations clearly.
          \override NoteColumn.force-hshift = #-3
          f8 f g
        }
      }
      {
        \voiceTwo
        f8.[ g16]
        \oneVoice
      } >>
    a8( b) c
  }
  \new Lyrics \lyricsto "lahlah" {
    Ju -- ras -- sic Park
  }
  \new Lyrics \lyricsto "lahlah" {
    % Tricky: need to set associatedVoice
    % one syllable too soon!
    \set associatedVoice = "alternative" % applies to "ran"
    Ty --
    ran --
    no --
    \set associatedVoice = "lahlah" % applies to "rus"
    sau -- rus Rex
  } >>
```



El texto del primer verso se fija a la melodía llamada ‘lahlah’ de la forma usual, pero el segundo verso se fija inicialmente al contexto `lahlah` y después se cambia a la melodía `alternative` para las sílabas desde ‘ran’ hasta ‘sau’ mediante las líneas:

```
\set associatedVoice = "alternative" % se aplica a "ran"
Ty --
ran --
no --
\set associatedVoice = "lahlah" % se aplica a "rus"
sau -- rus Rex
```

Aquí, `alternative` es el nombre del contexto de `Voice` que contiene el tresillo.

Observe la posición de la instrucción `\set associatedVoice`: parece estar situada una sílaba antes de lo normal, pero está bien así.

Nota: La instrucción `\set associatedVoice` se debe escribir una sílaba *antes* de aquella en la que queremos que se produzca el cambio a la voz nueva. Dicho de otra forma, el cambio de la voz asociada se produce una sílaba más tarde de lo esperado. Esto es por razones técnicas, y no es un fallo.

Imprimir los versos al final

En ocasiones es conveniente tener un verso ajustado a la música, y el resto añadido en forma de estrofa al final de la pieza. Esto se puede conseguir escribiendo los versos adicionales dentro de una sección `\markup` fuera del bloque Score principal de la partitura. Tenga en cuenta que existen dos formas distintas de forzar los saltos de línea al utilizar `\markup`.

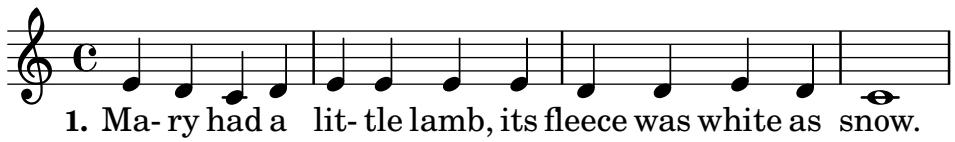
```
melody = \relative c' {
e d c d | e e e e |
d d e d | c1 |
}

text = \lyricmode {
\set stanza = #"1." Ma- ry had a lit- tle lamb,
its fleece was white as snow.
}

\score{ <<
  \new Voice = "one" { \melody }
  \new Lyrics \lyricsto "one" \text
>>
  \layout { }
}
\markup { \column{
  \line{ Verse 2. }
  \line{ All the children laughed and played }
  \line{ To see a lamb at school. }
  }
}
\markup{
  \wordwrap-string #"
  Verse 3.

  Mary took it home again,

  It was against the rule."
}
```



Verse 2.
All the children laughed and played
To see a lamb at school.

Verse 3.
Mary took it home again,
It was against the rule.

Imprimir los versos al final en varias columnas

Cuando una pieza tiene muchos versos, a menudo se imprimen en varias columnas a lo largo de toda la página. Con frecuencia un número de verso fuera del margen precede a cada verso. El ejemplo siguiente muestra cómo producir dicha salida en LilyPond.
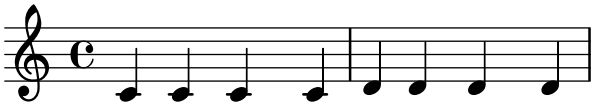
```
melody = \relative c' {
  c4 c c c | d d d d
}

text = \lyricmode {
  \set stanza = #"1." This is verse one.
  It has two lines.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \fill-line {
    \hspace #0.1 % moves the column off the left margin;
     % can be removed if space on the page is tight
     \column {
      \line { \bold "2."
        \column {
          "This is verse two."
          "It has two lines."
        }
      }
      \combine \null \vspace #0.1 % adds vertical spacing between verses
      \line { \bold "3."
        \column {
          "This is verse three."
          "It has two lines."
        }
      }
    }
    \hspace #0.1 % adds horizontal spacing between columns;
    \column {
      \line { \bold "4."
        \column {
          "This is verse four."
          "It has two lines."
        }
      }
      \combine \null \vspace #0.1 % adds vertical spacing between verses
      \line { \bold "5."
        \column {
```

```
          "This is verse five."
          "It has two lines."
        }
      }
    }
  \hspace #0.1 % gives some extra space on the right margin;
  % can be removed if page space is tight
  }
}
```



1. This is verse one. It has two lines.

2. This is verse two.
It has two lines.

3. This is verse three.
It has two lines.

4. This is verse four.
It has two lines.

5. This is verse five.
It has two lines.

Véase también

Referencia de funcionamiento interno: Sección “LyricText” in *Referencia de Funcionamiento Interno*, Sección “StanzaNumber” in *Referencia de Funcionamiento Interno*.

2.1.4 Canciones

Referencias para canciones

Por lo general, las canciones se escriben sobre tres pentagramas con la melodía del cantante en el pentagrama superior y dos pentagramas de acompañamiento de piano en la parte inferior. La letra del primer verso se imprime inmediatamente debajo del pentagrama superior. Si existe solamente una pequeña cantidad de versos adicionales, pueden imprimirse inmediatamente debajo del primero, pero si hay más versos de los que caben cómodamente allí, el segundo verso y siguientes se imprimen después de la música, como texto independiente.

Todos los elementos notacionales necesarios para escribir canciones se describen completamente en otras secciones de la documentación:

- Para montar la disposición de las pautas, véase Sección 1.6.1 [Impresión de los pentagramas], página 183.
- Para escribir música para piano, véase Sección 2.2 [Teclados y otros instrumentos de varios pentagramas], página 315.
- Para imprimir la letra de una línea melódica, véase Sección 2.1.1 [Notación común para música vocal], página 250.
- Para colocar la letra, véase [Posicionamiento vertical de la letra], página 264.
- Para escribir versos véase Sección 2.1.3 [Versos], página 282.
- Las canciones se imprimen frecuentemente con los acordes indicados mediante cifrado americano (los nombres de los acordes en letras mayúsculas) sobre las pautas. Esto se describe en Sección 2.7.2 [Imprimir los acordes], página 405.
- Para imprimir diagramas de posiciones de los acordes para el acompañamiento de guitarra o para otros instrumentos con trastes en el mástil, consulte “Marcados de diagramas de posiciones de acorde” dentro de Sección 2.4.1 [Notación común para cuerdas con trastes], página 330.

Véase también

Manual de aprendizaje: Sección “Canciones” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.1.1 [Notación común para música vocal], página 250, Sección 2.7.2 [Imprimir los acordes], página 405, Sección 1.6.1 [Impresión de los pentagramas], página 183, Sección 2.2 [Teclados y otros instrumentos de varios pentagramas], página 315, [Posicionamiento vertical de la letra], página 264, Sección 2.1.3 [Versos], página 282.

Fragmentos de código: Sección “Vocal music” in *Fragmentos de código*.

Hojas guía de acordes

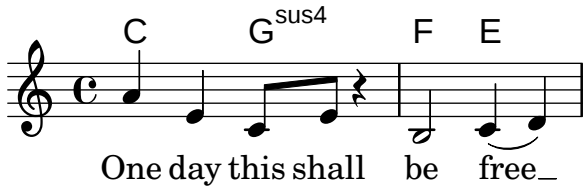
Se pueden imprimir hojas guía de acordes combinando partes vocales y el ‘modo de acordes’; esta sintaxis se explica en Sección 2.7 [Notación de acordes], página 399.

Fragmentos de código seleccionados

Hoja guía de acordes o «lead sheet» sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Véase también

Referencia de la notación: Sección 2.7 [Notación de acordes], página 399.

2.1.5 Música coral

Esta sección trata los asuntos de notación más directamente relacionados con la música coral. Esto incluye los anthems, las canciones por partes, los oratorios, etc.

Referencias para música coral

La notación de música coral se realiza normalmente sobre dos, tres o cuatro pentagramas dentro de un grupo de pautas `ChoirStaff`. Si se necesita un acompañamiento, se sitúa debajo dentro de un grupo `PianoStaff`, que se suele reducir de tamaño, para ensayar las obras corales *a cappella*. Las notas de cada parte vocal se sitúan dentro de un contexto `Voice`, y cara pentagrama recibe o una sola parte vocal (es decir, una `Voice`) o un par de partes vocales (es decir, dos `Voice`s).

Los textos se disponen en contextos `Lyrics`, bien debajo de cada pauta de música correspondiente, o bien una encima y una debajo de la pauta de música, si ésta contiene la música de las dos partes.

En otras partes del manual se describen completamente algunos otros temas sobre música coral:

- Hay una introducción a la creación de partituras vocales SATB en el Manual de aprendizaje, véase Sección "Partitura vocal a cuatro voces SATB" in *Manual de Aprendizaje*.
- También en el Manual de aprendizaje hay varias plantillas adecuadas para diversos estilos de música coral, véase Sección "Plantillas de conjuntos vocales" in *Manual de Aprendizaje*.
- Para ver más información acerca de `ChoirStaff` y `PianoStaff`, consulte [Agrupar pentagramas], página 184.
- Las figuras con cabezas de formas, como las que se usan en la notación del estilo Arpa Sacra y otros similares, se describen en [Cabezas de notas con formas diversas], página 39.
- Cuando dos partes vocales comparten un pentagrama, las plicas, ligaduras, etc. de la parte aguda se orientan hacia arriba, y los de la parte grave hacia abajo. Para hacerlo, utilice `\voiceOne` y `\voiceTwo`. Véase [Polifonía en un solo pentagrama], página 166.
- Cuando una parte vocal se divide temporalmente, debemos usar *Pasajes polifónicos temporales* (véase [Polifonía en un solo pentagrama], página 166).

Instrucciones predefinidas

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

Véase también

Manual de aprendizaje: Sección "Partitura vocal a cuatro voces SATB" in *Manual de Aprendizaje*, Sección "Plantillas de conjuntos vocales" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.7 [Orden de disposición de los contextos], página 584, [Agrupar pentagramas], página 184, [Cabezas de notas con formas diversas], página 39, [Polifonía en un solo pentagrama], página 166.

Fragmentos de código: Sección "Vocal music" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "ChoirStaff" in *Referencia de Funcionamiento Interno*, Sección "Lyrics" in *Referencia de Funcionamiento Interno*, Sección "PianoStaff" in *Referencia de Funcionamiento Interno*.

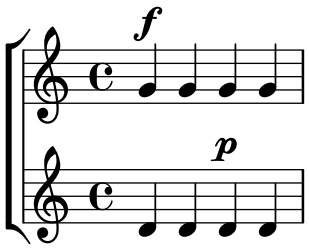
Disposiciones de la partitura para música coral

La música coral que contiene cuatro pentagramas, con o sin acompañamiento de piano, se suele disponer con dos sistemas por página. Dependiendo del tamaño de la página, conseguirlo puede requerir cambios en ciertos ajustes predeterminados. Se deben tener en cuenta los siguientes ajustes:

- Se puede modificar el tamaño global de pentagrama para alterar el tamaño general de los elementos de la partitura. Véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.
- Las diferentes distancias entre sistemas, pentagramas y líneas de texto se pueden ajustar de manera independiente. Véase Sección 4.4 [Espaciado vertical], página 536.
- Se pueden imprimir las dimensiones de las variables de disposición verticales como ayuda para el ajuste del espaciado vertical. Esta y otras posibilidades para hacer encajar la música en un número de páginas menor, se describen en Sección 4.6 [Encajar la música en menos páginas], página 564.
- Si el número de sistemas por página cambia de uno a dos, es costumbre indicarlo mediante una marca separadora de sistemas entre los dos sistemas. Véase [Separación de sistemas], página 190.
- Para ver más detalles sobre otras propiedades de formateo de las páginas, consulte Sección 4.1 [Disposición de la página], página 513.

Las indicaciones dinámicas se colocan por omisión debajo del pentagrama, pero en la música coral se suelen situar encima del pentagrama para evitar las colisiones con el texto. La instrucción predefinida `\dynamicUp` hace esta tarea para las indicaciones dinámicas en un único contexto `Voice`. Si hay muchos contextos de voz, esta instrucción predefinida tendría que colocarse en cada uno de ellos. Como alternativa, se puede usar su forma expandida para poner las indicaciones dinámicas de toda la partitura encima de sus pautas respectivas, como se muestra aquí:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
        \relative c'' { g4\f g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative c' { d4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override DynamicText.direction = #UP
      \override DynamicLineSpanner.direction = #UP
    }
  }
}
```



Instrucciones predefinidas

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Véase también

Referencia de la notación: Sección 4.6.2 [Cambiar el espaciado], página 565, Sección 4.6.1 [Mostrar el espaciado], página 564, Sección 4.6 [Encajar la música en menos páginas], página 564, Sección 4.1 [Disposición de la página], página 513, Sección 4.2 [Disposición de la partitura], página 524, [Separación de sistemas], página 190, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526, Sección 4.3.8 [Utilizar una voz adicional para los saltos de línea], página 534, Sección 4.4 [Espaciado vertical], página 536.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*, Sección “StaffGrouper” in *Referencia de Funcionamiento Interno*.

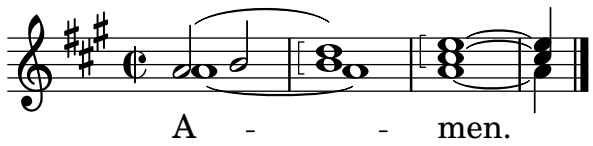
Voces divididas

Uso de arpeggioBracket para hacer más visible un divisi

El corchete de arpegios `arpeggioBracket` se puede usar para indicar la división de voces cuando no hay plicas que puedan ofrecer esta información. Se suele encontrar en la música coral.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
    >>
  }
  \layout { ragged-right = ##t }
}
```



Véase también

Referencia de la notación: Sección 1.3.3 [Expresiones como líneas], página 134.

2.1.6 Ópera y musicales

La música, letra y diálogos de las óperas y musicales se disponen normalmente de una o más de las siguientes formas:

- Una *Partitura del director* que contiene todas las partes orquestales y vocales, junto a notas guía del libreto si existen pasajes hablados.
- *Particellas orquestales* que contienen la música de los instrumentos individuales de la orquesta o banda.
- Una *Partitura vocal* que contiene todas las partes vocales con acompañamiento de piano. El acompañamiento es normalmente una reducción de orquesta, y en este caso es frecuente

indicar el nombre del instrumento original de la orquesta. Las partituras vocales a veces incluyen indicaciones de escena y notas guía del libreto.

- Un *Libro vocal* que contiene sólo las partes vocales (sin acompañamiento), a veces en combinación con el libreto.
- Un *Libreto* que contiene los diálogos completos que normalmente hay en los musicales, junto a la letra de las partes cantadas. Suelen incluirse también las indicaciones de escena. Se puede utilizar LilyPond para tipografiar libretos, pero dado que no contienen música, puede ser preferible algún método alternativo.

En la sección de referencias que aparece a continuación están relacionadas las secciones de la documentación de LilyPond que se ocupan de los temas necesarios para crear partituras en los estilos habituales de la ópera y los musicales. Después aparecen secciones que cubren las técnicas peculiares de la tipografía musical de las partituras de ópera y de musicales.

Referencias para ópera y musicales

- Una partitura de director contiene muchos pentagramas y legras agrupados. En [Agrupar pentagramas], página 184 se explican maneras de agrupar pentagramas. Para anidar unos grupos dentro de otros, consulte [Grupos de pentagramas anidados], página 188.
- La impresión de pentagramas vacíos en las partituras del director y vocales se suele inhibir. Para crear estas partituras, conocidas como “Partituras francesas” consulte [Ocultar pentagramas], página 198.
- La escritura de partituras orquestales se estudia en Sección 1.6.3 [Escritura de las particellas], página 201. Pueden ser de relevancia otras secciones del capítulo de Notación especializada, dependiendo de la orquestación utilizada. Muchos instrumentos son transpositores, consulte [Transposición de los instrumentos], página 24.
- Si el número de sistemas por página varía de una a otra página, es costumbre separar los sistemas con una marca separadora de sistemas. Véase See [Separación de sistemas], página 190.
- Para ver detalles acerca de otras propiedades de formateo de las páginas, consulte Sección 4.1 [Disposición de la página], página 513.
- Se pueden insertar notas guía de diálogos, instrucciones de escena y notas al pie, véase Sección 3.2.3 [Crear notas al pie], página 478 y Sección 1.8 [Texto], página 225. También se pueden añadir indicaciones de escena extensas con una sección de elementos de marcado independientes entre dos bloques `\score`, véase [Texto separado], página 231.

Véase también

Glosario musical: Sección “Frenched score” in *Glosario Musical*, Sección “Frenched staves” in *Glosario Musical*, Sección “transposing instrument” in *Glosario Musical*.

Referencia de la notación: Sección 3.2.3 [Crear notas al pie], página 478, [Agrupar pentagramas], página 184, [Ocultar pentagramas], página 198, [Transposición de los instrumentos], página 24, [Grupos de pentagramas anidados], página 188, Sección 4.1 [Disposición de la página], página 513, [Separación de sistemas], página 190, [Transposición], página 10, Sección 1.6.3 [Escritura de las particellas], página 201, Sección 1.8.1 [Escritura del texto], página 225.
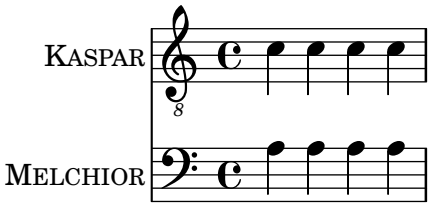
Snippets: Sección “Vocal music” in *Fragmentos de código*.

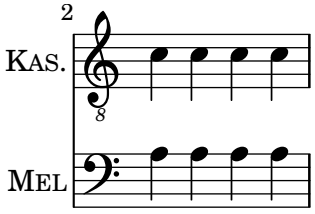
Nombres de los personajes

Se suelen mostrar los nombres de los personajes a la izquierda del pentagrama cuando éste está dedicado a dicho personaje solamente:

```
\score {
  <<
```
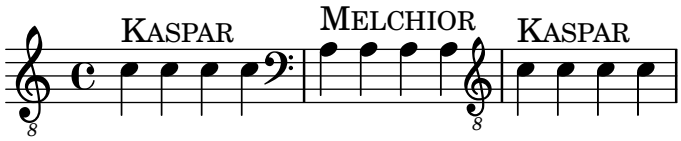
```
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative c' {
        \clef "G_8"
        c4 c c c
        \break
        c4 c c c
      }
    }
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Melchior
      \set Staff.shortVocalName = \markup \smallCaps Mel
      \clef "bass"
      \relative c' {
        a4 a a a
        a4 a a a
      }
    }
  >>
}
```





Cuando dos o más personajes comparten un pentagrama, el nombre del personaje se suele imprimir encima del pentagrama al principio de cada sección correspondiente a dicho personaje. Esto se puede hacer con elementos de marcado. Con frecuencia se usa un tipo de letra especial para este propósito.

```
\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c
\clef "bass"
a4^\markup \fontsize #1 \smallCaps Melchior
a a a
\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c
```
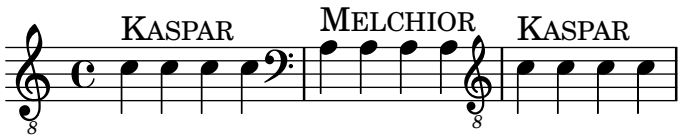


Como alternativa, si hay muchos cambios de personajes, puede ser más fácil establecer definiciones de "instrumentos" para cada instrumento en el nivel superior de la sintaxis, de forma que se puede usar `\instrumentSwitch` para indicar cada uno de los cambios.

```
\addInstrumentDefinition #"kaspar"
  #`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
     (shortInstrumentName . "Kas.")
     (clefGlyph . "clefs.G")
     (clefTransposition . -7)
     (middleCPosition . 1)
     (clefPosition . -2)
     (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Kaspar"))
     (midiInstrument . "voice oohs"))

\addInstrumentDefinition #"melchior"
  #`((instrumentTransposition . ,(ly:make-pitch 0 0 0))
     (shortInstrumentName . "Mel.")
     (clefGlyph . "clefs.F")
     (clefTransposition . 0)
     (middleCPosition . 6)
     (clefPosition . 2)
     (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Melchior"))
     (midiInstrument . "choir aahs"))

\relative c' {
  \instrumentSwitch "kaspar"
  c4 c c c
  \instrumentSwitch "melchior"
  a4 a a a
  \instrumentSwitch "kaspar"
  c4 c c c
}
```



Véase también

Referencia de la notación: [Nombres de instrumentos], página 201, Sección A.21 [Funciones de Scheme], página 773, Sección 1.8 [Texto], página 225, Sección A.10 [Instrucciones de marcado de texto], página 671.

Extender LilyPond: Sección "Construcción de elementos de marcado en Scheme" in *Extender*.

Guías musicales

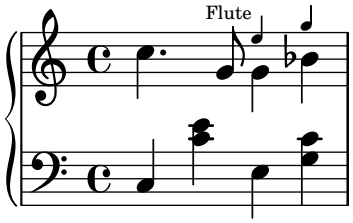
Se pueden insertar guías musicales en las partituras vocales, en los libros vocales y en las partes orquestales para indicar qué música de otra parte precede inmediatamente a una entrada. Asimismo, las guías se suelen insertar en la reducción de piano en las partituras vocales para indicar lo que está tocando cada parte orquestas. Esto sirve de ayuda al director cuando no está disponible una partitura del director.

El mecanismo básico para insertar guías se explica de forma exhaustiva en el texto principal, véase [Citar otras voces], página 204 y [Formateo de las notas guía], página 207. Pero cuando se deben insertar muchas guías, por ejemplo, como ayuda para el director en una partitura vocal, el nombre del instrumento se debe colocar cuidadosamente justo antes del comienzo de las notas guía, y cerca de ellas. El siguiente ejemplo muestra cómo se hace esto.

```
flute = \relative c'' {
  s4 s4 e g
}
\addQuote "flute" { \flute }

pianoRH = \relative c'' {
  c4. g8
  % position name of cue-ing instrument just before the cue notes,
  % and above the staff
  \new CueVoice {
    \override InstrumentSwitch.self-alignment-X = #RIGHT
    \set instrumentCueName = "Flute"
  }
  \cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}
```



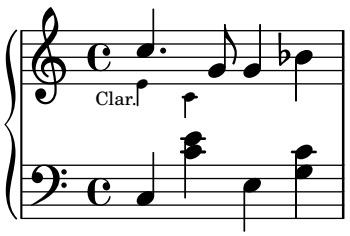
Si un instrumento transpositor se está citando, la parte instrumental debe especificar la tonalidad de manera que la conversión de las notas citadas se realice automáticamente. El ejemplo de abajo muestra esta transposición para un clarinete en Si bemol. Las notas de este ejemplo son bajas en el pentagrama, por lo que se especifica `DOWN` en `\cueDuring` (para que las plicas se orienten hacia abajo) y el nombre del instrumento se coloca debajo del pentagrama. Observe también que la voz de la mano derecha del piano está declarada explícitamente. Esto es así porque las notas citadas en este ejemplo comienzan al principio del primer compás y en caso contrario produciría que todas las notas de la mano derecha del piano se situaran dentro de un contexto `CueVoice`.

```
clarinet = \relative c' {
```

```
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  % position name of cue-ing instrument below the staff
  \new CueVoice {
    \override InstrumentSwitch.self-alignment-X = #RIGHT
    \override InstrumentSwitch.direction = #DOWN
    \set instrumentCueName = "Clar."
  }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



A partir de estos dos ejemplos debería quedar claro que insertar muchas notas guía en una partitura vocal puede ser tedioso, y la visión de las notas de la parte de piano quedaría entorpecida. Sin embargo, como muestra el siguiente fragmento de código, es posible definir una función musical para reducir la cantidad de texto que se teclea y para hacer que las notas del piano estén más claras.

Fragmentos de código seleccionados

Añadir notas guía orquestales a una partitura vocal

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro

argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `#UP` o `#DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
   (parser location instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
       \once \override TextScript.direction = $dir
       <>-\markup { \tiny #name }
       $music
     }
   #})

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }
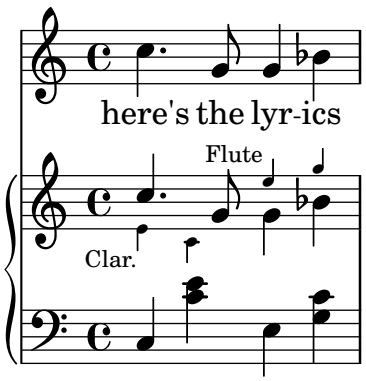
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
```

```
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Véase también

Glosario musical: Sección “cue-notes” in *Glosario Musical*.

Referencia de la notación: Sección 5.5.1 [Alineación de objetos], página 616, Sección 5.4.2 [Dirección y posición], página 601, [Formateo de las notas guía], página 207, [Citar otras voces], página 204, Sección 5.6 [Uso de las funciones musicales], página 629.

Fragmentos de código: Sección “Vocal music” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “InstrumentSwitch” in *Referencia de Funcionamiento Interno*, Sección “CueVoice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

`\cueDuring` inserta automáticamente un contexto `CueVoice` y todas las notas guía se colocan dentro de este contexto. Ello significa que no es posible tener dos secuencias de notas guía superpuestas mediante esta técnica. Las secuencias superpuestas se podrían introducir declarando explícitamente contextos `CueVoice` separados y usando `\quoteDuring` para extraer e insertar las notas citadas.

Música hablada
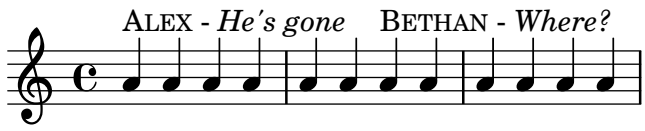
Efectos tales como el ‘parlato’ o el ‘Sprechgesang’ requieren que los intérpretes hablen sin altura definida pero con el ritmo adecuado; la notación de estos estilos se realiza mediante figuras con la cabeza en forma de cruz, como se muestra en [Cabezas de nota especiales], página 35.

Diálogos encima de la música

Los diálogos sobre la música se imprimen normalmente por encima de los pentagramas en un estilo inclinado, con el comienzo de cada frase ligado a un momento musical en particular.

Para interjecciones cortas, basta un simple elemento de marcado.

```
a4^\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
a4 a a^\markup { \smallCaps { Bethan - } \italic Where? } a
a4 a a a
```



Para frases más largas puede ser necesario expandir la música de forma que que las palabras quepan holgadamente. No está previsto en LilyPond hacer esto de manera completamente automática, y puede ser necesaria cierta intervención manual sobre la disposición.

Para frases largas o pasajes con un diálogo en disposición apretada, puede dar mejores resultados la utilización de un contexto de letra. El contexto Lyrics no se debe asociar con una voz musical; en vez de ello, cada sección del diálogo recibe una duración explícita. Si hay una pausa en medio del diálogo, se debe separar la palabra final del resto y dividirse la duración entre ellas de forma que la música subyacente reciba un espaciado continuo y adecuado.

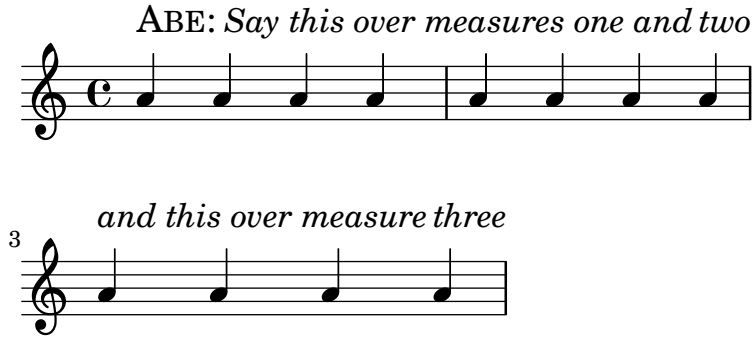
Si el diálogo se extiende a más de una línea, será necesario insertar `\break`s manualmente y ajustar la colocación del diálogo para evitar que se extienda más allá del margen derecho. La última palabra del último compás de una línea debe separarse también, como se ha visto antes.

He aquí un ejemplo que ilustra cómo puede hacerse.

```
music = \relative c'' {
  \repeat unfold 3 { a4 a a a }
}

dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
  }4*7
  "two"4 |
  \break
  "and this over measure"4*3
  "three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText.font-shape = #'italic
      \override LyricText.self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}
```



Véase también

Referencia de la notación: [Duración manual de las sílabas], página 257, Sección 1.8 [Texto], página 225.

Referencia de funcionamiento interno: Sección “LyricText” in *Referencia de Funcionamiento Interno*.

2.1.7 Cánticos salmos e himnos

La música y la letra de los cánticos, salmos e himnos suelen seguir un formato bien establecido dentro de cualquier iglesia en particular. Aunque los formatos pueden diferir de una iglesia a otra, los problemas de tipografiado que surgen son bastante similares, y se estudian en esta sección.

Referencias para cánticos y salmos

El tipografiado del canto gregoriano en distintos estilos de notación antigua se describe en Sección 2.9 [Notación antigua], página 420.

Véase también

Referencia de la notación: Sección 2.9 [Notación antigua], página 420.

Snippets: Sección “Vocal music” in *Fragmentos de código*.
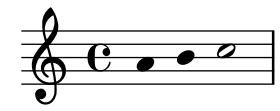
Preparar un cántico

La preparación moderna de los cánticos utiliza notación moderna con un número variable de elementos tomados de la notación antigua. Algunos de los elementos y métodos que considerar se presentan aquí.

Los cánticos usan con frecuencia figuras negras sin plica para indicar la altura, tomando las duraciones del ritmo hablado del texto.
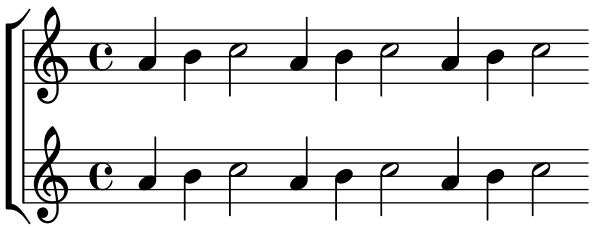
```
stemOff = { \hide Staff.Stem }

\relative c' {
  \stemOff
  a'4 b c2 |
}
```



Los cánticos con frecuencia omiten las líneas divisorias o utilizan barras de compás recortadas o discontinuas para indicar pausas en la música. Para omitir todas las barras de compás de todos los pentagramas, suprima completamente el grabador de barras:
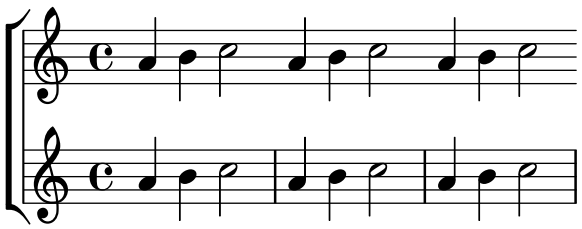
```
\score {
  \new StaffGroup <<
    \new Staff {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
    \new Staff {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
    }
  }
}
```



Las líneas divisorias también se pueden quitar pentagrama a pentagrama:
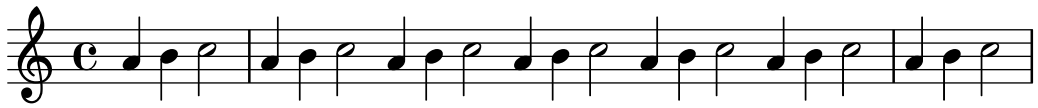
```
\score {
  \new ChoirStaff <<
    \new Staff
    \with { \remove "Bar_engraver" } {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
    \new Staff {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
```
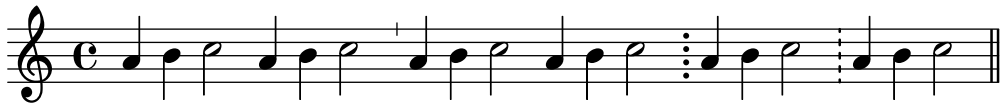
```
}
```



Para suprimir las líneas divisorias sólo de una sección de música, trátela como si fuera una cadencia. Si la sección es larga, podría necesitar insertar líneas divisorias "mudas" con `\bar ""` para indicar dónde se debe producir el salto de línea.

```
a4 b c2 |
\cadenzaOn
a4 b c2
a4 b c2
\bar ""
a4 b c2
a4 b c2
\cadenzaOff
a4 b c2 |
a4 b c2 |
```



Los silencios o pausas en los cánticos pueden indicarse mediante barras de compás modificadas.

```
a4
\cadenzaOn
b c2
a4 b c2
\bar "'"
a4 b c2
a4 b c2
\bar ";"
a4 b c2
\bar "!"
a4 b c2
\bar "||"
```
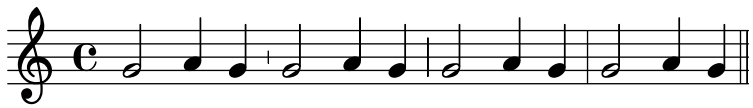


Como alternativa, se usa a veces la notación del canto gregoriano para las pausas o silencios, aunque el resto de la notación sea moderna. Este ejemplo utiliza una marca `\breathe` modificada:

```
divisioMinima = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-minima
  \once \override BreathingSign.Y-offset = #0
```

```
  \breathe
}
divisioMaior = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-maior
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
divisioMaxima = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-maxima
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
finalis = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
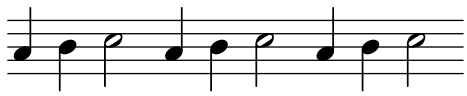
\score {
  \relative c'' {
    g2 a4 g
    \divisioMinima
    g2 a4 g
    \divisioMaior
    g2 a4 g
    \divisioMaxima
    g2 a4 g
    \finalis
  }
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
    }
  }
}
```



Los cánticos suelen omitir la indicación de compás y con frecuencia omiten la clave también.

```
\score {
  \new Staff {
    \relative c'' {
      a4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
```

```
      \Staff
      \remove "Bar_engraver"
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
    }
  }
}
```



Los cánticos de salmos en la tradición anglicana suelen ser *sencillos*, con siete compases de música, o *dobles*, con dos períodos de siete compases. Cada grupo de siete compases se divide en dos mitades, que corresponden a las dos mitades de cada verso, normalmente separados mediante una doble línea divisoria. Sólo se utilizan redondas y blancas. El primer compás de cada mitad siempre contiene un único acorde en redondas. Ésta es la "nota de recitado". Los cánticos se suelen centrar sobre la página.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

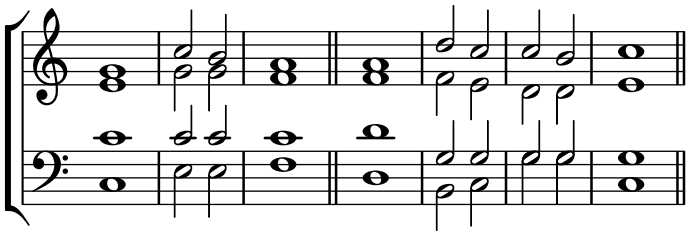
TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score {  % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
```

```
            \voiceOne
            \SopranoMusic
          >>
          \new Voice = "Alto" <<
            \voiceTwo
            \AltoMusic
          >>
        >>
        \new Staff <<
          \clef "bass"
          \global
          \new Voice = "Tenor" <<
            \voiceOne
            \TenorMusic
          >>
          \new Voice = "Bass" <<
            \voiceTwo
            \BassMusic
          >>
        >>
      >>
    >>
    \layout {
      \context {
        \Score
        \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/2)
      }
      \context {
        \Staff
        \remove "Time_signature_engraver"
      }
    }
  }  % End score
 }
}  % End markup
```



Otros enfoques para la preparación de este cántico se muestran en el primero de los siguientes fragmentos de código.

Fragmentos de código seleccionados

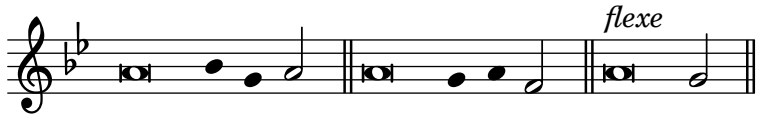
Notación de responsos o salmos

Este tipo de notación se utiliza para el canto de los Salmos, en que las estrofas no siempre tienen la misma longitud.

```
stemOff = \hide Staff.Stem
```

```
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}
```



Los cánticos y otros textos litúrgicos pueden disponerse de una forma más libre, y posiblemente usen elementos notacionales tomados de la música antigua. Con frecuencia la letra se imprime por debajo y alineada con las notas. En este caso, las notas se disponen espaciadas de acuerdo con las sílabas en vez de hacerlo con las duraciones de las notas.

Plantilla para notación de música antigua (transcripción moderna de canto gregoriano)

Este ejemplo muestra cómo hacer una transcripción moderna de canto gregoriano. El canto gregoriano no tiene compás ni plicas; utiliza solamente cabezas de nota de blanca y de negra, y unas marcas especiales que indican silencios de distintas longitudes.
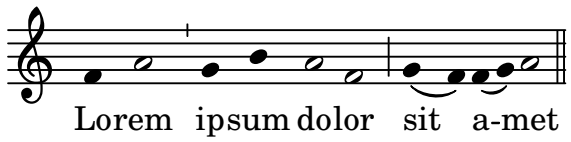
```
\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
```

```
      \hide Stem
    }
    \context {
      \Voice
      \override Stem.length = #0
    }
    \context {
      \Score
      barAlways = ##t
    }
  }
}
```



Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.9 [Notación antigua], página 420, [Barras de compás], página 94, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 575, Sección 2.9.4 [Tipografiado del canto gregoriano], página 432, [Música sin compasear], página 71, Sección 5.4.6 [Visibilidad de los objetos], página 608.

Puntuación de un salmo

El texto de un salmo anglicano se suele imprimir en versos separados debajo del canto.

Los cantos sencillos (con siete compases) se repiten para cada verso. Los cantos dobles (con 14 compases) se repiten para cada par de versos. Se insertan marcas dentro del texto para indicar la forma en que encajan con el canto. Cada verso se divide en dos mitades. Se suele utilizar un signo de dos puntos para indicar esta división. Esto corresponde a la doble línea divisoria de la música. El texto antes de los dos puntos se canta con los tres primeros compases de la música; el texto después de los dos puntos se canta con los últimos cuatro compases.

Se insertan líneas de compás simples (o en algunos casos una coma invertida u otro símbolo similar) entre el texto para indicar dónde caen las líneas divisorias de la música. En el modo de marcado, se puede introducir una línea de compás simple con el símbolo de comprobación de compás, la barra vertical, `|`.

```
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing | unto the | Lord : let }
        \line { us heartily rejoice in the | strength of | our }
        \line { sal- | -vation. }
      }
    }
  }
}
```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

otros símbolos pueden requerir glifos de las fuentes tipográficas `fetaMusic`. Para ver más detalles, consulte Sección 1.8.3 [Tipografías], página 246.

```
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing \tick unto the \tick Lord : let }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```

O come let us sing ' unto the ' Lord : let
us heartily rejoice in the ' strength of ' our
sal ' vation.

Donde hay una redonda en un compás, todo el texto que corresponde a ese compás se recita sobre esa nota en el ritmo de la palabra. Donde hay dos notas en un compás, suele haber solamente una o dos sílabas correspondientes. Si hay más de dos sílabas, se suele insertar un punto para indicar dónde se produce el cambio de nota.

```
dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us sing \tick unto \dot the \tick Lord : let
        }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```

O come let us sing ′ unto • the ′ Lord : let
us heartily rejoice in the ′ strength of ′ our
sal ′ vation.

En algunos salterios (libros de salmos) se usa un asterisco para indicar un corte en una sección recitada en lugar de una coma y las sílabas acentuadas o ligeramente alargadas se indican en estilo negrita.

```
dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { Today if ye will hear his voice * }
        \line {
          \concat { \bold hard en }
          | not your | hearts : as in the pro-
        }
        \line { vocation * and as in the \bold day of tempt- | }
        \line { -ation | in the | wilderness. }
      }
    }
  }
}
```

Today if ye will hear his voice *
harden | not your | hearts : as in the pro-
vocation * and as in the **day** of tempt- |
-ation | in the | wilderness.

En otros salterios se coloca un acento sobre la sílaba para indicar el énfasis.

```
tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us \concat {
            si \combine \tick ng
          }
          | unto the | Lord : let
        }
        \line {
          us heartily \concat {
            rejo \combine \tick ice
          }
          in the | strength of | our
        }
```

```
        \line { sal- | -vation. }
      }
    }
  }
}
```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

El uso del marcado para centrar texto y para disponer las líneas en columnas se describe en Sección 1.8.2 [Formatear el texto], página 233.

Casi todos estos elementos se muestran en uno u otro de los dos versos en la plantilla, véase Sección “Salmos” in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Salmos” in *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.8.3 [Tipografías], página 246, Sección 1.8.2 [Formatear el texto], página 233.

Compases parciales en melodías de himno

Con frecuencia las melodías de himno comienzan y terminan cada línea de música con compases parciales de forma que cada línea de música corresponde exactamente con una línea de texto. Esto requiere una instrucción `\partial` al principio de la música e instrucciones `\bar "|"` ó `\bar "||"` al final de cada línea.

Plantilla para himnos

Este fragmento de código muestra una forma de preparar un himno cuando cada línea comienza con un compás parcial. También muestra cómo añadir los versos como texto independiente debajo de la música.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
```

```
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic =  \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

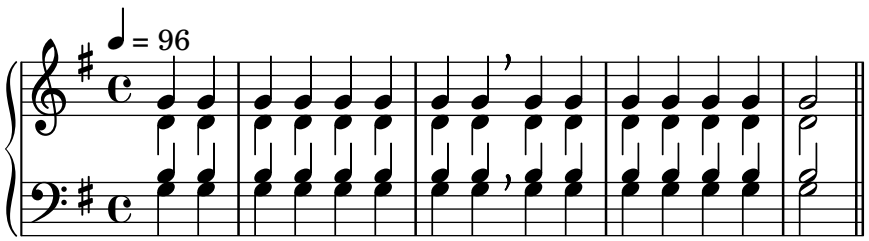
global = {
 \key g \major
}

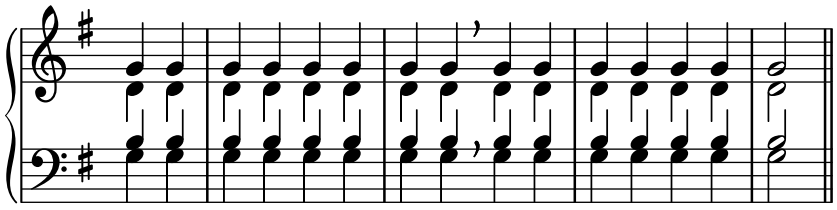
\score {  % Start score
  <<
    \new PianoStaff <<  % Start pianostaff
      \new Staff <<  % Start Staff = RH
        \global
        \clef "treble"
        \new Voice = "Soprano" <<  % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >>  % End Voice = "Soprano"
        \new Voice = "Alto" <<  % Start Voice = "Alto"
          \Timeline
          \voiceTwo
          \AltoMusic
        >>  % End Voice = "Alto"
      >>  % End Staff = RH
      \new Staff <<  % Start Staff = LH
        \global
        \clef "bass"
        \new Voice = "Tenor" <<  % Start Voice = "Tenor"
          \Timeline
          \voiceOne
          \TenorMusic
        >>  % End Voice = "Tenor"
        \new Voice = "Bass" <<  % Start Voice = "Bass"
          \Timeline
          \voiceTwo
          \BassMusic
        >>  % End Voice = "Bass"
      >>  % End Staff = LH
    >>  % End pianostaff
  >>
}  % End score

\markup {
  \fill-line {
    ""
    {
      \column {
```

```
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
    ""
  }
}

\paper {  % Start paper block
  indent = 0     % don't indent first system
  line-width = 130   % shorten line length to suit music
}  % End paper block
```





This is line one of the first verse
This is line two of the same
And here's line three of the first verse
And the last line of the same

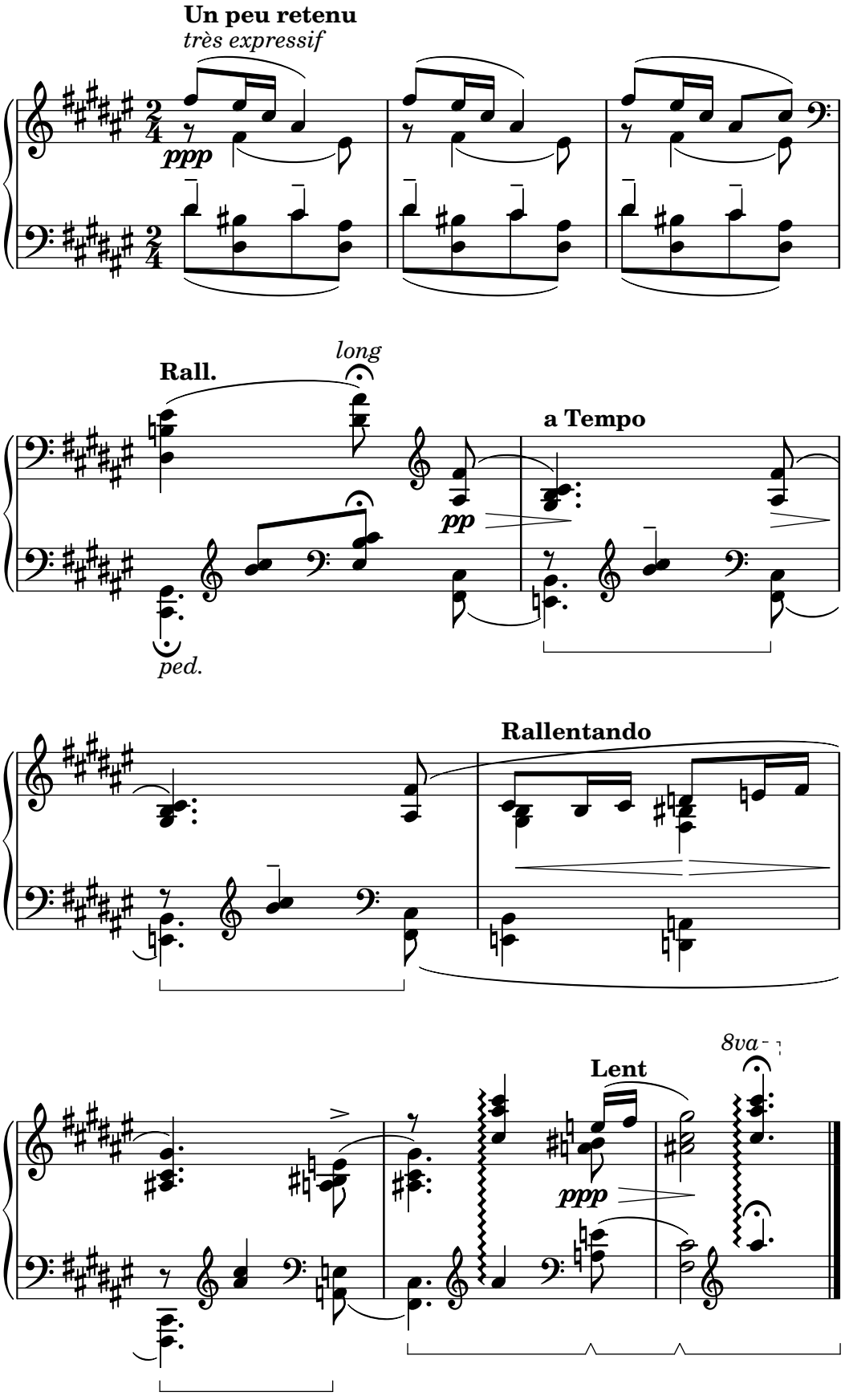
2.1.8 Música vocal antigua

Está contemplada la música vocal en estilo de tipografía antiguo, como se explica en Sección 2.9 [Notación antigua], página 420.

Véase también

Referencia de la notación: Sección 2.9 [Notación antigua], página 420.

2.2 Teclados y otros instrumentos de varios pentagramas



Esta sección se ocupa de varios aspectos de la notación musical que son exclusivos de los instrumentos de teclado y otros instrumentos cuya notación se realiza sobre varios pentagramas, como el arpa o el vibráfono. A los efectos de nomenclatura, en esta sección se denominan abreviadamente "teclados" a todo este grupo de instrumentos de varios pentagramas, aunque algunos de ellos no tienen teclado.

2.2.1 Notación común para instrumentos de teclado

Esta sección trata aspectos de notación que pueden aparecer en casi todos los instrumentos de teclado.

Referencias para teclados

La notación de los instrumentos de teclado se suele realizar con sistemas de piano. Éstos son dos pentagramas normales agrupados mediante una llave. También se utiliza esta notación para otros instrumentos de teclado. La música de órgano se escribe normalmente con dos pentagramas dentro de un grupo `PianoStaff` y un tercer pentagrama normal para los pedales.

Los pentagramas son hasta cierto punto independientes, pero a veces las voces pueden cruzarse entre los dos pentagramas. Esta sección trata técnicas de notación que son particulares de la música de teclado.

En otros lugares se tratan varios problemas comunes de la música de teclado:

- La música para teclado suele contener varias voces y el número de éstas puede variar ampliamente; esto se describe en [Resolución de las colisiones], página 170.
- La música para teclado se puede escribir en paralelo, como se describe en [Escribir música en paralelo], página 180.
- Los matices dinámicos se pueden colocar en un contexto `Dynamics`, entre los dos contextos `Staff` para alinear las indicaciones de matiz sobre una línea horizontal equidistante de los dos pentagramas; véase [Matices dinámicos], página 119.
- Las digitaciones se indican como puede verse en [Indicaciones de digitación], página 214.
- Las indicaciones de pedal de órgano se insertan como articulaciones, véase Sección A.13 [Lista de articulaciones], página 725.
- Las líneas verticales de rejilla se pueden mostrar como se describe en [Líneas de rejilla], página 222.
- La música para teclado incluye con frecuencia ligaduras *Laissez vibrer* así como ligaduras sobre arpegios y trémolos, descritos en [Ligaduras de unión], página 51.
- La colocación de arpegios a través de varias voces y pentagramas se cubre en [Arpegio], página 138.
- Las indicaciones de trémolo se describen en [Repeticiones de trémolo], página 159.
- Varios de los trucos que pueden ser necesarios en música para teclado se muestran en Sección "Ejemplos reales de música" in *Manual de Aprendizaje*.
- Las notas ocultas se pueden usar para producir ligaduras de unión que se cruzan entre voces, como se explica en Sección "Otras aplicaciones de los trucos" in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Ejemplos reales de música" in *Manual de Aprendizaje*, Sección "Otras aplicaciones de los trucos" in *Manual de Aprendizaje*.

Referencia de la notación: [Nombres de instrumentos], página 201, [Resolución de las colisiones], página 170, [Escribir música en paralelo], página 180, [Indicaciones de digitación], página 214, Sección A.13 [Lista de articulaciones], página 725, [Líneas de rejilla], página 222, [Ligaduras de unión], página 51, [Arpegio], página 138, [Repeticiones de trémolo], página 159.

Referencia de funcionamiento interno: Sección “PianoStaff” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Cambiar de pentagrama manualmente

Las voces se pueden cambiar de pentagrama manualmente usando la instrucción

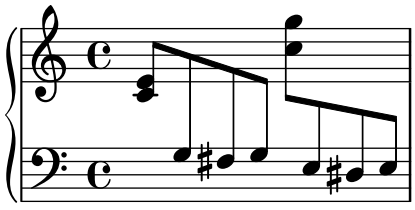
`\change Staff = nombre_del_pentagrama`

La cadena *nombre_del_pentagrama* es el nombre del pentagrama. Cambia la voz actual de su pentagrama en curso al pentagrama de nombre *nombre_del_pentagrama*. Los valores más corrientes para *nombre_del_pentagrama* son `"arriba"` y `"abajo"`, o bien `"MD"` y `"MI"`.

El pentagrama al que salta la voz debe existir en el momento del salto. Si es necesario, los pentagramas se tienen que “mantener con vida”, véase Sección 5.1.3 [Mantener vivos los contextos], página 573.

Las notas de pentagrama cruzado reciben la barra automáticamente:

```
\new PianoStaff <<
  \new Staff = "up" {
    <e' c'>8
    \change Staff = "down"
    g8 fis g
    \change Staff = "up"
    <g'' c''>8
    \change Staff = "down"
    e8 dis e
    \change Staff = "up"
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```

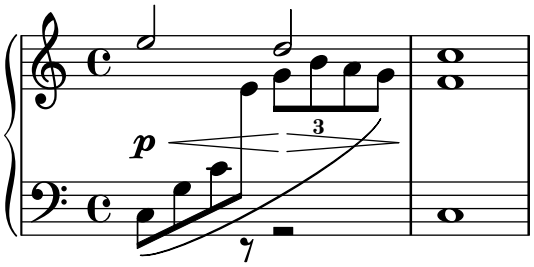


Si el barrado necesita algún trucaje, efectúe en primer lugar los cambios en las direcciones de las plicas. Las posiciones de las barras se miden posteriormente a partir del centro del pentagrama que está más cerca de la barra. Para ver un ejemplo sencillo de trucaje de las barras, consulte Sección “Arreglar notación con superposiciones” in *Manual de Aprendizaje*.

Se puede producir una superposición en la notación cuando las voces se cruzan de un pentagrama a otro:

```
\new PianoStaff <<
  \new Staff = "up" {
    \voiceOne
    % Make space for fingering in the cross-staff voice
    \once\override DynamicLineSpanner.staff-padding = #4
```

```
    e''2\p\< d''\>
    c''1\!
  }
  \new Staff = "down" <<
    {
      \clef bass
      s4. e,8\rest g,2\rest
      c1
    } \\ {
      c8\( g c'
      \change Staff = "up"
      e' g' b'-3 a' g'\)
      f'1
    }
  >>
>>
```



La plica y la ligadura de expresión se superponen a la línea del símbolo de dinámica porque la resolución automática de colisiones se suspende para las barras, ligaduras y otros objetos de extensión que conectan notas situadas en distintos pentagramas, así como para las plicas y articulaciones si su colocación está afectada por un objeto de extensión de pentagrama cruzado. Las colisiones resultantes se pueden resolver manualmente donde sea necesario, utilizando los métodos descritos en la sección Sección “Arreglar notación con superposiciones” in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Arreglar notación con superposiciones” in *Manual de Aprendizaje*.

Referencia de la notación: [Plicas], página 219, [Barras automáticas], página 79, Sección 5.1.3 [Mantener vivos los contextos], página 573.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” in *Referencia de Funcionamiento Interno*, Sección “ContextChange” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El mecanismo para evitar las colisiones de barras de corchea no funciona para las barras automáticas que terminan justo antes de un cambio de pauta. En este caso, utilice barras manuales.
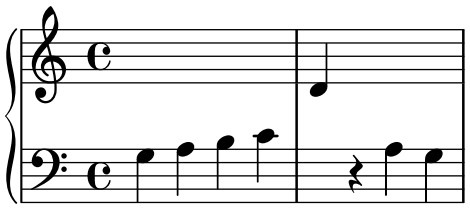
Cambiar de pentagrama automáticamente

Se puede hacer que las voces cambien automáticamente entre los pentagramas inferior y superior. La sintaxis para conseguir esto es

`\autochange ...`*`música`*`...`

Esto creará dos pentagramas dentro del grupo actual de pentagramas (normalmente un `PianoStaff`), llamados `"up"` (arriba) y `"down"` (abajo). El pentagrama inferior estará en clave de Fa de forma predeterminada. El cambiador automático hace el cambio basándose en la altura (el Do central es el punto de inflexión), y se adelanta saltando sobre los silencios para hacer cambios por adelantado.

```
\new PianoStaff {
  \autochange {
    g4 a b c'
    d'4 r a g
  }
}
```



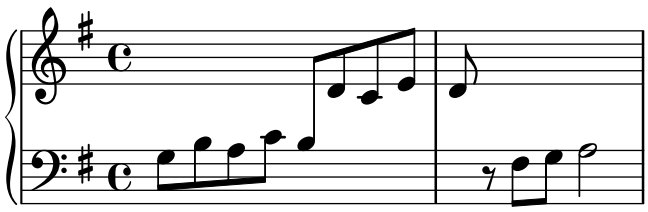
Una sección marcada `\relative` que se encuentra fuera de `\autochange` no tiene efecto sobre las notas de la música; por tanto, en caso necesario escriba `\relative` dentro de `\autochange`.

Si se necesita más control sobre los pentagramas individuales, se pueden crear manualmente con los nombres `"up"` y `"down"`. Entonces, la instrucción `\autochange` alternará su voz entre los pentagramas existentes.

Nota: Si se crean los pentagramas manualmente, *se deben* llamar `"up"` y `"down"` (en inglés).

Por ejemplo, esto es necesario para colocar una armadura de tonalidad en el pentagrama inferior:

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melodiaUno" {
      \key g \major
      \autochange \relative c' {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
>>
```



Véase también

Referencia de la notación: [Cambiar de pentagrama manualmente], página 317.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “AutoChangeMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

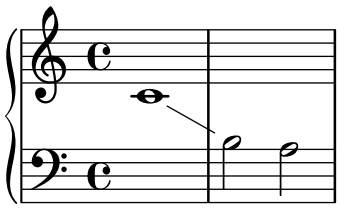
Los cambios de pentagrama pueden no acabar en lugares óptimos. Para conseguir una salida de alta calidad es mejor especificar manualmente los cambios de pentagrama.

Los acordes no se dividen entre los pentagramas; se asignan a un pentagrama en función de la primera nota nombrada dentro de la construcción del acorde.

Líneas de cambio de pentagrama

Cuando una voz cambia a otro pentagrama, se puede imprimir automáticamente una línea que conecte las notas:

```
\new PianoStaff <<
  \new Staff = "one" {
    \showStaffSwitch
    c1
    \change Staff = "two"
    b2 a
  }
  \new Staff = "two" {
    \clef bass
    s1*2
  }
>>
```



Instrucciones predefinidas

`\showStaffSwitch`, `\hideStaffSwitch`.

Véase también

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_head_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” in *Referencia de Funcionamiento Interno*.
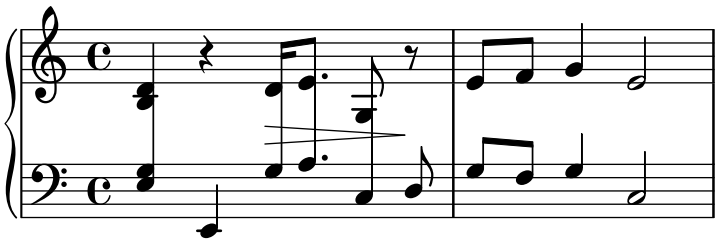
Plicas de pentagrama cruzado

Se pueden hacer acordes que cruzan los pentagramas usando el grabador `Span_stem_engraver`. Debe tenerse cuidado para estar seguro de que las barras automáticas no unen las notas de una pauta cuando no es necesario que estén unidas en la otra pauta.

```
\layout {
  \context {
```

```
    \PianoStaff
    \consists #Span_stem_engraver
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}
```



Por el momento no se puede especificar este grabador por su nombre entre comillas, sino prefijando el nombre con un símbolo de almohadilla `#`, a causa de la manera en que está implementado.

Fragmentos de código seleccionados

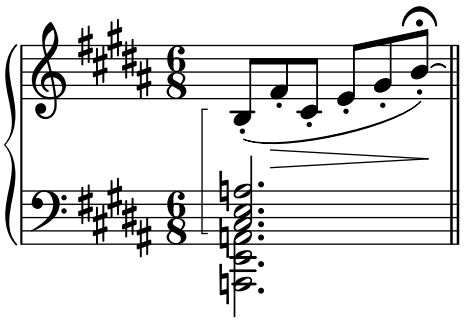
Indicar acordes de pentagrama cruzado con corchetes de arpegio

Un corchete de arpegio puede indicar que se tienen que tocar con la misma mano notas que están en dos pentagramas distintos. Para hacerlo, el `PianoStaff` se debe configurar para que acepte símbolos de arpegio de pentagrama cruzado y los símbolos de arpegio se deben configurar a la forma de corchete en el contexto de `PianoStaff`.

(Debussy, Les collines d'Anacapri, m. 65)

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio.stencil = #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
      b8-.(\arpeggio fis'-.\> cis-. e-. gis-. b-.)\!\fermata^\laissezVibrer
      \bar "||"
    }
```

```
  }
  \new Staff {
    \relative c' {
      \clef bass
      \key b \major
      <<
        {
          <a e cis>2.\arpeggio
        }
        \\
        {
          <a, e a,>2.
        }
      >>
    }
  }
>>
```



Véase también

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem” in *Referencia de Funcionamiento Interno*.
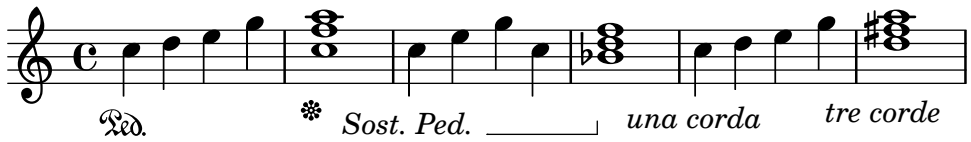
2.2.2 Piano

Esta sección trata de aspectos de la notación directamente relacionados con el piano.

Pedales de piano
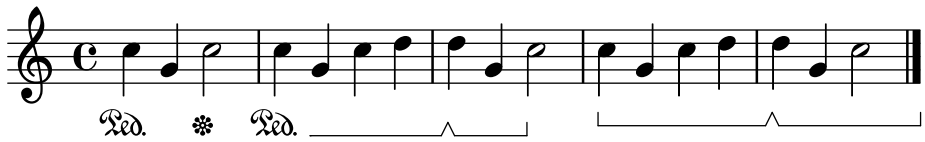
En general, los pianos tienen tres pedales que alteran la forma de producir el sonido: el pedal de mantenimiento (pedal *derecho* o pedal *forte*), el pedal *sostenuto* (*sos.*, pedal central) y el pedal *una corda* (*U.C.*, pedal izquierdo). Los pedales de mantenimiento como el pedal derecho del piano se encuentran también en el vibráfono y en la celesta.

```
c4\sustainOn d e g
<c, f a>1\sustainOff
c4\sostenutoOn e g c,
<bes d f>1\sostenutoOff
c4\unaCorda d e g
<d fis a>1\treCorde
```



Existen tres estilos de indicaciones de pedal: textual, corchete y mixta. El pedal derecho y el pedal de una corda utilizan el estilo textual de forma predeterminada, mientras que el pedal sostenuto utiliza el método mixto de forma predeterminada.

```
c4\sustainOn g c2\sustainOff
\set Staff.pedalSustainStyle = #'mixed
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2\sustainOff
\set Staff.pedalSustainStyle = #'bracket
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2
\bar "|."
```



La colocación de las instrucciones de pedal se corresponden con el movimiento físico del pedal derecho en el transcurso de la ejecución. La pedalización hasta la barra final de compás se indica mediante la omisión de la instrucción final de levantar el pedal.

Las indicaciones de pedal se pueden poner dentro de un contexto `Dynamics`, que los alinea sobre una línea horizontal.

Véase también

Referencia de la notación: [Ligaduras de unión], página 51.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SustainPedal” in *Referencia de Funcionamiento Interno*, Sección “SustainPedalLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “SustainEvent” in *Referencia de Funcionamiento Interno*, Sección “SostenutoPedal” in *Referencia de Funcionamiento Interno*, Sección “SostenutoPedalLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “SostenutoEvent” in *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedal” in *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedalLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “UnaCordaEvent” in *Referencia de Funcionamiento Interno*, Sección “PianoPedalBracket” in *Referencia de Funcionamiento Interno*, Sección “Piano_pedal_engraver” in *Referencia de Funcionamiento Interno*.

2.2.3 Acordeón

Esta sección se ocupa de la notación que es exclusiva del acordeón.

Símbolos de Discanto

Los acordeones se suelen construir con más de un conjunto de lengüetas que pueden estar al unísono, una octava por encima, o una octava por debajo de la altura escrita. Cada fabricante de acordeones utiliza distintos nombres para los *cambios* que seleccionan las distintas combinaciones de lengüetas, como *oboe*, *musette* o *bandoneón*, de manera que ha entrado en uso un sistema de signos para simplificar las instrucciones de ejecución.

Fragmentos de código seleccionados
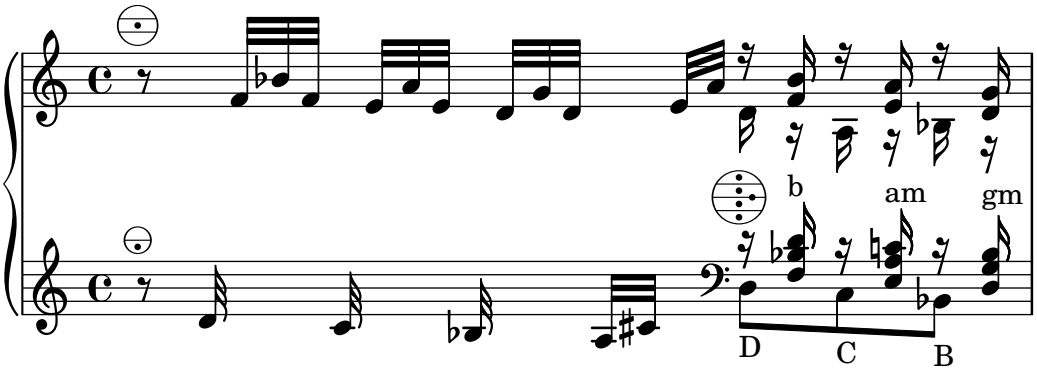
Símbolos de registros de acordeón

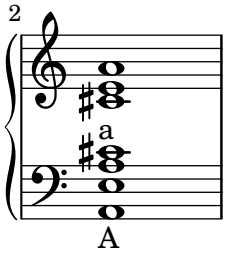
Los símbolos de registración para el acordeón están disponibles como elementos `\markup` así como en forma de eventos musicales autónomos (pues los cambios de registro suelen producirse entre eventos musicales reales). Los registros de bajo no están muy estandarizados. Pueden verse las instrucciones disponibles en la sección Sección “Registros de acordeón” in *Referencia de la Notación*.

```
\layout { ragged-right = ##t }

#(use-modules (scm accreg))

\new PianoStaff
<<
  \new Staff \relative
  { \clef treble \discant "10" r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    << { r16 <f bes> r <e a> r <d g> } \\ { d r a r bes r } >> | <cis e a>1 }
  \new Staff \relative
  { \clef treble \freeBass "1" r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    << { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
         <e a cis>1^"a" } \\
       { d8_"D" c_"C" bes_"B" | a1_"A" }
     >>
  }
>>
```





Véase también

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

2.2.4 Arpa

Esta sección trata sobre asuntos de notación específicos del arpa.

Referencias para notación de arpa

Ciertas características comunes de la música de arpa se estudian en otras partes de la documentación:

- El glissando es la técnica más característica del arpa, [Glissando], página 134.
- El *bisbigliando* se escribe como un trémolo, [Repeticiones de trémolo], página 159.
- Los armónicos naturales se estudian bajo el epígrafe [Armónicos], página 327.
- Para los arpegios dirigidos y los no arpegios, véase [Arpegio], página 138.
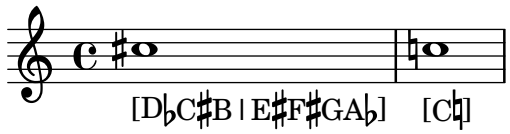
Véase también

Referencia de la notación: [Repeticiones de trémolo], página 159, [Glissando], página 134, [Arpegio], página 138, [Armónicos], página 327.

Pedales de arpa

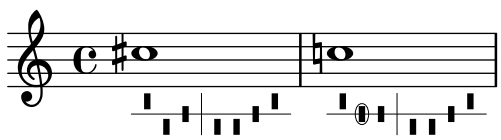
Las arpas tienen siete cuerdas por octava que pueden sonar a la altura natural, bemol o sostenido. En el arpa de palancas o arpa celta, cada cuerda se ajusta individualmente, pero en las arpas de pedales todas las cuerdas con igual nombre de nota se controlan por un único pedal. Visto de izquierda a derecha desde la posición del ejecutante, los pedales son Re, Do y Si a la izquierda y Mi, Fa, Sol y La a la derecha. La posición de los pedales se puede indicar con indicaciones textuales:

```
\textLengthOn
cis1_\markup \concat \vcenter {
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c!1_\markup \concat \vcenter {
  [ C \natural ] }
```



o mediante diagramas de pedal:

```
\textLengthOn
cis1_\markup { \harp-pedal #"^v-|vv-^" }
c!1_\markup { \harp-pedal #"^o--|vv-^" }
```
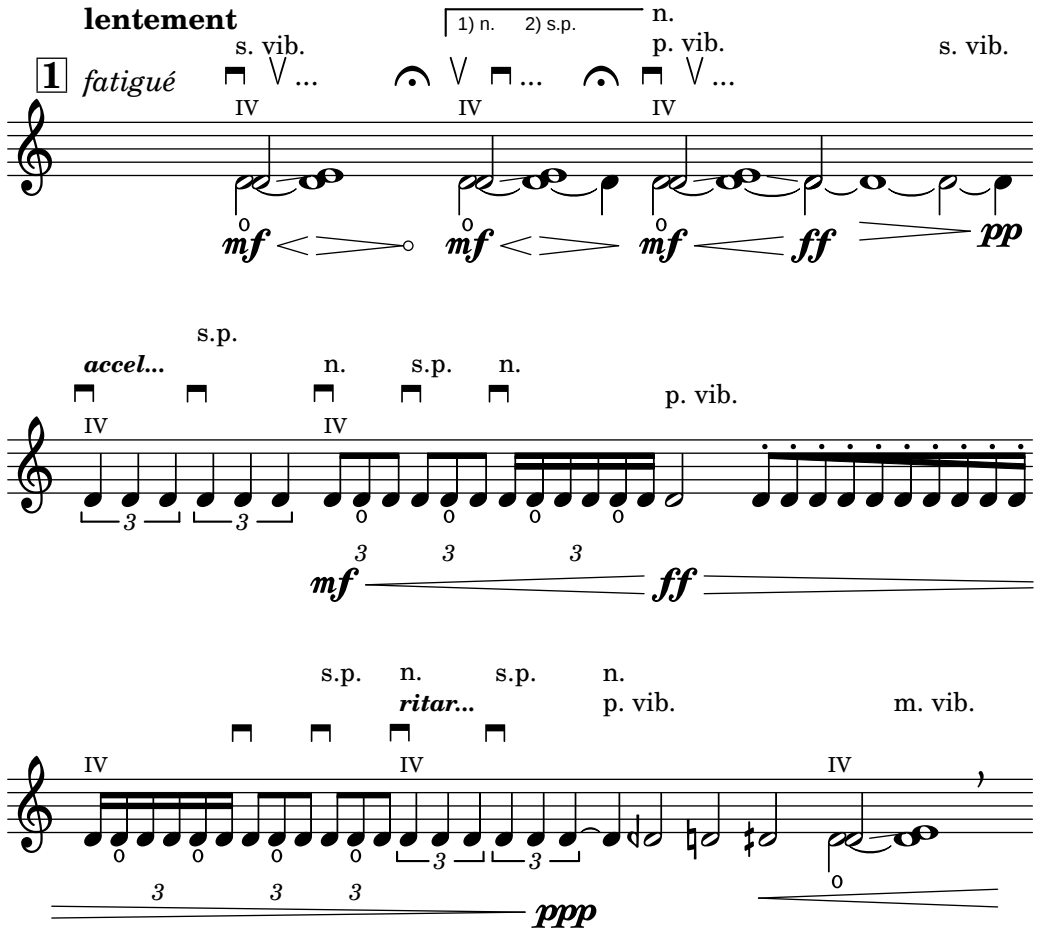


La instrucción `\harp-pedal` acepta una cadena de caracteres en la que `^` es la posición alta del pedal (altura bemol), `-` es la posición intermedia (altura natural), `v` es la posición baja (altura sostenido), y `|` es la línea vertical separadora. Si se antepone `o`, el siguiente símbolo de pedal se rodea por una elipse.

Véase también

Referencia de la notación: [Guiones de texto], página 226, Sección A.10.5 [Instrument Specific Markup], página 708.

2.3 Instrumentos de cuerda sin trastes



Esta sección aporta información y referencias que resultan útiles si se escribe música para instrumentos de cuerda sin trastes, principalmente instrumentos de cuerda orquestales.

2.3.1 Notación común para cuerdas sin trastes

Hay poca notación especializada para los instrumentos de cuerdas sin trastes. La notación de la música se realiza en un solo pentagrama, y se suele usar una sola voz. Se pueden necesitar dos voces para algunos pasajes en dobles cuerdas o en divisi.

Referencias para cuerdas sin trastes

La mayor parte de la notación que resulta de utilidad para las cuerdas orquestales y otros instrumentos de arco se estudia en otras partes del manual:

- Las indicaciones textuales como “pizz.” y “arco” se añaden como texto simple: véase [Guiones de texto], página 226.
- Las digitaciones, entre ellas la indicación de pulgar, se describen en [Indicaciones de digitación], página 214.
- Las dobles cuerdas se indican normalmente escribiendo un acorde, véase [Notas en acorde], página 161. Hay otras indicaciones que se pueden añadir para la interpretación de acordes, véase [Arpegio], página 138.
- Hay plantillas para cuarteto de cuerda en Sección “Plantillas de cuarteto de cuerda” in *Manual de Aprendizaje*. Otras se muestran en la sección de fragmentos de código.

Véase también

Manual de aprendizaje: Sección “Platillas de cuarteto de cuerda” in *Manual de Aprendizaje*.

Referencia de la notación: [Guiones de texto], página 226, [Indicaciones de digitación], página 214, [Notas en acorde], página 161, [Arpegio], página 138.
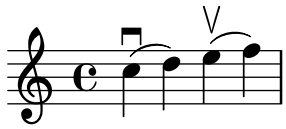
Fragmentos de código: Sección “Unfretted strings” in *Fragmentos de código*.

Indicaciones de arco

Las indicaciones de arco se crean como articulaciones, que se describen en [Articulaciones y ornamentos], página 116.
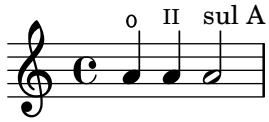
Las instrucciones de arco `\upbow` (arco arriba) y `\downbow` (arco abajo) se usan con ligaduras de expresión de la siguiente manera:

```
c4(\downbow d) e(\upbow f)
```



y el ejemplo siguiente muestra tres formas distintas de indicar un La sobre una cuerda al aire del violín:

```
a4 \open
a^\markup { \teeny "II" }
a2^\markup { \small "sul A" }
```



Instrucciones predefinidas

`\downbow`, `\upbow`, `\open`.

Véase también
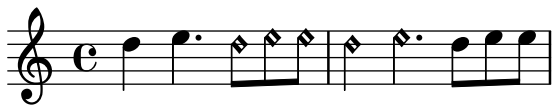
Referencia de la notación: [Articulaciones y ornamentos], página 116, [Ligaduras de expresión], página 127.

Armónicos

Armónicos naturales

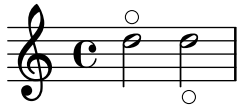
La notación de los armónicos naturales se puede realizar de varias formas. Generalmente, una nota con la cabeza en forma de rombo significa tocar (sin apretar) la cuerda en el lugar donde se pisaría la nota si no fuese un rombo.

```
d4 e4.
\harmonicsOn
d8 e e
d4 e4.
\harmonicsOff
d8 e e
```



Como posibilidad alternativa está la de mostrar una cabeza normal de nota en la altura de la nota que debe sonar, con un pequeño círculo que indica que se debe tocar como armónico:

```
d2^\flageolet d_\flageolet
```
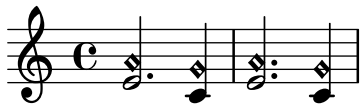


Se puede hacer un círculo de menor tamaño, véase la lista de fragmentos de código en [Referencias para cuerdas sin trastes], página 326.

Armónicos artificiales

La notación de los armónicos artificiales se realiza con dos notas, una con una cabeza normal que indica la posición donde se pisa, y otra con una cabeza en forma de rombo hueco para indicar la posición donde se roza la cuerda (sin pisar) para producir el armónico.

Los armónicos artificiales indicados con `\harmonic` no presentan los puntillos. La propiedad de contexto `harmonicDots` se debe activar si se necesitan los puntillos.

```
<e a\harmonic>2.  <c g'\harmonic>4
\set harmonicDots = ##t
<e a\harmonic>2.  <c g'\harmonic>4
```



Nota: `\harmonic` **se debe** colocar dentro de una construcción de acorde incluso si solamente hay una nota. Normalmente se usaría `\harmonicsOn` en esta situación.

Véase también

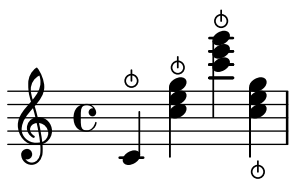
Glosario musical: Sección “harmonics” in *Glosario Musical*.

Referencia de la notación: [Cabezas de nota especiales], página 35, [Referencias para cuerdas sin trastes], página 326.
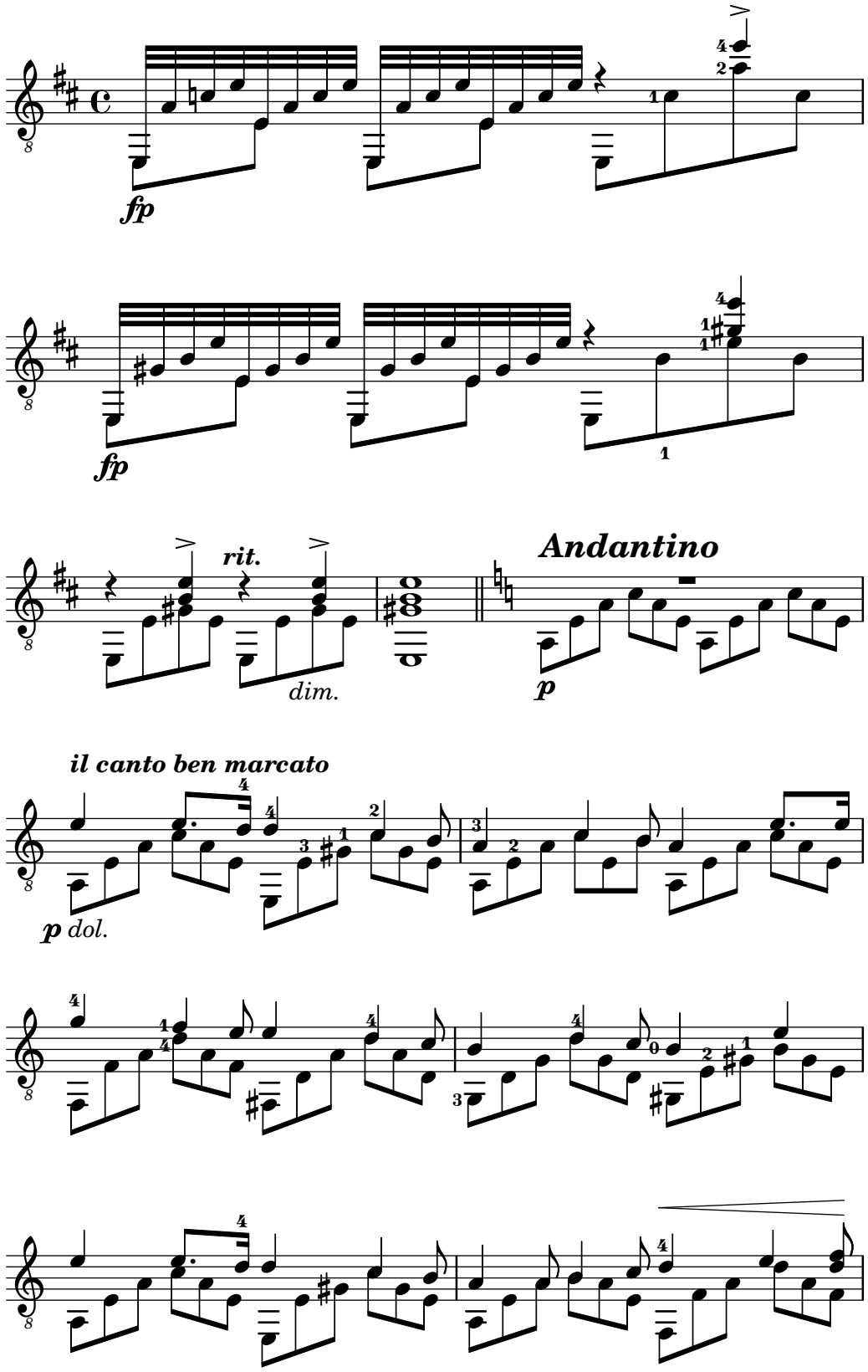
Snap (Bartók) pizzicato

El *snap pizzicato* (conocido también como “pizzicato de Bartók”) es un tipo de pizzicato en el que la cuerda se pulsa hacia arriba (y no lateralmente) de forma deliberada, de forma que golpea la madera del diapasón.

```
c4\snappizzicato
<c' e g>4\snappizzicato
<c' e g>4^\snappizzicato
<c, e g>4_\snappizzicato
```



2.4 Instrumentos de cuerda con trastes



Esta sección trata varios aspectos de la notación musical que son exclusivos de los instrumentos de cuerda con trastes.

2.4.1 Notación común para cuerdas con trastes

Esta sección se ocupa de la notación común exclusiva de los instrumentos de cuerda con trastes.

Referencias para cuerdas con trastes

La notación musical para instrumentos de cuerda con trastes se realiza normalmente en una sola pauta, ya sea en notación musical tradicional o en tablatura. A veces se combinan los dos tipos, y es común en música popular utilizar diagramas de acordes por encima de un pentagrama de notación tradicional. La guitarra y el banjo son instrumentos transpositores, que suenan una octava por debajo de lo escrito. Las partituras para estos instrumentos deben usar la clave de Sol octava baja `"treble_8"` (o `\transposition c` para obtener una salida MIDI correcta). Otros elementos relativos a los instrumentos de cuerda con trastes se estudian en otras partes de manual:

- Las digitaciones se indican como se explica en [Indicaciones de digitación], página 214.
- Las instrucciones para ligaduras *Laissez vibrer* así como ligaduras de unión sobre arpegios y trémolos están en [Ligaduras de unión], página 51.
- Las instrucciones para el manejo de varias voces se encuentran en [Resolución de las colisiones], página 170.
- Las instrucciones para la indicación de armónicos está en [Armónicos], página 327.

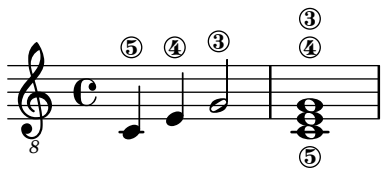
Véase también

Referencia de la notación: [Indicaciones de digitación], página 214, [Ligaduras de unión], página 51, [Resolución de las colisiones], página 170, [Nombres de instrumentos], página 201, [Escribir música en paralelo], página 180, [Arpegio], página 138, Sección A.13 [Lista de articulaciones], página 725, [Clave], página 16, [Transposición de los instrumentos], página 24.

Indicación de los números de cuerda

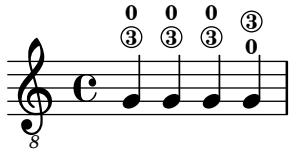
Se puede indicar la cuerda en que se debe tocar una nota añadiendo `\` *número* a una nota.

```
\clef "treble_8"
c4\5 e\4 g2\3
<c,\5 e\4 g\3>1
```



Cuando se usan al mismo tiempo digitaciones e indicaciones del número de cuerda, su colocación se puede controlar por el orden en que aparecen estos elementos en el código *solamente* si aparecen dentro de un acorde explícito: aplicado a acordes completos o notas individuales *fuera* de acordes, las digitaciones se colocan usando un mecanismo distinto.

```
\clef "treble_8"
g4\3-0
g-0\3
<g\3-0>
<g-0\3>
```

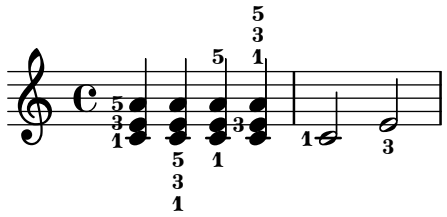


Fragmentos de código seleccionados

Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación. Para que se tenga en cuenta la orientación de las digitaciones, se debe utilizar una construcción de acorde `<>` aunque sea una sola nota.

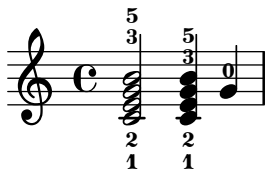
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Nota: se debe usar una construcción de acorde `<>`, aunque sea una sola nota.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



Véase también

Referencia de la notación: [Indicaciones de digitación], página 214.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

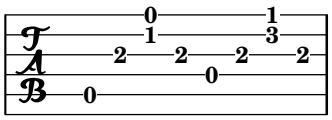
Referencia de funcionamiento interno: Sección “StringNumber” in *Referencia de Funcionamiento Interno*, Sección “Fingering” in *Referencia de Funcionamiento Interno*.

Tablaturas predeterminadas

La notación para instrumentos de cuerda pulsada se realiza con frecuencia utilizando un sistema de dedos y cifras, o tablatura. A diferencia de la notación tradicional, las notas no se designan mediante figuras, sino mediante números (o símbolos semejantes a letras en la intavolatura histórica). Las líneas de la tablatura indican sobre qué cuerda se debe tocar la nota, y el número sobre una línea indica el traste en que la cuerda correspondiente debe pisarse. Las notas que se deben tocar simultáneamente se alinean en sentido vertical.

De forma predeterminada, la cuerda 1 es la más aguda y corresponde a la línea superior de la pauta `TabStaff`. La afinación predeterminada de las cuerdas del `TabStaff` es la afinación estándar de guitarra (con 6 cuerdas). Las notas se imprimen como tablatura, usando los contextos `TabStaff` y `TabVoice`. Se añade automáticamente una clave caligráfica de tablatura.

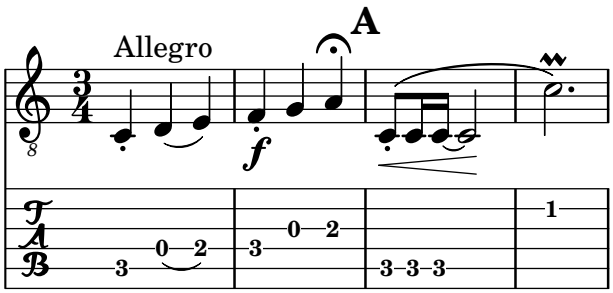
```
\new TabStaff \relative c' {
  a,8 a' <c e> a
  d,8 a' <d f> a
}
```



Las tablaturas predeterminadas no contienen símbolos para la duración de las notas ni ningún otro símbolo musical, como p.ej. indicaciones expresivas.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c~ c2\!
  c'2.\prall\)
}

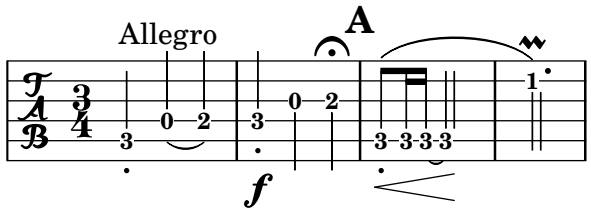
\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff { \symbols }
  >>
}
```



Si queremos que todos los símbolos musicales que se usan en la notación tradicional aparezcan también en la tablatura, tenemos que aplicar la instrucción `\tabFullNotation` dentro de un contexto `TabStaff`. Tenga en cuenta que, en la tablatura, las blancas tienen doble plica para poder distinguirlas de las negras.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c~ c2\!
  c'2.\prall\)
}

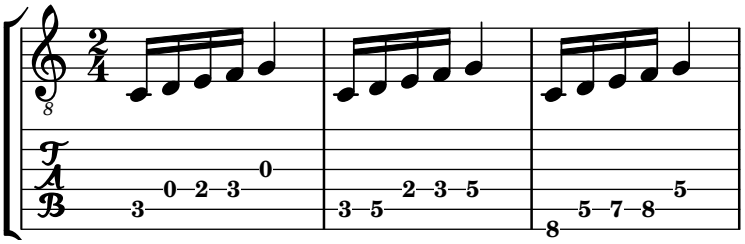
\score {
  \new TabStaff {
    \tabFullNotation
    \symbols
  }
}
```



De forma predeterminada, las notas se asignan a la posición más baja del mástil (primera posición). Las cuerdas al aire se toman como preferencia, automáticamente. Si queremos que una nota determinada se toque sobre una cuerda concreta, podemos añadir una indicación de número de cuerda al nombre de la nota. Si no queremos que las indicaciones de número de cuerda aparezcan en la notación tradicional, podemos sobreescribir el sello correspondiente. Generalmente es mucho más cómodo definir la posición utilizando el valor de `minimumFret`. El valor predeterminado de es 0.

Incluso si `minimumFret` está establecido, se usan las cuerdas al aire siempre que sea posible. Este comportamiento se puede cambiar fijando `restrainOpenStrings` al valor `#t`.

```
\layout { \omit Voice.StringNumber }
\new StaffGroup <<
   \new Staff \relative c {
     \clef "treble_8"
     \time 2/4
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     c,16 d e f g4
   }
   \new TabStaff \relative c {
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     \set TabStaff.minimumFret = #5
     \set TabStaff.restrainOpenStrings = ##t
     c,16 d e f g4
   }
>>
```



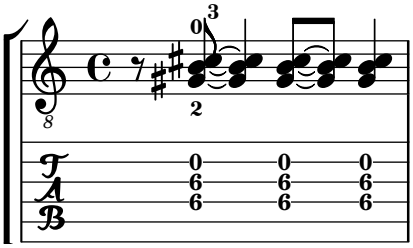
Las construcciones de acorde se pueden repetir mediante el símbolo de repetición de acordes `q`. En combinación con las tablaturas, su comportamiento de eliminar números de cuerda y de dedos junto a otros eventos es fastidioso, por lo que quizá quiera ejecutar

```
\chordRepeats #'(string-number-event fingering-event)
```

explícitamente sobre expresiones musicales en las tablaturas que utilicen [Repetición de acordes], página 163. Esta instrucción en particular es tan común que está disponible como `\tabChordRepeats`.

```
guitar = \relative c' {
  r8 <gis-2 cis-3 b-0>~ q4 q8~ q q4
}

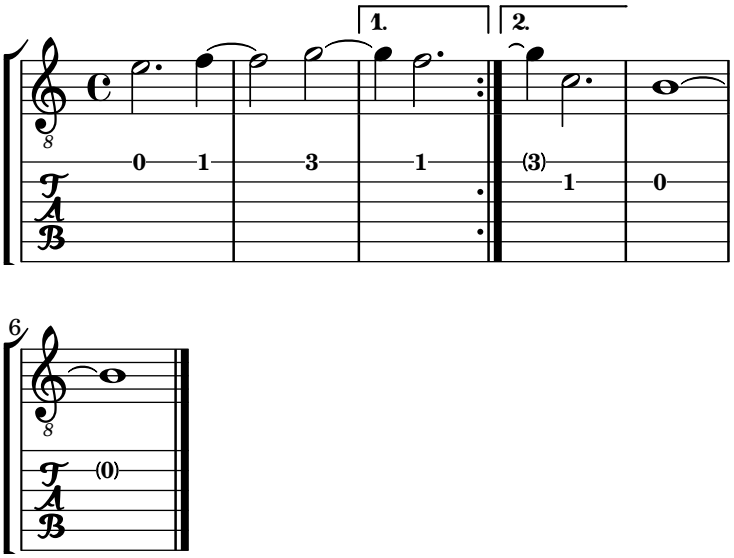
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \guitar
  }
  \new TabStaff {
    \tabChordRepeats \guitar
  }
>>
```



Las ligaduras de unión que cruzan saltos de línea llevan paréntesis de forma predeterminada. Lo mismo rige para la casilla de segunda vez de una repetición.

```
ties = \relative c' {
  \repeat volta 2 {
    e2. f4~
    f2 g2~
  }
  \alternative {
     { g4 f2. }
     { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|."
}
```

```
\score {
  <<
    \new StaffGroup  <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
        \ties
      }
    >>
  >>
  \layout {
  indent = #0
  ragged-right = ##t
  }
}
```

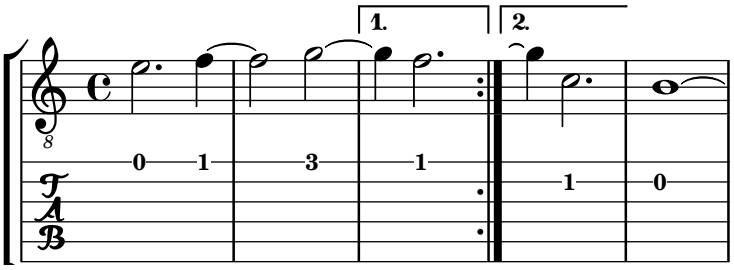


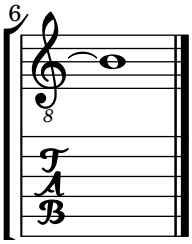
La instrucción `\hideSplitTiedTabNotes` cancela el comportamiento de imprimir los números de traste entre paréntesis:

```
ties = \relative c' {
  \repeat volta 2 {
    e2. f4~
    f2 g2~ }
  \alternative {
    { g4 f2. }
    { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|."
}

\score {
```

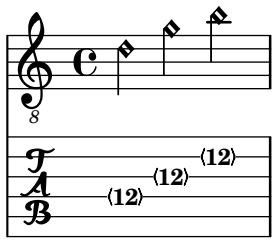
```
  <<
    \new StaffGroup  <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
      \hideSplitTiedTabNotes
        \ties
      }
    >>
  >>
  \layout {
  indent = #0
  ragged-right = ##t
  }
}
```





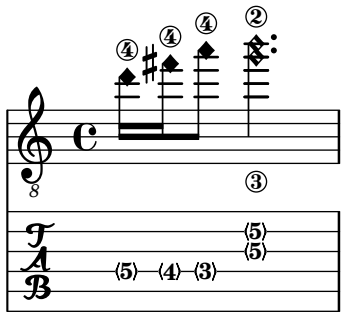
Se pueden añadir indicaciones de armónicos a la notación de tablatura como sus alturas de sonido:

```
\layout { \omit Voice.StringNumber }
firstHarmonic = {
  d'4\4\harmonic
  g'4\3\harmonic
  b'2\2\harmonic
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \firstHarmonic
    }
    \new TabStaff { \firstHarmonic }
  >>
}
```



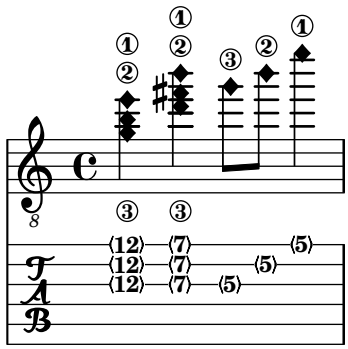
Observe que la instrucción `\harmonic` se debe adjuntar siempre a notas individuales (posiblemente dentro de un acorde) en lugar de a acordes completos. Sólo tiene sentido para armónicos sobre una cuerda abierta en el duodécimo traste. Cualquier otro armónico debe estar calculado por parte de LilyPond. Se puede conseguir esto indicando el traste en que debe rozar la cuerda el dedo de la mano que pisa.

```
fretHarmonics = {
  \harmonicByFret #5 d16\4
  \harmonicByFret #4 d16\4
  \harmonicByFret #3 d8\4
  \harmonicByFret #5 <g\3 b\2>2.
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \fretHarmonics
    }
    \new TabStaff { \fretHarmonics }
  >>
}
```



De forma alternativa, se pueden calcular los armónicos definiendo la razón de las longitudes de cuerda por encima y por debajo de la digitación de armónico.

```
ratioHarmonics = {
  \harmonicByRatio #1/2 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/3 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/4 { g8\3 b8\2 e'4\1 }
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \ratioHarmonics
    }
    \new TabStaff { \ratioHarmonics }
  >>
}
```

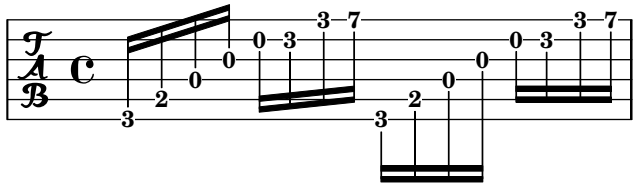


Fragmentos de código seleccionados

Comportamiento de las plicas y las barras de corchea en tablaturas

La dirección de las plicas se contola de la misma forma en la tablatura que en la notación tradicional. Las barras se pueden poner horizontales, como se muestra en este ejemplo.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = #10000
    g,,16 b d g b d g b
  }
}
```



Polifonía en tablaturas

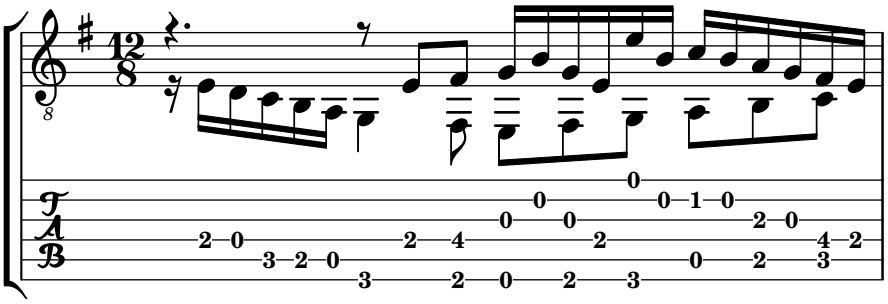
La polifonía se crea de la misma forma en un `TabStaff` que en una pauta normal.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  <<
    \new StaffGroup = "tab with traditional" <<
      \new Staff = "guitar traditional" <<
        \clef "treble_8"
        \context Voice = "upper" \upper
        \context Voice = "lower" \lower
```

```
      >>
      \new TabStaff = "guitar tab" <<
        \context TabVoice = "upper" \upper
        \context TabVoice = "lower" \lower
      >>
    >>
  >>
}
```



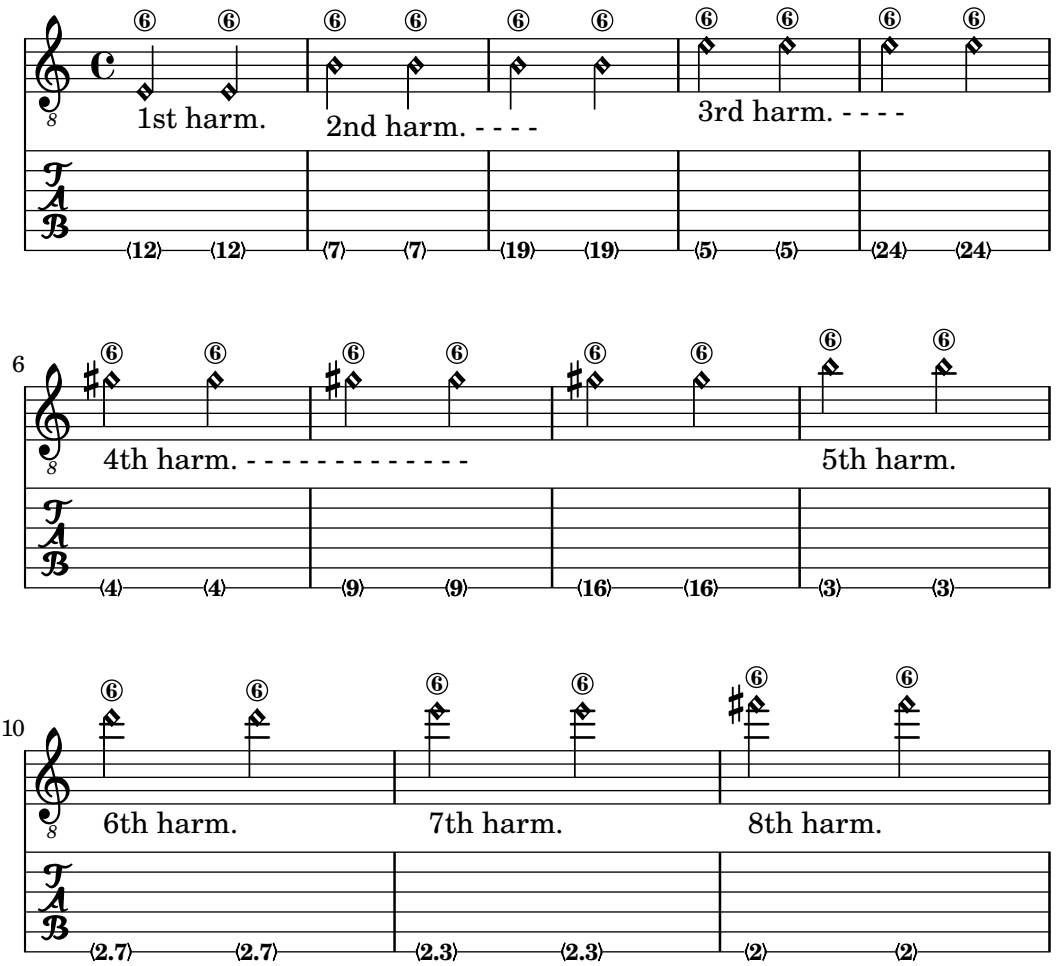
Referencia para armónicos sobre cuerdas al aire

Referencia para armónicos sobre cuerdas al aire (armónicos naturales):

```
openStringHarmonics = {
  %first harmonic
  \harmonicByFret #12 e,2\6_\markup{"1st harm."}
  \harmonicByRatio #1/2 e,\6
  %second harmonic
  \harmonicByFret #7 e,\6_\markup{"2nd harm. - - - -"}
  \harmonicByRatio #1/3 e,\6
  \harmonicByFret #19 e,\6
  \harmonicByRatio #2/3 e,\6
  %\harmonicByFret #19 < e,\6 a,\5 d\4 >
  %\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >
  %third harmonic
  \harmonicByFret #5 e,\6_\markup{"3rd harm. - - - -"}
  \harmonicByRatio #1/4 e,\6
  \harmonicByFret #24 e,\6
  \harmonicByRatio #3/4 e,\6
  \break
  %fourth harmonic
  \harmonicByFret #4 e,\6_\markup{"4th harm. - - - - - - - - - - - - -"}
  \harmonicByRatio #1/5 e,\6
  \harmonicByFret #9 e,\6
  \harmonicByRatio #2/5 e,\6
  \harmonicByFret #16 e,\6
  \harmonicByRatio #3/5 e,\6
  %fifth harmonic
  \harmonicByFret #3 e,\6_\markup{"5th harm."}
  \harmonicByRatio #1/6 e,\6
  \break
  %sixth harmonic
  \harmonicByFret #2.7 e,\6_\markup{"6th harm."}
  \harmonicByRatio #1/7 e,\6
```

```
  %seventh harmonic
  \harmonicByFret #2.3 e,\6_\markup{"7th harm."}
  \harmonicByRatio #1/8 e,\6
  %eighth harmonic
  \harmonicByFret #2 e,\6_\markup{"8th harm."}
  \harmonicByRatio #1/9 e,\6
}

\score {
  <<
    \new Staff {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \openStringHarmonics
      }
    }
  >>
}
```



Armónicos sobre cuerdas pisadas en tablatura

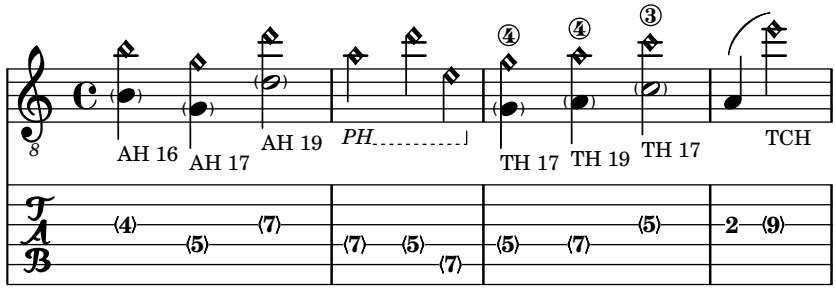
Armónicos sobre cuerdas pisadas (armónicos artificiales):

```
pinchedHarmonics = {
   \textSpannerDown
   \override TextSpanner.bound-details.left.text =
      \markup {\halign #-0.5 \teeny "PH" }
      \override TextSpanner.style =
         #'dashed-line
   \override TextSpanner.dash-period = #0.6
   \override TextSpanner.bound-details.right.attach-dir = #1
   \override TextSpanner.bound-details.right.text =
      \markup { \draw-line #'(0 . 1) }
   \override TextSpanner.bound-details.right.padding = #-0.5
}

harmonics = {
  %artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup{ \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup{ \teeny "AH 17" }
  <\parenthesize d' d''\harmonic>2_\markup{ \teeny "AH 19" }
  %pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d''\harmonic>4
  <e'\harmonic>4\stopTextSpan
  %tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4_\markup{ \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4_\markup{ \teeny "TH 19" }
  <\parenthesize c'\3 c''\harmonic>2_\markup{ \teeny "TH 17" }
  %touch harmonics (TCH)
  a4( <e''\harmonic>2. )_\markup{ \teeny "TCH" }
}

frettedStrings = {
  %artificial harmonics (AH)
  \harmonicByFret #4 g4\3
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 g2\3
  %pinched harmonics (PH)
  \harmonicByFret #7 d2\4
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 a4\5
  %tapped harmonics (TH)
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 d4\4
  \harmonicByFret #5 g2\3
  %touch harmonics (TCH)
  a4 \harmonicByFret #9 g2.\3
}
```

```
\score {
  <<
    \new Staff {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}
```



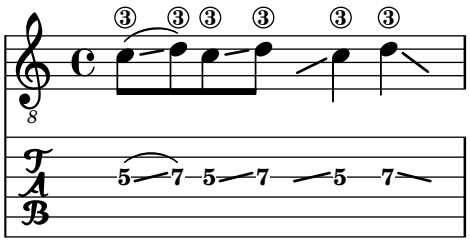
Deslizamientos en tablatura

Los deslizamientos se pueden componer tipográficamente tanto en los contextos de `Staff` como en los de `TabStaff`:

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
  \stemDown \hideNotes
  g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>
  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = #4
      \override Glissando.springs-and-rods =
```

```
                          #ly:spanner::set-spacing-rods
      \override Glissando.thickness = #2
    }
  }
}
```

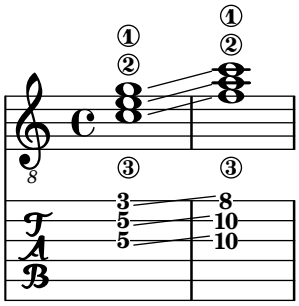


Glissando de acordes en tablatura

Los deslizamientos para acordes se pueden indicar tanto en el contexto `Staff` como en `TabStaff`. Los números de cuerda son necesarios para TabStaff porque los cálculos de cuerda automáticos son diferentes para los acordes y para notas sueltas.

```
myMusic = \relative c' {
  <c\3 e\2 g\1>1 \glissando <f\3 a\2 c\1>
}

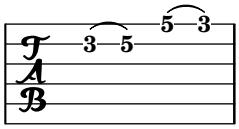
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
    }
    \new TabStaff {
      \myMusic
    }
  >>
}
```



Hammer on and pull off

Hammer-on and pull-off can be obtained using slurs.

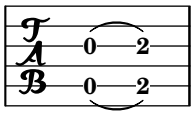
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Hammer on and pull off using voices

The arc of hammer-on and pull-off is upwards in voices one and three and downwards in voices two and four:

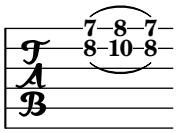
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Hammer on and pull off using chords

When using hammer-on or pull-off with chorded notes, only a single arc is drawn. However 'double arcs' are possible by setting the `doubleSlurs` property to `#t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Véase también

Referencia de la notación: [Repetición de acordes], página 163, [Repeticiones explícitas], página 154, [Plicas], página 219, [Armónicos], página 327, [Glissando], página 134.

Fragmentos de código: Sección "Fretted strings" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TabNoteHead" in *Referencia de Funcionamiento Interno*, Sección "TabStaff" in *Referencia de Funcionamiento Interno*, Sección "TabVoice" in *Referencia de Funcionamiento Interno*, Sección "Beam" in *Referencia de Funcionamiento Interno*.

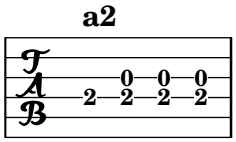
Advertencias y problemas conocidos

Los acordes no se tratan de una forma especial, y de aquí que el selector automático de la cuerda puede elegir fácilmente la misma cuerda para dos notas del acorde.

Para manejar `\partcombine`, es necesario que `TabStaff` utilice voces especialmente creadas:

```
melodia = \partcombine { e4 g g g } { e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "one" s1
```

```
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



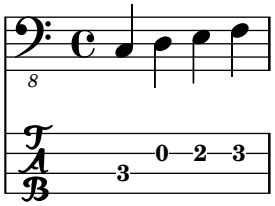
Los efectos especiales de guitarra se limitan a armónicos y *slides*.

Tablaturas personalizadas

La tablatura en LilyPond calcula automáticamente el traste para cada nota, basándose en la cuerda a que está asignada la nota. Para hacerlo, es necesario especificar la afinación de las cuerdas. La afinación de las cuerdas se da en la propiedad `stringTunings`.

LilyPond trae afinaciones predefinidas para el banjo, la mandolina, la guitarra, el bajo, el ukelele, el violín, la viola, el violoncello y el contrabajo. LilyPond establece automáticamente la transposición correcta para las afinaciones predefinidas. El ejemplo siguiente es para bajo, que suena una octava por debajo de lo escrito.

```
<<
  \new Voice \with {
    \omit StringNumber
  } {
    \clef "bass_8"
    \relative c, {
      c4 d e f
    }
  }
  \new TabStaff \with {
    stringTunings = #bass-tuning
  } {
    \relative c, {
      c4 d e f
    }
  }
>>
```



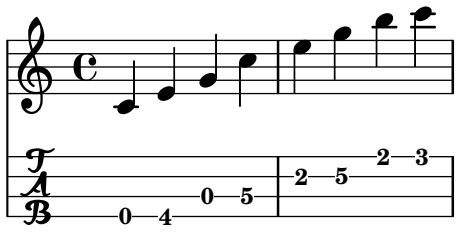
La afinación por omisión es `guitar-tuning` (la afinación estándar Mi-La-Re-Sol-Si-Mi). Otras afinaciones predefinidas son `guitar-open-g-tuning` (Sol Mayor al aire, Re-Sol-Re-Sol-Si-Re), `mandolin-tuning` (mandolina) y `banjo-open-g-tuning` (banjo con Sol Mayor al aire). Las afinaciones predefinidas están en '`ly/string-tunings-init.ly`'.

Es posible crear cualquier afinación de cuerdas que se desee. Se puede usar la función `\stringTuning` para definir una afinación de cuerdas y que puede usarse para establecerla como el valor de `stringTunings` para el contexto actual.

Su argumento es una construcción de acorde que define la altura de las notas de cada cuerda al aire en dicha afinación. La construcción de acorde debe estar en el modo de octava absoluto, véase [Escritura de octava absoluta], página 1. La cuerda con el número más alto (generalmente la más grave) debe ir primero en el acorde. Por ejemplo, podemos definir una afinación para un instrumento de cuatro cuerdas con las alturas `a''`, `d''`, `g'` y `c'`:

```
mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \mynotes
  }
  \new TabStaff {
    \set Staff.stringTunings = \stringTuning <c' g' d'' a''>
    \mynotes
  }
>>
```



La propiedad `stringTunings` también se utiliza por parte de `FretBoards` para calcular los diagramas de posición automáticos.

Las afinaciones de cuerdas se utilizan como parte de la clave de asociación para los diagramas de posición predefinidos (véase [Diagramas predefinidos de trastes], página 357).

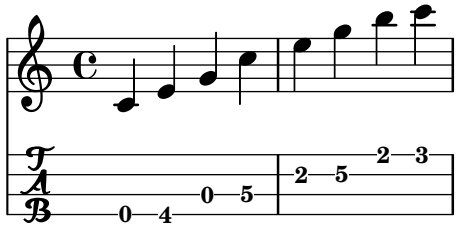
El ejemplo anterior también se podría haber escrito como sigue:

```
custom-tuning = \stringTuning <c' g' d'' a''>

mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \mynotes
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #custom-tuning
```

```
    \mynotes
  }
>>
```



Internamente, una afinación de las cuerdas es una lista de Scheme de alturas, una por cada cuerda, ordenadas de 1 a N, donde la cuerda 1 está arriba en la pauta de tablatura y la cuerda N está abajo. Esto resulta generalmente en un orden de la nota más aguda a la más grave, pero algunos instrumentos (p.ej. el ukelele) no tienen las cuerdas dispuestas en orden de altura.

La altura de una cuerda dentro de una lista de alturas de cuerdas es un objeto altura de LilyPond. Los objetos altura se crean con la función de Scheme `ly:make-pitch` (véase Sección A.21 [Funciones de Scheme], página 773).

`\stringTuning` crea dicho objeto a partir de la entrada de acordes.

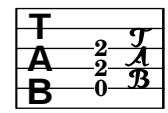
LilyPond calcula automáticamente el número de líneas dentro del `TabStaff` y el número de cuerdas dentro de un `FretBoard` calculado automáticamente como el número de elementos de `stringTunings`.

Para permitir a todos los contextos TabStaff usar la misma afinación personalizada por omisión, podemos usar

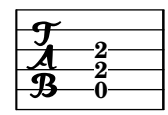
```
\layout {
  \context {
    \TabStaff
    stringTunings = \stringTuning <c' g' d'' a''>
  }
}
```

También se puede usar una clave moderna de tablatura.

```
\new TabStaff {
  \clef moderntab
  <a, e a>1
  \break
  \clef tab
  <a, e a>1
}
```



2



La clave moderna de tablatura contempla tablaturas de 4 a 7 cuerdas.

Véase también

Referencia de la notación: [Escritura de octava absoluta], página 1, [Diagramas predefinidos de trastes], página 357, Sección A.21 [Funciones de Scheme], página 773.

Archivos instalados: ‘`ly/string-tunings-init.ly`’, ‘`scm/tablature.scm`’.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Tab_note_heads_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

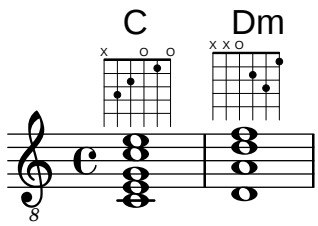
Los cálculos de tablatura automáticos no funcionan bien en la mayoría de los casos para instrumentos en que la afinación de las cuerdas no varía de forma monótona con el número de cuerda, como el ukelele.

Marcas de diagramas de trastes

Se pueden añadir diagramas de trastes a la música como elementos de marcado sobre la nota deseada. El marcado contiene información sobre el diagrama de trastes deseado. Existen tres interfaces distintos de marcado de diagramas de trastes: standard (estándar), terse (escueto) y verbose (prolijo). Los tres interfaces producen marcados equivalentes, pero tienen cantidades variables de información en la cadena de marcado. Hay más detalles sobre la sintaxis de las distintos cadenas de marcado que se usan para definir diagramas de posiciones en Sección A.10.5 [Instrument Specific Markup], página 708.

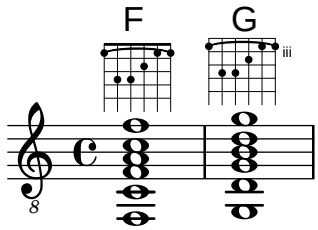
La cadena de marcado de los diagramas estándar de trastes indica el número de la cuerda y el número de traste en que se coloca cada uno de los puntos sobre la cuerda. Además se pueden indicar cuerdas al aire y cuerdas mudas (que no se tocan).

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram #"6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
>>
```



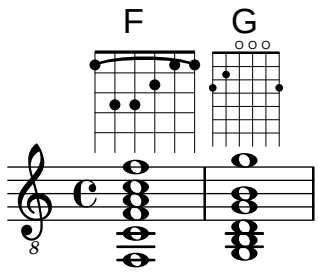
Las indicaciones de cejilla se pueden añadir al diagrama a partir de la cadena de marcado del diagrama de trastes.

```
<<
  \new ChordNames {
     \chordmode {
       f1 g
     }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram #"c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram #"c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
    }
  }
>>
```



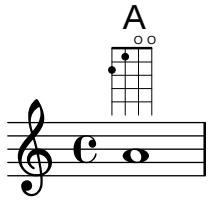
El tamaño del diagrama de trastes y el número de trastes del diagrama, se puede cambiar en la cadena de marcado `fret-diagram`.

```
<<
  \new ChordNames {
     \chordmode {
       f1 g
     }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram #"s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram #"h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
    }
  }
>>
```



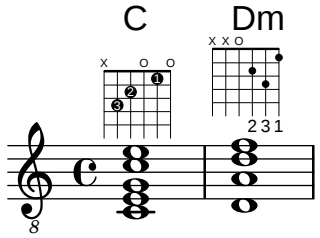
Se puede cambiar el número de cuerdas de un diagrama de trastes para que se adapte a distintos instrumentos como el bajo y el ukelele, con la cadena de marcado del diagrama.

```
<<
  \new ChordNames {
    \chordmode {
      a1
    }
  }
  \new Staff {
    % An 'A' chord for ukulele
    a'1^\markup {
      \fret-diagram #"w:4;4-2-2;3-1-1;2-o;1-o;"
    }
  }
>>
```



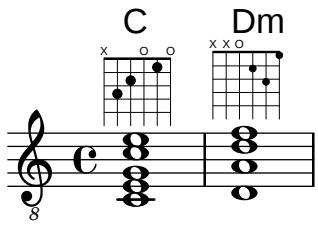
Se pueden añadir indicaciones de digitación, y la posición de las etiquetas de los dedos se puede controlar mediante la cadena de marcado del diagrama.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram #"f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram #"f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
    }
  }
>>
```



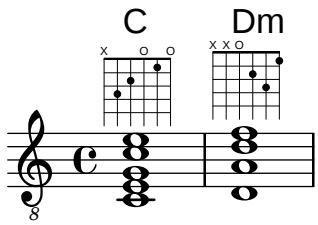
Se puede controlar el radio y la posición de los puntos con la cadena de marcado `fret-diagram`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram #"d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram #"p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
>>
```



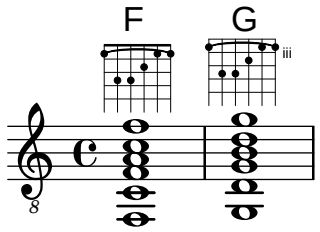
La cadena de marcado de diagrama de trastes escueta, `fret-diagram-terse`, omite los números de la cuerda; el número de cuerda viene implícito por la presencia del punto y coma. Hay un punto y coma por cada cuerda del diagrama. El primer punto y coma corresponde al número de cuerda más alto, y el último punto y coma corresponde a la primera cuerda. Se pueden indicad cuerdas mudas, al aire y números de traste.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-terse #"x;3;2;o;1;o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram-terse #"x;x;o;2;3;1;"
    }
  }
>>
```



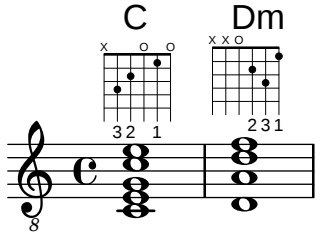
Se pueden incluir números de cejilla en la cadena de marcado escueta `fret-diagram-terse`.

```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram-terse #"1-(;3;3;2;1;1-);"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram-terse #"3-(;5;5;4;3;3-);"
    }
  }
>>
```



Se pueden incluir indicaciones de digitación en la cadena de marcado escueta `fret-diagram-terse`

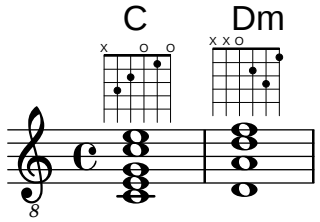
```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram-terse #"x;x;o;2-2;3-3;1-1;"
    }
  }
>>
```



Otras propiedades de los diagramas de trastes se deben ajustar usando `\override` al utilizar el marcado escueto con fret-diagram-terse.

La cadena de marcado prolija `fret-diagram-verbose` está en el formato de una lista de Scheme. Cada elemento de la lista indica la colocación un elemento sobre el diagrama de trastes.

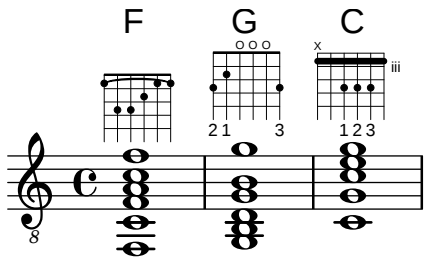
```
<<
    \new ChordNames {
      \chordmode {
        c1 d:m
      }
    }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    }
    <d a d' f'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
      )
    }
  }
>>
```



Se pueden incluir digitaciones y cejillas en una cadena de marcado prolija `fret-diagram-verbose`. Es exclusiva del interfaz de fret-diagram-verbose la indicación 'capo' que se puede colocar sobre el diagrama de posición. La indicación de capo es una barra gruesa que cubre todas las cuerdas. El traste que tiene el capo es el más bajo del diagrama de posición.

```
<<
    \new ChordNames {
      \chordmode {
        f1 g c
      }
```

```
    }
  \new Staff {
    \clef "treble_8"
    \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
    <f, c f a c' f'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
      )
    }
    <c g c' e' g'>1^\markup {
      \fret-diagram-verbose #'(
        (capo 3)
        (mute 6)
        (place-fret 4 5 1)
        (place-fret 3 5 2)
        (place-fret 2 5 3)
      )
    }
  }
>>
```



Todas las otras propiedades de diagrama de trastes se deben ajustar utilizando `\override` cuando se usa el marcado prolijo con fret-diagram-verbose.

El aspecto gráfico de un diagrama de trastes se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*. Para un marcado de diagrama de trastes, las propiedades del interface pertenecen a `Voice.TextScript`.

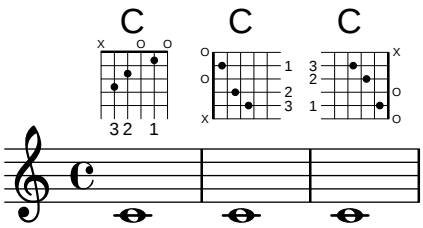
Fragmentos de código seleccionados

Modificar la orientación de los trastes

Los diagramas de posiciones de acordes se pueden orientar de tres formas. De manera predeterminada se alinena la cuerda o traste superior en las distintas alineaciones.

```
\include "predefined-guitar-fretboards.ly"

<<
  \chords {
    c1
    c1
    c1
  }
  \new FretBoards {
    \chordmode {
      c1
      \override FretBoard.fret-diagram-details.orientation =
        #'landscape
      c1
      \override FretBoard.fret-diagram-details.orientation =
        #'opposing-landscape
      c1
    }
  }
  \new Voice {
    c'1
    c'1
    c'
  }
>>
```



Personalizar diagramas de posiciones de marcado

Se pueden establecer las propiedades de los diagramas de posiciones a través de `'fret-diagram-details`. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobreescrituras) al objeto `Voice.TextScript` o directamente al elemento de marcado.

```
<<
  \chords { c1 | c | c | d }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = #'1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white
```

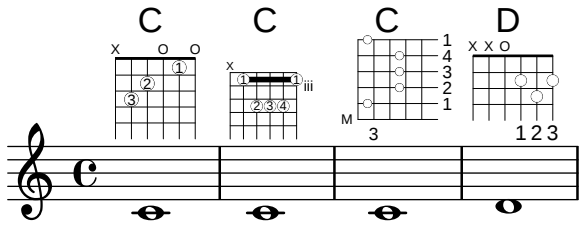
```
%% C major for guitar, no barre, using defaults
   % terse style
c'1^\markup { \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;" }

%% C major for guitar, barred on third fret
   % verbose style
   % size 1.0
   % roman fret label, finger labels below string, straight barre
c'1^\markup {
  % standard size
  \override #'(size . 1.0) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . in-dot)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
   % verbose style
   % landscape orientation, arabic numbers, M for mute string
   % no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (number-type . arabic)
               (label-dir . -1)
               (mute-string . "M")
               (orientation . landscape)
               (barre-type . none)
               (xo-font-magnification . 0.4)
               (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
                             (place-fret 5 3 1)
                             (place-fret 4 5 2)
                             (place-fret 3 5 3)
                             (place-fret 2 5 4)
                             (place-fret 1 3 1)
                             (barre 5 1 3))
  }
}

%% simple D chord
```

```
      % terse style
      % larger dots, centered dots, fewer frets
      % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                  (finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3))) {
        \fret-diagram-terse #"x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Véase también

Referencia de la notación: Sección A.10.5 [Instrument Specific Markup], página 708.

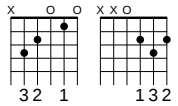
Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*.

Diagramas predefinidos de trastes

Se pueden imprimir los diagramas de trastes usando el contexto `FretBoards`. De forma predeterminada, el contexto `FretBoards` imprime diagramas de trastes que están almacenados en una tabla de búsqueda:

```
\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode {
    c1 d
  }
}
```



Los diagramas de traste predefinidos están en el archivo ‘`predefined-guitar-fretboards.ly`’. Los diagramas de trastes se almacenan tomando como base las notas de un acorde y el valor de `stringTunings` que se está utilizando en cada momento. ‘`predefined-guitar-fretboards.ly`’ contiene diagramas de traste predefinidos sólo para `guitar-tuning`. Los diagramas de traste predefinidos se pueden añadir para otros instrumentos u otras afinaciones siguiendo los ejemplos que aparecen en ‘`predefined-guitar-fretboards.ly`’.

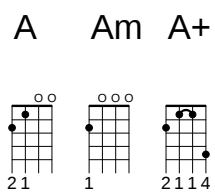
Los diagramas de posiciones para el ukelele están en el archivo ‘`predefined-ukulele-fretboards.ly`’.

```
\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}
```



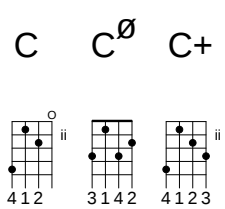
Los diagramas de posiciones para la mandolina están contenidos en el archivo ‘predefined-mandolin-fretboards.ly’.

```
\include "predefined-mandolin-fretboards.ly"

myChords = \chordmode { c1 c:m7.5- c:aug }

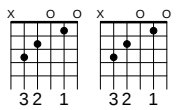
\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}
```



Las notas de los acordes se pueden introducir como música simultánea o bien usando el modo de acordes (véase [Panorámica del modo de acordes], página 399).

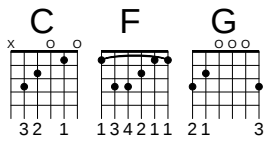
```
\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode { c1 }
  <c' e' g'>1
}
```



Es corriente que se impriman juntos los nombres de acorde y los diagramas de traste. Esto se puede hacer poniendo un contexto de nombres de acorde `ChordNames` en paralelo con un contexto de trastes `FretBoards` y dando a los dos contextos el mismo contenido musical.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

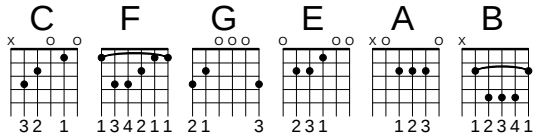
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Los diagramas de traste predefinidos se pueden transportar, en la medida en que esté almacenado un diagrama para el acorde transportado, en la tabla de diagramas de traste.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

mychordlist = {
  \mychords
  \transpose c e { \mychords }
}
<<
  \new ChordNames {
    \mychordlist
  }
  \new FretBoards {
    \mychordlist
  }
>>
```

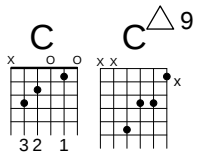


La tabla de diagramas de traste predefinidos para la guitarra contiene ocho acordes (mayor, menor, aumentado, disminuido, séptima dominante, séptima mayor, menor séptima y novena domniante) para 17 tonalidades distintas. La tabla de diagramas predefinidos para el ukelele contiene estos acordes y además otros tres (sexta mayor, segunda suspendida y cuarta suspendida). Se puede ver una lista completa de los diagramas de traste predefinidos en [Diagramas predefinidos de trastes], página 357. Si no hay una entrada en la tabla para un acorde, el grabador

FretBoards calcula un cadena de diagrama fret-diagram usando la funcionalidad de diagramas automáticos que se describe en [Diagramas de traste automáticos], página 367.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 c:maj9
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



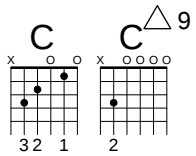
Se pueden añadir diagramas de posiciones a la tabla de diagramas de posiciones. Para añadir un diagrama debemos especificar la tabla de correspondencias para el diagrama, el acorde del diagrama, la afinación utilizada y una definición del diagrama. Normalmente la tabla de correspondencias será *default-fret-table*. La definición del diagrama puede ser una cadena de definición escueta fret-diagram-terse o una lista de marcados prolija fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c:maj9 }
                        #guitar-tuning
                        #"x;3-2;o;o;o;o;"

mychords = \chordmode {
  c1 c:maj9
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



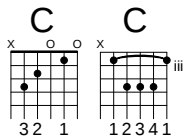
Se pueden almacenar distintos diagramas de traste para el mismo acorde usando distintas octavas para las notas. Las octavas de diferencia deben ser al menos dos por encima o por debajo de la octava predeterminada, porque las octavas inmediatas superior e inferior de la predeterminada se utilizan para los trasteros transpositores.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c'' }
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'bes guitar-tuning))

mychords = \chordmode {
  c1 c''
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Además de los diagramas de traste, LilyPond almacena una lista interna de formas de acorde. Las formas de acorde son diagramas de traste que se pueden desplazar por el mástil para dar acordes distintos. Se pueden añadir formas de acorde a la lista interna y luego usarlas para definir diagramas de posición predefinidos. Dado que se pueden mover a distintas posiciones dentro del mástil, las formas de acorde normalmente no contienen cuerdas al aire. Como los diagramas de posiciones, las formas de acorde se pueden introducir como cadenas escuetas fret-diagram-terse o como listas de marcado prolijas fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"

% Add a new chord shape

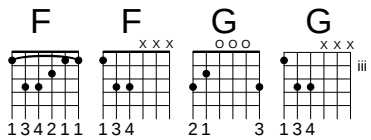
\addChordShape #'powerf #guitar-tuning #"1-1;3-3;3-4;x;x;x;"

% add some new chords based on the power chord shape

\storePredefinedDiagram #default-fret-table
                        \chordmode { f'' }
                        #guitar-tuning
                        #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table
                        \chordmode { g'' }
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'powerf guitar-tuning))
```

```
mychords = \chordmode{
  f1 f'' g g''
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



El aspecto gráfico de un diagrama de traste se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección "fret-diagram-interface" in *Referencia de Funcionamiento Interno*. Para un diagrama de trastes predefinido, las propiedades de interface pertenecen a `FretBoards.FretBoard`.

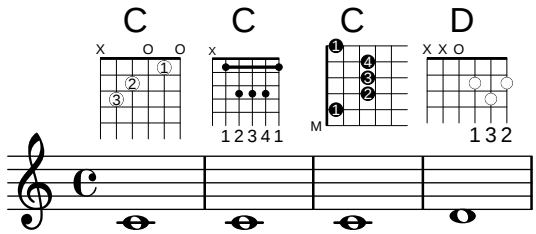
Fragmentos de código seleccionados

Personalizar los diagramas de posiciones

Se pueden establecer las propiedades de los diagramas de posiciones de acordes por medio de `'fret-diagram-details`. Para los diagramas de posiciones de FretBoard, se aplican los overrides (sobreescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobreescritura de propiedades.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        #"x;1-1-(;3-2;3-3;3-4;1-1-);"
<<
  \new ChordNames {
    \chordmode { c1 | c | c | d }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = #'1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \once \override FretBoard.size = #'1.0
      \once \override FretBoard.fret-diagram-details.barre-type = #'straight
      \once \override FretBoard.fret-diagram-details.dot-color = #'black
      \once \override FretBoard.fret-diagram-details.finger-code = #'below-string
      c'
      \once \override FretBoard.fret-diagram-details.barre-type = #'none
      \once \override FretBoard.fret-diagram-details.number-type = #'arabic
```

```
      \once \override FretBoard.fret-diagram-details.orientation = #'landscape
      \once \override FretBoard.fret-diagram-details.mute-string = #"M"
      \once \override FretBoard.fret-diagram-details.label-dir = #LEFT
      \once \override FretBoard.fret-diagram-details.dot-color = #'black
      c'
      \once \override FretBoard.fret-diagram-details.finger-code = #'below-string
      \once \override FretBoard.fret-diagram-details.dot-radius = #0.35
      \once \override FretBoard.fret-diagram-details.dot-position = #0.5
      \once \override FretBoard.fret-diagram-details.fret-count = #3
      d
    }
  }
  \new Voice {
    c'1 | c' | c' | d'
  }
>>
```



Definición de posiciones predefinidas para otros instrumentos

Se pueden añadir diagramas de posiciones predefinidas para instrumentos nuevos además de los estándar que se usan para la guitarra. Este archivo muestra cómo se hace, definiendo una afinación nueva y unas cuantas posiciones para el cuatro venezolano.

Este archivo también muestra cómo se pueden incluir las digitaciones en los acordes que se usan como puntos de referencia para la búsqueda de acordes en la tabla, y mostrarse en el diagrama de posiciones y la tablatura `TabStaff`, pero no en la música.

Estas posiciones no se pueden transportar porque contienen información de las cuerdas. Hay planes para corregir esto en un futuro.

```
% add FretBoards for the Cuatro
%   Note: This section could be put into a separate file
%      predefined-cuatro-fretboards.ly
%      and \included into each of your compositions

cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
```

```
                        #"o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        #"o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        #"o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
                        #cuatroTuning
                        #"o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
                        #cuatroTuning
                        #"2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly


#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

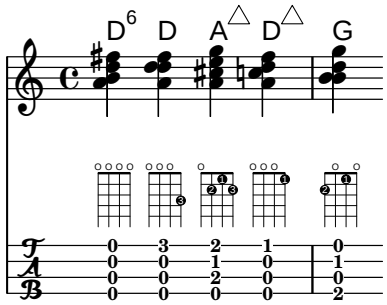
    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%        #'(fret-diagram-details string-count) = #'4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }
```

```
    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/16)
    }
  }
  \midi { }
}
```



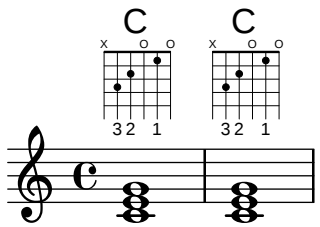
Cambios de acorde de posiciones de trastes

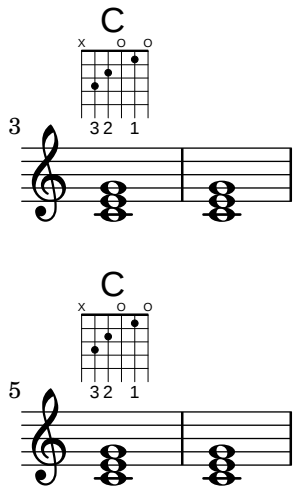
Se puede hacer que los diagramas de posiciones se muestren sólo cuando el acorde cambia o al comienzo de una nueva línea.

```
\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
  \new ChordNames { \myChords }
  \new FretBoards { \myChords }
  \new Staff { \myChords }
>>
```





Tablas alternativas de diagramas de posiciones

Se pueden crear tablas alternativas de diagramas de posiciones. Se utilizarían para tener diagramas altenativos para un acorde dado.

Para usar una tabla alternativa de diagramas de posiciones, se debe crear la tabla primero. Después se añaden los diagramas a la tabla.

La tabla de diagramas de posiciones que se crea puede estar vacía o se puede copiar a partir de una tabla existente.

La tabla a usar en la impresión de los diagramas predefinidos se selecciona por medio de la propiedad `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table
#(define custom-fretboard-table-one (make-fretboard-table))

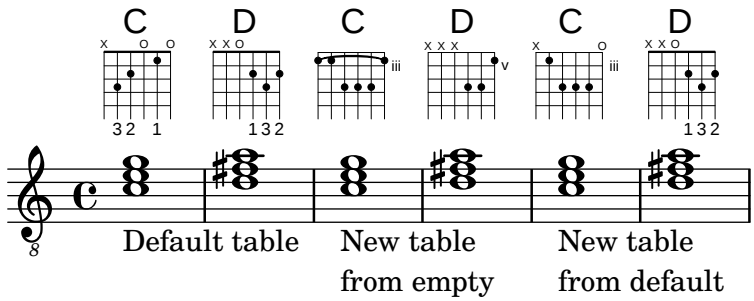
% Make a new fretboard table as a copy of default-fret-table
#(define custom-fretboard-table-two (make-fretboard-table default-fret-table))

% Add a chord to custom-fretboard-table-one
\storePredefinedDiagram #custom-fretboard-table-one
                        \chordmode{c}
                        #guitar-tuning
                        "3-(;3;5;5;5;3-);"

% Add a chord to custom-fretboard-table-two
\storePredefinedDiagram #custom-fretboard-table-two
                        \chordmode{c}
                        #guitar-tuning
                        "x;3;5;5;5;o;"

<<
  \chords {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  \new FretBoards {
    \chordmode {
      \set predefinedDiagramTable = #default-fret-table
      c1 | d1 |
```

```
      \set predefinedDiagramTable = #custom-fretboard-table-one
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-two
      c1 | d1 |
    }
  }
  \new Staff {
    \clef "treble_8"
    <<
      \chordmode {
        c1 | d1 |
        c1 | d1 |
        c1 | d1 |
      }
      {
        s1_\markup "Default table" | s1 |
        s1_\markup \column {"New table" "from empty"} | s1 |
        s1_\markup \column {"New table" "from default"} | s1 |
      }
    >>
  }
>>
```



Véase también

Referencia de la notación: [Tablaturas personalizadas], página 345, [Diagramas de traste automáticos], página 367, [Panorámica del modo de acordes], página 399, [Diagramas predefinidos de trastes], página 357.

Archivos de inicio: ‘`ly/predefined-guitar-fretboards.ly`’, ‘`ly/predefined-guitar-ninth-fretboards.ly`’, ‘`ly/predefined-ukulele-fretboards.ly`’, ‘`ly/predefined-mandolin-fretboards.ly`’.

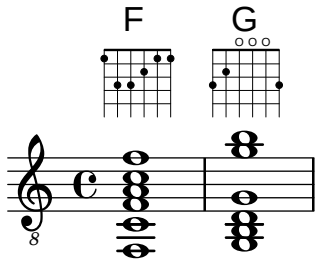
Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*.

Diagramas de traste automáticos

Se pueden crear diagramas de traste automáticamente a partir de las notas introducidas usando el contexto `FretBoards`. Si no está disponible ningún diagrama predefinido para las notas introducidas en la afinación `stringTunings` activa, este contexto calcula las cuerdas y los trastes que se pueden usar para tocar las notas.

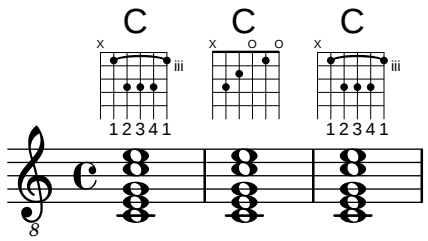
```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new FretBoards {
    <f, c f a c' f'>1
    <g,\6 b, d g b g'>1
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1
    <g, b, d g b' g'>1
  }
>>
```



Puesto que de forma predeterminada no se carga ningún diagrama predefinido, el comportamiento predeterminado es el cálculo automático de los diagramas de traste. Una vez que los diagramas predeterminados se han cargado, se puede habilitar e inhabilitar el cálculo automático con instrucciones predefinidas:

```
\storePredefinedDiagram #default-fret-table
                        <c e g c' e'>
                        #guitar-tuning
                        #"x;3-1-(;5-2;5-3;5-4;3-1-1-);"
<<
  \new ChordNames {
    \chordmode {
      c1 c c
    }
  }
  \new FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>1
    \predefinedFretboardsOn
    <c e g c' e'>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <c e g c' e'>1
    <c e g c' e'>1
```

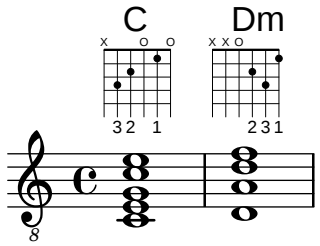
```
  }
>>
```



A veces el calculador de trastes es incapaz de encontrar un diagrama aceptable. Esto se puede remediar generalmente mediante la asignación manual de una nota a una cuerda. En muchos casos sólo hay que colocar manualmente una nota sobre una cuerda; el resto de las notas se situará en el lugar adecuado por parte del contexto `FretBoards`.

Se pueden añadir digitaciones a los diagramas de traste del contexto FretBoard.

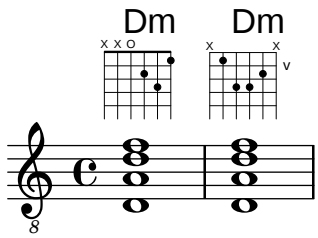
```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new FretBoards {
    <c-3 e-2 g c'-1 e'>1
    <d a-2 d'-3 f'-1>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <d a d' f'>1
  }
>>
```



El traste mínimo que usar en el cálculo de cuerdas y trastes para el contexto FretBoard se puede fijar con la propiedad `minimumFret`.

```
<<
  \new ChordNames {
    \chordmode {
      d1:m d:m
    }
  }
  \new FretBoards {
    <d a d' f'>1
    \set FretBoards.minimumFret = #5
```

```
    <d a d' f'>1
  }
  \new Staff {
    \clef "treble_8"
    <d a d' f'>1
    <d a d' f'>1
  }
>>
```



Las cuerdas y los trastes para el contexto `FretBoards` dependen de la propiedad `stringTunings`, que tiene el mismo significado que en el contexto de tablatura TabStaff. Consulte [Tablaturas personalizadas], página 345 para ver más información sobre la propiedad `stringTunings`.

El aspecto gráfico de un diagrama de trastes se puede personalizar de acuerdo a las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*. Para un diagrama de `FretBoards`, las propiedades del interface pertenecen a `FretBoards.FretBoard`.

Instrucciones predefinidas

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Véase también

Referencia de la notación: [Tablaturas personalizadas], página 345.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los cálculos de tablatura automáticos no funcionan bien para instrumentos con afinaciones no monótonas.
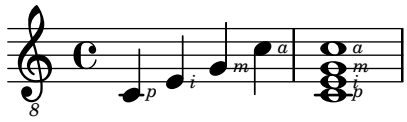
Digitaciones de la mano derecha

Las digitaciones de la mano derecha *p-i-m-a* se deben introducir usando `\rightHandFinger` seguido de un número.

> **Nota:** Si el número se introduce en notación de Scheme, recuerde escribir al final un espacio seguido de un ángulo de cierre `>` o similar.

```
\clef "treble_8"
c4\rightHandFinger #1
e\rightHandFinger #2
g\rightHandFinger #3
c\rightHandFinger #4
```

```
<c,\rightHandFinger #1 e\rightHandFinger #2
 g\rightHandFinger #3 c\rightHandFinger #4 >1
```



Por brevedad, se puede escribir `\rightHandFinger` como algo más corto, como por ejemplo `RH`:

```
RH=#rightHandFinger
```

Fragmentos de código seleccionados

Posicionamiento de digitaciones de mano derecha

Es posible ejercer un mayor control sobre la colocación de las digitaciones de la mano derecha estableciendo el valor de una propiedad específica, como se muestra en el ejemplo siguiente. Nota: se debe usar una construcción de acorde.

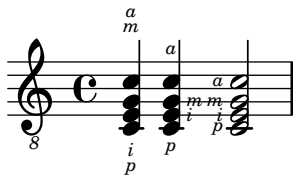
```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(left)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >2
}
```

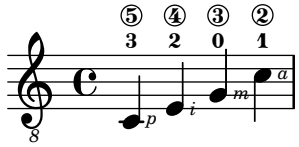


Digitaciones - indicación del número de cuerda y digitaciones de mano derecha

En este ejemplo se combinan las digitaciones de la mano izquierda, indicaciones del número de cuerda y digitaciones de la mano derecha.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5-\RH #1 >4
  <e-2\4-\RH #2 >4
  <g-0\3-\RH #3 >4
  <c-1\2-\RH #4 >4
}
```



Véase también

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StrokeFinger” in *Referencia de Funcionamiento Interno*.
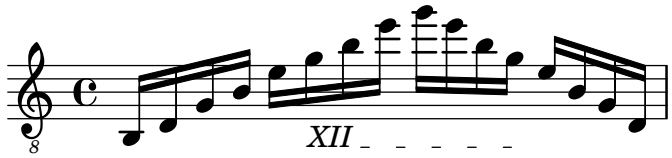
2.4.2 Guitarra

Casi todos los asuntos de notación relacionados con la música para guitarra están suficientemente cubiertos en la sección general sobre ajustes de instrumentos de trastes, pero hay algunos más de los que merece la pena hablar aquí. De forma ocasional, los usuarios quieren crear documentos del tipo de cancioneros que tengan sólo la letra de las canciones con indicaciones de acordes sobre ella. Dado que LilyPond es un tipografiador de música, no se recomienda para documentos que no tienen notación musical. Una alternativa mejor es un procesador de texto, editor de texto o, para usuarios con experiencia, un tipografiador como GuitarTeX.

Indicar la posición y la cejilla

Este ejemplo muestra cómo incluir indicaciones de posición y cejilla de guitarra:

```
\clef "treble_8"
b16 d g b e
\textSpannerDown
\override TextSpanner.bound-details.left.text = #"XII "
g16\startTextSpan
b16 e g e b g\stopTextSpan
e16 b g d
```



Véase también
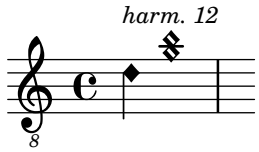
Referencia de la notación: [Extensiones de texto], página 227.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*, Sección “Expressive marks” in *Fragmentos de código*.

Indicar armónicos y notas tapadas

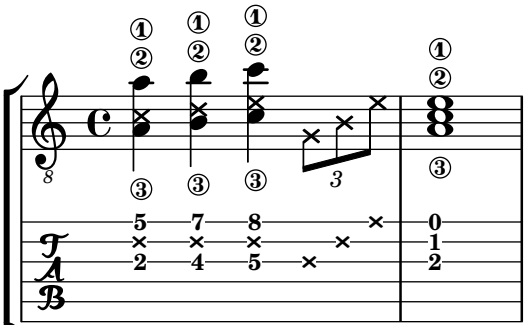
Se pueden usar cabezas de nota especiales para indicar notas apagadas o armónicos. Los armónicos se suelen explicar de forma más completa con un marcado de texto.

```
\relative c' {
  \clef "treble_8"
  \override Staff.NoteHead.style = #'harmonic-mixed
  d^\markup { \italic { \fontsize #-2 { "harm. 12" }}} <g b>1
}
```



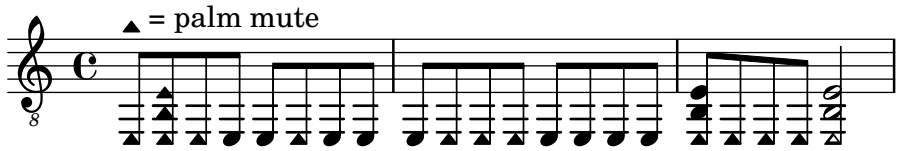
Las notas tapadas (llamadas también *notas muertas*) están contempladas dentro de los pentagramas normales y de tablatura:

```
music = \relative c' {
  < a\3 \deadNote c\2 a'\1 >4
  < b\3 \deadNote d\2 b'\1 >
  < c\3 \deadNote e\2 c'\1 >
  \deadNotesOn
  \tuplet 3/2 { g8 b e }
  \deadNotesOff
  < a,\3 c\2 e\1 >1
}
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \music
  }
  \new TabStaff {
    \music
  }
>>
```



Otra téctnica de ejecución (utilizada especialmente en la guitarra eléctrica) se llama *tapado de palma*. La cuerda es parcialmente tapada por la palma de la mano que pulsa (de ahí el nombre). Lilypond contempla la notación de las notas de estilo de tapado de palma mediante el cambio de la forma de la cabeza de nota a un triángulo.

```
\new Voice { % Warning: explicit Voice instantiation is
             %    required to have palmMuteOff work properly
             %    when palmMuteOn comes at the beginning of
             %    the piece.
  \relative c, {
    \clef "G_8"
    \palmMuteOn
    e8^\markup { \musicglyph #"noteheads.u2do"  = palm mute }
    < e b' e > e
    \palmMuteOff
    e e  \palmMute e e e |
    e8 \palmMute { e e e } e e e e |
    < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
  }
}
```



Véase también

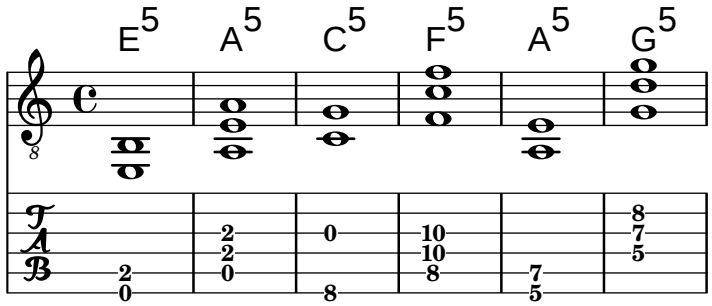
Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de la notación: [Cabezas de nota especiales], página 35, Sección A.9 [Estilos de cabezas de nota], página 671.

Indicación de acordes de potencia o de quinta vacía

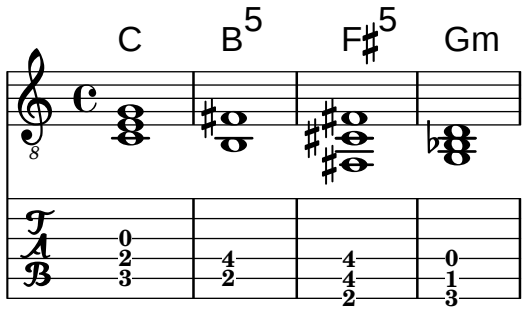
Los acordes de quinta vacía o acordes de potencia (*power chords*) y sus símbolos se pueden tipografiar en el modo de acordes o como construcciones de acorde normales:

```
ChordsAndSymbols = {
  \chordmode {
    \powerChords
    e,,1:1.5
    a,,1:1.5.8
    \set minimumFret = #8
    c,1:1.5
    f,1:1.5.8
  }
  \set minimumFret = #5
  <a, e>1
  <g d' g'>1
}
\score {
  <<
    \new ChordNames {
    \ChordsAndSymbols
    }
    \new Staff {
      \clef "treble_8"
      \ChordsAndSymbols
    }
    \new TabStaff {
      \ChordsAndSymbols
    }
  >>
}
```



Los símbolos de acordes de quinta vacía se desactivan automáticamente tan pronto como se utiliza cualquiera de los otros modificadores de acorde:

```
mixedChords = \chordmode {
  c,1
  \powerChords
  b,,1:1.5
  fis,,1:1.5.8
  g,,1:m
}
\score {
  <<
    \new ChordNames {
      \mixedChords
    }
    \new Staff {
      \clef "treble_8"
      \mixedChords
    }
    \new TabStaff {
      \mixedChords
    }
  >>
}
```



Véase también

Glosario musical: Sección “power chord” in *Glosario Musical*.

Referencia de la notación: [Acordes extendidos y alterados], página 402, [Impresión de los nombres de acorde], página 405.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

2.4.3 Banjo

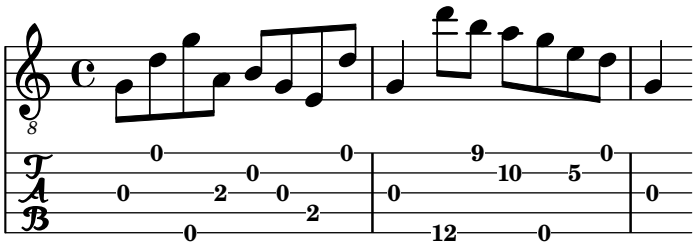
Tablaturas de banjo

LilyPond contempla el banjo de cinco cuerdas de una manera básica. Cuando haga tablaturas para banjo de cinco cuerdas, utilice la función de formato de tablatura de banjo para obtener los números correctos de los trastes para la quinta cuerda:

```
music = {
  g8 d' g'\5 a b g e d' |
  g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
  g4
}

<<
```

```
  \new Staff \with { \omit StringNumber }
  { \clef "treble_8"  \music }
  \new TabStaff \with {
    tablatureFormat = #fret-number-tablature-format-banjo
    stringTunings = #banjo-open-g-tuning
  }
  { \music }
>>
```



Se encuentran predefinidas varias afinaciones comunes para el banjo de cinco cuerdas: `banjo-c-tuning` (sol-Do-Sol-Si-Re), `banjo-modal-tuning` (sol-Re-Sol-Do-Re), `banjo-open-d-tuning` (Re Mayor al aire, la-Re-Fa#-La-Re) y `banjo-open-dm-tuning` (Re menor al aire, la-Re-Fa-La-Re).

Estas afinaciones se pueden convertir a afinaciones para banjo de cuatro cuerdas utilizando la función `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Véase también

Archivos de inicio: '`ly/string-tunings-init.ly`'

Fragmentos de código: Sección "Fretted strings" in *Fragmentos de código*.

2.5 Percusión

2.5.1 Notación común para percusión

La notación rítmica se emplea primordialmente para la percusión y la batería, pero también se puede utilizar para mostrar los valores rítmicos una melodía.

Referencias para percusión

- La notación de algunas percusiones se puede hacer sobre una pauta de ritmo; esto se estudia en [Mostrar los ritmos de la melodía], página 76 y [Crear instancias de pentagramas nuevos], página 183.
- La salida MIDI se trata en una sección aparte; véase Sección 3.5.6 [Percusión en MIDI], página 510.

Véase también

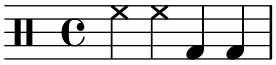
Referencia de la notación: [Mostrar los ritmos de la melodía], página 76, [Crear instancias de pentagramas nuevos], página 183. Sección 3.5.6 [Percusión en MIDI], página 510.

Fragmentos de código: Sección "Percussion" in *Fragmentos de código*.

Notación básica de percusión

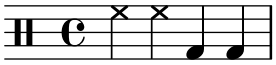
Las notas de percusión se pueden escribir en el modo `\drummode`, que es similar al modo estándar para introducir notas. La manera más fácil de escribir notas de percusión es utilizar la instrucción `\drums`, que crea el contexto y el modo de entrada apropiados para percusión:

```
\drums {
  hihat4 hh bassdrum bd
}
```



Esto es una abreviatura de
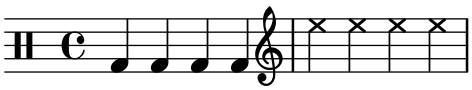
```
\new DrumStaff {
  \drummode {
    hihat4 hh bassdrum bd
  }
}
```



Cada elemento de un set de percusión lleva un nombre completo y un nombre abreviado, y los dos se pueden usar en la entrada. La lista completa de nombre de instrumentos de percusión se encuentra en Sección A.14 [Notas de percusión], página 726.

Observe que la notación normal de las notas con altura determinada (como `cis4`) en un contexto `DrumStaff` producen un mensaje de error. Las claves de percusión se añaden automáticamente al contexto `DrumStaff` pero también se pueden establecer explícitamente o utilizar otras claves.

```
\drums {
  \clef percussion
  bd4 bd bd bd
  \clef treble
  hh4 hh hh hh
}
```



Hay ciertos detalles respecto a cómo está contemplado el MIDI para los instrumentos de percusión; para ver los detalles consulte Sección 3.5.6 [Percusión en MIDI], página 510.

Véase también

Referencia de la notación: Sección 3.5.6 [Percusión en MIDI], página 510, Sección A.14 [Notas de percusión], página 726.
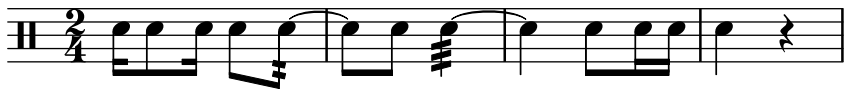
Archivos de inicio: ‘`ly/drumpitch-init.ly`’.

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Redobles

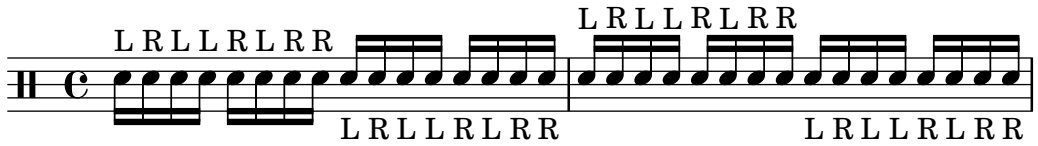
Los redobles se indican mediante tres barras cruzadas en la plica. Para las negras o notas más largas, las tres barras se muestran explícitamente, las corcheas se presentan con dos barras cruzadas (siendo la barra de corchea la tercera), y los redobles más breves que las corcheas tienen una barra cruzada para complementar las barras normales. Esto se consigue mediante la notación de trémolo, véase [Repeticiones de trémolo], página 159.

```
\drums {
  \time 2/4
  sn16 sn8 sn16 sn8 sn8:32 ~
  sn8 sn8 sn4:32 ~
  sn4 sn8 sn16 sn16
  sn4 r4
}
```



Los golpes de baqueta se pueden indicar mediante la colocación de los elementos de marcado `"R"` o `"L"` encima o debajo de las notas, como se estudia en Sección 5.4.2 [Dirección y posición], página 601. La propiedad `staff-padding` se puede sobreescribir para conseguir una línea de base satisfactoria.

```
\drums {
  \repeat unfold 2 {
    sn16^"L" sn^"R" sn^"L" sn^"L" sn^"R" sn^"L" sn^"R" sn^"R"
    \stemUp
    sn16_"L" sn_"R" sn_"L" sn_"L" sn_"R" sn_"L" sn_"R" sn_"R"
  }
}
```



Véase también

Referencia de la notación: [Repeticiones de trémolo], página 159.

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Percusión afinada

Ciertos instrumentos de percusión de altura determinada (p.ej.: el xilófono, el vibráfono y los timbales) se escriben usando pentagramas normales. Esto se estudia en otras secciones del manual.

Véase también

Referencia de la notación: Sección 3.5.6 [Percusión en MIDI], página 510.

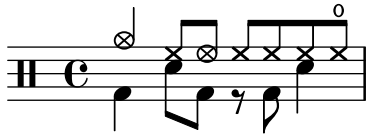
Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Pautas de percusión

Una parte de percusión para más de un instrumento, normalmente utiliza una pauta de varias líneas donde cada posición dentro de la pauta se refiere a un elemento de percusión. Para tipografiar la música, se deben interpretar las notas dentro de los contextos Sección “DrumStaff” in *Referencia de Funcionamiento Interno* y Sección “DrumVoice” in *Referencia de Funcionamiento Interno*.

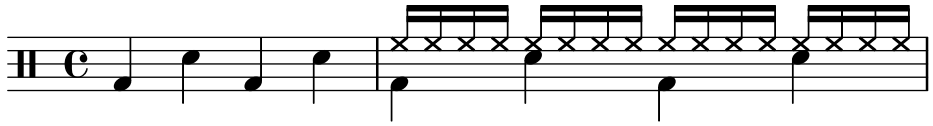
```
up = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
```

```
down = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



El ejemplo anterior muestra una notación polifónica prolija. La notación polifónica abreviada, descrita en Sección “Oigo voces” in *Manual de Aprendizaje*, también se puede usar. Por ejemplo,

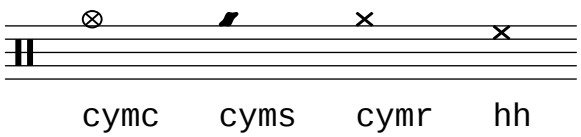
```
\new DrumStaff <<
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \\ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```

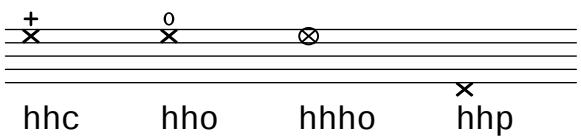


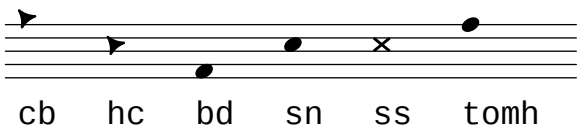
También existen otras posibilidades en lo referente a la disposición. Para usarlas, establezca la propiedad `drumStyleTable` en el contexto Sección “DrumVoice” in *Referencia de Funcionamiento Interno*. Se han predefinido las siguientes variables:
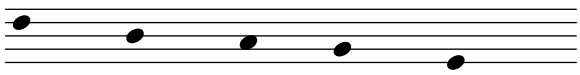
`drums-style`

Es la opción por defecto. Tipografía un típico set de percusión sobre pentagrama:



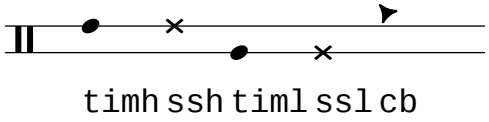




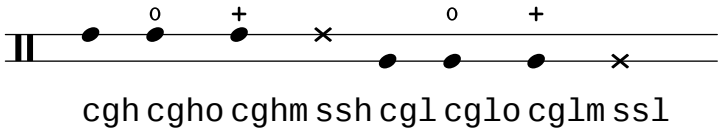


El esquema de percusión contempla seis tambores graves (toms) distintos. Cuando haya menos toms, sencillamente seleccione aquellos que producen el resultado deseado, es decir, para tener toms en las tres líneas centrales utilizará `tommh`, `tomml` y `tomfh`.

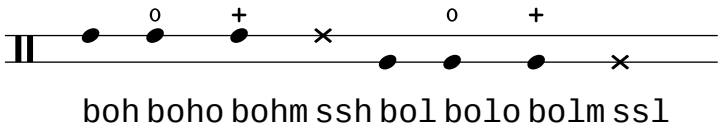
`timbales-style`
Esto tipografía timbales en una pauta de dos líneas:



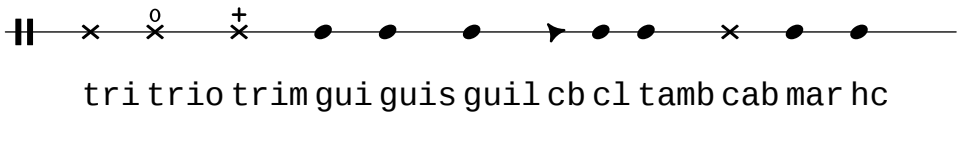
`congas-style`
Esto tipografía congas en una pauta de dos líneas:



`bongos-style`
Esto tipografía bongos sobre una pauta de dos líneas:



`percussion-style`
Para tipografiar toda clase de percusiones simples sobre pautas de una línea:



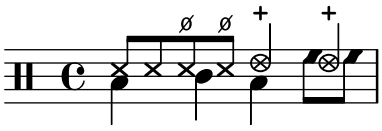
Pautas de percusión personalizadas

Si no le gusta ninguna de las listas predefinidas, puede definir su propia lista al principio de su archivo

```
#(define mydrums '(
         (bassdrum        default   #f           -1)
         (snare           default   #f           0)
         (hihat           cross     #f           1)
         (halfopenhihat   cross     "halfopen"   1)
         (pedalhihat      xcircle   "stopped"    2)
         (lowtom          diamond   #f           3)))
up = \drummode { hh8 hh hhho hhho hhp4 hhp }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
```

```
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



Fragmentos de código seleccionados

He aquí algunos ejemplos adicionales:

Dos bloques de madera, escritos con wbh (*woodblock-high*, bloque alto) y wbl (*woodblock-low* bloque bajo)

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
#(define mydrums '((hiwoodblock default #t  3)
                   (lowoodblock default #t -2)))

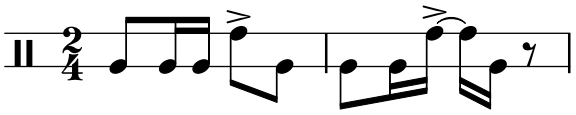
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % This is necessary; if not entered, the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
}

\new DrumStaff {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  % with this you load your new drum style table
  \woodstaff

  \drummode {
    \time 2/4
    wbl8 wbl16 wbl wbh8-> wbl |
    wbl8 wbl16 wbh-> ~ wbh wbl16 r8 |
  }
}
```



Observe que en este caso especial se debe alterar la longitud de la línea divisoria con `\override Staff.BarLine.bar-extent #'(de . a)`. En caso contrario resulta demasiado corta. También debe definir las posiciones de las dos líneas del pentagrama. Para más información sobre estos delicados asuntos, consulte [El símbolo del pentagrama], página 191.

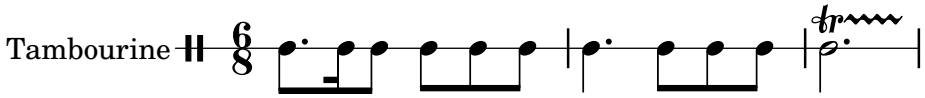
Una pandereta, escrita mediante 'tamb' (*tambourine*):

```
#(define mydrums '((tambourine default #t 0)))

tambustaff = {
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Tambourine"
}

\new DrumStaff {
  \tambustaff
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    \time 6/8
    tamb8. tamb16 tamb8 tamb tamb tamb |
    tamb4. tamb8 tamb tamb |
    % the trick with the scaled duration and the shorter rest
    % is neccessary for the correct ending of the trill-span!
    tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
  }
}
```



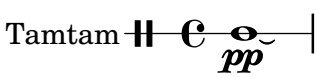
Música para gong, introducida con 'tt' (tam-tam):

```
#(define mydrums '((tamtam default #t 0)))

tamtamstaff = {
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Tamtam"
}

\new DrumStaff {
  \tamtamstaff
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    tt 1 \pp \laissezVibrer
  }
}
```
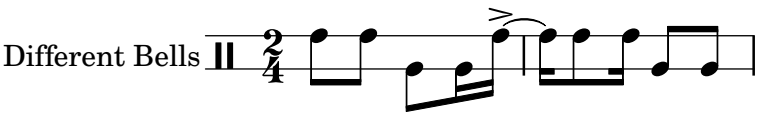


Dos campanas, introducidas con 'cb' (*cowbell*, cencerro) y 'rb' (*ridebell*, campana normal)

```
#(define mydrums '((ridebell default #t  3)
                   (cowbell  default #t -2)))

bellstaff = {
```

```
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Different Bells"
}

\new DrumStaff {
  \bellstaff
  \drummode {
    \time 2/4
    rb8 rb cb cb16 rb-> ~ |
    rb16 rb8 rb16 cb8 cb |
  }
}
```



Aquí un breve ejemplo procedente de la 'Historia del soldado' de Stravinsky:

```
#(define mydrums '((bassdrum   default #t  4)
                   (snare      default #t -4)
                   (tambourine default #t  0)))

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}
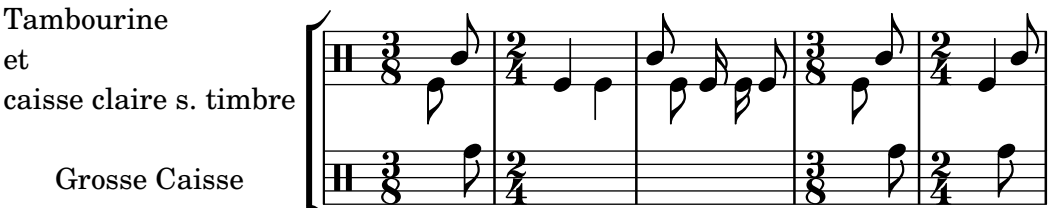
drumsA = {
  \context DrumVoice <<
    { \global }
    { \drummode {
        \autoBeamOff
        \stemDown sn8 \stemUp tamb s8 |
        sn4 \stemDown sn4 |
        \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
        \stemDown sn8 \stemUp tamb s8 |
        \stemUp sn4 s8 \stemUp tamb
      }
    }
  >>
}

drumsB = {
  \drummode {
    s4 bd8 s2*2 s4 bd8 s4 bd8 s8
  }
}
```

```
\layout {
  indent = #40
}

\score {
  \new StaffGroup <<
    \new DrumStaff {
      \set DrumStaff.instrumentName = \markup {
        \column {
          "Tambourine"
          "et"
          "caisse claire s. timbre"
        }
      }
      \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
      \drumsA
    }

   \new DrumStaff {
     \set DrumStaff.instrumentName = #"Grosse Caisse"
     \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
     \drumsB }
  >>
}
```



Véase también

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

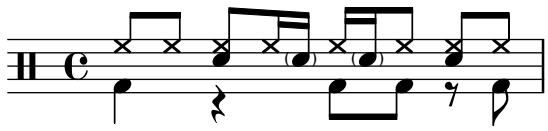
Referencia de funcionamiento interno: Sección “DrumStaff” in *Referencia de Funcionamiento Interno*, Sección “DrumVoice” in *Referencia de Funcionamiento Interno*.

Notas fantasma

Las notas fantasma para la batería e instrumentos de percusión se pueden crear utilizando la instrucción `\parenthesize` detallada en [Paréntesis], página 219.

```
\new DrumStaff
<<
  \context DrumVoice  = "1" { s1 }
  \context DrumVoice  = "2" { s1 }
  \drummode {
    <<
      {
        hh8[ hh] <hh sn> hh16
        \parenthesize sn  hh
        \parenthesize sn  hh8 <hh sn> hh
      } \\
```
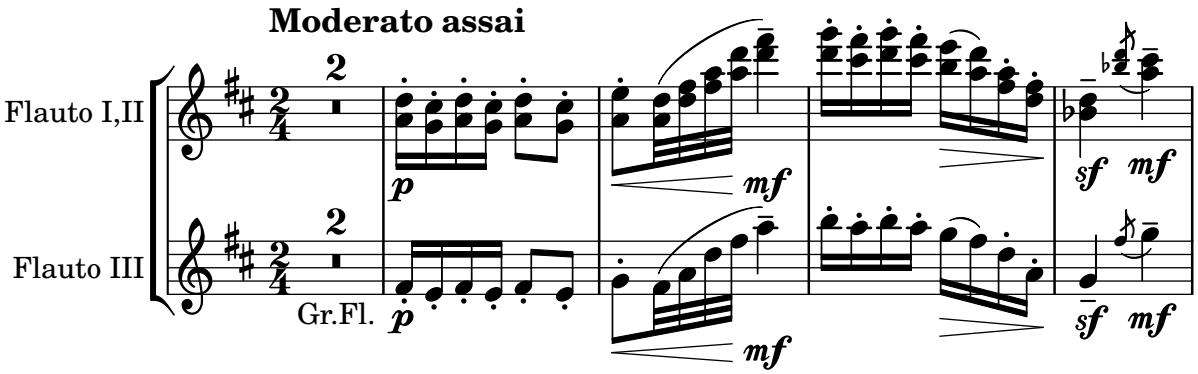
```
      {
        bd4 r4 bd8 bd r8 bd
      }
    >>
  }
>>
```



Véase también

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

2.6 Instrumentos de viento



Esta sección incluye elementos de notación musical que afloran al escribir para instrumentos de viento.

2.6.1 Notación común para instrumentos de viento

Esta sección trata la notación común para casi todos los instrumentos de viento.

Referencias para instrumentos de viento

Muchas cuestiones de la notación para instrumentos de viento tienen que ver con las respiraciones y los golpes de lengua:

- Las respiraciones se pueden especificar mediante silencios o marcas de respiración, véase [Marcas de respiración], página 131.
- La ejecución ligada se indica mediante ligaduras de expresión, véase [Ligaduras de expresión], página 127.
- Los distintos golpes de lengua, desde legato hasta staccato pasando por non legato, se presentan por lo general mediante articulaciones, en ocasiones combinadas con ligaduras de expresión, véase [Articulaciones y ornamentos], página 116 y Sección A.13 [Lista de articulaciones], página 725.
- El frullato se indica generalmente mediante una indicación de trémolo y una marca textual sobre la nota. Véase [Repeticiones de trémolo], página 159.

También existen otros aspectos de la notación musical que son de aplicación a los instrumentos de viento:

- Muchos instrumentos de viento son transpositores, véase [Transposición de los instrumentos], página 24.
- El portamento es característico del trombón, pero otros instrumentos de viento pueden realizar glissandos con llaves o válvulas. Véase [Glissando], página 134.
- Los glissandos de serie armónica, que son posibles en todos los metales pero bastante específicos de las trompas, se escriben en general como notas de adorno, véase [Notas de adorno], página 108.
- Las inflexiones de tono al final de una nota se tratan en [Caídas y elevaciones], página 133.
- Los golpes de llave o de válvula se suelen indicar con el estilo `cross` (aspas) de cabezas de nota, véase [Cabezas de nota especiales], página 35.
- Los instrumentos de viento-madera pueden sobreinflar las notas graves para producir armónicos. Éstos se muestran por medio de la articulación `flageolet`. Véase Sección A.13 [Lista de articulaciones], página 725.
- El uso de sordinas para los metales se suele indicar mediante marcas de texto, pero en los lugares donde se suceden muchos cambios rápidos es mejor utilizar las articulaciones `stopped` (tapado) y `open` (abierto). Véase [Articulaciones y ornamentos], página 116 y Sección A.13 [Lista de articulaciones], página 725.
- Las trompas tapadas se indican por medio de la articulación `stopped`. Véase [Articulaciones y ornamentos], página 116.

Fragmentos de código seleccionados

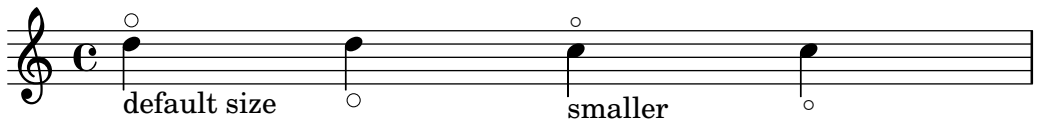
Cambiar el tamaño de la marca de \flageolet

Para hacer más pequeño el círculo de `\flageolet` (armónico) utilice la siguiente función de Scheme.

```
smallFlageolet =
#(let ((m (make-articulation "flageolet")))
   (set! (ly:music-property m 'tweaks)
         (acons 'font-size -3
                (ly:music-property m 'tweaks)))
   m)

\layout { ragged-right = ##f }

\relative c'' {
  d4^\flageolet_\markup { default size } d_\flageolet
  c4^\smallFlageolet_\markup { smaller } c_\smallFlageolet
}
```



Véase también

Referencia de la notación: [Marcas de respiración], página 131, [Ligaduras de expresión], página 127, [Articulaciones y ornamentos], página 116, Sección A.13 [Lista de articulaciones], página 725, [Repeticiones de trémolo], página 159, [Transposición de los instrumentos], página 24, [Glissando], página 134, [Notas de adorno], página 108, [Caídas y elevaciones], página 133, [Cabezas de nota especiales], página 35.

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

Digitaciones

Todos los instrumentos de viento aparte del trombón requieren el uso de los distintos dedos para producir cada nota. En los fragmentos de código que aparecen a continuación pueden verse algunos ejemplos de digitaciones.

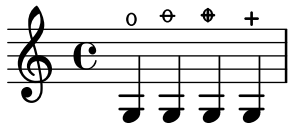
Se pueden producir diagramas de posiciones para instrumentos de viento madera,como se describe en Sección 2.6.3.1 [Diagramas de posiciones para viento madera], página 391.

Fragmentos de código seleccionados

Símbolos de digitación para instrumentos de viento

Se pueden conseguir símbolos especiales combinando glifos existentes, lo que es de utilidad para la notación de instrumentos de viento.

```
centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset =#(ly:make-simple-closure
    `(,+
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::centered-on-x-parent))
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::x-aligned-on-self))))
}
\score
{\relative c'
  {
    g\open
    \once \override TextScript.staff-padding = #-1.0 \centermarkup
    g^\markup{\combine \musicglyph #"scripts.open" \musicglyph
    #"scripts.tenuto"}
    \centermarkup g^\markup{\combine \musicglyph #"scripts.open"
    \musicglyph #"scripts.stopped"}
    g\stopped
  }
}
```



Diagramas de digitación para la flauta dulce

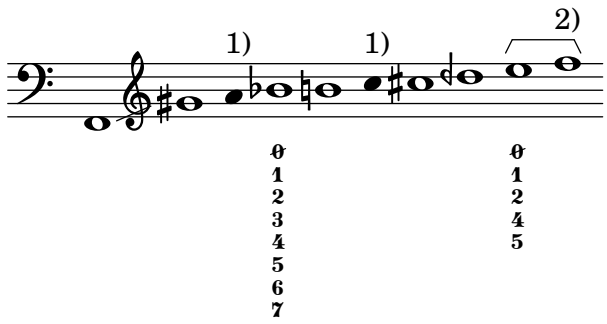
El ejemplo siguiente muestra cómo se pueden realizar diagramas de digitación para instrumentos de viento.

```
% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset =#(ly:make-simple-closure
  `(,+
  ,(ly:make-simple-closure (list
  ly:self-alignment-interface::centered-on-x-parent))
  ,(ly:make-simple-closure (list
  ly:self-alignment-interface::x-aligned-on-self))))
```

```
}

\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \omit Stem
    \omit Flag
    \consists "Horizontal_bracket_engraver"
  }
  {
    \clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4^\markup{1)}
    \centermarkup
    \once \override TextScript.padding = #2
    bes'1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 3 \finger 4
    \finger 5 \finger 6 \finger 7} }
    b'1*1/4
    c''4^\markup{1)}
    \centermarkup
    \once \override TextScript.padding = #2
    cis''1*1/4
    deh''1*1/4
    \centermarkup
    \once \override TextScript.padding = #2
    \once \override Staff.HorizontalBracket.direction = #UP
    e''1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 4
    \finger 5} }\startGroup
    f''1*1/4^\markup{2)}\stopGroup
  }
}
```



Véase también

Referencia de la notación: Sección 2.6.3.1 [Diagramas de posiciones para viento madera], página 391.

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

2.6.2 Gaita

Esta sección trata de la notación común para gaita.

Definiciones para la gaita

LilyPond contiene definiciones especiales para la música de gaita escocesa de las tierras altas; para usarlas, escriba
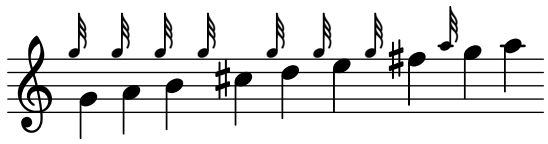
```
\include "bagpipe.ly"
```

al principio del archivo de entrada. De esta forma podrá escribir las notas de adorno especiales que son usuales en la música de gaita, mediante instrucciones breves. Por ejemplo, puede escribir `\taor` en lugar de

```
\grace { \small G32[ d G e] }
```

'`bagpipe.ly`' también contiene definiciones de alturas para las notas de la gaita en las octavas adecuadas, de forma que no se tenga que preocupar por `\relative` o por `\transpose`.

```
\include "bagpipe.ly"
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La música de gaita utiliza siempre la tonalidad de Re mayor (aunque esto no es completamente cierto). Sin embargo, puesto que es la única tonalidad que se puede utilizar, normalmente la armadura no se escribe. Para configurar esto de forma correcta, comience la música siempre con `\hideKeySignature` (ocultar armadura). Si, por algún motivo, quiere mostrar la armadura, puede usar `\showKeySignature` en su lugar.

Alguna música moderna utiliza digitaciones cruzadas sobre el Do y el Fa para bemolizar estas notas. Se puede indicar esto mediante `cflat` o `fflat`. De forma parecida, el sol agudo piobaireachd se puede escribir `gflat` cuando aparece en música ligera.

Véase también

Fragmentos de código: Sección "Winds" in *Fragmentos de código*.

Ejemplo de música de gaita

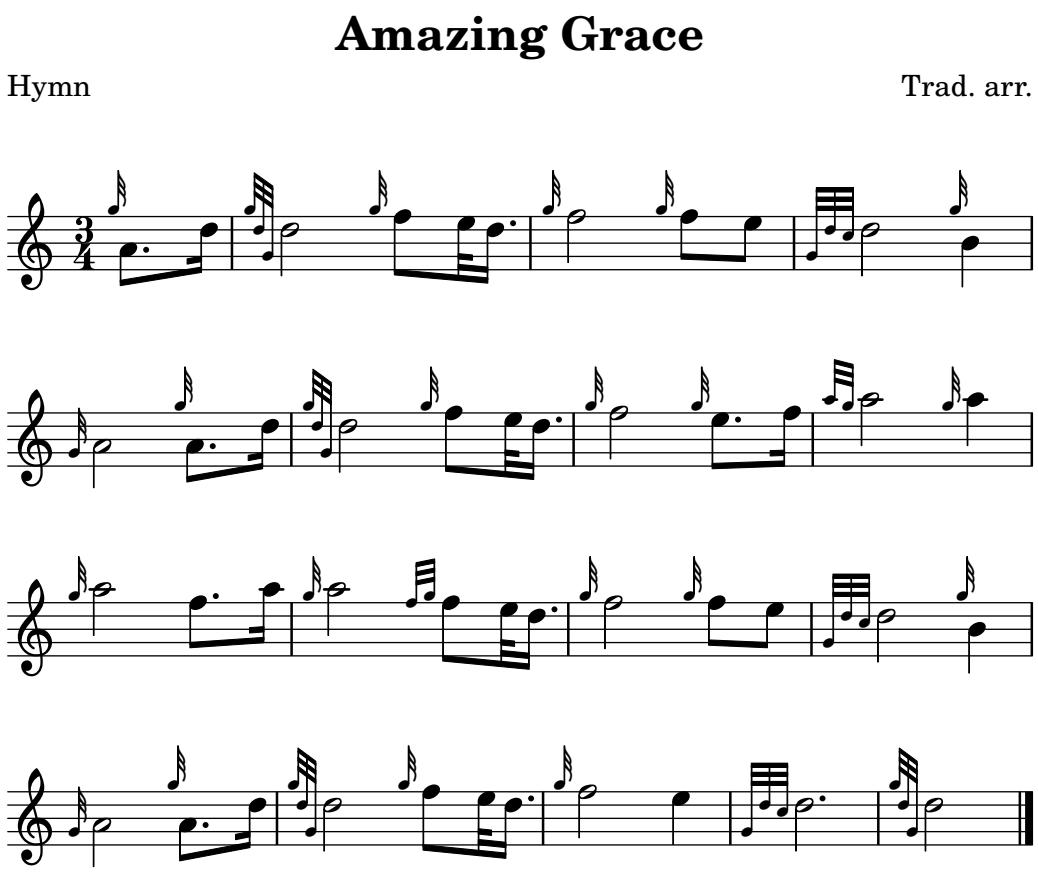
La conocida melodía Amazing Grace tiene este aspecto en notación de gaita.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
```

```
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdblf f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}
```



Véase también

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

2.6.3 Instrumentos de viento madera

Esta sección estudia la notación específica para instrumentos de viento madera.

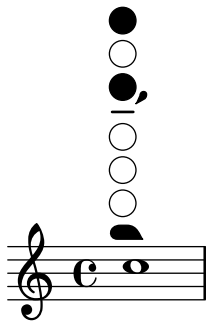
2.6.3.1 Diagramas de posiciones para viento madera

Se pueden utilizar diagramas de posiciones para los instrumentos de viento madera para indicar la digitación que usar al tocar ciertas notas, y están disponibles para los siguientes instrumentos:

- flautín
- flauta
- oboe
- clarinete
- clarinete bajo
- saxofón
- fagot
- contrafagot

Los diagramas se crean como elementos de marcado:

```
c1^ \markup \woodwind-diagram #'piccolo #'((lh . (gis))
                                           (cc . (one three))
                                           (rh . (ees)))
```



Las llaves pueden estar abiertas, parcialmente cubiertas, con el anillo pulsado o totalmente apretadas:

```
\textLengthOn
c1^\markup {
  \center-column {
    "one quarter"
    \woodwind-diagram #'flute #'((cc . (one1q))
                                 (lh . ())
                                 (rh . ()))
  }
}

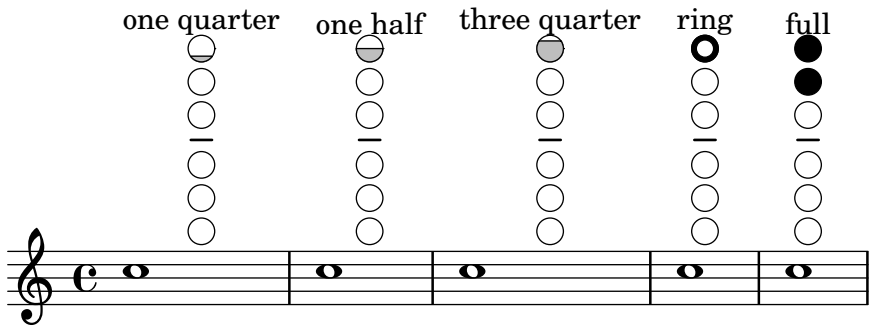
c1^\markup {
  \center-column {
    "one half"
    \woodwind-diagram #'flute #'((cc . (one1h))
                                 (lh . ())
                                 (rh . ()))
  }
}

c1^\markup {
  \center-column {
```

```
    "three quarter"
    \woodwind-diagram #'flute #'((cc . (one3q))
                                 (lh . ())
                                 (rh . ()))
  }
}

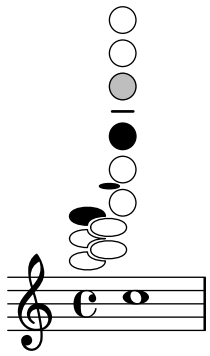
c1^\markup {
  \center-column {
    "ring"
    \woodwind-diagram #'flute #'((cc . (oneR))
                                 (lh . ())
                                 (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "full"
    \woodwind-diagram #'flute #'((cc . (oneF two))
                                 (lh . ())
                                 (rh . ()))
  }
}
```



Los trinos se indican mediante llaves sombreadas:

```
c1^\markup {
  \woodwind-diagram #'bass-clarinet
                    #'((cc . (threeT four))
                       (lh . ())
                       (rh . (b fis)))
}
```



Es posible imprimir una amplia variedad de trinos:

```
\textLengthOn
c1^\markup {
  \center-column {
    "one quarter to ring"
    \woodwind-diagram #'flute #'((cc . (one1qTR))
                                 (lh . ())
                                 (rh . ()))
  }
}

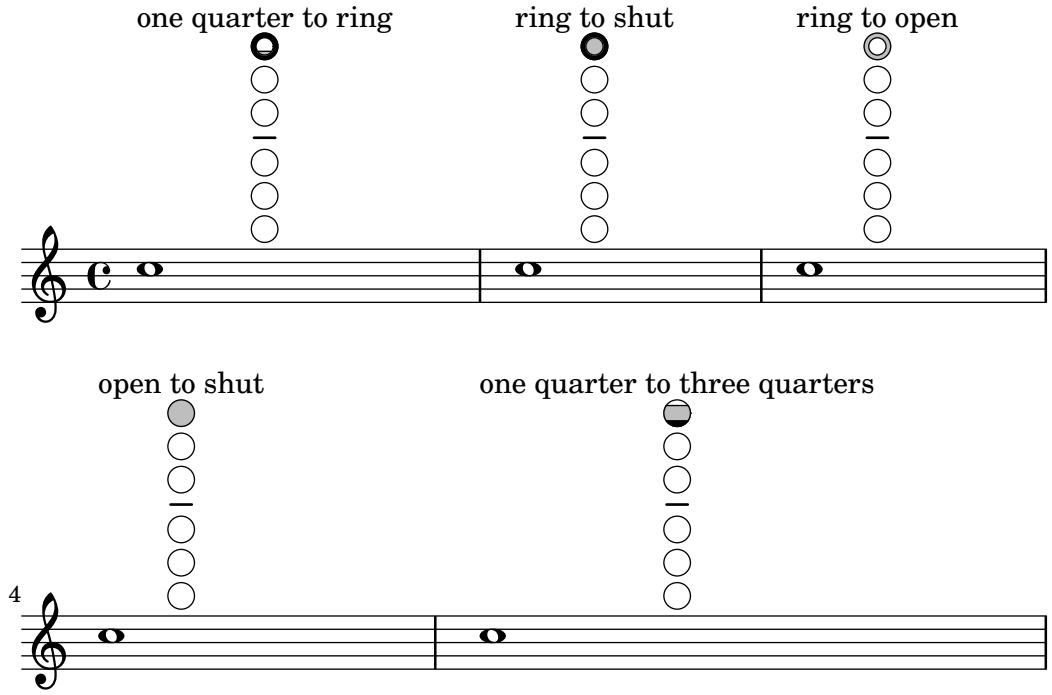
c1^\markup {
  \center-column {
    "ring to shut"
    \woodwind-diagram #'flute #'((cc . (oneTR))
                                 (lh . ())
                                 (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "ring to open"
    \woodwind-diagram #'flute #'((cc . (oneRT))
                                 (lh . ())
                                 (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "open to shut"
    \woodwind-diagram #'flute #'((cc . (oneT))
                                 (lh . ())
                                 (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "one quarter to three quarters"
    \woodwind-diagram #'flute #'((cc . (one1qT3q))
                                 (lh . ())
```

```
                                (rh . ()))
  }
}
```



La lista de todas las tonalidades y ajustes posibles para un instrumento dado se puede imprimir en la consola usando `#(print-keys-verbose 'flute)` o en el archivo de registro usando `#(print-keys-verbose 'flute (current-error-port))`, aunque no se pueden mostrar en la salida de música impresa.

Es posible crear diagramas nuevos, aunque requiere conocimientos de Scheme y quizá no esté al alcance de todos los usuarios. Los patrones de los diagramas están en '`scm/define-woodwind-diagrams.scm`' y en '`scm/display-woodwind-diagrams.scm`'.

Instrucciones predefinidas

Fragmentos de código seleccionados

Listado de los diagramas para viento madera

El fragmento de música que aparece a continuación presenta todos los diagramas de viento madera que se encuentran definidos en LilyPond por el momento.

```
\relative c' {
  \textLengthOn
  c1^
  \markup {
    \center-column {
      'tin-whistle
      " "
       \woodwind-diagram
                  #'tin-whistle
                  #'()
    }
  }
```

```
c1^
\markup {
  \center-column {
    'piccolo
    " "
     \woodwind-diagram
                #'piccolo
                #'()
  }
}

c1^
\markup {
  \center-column {
     'flute
     " "
     \woodwind-diagram
        #'flute
        #'()
  }
}
c1^\markup {
  \center-column {
    'oboe
    " "
    \woodwind-diagram
      #'oboe
      #'()
  }
}

c1^\markup {
  \center-column {
    'clarinet
    " "
    \woodwind-diagram
      #'clarinet
      #'()
  }
}

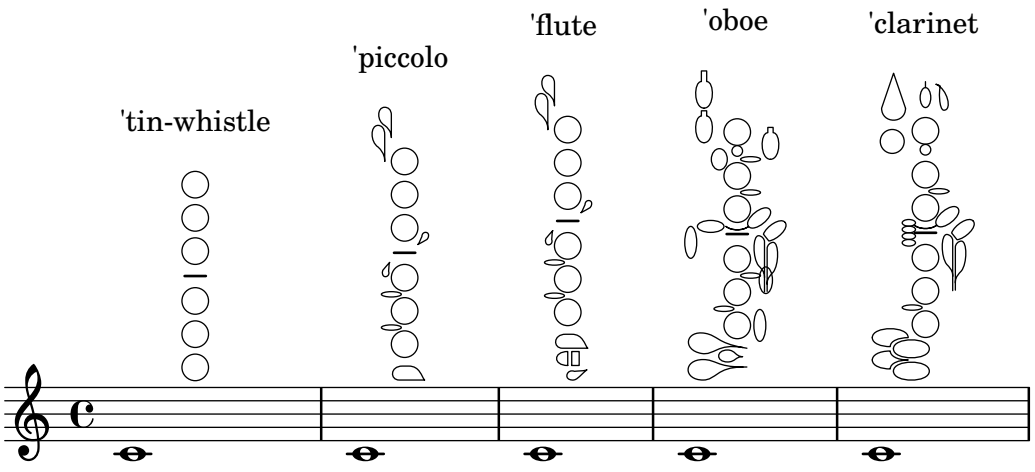
c1^\markup {
  \center-column {
    'bass-clarinet
    " "
    \woodwind-diagram
      #'bass-clarinet
      #'()
  }
}

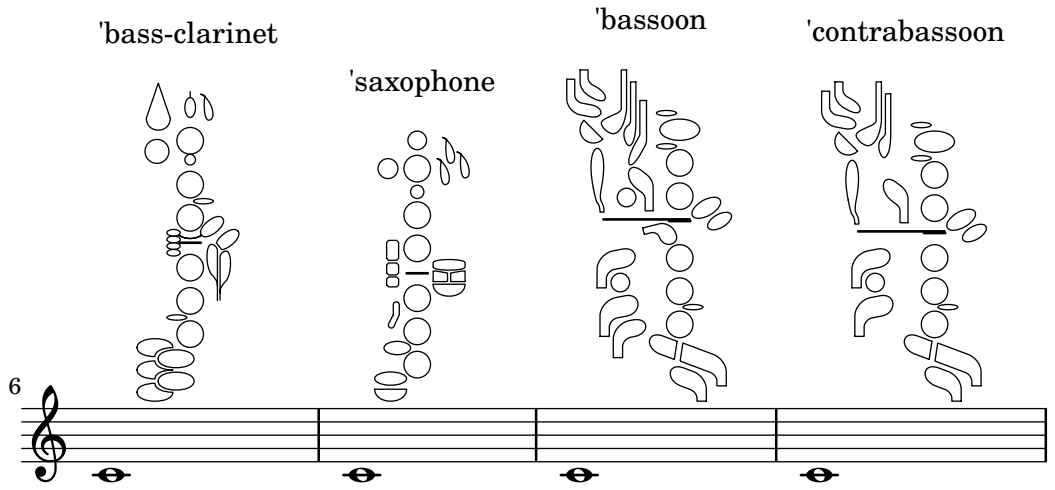
c1^\markup {
```

```
    \center-column {
      'saxophone
      " "
      \woodwind-diagram
        #'saxophone
        #'()
    }
  }

  c1^\markup {
    \center-column {
      'bassoon
      " "
      \woodwind-diagram
        #'bassoon
        #'()
    }
  }

  c1^\markup {
    \center-column {
      'contrabassoon
      " "
      \woodwind-diagram
        #'contrabassoon
        #'()
    }
  }
}
```



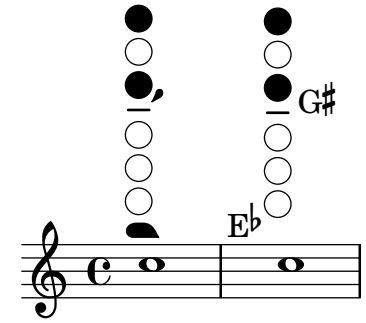


Diagramas para viento madera gráficos y textuales

En muchos casos, las llaves que no están en la columna central se pueden presentar por el nombre de la llave así como de forma gráfica.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'((cc . (one three))
         (lh . (gis))
         (rh . (ees)))

  c^\markup
    \override #'(graphical . #f) {
      \woodwind-diagram
        #'piccolo
        #'((cc . (one three))
           (lh . (gis))
           (rh . (ees)))
    }
}
```



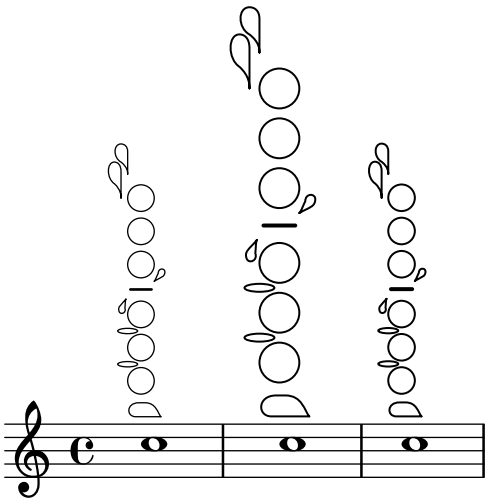
Modificar el tamaño de los diagramas de viento madera

Se puede cambiar el tamaño y grosor de las líneas de los diagramas de posiciones para instrumentos de viento madera.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
```

```
      #'piccolo
      #'()

  c^\markup
    \override #'(size . 1.5) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
  c^\markup
    \override #'(thickness . 0.15) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
}
```



Listas de llaves para los diagramas de viento madera

El fragmento de código que aparece a continuación produce una lista de todas las llaves y disposiciones de llave posibles para los diagramas de posiciones de instrumentos de viento madera, tal y como están definidos en '`scm/define-woodwind-diagrams.scm`'. La lista se muestra en el archivo de registro, pero no en la música. Si desea una salida por la consola, omita el `(current-error-port)` de las instrucciones.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
```

```
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))
```

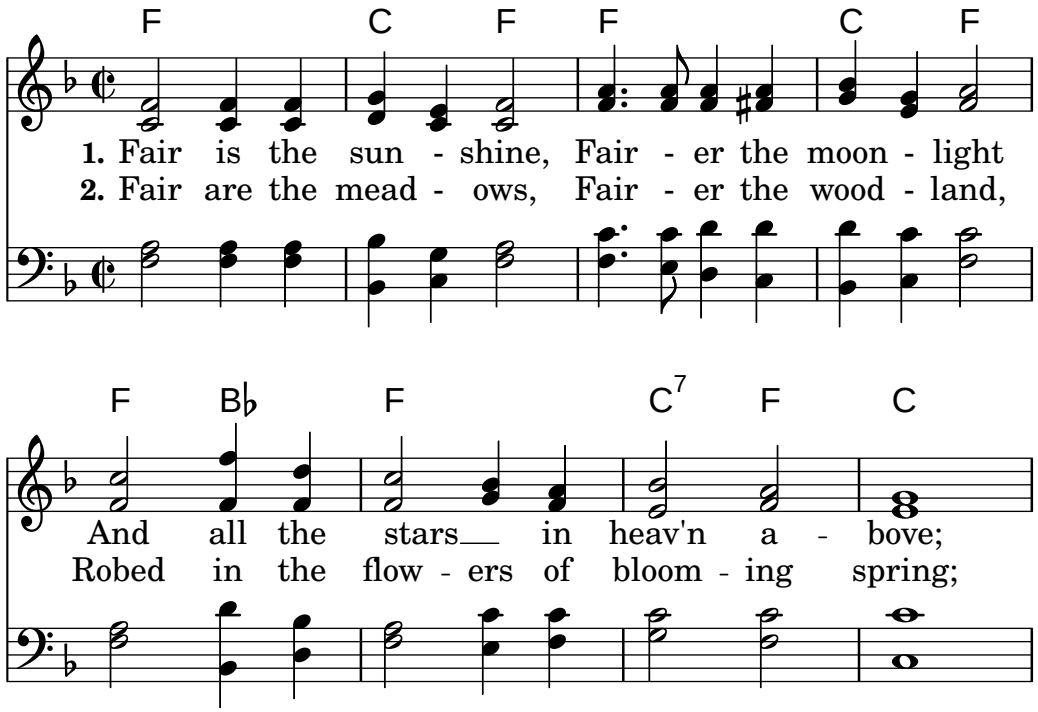
Véase también

Archivos de inicio: ‘scm/define-woodwind-diagrams.scm’, ‘scm/display-woodwind-diagrams.scm’.

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*, Sección “instrument-specific-markup-interface” in *Referencia de Funcionamiento Interno*.

2.7 Notación de acordes



Los acordes se pueden escribir en el modo de acordes, que reconoce ciertas convenciones europeas tradicionales de nomenclatura de acordes. También se pueden imprimir los nombres de los acordes. Además se puede imprimir notación de bajo cifrado.

2.7.1 Modo de acordes
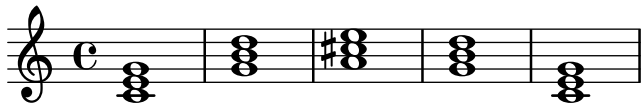
Para introducir acordes se utiliza el modo de acordes. Se usa un indicador de la estructura del acorde en lugar de la altura de sus notas.

Panorámica del modo de acordes

Los acordes se pueden escribir como música simultánea, como se explica en [Notas en acorde], página 161.

También se pueden escribir los acordes en el “modo de acordes”, que es un modo de entrada que se centra en las estructuras de los acordes dentro de la música europea tradicional (donde se conoce como «cifrado americano») más que en las notas concretas. Esto es muy práctico para los que están familiarizados con la utilización de nombres para describir los acordes. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 599.
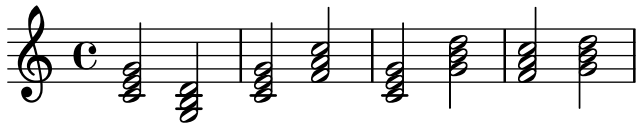
```
\chordmode { c1 g a g c }
```



Los acordes escritos con el modo de acordes son elementos musicales, y se pueden transportar igual que los acordes escritos mediante construcciones de música simultánea. `\chordmode` es absoluto, pues `\relative` no tiene ningún efecto sobre los bloques `chordmode`. Sin embargo, dentro de `\chordmode` las alturas absolutas son una octava más agudas que en el modo de notas.

Se pueden mezclar los modos de acorde y de nota en la música secuencial:

```
<c e g>2 <g b d>
\chordmode { c2 f }
<c e g>2 <g' b d>
\chordmode { f2 g }
```



Véase también

Glosario musical: Sección “chord” in *Glosario Musical*.

Referencia de la notación: [Notas en acorde], página 161, Sección 5.4.1 [Modos de entrada], página 599.

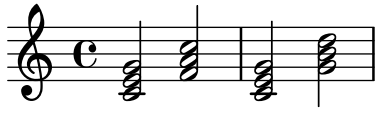
Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Advertencias y problemas conocidos

No se pueden usar las abreviaturas predefinidas de articulaciones y adornos sobre notas dentro del modo de acordes; véase [Articulaciones y ornamentos], página 116.
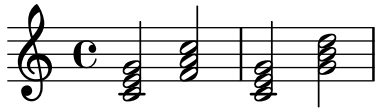
Cuando se mezclan los modos de acorde y de nota en música secuencial, y el modo de acordes aparece en primer lugar, el modo de notas crea un nuevo contexto de `Staff`:

```
\chordmode { c2 f }
<c e g>2 <g' b d>
```



Para evitar este comportamiento podemos crear explícitamente el contexto de `Staff`:

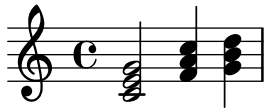
```
\new Staff {
  \chordmode { c2 f }
  <c e g>2  <g' b d>
}
```



Acordes más usuales
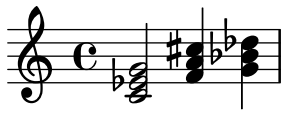
Las tríadas mayores se introducen escribiendo la fundamental y una duración opcional:

```
\chordmode { c2 f4 g }
```



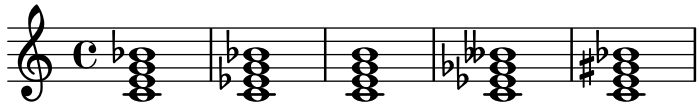
Las tríadas menores, aumentadas y disminuidas se escriben poniendo `:` y una cadena modificadora de variante después de la duración:

```
\chordmode { c2:m f4:aug g:dim }
```

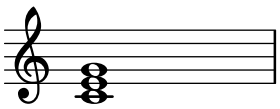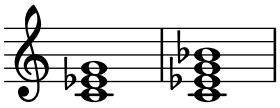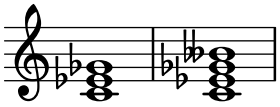
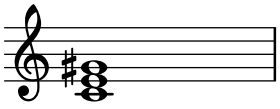


Se pueden crear acordes de séptima:

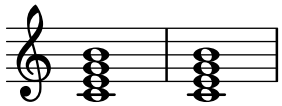
```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



La tabla que aparece más abajo muestra el efecto de los modificadores de variante sobre los acordes de tríada y de séptima. La séptima añadida de forma predeterminada a los acordes es menor, lo que hace del acorde de séptima dominante el acorde básico de séptima. Todas las alteraciones se consideran en relación con la séptima dominante. Hay una tabla más completa donde aparecen los usos de los modificadores en Sección A.2 [Modificadores de acorde más usuales], página 634.

Modificador	**Acción**	**Ejemplo**
Ninguno	Acción predeterminada; produce una tríada mayor.	
m, m7	Acorde menor. Este modificador baja la tercera.	
dim, dim7	Acorde disminuido. Este modificador baja la tercera, la quinta y, si existe, la séptima.	
aug	Acorde aumentado. Este modificador eleva la quinta.	

maj, maj7 — Acorde de séptima mayor. Este modificador añade una séptima elevada. El `7` que sigue a `maj` es opcional. NO utilice este modificador para crear una tríada mayor.



Véase también

Referencia de la notación: Sección A.2 [Modificadores de acorde más usuales], página 634, [Acordes extendidos y alterados], página 402.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Advertencias y problemas conocidos

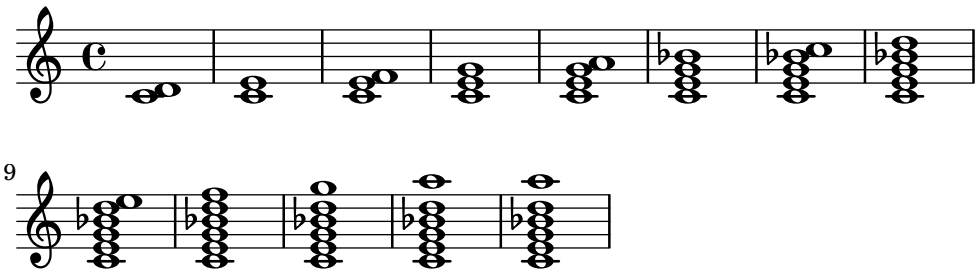
Sólo se puede usar un modificador por cada acorde, normalmente sobre la nota más aguda del mismo. Los acordes con más de un modificador se analizan sin producir errores ni advertencias, pero el resultado es impredecible. Los acordes que no se pueden conseguir con un solo modificador se deben alterar en sus notas individuales como se describe en [Acordes extendidos y alterados], página 402.

Acordes extendidos y alterados

Se pueden crear estructuras de acorde de complejidad arbitraria dentro del modo de acordes. Se puede usar la cadena modificadora para extender un acorde, añadir o quitar notas, elevar o bajar notas del acorde y añadir un bajo distinto o crear una inversión.

El primer número que sigue al `:` se considera que es el ámbito del acorde. El acorde se construye secuencialmente añadiendo terceras a la fundamental hasta que se alcanza el número especificado. Observe que la séptima añadida como parte de un acorde extendido es la séptima menor, no mayor. Si el ámbito no es una tercera (p.ej. 6), se añaden terceras hasta la tercera más alta inferior al ámbito, y después se añade la nota del ámbito. El mayor valor posible para el ámbito es 13. Cualquier valor mayor se interpreta como 13.

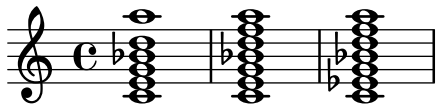
```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



Observe que `c:5` es idéntico a `c` (los dos producen una tríada de Do mayor).

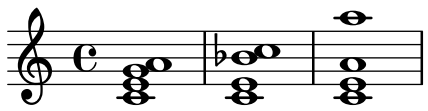
Puesto que una oncena sin alteración no suena bien cuando se combina con una trecena sin alteración, se elimina la oncena de los acordes de `:13` (a no ser que se añada explícitamente).

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



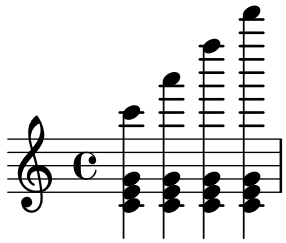
Se pueden añadir notas individuales a un acorde. Las adiciones siguen el ámbito y van prefijadas por un punto (`.`). La séptima normal que se añade a un acorde es la séptima menor, no mayor.

```
\chordmode {
  c1:5.6 c:3.7.8 c:3.6.13
}
```



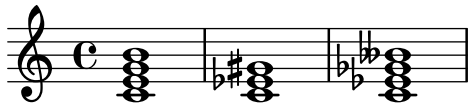
Las notas añadidas pueden ser tan agudas como se desee.

```
\chordmode {
  c4:5.15 c:5.20 c:5.25 c:5.30
}
```



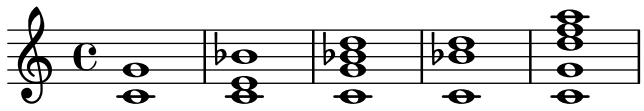
Las notas añadidas a los acordes se pueden alterar mediante la adición de los sufijos `-` o `+` al número. Para alterar una nota que se ha incluido automáticamente como parte de la estructura básica del acorde, añádala como una nota alterada.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



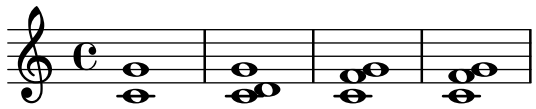
Una nota que se quiere suprimir de un acorde se indica dentro de la cadena modificadora prefijándola por un acento circunflejo `^`. Sólo se permite una supresión con `^` dentro de una cadena modificadora.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```



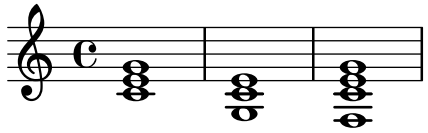
Se puede añadir el modificador `sus` a la cadena modificadora para crear acordes suspendidos. Esto elimina la tercera del acorde. Escriba bien `2` o bien `4` para añadir la segunda o la cuarta al acorde, respectivamente. `sus` equivale a `^3`; `sus4` equivale a `.4^3`.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4^3
}
```



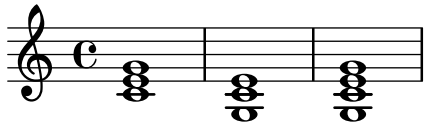
Se pueden especificar las inversiones (colocar en el bajo una nota distinta de la fundamental del acorde) y notas de bajo añadidas mediante la adición de `/`*nota* al acorde.

```
\chordmode {
  c1 c/g c/f
}
```



Se puede añadir una nota del bajo que forma parte del acorde, en lugar de ser movida por efecto de una inversión, mediante el uso de `/+`*nota*.

```
\chordmode {
  c1 c/g c/+g
}
```



Los modificadores de acorde que se pueden utilizar para producir una amplia variedad de acordes estándar se muestran en Sección A.2 [Modificadores de acorde más usuales], página 634.

Véase también
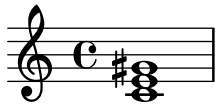
Referencia de la notación: Sección A.2 [Modificadores de acorde más usuales], página 634.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Advertencias y problemas conocidos

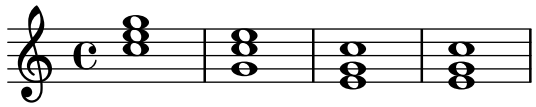
Cada nota sólo puede estar presente en el acorde una sola vez. Lo que sigue produce simplemente el acorde aumentado, porque `5+` se interpreta en último lugar.

```
\chordmode { c1:5.5-.5+ }
```



Sólo se puede crear la segunda inversión mediante la adición de un bajo. La primera inversión requiere cambiar la fundamental del acorde.

```
\chordmode {
  c'1: c':/g e:6-3-^5 e:m6-^5
}
```



2.7.2 Imprimir los acordes

Los acordes se pueden imprimir por su nombre, además de la impresión estándar como notas sobre un pentagrama.

Impresión de los nombres de acorde

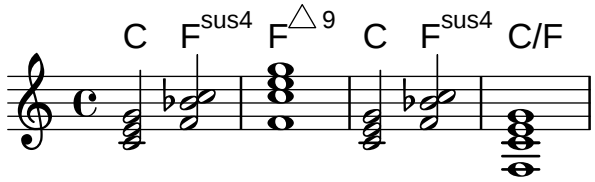
Los nombres de acorde se imprimen dentro del contexto `ChordNames`:

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```



Los acordes se pueden escribir como notas simultáneas o a través del uso del modo de acordes. El nombre de acorde que se imprime es el mismo independientemente del modo de entrada, a no ser que existan inversiones o bajos añadidos:
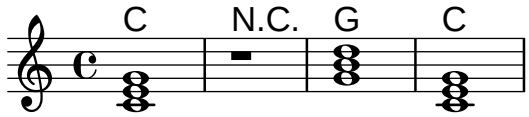
```
chordmusic = \relative c' {
  <c e g>2 <f bes c>
  <f c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
<<
  \new ChordNames {
    \chordmusic
  }
  {
    \chordmusic
  }
>>
```



Los silencios pasados a un contexto `ChordNames` hacen que se imprima el elemento de marcado `noChordSymbol` (símbolo de N.C., Sin Acorde).

```
<<
  \new ChordNames \chordmode {
    c1
    r1
    g1
    c1
  }
  \chordmode {
    c1
    r1
```

```
    g1
    c1
  }
>>
```



`\chords { ... }` es una forma abreviada de escribir `\new ChordNames { \chordmode { ... } }`.

```
\chords {
  c2 f4.:m g8:maj7
}
```




```
\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}
```

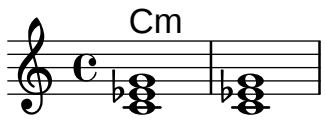


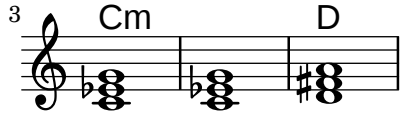
Fragmentos de código seleccionados

Imprimir los acordes cuando se produce un cambio

Se pueden imprimir los acordes exclusivamente al comienzo de las líneas y cuando cambia el acorde.

```
harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies
  }
  \new Staff {
    \relative c' { \harmonies }
  }
>>
```

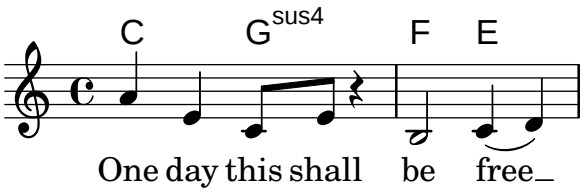




Hoja guía de acordes o «lead sheet» sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Véase también

Glosario musical: Sección “chord” in *Glosario Musical*.

Referencia de la notación: [Escribir música en paralelo], página 180.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” in *Referencia de Funcionamiento Interno*, Sección “ChordName” in *Referencia de Funcionamiento Interno*, Sección “Chord_name_engraver” in *Referencia de Funcionamiento Interno*, Sección “Volta_engraver” in *Referencia de Funcionamiento Interno*, Sección “Bar_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

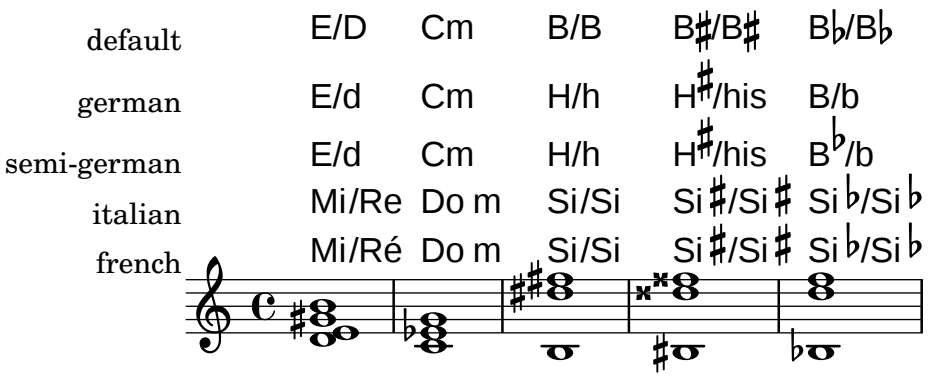
Los acordes que contienen inversiones o bajos alterados no reciben un nombre adecuado si se escriben usando música simultánea.

Nombres de acorde personalizados

No hay un sistema único para nombrar acordes. Las distintas tradiciones musicales usan distintos nombres para el mismo conjunto de acordes. También hay distintos símbolos impresos para un nombre de acorde dado. Los nombres y símbolos que se imprimen para los nombres de acorde dados se pueden personalizar.

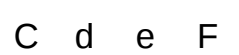
El esquema de nomenclatura de acordes predeterminado es un sistema para música de Jazz, propuesto por Klaus Ignatzek (véase Sección “Lista bibliográfica” in *Ensayo*). También funcionan otros dos esquemas de nomenclatura de acordes: una notación de acordes de Jazz alternativa y un esquema sistemático llamado Acordes de Banter. La notación de Jazz alternativa también se puede ver en el cuadro Sección A.1 [Cuadro de nombres de acordes], página 633.

Además de los distintos sistemas de nomenclatura, se usan diferentes nombres de nota para la fundamental en los distintos idiomas. Las variables predefinidas `\germanChords`, `\semiGermanChords`, `\italianChords` y `\frenchChords` establecen el valor de estas variables. El efecto se muestra aquí:



Algunos cancioneros alemanes indican los acordes menores como letras minúsculas, sin el sufijo *m*. Esto se puede conseguir estableciendo la propiedad `chordNameLowercaseMinor`:

```
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```



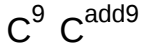
Si ninguno de los ajustes predeterminados proporciona el resultado deseado, se puede afinar la presentación de los nombres de acorde a través de las siguientes propiedades.

`chordRootNamer`
: El nombre de acorde se imprime normalmente como una letra para la fundamental con una alteración opcional. La transformación de la nota en la letra se realiza por parte de esta función. Los nombres de nota especiales (por ejemplo, la 'H' alemana para un acorde de Si) se pueden producir almacenando una función nueva en esta propiedad.

`majorSevenSymbol`
: Esta propiedad contiene el objeto de marcado que se usa para hacer el seguimiento de la salida de `chordRootNamer` para identificar un acorde de séptima mayor. Las opciones predefinidas son `whiteTriangleMarkup` (triángulo blanco) y `blackTriangleMarkup` (triángulo negro).

`additionalPitchPrefix`
: Cuando el acorde contiene notas añadidas, se pueden prefijar opcionalmente por un texto. Por omisión no hay ningún prefijo, para evitar la sobrecarga visual, pero para pequeñas cantidades de notas añadidas puede ser visualmente efectivo.

```
\new ChordNames {
  <c e g d'>    % add9
  \set additionalPitchPrefix = #"add"
  <c e g d'>    % add9
}
```

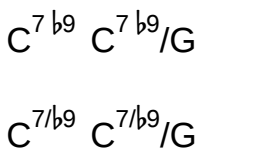


`chordNoteNamer`
: Cuando el nombre del acorde contiene notas adicionales aparte de la fundamental (p.ej., un bajo añadido), se utiliza esta función para imprimir la nota adicional. De forma predeterminada se imprime la nota usando `chordRootNamer`. La propiedad `chordNoteNamer` se puede establecer a una función especializada para cambiar este comportamiento. Por ejemplo, el bajo se puede imprimir en minúscula.

`chordNameSeparator`

Las diferentes partes del nombre de un acorde se separan normalmente mediante un pequeño espacio en blanco. Mediante el ajuste de `chordNameSeparator`, podemos usar cualquier marcado que deseemos como separador. Esto no afecta al separador entre un acorde y la nota del bajo: para personalizar eso, utilice `slashChordSeparator`.

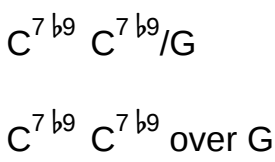
```
\chords {
  c4:7.9- c:7.9-/g
  \set chordNameSeparator = \markup { "/" }
  \break
  c4:7.9- c:7.9-/g
}
```



`slashChordSeparator`

Los acordes se pueden tocar sobre una nota del bajo distinta a la fundamental. Se llaman "inversiones" o "acordes de barra inclinada" porque la manera predeterminada de efectuar su notación es con una barra inclinada entre el acorde y la nota del bajo. Por tanto, el valor de `slashChordSeparator` es, por omisión, una barra inclinada, pero podemos cambiarlo a cualquier elemento de marcado que queramos.

```
\chords {
  c4:7.9- c:7.9-/g
  \set slashChordSeparator = \markup { " over " }
  \break
  c4:7.9- c:7.9-/g
}
```



`chordNameExceptions`

Esta propiedad es una lista de parejas. El primer elemento de cada pareja es un conjunto de notas utilizadas para identificar los elementos presentes en el acorde. El segundo elemento es un elemento de marcado que sigue a la salida de `chordRootNamer` para crear el nombre del acorde.

`minorChordModifier`

La notación de los acordes menores se suele hacer por medio del sufijo 'm' a la derecha de la letra que indica la fundamental del acorde. Sin embargo, otras convenciones dan preferencia a sufijos diferentes, como un signo menos.

```
\chords {
  c4:min f:min7
  \set minorChordModifier = \markup { "-" }
  \break
  c4:min f:min7
}
```

Cm Fm^{7}

C- F-^{7}

`chordPrefixSpacer`
: El modificador de los acordes menores tal y como viene determinado por `minorChordModifier` se imprime por lo general inmediatamente a la derecha de la fundamental. Se puede insertar un símbolo separador entre la fundamental y el modificador estableciendo `chordPrefixSpacer`. El símbolo separador no se usa cuando el acorde está alterado.

Instrucciones predefinidas

`\whiteTriangleMarkup`, `\blackTriangleMarkup`, `\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

Fragmentos de código seleccionados

Excepciones para los nombres de acorde

Se puede usar la propiedad `chordNameExceptions` para almacenar una lista de notaciones espaciales para acordes específicos.

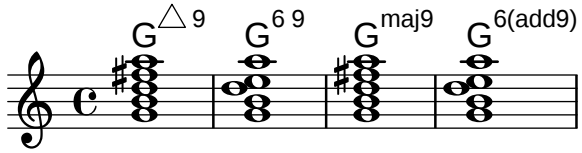
```
% modify maj9 and 6(add9)
% Exception music is chords with markups
chExceptionMusic = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

% Convert music to list and prepend to existing exceptions.
chExceptions = #( append
  ( sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

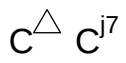
<< \context ChordNames \theMusic
   \context Voice \theMusic
>>
```



Nombre de acorde maj7

La presentación del acorde de séptima mayor se puede ajustar mediante majorSevenSymbol.

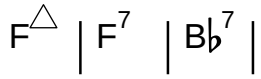
```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

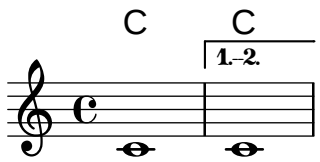
```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-2 . 2)
  \consists "Bar_engraver"
}
\chordmode {
  f1:maj7 f:7 bes:7
}
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

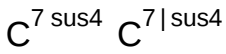
```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Modificación del separador de acordes

Se puede establecer el separador entre las distintas partes del nombre de un acorde para que sea cualquier elemento de marcado.

```
\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter | }
  c:7sus4
}
```



Véase también

Referencia de la notación: Sección A.1 [Cuadro de nombres de acordes], página 633, Sección A.2 [Modificadores de acorde más usuales], página 634.

El ensayo sobre grabado musical automatizado: Sección “Lista de referencias bibliográficas” in *Ensayo*.

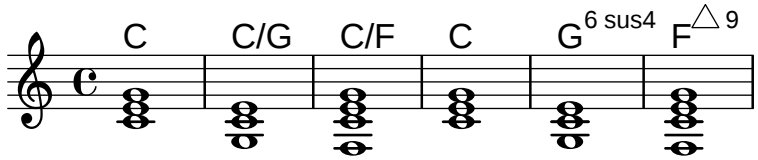
Archivos instalados: ‘`scm/chords-ignatzek.scm`’, ‘`scm/chord-entry.scm`’, ‘`ly/chord-modifier-init.ly`’.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.
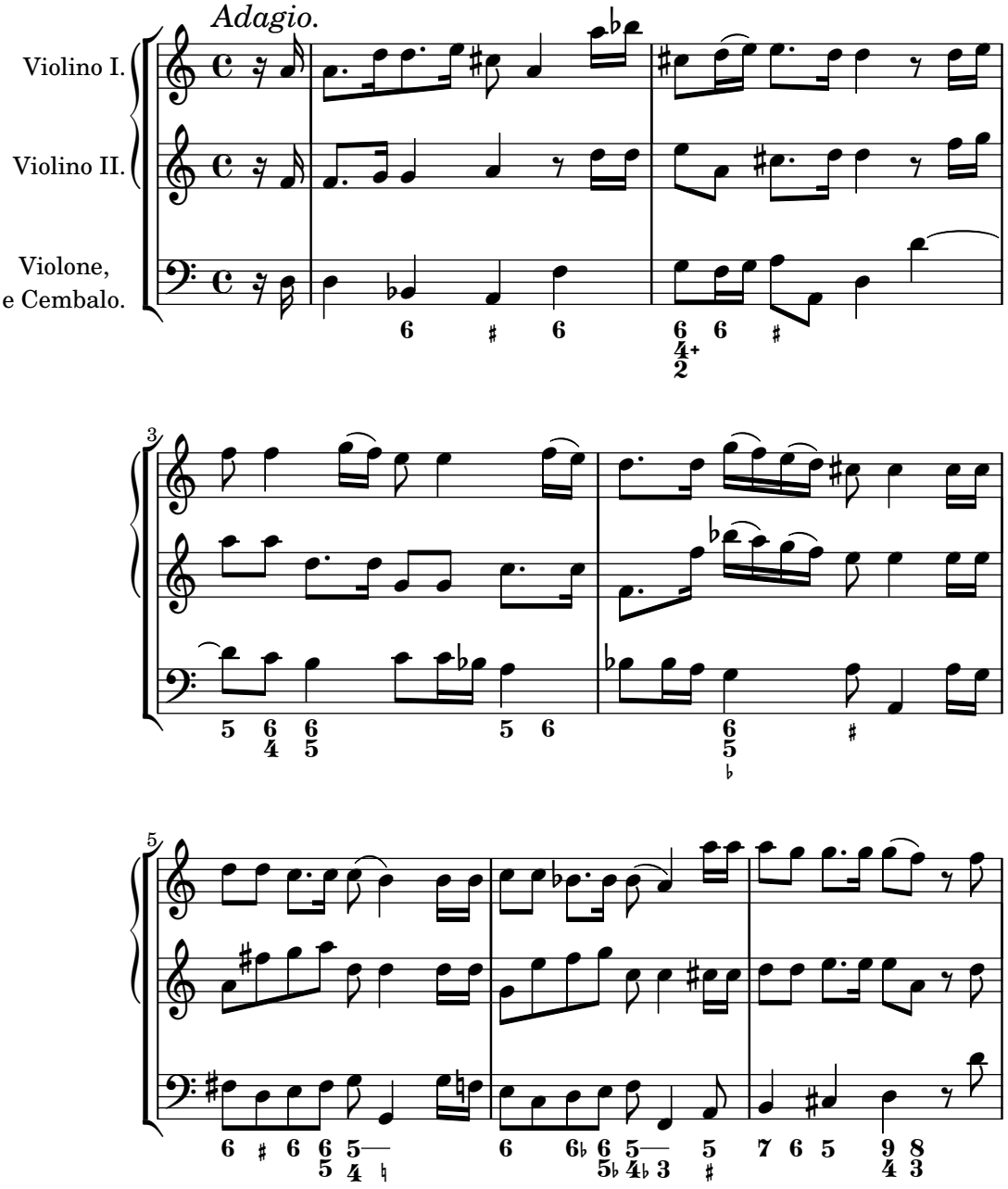
Advertencias y problemas conocidos

Los nombres de acorde se determinan a partir de las notas que están presentes en el mismo y de la información acerca de la estructura del acorde que ha podido ser introducida en `\chordmode`. Si el método de las notas simultáneas se usa para introducir los acordes, pueden resultar nombres no deseados a partir de las inversiones o de las notas del bajo.

```
myChords = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>
```



2.7.3 Bajo cifrado



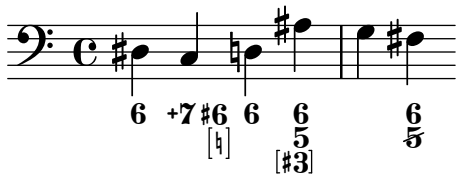
Se puede imprimir notación de bajo cifrado.

Introducción al bajo cifrado

LilyPond contempla el bajo cifrado, también conocido como basso continuo:

```
<<
  \new Voice { \clef bass dis4 c d ais g fis}
  \new FiguredBass {
    \figuremode {
      < 6 >4 < 7\+ >8 < 6+ [_!] >
      < 6 >4 <6 5 [3+] >
      < _ >4 < 6 5/>4
    }
```

```
  }
>>
```



El soporte para bajo cifrado se compone de dos partes: hay un modo de entrada, presentado por `\figuremode`, que admite la entrada de las cifras del bajo, y hay un contexto llamado `FiguredBass` que se ocupa de la impresión de los objetos del tipo `BassFigure`. El bajo cifrado también se puede imprimir dentro de contextos de `Staff`.

`\figures{ ... }` es una notación abreviada de `\new FiguredBass { \figuremode { ... } }`.

Aunque el soporte para bajo cifrado puede parecerse superficialmente al de los acordes, realmente es mucho más simple. El modo `\figuremode` simplemente almacena las cifras y el contexto `FiguredBass` las imprime tal y como se han introducido. No hay ninguna conversión a notas.

Véase también

Glosario musical: Sección “figured bass” in *Glosario Musical*.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Introducir el bajo cifrado

Se usa `\figuremode` para cambiar el modo de entrada al modo de cifras. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 599.

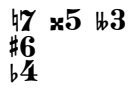
En el modo de cifras, un grupo de cifras del bajo está delimitado mediante ángulos simples, `<` y `>`. La duración se escribe después del `>`.

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```



Se pueden añadir alteraciones accidentales (incluso becuadros) a las cifras:

```
\figures {
  <7! 6+ 4-> <5++> <3-->
}
```



Se pueden indicar elementos del acorde aumentados y disminuidos:

```
\figures {
  <6\+ 5/> <7/>
}
```



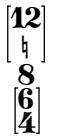
Se puede crear una barra invertida cruzando una cifra (usada normalmente para sextas elevadas):

```
\figures {
  <6> <6\\>
}
```



Se pueden incluir espacios verticales y corchetes en las cifras:

```
\figures {
  <[12 _!] 8 [6  4]>
}
```



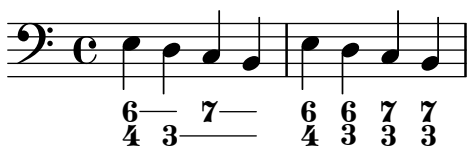
Se puede insertar cualquier elemento de marcado de texto como una cifra:

```
\figures {
  <\markup { \tiny \number 6 \super (1) } 5>
}
```



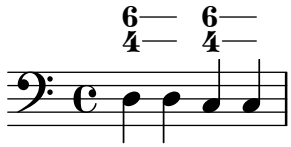
Se pueden usar líneas de continuación para indicar cifras que se repiten:

```
<<
  {
    \clef bass
    e4 d c b,
    e4 d c b,
  }
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 3> <7 3> <7 3>
    \bassFigureExtendersOff
    <6 4>4 <6 3> <7 3> <7 3>
  }
>>
```

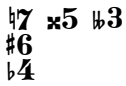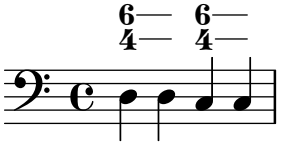


En este caso, las líneas de extensión sustituyen a las cifras existentes, a no ser que las líneas de continuación hayan sido terminadas explícitamente.

```
<<
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 4> <6\! 4\!> <6 4>
  }
  {
    \clef bass
    d4 d c c
  }
>>
```



La siguiente tabla resume los modificadores de cifrado disponibles.

Modificador	Propósito	Ejemplo
+, -, !	Alteraciones	
\+, /	Aumentaciones y disminuciones	
\\	Sexta elevada	
\!	Fin de línea de continuación	

Instrucciones predefinidas

`\bassFigureExtendersOn`, `\bassFigureExtendersOff`.

Fragmentos de código seleccionados

Cambiar las posiciones de las alteraciones del bajo cifrado

Las alteraciones y los signos «más» pueden aparecer antes o después de los números, según el valor de las propiedades `figuredBassAlterationDirection` y `figuredBassPlusDirection`.

```
\figures {
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassPlusDirection = #RIGHT
  <6\+> <5+> <6 4-> r
```

```
  \set figuredBassAlterationDirection = #LEFT
  <6\+> <5+> <6 4-> r
}
```



Véase también

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

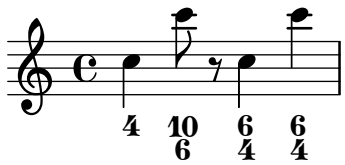
Referencia de funcionamiento interno: Sección “BassFigure” in *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” in *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” in *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” in *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*.

Imprimir el bajo cifrado

El bajo cifrado se puede imprimir usando el contexto `FiguredBass`, o dentro de la mayoría de los contextos de pentagrama.

Cuando se presenta dentro de un contexto de `FiguredBass`, la ubicación vertical de las cifras es independiente de las notas que están en el pentagrama.

```
<<
  \relative c'' {
    c4 c'8 r8 c,4 c'
  }
  \new FiguredBass {
    \figuremode {
      <4>4 <10 6>8 s8
      <6 4>4 <6 4>
    }
  }
>>
```

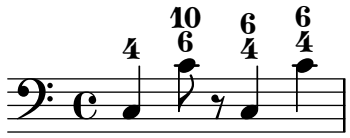


En el ejemplo de arriba, el contexto `FiguredBass` se debe instanciar explícitamente para evitar crear un segundo pentagrama vacío.

El bajo cifrado se puede añadir también a contextos de `Staff` directamente. En este caso, la posición vertical de las cifras se ajusta automáticamente.

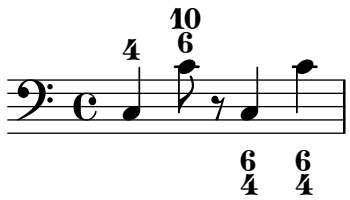
```
<<
  \new Staff = "myStaff"
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
  %% Put notes on same Staff as figures
  \context Staff = "myStaff"
  {
```

```
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



Cuando se escribe dentro de un contexto de pentagrama, el bajo cifrado se puede presentar encima o debajo del mismo.

```
<<
  \new Staff = "myStaff"
  \figuremode {
    <4>4 <10 6>8 s8
    \bassFigureStaffAlignmentDown
    <6 4>4 <6 4>
  }
  %% Put notes on same Staff as figures
  \context Staff = "myStaff"
  {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



Instrucciones predefinidas

`\bassFigureStaffAlignmentDown`, `\bassFigureStaffAlignmentUp`, `\bassFigureStaffAlignmentNeutral`.

Véase también

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.
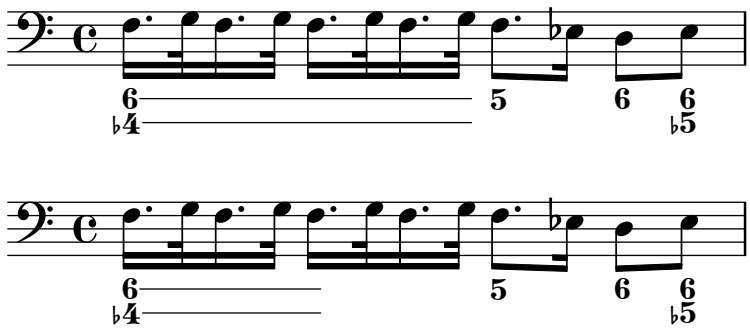
Referencia de funcionamiento interno: Sección “BassFigure” in *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” in *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” in *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” in *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Para asegurar que las líneas extensoras funcionan adecuadamente, lo más seguro es utilizar las mismas duraciones en la línea de cifras y en la línea del bajo.

```
<<
  {
```

```
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % The extenders are correct here, with the same rhythm as the bass
    \repeat unfold 4 { <6 4->16. <6 4->32 }
    <5>8. r16 <6>8 <6\! 5->
  }
>>
<<
  {
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % The extenders are incorrect here, even though the timing is the same
    <6 4->4 <6 4->4
    <5>8. r16 <6>8 <6\! 5->
  }
>>
```



2.8 Música contemporánea

A partir de comienzos del s.XX se ha producido una expansión masiva de los estilos y técnicas composicionales. Los nuevos desarrollos armónicos y rítmicos, una expansión del espectro de alturas y el desarrollo de un amplio abanico de nuevas técnicas instrumentales han venido acompañadas de una evolución y expansión paralelas de la notación musical. El objetivo de esta sección es proporcionar referencias e información relevante para el trabajo con estas nuevas técnicas notacionales.

2.8.1 Altura y armonía en la música contemporánea

Esta sección destaca los problemas de la notación de las alturas y la armonía en la música contemporánea.

Referencias para alturas y armonía en música contemporánea

- La notación estándar de cuartos de tono se estudia en [Nombres de las notas en otros idiomas], página 7.
- Las armaduras de tonalidad no estándar se estudian en [Armadura de la tonalidad], página 20.

- Las prácticas contemporáneas en la presentación de las alteraciones accidentales se estudian en [Alteraciones accidentales automáticas], página 26.

Notación microtonal

Armonía y armaduras de tonalidad contemporáneas

2.8.2 Enfoques contemporáneos del ritmo

Esta sección destaca los problemas de la notación de las duraciones en la música contemporánea.

Referencias para enfoque contemporáneos del ritmo

- Los tipos de compás de amalgama se estudian en [Indicación de compás], página 62.
- La notación polimétrica básica se estudia en [Notación polimétrica], página 72.
- Las barras en ángulo progresivas se estudian en [Barras progresivas], página 93.
- Las líneas divisorias de estilo Mensurstriche (líneas que están solamente entre los pentagramas) se estudian en [Agrupar pentagramas], página 184.

Grupos de valoración especial en música contemporánea

Compases contemporáneos

Notación polimétrica extendida

Barrado en música contemporánea

Líneas divisorias en música contemporánea

2.8.3 Notación gráfica

2.8.4 Técnicas de partitura contemporáneas

2.8.5 Nuevas técnicas instrumentales

2.8.6 Lecturas adicionales y partituras de interés

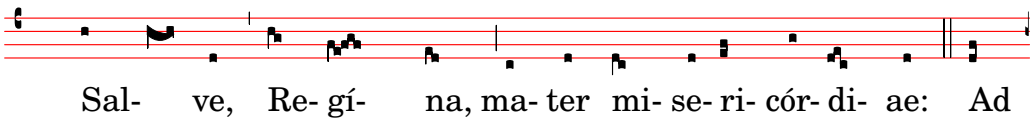
Esta sección sugiere libros, ejemplos musicales y otros recursos de utilidad en el estudio de la notación musical contemporánea.
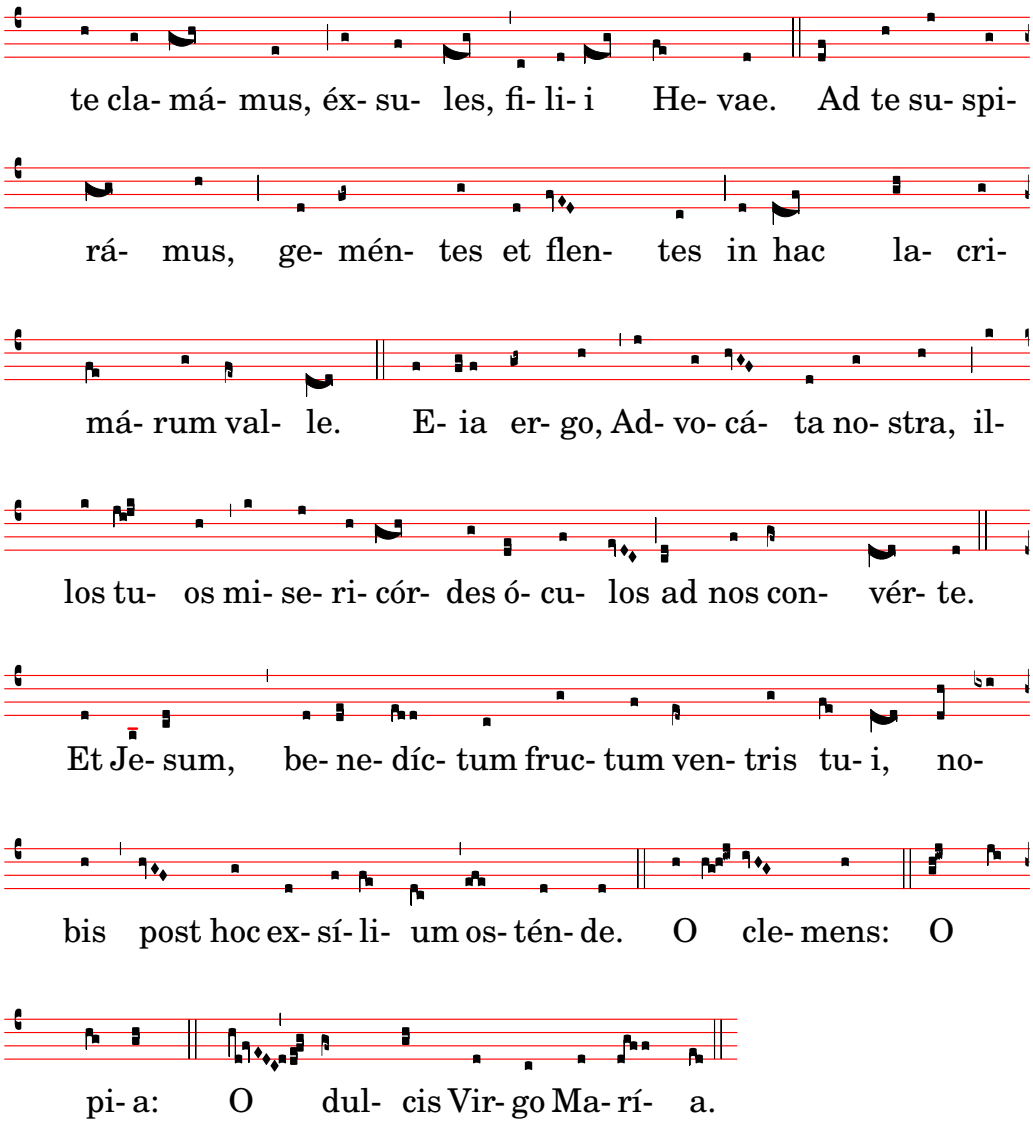
Libros y artículos sobre notación musical contemporánea

- *Music Notation in the Twentieth Century: A Practical Guidebook* de Kurt Stone [W. W. Norton, 1980]
- *Music Notation: A Manual of Modern Practice* de Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* de Alfred Blatter [Schirmer, 2nd ed. 1997]

Partituras y ejemplos musicales

2.9 Notación antigua





Entre las formas en que está contemplada la notación antigua se encuentra la posibilidad de tipografiar notación mensural, canto gregoriano y canto kievano en notación cuadrada. Se puede acceder a estas funcionalidades mediante la modificación de propiedades de estilo de objetos gráficos como la cabeza de las notas o los silencios, o bien mediante la utilización de uno de los contextos predefinidos para estos estilos.

Muchos objetos gráficos, como la cabeza y el corchete de las figuras, las alteraciones, la indicación de compás y los silencios ofrecen una propiedad `style` (estilo), que se puede cambiar para emular varios estilos diferentes de notación antigua. Véase

- [Cabezas de nota de la música mensural], página 427,
- [Alteraciones y armaduras de la música mensural], página 430,
- [Silencios de la música mensural], página 429,
- [Claves de la música mensural], página 425,
- [Claves de canto gregoriano], página 433,
- [Corchetes de la música mensural], página 428,
- [Indicaciones de compás de la música mensural], página 426.

Ciertos conceptos de notación se incluyen específicamente para la notación antigua:

- [Custos], página 424,
- [Divisiones], página 435,
- [Ligaduras], página 423.

Véase también

Glosario musical: Sección “custos” in *Glosario Musical*, Sección “ligature” in *Glosario Musical*, Sección “mensural notation” in *Glosario Musical*.

Referencia de la notación: [Cabezas de nota de la música mensural], página 427, [Alteraciones y armaduras de la música mensural], página 430, [Silencios de la música mensural], página 429, [Claves de canto gregoriano], página 433, [Corchetes de la música mensural], página 428, [Indicaciones de compás de la música mensural], página 426, [Custos], página 424, [Divisiones], página 435, [Ligaduras], página 423.

2.9.1 Panorámica de los estilos contemplados

Están disponibles tres estilos para la composición tipográfica del canto gregoriano:

- *Editio Vaticana* es un estilo completo para canto gregoriano, que sigue la apariencia de las ediciones de Solesmes, los libros de canto oficiales del Vaticano desde 1904. LilyPond contempla todos los signos de notación que se utilizan en este estilo, entre ellos las ligaduras, *custodias* y símbolos especiales como el quilisma y el oriscus.
- El estilo *Editio Medicaea* ofrece ciertas funcionalidades que se usan en las ediciones Medicaea (o de Ratisbona) que se usaban con anterioridad a las ediciones de Solesmes. Las diferencias más significativas con respecto al estilo *Vaticana* son las claves, que tienen trazos descendentes, y la cabeza de las figuras, que son cuadradas y regulares.
- El estilo *Hufnagel* (en “uña de herradura”) o *gótico* reproduce el estilo de escritura en los manuscritos de canto medievales de Alemania y Centro Europa. Recibe el nombre de la forma de la nota básica (la *virga*), que parece una pequeña uña.

Tres estilos emulan la apariencia de los manuscritos y ediciones impresas de música mensural tardomedievales y renacentistas:

- El estilo *Mensural* se asemeja bastante al estilo de escritura utilizado en los manuscritos tardomedievales y renacentistas tempranos, con la cabeza de sus figuras en forma de rombo pequeñas y estrechas, y los silencios que imitan un estilo dibujado a mano.
- El estilo *Neomensural* es una versión modernizada y estilizada del anterior: la cabeza de las figuras es más ancha y los silencios están formados por trazos rectilíneos. Este estilo está particularmente indicado, p.ej., para los *incipits* de transcripciones de fragmentos de música mensural.
- El estilo *Petrucci* recibe su nombre de Ottaviano Petrucci (1466-1539), el primer impresor en utilizar tipos móviles para la música (en su *Harmonice musices odhecaton*, 1501). Este estilo utiliza para las figuras una cabeza más grande que los otros estilos mensurales.

Baroque y *Classical* no son estilos completos pero difieren del estilo predeterminado solamente en ciertos detalles: algunas cabezas de nota (Baroque) y el silencio de negra (Classical).

Sólo el estilo mensural tiene alternativas para todos los aspectos de la notación. Así, no existen silencios ni corchetes en los estilos gregorianos, pues estos símbolos no se usan en la notación del canto llano, y el estilo Petrucci no posee corchetes ni alteraciones por sí mismo.

Cada elemento de la notación se puede cambiar independientemente de los otros, y así podemos usar corchetes mensurales, cabezas de petrucci, silencios de classical y claves de vaticana en la misma pieza, si queremos.

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*, Sección “flag” in *Glosario Musical*.

2.9.2 Notación antigua - funcionalidades comunes

Contextos predefinidos

Para el canto gregoriano y la notación mensural, están a nuestra disposición contextos predefinidos de voz y de pauta, que establecen todos los símbolos de notación a unos valores adecuados para estos estilos. Si nos satisfacen estos valores predeterminados, podemos a continuación introducir directamente las notas sin la necesidad de preocuparnos de los detalles sobre cómo personalizar un contexto. Véanse uno de los contextos predefinidos `VaticanaVoice`, `VaticanaStaff`, `MensuralVoice` y `MensuralStaff`.

- [Contextos del canto gregoriano], página 433,
- [Contextos de la música mensural], página 424.

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*.
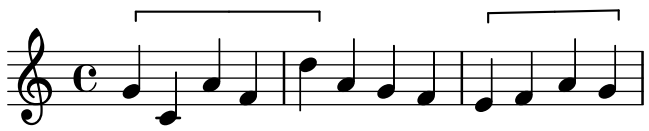
Referencia de la notación: [Contextos del canto gregoriano], página 433, [Contextos de la música mensural], página 424.

Ligaduras

Una ligadura es un símbolo gráfico que representa al menos dos notas distintas. Las ligaduras aparecieron originalmente en los manuscritos de la notación del canto gregoriano para denotar secuencias de notas ascendentes o descendentes en la misma sílaba. Se usan también en notación mensural.

Las ligaduras se escriben *encerrándolas* dentro de un par `\[` y `\]`. Ciertos estilos de ligadura podrían necesitar una sintaxis de entrada adicional específica para este tipo de ligadura en particular. De forma predeterminada, el grabador `LigatureBracket` se limita a colocar un corchete recto sobre la ligadura:

```
\relative c'' {
  \[ g c, a' f d' \]
  a g f
  \[ e f a g \]
}
```



Están disponibles otros dos estilos de ligaduras: el Vaticana para canto gregoriano, y el Mensural para música mensural (sólo están contempladas las ligaduras mensurales blancas para la música mensural, y con ciertas limitaciones). Para usar cualquiera de estos estilos, se debe sustituir el grabador predeterminado `Ligature_bracket_engraver` por uno de los grabadores especializados en ligaduras del contexto `Voice`, como se explica en [Ligaduras mensurales blancas], página 431 y [Ligaduras de neumas cuadrados gregorianos], página 437.

Véase también

Glosario musical: Sección “ligature” in *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 431, [Ligaduras de neumas cuadrados gregorianos], página 437.

Advertencias y problemas conocidos

Las ligaduras necesitan un espaciado especial que aún no se ha desarrollado. Como consecuencia, la mayor parte del tiempo existe una separación excesiva entre las ligaduras, y a menudo los

cortes de línea no son satisfactorios. Además, la letra de los cantos no se alinea correctamente con las ligaduras.

Las alteraciones accidentales no se deben imprimir dentro de las ligaduras, sino que deben reunirse e imprimirse juntas delante de ellas.

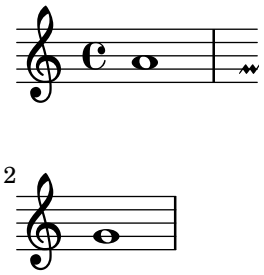
La sintaxis todavía utiliza el obsoleto estilo infijo `\[ expresión_musical \]`. Por motivos de consistencia, esto cambiará algún día al estilo postfijo `nota\[ ... nota\]`.

Custos

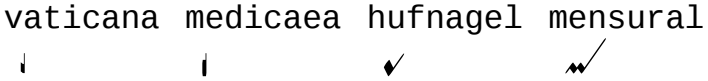
Un *custos* (en plural: *custodias*, que en latín significa "guarda") es un símbolo que aparece al final de una pauta. Anticipa la altura de la primera o primeras notas de la línea siguiente, ayudando al intérprete con los saltos de línea durante la ejecución.

Las custodias se utilizaron mucho en la notación musical hasta el siglo XVII. En nuestros días, sobreviven sólo en algunas formas particulares de notación musical como las ediciones contemporáneas de canto gregoriano como la *Editio Vaticana*. Hay distintos glifos para el custos que se utilizan en las diferentes variaciones de estilo notacional.

Para tipografiar las custodias, simplemente coloque un grabador `Custos_engraver` dentro del contexto `Staff` (el contexto del pentagrama) al declarar el bloque de `\layout` (el bloque de disposición de la página), y modifique el estilo del custos con una instrucción `\override` si lo desea, como se muestra en el ejemplo siguiente:



El glifo del custos se selecciona mediante la propiedad `style`. Los estilos contemplados son `vaticana`, `medicaea`, `hufnagel` y `mensural`. Se muestran en el siguiente fragmento:



Véase también

Glosario musical: Sección "custos" in *Glosario Musical*.

Fragmentos de código: Sección "Ancient notation" in *Fragmentos de código*.

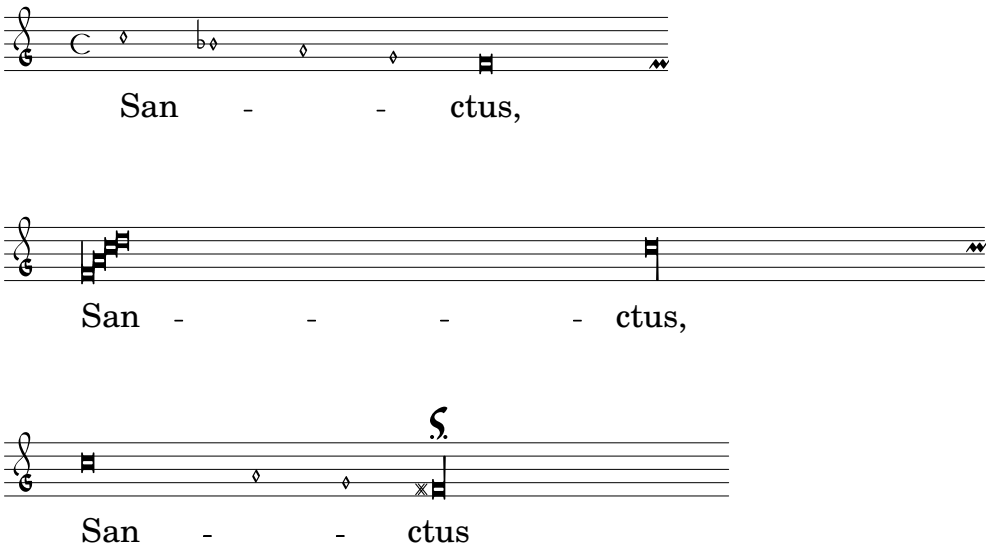
Referencia de funcionamiento interno: Sección "Custos" in *Referencia de Funcionamiento Interno*.

2.9.3 Tipografiar música mensural

Contextos de la música mensural

Los contextos predefinidos de voz `MensuralVoice` y de pauta `MensuralStaff` se pueden usar para grabar una pieza en estilo mensural. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de manera que pueda comenzar inmediatamente a escribir el canto, como lo demuestra el siguiente fragmento:

```
\score {
  <<
    \new MensuralVoice = "discantus" \relative c'' {
      \hide Score.BarNumber {
        c1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c\breve d\melismaEnd \]
        c\longa
        c\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



Véase también

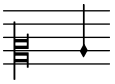
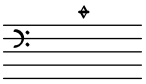
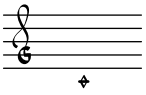
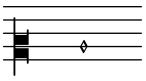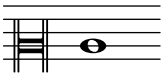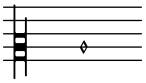
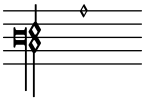
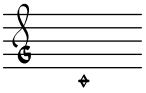
Glosario musical: Sección “mensural notation” in *Glosario Musical*.

Claves de la música mensural

La siguiente tabla presenta todas las claves antiguas que están contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian sólo en la línea en que se imprimen. En tales casos, se usa un número añadido al nombre para enumerar dichas claves, numerado desde la línea inferior hasta la superior. Puede forzar manualmente que se escriba un glifo de clave sobre una línea arbitraria, como se encuentra descrito en [Clave], página 16. La nota impresa a la derecha de cada clave en la columna de los ejemplos denota la situación del Do central (`c'`) con respecto a dicha clave.

Petrucci usaba claves de Do que tenían barras verticales en el lado izquierdo en un equilibrio variable, dependiendo de la línea de la pauta en que se imprimían.

Descripción	**Claves posibles**	**Ejemplo**

clave de Do, mensural	`mensural-c1`, `mensural-c2`, `mensural-c3`, `mensural-c4`, `mensural-c5`	
clave de Fa, mensural	`mensural-f`	
clave de Sol, mensural	`mensural-g`	
clave de Do, mensural negra	`blackmensural-c1`, `blackmensural-c2`, `blackmensural-c3`, `blackmensural-c4`, `blackmensural-c5`	
clave de Do, neomensural	`neomensural-c1`, `neomensural-c2`, `neomensural-c3`, `neomensural-c4`	
claves de Do en el estilo de Petrucci, para su uso en distintas líneas de la pauta (el ejemplo muestra la clave de Do en segunda)	`petrucci-c1`, `petrucci-c2`, `petrucci-c3`, `petrucci-c4`, `petrucci-c5`	
claves de Fa en el estilo de Petrucci, para su uso en distintas líneas de la pauta (el ejemplo muestra la lclave de Fa en tercera línea)	`petrucci-f3`, `petrucci-f4`, `petrucci-f5`	
clave de Sol en el estilo de Petrucci	`petrucci-g`	

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*, Sección “clef” in *Glosario Musical*.

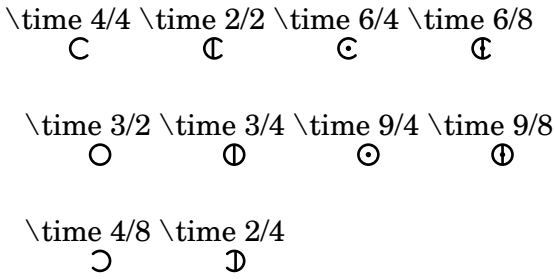
Referencia de la notación: [Clave], página 16.

Advertencias y problemas conocidos

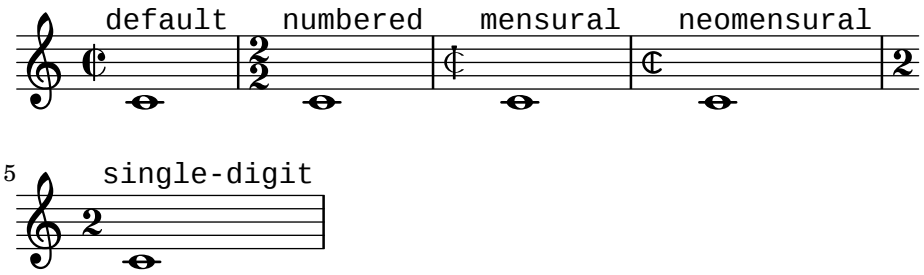
La clave de Sol mensural está asignada a la clave de Sol de Petrucci.

Indicaciones de compás de la música mensural

Los símbolos de mensuración (que son algo similar, pero no exactamente igual que las indicaciones de compás) están contemplados de forma limitada. Los glifos están configurados de forma fija para determinadas fracciones de tiempo. En otras palabras, para obtener un símbolo de mensuración determinado con la instrucción `\time n/m`, se tienen que escoger `n` y `m` según la siguiente tabla



Utilice la propiedad `style` del elemento gráfico `TimeSignature` para seleccionar las indicaciones de compás de la música antigua. Los estilos contemplados son `neomensural` y `mensural`. La tabla anterior utiliza el estilo `neomensural`. Los ejemplos siguientes muestran las diferencias que existen entre los estilos:



[Indicación de compás], página 62 ofrece una introducción general al uso de las indicaciones de compás.

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*.

Referencia de la notación: [Indicación de compás], página 62.

Advertencias y problemas conocidos

Las relaciones de duración de las notas no pueden cambiar con la indicación de compás, porque no son constantes. Por ejemplo, la razón de 1 breve = 3 semibreves (*tempus perfectum*) se puede hacer a mano mediante el establecimiento de

```
breveTP = #(ly:make-duration -1 0 3/2)
...
{ c\breveTP f1 }
```

Esto fija un valor para `breveTP` de 3/2 multiplicado por 2 = 3 veces una redonda.

Los símbolos `mensural68alt` y `neomensural68alt` (alternativos al símbolo de 6/8) no se puede seleccionar directamente con `\time`. En su lugar, utilice `\markup {\musicglyph #"timesig.mensural68alt" }`.

Cabezas de nota de la música mensural

Para la notación antigua, se puede elegir un estilo de cabezas de notas distinto al `default` (estilo predeterminado). Esto se consigue estableciendo la propiedad `style` del objeto `NoteHead` a los valores `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` o `semipetrucci`.

El estilo `baroque` se diferencia del estilo predeterminado `default` en que:

- Proporciona una cabeza de nota para la `maxima`, y
- Usa una forma cuadrada para las cabezas de `\breve`.

Los estilos `neomensural`, `mensural` y `petrucci` se diferencian del estilo `baroque` en que:

- Usan cabezas romboidales para las semibreves y todas las notas más cortas, y
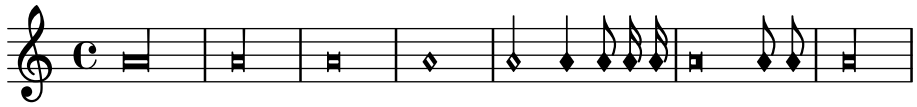
- Centran las plicas sobre las cabezas.

El estilo `blackpetrucci` produce notas con una cabeza que puede usarse en notación mensural negra o en secciones de coloratio dentro de la notación mensural blanca. A causa de que el estilo de la cabeza de las notas no influye en el número de corchetes, en este estilo la notación de la semimínima debe hacer como `a8*2`, no como `a4`, pues de lo contrario tendrá el aspecto de una mínima. El multiplicador puede ser distinto si se usa la coloratio, es decir, para la notación de tresillos.

Utilice el estilo `semipetrucci` para dibujar notas con cabeza semi-rellena (breves, longas y máximas).

El ejemplo siguiente muestra el estilo `petrucci`:

```
\set Score.skipBars = ##t
\autoBeamOff
\override NoteHead.style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead.style = #'semipetrucci
a'\breve*5/6
\override NoteHead.style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead.style = #'petrucci
a'\longa
```



Sección A.9 [Estilos de cabezas de nota], página 671 ofrece una visión de conjunto sobre la totalidad de los estilos de cabeza disponibles.

Véase también

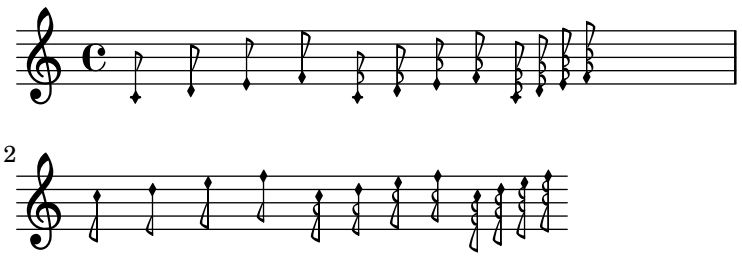
Glosario musical: Sección “mensural notation” in *Glosario Musical*, Sección “note head” in *Glosario Musical*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 671.

Corchetes de la música mensural

Utilice la propiedad `flag-style` del elemento gráfico `Stem` para seleccionar los corchetes de estilo antiguo. Aparte del estilo de corchete por defecto `default`, sólo está contemplado el estilo `mensural`

```
\override Flag.style = #'mensural
\override Stem.thickness = #1.0
\override NoteHead.style = #'mensural
\autoBeamOff
c8 d e f c16 d e f c32 d e f s8
c'8 d e f c16 d e f c32 d e f
```



Observe que la bandera más interna de cada uno de los corchetes mensurales se alinea verticalmente con una línea de la pauta.

No existe un estilo particular de corchete para la notación neo-mensural ni Petrucci. No existen corchetes en la notación del canto gregoriano.

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*, Sección “flag” in *Glosario Musical*.

Advertencias y problemas conocidos

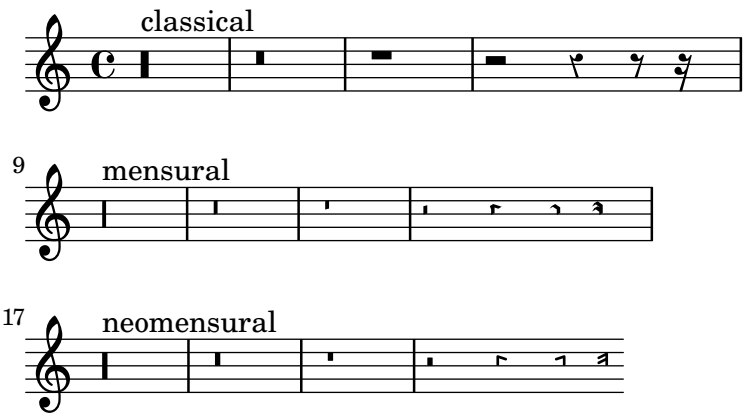
La alineación vertical de cada uno de los corchetes con una línea de la pauta da por supuesto que las plicas siempre terminan exactamente encima o bien exactamente en el medio de dos líneas de la pauta. Esto puede no ser siempre cierto cuando se utilizan las posibilidades de disposición avanzadas de la notación clásica (que, de todas formas, por lo común se encuentra fuera del ámbito de la notación mensural).

Silencios de la música mensural

Utilice la propiedad `style` del elemento gráfico `Rest` para seleccionar silencios antiguos. Los estilos contemplados son `classical`, `neomensural` y `mensural`. `classical` se diferencia del estilo por defecto `default` solamente en que el silencio de negra parece un silencio de corchea invertido horizontalmente. Los estilos `mensural` y `neomensural` reproducen el aspecto de los silencios de los manuscritos y ediciones impresas de hasta el s.XVI.

El ejemplo siguiente muestra los estilos `mensural` y `neomensural`:

```
\set Score.skipBars = ##t
\override Rest.style = #'classical
r\longa^"classical" r\breve r1 r2 r4 r8 r16 s \break
\override Rest.style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest.style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16
```



No existen silencios de fusa y semifusa específicos para los estilos mensural ni neo-mensural. En su lugar se tomarán los silencios del estilo predeterminado.

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*.

Referencia de la notación: [Silencios], página 54.

Fragmentos de código: Sección “Ancient notation” in *Fragmentos de código*.

Advertencias y problemas conocidos

El glifo para el silencio de máxima en el estilo mensural es en realidad un silencio de longa perfecta; use dos (o tres) silencios de longa para imprimir un silencio de máxima. Los sliencions de longa no se agrupan automáticamente, por lo que debe hacerse manualmente utilizando silencios con altura determinada.

Alteraciones y armaduras de la música mensural

El estilo `mensural` ofrece unos símbolos de sostenido y bemol distintos de los del estilo predeterminado. Si se solicita, el símbolo de becuadro se extrae del estilo `vaticana`.

mensural



El estilo de las alteraciones y de la armadura de la tonalidad se controlan a través de la propiedad `glyph-name-alist` de los elementos gráficos `Accidental` y `KeySignature`, respectivamente; p.ej.:

```
\override Staff.Accidental.glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

Véase también

Glosario musical: Sección “mensural notation” in *Glosario Musical*, Sección “Pitch names” in *Glosario Musical*, Sección “accidental” in *Glosario Musical*, Sección “key signature” in *Glosario Musical*.

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 5, [Alteraciones accidentales automáticas], página 26, [Armadura de la tonalidad], página 20.
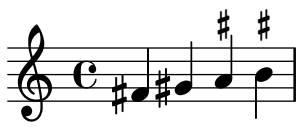
Referencia de funcionamiento interno: Sección “KeySignature” in *Referencia de Funcionamiento Interno*.

Alteraciones de anotación (*musica ficta*)

En la música europea de hasta aprox. 1600, se esperaba de los cantantes que alterasen cromáticamente ciertas notas según su propia iniciativa y de acuerdo a ciertas reglas. Esto recibe el nombre de *musica ficta*. En las transcripciones modernas, estas alteraciones accidentales se imprimen normalmente encima de la nota.
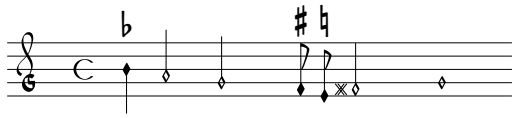
Están contempladas estas alteraciones sugeridas, y se pueden activar estableciendo `suggestAccidentals` al valor verdadero.

```
fis gis
\set suggestAccidentals = ##t
ais bis
```



Esto hará que se traten *todas* las alteraciones siguientes como *musica ficta* hasta que se desactive con `\set suggestAccidentals = ##f`. Una forma más práctica es usar `\once \set suggestAccidentals = ##t`, que puede incluso definirse como una forma breve muy conveniente:

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative c''
  \new MensuralVoice  {
        \once \set suggestAccidentals = ##t
  bes4 a2 g2 \ficta fis8 \ficta e! fis2 g1
  }
}
```



Véase también

Referencia de funcionamiento interno: Sección “Accidental_engraver” in *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” in *Referencia de Funcionamiento Interno*.

Ligaduras mensurales blancas

Las ligaduras mensurales blancas están contempladas de forma limitada.

Para grabar ligaduras mensurales blancas, en el bloque layout o de diseño de página reemplace el grabador `Ligature_bracket_engraver` por el grabador `Mensural_ligature_engraver` en el contexto `Voice`:

```
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Mensural_ligature_engraver"
  }
}
```

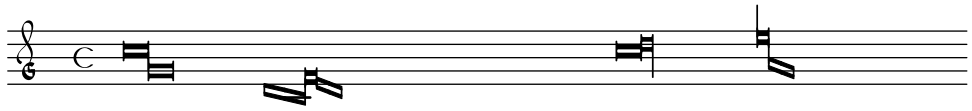
No existe ningún lenguaje de entrada adicional para describir la forma de una ligadura mensural blanca. Más bien la forma se determina exclusivamente a partir de la altura y duración de las notas comprendidas. Si bien este enfoque podría hacer que un usuario con poca experiencia tardase un tiempo en acostumbrarse a él, tiene la gran ventaja de que toda la información musical de la ligadura se conoce internamente. Esto no se requiere solamente para una salida MIDI correcta, sino que también permite la transcripción automática de las ligaduras.

En algunos lugares, dos notas consecutivas se pueden representar bien como dos cuadrados, bien como un paralelogramo oblicuo (forma de flexa). En tales casos la forma por omisión es dos cuadrados, pero puede requerirse una flexa fijando la propiedad `ligature-flexa` de la *segunda* cabeza. La longitud de una flexa se puede fijar mediante la propiedad `flexa-width` de la cabeza.
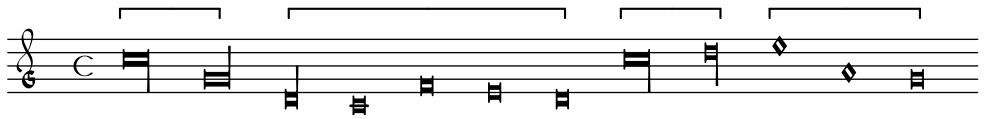
Por ejemplo:

```
\score {
  \relative c' {
    \set Score.timing = ##f
    \set Score.defaultBarType = "-"
    \override NoteHead.style = #'petrucci
    \override Staff.TimeSignature.style = #'mensural
    \clef "petrucci-g"
    \[ c'\maxima g \]
    \[ d\longa
       \override NoteHead.ligature-flexa = ##t
```

```
      \once \override NoteHead.flexa-width = #3.2
      c\breve f e d \]
    \[ c'\maxima d\longa \]
    \[ e1 a, g\breve \]
  }
  \layout {
    \context {
      \Voice
      \remove "Ligature_bracket_engraver"
      \consists "Mensural_ligature_engraver"
    }
  }
}
```



Si no se sustituye el grabador `Ligature_bracket_engraver` por el `Mensural_ligature_engraver`, la misma música se transcribe de la siguiente manera:



Véase también

Glosario musical: Sección “ligature” in *Glosario Musical*.

Referencia de la notación: [Ligaduras de neumas cuadrados gregorianos], página 437, [Ligaduras], página 423.

Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras es pobre. Las alteraciones accidentales pueden colisionar con las notas previas.

2.9.4 Tipografiado del canto gregoriano

Al componer tipográficamente una pieza en notación de canto gregoriano, el grabador `Vaticana_ligature_engraver` selecciona automáticamente la cabeza adecuada de las figuras, de manera que no es necesario establecer explícitamente el estilo de las cabezas. A pesar de ello se puede establecer el estilo de las cabezas, p.ej.: a `vaticana_punctum` para producir neumas de punctum. De forma similar, el grabador `Mensural_ligature_engraver` construye ligaduras mensurales automáticamente. Consulte [Ligaduras], página 423 para ver cómo funcionan los grabadores de ligaduras.

Véase también

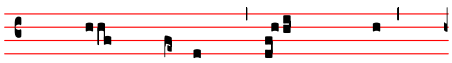
Glosario musical: Sección “ligature” in *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 431, [Ligaduras], página 423.

Contextos del canto gregoriano

El contexto predefinido de voz `VaticanaVoiceContext` y el de pauta `VaticanaStaffContext` se pueden utilizar para grabar una pieza de canto gregoriano en el estilo de la Editio Vaticana. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de forma que pueda inmediatamente comenzar a escribir el canto en sí, como lo demuestra el siguiente fragmento:

```
\include "gregorian.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}
```



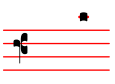
San- ctus, San- ctus,

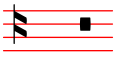
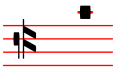
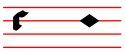


San- ctus

Claves de canto gregoriano

La tabla siguiente muestra todas las claves gregorianas contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian solamente en la línea en que se imprimen. En estos casos, para enumerar estas claves se usa un número a continuación del nombre de la clave, contando desde la línea inferior. Aún se puede forzar manualmente la composición tipográfica de un glifo de clave sobre una línea arbitraria, como se describe en [Clave], página 16. La nota que se imprime a la derecha junto a cada una de las claves en la columna de los ejemplos, denota el Do central (`c'`) respecto a dicha clave.

Descripción	**Claves posibles**	**Ejemplo**
Clave de Do, estilo Editio Vaticana	`vaticana-do1`, `vaticana-do2`, `vaticana-do3`	
Clave de Fa, estilo Editio Vaticana	`vaticana-fa1`, `vaticana-fa2`	

Clave de Do, estilo Editio Medicaea	`medicaea-do1`, `medicaea-do2`, `medicaea-do3`	
Clave de Fa, estilo Editio Medicaea	`medicaea-fa1`, `medicaea-fa2`	
Clave de Do, estilo hufnagel	`hufnagel-do1`, `hufnagel-do2`, `hufnagel-do3`	
Clave de Fa, estilo hufnagel	`hufnagel-fa1`, `hufnagel-fa2`	
Clave combinada de Do y Fa, estilo hufnagel	`hufnagel-do-fa`	

Véase también

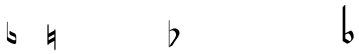
Glosario musical: Sección “clef” in *Glosario Musical*.

Referencia de la notación: [Clave], página 16.

Alteraciones y armaduras de canto gregoriano

Están disponibles las alteraciones de los tres estilos gregorianos:

vaticana medicaea hufnagel



Como se ve en el ejemplo, cada estilo no contempla todas las alteraciones. Al intentar acceder a una alteración que no está contemplada, LilyPond cambia a un estilo diferente.

El estilo de las alteraciones y las armaduras de tonalidad se controla a través de la propiedad `glyph-name-alist` de los grobs (objetos gráficos) Sección “Accidental” in *Referencia de Funcionamiento Interno* y Sección “KeySignature” in *Referencia de Funcionamiento Interno*, respectivamente; p.ej.:

```
\override Staff.Accidental.glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

Véase también

Glosario musical: Sección “accidental” in *Glosario Musical*, Sección “key signature” in *Glosario Musical*.

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 5, [Alteraciones accidentales automáticas], página 26, [Armadura de la tonalidad], página 20.

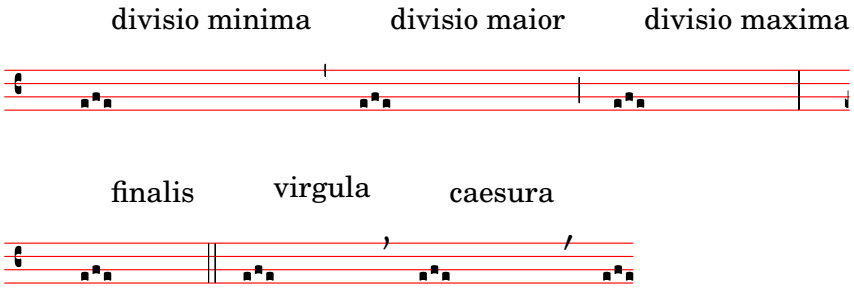
Referencia de funcionamiento interno: Sección “KeySignature” in *Referencia de Funcionamiento Interno*.

Divisiones

En la notación del canto gregoriano no existen silencios, sino [Divisiones], página 435.

Una *divisio* (en plural: *divisiones*, que en latín significa 'división') es un símbolo del contexto del pentagrama que se usa para estructurar la música gregoriana en frases y secciones. El significado musical de *divisio minima*, *divisio maior* y *divisio maxima* se podrían caracterizar como una pausa corta, media y larga (respectivamente), algo así como las marcas de respiración de [Marcas de respiración], página 131. El signo de *finalis* no sólo marca el final de un canto, sino que se usa también con frecuencia dentro de un único canto antifonal o responsorial para marcar el final de cada una de las secciones.

Para usar divisiones, incluya el archivo '`gregorian.ly`'. Contiene definiciones que podrá aplicar con tan sólo escribir `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` y `\finalis` en los lugares adecuados de la entrada. Algunas ediciones usan *virgula* o *caesura* en vez de divisio minima. Por ello, '`gregorian.ly`' define también las instrucciones `\virgula` y `\caesura`.



Instrucciones predefinidas

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Véase también

Glosario musical: Sección "caesura" in *Glosario Musical*, Sección "divisio" in *Glosario Musical*.

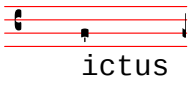
Referencia de la notación: [Marcas de respiración], página 131.

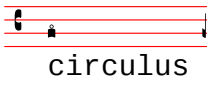
Archivos de inicio: '`ly/gregorian.ly`'.

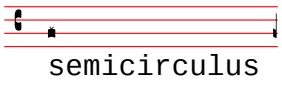
Articulaciones del canto gregoriano

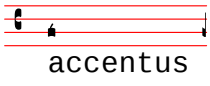
Además de los signos de articulación estándar que se hallan descritos en la sección [Articulaciones y ornamentos], página 116, se proveen signos de articulación diseñados específicamente para utilizarlos con la notación en el estilo de la *Editio Vaticana*.

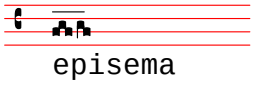
```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \override TextScript.font-family = #'typewriter
    \override TextScript.font-shape = #'upright
    \override Script.padding = #-0.1
    a\ictus_"ictus " \bar "" \break
    a\circulus_"circulus " \bar "" \break
    a\semicirculus_"semicirculus " \bar "" \break
    a\accentus_"accentus " \bar "" \break
    \[ a_"episema" \episemInitium \pes b \flexa a b \episemFinis \flexa a \]
  }
}
```











Véase también

Referencia de la notación: [Articulaciones y ornamentos], página 116.

Fragmentos de código: Sección “Ancient notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Episema” in *Referencia de Funcionamiento Interno*, Sección “EpisemaEvent” in *Referencia de Funcionamiento Interno*, Sección “Episema_engraver” in *Referencia de Funcionamiento Interno*, Sección “Script” in *Referencia de Funcionamiento Interno*, Sección “ScriptEvent” in *Referencia de Funcionamiento Interno*, Sección “Script_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Ciertas articulaciones se colocan demasiado cerca verticalmente de las cabezas de nota correspondientes.

Puntos de aumentación (*morae*)

Los puntos de aumentación, llamados también *morae*, se escriben con la función musical `\augmentum`. Observe que `\augmentum` está programado como una función musical unaria y no como un prefijo de cabeza. Se aplica solamente a la expresión musical que sigue inmediatamente. Esto es, `\augmentum \virga c` no tendrá ningún efecto visible. En su lugar, escriba `\virga \augmentum c` o `\augmentum {\virga c}`. Tenga en cuenta también que puede escribir `\augmentum {a g}` como abreviatura de `\augmentum a \augmentum g`.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



Véase también

Referencia de la notación: [Marcas de respiración], página 131.

Referencia de funcionamiento interno: Sección “BreathingSign” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Ancient notation” in *Fragmentos de código*.

Ligaduras de neumas cuadrados gregorianos

La notación de neumas cuadrados gregorianos está contemplada de forma limitada (siguiendo el estilo de la Editio Vaticana). El núcleo principal del conjunto de las ligaduras ya se pueden tipografiar, pero aún faltan aspectos esenciales para una composición tipográfica seria, como (entre otros) la alineación horizontal de varias ligaduras, la alineación de la letra y un correcto manejo de las alteraciones accidentales.

Se habilita el soporte de los neumas gregorianos por medio de la inclusión con `\include` del archivo '`gregorian.ly`' al principio del archivo. Esto hace que estén disponibles una cierta cantidad de instrucciones adicionales para producir los símbolos de los neumas que se usan en la notación de canto llano.

Las cabezas de las notas se pueden *modificar* y/o *unir*.

- La forma de la cabeza se puede modificar *precediendo* el nombre de la nota con una cualquiera de las instrucciones siguientes: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.
- Las ligaduras hablando en propiedad, (es decir, notas unidas), se producen escribiendo una de las instrucciones de unión `\pes` o `\flexa`, para el movimiento ascendente y descendente, respectivamente, *intercaladas* entre las notas que se han de unir.

Una nota sin modificaciones produce un *punctum*. Todos los demás neumas, incluso los neumas de una nota con forma distinta como la *virga*, se consideran en principio como ligaduras y por ello se deben escribir dentro de `\[...\]`.

Neumas de una nota:

- El *punctum* es la forma básica de nota (en el estilo *Vaticana*: un cuadrado con una ligera curvatura a modo de excelencia tipográfica). Además del *punctum* normal, están el *punctum inclinatum*, oblicuo, producido con el prefijo `\inclinatum`. El *punctum* normal se puede modificar con `\cavum`, que produce una nota vacía, y `\linea`, que traza líneas verticales a ambos lados de la nota.
- La *virga* tiene una plica descendente en el lado derecho. Se produce mediante el modificador `\virga`.

Ligaduras

A diferencia de casi todos los otros sistemas de notación de neumas, el aspecto tipográfico de las ligaduras no viene dictado directamente por las instrucciones de la entrada, sino que sigue unas convenciones que dependen del significado musical. Por ejemplo, una ligadura de tres notas con la forma musical bajo-alto-bajo, como `\[ a \pes b \flexa g \]`, produce un Torculus que consiste en tres cabezas de Punctum, mientras que la forma alto-bajo-alto, como `\[ a \flexa g \pes b \]`, produce un Porrectus con una forma de flexa curvada y una sola cabeza de Punctum. No existe ninguna instrucción para tipografiar explícitamente la forma de flexa curvada; la decisión de cuándo tipografiar una forma de flexa curvada está basada en la entrada musical. La idea de este enfoque es separar los aspectos musicales de la entrada, del estilo de notación de la salida. De esta forma, la misma entrada se puede reutilizar para tipografiar la misma música en un estilo diferente de notación de canto gregoriano.

Neumas licuescentes

Otra categoría fundamental de notas en el canto gregoriano es la de los llamados neumas licuescentes. Se utilizan bajo ciertas circunstancias al final de una sílaba que acaba en una letra 'licuescente', es decir, consonantes sonoras que pueden tener una altura tonal, (nasales, l, r, v, j, y sus diptongos equivalentes). Así, los neumas licuescentes nunca se utilizan aisladamente (aunque se pueden producir algunos de ellos), y siempre están al final de una ligadura.

Los neumas licuescentes se representan gráficamente de dos formas distintas más o menos intercambiables: con una nota pequeña o 'girando' la nota principal hacia arriba o hacia abajo.

La primera forma se produce haciendo un `pes` o una `flexa` normales y modificando la forma de la segunda nota: `\[ a \pes \deminutum b \]` , mientras que la segunda se hace modificando la forma de un neuma de una nota con `\auctum` y uno de los generadores de dirección `\descendens` o `\ascendens`, p.ej. `\[ \auctum \descendens a \]` .

Símbolos especiales

Hay una tercera categoría de símbolos que se hace a partir de un pequeño número de símbolos que tienen un significado especial (que, por cierto, casi siempre se conocen sólo vagamente): el *quilisma*, el *oriscus* y el *strophicus*. Todos ellos se producen anteponiendo al nombre de la nota el modificador correspondiente, `\quilisma`, `\oriscus` o `\stropha`.

Dentro de los delimitadores de ligadura `\[` y `\]`, se pueden acumular prácticamente cualquier cantidad de cabezas de nota para formar una sola ligadura, y los prefijos de cabeza como `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc. se pueden mezclar libremente. El uso del conjunto de reglas que subyace a la construcción de ligaduras en la tabla anterior está consecuentemente extrapolada. De esta manera se pueden crear un número infinito de ligaduras distintas.

Observe que el uso de esos símbolos en la propia música sigue ciertas reglas que LilyPond no comprueba. P.ej., el *quilisma* siempre es la nota intermedia de una ligadura ascendente, y suele caer sobre un intervalo de semitono, pero es perfectamente posible, si bien incorrecto, hacer un quilisma de una nota.

Además de los símbolos de nota, el archivo '`gregorian.ly`' define también las instrucciones `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` y `\IIJ`, que producen los caracteres correspondientes, p.ej. para utilizarlos en la letra, como marcas de sección, etc. Estas instrucciones utilizan caracteres de Unicode especiales y sólo funcionan si se usa una fuente tipográfica que los contemple.

La tabla siguiente muestra un conjunto limitado, pero representativo, de ligaduras gregorianas, junto a los fragmentos de código que las producen. La tabla está basada en la tabla de neumas extendidos del segundo volumen del Antiphonale Romanum (*Liber Hymnarius*), publicado en 1983 por los monjes de Solesmes. La primera columna da el nombre de la ligadura, con la forma principal en tipo negrita y las formas licuescentes en cursiva. La tercera forma muestra el fragmento de código que produce dicha ligadura, utilizando Sol, La y Si como alturas de ejemplo.

Neumas de una nota

Formas **Básica** y *Licuescente*	**Salida**	**Código de LilyPond**
Punctum		`\[ b \]`
		`\[ \cavum b \]`
		`\[ \linea b \]`

Punctum Auctum Ascendens

```
\[ \auctum \ascendens b \]
```



Punctum Auctum Descendens

```
\[ \auctum \descendens b \]
```



Punctum inclinatum

```
\[ \inclinatum b \]
```



Punctum Inclinatum Auctum

```
\[ \inclinatum \auctum b \]
```



Punctum Inclinatum Parvum

```
\[ \inclinatum \deminutum b \]
```


Virga



Ligaduras de dos notas

Clivis vel Flexa

```
\[ b \flexa g \]
```



Clivis Aucta Descendens

```
\[ b \flexa \auctum \descendens
g \]
```



Clivis Aucta Ascendens

```
\[ b \flexa \auctum \ascendens
g \]
```



Cephalicus

```
\[ b \flexa \deminutum g \]
```



Podatus o Pes

```
\[ g \pes b \]
```



Pes Auctus Descendens

```
\[ g \pes \auctum \descendens b
\]
```



Pes Auctus Ascendens

```
\[ g \pes \auctum \ascendens b
\]
```



Epiphonus

```
\[ g \pes \deminutum b \]
```



Pes Initio Debilis

```
\[ \deminutum g \pes b \]
```



Pes Auctus Descendens Initio Debilis

```
\[ \deminutum g \pes \auctum
\descendens b \]
```



Ligaduras de varias notas

Torculus

```
\[ a \pes b \flexa g \]
```



Torculus Auctus Descendens

```
\[ a \pes b \flexa \auctum
\descendens g \]
```



Torculus Deminutus

```
\[ a \pes b \flexa \deminutum g
\]
```



Torculus Initio Debilis

```
\[ \deminutum a \pes b \flexa g
\]
```



Torculus Auctus Descendens Initio Debilis

```
\[ \deminutum a \pes b \flexa
\auctum \descendens g \]
```



Torculus Deminutus Initio Debilis

```
\[ \deminutum a \pes b \flexa
\deminutum g \]
```



Porrectus

```
\[ a \flexa g \pes b \]
```



Porrectus Auctus Descendens

```
\[ a \flexa g \pes \auctum
\descendens b \]
```



Porrectus Deminutus

```
\[ a \flexa g \pes \deminutum b
\]
```



Climacus

```
\[ \virga b \inclinatum a
\inclinatum g \]
```



Climacus Auctus

```
\[ \virga b \inclinatum a
\inclinatum \auctum g \]
```



Climacus Deminutus

```
\[ \virga b \inclinatum a
\inclinatum \deminutum g \]
```



Scandicus

```
\[ g \pes a \virga b \]
```



Scandicus Auctus Descendens

```
\[ g \pes a \pes \auctum
\descendens b \]
```



Scandicus Deminutus

```
\[ g \pes a \pes \deminutum b \]
```



Símbolos especiales

Quilisma

```
\[ g \pes \quilisma a \pes b \]
```



Quilisma Pes Auctus Descendens

```
\[ \quilisma g \pes \auctum
\descendens b \]
```



Oriscus

```
\[ \oriscus b \]
```


Pes Quassus

```
\[ \oriscus g \pes \virga b \]
```



Pes Quassus Auctus Descendens

```
\[ \oriscus g \pes \auctum
\descendens b \]
```



Salicus

```
\[ g \oriscus a \pes \virga b \]
```



Salicus Auctus Descendens

```
\[ g \oriscus a \pes \auctum
\descendens b \]
```



(Apo)stropha

```
\[ \stropha b \]
```



Stropha Aucta

```
\[ \stropha \auctum b \]
```



Bistropha

```
\[ \stropha b \stropha b \]
```



Tristropha

```
\[ \stropha b \stropha b
\stropha b \]
```



Trigonus

```
\[ \stropha b \stropha b
\stropha a \]
```



Instrucciones predefinidas

Están contemplados los siguientes prefijos de cabeza: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Se pueden acumular los prefijos de cabeza, aunque con ciertas restricciones. Por ejemplo, se pueden aplicar `\descendens` o `\ascendens` a una cabeza, pero no los dos a la misma cabeza.

Se pueden ligar dos cabezas adyacentes con las instrucciones infijas `\pes` y `\flexa` para una línea melódica ascendente y descendente, respectivamente.

Utilice la función musical unaria `\augmentum` para añadir puntos de augmentum.

Véase también

Glosario musical: Sección “ligature” in *Glosario Musical*.

Referencia de la notación: [Ligaduras de neumas cuadrados gregorianos], página 437, [Ligaduras mensurales blancas], página 431, [Ligaduras], página 423.

Advertencias y problemas conocidos

Cuando aparece un punto de `\augmentum` al final de la última pauta dentro de una ligadura, a veces se posiciona defectuosamente en el sentido vertical. Para rodear el problema, escriba una nota adicional de salto (p.ej. `s8`) como última nota de la pauta.
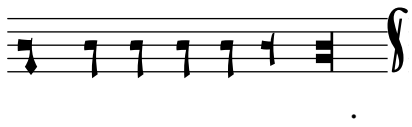
`\augmentum` debería estar implementado como un prefijo de cabeza en vez de una función musical unaria, de forma que `\augmentum` se pudiera entremezclar con los prefijos de cabeza en un orden arbitrario.

2.9.5 Tipografiado del canto kievano en notación cuadrada

Contextos del canto kievano

De la misma manera que con las notaciones mensural y gregoriana, pueden usarse los contextos predefinidos `KievanVoice` y `KievanStaff` para tipografiar una pieza en notación cuadrada. Estos contextos inician todas las propiedades de contexto necesarias a sus valores adecuados, de manera que pueda ponerse manos a la obra inmediatamente con la introducción del propio canto:

```
\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      \cadenzaOn
        c4 c c c c2 b\longa
        \bar "k"
    }
    \new Lyrics \lyricsto "melody" {
      Го -- спо -- ди по -- ми -- луй.
    }
  >>
}
```



Véase también

Glosario musical: Sección “kievan notation” in *Glosario Musical*.
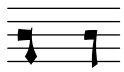
Advertencias y problemas conocidos

LilyPond contempla la notación del canto kievano en el estilo sinodal, tal y como se usa en los cantorales impresos por el Sacro Sínodo Ruso en la década de 1910 y recientemenre reimpreso por la Casa de Publicaciones del Patriarcado de Moscú. LilyPond no contempla las formas anteriores (menos corrientes) de notación kievana que se usaban en Galitzia para la notación del canto llano ruteno.

Claves del canto kievano

En la notación kievana solamente se utiliza una clave (la clave Tse-fa-ut). Se utiliza para indicar la posición del Do:

```
\clef "kievan-do"
\kievanOn
c
```



Véase también

Glosario musical: Sección “kievan notation” in *Glosario Musical*, Sección “clef” in *Glosario Musical*.
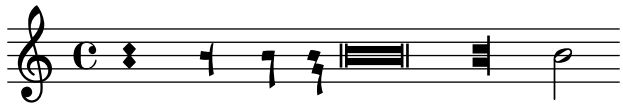
Referencia de la notación: [Clave], página 16.

Notas del canto kievano

Para la notación cuadrada del canto kievano, debe escogerse el estilo apropiado para la forma de la cabeza de las notas y se deben suprimir los corchetes y las plicas. Esto se consigue mediante una llamada a la función `\kievanOn`, que establece las propiedades adecuadas para la cabeza, plica y corchete de las notas. Una vez que las notas de estilo kievano ya no se necesiten, pueden revertirse estas propiedades llamando a la función `\kievanOff`.

La nota final del canto kievano, que suele ir al final de una pieza musical, puede seleccionarse estableciendo la duración al valor `\longa`. La marca kievana de recitativo, utilizada para indicar la entonación de varias sílabas sobre una sola nota, se puede seleccionar estableciendo la duración al valor `\breve`. El siguiente ejemplo muestra las diversas formas de la cabeza de las notas del canto kievano:

```
\autoBeamOff
\cadenzaOn
\kievanOn
b'1 b'2 b'4 b'8 b'\breve b'\longa
\kievanOff
b'2
```



Véase también

Glosario musical: Sección “kievan notation” in *Glosario Musical*, Sección “note head” in *Glosario Musical*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 671.

Advertencias y problemas conocidos
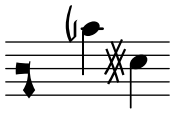
LilyPond determina de forma automática si debe trazarse la forma de una nota con la plica hacia arriba o hacia abajo. Sin embargo, cuando se está tipografiando un canto en notación cuadrada, es preceptivo que las plicas estén orientadas en la misma dirección dentro de un melisma. Esto puede hacerse manualmente fijando la propiedad `direction` del objeto `Stem`.

Alteraciones accidentales del canto kievano

El estilo `kievan` para las alteraciones accidentales se selecciona con la propiedad `glyph-name-alist` del grob `Accidental`. El estilo `kievan` provee unos signos de sostenido y de bemol diferentes del estilo predeterminado. No esiste el becuadro en la notación kievana. El símbolo del sostenido no se utiliza en la música sinodal pero puede aparecer en manuscritos más antiguos. Se ha incluido principalmente a efectos de compatibilidad.

```
\clef "kievan-do"
\override Accidental.glyph-name-alist =
 #alteration-kievan-glyph-name-alist
```

```
bes' dis,
```



Véase también
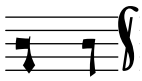
Glosario musical: Sección “kievan notation” in *Glosario Musical*, Sección “accidental” in *Glosario Musical*.

Referencia de la notación: [Alteraciones accidentales], página 5, [Alteraciones accidentales automáticas], página 26, Sección A.8 [La tipografía Feta], página 650.

Líneas divisorias del canto kievano

Normalmente se coloca una figura decorativa al final de una pieza de notación kievana, que puede denominarse como «doble barra final del canto kievano». Se puede invocar como `\bar "k"`.

```
\kievanOn
\clef "kievan-do"
c \bar "k"
```



Véase también

[Barras de compás], página 94, Sección A.8 [La tipografía Feta], página 650.

Melismas del canto kievano

Las notas que están dentro de un melisma del canto kievano se suelen colocar a poca distancia entre sí y cada melisma está separado de los demás por un espacio vacío. Esto se hace para permitir al cantor la identificación rápida de las estructuras melódicas del canto Znamenny. En LilyPond, los melismas se tratan como ligaduras y el especiado está implementado por el grabador `Kievan_ligature_engraver`.
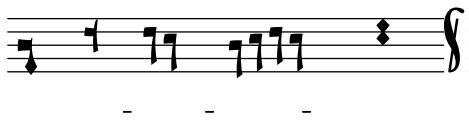
Cuando se usan los contextos `KievanVoice` y `KievanStaff`, se habilita el grabador `Kievan_ligature_engraver` de forma predeterminada. En otros contextos, se puede invocar sustituyendo el grabador `Ligature_bracket_engraver` por `Kievan_ligature_engraver` en el bloque layout:

```
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Kievan_ligature_engraver"
  }
}
```

El espaciado entre las notas que están dentro de una misma ligadura kievana puede controlatse estableciendo la propiedad `padding` de `KievanLigature`.

El ejemplo siguiente muestra el uso de las ligaduras del canto kievano:

```
\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      \cadenzaOn
        e2 \[ e4( d4 ) \] \[ c4( d e  d ) \] e1 \bar "k"
    }
    \new Lyrics \lyricsto "melody" {
      Га -- ври -- и -- лу
    }
  >>
}
```



Véase también

Glosario musical: Sección “ligature” in *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 431, [Ligaduras de neumas cuadrados gregorianos], página 437, [Ligaduras], página 423.

Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras es mediocre.

2.9.6 Trabajar con música antigua - escenarios y soluciones

El trabajo con música antigua implica con frecuencia tareas adicionales que difieren considerablemente de la notación moderna para la que se diseñó LilyPond. En el resto de esta sección se perfilan unos cuantos escenarios típicos, con sugerencias de soluciones. Entre ellos están los siguientes:

- cómo hacer incipits (o sea, material a modo de prefacio que indica cómo era el original) a las transcripciones modernas de música mensural;
- cómo conseguir el diseño *Mensurstriche* que se usa con frecuencia para las transcripciones modernas de música polifónica;
- cómo transcribir canto gregoriano en notación moderna;
- cómo generar notación antigua y moderna a partir de la misma fuente.

Incipits

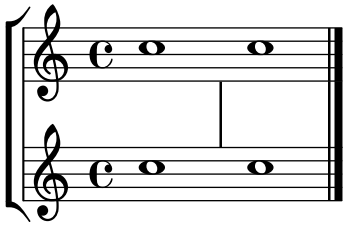
Continuará...

Disposición tipo «Mensurstriche»

Mensurstriche (‘líneas de mensuración’) es el término aceptado para las barras de compás que se trazan entre los pentagramas de un sistema pero no dentro de los propios pentagramas. Es una forma común de preservar el aspecto visual de las duraciones del original, es decir, sin tener que dividir las notas sincopadas en las barras de compás, y al mismo tiempo ofrecer la ayuda orientativa que proporcionan las barras de compás.

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra estableciendo la propiedad `transparent`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Transcripción de canto gregoriano

El canto gregoriano se puede transcribir a notación moderna con ciertos trucos sencillos.

Plicas. Todas las plicas se pueden eliminar de una vez retirando el grabador `Stem_engraver` del contexto de voz:

```
\layout {
  ...
  \context {
    \Voice
      \remove "Stem_engraver"
  }
}
```

Sin embargo, en ciertos estilos de transcripción, se usan plicas de forma ocasional, por ejemplo para indicar la transición a partir de un recitativo monotónico a un gesto melódico fijo. En estos casos podemos usar bien `\hide Stem` o bien `\override Stem.length = #0`, y restaurar la plica cuando se necesite con el correspondiente `\once \override Stem.transparent = ##f` (véase el ejemplo de abajo).

Compás. Para el canto no medido, existen diversas alternativas.

Se puede eliminar el grabador Time_signature_engraver del contexto de Staff sin ningún efecto secundario negativo. La alternativa, que es hacer transparente la indicación de compás, deja un espacio en blanco en la partitura, pues la indicación invisible sigue ocupando un espacio.
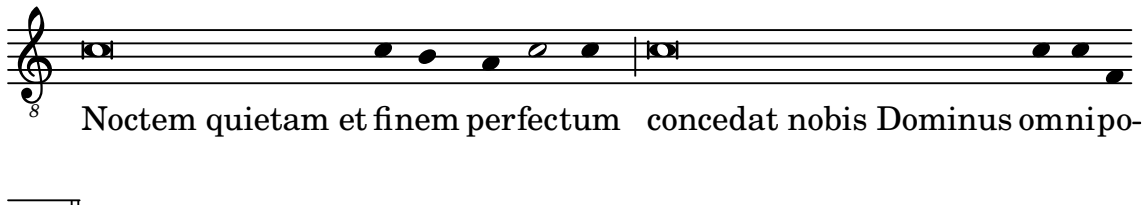
En muchos casos da un buen resultado `\set Score.timing = ##f`. Otra alternativa es utilizar `\cadenzaOn` y `\cadenzaOff`.

Para quitar las barras de compás, el enfoque radical consiste en retirar (mediante la instrucción `\remove`) el grabador Bar_engraver del contexto de pentagrama Staff. Una vez más, en vez de ello podemos usar `\hide BarLine` si se necesita ocasionalmente una barra de compás.

Un tipo de transcripción muy común es el canto de recitativo, en el que la nota repetida se indica con una breve. Se puede tratar el texto del tono de recitativo de dos maneras. Podemos ponerlo como una sola sílaba alineada por la izquierda:

```
\include "gregorian.ly"
chant = \relative c' {
  \clef "G_8"
  c\breve c4 b4 a c2 c4  \divisioMaior
  c\breve c4 c f, f \finalis
}

verba = \lyricmode {
  \once \override LyricText.self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText.self-alignment-X = #-1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}
\score {
  \new Staff <<
  \new Voice = "melody" \chant
  \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
  }
}
```

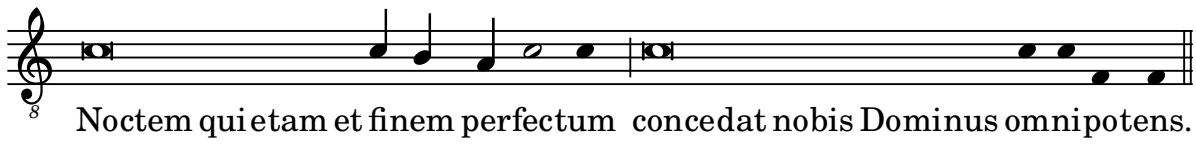


Esto funciona bien en la medida en que el texto no comprenda un salto de línea. Si es el caso, una alternativa sería añadir notas ocultas a la partitura, en combinación con cambios en la visibilidad de las plicas:

```
\include "gregorian.ly"
chant = \relative c' {
  \clef "G_8"
  \set Score.timing = ##f
  c\breve \hide NoteHead  c c c c c
  \undo \hide NoteHead
  \override Stem.transparent = ##f \stemUp c4 b4 a
  \hide Stem c2 c4  \divisioMaior
  c\breve \hide NoteHead c c c c c c c
  \undo \hide NoteHead c4 c f, f \finalis
}
```

```
verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics \lyricsto "melody" \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \hide BarLine
      \hide Stem
    }
  }
}
```
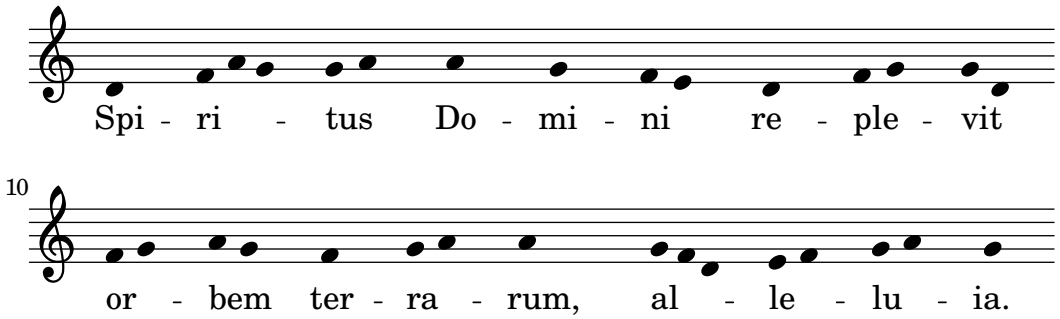


Otra situación común es la transcripción de cantos neumáticos o melismáticos, es decir, cantos con un número variable de notas en cada sílaba. En este caso, podemos establecer los grupos de sílabas claramente separados, normalmente también las subdivisiones de un melisma mayor. Una forma de conseguirlo es usar un `\time` fijo, p.ej. 1/4, y dejar que cada sílaba o grupo de notas llene uno de esos compases, con la ayuda de grupos de valoración especial o duraciones más breves. Si las líneas divisorias y todas las demás indicaciones rítmicas se hacen transparentes, y el espacio que rodea las líneas divisorias se aumenta, ello proporciona una aceptable representación del original, en notación moderna.

Para evitar que las sílabas de distinta anchura (como "-ri" y "-rum") dispongan los grupos de notas de las sílabas de forma irregular, se puede ajustar la propiedad `'X-extent` del objeto `LyricText` a un valor fijo. Otra forma más engorrosa sería añadir las sílabas como elementos de `\markup`. Si se necesitan más ajustes posteriormente, esto se puede hacer fácilmente con 'notas' `s`.

```
spiritus = \relative c' {
  \time 1/4
  \override Lyrics.LyricText.X-extent  = #'(0 . 3)
  d4 \tuplet 3/2 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4  s
  \tuplet 3/2 { g8 f d } e f g a g4
}

spirLyr = \lyricmode {
  Spi -- ri -- _ _ tus  _ Do -- mi -- ni  _ re -- ple -- _ vit _
  or -- _ bem _  ter -- ra -- _ rum, al -- _ _ le -- _ lu
  -- _ ia.
}
```

```
\score {
  \new Staff <<
    \new Voice = "chant" \spiritus
    \new Lyrics = "one" \lyricsto "chant" \spirLyr
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \override BarLine.X-extent = #'(-1 . 1)
      \hide Stem
      \hide Beam
      \hide BarLine
      \hide TupletNumber
    }
  }
}
```



Antigua y moderna desde la misma fuente

Continuará...

Marcado editorial

Continuará...

2.10 Músicas del mundo

El propósito de esta sección es remarcar los problemas de notación que son de relevancia a las tradiciones musicales distintas a la occidental.

2.10.1 Notación común para músicas no occidentales

Esta sección estudia la forma de introducir e imprimir partituras musicales que no pertenecen a la tradición clásica occidental, también conocida como *Período de la práctica común*.

Extensión de la notación y los sistemas de afinación

La notación clásica estándar (también conocida como la notación del *Período de la práctica común*) se usa normalmente en todo tipo de música, sin limitación a la música occidental 'clásica'. Esta notación se estudia en Sección 1.1.1 [Escritura de notas], página 1, y los distintos nombres de las notas que se pueden utilizar están explicados en [Nombres de las notas en otros idiomas], página 7.

Sin embargo, muchos tipos de música no occidental (y algunas clases de música folk occidental y tradicional) emplea sistemas de afiniación altenativos o extendidos que no encajan inmediatamente dentro de la notación clásica estándar.

En algunos casos aún se usa la notación estándar, estando implícitas las diferencias de altura. Por ejemplo, la notación de la *música árabe* se realiza con alteraciones accidentales de semitono y de cuarto de tonoc, estando determinadas por el contexto las exactas alteraciones de altura. Se usan por lo general los nombres de nota italianos, mientras que el archivo de inicio '`arabic.ly`' proporciona un conjunto adecuado de macros y definiciones que extienden la notación estándar. Para ver más detalles, consulte Sección 2.10.2 [Música árabe], página 452.

Otros tipos de música requieren notaciones extendidas o singulares. La *música clásica turca* o música otomana, por ejemplo, emplea formas melódicas conocidas como *makamlar*, cuyos intervalos están basados en divisiones de 1/9 de tono. Se siguen usando las notas del pentagrama occidental, pero con alteraciones accidentales exclusivas de la música turca, que están definidas en el archivo '`makam.ly`'. Para ver más información sobre la música clásica turca y los makamlar, consulte Sección 2.10.3 [Música clásica de Turquía], página 457.

PAra localizar los archivos de inicio como '`arabic.ly`' o '`makam.ly`' en su sistema, consulte Sección "Otras fuentes de información" in *Manual de Aprendizaje*.
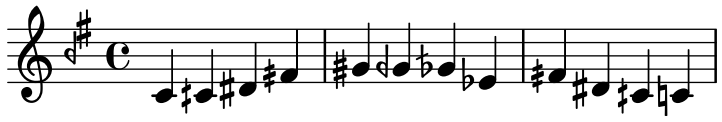
Fragmentos de código seleccionados

Ejemplo de «Makam»

El «Makam» es un tipo de melodía de Turquía que utiliza alteraciones microtonales de 1/9 de tono. Consulte el archivo de inicio '`makam.ly`' (véase el 'Manual de aprendizaje 2.18.2, 4.6.3 Otras fuentes de información' para averiguar la situación de este archivo) para ver detalles de los nombres de las notas y las alteraciones.

```
% Initialize makam settings
\include "makam.ly"

\relative c' {
  \set Staff.keySignature = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Véase también

Glosario musical: Sección "Common Practice Period" in *Glosario Musical*, Sección "makamlar" in *Glosario Musical*.

Manual de aprendizaje: Sección "Otras fuentes de inforamción" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.1.1 [Escritura de notas], página 1, [Nombres de las notas en otros idiomas], página 7, Sección 2.10.2 [Música árabe], página 452, Sección 2.10.3 [Música clásica de Turquía], página 457.

2.10.2 Música árabe

Esta sección destaca asuntos relativos a la música árabe.

Referencias para música árabe

La música árabe ha sido principalmente una tradición oral hasta el momento. Cuando se transcribe la música, se suele hacer en la forma de un esbozo sobre el que los intérpretes deben

improvisar ampliamente. En una medida creciente, se está adoptando la notación occidental con ciertas variantes para preservar y transmitir la música árabe.

Algunos elementos de la notación musical occidental como la transcripción de acordes o las particellas independientes, no se necesitan para tipografiar las piezas árabes, más tradicionales. Sin embargo hay ciertos problemas, como la necesidad de indicar intervalos que tienen una medida intermedia entre el semitono y el tono, además de los intervalos mayores y menores de la música occidental. También existe la necesidad de agrupar e indicar los numerosos maqams (modos) que forman parte de la música árabe.

Por lo general la notación musical árabe no trata de indicar con precisión los elementos microtonales que se encuentran presentes en la práctica musical.

Son de relevancia para la música árabe ciertos asuntos que se encuentran cubiertos en otras partes del manual:

- Los nombres de las notas y sus alteraciones (entre ellos, cuartos de tono) se pueden realizar como se estudia en Sección 2.10.1 [Notación común para músicas no occidentales], página 451.
- Las armaduras adicionales también pueden realizarse como se describe en [Armadura de la tonalidad], página 20.
- Los compases complejos pueden requerir agrupaciones manuales de las notas tal y como se describe en [Barras manuales], página 90.
- Los *takasim* o improvisaciones libres desde el punto de vista rítmico se pueden escribir omitiendo las barras de compás como aparece en [Música sin compasear], página 71.

Véase también

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 451, [Armadura de la tonalidad], página 20, [Barras manuales], página 90.
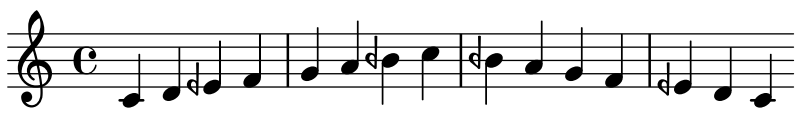
Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Nombres de nota árabes

Los nombres árabes tradicionales de las notas pueden llegar a ser bastante largos y no son adecuados para la escritura musical, y por ello no se utilizan. Los nombres ingleses de las notas no son muy familiares dentro de la educación musical árabe, por lo que en su lugar se utilizan los nombres italo-franco-españoles o del solfeo (`do, re, mi, fa, sol, la, si`); también se pueden utilizar modificadores (alteraciones). Los nombres italianos de las notas y las alteraciones accidentales están explicados en [Nombres de las notas en otros idiomas], página 7; el usao de la notación occidental estándar para realizar la notación de música no occidental se discute en Sección 2.10.1 [Notación común para músicas no occidentales], página 451.
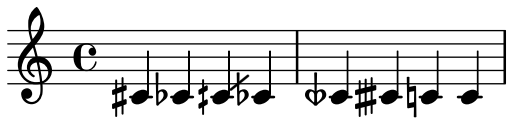
Por ejemplo, así es como puede ser la notación de la escala árabe *rast*:

```
\include "arabic.ly"
\relative do' {
  do re misb fa sol la sisb do sisb la sol fa misb re do
}
```



El símbolo del semi-bemol no coincide con el símbolo que se utiliza en la notación árabe. Se puede usar el símbolo `\dwn` definido en ‘`arabic.ly`’ precediendo a un bemol como vía alternativa si es importante usar el símbolo árabe específico del semi-bemol. El aspecto del símbolo del semi-bemol dentro de la armadura no se puede alterar mediante este método.

```
\include "arabic.ly"
\relative do' {
  \set Staff.extraNatural = ##f
  dod dob dosd \dwn dob dobsb dodsd do do
}
```



Véase también

Referencia de la notación: [Nombres de las notas en otros idiomas], página 7, Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Armaduras de tonalidad árabes

Además de las armaduras de tonalidad mayores y menores, están definidas en ‘`arabic.ly`’ las siguientes tonalidades: *bayati*, *rast*, *sikah*, *iraq* y *kurd*. Estas armaduras de tonalidad definen una pequeña cantidad de grupos de maqam, en contraste con los numerosos maqams que son de uso común.

En general, un maqam usa la armadura de su grupo o la de un grupo vecino, y las alteraciones que difieren se van marcando a lo largo de la música.

Por ejemplo, para indicar la armadura de una pieza en el maqam muhayer:

```
\key re \bayati
```

Aquí, *re* es la altura predeterminada del maqam muhayer, y *bayati* es el nombre del maqam básico dentro del grupo.

Al tiempo que la armadura indica el grupo, el título suele indicar el maqam específico, y así en este ejemplo el nombra del maqam muhayer debe aparecer en el título.

Se pueden indicar de la misma forma otros maqams dentro del mismo grupo bayati, como aparece en la tabla a continuación: bayati, hussaini, saba y ushaq. Todos ellos son variantes del maqam básico y más común dentro del grupo, que es bayati. Normalmente se diferencian del maqam básico en su tetracordo superior, o en ciertos detalles de discurso que no cambian su naturaleza fundamental como parientes.

El otro maqam del grupo (Nawa) se relaciona con el bayati a través de una modulación que se indica en la tabla entre paréntesis para aquellos maqams que son modulaciones de su maqam básico. Los maqams árabes admiten sólo modulaciones limitadas, a causa de la naturaleza de los instrumentos musicales árabes. El Nawa se puede indicar de la siguiente forma:

```
\key sol \bayati
```

En la música árabe, el mismo término que se utiliza para indicar un grupo de maqam, como bayati, es también un maqam que normalmente es el más importante del grupo, y podemos verlo como el maqam básico.

A continuación presentamos una posibilidad de agrupación que asigna los maqams más comunes a armaduras de tonalidad:

grupo de maqam	**tono**	**final**	**otros maqams del grupo (final)**
ajam	major	sib	jaharka (fa)
bayati	bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)

hijaz	kurd	re	shahnaz, shad arban (sol), hijazkar (do)
iraq	iraq	sisb	-
kurd	kurd	re	hijazkar kurd (do)
nahawand	minor	do	busalik (re), farah faza (sol)
nakriz	minor	do	nawa athar, hisar (re)
rast	rast	do	mahur, yakah (sol)
sikah	sikah	misb	huzam

Fragmentos de código seleccionados

Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keySignature` property, dentro del contexto `Staff`.
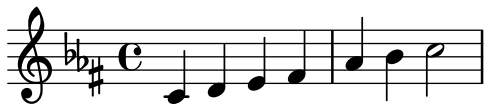
Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

`\set Staff.keySignature = #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)` donde, para cada elemento dentro de la lista, `octava` especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), `paso` especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y `alteración` es `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (observe la coma precedente.)

De forma alternativa, para cada elemento de la lista el uso del formato más conciso `(paso . alteración)` especifica que la misma alteración debe estar en todas las octavas.

He aquí un ejemplo de una posible armadura para generar una escala exátona:

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                              ((0 . 5) . ,FLAT)
                              ((0 . 3) . ,SHARP))
  c4 d e fis
  aes4 bes c2
}
```



Véase también

Glosario musical: Sección “maqam” in *Glosario Musical*, Sección “bayati” in *Glosario Musical*, Sección “rast” in *Glosario Musical*, Sección “sikah” in *Glosario Musical*, Sección “iraq” in *Glosario Musical*, Sección “kurd” in *Glosario Musical*.

Referencia de la notación: [Armadura de la tonalidad], página 20.

Manual de aprendizaje: Sección “Alteraciones accidentales y armaduras” in *Manual de Aprendizaje*.

Referencia de funcionamiento interno: Sección “KeySignature” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “World music” in *Fragmentos de código*, Sección “Pitches” in *Fragmentos de código*.

Indicaciones de compás árabes

Ciertas formas clásicas de música árabe y turca como el *Semai* utilizan compases poco comunes como el 10/8. Ello puede derivar en una agrupación automática de las notas muy distinta a la de las partituras impresas existentes, donde las notas pueden no estar agrupadas sobre el pulso, sino de una forma difícil de emular ajustando el barrado automático. La alternativa sería desactivar el barrado automático y realizar un barrado manual de las notas. Incluso si no es necesaria una correspondencia con música escrita ya existente, puede seguir siendo deseable ajustar el comportamiento del barrado automático y/o utilizar indicaciones de compás compuestas.

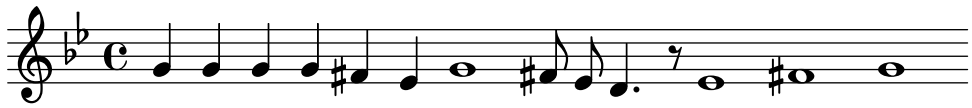
Fragmentos de código seleccionados

Improvisación de música árabe

Para las improvisaciones o *taqasim* que son libres durante unos momentos, se puede omitir la indicación de compás y se puede usar `\cadenzaOn`. Podría ser necesario ajustar el estilo de alteraciones accidentales, porque la ausencia de líneas divisorias hará que la alteración aparezca una sola vez. He aquí un ejemplo de cómo podría ser el comienzo de una improvisación *hijaz*:

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Véase también

Glosario musical: Sección “semai” in *Glosario Musical*, Sección “taqasim” in *Glosario Musical*.

Referencia de la notación: [Barras manuales], página 90, [Barras automáticas], página 79, [Música sin compasear], página 71, [Alteraciones accidentales automáticas], página 26, [Establecer el comportamiento de las barras automáticas], página 81, [Indicación de compás], página 62.
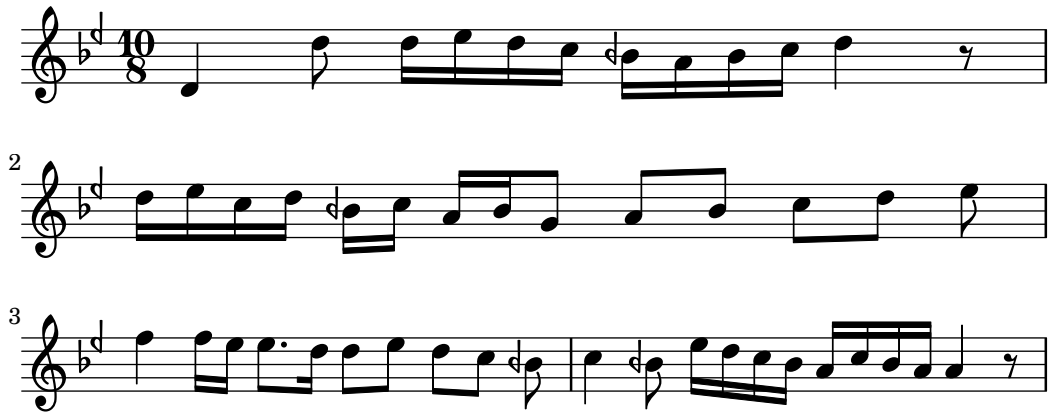
Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Ejemplo de música árabe

A continuación aparece una plantilla que emplea también el comienzo de un *Semai* turco muy común en la educación musical árabe con el objeto de ilustrar algunas de las peculiaridades de la notación árabe, como los intervalos intermedios y los modos poco usuales que tratamos en esta sección.

```
\include "arabic.ly"
\score {
  \relative re' {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8
```

```
    re4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re  [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
}
```



Véase también

Fragmentos de código: Sección "World music" in *Fragmentos de código*.

Lecturas adicionales sobre música árabe

1. *The music of the Arabs*, de Habib Hassan Touma [Amadeus Press, 1996], contiene un tratado de los maqams y sus formas de agrupación.

   También existen varios sitios web que explican los maqams, y algunos de ellos ofrecen ejemplos de audio, como:

   - http://www.maqamworld.com/
   - http://www.turath.org/

   Existen ciertas variantes en los detalles sobre la forma en que se agrupan los maqams, a pesar del relativo acuerdo existente acerca de los criterios de agrupación de maqams en función de los tetracordos inferiores comunes, o por medio de la modulación.

2. No es completamente consistente, incluso a veces dentro del mismo texto, la forma en que se deben especificar las armaduras para los maqams concretos. Sin embargo, es usual utilizar una armadura por grupo, más que una armadura distinta para cada maqam concreto.

   Varios métodos de los siguientes autores para el *Ud*, el laúd árabe, contienen ejemplos de composiciones turcas y árabes, principalmente.

   - Charbel Rouhana
   - George Farah
   - Ibrahim Ali Darwish Al-masri

2.10.3 Música clásica de Turquía

Esta sección destaca los problemas de relevancia para la notación de la música clásica de Turquía.

Referencias para música clásica de Turquía

La música clásica de Turquía se desarrollo en el imperio Otomano durante un período aproximadamente contemporáneo con la música clásica de Europa, y se ha prolongado hasta los ss.XX y XXI como una viva y característica tradición con sus propias formas composicionales, teorías y estilos de interpretación. Entre sus características más sobresalientes está el uso de intervalos microtonales basados en 'comas' de 1/9 de tono, a partir de los que se construyen las formas melódicas conocidas como *makam* (en plural *makamlar*).

Ciertos asuntos de relevancia respecto a la música clásica de Turquía se tratan en otras partes:

- Los nombres especiales de las notas y las alteraciones se dan en Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

Nombres de las notas en turco

Las notas en la música clásica de Turquía tienen tradicionalmente nombres únicos, y la base de las alturas sobre divisiones de 1/9 de tono significan que los makamlar utilizan un conjunto de intervalos completamente diferente de las escalas y modos occidentales: *koma* (1/9 de tono), *eksik bakiye* (3/9), *bakiye* (4/9), *kücük mücenneb* (5/9), *büyük mücenneb* (8/9), *tanîni* (un tono) y *artık ikili* (12/9 ó 13/9 de tono).

Desde el punto de vista de la notación moderna, es conveniente utilizar las notas occidentales (do, re, mi, . . . ó c, d, e, . . .) con alteraciones especiales que elevan o bajan las notas en intervalos de 1/9, 4/9, 5/9 y 8/9 de tono. Estas alteraciones están definidas en el archivo '`makam.ly`'.

La tabla siguiente relaciona:

- el nombre de estas alteraciones accidentales especiales,
- el sufijo de la alteración que se debe escribir detrás de las notas, y
- su alteración en altura expresada como una fracción de un tono.

Nombre de la alteración	**sufijo**	**alteración de altura**
büyük mücenneb (sostenido)	-bm	+8/9
kücük mücenneb (sostenido)	-k	+5/9
bakiye (sostenido)	-b	+4/9
koma (sostenido)	-c	+1/9
koma (bemol)	-fc	-1/9
bakiye (bemol)	-fb	-4/9
kücük mücenneb (bemol)	-fk	-5/9
büyük mücenneb (bemol)	-fbm	-8/9

Para ver una explicación más general sobre la notación musical no occidental, consulte Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

Véase también

Glosario musical: Sección "makam" in *Glosario Musical*, Sección "makamlar" in *Glosario Musical*.

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 451.

3 Entrada y salida generales

Esta sección trata de asuntos relacionados con la entrada y salida generales de LilyPond, más que con la notación específica.

3.1 Estructura del código de entrada

El formato principal de entrada para LilyPond son archivos de texto. Por convenio, el nombre de estos archivos termina en '`.ly`'.

3.1.1 Estructura de una partitura

Un bloque `\score` debe contener una expresión musical única delimitada por un par de llaves:
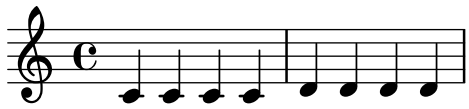
```
\score {
  ...
}
```

> **Nota:** Sólo puede haber **una** expresión musical dentro del bloque `\score`, y **debe** estar encerrada entre llaves.

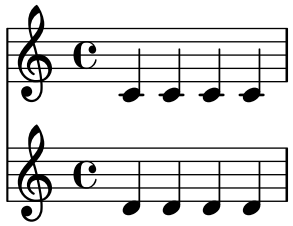
Esta expresión musical única puede tener cualquier tamaño, y puede contener otras expresiones musicales hasta cualquier grado de complejidad. Todos los siguientes ejemplos son expresiones musicales:

```
{ c'4 c' c' c' }
```

```
{
  { c'4 c' c' c' }
  { d'4 d' d' d' }
}
```



```
<<
  \new Staff { c'4 c' c' c' }
  \new Staff { d'4 d' d' d' }
>>
```



```
{
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { \flute }
      \new Staff { \oboe }
    >>
    \new StaffGroup <<
      \new Staff { \violinI }
```

```
      \new Staff { \violinII }
    >>
  >>
}
```

Los comentarios son una excepción a esta regla general (para ver otras excepciones, consulte Sección 3.1.5 [Estructura del archivo], página 463). Tanto los comentarios de una línea como los que están delimitados por `%{ ... %}` se pueden escribir en cualquier lugar dentro del archivo de entrada. Se pueden escribir dentro o fuera del bloque `\score`, y dentro o fuera de la expresión musical única dentro del bloque `\score`.

Recuerde que incluso dentro de un archivo que contenga solamente un bloque `\score`, éste se encuentra incluido implícitamente dentro de un bloque `\book`. Un bloque `\book` dentro de un archivo fuente produce al menos un archivo de salida, y de forma predeterminada el nombre del archivo de salida que se produce está derivado del nombre del archivo de entrada, de forma que '`fandangoforelephants.ly`' da como resultado '`fandangoforelephants.pdf`'.

(Para ver más detalles acerca de los bloques `\book`, consulte Sección 3.1.2 [Varias partituras en un libro], página 460, Sección 3.1.3 [Varios archivos de salida a partir de uno de entrada], página 461 y Sección 3.1.5 [Estructura del archivo], página 463.)

Véase también

Manual de aprendizaje: Sección "Trabajar sobre los archivos de entrada" in *Manual de Aprendizaje*, Sección "Explicación de las expresiones musicales" in *Manual de Aprendizaje*, Sección "La partitura es una (única) expresión musical compuesta" in *Manual de Aprendizaje*.

3.1.2 Varias partituras en un libro

Un documento puede contener varias piezas de música y texto. Son ejemplos un libro de estudios o una parte orquestal con varios movimientos. Cada movimiento se introduce con un bloque `\score`:

```
\score {
  ...música...
}
```

y los textos se introducen con un bloque `\markup`:

```
\markup {
  ...texto...
}
```

Todos los movimientos y textos que aparecen en el mismo archivo '`.ly`' se tipografían normalmente en la forma de un solo archivo de salida.

```
\score {
  ...
}
\markup {
  ...
}
\score {
  ...
}
```

Una excepción importante se da dentro de los documentos de lilypond-book, en los que tenemos que escribir explícitamente un bloque `\book`, pues en caso contrario sólo aparecerá en la salida el primer `\score` o `\markup`.

El encabezamiento de cada pieza se puede poner dentro del bloque `\score`. Antes de cada movimiento, se imprime el nombre de la pieza (`piece`) extraído del encabezamiento. El título

del libro entero se puede poner dentro del bloque `\book`, pero si no está presente se inserta el encabezamiento `\header` que está al principio del archivo.

```
\header {
  title = "Ocho miniaturas"
  composer = "Igor Stravinsky"
}
\score {
  ...
  \header { piece = "Romanze" }
}
\markup {
   ...texto de la segunda estrofa...
}
\markup {
   ...texto de la tercera estrofa...
}
\score {
  ...
  \header { piece = "Menuetto" }
}
```

Se pueden agrupar distintas piezas de música en partes de un libro utilizando bloques `\bookpart`. Las partes de libro se separan por medio de saltos de página, y pueden dar comienzo con un título, como el libro propiamente dicho, mediante la especificación del bloque `\header`.

```
\bookpart {
  \header {
    title = "Título del libro"
    subtitle = "Primera parte"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Segunda parte"
  }
  \score { ... }
  ...
}
```

3.1.3 Varios archivos de salida a partir de uno de entrada

Si queremos varios archivos de salida a partir del mismo archivo '`.ly`', podemos escribir varios bloques `\book`, donde cada uno de estos bloques `\book` dará como resultado un archivo de salida distinto. Si no especificamos ningún bloque `\book` en el archivo, LilyPond trata implícitamente todo el archivo como un solo bloque `\book`, véase Sección 3.1.5 [Estructura del archivo], página 463.

Al producir varios archivos a partir de un solo archivo fuente, Lilypond se asegura de que ninguno de los archivos de salida de ningún bloque `\book` sobreescribe el archivo de salida producido por un `\book` anterior del mismo archivo de entrada.

Lo hace añadiendo un sufijo al nombre de la salida para cada libro `\book` que utiliza el nombre del archivo de salida predeterminado, derivado del archivo fuente de entrada.

El comportamiento predeterminado es añadir un sufijo con el número de versión para cada nombre que pueda coincidir; así:

```
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
```

en el archivo fuente ‘`eightminiatures.ly`’, produce

- ‘`eightminiatures.pdf`’,
- ‘`eightminiatures-1.pdf`’ y
- ‘`eightminiatures-2.pdf`’.

3.1.4 Nombres de los archivos de salida

Lilypond le ofrece unos mecanismos que le permiten controlar los nombres de archivo que se utilizan por parte de los diversos motores finales al producir los archivos de salida.

En la sección anterior vimos cómo Lilypond evita las coincidencias de nombres al producir varias salidas a partir de un único archivo fuente de entrada. También podemos especificar nuestros propios sufijos para cada bloque `\book`, de manera que podemos producir, por ejemplo, archivos con los nombres ‘`eightminiatures-Romanze.pdf`’, ‘`eightminiatures-Menuetto.pdf`’ y ‘`eightminiatures-Nocturne.pdf`’ añadiendo una declaración `\bookOutputSuffix` dentro de cada bloque `\book`.

```
\book {
  \bookOutputSuffix "Romanze"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Menuetto"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Nocturne"
  \score { ... }
  \paper { ... }
}
```

También podemos especificar un nombre de archivo de salida diferente para cada bloque `book`, mediante la utilización de declaraciones `\bookOutputName`:

```
\book {
  \bookOutputName "Romanze"
  \score { ... }
  \paper { ... }
}
```

```
\book {
  \bookOutputName "Menuetto"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputName "Nocturne"
  \score { ... }
  \paper { ... }
}
```

El código anterior produce los siguientes archivos de salida:

- 'Romanze.pdf',
- 'Menuetto.pdf' y
- 'Nocturne.pdf'.

3.1.5 Estructura del archivo

Un archivo '.ly' puede contener cualquier número de expresiones del nivel superior, donde una expresión del nivel superior es una de las siguientes posibilidades:

- Una definición de salida, como \paper, \midi y \layout. Tal definición en el nivel más alto cambia los ajustes predeterminados para todo el libro. Si se introduce más de una de estas definiciones del mismo tipo en el nivel más alto, las definiciones se combinan, pero en situaciones de conflicto las definiciones más tardías tienen preferencia. Para ver los detalles sobre cómo afecta al bloque \layout, consulte Sección 4.2.1 [El bloque \layout], página 524.
- Una expresión directa de Scheme, como `#(set-default-paper-size "a7" 'landscape)` o `#(ly:set-option 'point-and-click #f)`.
- Un bloque \header. Esto establece el bloque de encabezamiento global (es decir, al principio del archivo). Es el bloque que contiene los ajustes predeterminados de los campos de encabezamientos como el compositor, título, etc. para todos los libros que están dentro del archivo (véase [Explicación de los títulos], página 465).
- Un bloque \score. Esta partitura se reúne con otras partituras del nivel superior, y se combina con ellas formando un \book (libro) único. Este comportamiento se puede cambiar estableciendo el valor de la variable toplevel-score-handler en el nivel superior. El *handler* predeterminado se define en el archivo de inicio '../scm/lily.scm'.
- Un bloque \book combina lógicamente varios movimientos (es decir, varios bloques \score) en un solo documento. Si hay varias partituras \score, se crea un solo archivo de salida para cada bloque \book, en el que se concatenan todos los movimientos correspondientes. La única razón para especificar explícitamente bloques \book en un archivo de entrada '.ly' es si queremos crear varios archivos de salida a partir de un único archivo de entrada. Una excepción se da dentro de los documentos de lilypond-book, en los que tenemos que escribir explícitamente un bloque \book si queremos tener más de una única partitura \score o marcado \markup en el mismo ejemplo. Se puede cambiar este comportamiento estableciendo el valor de la variable toplevel-book-handler en el nivel superior. El *handler* predeterminado se define en el archivo de inicio '../scm/lily.scm'.
- Un bloque \bookpart. Un libro puede dividirse en varias partes, utilizando bloques \bookpart, con objeto de facilitar la inserción de saltos de página o usar distintos ajustes de \paper en las distintas partes.
- Una expresión musical compuesta, como

  ```
  { c'4 d' e'2 }
  ```

Esto añade la pieza al `\score` y la conforma dentro de un libro único junto a todas las demás partituras `\score` y expresiones musicales del nivel superior. En otras palabras, un archivo que contiene sólo la expresión musical anterior se traduce a

```
\book {
  \score {
    \new Staff {
      \new Voice {
        { c'4 d' e'2 }
      }
    }
    \layout { }
  }
  \paper { }
  \header { }
}
```

Se puede cambiar este comportamiento estableciendo el valor de la variable `toplevel-music-handler` en el nivel más alto. El *handler* predeterminado se define en el archivo de inicio '`../scm/lily.scm`'.

- Un texto de marcado, por ejemplo una estrofa

  ```
  \markup {
     2.  Estrofa dos, línea uno
  }
  ```

  Los textos de marcado se colocan encima, en medio o debajo de las partituras o expresiones musicales, según donde aparezcan escritos.

- Una variable, como

  ```
  fulano = { c4 d e d }
  ```

  Esto se puede utilizar más tarde en el archivo escribiendo `\fulano`. El nombre de las variables ha de ser exclusivamente alfabético; sin números, guiones ni barras bajas.

El ejemplo siguiente muestra tres cosas que se pueden introducir en el nivel superior:

```
\layout {
  % No justificar la salida
  ragged-right = ##t
}

\header {
   title = "Do-re-mi"
}

{ c'4 d' e2 }
```

En cualquier punto del archivo se pueden introducir cualquiera de las instrucciones léxicas siguientes:

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Un comentario de una línea, determinado por un símbolo `%` al principio.
- Un comentario de varias líneas, delimitado por `%{ ... %}`.

Por lo general se ignoran los espacios entre elementos del flujo de entrada, y se pueden omitir o aumentar tranquilamente para mejorar la legibilidad. Sin embargo, los espacios se deben utilizar siempre para evitar errores, en las siguientes situaciones:

- Alrededor de todas y cada una de las llaves curvas de apertura y cierre.
- A continuación de las instrucciones y variables, es decir, todos los elementos que comienzan con un signo \ de barra invertida.
- A continuación de todo elemento que se deba interpretar como expresión de Scheme, es decir, todo elemento que comience por un signo `#`.
- Para separar los elementos de una expresión de Scheme.
- Dentro de `lyricmode` antes y después de las instrucciones `\set` y `\override`.

Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” in *Manual de Aprendizaje*.

Referencia de la notación: [Explicación de los títulos], página 465, Sección 4.2.1 [El bloque `\layout`], página 524.

3.2 Títulos y encabezamientos

Casi toda la música impresa tiene un título y el nombre del compositor; ciertas piezas tienen mucha más información.

3.2.1 Crear títulos, encabezamientos y pies de página

Explicación de los títulos

Cada bloque `\book` de un solo archivo de entrada produce un archivo de salida distinto, véase Sección 3.1.5 [Estructura del archivo], página 463. Dentro de cada archivo de salida están disponibles tres tipos de zonas para títulos: *Títulos de libro* al principio de cada libro, *Títulos de parte de libro* al principio de cada parte de un libro y *Títulos de partitura* al comienzo de una partitura.

Los valores de los campos para encabezamientos tales como `title` (para el título) y `composer` (para el autor) se establecen en bloques `\header` (para ver la sintaxis de los bloques `\header` y una lista completa de los campos disponibles de forma predeterminada, consulte [Presentación predeterminada de los títulos de partes de libro y partitura], página 469). Tanto los títulos de libro como los títulos de parte de libro y los títulos de partitura pueden contener los mismos campos, aunque de forma predeterminada los campos de los títulos de partitura se limitan a `piece` (pieza) y `opus`.

Los bloques `\header` se pueden situar en cuatro lugares distintos para formar una jerarquía descendente de bloques `\header`:

- Al principio del archivo de entrada, antes de cualquier otro bloque `\book`, `\bookpart` o `\score`.
- Dentro de un bloque `\book` pero fuera de los bloques `\bookpart` y `\score` que están dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` dentro de dicha parte de libro.
- Después de la expresión musical de un bloque `\score`.

Los valores de los campos se fitran a través de esta jerarquía, persistiendo aquellos que se establecen más arriba a no ser que se sobreescriban por un valor establecido más abajo en la jerarquía, y así:

- El título de un libro deriva de los campos establecidos al principio del archivo de entrada, modificado por campos establecidos en el bloque `\book`. Los campos resultantes se usan para imprimir el título de libro para dicho libro, siempre y cuando haya otro material que genere una página al comienzo del libro, antes de la primera parte de libro. Es suficiente una sola instrucción `\pageBreak`.
- Un título de parte de libro se deriva de los campos fijados al principio del archivo de entrada, modificado por los campos establecidos dentro del bloque `\book`, y posteriormente modificado por los campos que se fijan dentro del bloque `\bookpart`. Los valores resultantes se usan para imprimir el título de parte de libro para esa parte.
- Un título de partitura se deriva de los campos que se fijan al principio del archivo de entrada, modificados por los campos fijados dentro del bloque `\book`, posteriormente modificados por los campos que se fijan dentro del bloque `\bookpart` y finalmente modificados por los campos establecidos dentro del bloque `\score`. Los valores restulantes se utilizan para imprimir el título de partitura. Observe, sin embargo, que de forma predeterminada solamente se imprimen los campos `piece` y `opus` en los títulos de partitura a no ser que la variable de `\paper`, `print-all-headers`, tenga el valor `#t`.

Nota: Recuerde que al poner un bloque `\header` dentro de un bloque `\score`, la expresión musical debe aparecer antes del bloque `\header`.

No es necesario aportar bloques `\header` en los cuatro lugares: cualquiera de ellos o incluso todos pueden omitirse. De forma similar, los archivos de entrada sencillos pueden omitir los bloques `\book` y `\bookpart`, dejando que se creen de forma implícita.

Si el lobro solamente tiene una partitura, el bloque `\header` debiera situarse normalmente al principio del archivo de forma que solamente se produce una zona de título de parte de libro, haciendo que estén disponibles para su uso todos los títulos de encabezamientos.

Si el libro tiene varias partituras, son posibles distintas combinaciones de bloques `\header`, según los distintos tipos de publicaciones musicales. Por ejemplo, si la publicación contiene varias piezas del mismo autor, un bloque `\header` colocado al principio del archivo que especifique el título del libro y el autor, con bloques `\header` en cada bloque `\score` que especifique la `piece` y/o el `opus` sería lo más adecuado, como aquí:

```
\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \new Staff \relative g, {
    \clef bass
    \key g \major
    \repeat unfold 2 { g16( d' b') a b d, b' d, } |
    \repeat unfold 2 { g,16( e' c') b c e, c' e, } |
  }
  \header {
    piece = "Prélude."
  }
}
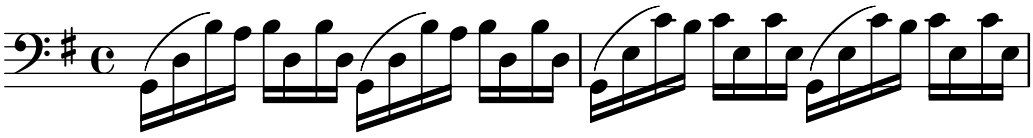
\score {
  \new Staff \relative b {
```

```
    \clef bass
    \key g \major
    \partial 16 b16 |
    <g, d' b'~>4 b'16 a( g fis) g( d e fis) g( a b c) |
    d16( b g fis) g( e d c) b(c d e) fis( g a b) |
  }
  \header {
    piece = "Allemande."
  }
}
```
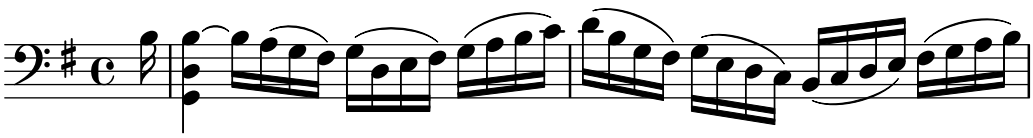
SUITE I.

J. S. Bach.

Prélude.



Allemande.



Son posibles disposiciones más complejas. Por ejemplo, los campos de texto extraídos del bloque `\header` de un libro se pueden imprimir en todos los títulos de partitura, con ciertos campos sobreescritos y otros suprimidos manualmente:

```
\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Do not display the tagline for this book
    tagline = ##f
  }
  \markup { \vspace #1 }
  \score {
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
    \header {
      title = "PRAELUDIUM I"
      opus = "BWV 846"
```

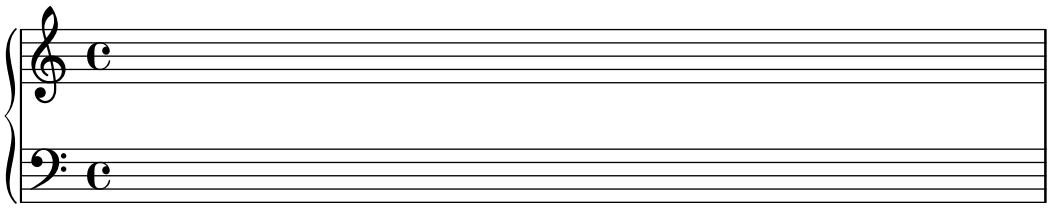
```
      % Do not display the subtitle for this score
      subtitle = ##f
    }
  }
  \score {
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
    \header {
      title = "FUGA I"
      subsubtitle = "A 4 VOCI"
      opus = "BWV 846"
      % Do not display the subtitle for this score
      subtitle = ##f
    }
  }
}
```

DAS WOHLTEMPERIRTE CLAVIER

TEIL I

PRAELUDIUM I

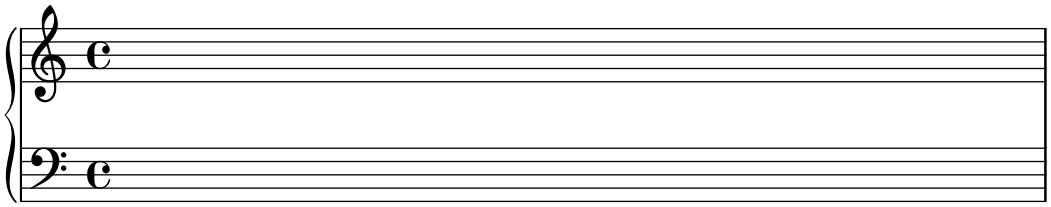
BWV 846



FUGA I

A 4 VOCI

BWV 846



Véase también

Referencia de la notación: Sección 3.1.5 [Estructura del archivo], página 463, [Presentación predeterminada de los títulos de partes de libro y partitura], página 469, [Personalización de los títulos], página 473.

Presentación predeterminada de los títulos de partes de libro y partitura

Este ejemplo muestra todas las variables de `\header`:

```
\book {
  \header {
      % Los siguientes campos están centrados
    dedication = "Dedication (dedicatoria)"
    title = "Title (título)"
    subtitle = "Subtitle (subtítulo)"
    subsubtitle = "Subsubtitle (sub-subtítulo)"
      % Los siguientes campos se distribuyen regularmente sobre una línea
      % el campo "instrument" también aparece en las siguientes páginas
    instrument = \markup \with-color #green "Instrument (instrumento)"
    poet = "Poet (poeta)"
    composer = "Composer (autor)"
      % Los siguientes campos se sitúan en los extremos opuestos de la misma línea
    meter = "Meter (metro)"
    arranger = "Arranger (arreglista)"
      % Los siguientes campos están centrados en la parte inferior
    tagline = "la tagline o línea de etiqueta va al final de la úitlma página"
    copyright = "el copyright va al final de la primera página"
  }
  \score {
    { s1 }
    \header {
        % Los siguientes campos se colocan en los extremos de una línea
      piece = "Piece (pieza) 1"
      opus = "Opus 1"
    }
  }
  \score {
    { s1 }
    \header {
        % Los siguientes campos se sitúan en los extremos de una línea
      piece = "Piece (pieza) 2 en la misma página"
      opus = "Opus 2"
    }
  }
  \pageBreak
  \score {
    { s1 }
    \header {
        % Los siguientes campos se sitúan en los extremos de una línea
      piece = "Piece (pieza) 3 en una página nueva"
      opus = "Opus 3"
    }
  }
}
```

Dedication (dedicatoria)

Title (título)

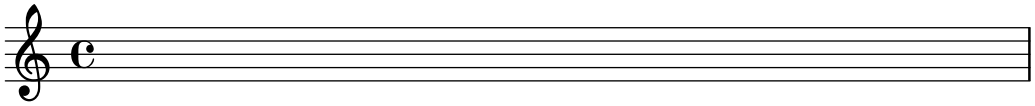
Subtitle (subtítulo)

Subsubtitle (sub-subtítulo)

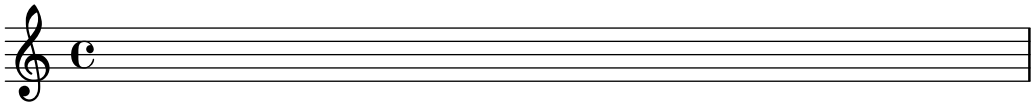
Poet (poeta) **Instrument (instrumento)** Composer (autor)

Meter (metro) Arranger (arreglista)

Piece (pieza) 1 Opus 1



Piece (pieza) 2 en la misma página Opus 2



el copyright va al final de la primera página

2 Instrument (instrumento)

Piece (pieza) 3 en una página nueva Opus 3



la tagline o línea de etiqueta va al final de la úitlma página

Observe que

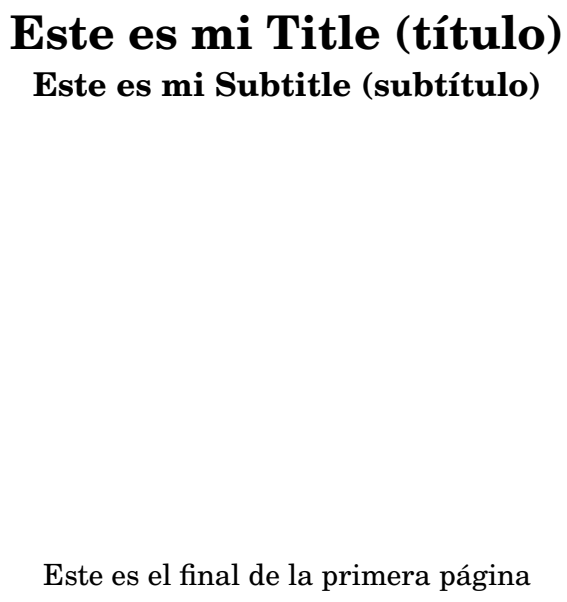
- El nombre de instrumento se repite en todas las páginas.
- Solamente `piece` (la pieza) y `opus` se imprimen en una `\score` cuando la variable de papel `print-all-headers` se establece a `##f` (el valor predeterminado).
- Los campos de texto que se dejan sin establecer dentro de un bloque `\header` se sustituyen por elementos de marcado nulos `\null` de manera que no ocupen espacio innecesariamente.
- Los ajustes predeterminados para `scoreTitleMarkup` sitúan los campos de texto `piece` y `opus` en extremos opuestos de la misma línea.

Para cambiar la disposición predeterminada, véase [Personalización de los títulos], página 473.
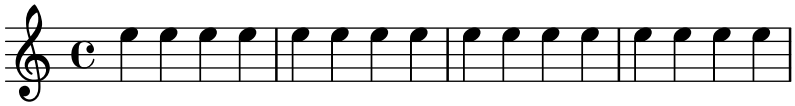
Si un bloque `\book` comienza inmediatamente con un bloque `\bookpart`, no se imprime ningún título de libro, pues no existe ninguna página en que imprimirlo. Si es necesario un título de libro, inicie el bloque `\book` con algún material de marcado o con una instrucción `\pageBreak`.

Utilice la variable `breakbefore` dentro de un bloque `\header` que está propiamente dentro de un bloque `\score` para hacer que los títulos del bloque `\header` del nivel superior aparezcan ocupando toda la primera página, empezando la música (definida en el bloque `\score`) en la página siguiente.

```
\book {
  \header {
    title = "Este es mi Title (título)"
    subtitle = "Este es mi Subtitle (subtítulo)"
    copyright = "Este es el final de la primera página"
  }
  \score {
    \repeat unfold 4 { e'' e'' e'' e'' }
    \header {
      piece = "Esto es la música"
      breakbefore = ##t
    }
  }
}
```



2
Esto es la música



Music engraving by LilyPond 2.18.2—www.lilypond.org

Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” in *Manual de Aprendizaje*

Referencia de la notación: [Personalización de los títulos], página 473, Sección 3.1.5 [Estructura del archivo], página 463.

Archivos de inicio: ‘`ly/titling-init.ly`’.

Disposición predeterminada de las cabeceras y pies de página

Las *cabeceras* y los *pies* de página son líneas de texto que aparecen en la parte más alta y en la más baja de las páginas separadas del texto principal de un libro. Se controlan mediante las siguientes variables de `\paper`:

- `oddHeaderMarkup` (marcado de cabecera impar)
- `evenHeaderMarkup` (marcado de cabecera par)
- `oddFooterMarkup` (marcado de pie impar)
- `evenFooterMarkup` (marcado de pie par)
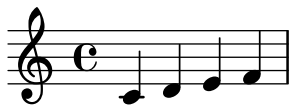
Estas variables de marcado sólo pueden acceder a los campos de texto extraídos de bloques `\header` del nivel superior (que se aplican a todas las partituras del libro) y se definen en el archivo ‘`ly/titling-init.ly`’. De forma predeterminada:

- los números de página se sitúan automáticamente en el extremo superior izquierdo (si es par) o superior derecho (si es impar), a partir de la segunda página.
- el campo de texto `instrument` se sitúa en el centro en cada página, a partir de la segunda página.
- el texto de `copyright` se centra en la parte baja de la primera página.
- la línea `tagline` se centra al final de la última página, y debajo del texto de `copyright` si sólo hay una página.

La línea de etiqueta tagline predeterminada se puede cambiar añadiendo un campo `tagline` en el bloque `\header` del nivel superior.

```
\book {
  \header {
    tagline = "... music notation for Everyone"
  }
  \score {
```

```
    \relative c' {
      c4 d e f
    }
  }
}
```



... music notation for Everyone

Para eliminar la línea `tagline`, establezca su valor a `##f`.

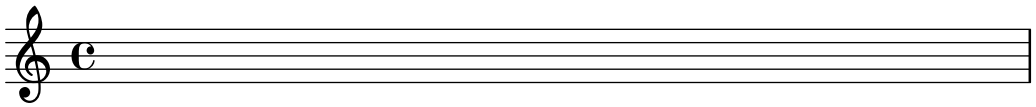
3.2.2 Títulos, encabezamientos y pies de página personalizados

Formateo personalizado del texto de los bloques de título

Se pueden usar instrucciones `\markup` estándar para personalizar el texto de cualquier cabecera, pie o título dentro del bloque `\header`.

```
\score {
  { s1 }
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    opus = \markup { \italic "BWV 846" }
  }
}
```

PRAELUDIUM I *BWV 846*



Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 233.

Personalización de los títulos

Las instrucciones `\markup` dentro del bloque `\header` son útiles para dar un formato simple al texto, pero no permiten un control preciso sobre la colocación de los títulos. Para personalizar la colocación de los campos de texto, cambie una o las dos variables de `\paper` siguientes:

- `bookTitleMarkup` (marcado de título del libro)
- `scoreTitleMarkup` (marcado de título de la partitura)

La colocación de los títulos cuando se utilizan los valores predeterminados de estas variables de `\markup` se muestra en los ejemplos de [Presentación predeterminada de los títulos de partes de libro y partitura], página 469.

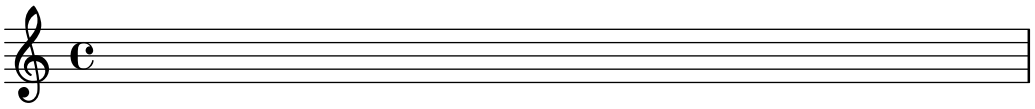
Los ajustes predeterminados para `scoreTitleMarkup` tal y como están definidos en el archivo '`ly/titling-init.ly`' son:

```
scoreTitleMarkup = \markup { \column {
  \on-the-fly \print-all-headers { \bookTitleMarkup \hspace #1 }
  \fill-line {
    \fromproperty #'header:piece
    \fromproperty #'header:opus
  }
}
}
```

Esto sitúa los campos de texto `piece` y `opus` en extremos opuestos de la misma línea:

```
\score {
  { s1 }
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
}
```

PRAELUDIUM I BWV 846



Este ejemplo redefine `scoreTitleMarkup` de manera que el campo de texto `piece` aparece centrado y en un tipo de letra grande y en negrita.
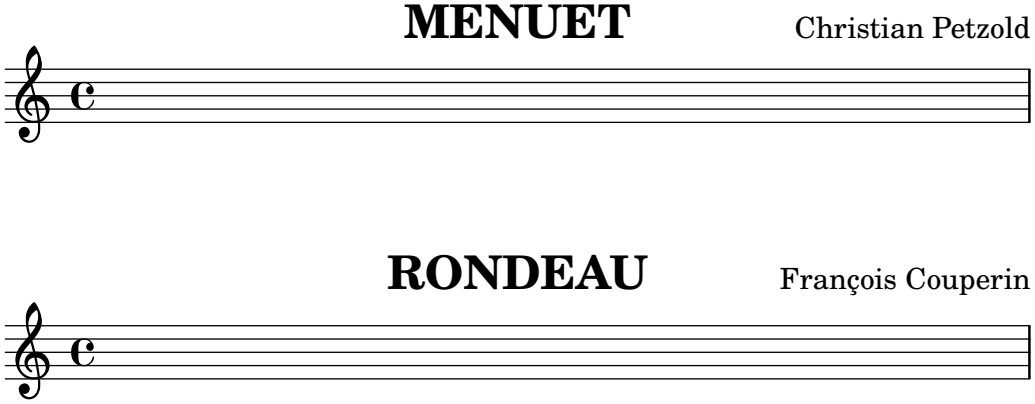
```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
    { s1 }
    \header {
      piece = "PRAELUDIUM I"
      opus = "BWV 846"
    }
  }
}
```



Los campos de texto que normalmente no son efectivos dentro de los bloques `\header` de una partitura se pueden imprimir el la zona del Título de la partitura si se coloca `print-all-headers` dentro del bloque `\paper`. Una desventaja de la utilización de este método es que los campos de texto que están orientados específicamente para la zona del título de parte de libro han de suprimirse manualmente en cada uno de los bloques `\score`. Véase [Explicación de los títulos], página 465.
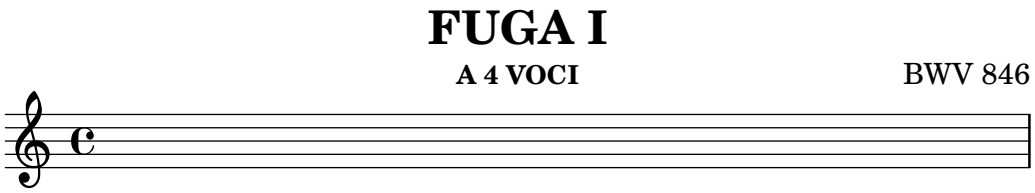
Para evitarlo, añada el campo de texto deseado a la definición de `scoreTitleMarkup`. En el ejemplo siguiente, el campo de texto `composer` (asociado normalmente con `bookTitleMarkup`) se añade a `scoreTitleMarkup`, permitiendo que cada partitura muestre un compositor diferente:

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
  }
  \header { tagline = ##f }
  \score {
    { s1 }
    \header {
      piece = "MENUET"
      composer = "Christian Petzold"
    }
  }
  \score {
    { s1 }
    \header {
      piece = "RONDEAU"
      composer = "François Couperin"
    }
  }
}
```



También podemos crear nuestros propios campos de texto personalizados, y referirnos a ellos en la definición del elemento de marcado.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \override #`(direction . ,UP) {
          \dir-column {
            \center-align \fontsize #-1 \bold
              \fromproperty #'header:mycustomtext %% User-defined field
            \center-align \fontsize #4 \bold
              \fromproperty #'header:piece
          }
        }
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
    { s1 }
    \header {
      piece = "FUGA I"
      mycustomtext = "A 4 VOCI" %% User-defined field
      opus = "BWV 846"
    }
  }
}
```



Véase también

Referencia de la notación: [Explicación de los títulos], página 465.

Disposición personalizada de cabeceras y pies de página

Las instrucciones `\markup` dentro del bloque `\header` son de utilidad para dar formato al texto de una manera sencilla, pero no permiten un control preciso sobre la colocación de las cabeceras y los pies de página. Para personalizar la colocación de los campos de texto, use una o más de las siguientes variables de `\paper`:

- `oddHeaderMarkup` (marcado de encabezamiento impar)
- `evenHeaderMarkup` (marcado de encabezamiento par)
- `oddFooterMarkup` (marcado de pie de página impar)
- `evenFooterMarkup` (marcado de pie de página par)

La instrucción de marcado `\on-the-fly` se puede utilizar para añadir elementos de forma condicional al texto de encabezamiento y pie de página definido dentro del bloque `\paper`, usando la sintaxis siguiente:

```
variable = \markup {
  ...
  \on-the-fly  \procedimiento  marcado
  ...
}
```

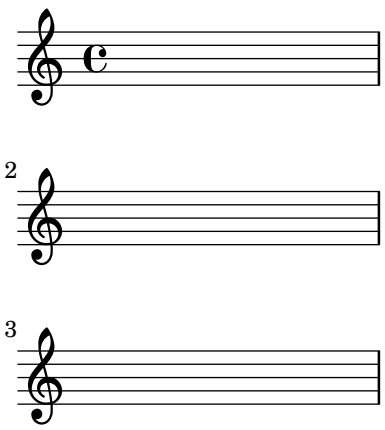
El *procedimiento* se llama cada vez que se evalúa la instrucción `\markup` en que ésta aparece. El *procedimiento* debería comprobar si se cumple una condición determinada e interpretar (es decir: imprimir) el argumento *marcado* si, y sólo si, la condición es verdadera.

Se proveen un cierto número de procedimientos ya hechos para la comprobación de diversas condiciones:

Nombre del procedimiento	**Condición que se comprueba**
print-page-number-check-first	¿debería imprimirse esta página?
create-page-number-stencil	¿es print-page-numbers verdadero?
print-all-headers	¿es print-all-headers verdadero?
first-page	¿es la primera página del libro?
(on-page nmbr)	¿es el número de página = nmbr?
last-page	¿es la última página del libro?
not-first-page	¿no es la primera página del libro?
part-first-page	¿es la primera página de la parte de libro?
part-last-page	¿es la última página de la parte de libro?
not-single-page	¿es el núm. de páginas en la parte de libro > 1?

El ejemplo siguiente centra los números de página en la parte baja de las páginas. En primer lugar, los ajustes predeterminados para `oddHeaderMarkup` y `evenHeaderMarkup` se eliminan definiéndolos como un marcado *nulo*. Después de esto, se redefine `oddFooterMarkup` con el número de página centrado. Finalmente, `evenFooterMarkup` recibe la misma disposición definiéndola como `\oddFooterMarkup`:

```
\book {
  \paper {
    print-page-number = ##t
    print-first-page-number = ##t
    oddHeaderMarkup = \markup \null
    evenHeaderMarkup = \markup \null
    oddFooterMarkup = \markup {
      \fill-line {
        \on-the-fly \print-page-number-check-first
        \fromproperty #'page:page-number-string
      }
    }
    evenFooterMarkup = \oddFooterMarkup
  }
  \score {
    \new Staff { s1 \break s1 \break s1 }
  }
}
```



1

Se pueden combinar varias condiciones de `\on-the-fly` con un operador 'and', por ejemplo:

```
\on-the-fly \first-page
\on-the-fly \last-page
{ \markup ... \fromproperty #'header: ... }
```

determina si la salida es una sola página.

Véase también

Referencia de la notación: [Explicación de los títulos], página 465, [Presentación predeterminada de los títulos de partes de libro y partitura], página 469.

Archivos instalados: '`../ly/titling-init.ly`'.

3.2.3 Crear notas al pie

Se pueden crear dos tipos de notas al pie: automáticas y manuales.

Notas al pie dentro de expresiones musicales

Panorámica de las notas al pie

Las notas al pie dentro de expresiones musicales caen dentro de dos categorías:

Notas al pie basadas en eventos
: Se adjuntan a un evento en particular. Son ejemplos de tales eventos las notas sueltas, las articulaciones (como indicaciones de digitación, acentos, indicaciones dinámicas) y los post-eventos (como las ligaduras de expresión y las barras de corchea manuales). La forma general de las notas al pie basadas en eventos es como sigue:

  `[`*`dirección`*`] \footnote [`*`marca`*`] `*`desplazamiento texto-de-la-nota música`*

Notas al pie basadas en tiempo
: Se ligan a un punto temporal determinado, dentro de un contexto musical. Algunas instrucciones como `\time` y `\clef` en realidad no usan eventos para crear objetos como la indicación de compás y la clave. Un acorde tampoco crea un evento por sí mismo: su plica o corchete se crea al final de un paso de tiempo (nominalmente, a través de uno de los eventos de nota que contiene). Una nota al pie basada en tiempo permite anotar tales objetos de presentación sin referirse a ningún evento.

  Una nota al pie basada en tiempo permite que tales objetos de presentación se puedan anotar sin hacer referencia a un evento. La forma general para las notas al pie basadas en tiempo es:

  `\footnote [`*`marca`*`] `*`desplazamiento texto-de-la-nota [Contexto].Nombre-del-Grob`*

Los elementos para las dos formas son como sigue:

dirección Si (y sólo si) el `\footnote` se está aplicando a un post-evento o articulación, debe ir precedida de un indicador de dirección (`-, _, ^`) con el objeto de adjuntar la *música* (con una marca de nota al pie) a la nota o silencio precedente.

marca es un elemento de marcado o cadena de caracteres que especifica la marca de nota al pie que se usa para marcar tanto el punto de referencia como la propia nota al pie en la parte inferior de la página. Se puede omitir (o, de forma equivalente, sustituirse por `\default`) en cuyo caso se genera un número secuencialmente de forma automática. Tales secuencias numéricas se reinician en cada página que contiene una nota al pie.

desplazamiento
es una pareja de números tal como '`#(2 . 1)`' que especifica los desplazamientos en X y en Y en unidades de espacios de pentagrama a partir del límite del objeto en que se desea situar la marca. Los valores de desplazamiento positivos se toman a partir del borde superior derecho, los valores negativos a partir del borde inferior izquierdo y cero implica que la marca se centra sobre el borde.

Contexto es el contexto en que se crea el grob que recibe la nota al pie. Se puede omitir si el grob está en un contexto del nivel inferior, p.ej un contexto `Voice`.

Nombre-del-Grob
especifica un tipo de grob que marcar (como '`Flag`' para el corchete). Si se da, la nota al pie no se adjunta a una expresión musical en particular, sino a todos los grobs del tipo especificado que ocurren en dicho momento de tiempo musical.

texto-de-la-nota
es el elemento de marccado o cadena de caracteres que especifica el texto de la nota al pie que utilizar en la parte baja de la página.
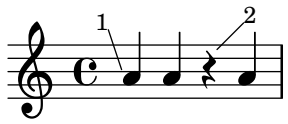
música es el evento musical, post-evento o articulación que se está anotando.

Notas al pie basadas en eventos

Una nota al pie se adjunta a un objeto de presentación causado directamente por el evento que corresponde a *música*, con la sintaxis siguiente:

```
\footnote [marca] desplazamiento texto-de-la-nota música
```

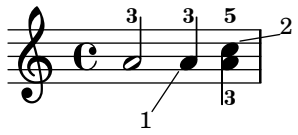
```
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(-1 . 3) "Una nota" a4
    a4
    \footnote #'(2 . 2) "Un silencio" r4
    a4
  }
}
```



[1]Una nota
[2]Un silencio

No es posible marcar un acrode *completo* con una nota al pie basada en evento: un acorde, incluso aquel que contiene una sola nota, no produce un verdadero evento por sí mismo. Sin embargo, las notas individuales que están *dentro* del acorde sí se pueden marcar:

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(2 . 3) "No funciona" <a-3>2
    <\footnote #'(-2 . -3) "Funciona" a-3>4
    <a-3 \footnote #'(3 . 1/2) "También funciona" c-5>4
  }
}
```

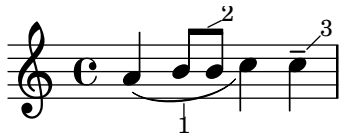


[1]Funciona
[2]También funciona

Si se desea que la nota al pie se adjunte a un post-evento o articulación, la instrucción `\footnote` *debe* ir precedida por un indicador de dirección, `-`, `_`, `^`, y seguida por el post-evento o articulación que se desea anotar como argumento *música*. En esta forma, la `\footnote` se puede considerar como una simple copia de su último argumento con una marca de nota al pie adjunta. La sintaxis es:

dirección `\footnote [`*marca*`]` *desplazamiento texto-de-la-nota música*

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    a4_\footnote #'(0 . -1) "Ligadura forzada hacia abajo" (
    b8^\footnote #'(1 . 0.5) "Barra manual forzada hacia arriba" [
    b8 ]
    c4 )
    c-\footnote #'(1 . 1) "Tenuto" --
  }
}
```

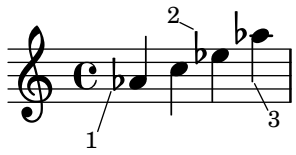


[1]Ligadura forzada hacia abajo
[2]Barra manual forzada hacia arriba
[3]Tenuto

notas al pie basadas en tiempo

Si el objeto de presentación que recibe la nota al pie está causado *indirectamente* por un evento (como un objeto `Accidental`, alteración, o `Stem`, plica, causados por un evento `NoteHead`, cabeza de nota), el nombre del grob *GrobName* del objeto de presentación es necesario después del texto de la nota al pie, en lugar de *música*:

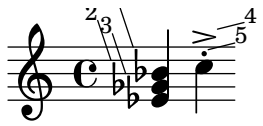
```
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(-1 . -3) "Un bemol" Accidental
    aes4 c
    \footnote #'(-1 . 0.5) "Otro bemol" Accidental
    ees
    \footnote #'(1 . -2) "Una plica" Stem
    aes
  }
}
```



[1]Un bemol
[2]Otro bemol
[3]Una plica

Sin embargo, observe que cuando se especifica el nombre de un grob, se adjunta una nota al pie a todos los grobs de ese tipo dentro del instante de tiempo actual:
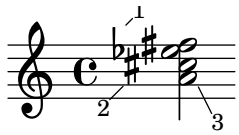
```
\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote #'(-1 . 3) "Un bemol" Accidental
    <ees ges bes>4
    \footnote #'(2 . 0.5) "Articulación" Script
    c'->-.
  }
}
```



[1]Un bemol
[2]Un bemol
[3]Un bemol
[4]Articulación
[5]Articulación

Una nota dentro de un acorde puede recibir una nota al pie individual, basada en evento. El único grob causado directamente por una nota de un acorde es '`NoteHead`', por lo que la instrucción de nota al pie basada en evento *solamente* es apta para añadir una nota al pie a la '`NoteHead`' dentro de un acorde. Todos los demás grobs de notas de acorde están causados indirectamente. La propia instrucción `\footnote` no ofrece ninguna sintaxis para especificar al mismo tiempo un tipo concreto de grob *y también* un evento concreto al que adjuntar la nota. Sin embargo, se puede usat una instrucción `\footnote` basada en tiempo para especificar el tipo de grob, y luego añadir la instrucción `\single` como prefijo, con el objeto de que se aplique solamente al evento siguiente:

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    < \footnote #'(1 . -2) "An A" a
      \single \footnote #'(-1 . -1) "Un sostenido" Accidental
      cis
      \single \footnote #'(0.5 . 0.5) "Un bemol" Accidental
      ees fis
    >2
  }
}
```



[1]Un bemol
[2]Un sostenido
[3]An A

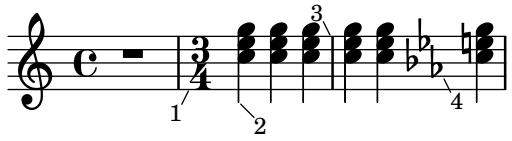
Nota: Cuando las notas al pie se adjuntan a varios elementos musicales que están dentro del mismo momento musical, como ocurre en el ejemplo anterior, las notas al pie se numeran desde los elementos más altos a los más bajos tal y como aparecen en la salida impresa, no en el orden en que se escriben en el flujo de entrada.

Los objetos de presentación como las claves y las armaduras de cambio de tonalidad están, más bien, causadas por propiedades que cambian, y no por eventos. Otros, como las líneas divisorias y los números de compás, son una consecuencia directa del paso del tiempo. Por esta razón, las notas al pie sobre tales objetos tienen que basarse en su tiempo musical. Las notas

al pie basadas en tiempo so preferibles también cuando se marcan cosas como las plicas y las barras de corchea en *acordes*: dado que estas funcionalidades de cada acorde individual están asignadas normalmente a *un solo* evento dentro del acorde, sería imprudente descansar en una elección particular.

En las notas al pie basadas en tiempo, siempre se debe especificar explícitamente el objeto de presentación en cuestión, y si el grob se crea dentro de un contexto distinto al de nivel más bajo, debe especificarse el contexto apropiado.
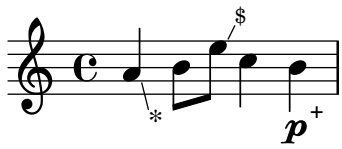
```
\book {
  \header { tagline = ##f }
  \relative c'' {
    r1 |
    \footnote #'(-0.5 . -1) "Cambio de compás" Staff.TimeSignature
    \time 3/4
    \footnote #'(1 . -1) "Plica de acorde" Stem
    <c e g>4 q q
    \footnote #'(-0.5 . 1) "Línea divisoria" Staff.BarLine
    q q
    \footnote #'(0.5 . -1) "Cambio de tonalidad" Staff.KeySignature
    \key c\minor
    q
  }
}
```



[1]Cambio de compás
[2]Plica de acorde
[3]Línea divisoria
[4]Cambio de tonalidad

Se pueden usar marcas personalizadas como alternativas a las marcas numéricas, y se puede suprimir la línea de anotación que une el objeto marcado con su marca:

```
\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote "*" #'(0.5 . -2) \markup { \italic "* La primera nota" } a'4
    b8
    \footnote \markup { \super "$" } #'(0.5 . 1)
      \markup { \super "$" \italic " La segunda nota" } e
    c4
    \once \override Score.FootnoteItem.annotation-line = ##f
    b-\footnote \markup \tiny "+" #'(0.1 . 0.1)
      \markup { \super "+" \italic " Editorial" } \p
  }
}
```



* *La primera nota*
$ *La segunda nota*
+ *Editorial*

Se muestran más ejemplos de marcas personalizadas en [Notas al pie en texto independiente], página 484.

Notas al pie en texto independiente

Se utilizan dentro de un elemento de marcado fuera de cualquier expresión musical. No se traza una líne hasta el punto de referencia: la marca simplemente sigue al elemento de marcado que se referencia. Se pueden insertar las marcas automáticamente, en cuyo caso son numéricas. De forma alternativa, se pueden aportar manualmente marcas personalizadas.

Las notas al pie para texto independiente con marcas automáticas y personalizadas se crean de distintas formas.

Notas al pie en texto independiente con marcas automáticas

La sintaxis de una nota al pue sobre texto independiente con marcas automáticas es

```
\markup { ... \auto-footnote texto nota ... }
```

Los elementos son:
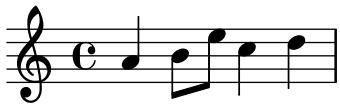
texto es el elemento de marcado o cadena de caracteres que se desea anotar.

nota es el elemento de marcado o cadena de caracteres que especifica el texto de la nota al pie que utilizar en la parte baja de la página.

Por ejemplo:

```
\book {
  \header { tagline = ##f }
  \markup {
    "A simple"
    \auto-footnote "tune" \italic " By me"
    "is shown below.  It is a"
    \auto-footnote "recent" \italic " Aug 2012"
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}
```

A simple tune[1] is shown below. It is a recent[2] composition.



[1] *By me*
[2] *Aug 2012*

Notas al pie sobre texto independiente con marcas personalizadas

La sintaxis de una nota al pie sobre texto independiente con marcas personalizadas es

```
\markup { ... \footnote marca nota ... }
```

Los elementos son:
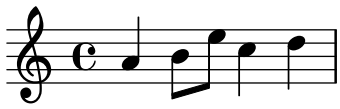
marca es un elemento de marcado o cadena de caracteres que especifica la marca que se usa para señalar el punto de referencia. Observe que esta marca *no* se inserta automáticamente antes de la propia nota al pie.

nota es el elemento de marcado o cadena de caracteres que especifica el texto de la nota que aparece en la parte baja de la página, precedida por la *marca* especificada.

Se puede usar como marca un carácter fácil de teclear, como * ó +, según se muestra en [Notas al pie dentro de expresiones musicales], página 478. De forma alternativa, se pueden usar alias en ASCII (véase [Alias de ASCII], página 499):

```
\book {
  \paper { #(include-special-characters) }
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote "*" \italic "* By me"
    "is shown below.  It is a recent"
    \footnote \super &dagger; \concat {
      \super &dagger; \italic " Aug 2012"
    }
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}
```

A simple tune * is shown below. It is a recent † composition.
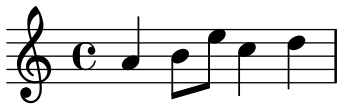


* *By me*
† *Aug 2012*

También pueden utilizarse códigos de caracteres Unicode para especificar las marcas (véase [Unicode], página 498):

```
\book {
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote \super \char##x00a7 \concat {
      \super \char##x00a7 \italic " By me"
    }
    "is shown below.  It is a recent"
    \footnote \super \char##x00b6 \concat {
      \super \char##x00b6 \italic " Aug 2012"
    }
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}
```

A simple tune § is shown below. It is a recent ¶ composition.



§ *By me*
¶ *Aug 2012*

Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” in *Manual de Aprendizaje*.

Referencia de la notación: ⟨undefined⟩ [ASCII aliases], página ⟨undefined⟩, [Globos de ayuda], página 221, ⟨undefined⟩ [List of special characters], página ⟨undefined⟩, [Indicaciones de texto], página 229, [Guiones de texto], página 226, [Unicode], página 498.

Referencia de funcionamiento interno: Sección “FootnoteEvent” in *Referencia de Funcionamiento Interno*, Sección “FootnoteItem” in *Referencia de Funcionamiento Interno*, Sección “FootnoteSpanner” in *Referencia de Funcionamiento Interno*, Sección “Footnote_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Varias notas al pie en la misma página solo pueden ir apiladas unas sobre otras; no se pueden imprimir en la misma línea.

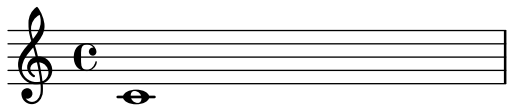
Las notas al pie no se pueden adjuntar a `MultiMeasureRests` (silencios multicompás), a barras de corchea automáticas ni a la letra de las canciones.

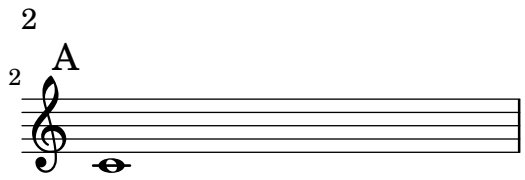
Las notas al pie pueden colisionar con las pautas, objetos `\markup`, otras marcas de notas al pie y líneas de anotación.

3.2.4 Referencia a números de página

Un lugar determinado de una partitura puede señalizarse usando la instrucción `\label` (etiqueta), ya sea en el nivel superior o dentro de la música. Podemos referirnos a esta etiqueta más tarde dentro de un elemento de marcado, para obtener el número de la página en la que se sitúa el punto señalizado, usando la instrucción de marcado `\page-ref`.

```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'1
    }
  }
  \markup { La primera partitura comienza en la página \page-ref #'firstScore "0" "?" }
  \markup { La marca A está en la página \page-ref #'markA "0" "?" }
}
```





La primera partitura comienza en la página 1
La marca A está en la página 2

La instrucción de marcado `\page-ref` toma tres argumentos:

1. la etiqueta, un símbolo de scheme, p.ej. `#'firstScore`;
2. un elemento de marcado que se usará como medidor para estimar las dimensiones del marcado;
3. un elemento de marcado que se utilizará en sustitución del número de página si la etiqueta es desconocida.

El motivo de que se necesite un medidor es que en el momento en que se están interpretando los marcados, los saltos de página aún no se han producido y por tanto los números de página no se conocen todavía. Para sortear este inconveniente, la interpretación real del marcado se retrasa hasta un momento posterior; sin embargo, las dimensiones del marcado se tienen que conocer de antemano, así que se usa el medidor para decidir estas dimensiones. Si el libro tiene entre 10 y 99 páginas, el medidor puede ser `"00"`, es decir, un número de dos dígitos.

Instrucciones predefinidas

`\label`, `\page-ref`.

3.2.5 Índice general

Se puede insertar un índice general o tabla de contenidos utilizando la instrucción `\markuplist \table-of-contents`. Los elementos que deben aparecer en la tabla de contenidos se introducen con la instrucción `\tocItem`, que se puede usar en el nivel más alto de la jerarquía del código, o dentro de una expresión musical.

```
\markuplist \table-of-contents
\pageBreak

\tocItem \markup "Primera partitura"
\score {
  {
    c'4  % ...
    \tocItem \markup "Un punto concreto dentro de la primera partitura"
    d'4  % ...
  }
}

\tocItem \markup "Segunda partitura"
\score {
  {
    e'4 % ...
  }
}
```

Los elementos de marcado que se usan para dar formato al índice general se encuentran definidos dentro del bloque `\paper`. Los elementos predeterminados son `tocTitleMarkup`, para

dar formato al título de la tabla, y `tocItemMarkup`, para aplicar formato a los elementos del índice, que consisten en el título del elemento y el número de página. Estas variables se pueden cambiar por parte del usuario:

```
\paper {
  %% Traducir el título del índice al español:
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Índice general" \null }
    \hspace #1
  }
  %% usar una fuente mayor
  tocItemMarkup = \markup \large \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
  }
}
```

Observe la forma en que nos referimos al texto y al número de página del elemento de índice, dentro de la definición `tocItemMarkup`.

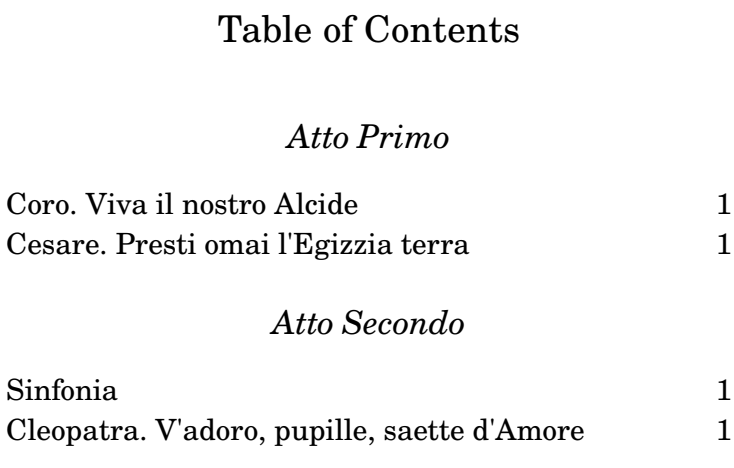
También se pueden definir nuevas instrucciones y elementos de marcado para crear índices generales más elaborados:

- en primer lugar, defina una variable de marcado nueva dentro del bloque `\paper`
- luego defina una función de música cuyo propósito es añadir un elemento al índice general utilizando esta variable de marcado del bloque `\paper`.

En el ejemplo siguiente se define un estilo nuevo para introducir los nombres de los actos de una ópera en el índice general:

```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (parser location text) (markup?)
   (add-toc-item! 'tocActMarkup text))
```



Se pueden añadir puntos de relleno entre un elemento y su número de página:

```
\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
  \markuplist \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
```

Table of Contents

Allegro 1
Largo 1

Véase también

Archivos de inicio: ‘`ly/toc-init.ly`’.

Instrucciones predefinidas

`\table-of-contents`, `\tocItem`.

3.3 Trabajar sobre los archivos de entrada

3.3.1 Inclusión de archivos de LilyPond

Un proyecto grande se puede dividir en varios archivos. Para referirse a otro archivo, utilice

```
\include "otroarchivo.ly"
```

La línea `\include "otroarchivo.ly"` equivale a pegar todo el contenido de ‘`otroarchivo.ly`’ en el archivo actual en el lugar en que aparece el `\include`. Por ejemplo, en un proyecto grande podríamos querer archivos distintos para cada parte instrumental y crear un archivo de “partitura completa” que reúne los archivos individuales de los instrumentos. Normalmente el archivo incluido define un cierto número de variables que a partir de entonces quedan disponibles para poderlas utilizar en el archivo de la partitura completa. En los archivos incluidos se pueden marcar las secciones etiquetadas para ayudar a hacerlas utilizables en distintos lugares de la partitura, véase Sección 3.3.2 [Distintas ediciones a partir de una misma fuente], página 492.

Los archivos que están en el directorio de trabajo actual se pueden referenciar simplemente especificando el nombre después de la instrucción `\include`. Los archivos en otros lugares se pueden incluir proporcionando una referencia de ruta completa o una ruta relativa (pero utilice la barra inclinada normal del UNIX, /, no la barra invertida de DOS/Windows, \, como separador de directorio). Por ejemplo, si ‘`material.ly`’ está situado un directorio por encima del directorio de trabajo actual, utilice

```
\include "../material.ly"
```

o si los archivos de las partes orquestales incluidas están todos situados en un subdirectorio llamado ‘`partes`’ dentro del directorio actual, use

```
\include "partes/VI.ly"
\include "partes/VII.ly"
```

```
... etc.
```

Los archivos de inclusión también pueden contener enunciados `\include` a su vez. De forma predeterminada, estos enunciados `\include` de segundo nivel no se interpretan hasta que se incorporan al archivo principal, por lo que los nombres de archivo que especifican deben ser relativos al directorio que contiene el archivo principal, no al directorio que contiene el archivo incluido. Sin embargo, este comportamiento se puede modificar globalmente pasando la opción '`-drelative-includes`' en la línea de órdenes (o mediante la adición de `#(ly:set-option 'relative-includes #t)` al principio del archivo principal de entrada).

Cuando `relative-includes` está establecido al valor `#t`, la ruta de cada instrucción `\include` se considera relativa al archivo que contiene dicha instrucción. Este comportamiento es el recomendado y se convertirá en el comportamiento predeterminado en una versión futura de lilypond.

Tanto los archivos relativos al directorio principal como los archivos relativos a algún otro directorio se pueden `\include`ar mediante el establecimiento de `relative-includes` al valor `#t` o a `#f` en los lugares apropiados dentro de los archivos. Por ejemplo, si se ha creado una biblioteca genérica, libA, que a su vez utiliza sub-archivos que se `\include`an por medio del archivo de entrada de dicha biblioteca, esas instrucciones `\include` tendrán que venir precedidas por `#(ly:set-option #relative-includes #t)` de manera que se interpreten correctamente al construir el archivo de entrada `.ly` principal, de esta forma:

```
libA/
  libA.ly
  A1.ly
  A2.ly
  ...
```

entonces el archivo de entrada, `libA.ly`, contiene

```
#(ly:set-option 'relative-includes #t)
\include "A1.ly"
\include "A2.ly"
...
% vuelta al ajuste predeterminado
#(ly:set-option 'relative-includes #f)
```

Cualquier archivo '`.ly`' puede entonces incluir la biblioteca completa, simplemente con

```
\include "~/libA/libA.ly"
```

Se pueden usar estructuras de archivo más complejas instando los cambios en los lugares adecuados.

También se pueden incluir archivos de un directorio que está en una ruta de búsqueda especificada como opción al invocar a LilyPond desde la línea de órdenes. Los archivos incluidos se especifican usando solamente su nombre de archivo. Por ejemplo, para compilar mediante este método un archivo '`principal.ly`' que incluye archivos situados en un subdirectorio llamado '`partes`', cambie al directorio que contiene '`principal.ly`' e introduzca

```
lilypond --include=partes principal.ly
```

y en '`principal.ly`' escriba

```
\include "VI.ly"
\include "VII.ly"
... etc.
```

Los archivos que se incluyen en muchas partituras se pueden poner en el directorio de LilyPond '`../ly`' (la localización de este directorio depende de la instalación: véase Sección "Otras

fuentes de información" in *Manual de Aprendizaje*). Estos archivos se pueden incluir sencillamente por su nombre en un enunciado `\include`. De esta forma se incluyen los archivos dependientes del idioma como '`espanol.ly`'.

LilyPond incluye de forma predeterminada ciertos archivos cuando se inicia el programa. Estas inclusiones no son visibles para el usuario, pero los archivos se pueden identificar ejecutando `lilypond --verbose` desde la línea de órdenes. Esto presentará una lista de rutas y archivos que utiliza LilyPond, junto a muchas otras indicaciones. De forma alternativa, pueden verse explicaciones sobre los archivos más importantes de este conjunto en Sección "Otras fuentes de información" in *Manual de Aprendizaje*. Estos archivos se pueden editar, pero los cambios realizados sobre ellos se perderán al instalar una nueva versión de LilyPond.

Pueden verse ejemplos sencillos sobre el uso de `\include` en Sección "Partituras y particellas" in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Otras fuentes de información" in *Manual de Aprendizaje*, Sección "Partituras y particellas" in *Manual de Aprendizaje*.

Advertencias y problemas conocidos

Si un archivo incluido recibe un nombre igual al de uno de los archivos de instalación de LilyPond, el archivo que tiene prioridad es el de los archivos de instalación de LilyPond.

3.3.2 Distintas ediciones a partir de una misma fuente

Se pueden usar varios métodos para generar la producción de distintas versiones de una partitura a partir de la misma fuente musical. Las variables son quizá el más útil para combinar secciones extensas de música y/o anotaciones. Las etiquetas son más útiles para seleccionar una sección de entre varias secciones de música alternativas, más cortas, y se pueden usar también para dividir piezas de música en distintos trozos.
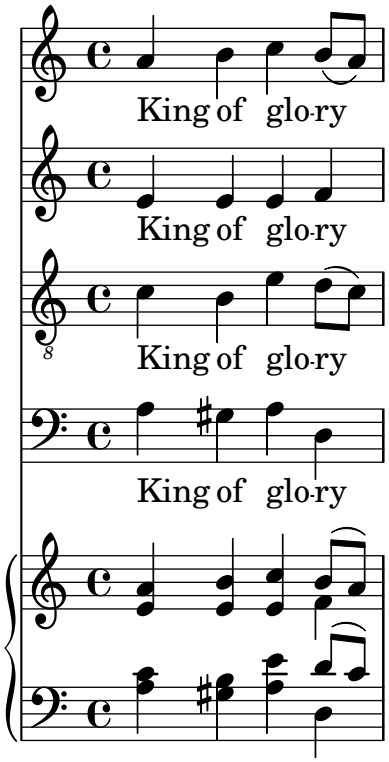
Cualquiera que sea el método utilizado, la separación de la notación de la estructura de la partitura hará que sea más fácil cambiar la estructura dejando intacta la notación.

Uso de las variables

Si las secciones de la música se definen dentro de variables, se pueden reutilizar en distintas partes de la partitura, véase Sección "Organizar las piezas mediante variables" in *Manual de Aprendizaje*. Por ejemplo, una partitura vocal *a cappella* con frecuencia incluye una reducción de piano de las partes para ensayar que es idéntica a la música vocal, por lo que ésta sólo se tiene que escribir una vez. Se puede combinar sobre un solo pentagrama la música de dos variables, véase [Combinación automática de las partes], página 175. He aquí un ejemplo:

```
sopranoMusic = \relative c'' { a4 b c b8( a) }
altoMusic = \relative g' { e4 e e f }
tenorMusic = \relative c' { c4 b e d8( c) }
bassMusic = \relative c' { a4 gis a d, }
allLyrics = \lyricmode {King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
```

```
  }
  \new Lyrics \allLyrics
  \new Staff = "Bass" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "RH" {
      \set Staff.printPartCombineTexts = ##f
      \partcombine
      \sopranoMusic
      \altoMusic
    }
    \new Staff = "LH" {
      \set Staff.printPartCombineTexts = ##f
      \clef "bass"
      \partcombine
      \tenorMusic
      \bassMusic
    }
  >>
>>
```



Se pueden producir partituras distintas que presenten sólo las partes vocales o sólo la parte de piano, cambiando solamente los enunciados estructurales, sin modificar la notación musical.

Para partituras extensas, las definiciones de variable se pueden poner en archivos separados que se incluyen más tarde, véase Sección 3.3.1 [Inclusión de archivos de LilyPond], página 490.

Uso de etiquetas

La instrucción `\tag #'parteA` marca una expresión musical con el nombre *parteA*. Las expresiones etiquetadas de esta manera se pueden seleccionar o filtrar más tarde por su nombre, usando bien `\keepWithTag #'nombre` o bien `\removeWithTag #'nombre`. El resultado de la aplicación de estos filtros a la música etiquetada es como sigue:

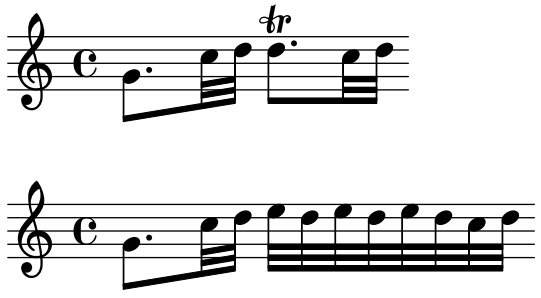
Filtro	**Resultado**
Música etiquetada precedida de `\keepWithTag #'nombre` o de `\keepWithTag #'(nombre1 nombre2...)`	Se incluye la música no etiquetada y la música etiquetada con el nombre o nombres de etiqueta dados; se excluye la música etiquetada con cualquier otro nombre de etiqueta.
Música etiquetada precedida de `\removeWithTag #'nombre` o `\removeWithTag #'(nombre1 nombre2...)`	Se incluye la música no etiquetada y la música que no está etiquetada con ninguno de los nombres de etiqueta dados; se excluye la música etiquetada con los nombres de etiqueta dados.
Música etiquetada no precedida de `\keepWithTag` ni de `\removeWithTag`	Se incluye toda la música etiquetada y no etiquetada.

Los argumentos de las instrucciones `\tag`, `\keepWithTag` y `\removeWithTag` deben ser un símbolo (como `#'partitura` o `#'parte`), seguido de una expresión musical.

En el ejemplo siguiente vemos dos versiones de una pieza musical, una que muestra trinos con la notación usual y otra con los trinos expandidos explícitamente:

```
music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
 }

\score {
  \keepWithTag #'trills \music
}
\score {
  \keepWithTag #'expand \music
}
```

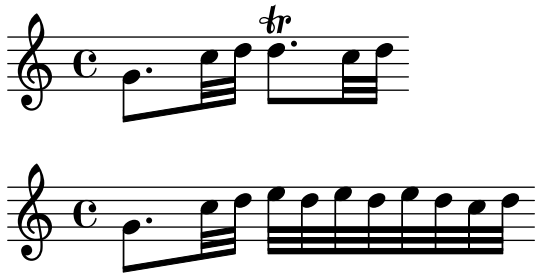


De forma alternativa, a veces es más fácil excluir secciones de música:

```
music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand {\repeat unfold 3 { e32 d } }
```

```
  c32 d
 }

\score {
  \removeWithTag #'expand
  \music
}
\score {
  \removeWithTag #'trills
  \music
}
```



El filtrado con etiquetas se puede aplicar a articulaciones, textos, etc. anteponiendo

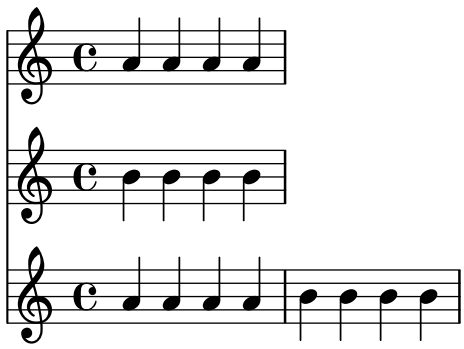
`-\tag #'`*`mi-etiqueta`*

a una articulación. Por ejemplo, esto define una nota con una indicación opcional de digitación y una nota con una anotación condicional:

```
c1-\tag #'finger ^4
c1-\tag #'warn ^"¡Ojo!"
```
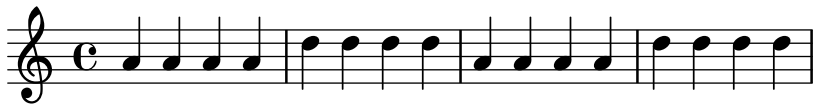
Se pueden poner varias etiquetas sobre expresiones con varias entradas de `\tag`, o combinando varias etiquetas en una lista de símbolos:

```
music = \relative c'' {
  \tag #'a \tag #'both { a4 a a a }
  \tag #'(b both) { b4 b b b }
}
<<
\keepWithTag #'a \music
\keepWithTag #'b \music
\keepWithTag #'both \music
>>
```



Se pueden aplicar varios filtros `\removeWithTag` a una sola expresión musical para quitar varias secciones etiquetadas con nombres distintos. De forma alternativa, podemos usar una sola instrucción `\removeWithTag` con una lista de etiquetas.

```
music = \relative c'' {
  \tag #'A { a4 a a a }
  \tag #'B { b4 b b b }
  \tag #'C { c4 c c c }
  \tag #'D { d4 d d d }
}
\new Voice {
  \removeWithTag #'B
  \removeWithTag #'C
  \music
  \removeWithTag #'(B C)
  \music
}
```
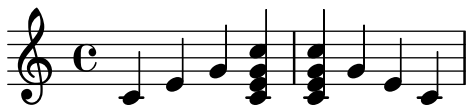


Dos o más filtros `\keepWithTag` aplicados a una sola expresión musical producen la eliminación de *todas* las secciones etiquetadas, porque el primer filtro quita todas las secciones etiquetadas excepto la que se nombra, y el segundo filtro elimina incluso esa sección etiquetada. Por lo general, querrá usar una sola instrucción `\keepWithTag` con una lista de varias etiquetas: de esta forma solamente se suprimirán las secciones etiquetadas que no aparezcan en *ninguna* de las etiquetas.

A veces podemos desear dividir la música en un determinado lugar dentro de una expresión musical existente. Podemos usar `\pushToTag` y `\appendToTag` para añadir material delante o al final de los `elements` de una construcción musical existente. No toda construcción musical tiene la parte `elements`, pero las músicas secuenciaes y simultáneas son dos apuestas seguras:

```
test = { \tag #'here { \tag #'here <<c''>> } }

{
  \pushToTag #'here c'
  \pushToTag #'here e'
  \pushToTag #'here g' \test
  \appendToTag #'here c'
  \appendToTag #'here e'
  \appendToTag #'here g' \test
}
```



Las dos instrucciones admiten una etiqueta, el material que dividir en cada ocurrencia de la etiqueta, y la expresión etiquetada. Las instrucciones aseguran que se copia todo lo que cambian, de forma que el código original `\test` retiene su significado.

Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” in *Manual de Aprendizaje*.

Referencia de la notación: [Combinación automática de las partes], página 175, Sección 3.3.1 [Inclusión de archivos de LilyPond], página 490.

Advertencias y problemas conocidos

Al llamar a `\relative` sobre una expresión musical obtenida mediante el filtrado de música con las instrucciones `\keepWithTag` o `\removeWithTag`, podría ocurrir que cambiaran las relaciones de octava, dado que solo se tendrán en cuenta las alturas de las notas que efectivamente permanecen en la expresión filtrada. La aplicación de `\relative` en primer lugar, antes de `\keepWithTag` o de `\removeWithTag`, evita este peligro porque en este caso `\relative` actúa sobre todas las notas tal y como se introducen.

Uso de ajustes globales

Se pueden incluir ajustes globales a partir de un archivo distinto:

```
lilypond -dinclude-settings=MIS_AJUSTES.ly MI_PARTITURA.ly
```

En archivos diferentes se pueden almacenar grupos de ajustes como el tamaño de la página o las fuentes tipográficas. Ello permite hacer diferentes ediciones de la mism a partitura así como aplicar ajustes estándar a muchas partituras, simplemente por medio de la especificación del archivo de ajustes adecuado.

Esta técnica también funciona bien con el uso de hojas de estilo, como se estudia en Sección "Hojas de estilo" in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Organizar las piezas mediante variables" in *Manual de Aprendizaje*, Sección "Hojas de estilo" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 3.3.1 [Inclusión de archivos de LilyPond], página 490.

3.3.3 Caracteres especiales

Codificación del texto

LilyPond usa el conjunto de caracteres definido por el consorcio Unicode y la norma ISO/IEC 10646. Define un nombre único y un código para los conjuntos de caracteres que se utilizan en prácticamente todos los idiomas modernos y también en muchos otros. Unicode se puede implementar utilizando varios esquemas de codificación distintos. LilyPond usa la codificación UTF-8 (UTF son las siglas de *Unicode Transformation Format*, o formato de transformación de Unicode) que representa todos los caracteres comunes de Latin en un solo byte, y representa otros caracteres usando un formato de longitud variable de hasta cuatro bytes.
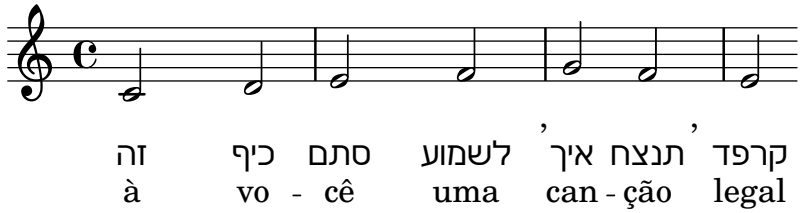
El aspecto visual real de los caracteres viene determinado por los glifos que se definen en las fuentes tipográficas concretas que se tengan disponibles: una fuente tipográfica define la asignación de un subconjunto de los códigos de Unicode a glifos. LilyPond usa la biblioteca Pango para representar y disponer tipográficamente textos multilingües.

LilyPond no realiza ninguna conversión en la codificación de la entrada. Esto significa que cualquier text, ya sea el título, la letra de la canción o una instrucción musical que contenga caracteres distintos a los del conjunto ASCII, se deben codificar en UTF-8. La forma más fácil de escribir dicho texto es utilizar un editor preparado para Unicode y guardar el archivo con la codificación UTF-8. Casi todos los editores modernos populares contemplan el UTF-8, por ejemplo lo hacen vim, Emacs, jEdit, y GEdit. Todos los sistemas MS Windows posteriores a NT usan Unicode como codificación de caracteres nativa, de manera que incluso el accesorio Bloc de Notas (Notepad) puede editar y guardar un archivo en el formato UTF-8. Una alternativa más funcional para Windows es BabelPad.

Si un archivo de entrada que contiene un carácter que no es ASCII, no se guarda en el formato UTF-8, se genera el mensaje de error

```
FT_Get_Glyph_Name () error: argumento inválido
```

He aquí un ejemplo que muestra texto cirílico, hebreo y portugués:



Unicode

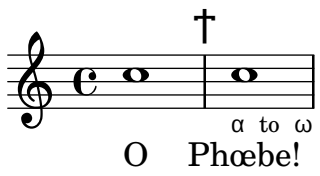
Para introducir un carácter aislado para el que se conoce el punto de código Unicode pero no está disponible en el editor que se está usando, use `\char ##xhhhh` o bien `\char #dddd` dentro de un bloque `\markup`, donde `hhhh` es el código hexadecimal del carácter en cuestión y `dddd` es su valor decimal correspondiente. Pueden omitirse los ceros iniciales, pero es costumbre indicar los cuatro caracteres en la representación hexadecimal (observe que *no* debe utilizarse la codificación UTF-8 del punto de código Unicode después de `\char`, pues la codificación UTF-8 contiene bits adicionales que indican el número de octetos). Hay tablas de códigos Unicode y un índice de nombres de caracteres que da el punto de código en hexadecimal para cualquier carácter en el portal del Consorcio Unicode, http://www.unicode.org/.

Por ejemplo, tanto `\char ##x03BE` como `\char #958` insertan el carácter Unicode U+03BE, que tiene el nombre Unicode “Letra griega Xi pequeña”.

Se puede escribir de esta forma cualquier punto de código Unicode, y si todos los caracteres especiales se escriben en este formato no es necesario guardar el archivo de entrada en formato UTF-8. Por supuesto, debe estar instalada y estar disponible para LilyPond una fuente tipográfica que contenga codificados todos estos caracteres.

El ejemplo siguiente muestra valores hexadecimales Unicode que se usan en cuatro lugares: en una llamada de ensayo, como texto de articulación, en la letra y como texto independiente bajo la partitura:

```
\score {
  \relative c'' {
    c1 \mark \markup { \char ##x03EE }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { O \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2012" \char ##x00A9 }
```



Copyright 2008--2012 ©

Para escribir el símbolo de copyright en la nota de créditos, utilice:

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

Alias de ASCII

Se puede incluir una lista de alias de ASCII para caracteres especiales:

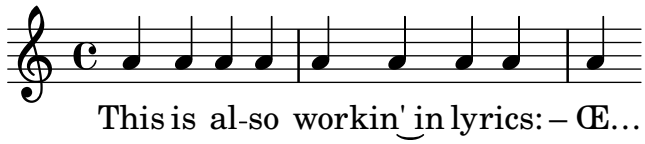
```
\paper {
  #(include-special-characters)
}

\markup "&flqq; &ndash; &OE;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    This is al -- so wor -- kin'~in ly -- rics: &ndash;_&OE;&hellip;
  }
}

\markup \column {
  "The replacement can be disabled:"
  "&ndash; &OE; &hellip;"
  \override #'(replacement-alist . ()) "&ndash; &OE; &hellip;"
}
```

« – Œuvre incomplète… »



The replacement can be disabled:
– Œ …
– &OE; …

También podemos hacer nuestros propios alias, ya sea de forma global:

```
\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}
\markup "A 100 dpi."
```

A hundred dots per inch.

o local:

```
\markup \replace #'(("100" . "hundred")
                    ("dpi" . "dots per inch")) "A 100 dpi."
```

A hundred dots per inch.

Véase también

Referencia de la notación: Sección A.12 [Lista de caracteres especiales], página 723.

Archivos instalados: ‘`ly/text-replacements.ly`’.

3.4 Controlar la salida

3.4.1 Extracción de fragmentos de música

Es posible citar pequeños fragmentos de una partitura grande directamente a partir de la salida. Puede compararse con la separación de una pieza de una partitura, recortándola con tijeras.

Esto se hace definiendo los compases que se deben recortar. Por ejemplo, la inclusión de la siguiente definición

```
\layout {
  clip-regions
  = #(list
      (cons
       (make-rhythmic-location 5 1 2)
       (make-rhythmic-location 7 3 4)))
}
```

extrae un fragmento empezando por la mitad del quinto compás y finalizando en el séptimo compás. El significado de `5 1 2` es: después de una blanca (1/2) en el compás 5, y el de `7 3 4` después de tres negras del compás 7.

Se pueden definir más zonas de recorte añadiendo más parejas de definición de tiempo rhythmic-location a la lista.

Para poder utilizar esta funcionalidad, LilyPond se debe invocar con la opción '`-dclip-systems`'. Los recortes aparecen en la salida como archivos EPS, y se convierten en PDF y PNG si estos formatos también están activados.

Para ver más información sobre los formatos de salida, consulte Sección "Invocar a LilyPond" in *Utilización del Programa*.

3.4.2 Saltar la música corregida

Al escribir o copiar música, normalmente sólo es interesante de ver y corregir la música cercana al final (donde estamos añadiendo las notas). Para acelerar este proceso de corrección, es posible saltar la composición tipográfica de todos excepto unos pocos de los últimos compases. Esto se consigue poniendo

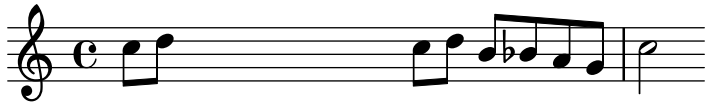
```
showLastLength = R1*5
\score { ... }
```

en nuestro archivo fuente. Esto hará que se tracen sólo los últimos cinco compases (suponiendo un compás de 4/4) de cada una de las partituras `\score` del archivo de entrada. Para piezas largas, el tipografiado de únicamente una parte pequeña es con frecuencia un orden de magnitud más rápido que el de la obra completa. Si estamos trabajando sobre el principio de una partitura que ya hemos tipografiado (p.ej., para añadir una parte instrumental nueva), también puede ser útil la propiedad `showFirstLength`.

Este mecanismo de pasar por alto partes de un partitura se puede controlar con un grano más fino mediante la propiedad `Score.skipTypesetting`. Cuando su valor está establecido, no se lleva a cabo ningún tipografiado en absoluto.

Esta propiedad se usa también para controlar la salida al archivo MIDI. Observe que pasa por alto todos los eventos, incluidos el tempo y los cambios de instrumento. Está usted advertido.

```
c8 d
\set Score.skipTypesetting = ##t
e8 e e e e e e e
\set Score.skipTypesetting = ##f
c8 d b bes a g c2
```



En música polifónica, `Score.skipTypesetting` afecta a todas las voces y pentagramas, ahorrando más tiempo incluso.

3.4.3 Formatos de salida alternativos

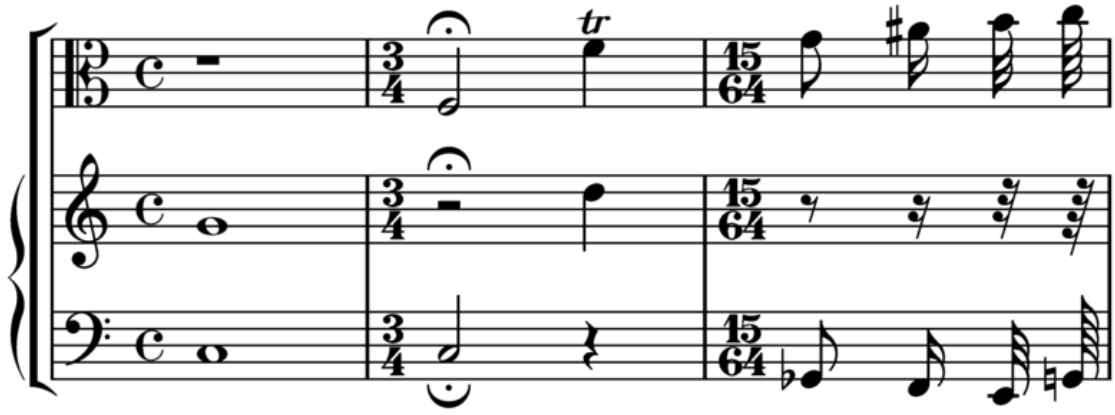
Los formatos de salida predeterminados para la partitura impresa son el formato de documento portátil (PDF) y PostScript (PS). Los formatos de salida Gráficos de vector escalables (SVG), PostScript encapsulado (EPS) y Gráficos de red portátiles (PNG) también están disponibles a través de opciones de la línea de órdenes, véase Sección “Opciones básicas de la línea de órdenes para LilyPond” in *Utilización del Programa*.

3.4.4 Sustituir la tipografía de la notación

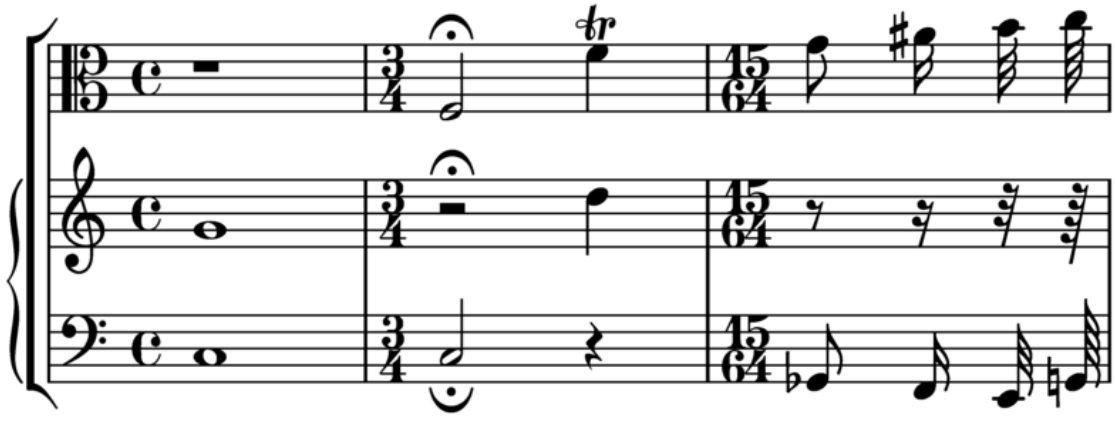
Gonville es una alternativa a la tipografía Feta que se utiliza en LilyPond y se puede descargar de:

`http://www.chiark.greenend.org.uk/~sgtatham/gonville/`

Aquí presentamos algunos compases de muestra tipografiados con la fuente Gonville:



Aquí hay unos compases de muestra tipografiados en la fuente Feta de LilyPond:



Instrucciones de instalación para MacOS

Descargue y extraiga el archivo zip. Copie la carpeta `lilyfonts` a ‘*SHARE_DIR*`/lilypond/current`’; para más información, consulte Sección “Otras fuentes de información” in *Manual de Aprendizaje*. Renombre la carpeta `fonts` existente a `fonts_orig` y la carpeta `lilyfonts` a `fonts`. Para volver a la fuente Feta, invierta el proceso.

Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” in *Manual de Aprendizaje*.

Advertencias y problemas conocidos

Gonville no se puede usar para tipografiar notación de ‘Música Antigua’ y es probable que cualquier nuevo glifo en versiones posteriores de LilyPond no existan en la familia tipográfica Gonville. Diríjase a la página web del autor para obtener más información sobre éste y otros asuntos, entre ellos el régimen de licencias de Gonville.

3.5 Salida MIDI

El MIDI (Musical Instrument Digital Interface, Interfase Digital para Instrumentos Musicales) es un estándar para interconectar y controlar instrumentos musicales electrónicos. Un archivo o secuencia MIDI es una serie de notas dentro de un conjunto de pistas. No es un archivo de sonidos reales; se necesita un programa reproductor especial para traducir la serie de notas en sonidos de verdad.

Cualquier música se puede convertir a archivos MIDI, de manera que podamos escuchar lo que hayamos introducido. Esto es muy conveniente para comprobar la corrección de la música; las octavas equivocadas o las alteraciones erróneas se ponen de relieve muy claramente al escuchar la salida MIDI.

La salida estándar de MIDI es un poco primitiva; de forma opcional, se encuentra disponible una salida MIDI mejorada y mucho más realista, por medio de Sección 3.5.7 [El script Articulate], página 511.

La salida MIDI reserva un canal por cada pentagrama, y reserva el canal 10 para la percusión. Sólo hay 16 canales MIDI por cada dispositivo, por lo que si la partitura tiene más de 15 pautas, los canales MIDI se reutilizarán.

3.5.1 Crear archivos MIDI

Para crear un MIDI a partir de un archivo de LilyPond, inserte un bloque `\midi` dentro de un bloque `\score`;

```
\score {
  ...música...
  \layout { }
  \midi { }
}
```

Si hay *solamente* un bloque `\midi` dentro de un `\score` (es decir, sin ningún bloque `\layout`), entonces *solamente* se producirá MIDI. No se produce ninguna salida impresa de notación musical.

```
\score {
  ...música...
  \midi { }
}
```

Se traducen correctamente a la salida MIDI la altura y duración de las notas, las ligaduras de unión, matices y cambios de tempo. Las ‘marcas’ dinámicas se traducen a niveles de volumen con una ‘fracción fija’ del rango de volumen MIDI disponible; Los crescendi y decrescendi hacen que el volumen varíe linealmente entre sus dos extremos.

Se reflejan en la salida de MIDI todas las indicaciones de `\tempo`, incluidos todos los cambios de tempo siguientes, especificados dentro de la notación musical.

Por lo general basta con dejar vacío el bloque `\midi`, pero puede contener disposiciones distintas de los contextos, definiciones de contexto nuevas o código que establezca valores para

propiedades. En el ejemplo que aparece a continuación se fija el tempo a 72 negras por minuto, pero *solamente* para la reproducción de audio a través de MIDI.

```
\score {
  ...music...
  \midi {
    \tempo 4 = 72
  }
}
```

Observe que `\tempo` es en realidad una instrucción para establecer valores para propiedades durante la interpretación de la música, y en el contexto de las definiciones de salida, como un bloque `\midi`, se reinterpreta como si fuese una modificación de contexto.

Las definiciones de contexto siguen la misma sintaxis que las de un bloque `\layout`;

```
\score {
  ...música...
  \midi {
    \context {
      \Voice
      \remove "Dynamic_performer"
    }
  }
}
```

elimina de la salida MIDI el efecto de los matices dinámicos. Los módulos de traducción para el sonido se llaman 'performers' (intérpretes).

Fragmentos de código seleccionados
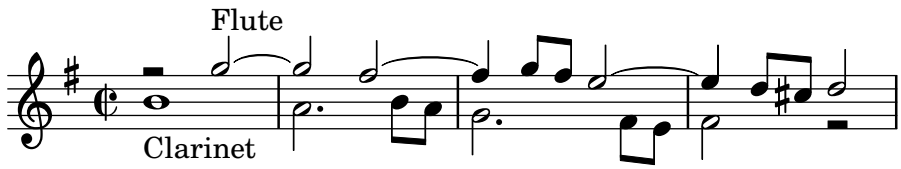
Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos sólo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
```

```
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```



Advertencias y problemas conocidos

Algunos sistemas operativos requieren una extensión de archivo *específica* para los archivos MIDI. Si se prefiere una extensión distinta, inserte la línea siguiente en el nivel superior del archivo de entrada, antes del comienzo de cualquier bloque `\book`, `\bookpart` o `\score`;

```
#(ly:set-option 'midi-extension "mid")
```

De esta manera quedará establecida la extensión predeterminada para los archivos MIDI a `.mid`.

De forma alternativa, se puede proporcionar una opción en la línea de órdenes de la consola:

```
lilypond -dmidi-extension=mid MiArchivo.ly
```

Los cambios del volumen MIDI sólo tienen lugar al principio de la nota, por lo que los crescendi y decrescendi no pueden afectar al volumen de una sola nota.

Puede que algunos propgramas reproductores de MIDI no siempre manejen correctamente los cambios de tempo en la salida MIDI.

Véase también

Archivos instalados: '`../ly/performer-init.ly`'.

Manual de aprendizaje: Sección "Otras fuentes de información" in *Manual de Aprendizaje*.

3.5.2 Instrumentos MIDI

El instrumento MIDI que se quiere utilizar se establece fijando la propiedad `Staff.midiInstrument` al nombre del instrumento. El nombre se debe elegir de entre los que están en la lista que aparece en Sección A.6 [instrumentos MIDI], página 648.

```
\new Staff {
  \set Staff.midiInstrument = #"glockenspiel"
```

```
  ...notas...
}
\new Staff \with {midiInstrument = #"cello"} {
  ...notas...
}
```

Si el instrumento elegido no coincide exactamente con uno de los instrumentos de la lista de instrumentos MIDI, se usará el instrumento Piano de Cola (`"acoustic grand"`).

3.5.3 ¿Qué sale por el MIDI?

Contemplado en el MIDI

Los siguientes elementos de notación se reflejan en la salida MIDI:

- Alturas
- Microtonos (véase [Alteraciones accidentales], página 5. La reproducción necesita un reproductor que contemple la curvatura de tono o *pitch bend*.)
- Acordes escritos como nombres de acorde
- Ritmos escritos como duraciones de las notas, incluidos los grupos de valoración especial
- Trémolos escritos sin ':[*número*]'
- Ligaduras de unión
- Matices o indicaciones de dinámica
- Crescendi y decrescendi sobre varias notas
- Cambios de tempo insertados con una marca metronómica
- Letra de las canciones

Utilizando Sección 3.5.7 [El script Articulate], página 511, a la lista anterior se añaden una serie de elementos:

- Articulaciones (ligaduras de expresión, picados, etc)
- Trinos, mordentes circulares
- Rallentando y accelerando

No contemplado en el MIDI

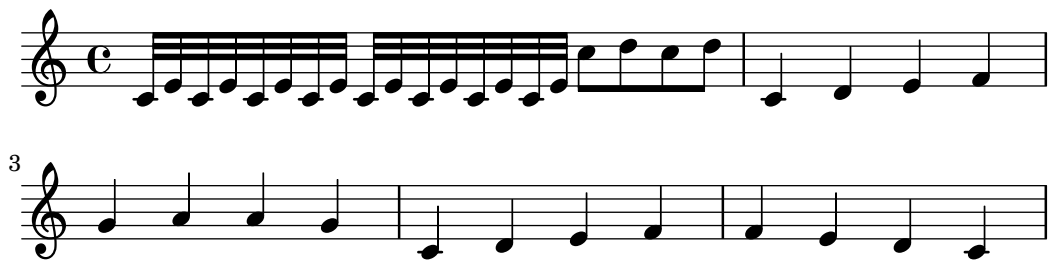
Los siguientes elementos de notación no tienen ningún efecto sobre la salida MIDI, a no ser que utilicemos Sección 3.5.7 [El script Articulate], página 511:

- Duraciones escritas como anotaciones, p.ej. el swing
- Cambios de tempo escritos como anotaciones sin marca de tempo
- Staccato y otras articulaciones y ornamentos
- Ligaduras de expresión y de fraseo
- Crescendi y decrescendi sobre una nota única
- Trémolos introducidos con ':[*número*]'
- Bajo cifrado
- Acordes microtonales

3.5.4 Repeticiones y MIDI

Con un poco de trucaje, se puede hacer que cualquier tipo de repetición esté presente en la salida MIDI. Esto se consigue mediante la aplicación de la función musical `\unfoldRepeats`. Esta función cambia todas las repeticiones a repeticiones desplegadas.

```
\unfoldRepeats {
  \repeat tremolo 8 { c'32 e' }
  \repeat percent 2 { c''8 d'' }
  \repeat volta 2 { c'4 d' e' f' }
  \alternative {
    { g' a' a' g' }
    { f' e' d' c' }
  }
}
\bar "|."
```



En partituras con varias voces, el desplegado de las repeticiones en la salida MIDI solo se produce correctamente si *todas y cada una* de las voces incluye las repeticiones de manera completa y explícita.

Al crear un archivo de partitura que use `\unfoldRepeats` para el MIDI, es necesario hacer dos bloques `\score`: uno para el MIDI (con repeticiones desplegadas) y otro para la notación impresa (con repeticiones de volta –primera y segunda vez–, tremolo –trémolo–, y percent –repetición de compases–). Por ejemplo,

```
\score {
  ...música...
  \layout { ... }
}
\score {
  \unfoldRepeats ...música...
  \midi { ... }
}
```

3.5.5 Control de los matices en el MIDI

La dinámica en el MIDI está implementada por medio del intérprete Dynamic_performer que reside de forma predeterminada dentro del contexto Voice. Es posible controlar el volumen MIDI general, el volumen relativo de los matices dinámicos y el volumen relativo de los distintos instrumentos.

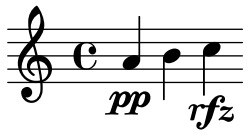
Indicaciones dinámicas

Los matices o indicaciones de dinámica se traducen a una fracción fija del rango de volumen MIDI disponible. Las fracciones predeterminadas se extienden desde 0.25 para *ppppp* hasta 0.95 para *fffff*. El conjunto de indicaciones dinámicas y las fracciones asociadas pueden verse en '`../scm/midi.scm`', véase Sección "Otras fuentes de información" in *Manual de Aprendizaje*. Este conjunto de fracciones se puede cambiar o extender proporcionando una función que toma como argumento una indicación dinámica y devuelve la fracción requerida, y estableciendo `Score.dynamicAbsoluteVolumeFunction` a esta función.

Por ejemplo, si se requiere una indicación dinámica *rinforzando*, `\rfz`, ésta no tiene ningún efecto predeterminado sobre el volumen MIDI, pues no está incluida en el conjunto predeterminado. De igual manera, si se define una indicación dinámica nueva con `make-dynamic-script`, ésta tampoco estará incluida en el conjunto predeterminado. El ejemplo siguiente muestra cómo hay que añadir el volumen MIDI para estas indicaciones dinámicas. La función Scheme establece la fracción a 0.9 si se encuentra una indicación dinámica rfz, y en caso contrario llama a la función predeterminada.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative c'' {
        a4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Como alternativa, si se necesitara redefinir la tabla completa de fracciones, sería mejor usar el procedimiento *default-dynamic-absolute-volume* que aparece en '`../scm/midi.scm`' y la tabla asociada como modelo. El último ejemplo de esta sección muestra cómo se puede hacer esto.

Volumen maestro MIDI
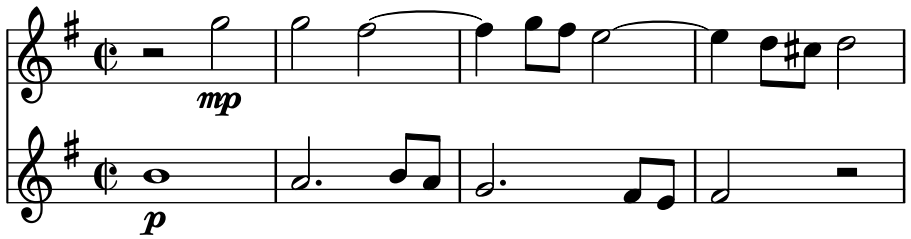
El volumen general máximo y mínimo de las indicaciones dinámicas de MIDI se controla estableciendo valores para las propiedades `midiMinimumVolume` y `midiMaximumVolume` en el nivel de `Score`. Estas propiedades tienen efecto solamente al inicio de una voz y sobre las indicaciones dinámicas. La fracción que corresponde a cada indicación dinámica se modifica con esta fórmula:

```
midiMinimumVolume + (midiMaximumVolume - midiMinimumVolume) * fracción
```

En el ejemplo siguiente, el rango de intensidad dinámica del volumen general MIDI está limitado al intervalo desde 0.2 hasta 0.5.

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis~
```

```
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout {}
  \midi {
    \tempo 2 = 72
    \context {
      \Score
      midiMinimumVolume = #0.2
      midiMaximumVolume = #0.5
    }
  }
}
```
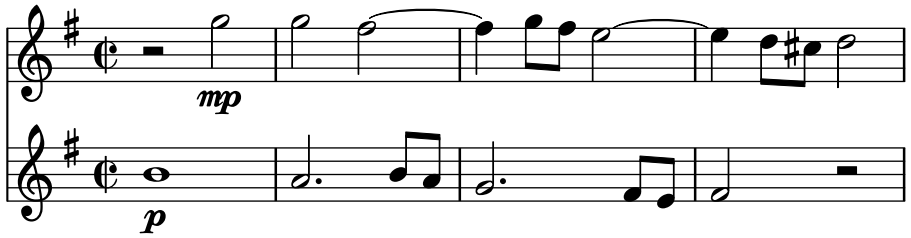


Balance entre instrumentos (i)

Si las propiedades de volumen MIDI máximo y mínimo están establecidas en el contexto `Staff` se pueden controlar los volúmenes relativos de los instrumentos MIDI. Esto proporciona un ecualizador de instrumentos básico que puede realzar significativamente la calidad de la salida MIDI.

En este ejemplo se reduce el volumen del clarinete con relación al de la flauta.

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \set Staff.midiMinimumVolume = #0.7
      \set Staff.midiMaximumVolume = #0.9
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
```

```
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \set Staff.midiMinimumVolume = #0.3
      \set Staff.midiMaximumVolume = #0.6
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout {}
  \midi {
    \tempo 2 = 72
  }
}
```



Balance entre instrumentos (ii)

Si las propiedades de volumen MIDI máximo y mínimo no están establecidas, de forma predeterminada LilyPond aplica un pequeño grado de ecualización a algunos instrumentos. Los instrumentos y la ecualización que se aplica se muestra en la tabla *instrument-equalizer-alist* en '`../scm/midi.scm`'.

El ecualizador básico predeterminado se puede sustituir estableciendo `instrumentEqualizer` en el contexto `Score` a un procedimiento Scheme nuevo que acepte el nombre de un instrumento MIDI como único argumento y devuelva un par de fracciones que dan los volúmenes máximo y mínimo que se aplican a dicho instrumento. Esta sustitución se hace de la misma forma que mostramos para restablecer la fracción de volumen absoluto `dynamicAbsoluteVolumeFunction` al comienzo de esta sección. El ecualizador predeterminado, *default-instrument-equalizer*, en '`../scm/midi.scm`', muestra cómo se puede escribir tal procedimiento.
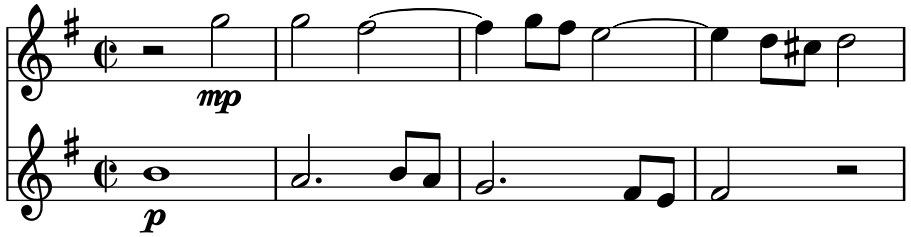
El ejemplo siguiente establece los volúmenes relativos de la flauta y del clarinete a los mismos valores que el ejemplo anterior.

```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(
      ("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))
```

```
#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \tempo 2 = 72
  }
}
```



3.5.6 Percusión en MIDI

La notación de los instrumentos de percusión se realiza normalmente en un contexto `DrumStaff` y cuando se hace de esta forma se les da salida correctamente por el canal MIDI 10, pero ciertos instrumentos de percusión de altura determinada, como el xilófono, marimba, vibráfono, timbales, etc., se tratan como instrumentos "normales" y la música para estos instrumentos se debe introducir en contextos de `Staff` normales, no en un contexto `DrumStaff`, para obtener la salida MIDI correcta.

Ciertos sonidos de altura indeterminada que están incluidos en el estándar General MIDI, como el tom melódico, el tambor taiko, los tambores sintéticos, etc., no se pueden acceder a

través del canal MIDI 10, por lo que la notación para dichos instrumentos se debe introducir también en un contexto normal de `Staff`, utilizando las alturas normales adecuadas.

Muchos instrumentos de percusión no están incluidos dentro del estándar General MIDI, p.ej. las castañuelas. El método más fácil (aunque no satisfactorio) de producir una salida MIDI al escribir para estos instrumentos es sustituirlos por el sonido más parecido del conjunto estándar.

Advertencias y problemas conocidos

Debido a que el estándar General MIDI no contiene golpes de aro (*rim shots*), para este propósito se utiliza en su lugar el golpe lateral de baqueta (*sidestick*).

3.5.7 El script Articulate

Es posible obtener una salida MIDI más real si se utiliza el script Articulate. Éste trata de tener en cuenta las articulaciones (ligaduras de expresión, picados, etc), mediante la sustitución de las notas con secuencias musicales de silencios y notas escaladas en el tiempo de forma apropiada. También trata de desplegar los trinos, grupetos circulares, etc. y responder a las indicaciones de rallentando y accelerando.

Para utilizar el script Articulate, tenemos que incluirlo al principio de nuestro archivo de entrada:

```
\include "articulate.ly"
```

y en la sección `\score`, hacer lo siguiente:

```
\unfoldRepeats \articulate <<
resto de la partitura...
>>
```

Después de alterar el código de entrada de esta manera el resultado visual se altera profundamente, pero el bloque `\midi` estándar produce un archivo MIDI mejorado.

Si bien no es esencial para que funcione el script Articulate, podemos insertar la instrucción `\unfoldRepeats` según aparece en el ejemplo anterior, dado que habilita la ejecución de abreviaturas tales como los *trinos*.

Advertencias y problemas conocidos

Articulate acorta los acordes, y ciertos fragmentos de música (especialmente la música de órgano) es posible que suene peor.

3.6 Extraer información musical

Además de crear un resultado visual y MIDI, LilyPond es capaz de presentar información musical en forma de texto.

3.6.1 Displaying LilyPond notation

La impresión textual de una expresión musical en notación de LilyPond puede hacerse con la función musical `\displayLilyMusic`. Para ver el resultado, lo más usual es llamar a LilyPond mediante la línea de órdenes. Por ejemplo,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

imprime lo siguiente:

```
{ a,4 cis e fis g }
```

De forma predeterminada, LilyPond imprime estos mensajes en la consola junto al resto de los mensajes de la compilación de LilyPond. Para discernir entre estos mensajes y guardar el resultado de `\displayLilyMusic`, redireccione la salida hacia un archivo.

```
lilypond archivo.ly >resultado.txt
```

3.6.2 Impresión de las expresiones musicales de Scheme

Véase Sección “Presentación de las expresiones musicales” in *Extender*.

3.6.3 Guardar los eventos musicales en un archivo

Los eventos musicales se pueden guardar en un archivo pentagrama a pentagrama mediante la inclusión de un archivo en nuestra partitura principal.

```
\include "event-listener.ly"
```

De esta forma se crean uno o más archivos llamados ‘`FILENAME-STAFFNAME.notes`’ ó ‘`FILENAME-unnamed-staff.notes`’ para cada pentagrama. Observe que si tiene más de un pentagrama sin nombre, los eventos de todos los pentagramas se mezclarán entre sí dentro del mismo archivo. El resultado tiene el aspecto siguiente:

```
0.000   note     57       4   p-c 2 12
0.000   dynamic  f
0.250   note     62       4   p-c 7 12
0.500   note     66       8   p-c 9 12
0.625   note     69       8   p-c 14 12
0.750   rest     4
0.750   breathe
```

La sintaxis consiste en una línea delimitada por caracteres de tabulación, con dos campos fijos en cada línea seguidos de parámetros opcionales.

`tiempo  tipo  ...parámetros...`

Esta información se puede leer fácilmente por parte de otros programas como guiones de Python, y pueden ser muy útiles para aquellos investigadores que desean realizar un análisis musical o hacer experimientos de reproducción con LilyPond.

Advertencias y problemas conocidos

No todos los eventos musicales de lilypond están contemplados por ‘`event-listener.ly`’. Se pretende que sea una “prueba de concepto” bien realizada. Si algunos eventos que quiere ver no aparecen incluidos, copie ‘`event-listener.ly`’ en su carpeta de lilypond y modifique el archivo de forma que produzca la información que desea.

4 Problemas de espaciado

La disposición global sobre el papel viene determinada por tres factores: el diseño de página, los saltos de línea y el espaciado. Estos factores influyen entre sí. La elección de uno u otro espaciado determina qué densidad termina teniendo cada sistema de música. Esto influye en el lugar en que se eligen los saltos de línea y, en último término, en el número de páginas que tiene una partitura.

Considerado globalmente, este proceso se produce en cuatro fases: en primer lugar se eligen unas distancias elásticas o 'muelles', basados en la duración de las figuras. Se prueban todas las combinaciones de saltos de línea posibles, y para cada una de ellas se calcula una puntuación de 'maldad'. Después se hace una estimación de la altura de cada uno de los posibles sistemas. Finalmente se selecciona una combinación de saltos de página y de línea de tal forma que ni el espaciado horizontal ni el vertical estén demasiado estrechos ni estirados.

Hay dos tipos de bloques que pueden contener ajustes de disposición: `\paper {...}` y `\layout {...}`. El bloque `\paper` contiene ajustes de disposición de la página que se espera sean los mismos para todas las partituras de un libro o parte de libro, como la altura de la hoja o si se imprimen los números de página, etc. Véase Sección 4.1 [Disposición de la página], página 513. El bloque `\layout` contiene ajustes de disposición de la partitura, como el número de sistemas que utilizar, o la separación entre grupos de pentagramas, etc. Véase Sección 4.2 [Disposición de la partitura], página 524.

4.1 Disposición de la página

Esta sección estudia las opciones de disposición de la página para el bloque `\paper`.

4.1.1 El bloque \paper

Los bloques `\paper` pueden aparecer en tres lugares diferentes para formar una jerarquía descendente de bloques `\paper`:

- Al principio del archivo de entrada, antes de cualquier bloque `\book`, `\bookpart`, o `\score`.
- Dentro de un bloque `\book` pero fuera de cualquier bloque `\bookpart` o `\score` que estén dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` que esté dentro de esa parte de libro.

No se puede poner un bloque `\paper` dentro de un bloque `\score`.

Los valores de los campos filtran esta jerarquía, persistiendo los valores establecidos más arriba en la jerarquía a no ser que se sobreescriban por algún valor establecido más abajo en la jerarquía.

Pueden aparecer varios bloques `\paper` en cada uno de los niveles, por ejemplo como parte de distintos archivos incluidos con `\include`. Si es el caso, los campos en cada nivel se combinan, tomando precedencia los campos que aparecen más tarde si se producen campos duplicados.

Entre los ajustes que pueden aparecer dentro de un bloque `\paper` están los siguientes:

- la función de Scheme `set-paper-size`,
- variables de `\paper` utilizadas para personalizar la disposición de la página, y
- definiciones de marcado utilizadas para personalizar la disposición de los encabezamientos, pies y títulos.

La función `set-paper-size` se estudia en la sección siguiente, Sección 4.1.2 [Tamaño del papel y escalado automático], página 514. Las variables de `\paper` que se ocupan de la disposición de la página se estudian en secciones posteriores. Las definiciones de elementos de marcado

que se encargan de los encabezamientos, pies y títulos se estudian en Sección 3.2.2 [Títulos encabezamientos y pies de página personalizados], página 473.

Casi todas las variables de `\paper` funcionan solamente dentro de un bloque `\paper`. Las pocas que funcionan también en un bloque `\layout` están relacionadas en Sección 4.2.1 [El bloque \layout], página 524.

Excepto cuando se especifica lo contrario, todas las variables de `\paper` que corresponden a distancias sobre la página se miden en milímetros, a no ser que el usuario especifique unas unidades diferentes. Por ejemplo, la declaración siguiente establece el margen superior `top-margin` a diez milímetros:

```
\paper {
  top-margin = 10
}
```

Para fijarlo en `0.5` pulgadas, utilice el sufijo de unidades `\in`:

```
\paper {
  top-margin = 0.5\in
}
```

Los sufijos de unidades que están disponibles son `\mm`, `\cm`, `\in` y `\pt`. Estas unidades son valores simples para convertir a partir de milímetros; están definidos en '`ly/paper-defaults-init.ly`'. Por claridad, al usar milímetros, se suele escribir el sufijo `\mm` aunque no es necesario técnicamente.

También es posible definir valores de `\paper` utilizando Scheme. El equivalente de Scheme del ejemplo anterior es:

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

Véase también

Referencia de la notación: Sección 4.1.2 [Tamaño del papel y escalado automático], página 514, Sección 3.2.2 [Títulos encabezamientos y pies de página personalizados], página 473, Sección 4.2.1 [El bloque \layout], página 524.

Archivos instalados: '`ly/paper-defaults-init.ly`'.

4.1.2 Tamaño del papel y escalado automático

Fijar el tamaño del papel

'A4' es el valor predeterminado si no se establece ningún tamaño de página explícitamente. Sin embargo, existen dos funciones que se pueden usar para modificarlo:

`set-default-paper-size`

```
#(set-default-paper-size "quarto")
```

que siempre se debe colocar en el ámbito superior, y

`set-paper-size`

```
\paper {
  #(set-paper-size "tabloid")
}
```

que se debe colocar siempre dentro de un bloque `\paper`.

Si se usa la función `set-default-paper-size` en el ámbito superior jerárquico, debe ir antes de cualquier bloque `\paper`. `set-default-paper-size` establece el tamaño de todas las páginas, mientras que `set-paper-size` establece sólo el tamaño de las páginas a las que se aplica el bloque

\paper. Por ejemplo, si el bloque \paper está al principio del archivo, aplica el tamaño de papel a todas las páginas. Si el bloque \paper está dentro de un bloque \book, entonces el tamaño del papel sólo se aplica a ese libro.

Cuando se usa la función set-paper-size, se debe colocar *antes* de cualquier otra función que se utilice dentro del mismo bloque \paper. Véase [Escalado automático al tamaño del papel], página 515.

Los tamaños de página están definidos en el archivo 'scm/paper.scm', y aunque es posible añadir tamaños personalizados, se sobreescribirán en las actualizaciones del software posteriores. La lista de todos los tamaños de papel disponibles está en Sección A.5 [Tamaños de página predefinidos], página 644.

Se puede usar la siguiente instrucción en el archivo para añadir un tamaño de página personalizado que se puede usar después con set-default-paper-size o con set-paper-size según convenga,

```
#(set! paper-alist (cons '("mi tamaño" . (cons (* 15 in) (* 3 in))) paper-alist))

\paper {
  #(set-paper-size "mi tamaño")
}
```

Se puede usar cualquiera de las unidades in (pulgadas), cm (centímetros) o mm (milímetros).

Si se añade el símbolo 'landscape a la función del tamaño de la página, el papel se gira 90 grados y se establece adecuadamente la anchura de las líneas.

```
#(set-default-paper-size "a6" 'landscape)
```

Se pueden intercambiar las dimensiones del papel *sin que la impresión resulte girada* (como cuando se imprime en tamaño tarjeta postal, o al crear gráficos para su inclusión en lugar de un documento independiente) añadiendo 'landscape' después del nombre del propio tamaño del papel:

```
#(set-default-paper-size "a6landscape")
```

Cuando el tamaño del papel termina de forma explícita con 'landscape' o con 'portrait', la presencia de un símbolo 'landscape *solamente* afecta a la orientación de la impresión, no a las dimensiones del papel que se usan para la disposición.

Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 515, Sección A.5 [Tamaños de página predefinidos], página 644.

Archivos instalados: 'scm/paper.scm'.

Escalado automático al tamaño del papel

Si se cambia el tamaño del papel con una de las funciones de Scheme (set-default-paper-size o set-paper-size), los valores de algunas variables de \paper se escalan automáticamente al nuevo tamaño. Para sortear el escalado automático para una variable determinada, fije el valor de la variable después de establecer el tamaño del papel. Observe que el escalado automático no se activa mediante el ajuste de las variables paper-height o de paper-width, incluso aunque paper-width puede influir sobre otros valores (esto es distinto al escalado y se estudia más abajo). Las funciones set-default-paper-size y set-paper-size se describen en [Fijar el tamaño del papel], página 514.

Las dimensiones verticales afectadas por el escalado automático son top-margin y bottom-margin (véase Sección 4.1.3 [Variables de espaciado de \paper verticales fijas], página 516). Las dimensiones horizontales afectadas por el escalado automático son left-margin, right-margin, inner-margin, outer-margin, binding-offset, indent

y `short-indent` (véase Sección 4.1.5 [Variables de espaciado de `\paper` horizontales], página 519).

Los valores predeterminados para estas dimensiones están fijados en el archivo '`ly/paper-defaults-init.ly`', usando variables internas llamadas `top-margin-default`, `bottom-margin-default`, etc. Estos son los valores que resultan del tamaño predeterminado del papel `a4`. Como referencia, con el papel `a4` la altura `paper-height` es `297\mm` y la anchura `paper-width` es `210\mm`.

Véase también

Referencia de la notación: Sección 4.1.3 [Variables de espaciado de `\paper` verticales fijas], página 516, Sección 4.1.5 [Variables de espaciado de `\paper` horizontales], página 519.

Archivos de inicio: '`ly/paper-defaults-init.ly`', '`scm/paper.scm`'.

4.1.3 Variables de espaciado de `\paper` verticales fijas

Nota: Algunas dimensiones de `\paper` se escalan automáticamente al tamaño del papel, lo que puede llevar a un comportamiento no esperado. Véase [Escalado automático al tamaño del papel], página 515.

Los valores predeterminados (previos al escalado) están definidos en el archivo '`ly/paper-defaults-init.ly`'.

`paper-height (altura del papel)`
: Altura de la página, no fijada de forma predeterminada. Observe que el escalado automático de algunas dimensiones verticales no resulta afectado por esto.

`top-margin (margen superior)`
: Margen entre el extremo superior de la página y la parte superior de la zona imprimible. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello.

`bottom-margin (margen inferior)`
: Margen entre la parte inferior de la zona imprimible y el extremo inferior de la página. Si se modifica el tamaño del papel, se escala de acuerdo con ello el valor predeterminado de esta dimensión.

`ragged-bottom (sin justificar por abajo)`
: Si se fija al valor verdadero, los sistemas no se reparten verticalmente hasta abajo de la página. Esto no afecta a la última página. Se debe fijar a verdadero para piezas que sólo tienen dos o tres sistemas por página, como por ejemplo las partituras orquestales.

`ragged-last-bottom (sin justificar la última por abajo)`
: Si se establece al valor falso, los sistemas se reparten verticalmente hasta abajo de la última página. Las piezas que llenan sobradamente dos o más páginas deben tener este valor fijado a falso. También afecta a la última página de las partes de libro, es decir, partes de un libro creadas con bloques `\bookpart`.

Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 515.

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Advertencias y problemas conocidos

Los títulos (extraídos del bloque `\header`) se tratan como sistemas, así pues `ragged-bottom` y `ragged-last-bottom` aumentan la separación entre los títulos y el primer sistema de la partitura.

Los tamaños de papel definidos explícitamente sobreescriben cualquier ajuste de los márgenes superior o inferior definidos por el usuario.

4.1.4 Variables de espaciado de `\paper` verticales flexibles

En casi todos los casos es preferible que las distancias verticales entre ciertos elementos (como márgenes, títulos, sistemas y las distintas partituras) sean flexibles, de manera que se amplíen y compriman adecuadamente dependiendo de la situación. Están disponibles un cierto número de variables del bloque `\paper` (relacionadas más abajo) para realizar un ajuste fino del comportamiento de estas dimensiones frente a su ampliación o compresión.

Observe que las variables de `\paper` que se estudian en esta sección no controlan el espaciado de los pentagramas que están dentro de los sistemas individuales. El espaciado dentro de los sistemas se controla por medio de propiedades de grob, con ajustes que se escriben normalmente dentro de un bloque `\score` o `\layout`, y no dentro del bloque `\paper`. Véase Sección 4.4.1 [Espaciado vertical flexible dentro de los sistemas], página 536.

Estructura de las listas-A de espaciado vertical flexible

Cada una de las variables de espaciado de `\paper` flexibles es una lista-A (lista asociativa) que contiene cuatro *claves*:

- `basic-distance` (distancia básica): distancia vertical, medida en espacios de pentagrama, entre los *puntos de referencia* de los dos elementos, cuando no resulte ninguna colisión, y no se produzca ninguna ampliación o compresión. El punto de referencia de un elemento de marcado (de título o del nivel jerárquico superior) es su punto más alto, y el punto de referencia de un sistema es el centro vertical del `StaffSymbol` más cercano (incluso si se trata de una línea que no es un pentagrama, como un contexto `Lyrics`). Los valores de `distancia básica` menores de `padding` o de `minimum-distance` no son significativos, porque la distancia resultante nunca será menor de `padding` o de `minimum-distance`.
- `minimum-distance` (distancia mínima): la distancia vertical mínima permitida, medida en espacios de pentagrama, entre los puntos de referencia de los dos elementos, cuando se produce una compresión. Los valores de `minimum-distance` menores de `padding` no son significativos, porque la distancia resultante nunca será menor de `padding`.
- `padding` (relleno): mínimo espacio vertical vacío necesario entre los rectángulos circundantes (o las líneas de horizonte) de los dos elementos, medido en espacios de pentagrama.
- `stretchability` (ampliabilidad): medida sin unidades de la propensión relativa de esta dimensión a ampliarse. Si es cero, la distancia no se ampliará (a no ser que resultasen colisiones). Si es positiva, la significación del valor de ampliabilidad de una dimensión concreta depende solamente de su relación con los valores de `stretchability` de las otras dimensiones. Por ejemplo, si una dimensión tiene el doble de `stretchability` que otra, se ampliará con el doble de facilidad. Los valores deben ser no negativos y finitos. El valor `+inf.0` desencadena un error de programación y se ignora, pero se puede usar `1.0e7` para un muelle casi infinitamente ampliable. Si no se fija un valor, el valor predeterminado se establece a `basic-distance`. Observe que la propensión de la dimensión a *comprimirse* no se puede establecer directamente por el usuario y es igual a (`basic-distance` − `minimum-distance`).

Si una página tiene un margen inferior no justificado, la distancia resultante es la mayor de:

- `basic-distance`,
- `minimum-distance`, y
- `padding` más la menor distancia necesaria para eliminar las colisiones.

Para partituras de varias páginas con la parte inferior no justificada en la última página, ésta usará el mismo espaciado que la página anterior siempre y cuando haya espacio suficiente para ello.

Los métodos específicos para la modificación de las listas-A se estudian en Sección 5.3.6 [Modificación de las listas-A], página 597. El siguiente ejemplo muestra las dos formas en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor individualmente, y la segunda redefine la variable completamente:

```
\paper {
  system-system-spacing #'basic-distance = #8
  score-system-spacing =
    #'((basic-distance . 12)
       (minimum-distance . 6)
       (padding . 1)
       (stretchability . 12))
}
```

Lista de variables de espaciado de \paper verticales flexibles

Los nombres de estas variables siguen el formato *`superior-inferior`*`-spacing`, donde *`superior`* e *`inferior`* son los elementos que van a ser espaciados. Cada distancia se mide entre los puntos de referencia de los dos elementos (véase más arriba la descripción de la estructura de la lista-A). Observe que en estos nombres de variable, el término '`markup`' se refiere tanto a *elementos de marcado de título* (`bookTitleMarkup` o `scoreTitleMarkup`) como a *elementos de marcado del nivel superior* (véase Sección 3.1.5 [Estructura del archivo], página 463). Todas las distancias se miden en espacios de pentagrama.

Los ajustes predeterminados están definidos en el archivo '`ly/paper-defaults-init.ly`'.

`markup-system-spacing`
: distancia entre un elemento de marcado (de título o del nivel superior) y el sistema que le sigue.

`score-markup-spacing`
: distancia entre el último sistema de una partitura y el elemento de marcado (de título o del nivel superior) que le sigue.

`score-system-spacing`
: distancia entre el último sistema de una partitura y el primer sistema de la partitura que le sigue, cuando no existe ningún elemento de marcado (de título o del nivel superior) entre ellos.

`system-system-spacing`
: distancia entre dos sistemas dentro de la misma partitura.

`markup-markup-spacing`
: distancia entre dos elementos de marcado (de título o del nivel superior).

`last-bottom-spacing`
: distancia desde el último sistema o elemento de marcado del nivel superior en una página, hasta la parte inferior de la zona imprimible (es decir, el extremo superior del margen inferior).

`top-system-spacing`
: distancia desde la parte superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer sistema de una página, cuando no hay ningún elemento de marcado (de título o del nivel superior) entre los dos.

`top-markup-spacing`
: distancia desde el extremo superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer elemento de marcado (de título o del nivel superior) sobre una página, cuando no hay ningún sistema entre los dos.

Véase también

Referencia de la notación: Sección 4.4.1 [Espaciado vertical flexible dentro de los sistemas], página 536.

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

4.1.5 Variables de espaciado de \paper horizontales

Nota: Algunas dimensiones de `\paper` se escalan automáticamente al tamaño del papel, lo que puede dar lugar a un comportamiento distinto al esperado. Véase [Escalado automático al tamaño del papel], página 515.

Variables de \paper para la anchura y los márgenes

Los valores predeterminados (antes del escalado) que no están relacionados aquí se encuentran definidos en el archivo '`ly/paper-defaults-init.ly`'.

`paper-width`
: Anchura de la página, sin fijar de forma predeterminada. Aunque `paper-width` no tiene ningún efecto sobre el escalado automático de algunas dimensiones horizontales, sí influye sobre la variable `line-width`. Si están establecidas las dos variables `paper-width` y `line-width`, entonces también se actualizan `left-margin` y `right-margin`. Véase también `check-consistency`.

`line-width`
: Dimensiones horizontales de las líneas del pentagrama en los sistemas sin sangrado y justificados, igual a `(paper-width − left-margin − right-margin)` cuando está sin fijar. Si `line-width` está fijado, y tanto `left-margin` como `right-margin` están sin fijar, entonces los márgenes se actualizan para centrar los sistemas automáticamente sobre la página. Véase también `check-consistency`. Esta variable también se puede fijar dentro de un bloque `\layout`.

`left-margin`
: Margen entre el extremos izquierdo de la página y el comienzo de las líneas del pentagrama en los sistemas sin sangrado. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si no se fija el valor de `left-margin`, y tanto `line-width` como `right-margin` están fijados, entonces se establece el valor de `left-margin` a `(paper-width − line-width − right-margin)`. Si solamente está fijado `line-width`, entonces los dos márgenes se establecen a `((paper-width − line-width) / 2)`, y como consecuencia los sistemas se centran sobre la página. Véase también `check-consistency`.

`right-margin`
: Margen entre el extremo derecho de la página y el final de las líneas del pentagrama en sistemas justificados. Si el tamaño de la página se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si `right-margin` está sin fijar, y tanto `line-width` como `left-margin` están fijados, el valor de

`right-margin` se establece a `(paper-width − line-width − left-margin)`. Si solamente está fijado `line-width`, los dos márgenes se establecen a `((paper-width − line-width) / 2)`, y como consecuencia los sistemas aparecen centrados sobre la página. Véase también `check-consistency`.

`check-consistency`
: Si se establece al valor verdadero, se imprime una advertencia en caso de que la suma de `left-margin`, `line-width` y `right-margin` no coincida exactamente con el valor de `paper-width`, y se sustituye cada uno de estos valores (excepto `paper-width`) con su valor predeterminado (escalado al tamaño del papel si es necesario). Si está fijado al valor falso, se ignora cualquier inconsistencia y se permite que los sistemas se salgan del borde del papel.

`ragged-right`
: Si está fijado al valor verdadero, los sistemas no cubren el ancho de la línea. En lugar de ello, los sistemas finalizan en su longitud horizontal natural. Valor predeterminado: `#t` para las partituras con un solo sistema, y `#f` para partituras que tienen dos o más sistemas. Esta variable también se puede establecer dentro de un bloque `\layout`.

`ragged-last`
: Si está fijado al valor verdadero, el último sistema de la partitura no llena toda la anchura de la línea. En lugar de ello, el último sistema finaliza en su longitud horizontal natural. Valor predeterminado: `#f`. Esta variable también puede ajustarse dentro de un bloque `\layout`.

Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 515.

Archivos de inicio: ‘`ly/paper-defaults-init.ly`’.

Advertencias y problemas conocidos

Los tamaños de papel definidos explícitamente sobreescriben cualquier ajuste de los márgenes izquierdo o derecho definidos por el usuario.

Variables de \paper para el modo de doble cara

Los valores predeterminados (antes del escalado) se definen en ‘`ly/paper-defaults-init.ly`’.

`two-sided`
: Si tiene el valor verdadero, utilizar `inner-margin`, `outer-margin` y `binding-offset` para determinar los márgenes dependiendo de si el número de la página es par o impar. Esto sobreescribe a `left-margin` y a `right-margin`.

`inner-margin`
: Margen que todas las páginas tienen en el lado interno si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

`outer-margin`
: Margen que todas las páginas tienen en el lado externo si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

`binding-offset`
: Medida en que se aumenta `inner-margin` para asegurar que no se oculta nada a causa de la encuadernación. Si el tamaño del papel se modifica, el valor predeter-

minado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 515.

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Variables de \paper para desplazamientos y sangrados

Los valores predeterminados (antes del escalado) que no se relacionan aquí están definidos en el archivo '`ly/paper-defaults-init.ly`'.

`horizontal-shift`
: Medida en que todos los sistemas (incluidos los títulos de cabecera y los separadores de sistemas) se desplazan a la derecha. Predeterminado: `0.0`.

`indent`
: Nivel de sangrado para el primer sistema de una partitura. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Esta variable también se puede ajustar dentro de un bloque `\layout`.

`short-indent`
: Nivel de sangrado para todos los sistemas de una partitura excepto el primero. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Esta variable también se puede ajustar dentro de un bloque `\layout`.

Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 515.

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

4.1.6 Otras variables de \paper

Variables de \paper para los saltos de línea

`max-systems-per-page`
: Número máximo de sistemas que se colocan sobre una página. Solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`min-systems-per-page`
: Número mínimo de sistemas que se colocan sobre una página. Puede causar que las páginas aparezcan atiborradas si el valor es demasiado grande En el momento actual solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`systems-per-page`
: Número de sistemas que se deben colocar en cada página. En el momento actual solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`system-count`
: Número de sistemas que utilizar para una partitura. Valor predeterminado: no fijado. Esta variables también se puede ajustar dentro de un bloque `\layout`.

Véase también

Referencia de la notación: Sección 4.3.1 [Saltos de línea], página 527.

Variables de \paper para los saltos de página

Los valores predeterminados que no están relacionados aquí, se definen en el archivo ‘`ly/paper-defaults-init.ly`’

`page-breaking`
: Algoritmo de saltos de página que utilizar. Se puede elegir entre `ly:minimal-breaking`, `ly:page-turn-breaking`, `ly:one-line-breaking` y `ly:optimal-breaking` (predeterminado).

`page-breaking-system-system-spacing`
: Engaña a la parte del programa encargada de los saltos de página para que piense que `system-system-spacing` está establecido a un valor distinto al que tiene realmente. Por ejemplo, si `page-breaking-system-system-spacing #'padding` está establecido a algo que es mucho mayor que `system-system-spacing #'padding`, entonces el divisor de páginas pone menos sistemas en cada página. Valor predeterminado: sin ajustar.

`page-count`
: Número de páginas que utilizar en una partitura, no ajustado de forma predeterminada.

Las siguientes variables son efectivas solamente cuando `page-breaking` está sijado a `ly:page-turn-breaking`. Los saltos de página se seleccionan entonces de forma que se minimice el número de pasos de página. Dado que los pasos de página son necesarios cuando se pasa de una página impar a una página par, normalmente resultará favorecida aquella disposición en que la última página sea impar. Los lugares en que se prefieran pasos de página pueden indicarse manualmente insertando `\allowPageTurn` o automáticamente incluyendo el grabador `Page_turn_engraver` (véase Sección 4.3.4 [Paso de página óptimo], página 531).

Si no hay un número suficiente de posibilidades para escoger a la hora de insertar unos pasos de página adecuados, LilyPond puede insertar una página vacía dentro de una partitura, entre partituras (si hay dos o más partituras) o finalizando una partitura en una página de numeración par. Los valores de las tres siguientes variables pueden incrementarse para que estas acciones se vuelvan menos probables.

Los valores son penalizaciones, es decir, cuando más alto es el valor menos probables será la acción asociada relativa a otras posibilidades.

`blank-page-penalty (penalización por página vacía)`
: Penalización por tener una página vcía en mitad de una partitura. Si `blank-page-penalty` es grande y está seleccionado `ly:page-turn-breaking`, entonces será menos probable que LilyPond inserte una página en medio de una partitura. En lugar de ello, intercala espacio en la música hasta que se rellene la página vacía y la siguiente. Predeterminado: 5.

`blank-last-page-penalty (penalización por última página vacía)`
: Penalización por terminar la partitura sobre una página par. Si `blank-last-page-penalty` es grande y está seleccionado `ly:page-turn-breaking`, entonces será menos probable que LilyPond produzca una partitura en que la última página tenga numeración par. En su lugar, ajustará el espaciado de manera que use una página más o una página menos. Predeterminado: 0.

`blank-after-score-page-penalty (penalización por página vacía después de una partitura)`
: Penalización por tener una página vacía después del final de una partitura y antes de la siguiente. De forma predeterminada, esta penalización es más pequeña que `blank-page-penalty`, de forma que se insertan páginas vacías después de las partituras como preferencia sobre la inserción de páginas vacías dentro de una partitura. Predeterminado: 2.

Véase también

Referencia de la notación: Sección 4.3.2 [Saltos de página], página 530, Sección 4.3.3 [Saltos de página óptimos], página 531, Sección 4.3.4 [Paso de página óptimo], página 531, Sección 4.3.5 [Saltos de página mínimos], página 532, Sección 4.3.6 [Saltos de página de una línea], página 532.

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Variables de \paper para la numeración de las páginas

Los valores predeterminados que no se relacionan aquí están definidos en el archivo '`ly/paper-defaults-init.ly`'

`auto-first-page-number`
: El algoritmo de división de páginas está afectado por el hecho de que el número de la primera página sea par o impar. Si está establecido al valor verdadero, el algoritmo de división de páginas decide si comenzar con un número par o impar. Esto hace que el número de la primera página se quede como está, o que se aumente en una unidad. Predeterminado: `#f`.

`first-page-number`
: Valor del número de pátina en la primera página.

`print-first-page-number`
: Si está fijado a verdadero, se imprime un número de página en la primera página.

`print-page-number`
: Si está fijado a falso, no se imprimen los números de página.

Véase también

Archivos de inicio: '`ly/paper-defaults-init.ly`'.

Advertencias y problemas conocidos

Los números de página impares siempre están a la derecha. Si quiere que la música comience en la página 1 debe haber una página en blanco en el reverso de la página de cubierta de manera que la página 1 esté en el lado derecho.

Variables de \paper diversas

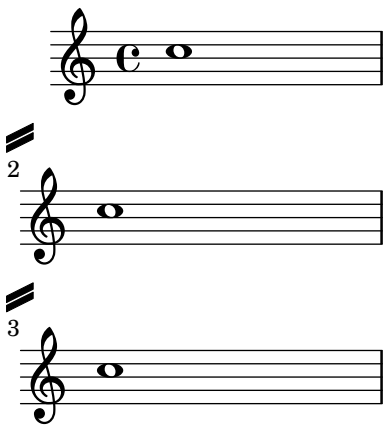
`page-spacing-weight`
: Importancia relativa del espacio (vertical) de las páginas y el espaciado (horizontal) de las líneas. Los valores altos hacen que el espaciado de la página tenga más importancia. Predeterminado: `10`.

`print-all-headers`
: Si está fijado a verdadero, imprime todos los encabezamientos para cada una de las partituras `\score` de la salida. Normalmente sólo se imprimen las variables de encabezamiento `piece` y `opus`. Valor predeterminado: `#f`.

`system-separator-markup`
: Objeto de marcado que se inserta entre los sistemas. Se suele usar para partituras orquestales. Predeterminado: sin establecer. El elemento de marcado `\slashSeparator`, definido en el archivo '`ly/titling-init.ly`', se aporta como un valor predeterminado adecuado, por ejemplo:

```
#(set-default-paper-size "a8")

\book {
  \paper {
    system-separator-markup = \slashSeparator
  }
  \header {
    tagline = ##f
  }
  \score {
    \relative c'' { c1 \break c1 \break c1 }
  }
}
```



Véase también

Archivos de inicio: '`ly/titling-init.ly`'.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Advertencias y problemas conocidos

El encabezamiento de página predeterminado pone el número de página y el campo `instrument` del bloque `\header` sobre la misma línea.

4.2 Disposición de la partitura

Esta sección trata de las opciones de disposición de la partitura para el bloque `\layout`.

4.2.1 El bloque `\layout`

Así como el bloque `\paper` contiene ajustes relativos al formateo de la página para todo el documento, el bloque `\layout` contiene ajustes para la disposición específica de las partituras. Para fijar las opciones de disposición de partituras de forma global, escríbalas en un bloque `\layout` del nivel jerárquico superior. Para fijar opciones de disposición para una partitura individual escríbalas dentro de un bloque `\layout` dentro del bloque `\score`, después de la música. Entre los ajustes que pueden aparecer en un bloque `\layout` se encuentran los siguientes:

- la función de Scheme `layout-set-staff-size`,
- las modificaciones de contexto de los bloques `\context`, y
- las variables de `\paper` que afectan a la disposición de la partitura.

La función `layout-set-staff-size` se estudia en la sección siguiente, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526. Las modificaciones de contexto se estudian en un capítulo distinto; véase Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 575 y Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 577. Las variables de `\paper` que pueden aparecer dentro de un bloque `\layout` son:

- `line-width`, `ragged-right` y `ragged-last` (véase [Variables de `\paper` para la anchura y los márgenes], página 519)
- `indent` y `short-indent` (véase [Variables de `\paper` para desplazamientos y sangrados], página 521)
- `system-count` (véase [Variables de `\paper` para los saltos de línea], página 521)

He aquí un ejemplo de un bloque `\layout`:

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGrouper.staff-staff-spacing.basic-distance = #8
  }
  \context {
    \Voice
    \override TextScript.padding = #1
    \override Glissando.thickness = #3
  }
}
```

Se pueden escribir varios bloques `\layout` como expresiones del nivel superior. Esto puede ser útil, por ejemplo, si hay distintos ajustes almacenados en diversos archivos y se incluyen opcionalmente. Internamente, cuando se evalúa un bloque `\layout`, se hace una copia de la configuración de `\layout` actual, y después se aplican los cambios definidos en el bloque; el resultado se guarda como la nueva configuración actual. Desde el punto de vista del usuario, los bloques `\layout` se combinan, pero en situaciones de conflicto (cuando se cambia la misma propiedad en diferentes bloques) las definiciones más tarías toman precedencia.

Por ejemplo: si este bloque:

```
\layout {
  \context {
    \Voice
    \override TextScript.color = #magenta
    \override Glissando.thickness = #1.5
  }
}
```

se coloca después del del ejemplo anterior, las sobreescrituras de `'padding` y de `'color` para `TextScript` se combinan, pero la última sobreescritura `'thickness` para `Glissando` sustituye (u oculta) a la anterior.

Los bloques `\layout` se pueden asignar a variables para una reutilización posterior, pero la forma en que esto funciona es ligera y significativamente diferente del hecho de escribirlas literalmente.

Si se define una variable de la siguiente manera:

```
variable_de_Layout = \layout {
  \context {
    \Voice
    \override NoteHead.font-size = #4
  }
}
```

guardará la configuración actual de `\layout` con la sobreescritura añadida de `NoteHead.font-size`, pero esta combinación *no* se guarda como la nueva configuración actual. Tenga en cuenta que la 'configuración actual' se lee cuando la variable se define y no cuando se usa, de manera que el contenido de la variable depende de su posición dentro del código fuente.

La variable se puede entonces usar dentro de otro bloque `\layout`, por ejemplo:

```
\layout {
  \variable_de_layout
  \context {
    \Voice
    \override NoteHead.color = #red
  }
}
```

Un bloque `\layout` que contiene una variable, como en el ejemplo anterior, *no* copia la configuración actual, sino que en lugar de ello usa el contenido de `\variable_de_layout` como la configuración base para las adiciones posteriores. Esto significa que cualquier cambio definido entre la definición y el uso de la variable, se pierde.

Si la `variable_de_layout` se define (o se incluye con `\include`) inmediatamente antes de que se utilice, su contenido es sencillamente la configuración en curso más las sobreescrituras que se hayan definido dentro de él. Así pues, en el ejemplo anterior que muestra el uso de `\variable_de_layout` el último bloque `\layout` consistiría en:

```
  TextScript.padding = #1
  TextScript.color = #magenta
  Glissando.thickness = #1.5
  NoteHead.font-size = #4
  NoteHead.color = #red
```

más las sobreescrituras de `indent` y de `StaffGrouper`.

Pero si la variable ya hubiera sido definida antes del primer bloque `\layout` la configuración actual contendría ahora solamente

```
  NoteHead.font-size= #4 % (escrita en la definición de la variable)
  NoteHead.color = #red % (añadida después del uso de la variable)
```

Si se planifica cuidadosamente, las variables de `\layout` pueden ser una herramienta valiosa para estructurar el diseño de la disposición de las fuentes, y tambiién para reiniciar la configuración de `\layout` a un estado conocido.

Véase también

Referencia de la notación: Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 577.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

4.2.2 Establecer el tamaño del pentagrama

El **tamaño de pentagrama** predeterminado se establece en 20 puntos. Esto se puede modificar de dos maneras:

Para establecer globalmente el tamaño del pentagrama para todas las partituras de un archivo (o en un bloque `book`, para ser exactos), utilice `set-global-staff-size`.

```
#(set-global-staff-size 14)
```

Esto establece el tamaño global predeterminado a una altura de pentagrama de 14pt y escala todas las tipografías según corresponda.

Para establecer el tamaño del pentagrama de forma individual para cada partitura, use

```
\score{
  ...
  \layout {
    #(layout-set-staff-size 15)
  }
}
```

La fuente tipográfica Feta ofrece símbolos musicales a ocho tamaños distintos. Cada fuente tipográfica está afinada para un tamaño de pentagrama distinto: a un tamaño menor la tipografía se vuelve más pesada, para que coincida con las líneas de pentagrama relativamente más gruesas. Los tamaños de tipografía recomendados están relacionados en la tabla siguiente:

nombre de la fuente	**altura del pentagrama (pt)**	**altura del pentagrama (mm)**	**usos**
feta11	11.22	3.9	partituras de bolsillo
feta13	12.60	4.4	
feta14	14.14	5.0	
feta16	15.87	5.6	
feta18	17.82	6.3	cancioneros
feta20	20	7.0	particellas estándar
feta23	22.45	7.9	
feta26	25.2	8.9	

Estas tipografías están disponibles en cualquier tamaño. La propiedad de contexto `fontSize` y la propiedad de disposición `staff-space` (en Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*) se pueden usar para afinar el tamaño de los pentagramas individuales. Los tamaños de pentagrama individuales están en relación al tamaño global.

Véase también

Referencia de la notación: [Seleccionar el tamaño de la tipografía para la notación], página 213.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

`layout-set-staff-size` no cambia la distancia entre las líneas del pentagrama.
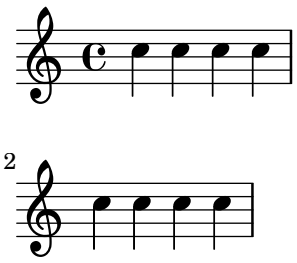
4.3 Saltos

4.3.1 Saltos de línea

Normalmente los saltos de línea se determinan automáticamente. Se eligen de forma que las líneas no aparezcan demasiado apretadas ni demasiado sueltas, y que las líneas consecutivas tengan una densidad similar.
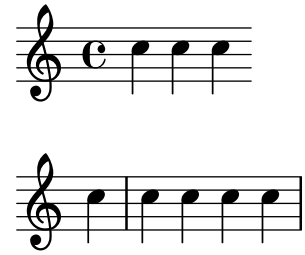
Para forzar manualmente un salto de línea sobre la línea divisoria, utilice la instrucción `\break`:

```
c4 c c c | \break
c4 c c c |
```



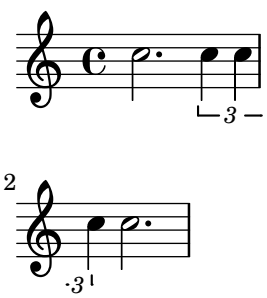
De forma predeterminada se ignora cualquier `\break` en la mitad de un compás, y se imprime una advertencia. Para forzar un salto de línea en medio de un compás, añada una barra de compás invisible con '`\bar ""`':

```
c4 c c
\bar "" \break
c |
c4 c c c |
```
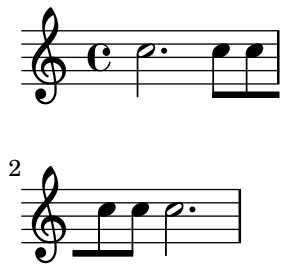


También se ignora un `\break` que se produce en la línea divisoria si el compás anterior termina en medio de una nota, como cuando un grupo de valoración especial inicia y termina en compases diferentes. Para permitir que las instrucciones `\break` funcionen en estas situaciones, elimine el grabador `Forbid_line_break_engraver` del contexto `Voice`. Observe que los saltos de línea forzados manualmente se deben añadir en paralelo con la música:

```
\new Voice \with {
  \remove "Forbid_line_break_engraver"
} \relative c'' {
  <<
    { c2. \tuplet 3/2 { c4 c c } c2. | }
    { s1 | \break s1 | }
  >>
}
```



De forma similar, los saltos de línea están normalmente prohibidos cuando hay barras de corchea que cruzan la línea divisoria. Se puede cambiar este comportamiento fijando `\override Beam.breakable = ##t`:

```
\override Beam.breakable = ##t
c2. c8[ c | \break
c8 c] c2. |
```



La instrucción `\noBreak`, prohíbe un salto de línea en la barra divisoria en que se inserta.

Los ajustes más básicos que influyen sobre el espaciado de las líneas son `indent` y `line-width`. Se establecen dentro del bloque `\layout`. Controlan el sangrado de la primera línea de música, y la longitud de las líneas.

Si se establece `ragged-right` a verdadero en el bloque `\layout`, los sistemas terminan en su longitud horizontal natural, en lugar de distribuirse horizontalmente para llenar toda la línea. Esto es de utilidad para fragmentos cortos, y para efectuar una comprobación de lo apretado que es el espaciado natural.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza.

```
\layout {
  indent = 0\mm
  line-width = 150\mm
  ragged-last = ##t
}
```

Para saltos de línea a intervalos regulares utilice `\break` separado mediante desplazamientos con `\skip` y repetidos con `\repeat`. Por ejemplo, esto haría que los 28 compases siguientes (suponiendo un compás de 4/4) se dividan cada cuatro compases, y sólo en dichos lugares:

```
<<
  \repeat unfold 7 {
    s1 \noBreak s1 \noBreak
    s1 \noBreak s1 \break
  }
  { la música real... }
>>
```

Instrucciones predefinidas

`\break`, `\noBreak`.

Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de línea], página 521.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LineBreakEvent” in *Referencia de Funcionamiento Interno*.

4.3.2 Saltos de página

Se puede sobreescribir el mecanismo predeterminado de salto de página insertando instrucciones `\pageBreak` o `\noPageBreak`. Estas instrucciones son análogas a `\break` y `\noBreak`. Se deben insertar en una línea divisoria. Estas instrucciones fuerzan y prohíben, respectivamente, la eventualidad de un salto de página. Por supuesto, la instrucción `\pageBreak` también fuerza un salto de línea.

Las instrucciones `\pageBreak` y `\noPageBreak` también se pueden insertar en el nivel más alto, entre las partituras y los elementos de marcado situados en el nivel superior.

Hay ajustes análogos a `ragged-right` y `ragged-last` que tienen el mismo efecto sobre el espaciado vertical: `ragged-bottom` y `ragged-last-bottom`. Si están establecidos a `#t` los sistemas de todas las páginas o sólo de la última página, respectivamente, no se verán justificados verticalmente. Véase Sección 4.1.3 [Variables de espaciado de `\paper` verticales fijas], página 516.

Los saltos de página se calculan por medio de la función `page-breaking`. LilyPond ofrece tres algoritmos para el cómputo de los saltos de página: `ly:optimal-breaking`, `ly:page-turn-breaking` y `ly:minimal-breaking`. El predeterminado es `ly:optimal-breaking`, pero el valor se puede cambiar en el bloque `\paper`:

```
\paper {
  page-breaking = #ly:page-turn-breaking
}
```

Cuando un libro tiene muchas partituras y páginas, puede ser difícil resolver el problema de los saltos de página, necesitando mucha memoria y prolongados tiempos de procesamiento. Para facilitar el proceso de división en páginas, se usan los bloques `\bookpart` para dividir el libro en varias partes: los saltos de página se producen de manera independiente en cada parte. También se pueden usar diferentes funciones de división en páginas para las distintas partes del libro.

```
\bookpart {
  \header {
    subtitle = "Prefacio"
  }
  \paper {
     %% En una parte que consiste en texto principalmente,
     %% puede ser preferible ly:minimal-breaking
     page-breaking = #ly:minimal-breaking
  }
  \markup { ... }
  ...
}
\bookpart {
  %% En esta parte, consistente en música, se usa la función
  %% óptima predeterminada de saltos de página.
  \header {
    subtitle = "Primer movimiento"
  }
  \score { ... }
  ...
}
```

Instrucciones predefinidas

`\pageBreak`, `\noPageBreak`.

Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de página], página 522.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.3 Saltos de página óptimos

La función `ly:optimal-breaking` es el método predeterminado de LilyPond para determinar los saltos de página. Intenta hallar una división de páginas que haga mínimos el apretujamiento y la distensión, tanto horizontal como verticalmente. A diferencia de `ly:page-turn-breaking`, no tiene un concepto de los pasos de página.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.4 Paso de página óptimo

Con frecuencia es necesario encontrar una configuración de división de páginas de manera que haya un silencio al final de una página de cada dos. De esta forma, el músico puede pasar la página sin perder notas. La función `ly:page-turn-breaking` trata de encontrar una división de páginas que haga mínimos el apretujamiento y el estiramiento, pero con la restricción añadida de que sólo se permite introducir vueltas de página en los lugares especificados.

Hay dos etapas en el uso de esta función de división de páginas. En primer lugar debemos habilitarlo en el bloque `\paper`, como se explicó en Sección 4.3.2 [Saltos de página], página 530. Entonces debemos decirle a la función dónde nos gustaría permitir los saltos de página.

Hay dos formas de conseguir la segunda tarea. Primero, podemos especificar manualmente cada uno de los pasos de página potenciales, insertando `\allowPageTurn` en nuestro archivo de entrada en los lugares adecuados.

Si esto es demasiado tedioso, podemos añadir un grabador `Page_turn_engraver` a un contexto Staff o Voice. El grabador `Page_turn_engraver` analizará el contexto en busca de secciones sin notas (observe que no busca silencios, sino la ausencia de notas. Se hace así para que la polifonía en un solo pentagrama con silencios en una de las voces no arruine la labor del grabador `Page_turn_engraver`). Cuando encuentra una sección sin notas suficientemente larga, el grabador `Page_turn_engraver` inserta un `\allowPageTurn` en la última barra de compás de dicha sección, a no ser que haya una barra ‘especial’ de compás (como una doble barra), en cuyo caso se insertará el `\allowPageTurn` en la última barra ‘especial’ de compás de la sección.

El grabador `Page_turn_engraver` lee la propiedad de contexto `minimumPageTurnLength` para determinar qué longitud debe tener una sección sin notas antes de que se considere la posibilidad de un paso de página. El valor predeterminado para `minimumPageTurnLength` es `(ly:make-moment 1/1)`. Si quiere inhabilitar las vueltas de página, puede establecerlo a algún valor muy grande.

```
\new Staff \with { \consists "Page_turn_engraver" }
{
  a4 b c d |
  R1 | % aquí se permite un salto de página
  a4 b c d |
  \set Staff.minimumPageTurnLength = #(ly:make-moment 5/2)
  R1 | % aquí no se permite un salto de página
  a4 b r2 |
  R1*2 | % aquí se permite un salto de página
  a1
}
```

El grabador `Page_turn_engraver` detecta las repeticiones de primera y segunda vez. Sólo permite un pase de página durante la repetición si hay suficiente tiempo al principio y al final de la repetición para volver a pasar la página hacia atrás. El grabador `Page_turn_engraver` también puede inhabilitar los pasos de página si la repetición es muy corta. Si establecemos la propiedad de contexto `minimumRepeatLengthForPageTurn` entonces el grabador `Page_turn_engraver` sólo permitirá los pases de página en las repeticiones cuya duración sea mayor que este valor.

Las instrucciones de paso de página, `\pageTurn`, `\noPageTurn` y `\allowPageTurn`, se pueden usar también en el nivel más elevado del código, entre las partituras y los elementos de marcado del nivel superior.

Instrucciones predefinidas

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de línea], página 521.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

Sólo debería haber un grabador `Page_turn_engraver` dentro de una partitura. Si hay más de uno, se interferirán entre sí.

4.3.5 Saltos de página mínimos

La función `ly:minimal-breaking` efectúa unos cálculos mínimos para determinar los saltos de página: completa una página con tantos sistemas como sea posible antes de continuar con la siguiente. Así, puede preferirse para partituras con muchas páginas, donde las otras funciones de salto de página pueden resultar demasiado lentas o ávidas de memoria, o con una gran cantidad de textos. Se habilita utilizando:

```
\paper {
  page-breaking = #ly:minimal-breaking
}
```

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.6 Saltos de página de una línea

La función `ly:one-line-breaking` es un algoritmo de saltos de página para uso especial que pone cada partitura en una página y en una sola línea. Esta función de salto de página no compone los títulos ni los márgenes; solamente se imprime la partitura.

El ancho de la página se ajusta de forma que la partitura más larga quepa en una línea. Concretamente, las variables `paper-width`, `line-width` y `indent` del bloque `\paper` se ignoran, aunque `left-margin` y `right-margin` aún se respetan. La altura de la página se deja sin cambios.

4.3.7 Saltos de línea explícitos

Lily a veces rechaza las instrucciones `\break` y `\pageBreak` explícitas. Hay dos instrucciones para sobreescribir este comportamiento:

```
\override NonMusicalPaperColumn.line-break-permission = ##f
\override NonMusicalPaperColumn.page-break-permission = ##f
```

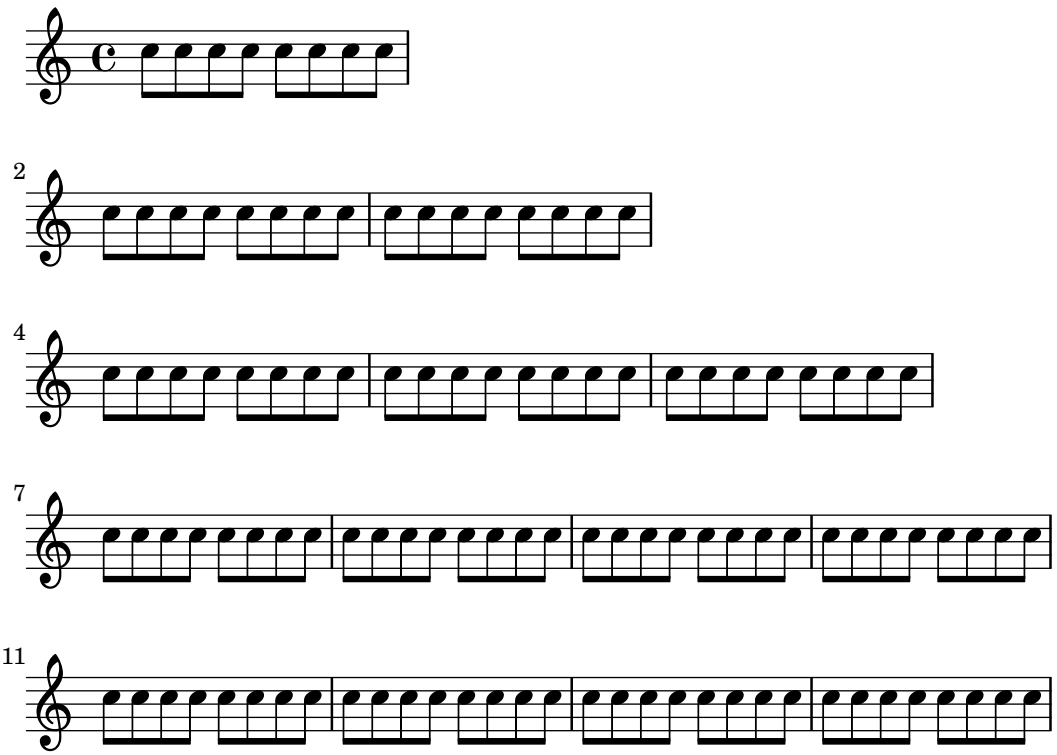
Cuando se sobreescribe el valor de `line-break-permission` a falso, Lily inserta saltos de línea en las instrucciones `\break` explícitas y en ningún otro lugar. Cuando se sobreescribe

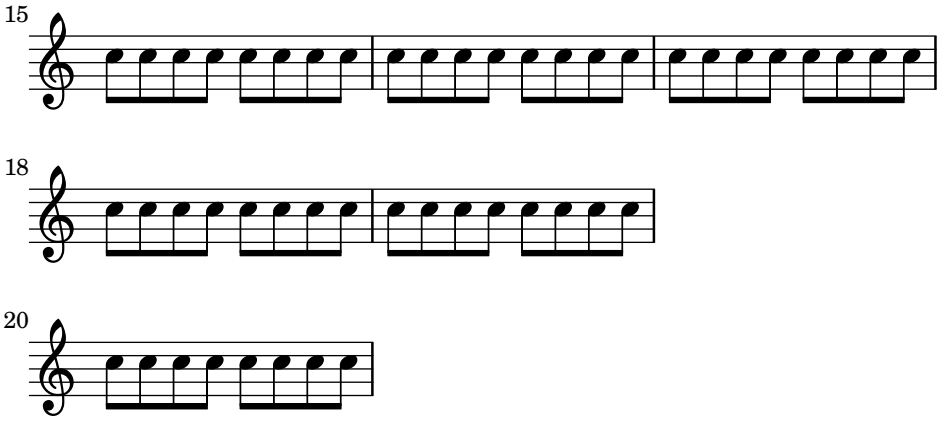
el valor de `page-break-permission` a falso, Lily inserta saltos de página en las instrucciones `\pageBreak` explícitas y en ningún otro lugar.

```
\paper {
  indent = #0
  ragged-right = ##t
  ragged-bottom = ##t
}

music = \relative c'' { c8 c c c }

\score {
  \new Staff {
    \repeat unfold 2 { \music } \break
    \repeat unfold 4 { \music } \break
    \repeat unfold 6 { \music } \break
    \repeat unfold 8 { \music } \pageBreak
    \repeat unfold 8 { \music } \break
    \repeat unfold 6 { \music } \break
    \repeat unfold 4 { \music } \break
    \repeat unfold 2 { \music }
  }
  \layout {
    \context {
      \Score
      \override NonMusicalPaperColumn.line-break-permission = ##f
      \override NonMusicalPaperColumn.page-break-permission = ##f
    }
  }
}
```





Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.8 Utilizar una voz adicional para los saltos de línea

La información sobre saltos de línea y de página suele aparecer directamente entremezclado dentro del código de notas.

```
music = \relative c'' { c4 c c c }
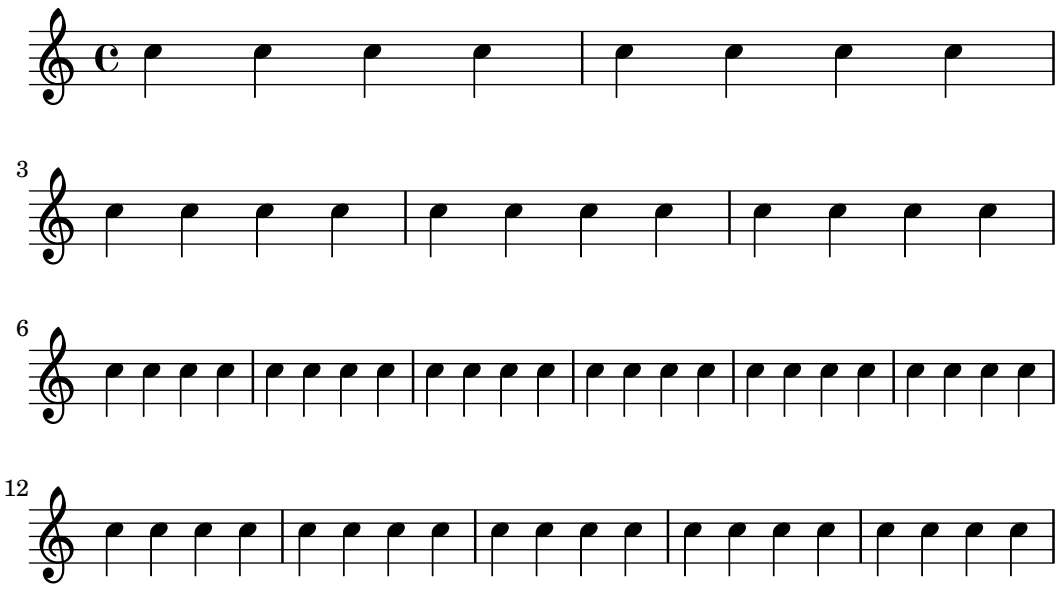
\score {
  \new Staff {
    \repeat unfold 2 { \music } \break
    \repeat unfold 3 { \music }
  }
}
```

Esto hace sencillas de introducir a las instrucciones `\break` y `\pageBreak`, pero mezcla la escritura de música con información que especifica cómo se debe disponer la música sobre la página. Podemos mantener la introducción de música y la información de saltos de línea y de página en dos lugares separados mediante la introducción de una voz adicional que contenga los saltos. Esta voz adicional contiene solamente desplazamientos o ‘skips’ junto con los `\break`, `pageBreak` y otras informaciones sobre la disposición de los saltos.

```
music = \relative c'' { c4 c c c }

\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    \new Staff <<
      \new Voice {
        s1 * 2 \break
        s1 * 3 \break
        s1 * 6 \break
        s1 * 5 \break
      }
      \new Voice {
        \repeat unfold 2 { \music }
        \repeat unfold 3 { \music }
        \repeat unfold 6 { \music }
        \repeat unfold 5 { \music }
      }
    >>
```

```
  }
}
```



Este patrón resulta especialmente útil cuando se sobreescribe `line-break-system-details` y las otras útiles (pero largas) propiedades de `NonMusicalPaperColumnGrob`, como se explica en Sección 4.4 [Espaciado vertical], página 536.

```
music = \relative c'' { c4 c c c }

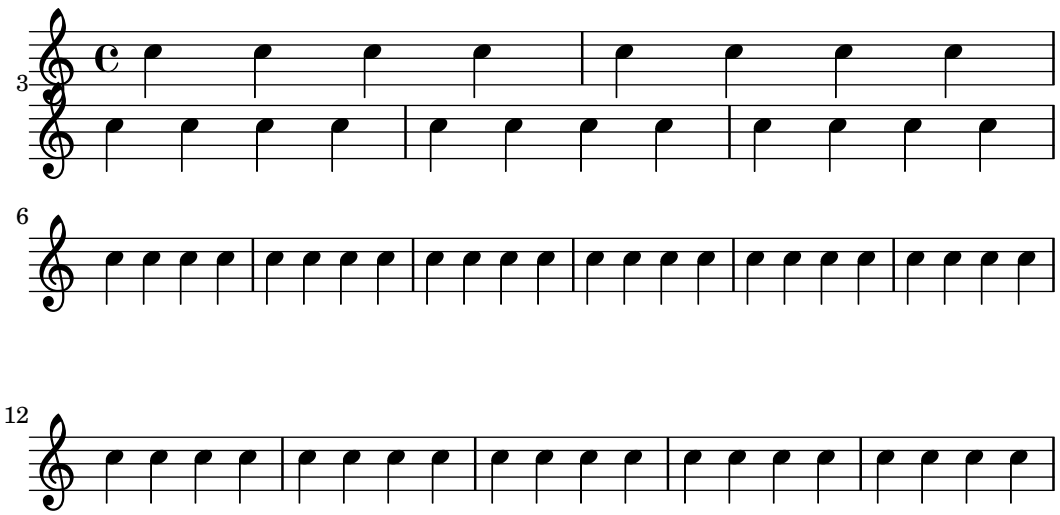
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    \new Staff <<
      \new Voice {
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 0))
        s1 * 2 \break

        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 5))
        s1 * 3 \break

        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 15))
        s1 * 6 \break

        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 30))
        s1 * 5 \break
      }
      \new Voice {
        \repeat unfold 2 { \music }
        \repeat unfold 3 { \music }
        \repeat unfold 6 { \music }
```

```
        \repeat unfold 5 { \music }
      }
    >>
  }
}
```



Véase también

Referencia de la notación: Sección 4.4 [Espaciado vertical], página 536.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.4 Espaciado vertical

El espaciado vertical está controlado por tres factores: la cantidad de espacio disponible (es decir, el tamaño del papel y los márgenes), la separación entre los sistemas, y la separación entre los pentagramas dentro de un sistema.

4.4.1 Espaciado vertical flexible dentro de los sistemas

Tres mecanismos distintos controlan el espaciado vertical flexible dentro de los sistemas, uno por cada una de las categorías siguientes:

- *pautas sin agrupar*,
- *pautas agrupadas* (pautas dentro de un grupo de pentagramas como `ChoirStaff`, etc.), y
- *líneas que no son pautas* (como `Lyrics`, `ChordNames`, etc.).

La altura de cada sistema se determina en dos fases. Primero, todos los pentagramas se disponen según la cantidad de espacio vacío disponible. Después, las líneas que no son pautas se distribuyen entre las pautas.

Observe que los mecanismos de espaciado estudiados en esta sección solamente controlan el espaciado vertical de las pautas y líneas que no son pautas dentro de los sistemas individuales. El espaciado vertical entre distintos sistemas, partituras, marcados y márgenes se controla mediante variables de `\paper` que se estudian en Sección 4.1.4 [Variables de espaciado de `\paper` verticales flexibles], página 517.

Propiedades de espaciado dentro de los sistemas

Los mecanismos de espaciado vertical dentro del sistema están controlados por dos conjuntos de propiedades de grob. El primer conjunto está asociado con el grob `VerticalAxisGroup`, que se

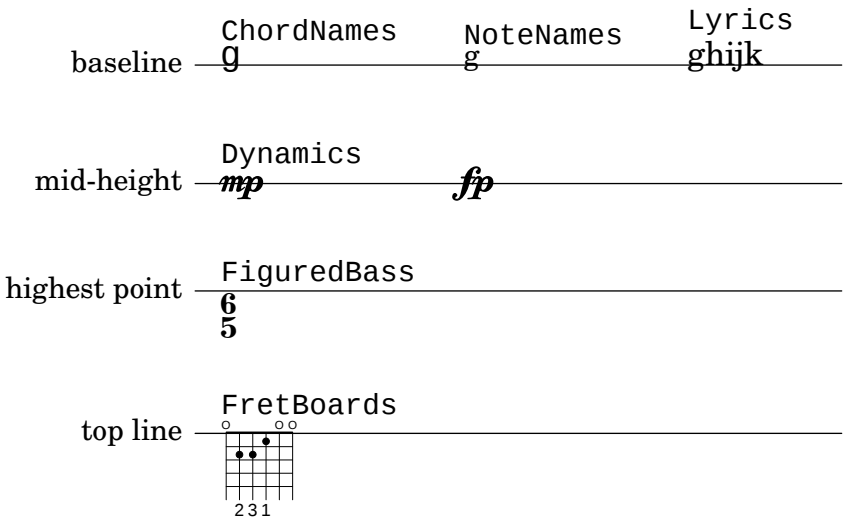
crea por parte de todas las pautas y líneas que no son pautas. El segundo conjunto está asociado con el grob `StaffGrouper`, que puede crearse por parte de los grupos de pentagramas, pero solamente si se le llama explícitamente. Estas propiedades se describen individualmente al final de la presente sección.

Los nombres de estas propiedades (excepto para `staff-affinity`) siguen el formato *`elemento1`*`-`*`elemento2`*`-spacing`, donde *`elemento1`* y *`elemento2`* son los elementos que se van a espaciar. Observe que *`elemento2`* no está necesariamente por debajo de *`elemento1`*; por ejemplo, `nonstaff-relatedstaff-spacing` mide hacia arriba a partir de la línea del tipo no-pauta si `staff-affinity` es `UP`.

Cada distancia se mide entre los *puntos de referencia* de los dos elementos. El *punto de referencia* para un pentagrama y otro tipo de pauta es el centro vertical de su `StaffSymbol` (es decir, la línea central si `line-count` es un número impar; el espacio central si `line-count` es par). Los puntos de referencia para las líneas individuales que no son pautas aparecen en la tabla siguiente:

Línea que no es una pauta	**Punto de referencia**
`ChordNames`	línea de base
`NoteNames`	línea de base
`Lyrics`	línea de base
`Dynamics`	la mitad de la altura de la 'm'
`FiguredBass`	punto más alto
`FretBoards`	línea superior

En la siguiente imagen, las líneas horizontales indican las posiciones de estos puntos de referencia:



Cada una de las propiedades de grob del espaciado vertical (excepto `staff-affinity`) usa la misma estructura de lista-A que las variables de espaciado del bloque `\paper` que se han estudiado en Sección 4.1.4 [Variables de espaciado de \paper verticales flexibles], página 517. En Sección 5.3.6 [Modificación de las listas-A], página 597 se estudian métodos específicos para la modificación de las listas-A. Las propiedades de los grobs deben ajustarse con un `\override` dentro de un bloque `\score` o `\layout`, y no dentro de un bloque `\paper`.

El ejemplo que sigue muestra las dos maneras en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor de forma individual, y la segunda redefine la propiedad completamente:

```
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 10)
       (minimum-distance . 9)
       (padding . 1)
       (stretchability . 10))
} { ... }
```

Para modificar globalmente cualquiera de los ajustes de espaciado, póngalos dentro del bloque `\layout`:

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
  }
}
```

Los ajustes estándar para las propiedades de grob del espaciado vertical están relacionadas en Sección "VerticalAxisGroup" in *Referencia de Funcionamiento Interno* y en Sección "StaffGrouper" in *Referencia de Funcionamiento Interno*. Las sobreescrituras predeterminadas para los tipos específicos de líneas que no son pautas están relacionadas en las descripciones de contexto correspondientes en Sección "Contexts" in *Referencia de Funcionamiento Interno*.

Propiedades del grob `VerticalAxisGroup`

Las propiedades de `VerticalAxisGroup` se suelen ajustar con una instrucción `\override` en el nivel de `Staff` (o equivalente).

`staff-staff-spacing`
: Se utiliza para determinar la distancia entre la pauta actual y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si hay más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica a la pauta inferior de un sistema.

  Inicialmente, el valor de espaciado `staff-staff-spacing` de un grupo `VerticalAxisGroup` es una función de Scheme que aplica las propiedades del `StaffGrouper` si la pauta forma parte de un grupo, o el `default-staff-staff-spacing` de la pauta en caso contrario. Esto hace posible que las pautas estén espaciadas de forma diferente cuando están agrupadas. Para conseguir un espaciado regular sin importar el agrupamiento, esta función se puede sustituir por una lista-A de espaciado flexible, utilizando la forma de sobreescritura de redefinición completa que se ha mostrado más arriba.

`default-staff-staff-spacing`
: Una lista-A de espaciado flexible que define el espacio `staff-staff-spacing` que se usará para las pautas sin agrupar, a no ser que se haya sobreescrito explícitamente `staff-staff-spacing` con una instrucción `\override`.

`staff-affinity`
: Dirección de la pauta que utilizar para el espaciado de la línea actual que no es pauta. Se puede elegir entre `UP` (arriba), `DOWN` (abajo) y `CENTER` (centro). Si es `CENTER`, la línea del tipo no-pauta se coloca equidistante entre las dos pautas contiguas a ambos

lados, a no ser que lo impidan colisiones u otras restricciones de espaciado. Las líneas del tipo no-pauta adyacentes han de llevar valores de `staff-affinity` no crecientes desde la parte superior hasta la inferior, p.ej. una línea del tipo no-pauta establecida a `UP` no debería seguir inmediatamente a otra que esté establecida a `DOWN`. Las líneas del tipo no-pauta en la parte superior de un sistema deben usar `DOWN`; las de la parte inferior deben usar `UP`. El ajuste de `staff-affinity` para una pauta hace que se le trate como una línea que no es una pauta. El establecimiento de `staff-affinity` a `#f` para una línea que no es una pauta hace que se trate como una pauta. Establecer `staff-affinity` a `UP`, `CENTER`, o `DOWN` hace que el pentagrama resulte espaciado como si fuera una línea que no es una pauta.

`nonstaff-relatedstaff-spacing`
: Distancia entre la línea del tipo no-pauta actual y la pauta más próxima en la dirección de `staff-affinity`, si no hay ninguna línea que no sea una pauta entre las dos, y `staff-affinity` es `UP` o `DOWN`. Si `staff-affinity` es `CENTER`, entonces se usa `nonstaff-relatedstaff-spacing` para las pautas más próximas *a los dos lados*, incluso aunque aparezcan otras líneas del tipo no-pauta entre la pauta actual y una de las otras. Esto significa que la colocación de una línea que no es una pauta depende tanto de las pautas que la rodean como de las líneas que no son pautas y que la rodean. Establecer la ampliabilidad `stretchability` de uno de estos tipos de espaciado a un valor más pequeño provoca que este espaciado domine. El establecimiento de `stretchability` a un valor mayor hace que dicho espaciado tenga un menor efecto.

`nonstaff-nonstaff-spacing`
: Distancia entre la línea actual del tipo no-pauta y la siguiente línea del tipo no-pauta en la dirección de `staff-affinity`, si las dos están al mismo lado de la pauta relacionada, y `staff-affinity` es `UP` o `DOWN`.

`nonstaff-unrelatedstaff-spacing`
: Distancia entre la línea actual del tipo no-pauta y la pauta en la dirección opuesta de `staff-affinity`, si no hay ninguna otra línea del tipo no-pauta entre las dos, y `staff-affinity` es `UP` o `DOWN`. Se puede usar, por ejemplo, para requerir una cantidad de relleno mínima entre una línea de `Lyrics` y la pauta a la que no pertenece.

Propiedades del grob `StaffGrouper`

Las propiedades de `StaffGrouper` se suelen ajustar con una instrucción `\override` en el nivel de `StaffGroup` (o equivalente).

`staff-staff-spacing`
: Distancia entre pautas consecutivas dentro del grupo de pautas actual. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobreescribir con distintos ajustes de espaciado para dicha pauta.

`staffgroup-staff-spacing`
: Distancia entre la última pauta del grupo en curso y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si existen una o más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica al pentagrama o pauta inferior de un sistema. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobreescribir con distintos ajustes de espaciado para dicha pauta.

Véase también

Referencia de la notación: Sección 4.1.4 [Variables de espaciado de `\paper` verticales flexibles], página 517, Sección 5.3.6 [Modificación de las listas-A], página 597.

Archivos de inicio: '`ly/engraver-init.ly`', '`scm/define-grobs.scm`'.

Referencia de funcionamiento interno: Sección "Contexts" in *Referencia de Funcionamiento Interno*, Sección "VerticalAxisGroup" in *Referencia de Funcionamiento Interno*, Sección "StaffGrouper" in *Referencia de Funcionamiento Interno*.

Espaciado de pautas no agrupadas

Las *pautas*, tales como los pentagramas (`Staff`), pautas de percusión (`DrumStaff`) o de tablatura (`TabStaff`), etc. son contextos que pueden contener uno o más contextos de voz, pero no pueden contener otras pautas.

Las siguientes propiedades afectan al espaciado de las pautas *no agrupadas*:

- Propiedades de `VerticalAxisGroup`:
  - `default-staff-staff-spacing`
  - `staff-staff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 536.

Están implicadas ciertas propiedades adicionales para las pautas que son parte de un grupo; véase [Espaciado de pautas agrupadas], página 541.

El ejemplo siguiente muestra cómo la propiedad `default-staff-staff-spacing` puede afectar al espaciado de pautas no agrupadas. Las mismas sobreescrituras aplicadas a `staff-staff-spacing` tendrían el mismo efecto, pero también se aplicaría en caso de que las pautas estuvieran combinadas en un grupo o grupos.

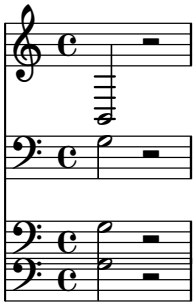
```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 8)
         (minimum-distance . 7)
         (padding . 1))
  }
}

<<
  % The very low note here needs more room than 'basic-distance
  % can provide, so the distance between this staff and the next
  % is determined by 'padding.
  \new Staff { b,2 r | }

  % Here, 'basic-distance provides enough room, and there is no
  % need to compress the space (towards 'minimum-distance) to make
  % room for anything else on the page, so the distance between
  % this staff and the next is determined by 'basic-distance.
  \new Staff { \clef bass g2 r | }

  % By setting 'padding to a negative value, staves can be made to
  % collide.  The lowest acceptable value for 'basic-distance is 0.
  \new Staff \with {
```

```
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 3.5)
         (padding . -10))
  } { \clef bass g2 r | }
  \new Staff { \clef bass g2 r | }
>>
```



Véase también

Archivos de inicio: ‘`scm/define-grobs.scm`’.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*.

Espaciado de pautas agrupadas

En partituras grandes como las orquestales, es común colocar los pentagramas en grupos. El espacio entre los grupos suele ser mayor que el espacio que hay entre los pentagramas dentro del mismo grupo.

Los grupos de pautas, *Staff-groups* (tales como `StaffGroup`, `ChoirStaff`, etc.) son contextos que pueden contener al mismo tiempo uno o más pentagramas o pautas.

Las siguientes propiedades afectan al espaciado de las pautas dentro de los grupos:

- Propiedades de `VerticalAxisGroup`:
  - `staff-staff-spacing`
- Propiedades de `StaffGrouper`:
  - `staff-staff-spacing`
  - `staffgroup-staff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 536.

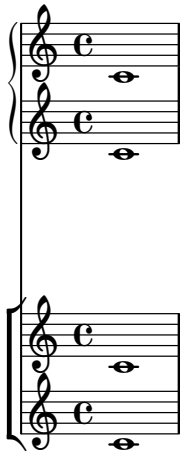
El ejemplo siguiente muestra cómo pueden afectar las propiedades del grob `StaffGrouper` al espaciado de las pautas agrupadas:

```
\layout {
  \context {
    \Score
    \override StaffGrouper.staff-staff-spacing.padding = #0
    \override StaffGrouper.staff-staff-spacing.basic-distance = #1
  }
}

<<
```

```
  \new PianoStaff \with {
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #20
  } <<
    \new Staff { c'1 }
    \new Staff { c'1 }
  >>

  \new StaffGroup <<
    \new Staff { c'1 }
    \new Staff { c'1 }
  >>
>>
```



Véase también

Archivos de inicio: ‘`scm/define-grobs.scm`’.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*, Sección “StaffGrouper” in *Referencia de Funcionamiento Interno*.

Espaciado de las líneas que no son pautas

Las *Líneas que no son pautas* (tales como `Lyrics`, `ChordNames`, etc.) son contextos cuyos objetos de presentación se imprimen como pentagramas (es decir, en líneas horizontales dentro de los sistemas). Específicamente, las líneas que no son pautas son contextos del tipo no-pauta que crean el objeto de presentación `VerticalAxisGroup`.

Las siguientes propiedades afectan al espaciado de las líneas que no son pautas:

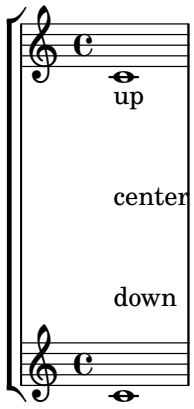
- Propiedades de `VerticalAxisGroup`:
  - `staff-affinity`
  - `nonstaff-relatedstaff-spacing`
  - `nonstaff-nonstaff-spacing`
  - `nonstaff-unrelatedstaff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 536.

El ejemplo siguiente muestra cómo la propiedad `nonstaff-nonstaff-spacing` puede afectar el espaciado de líneas consecutivas que no son pautas. Aquí, mediante el establecimiento de la clave de ampliabilidad `stretchability` a un valor muy grande, la línea de letra es capaz de ampliarse mucho más de lo que es usual:

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.nonstaff-nonstaff-spacing.stretchability = #1000
  }
}

\new StaffGroup
<<
  \new Staff \with {
    \override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 30))
  } { c'1 }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #UP
  } \lyricmode { up }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #CENTER
  } \lyricmode { center }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #DOWN
  } \lyricmode { down }
  \new Staff { c'1 }
>>
```



Véase también

Archivos de inicio: ‘`ly/engraver-init.ly`’, ‘`scm/define-grobs.scm`’.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Contexts” in *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*.

4.4.2 Posicionamiento explícito de los pentagramas y los sistemas

Una forma de entender los mecanismos del espaciado vertical flexible que se han explicado más arriba es como una colección de ajustes que controlan la cantidad de relleno vertical entre los pentagramas y los sistemas.

Es posible enfrentarse al espaciado vertical de una forma distinta utilizando `NonMusicalPaperColumn.line-break-system-details`. Mientras que los mecanismos del espaciado vertical flexible especifican relleno vertical,

`NonMusicalPaperColumn.line-break-system-details` puede especificar posiciones verticales exactas sobre la página.

`NonMusicalPaperColumn.line-break-system-details` acepta una lista asociativa de tres ajustes diferentes:

- `X-offset` (desplazamiento en X)
- `Y-offset` (desplazamiento en Y)
- `alignment-distances` (distancias de alineación)

Las sobreescrituras de los objetos gráficos, entre ellas las de `NonMusicalPaperColumn` que aparecen más abajo, pueden ocurrir en tres lugares distintos dentro de un archivo de entrada:

- directamente en medio de las notas
- en un bloque `\context`
- en el bloque `\with`

Cuando sobreescribimos `NonMusicalPaperColumn`, usamos la instrucción `\override` usual en los bloques `\context` y en el bloque `\with`. Por otra parte, cuando sobreescribimos `NonMusicalPaperColumn` en medio de las notas, debemos usar la instrucción especial `\overrideProperty`. He aquí algunas sobreescrituras de `NonMusicalPaperColumn` de ejemplo con la instrucción especial `\overrideProperty`:

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20))

\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((Y-offset . 40))

\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
     (Y-offset . 40))
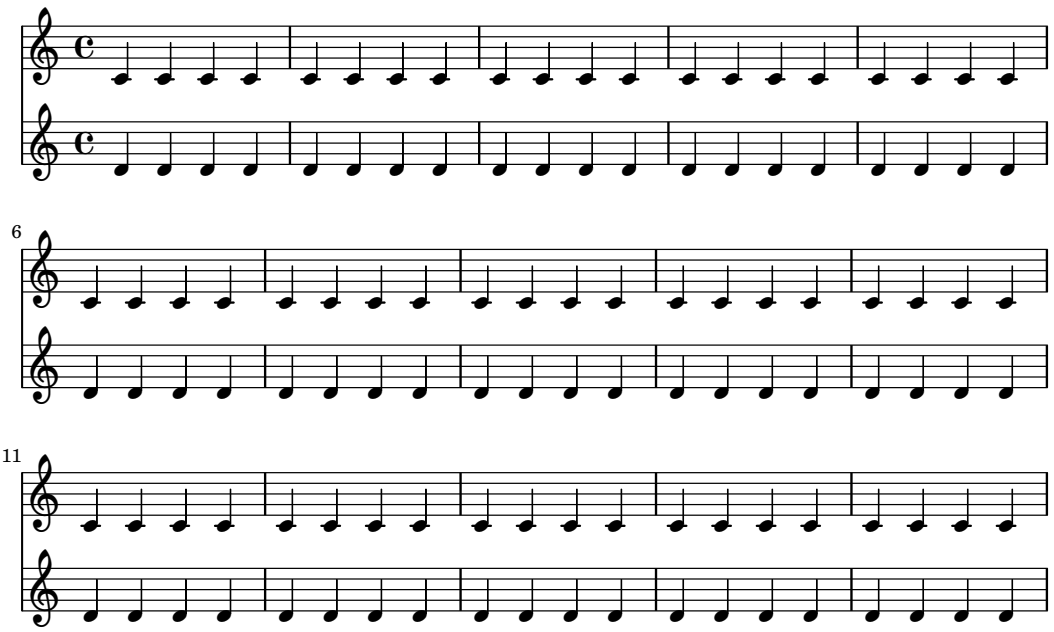
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((alignment-distances . (15)))

\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
     (Y-offset . 40)
     (alignment-distances . (15)))
```

Para entender cómo funciona cada uno de los distintos ajustes, empezamos observando un ejemplo que no incluye absolutamente ninguna sobreescritura.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
```

```
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```

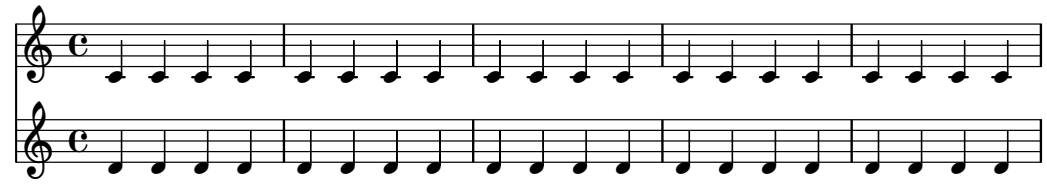


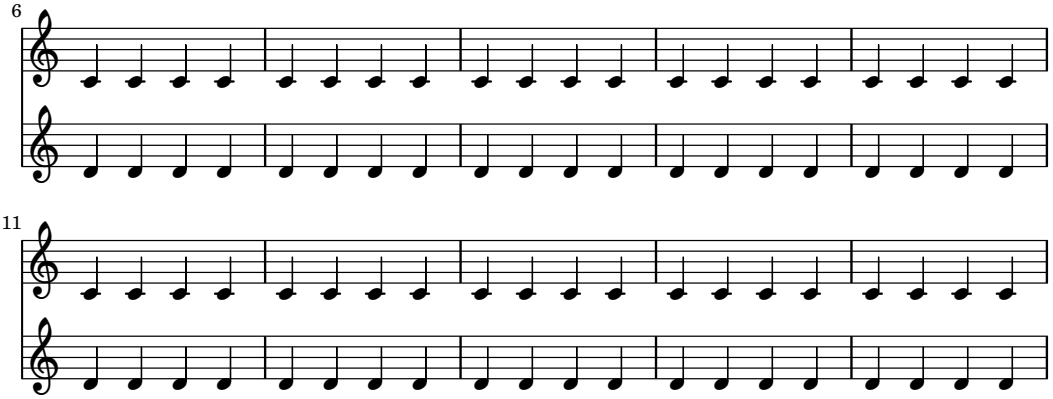
Esta partitura mantiene aislada la información de los saltos de línea y de página en una voz dedicada a ello. Esta técnica de crear una voz de saltos ayuda a mantener separada de la escritura de notas conforme el ejemplo se vuelve más complicado. Consulte Sección 4.3.8 [Utilizar una voz adicional para los saltos de línea], página 534.

Los saltos explícitos dividen la música de forma regular en seis compases por línea. El espaciado vertical es el resultado de los ajustes predeterminados de LilyPond. Para establecer explícitamente el punto de origen vertical de cada sistema, podemos establecer el par `Y-offset` en el atributo `line-break-system-details` del grob (objeto gráfico) `NonMusicalPaperColumn`:

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 0))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 40))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 60))
          s1*5 \break
        }
```

```
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```
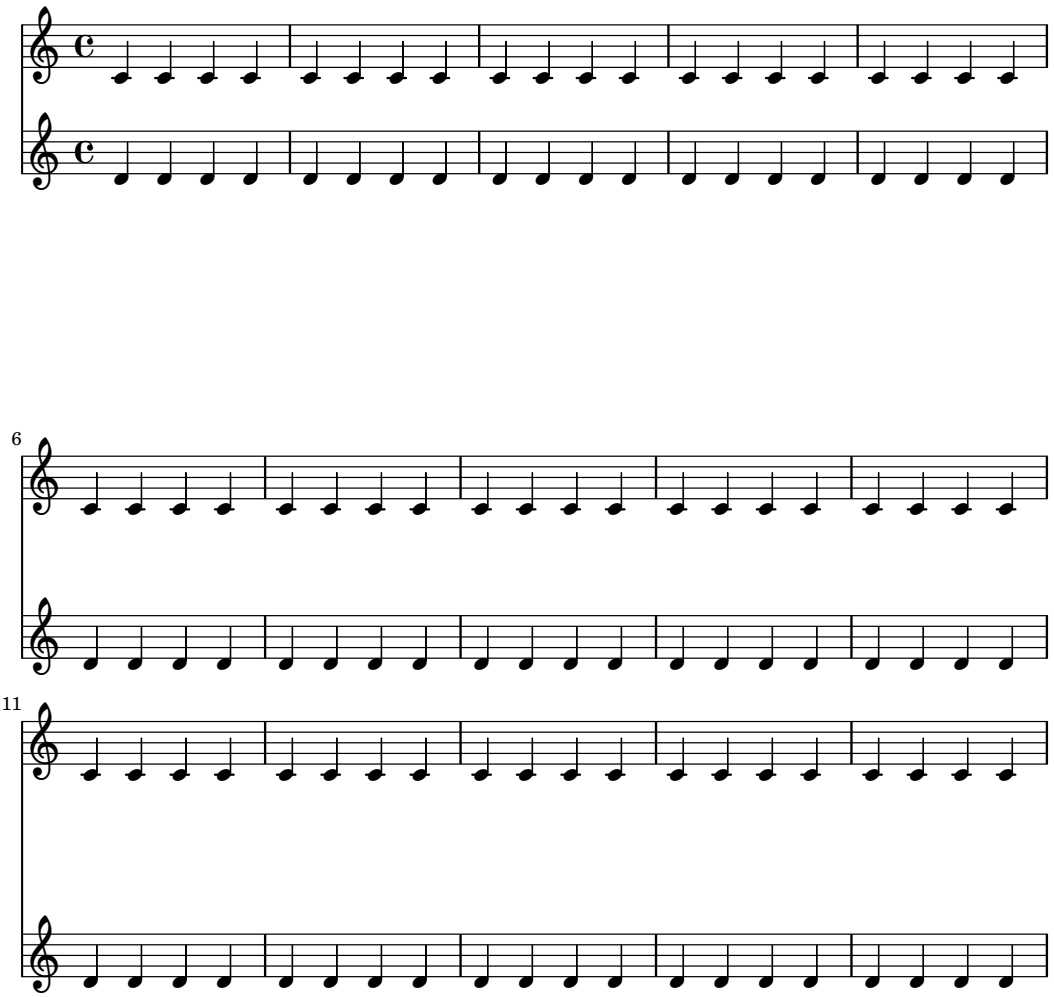




Observe que `line-break-system-details` toma una lista asociativa de una cantidad de valores potencialmente elevada, pero aquí sólo fijamos un valor. Observe también que aquí la propiedad `Y-offset` determina la posición vertical exacta sobre la página en la que se trazará cada uno de los nuevos sistemas.

Ahora que hemos establecido explícitamente el punto de origen vertical de cada sistema, podemos también establecer manualmente las distancias verticales entre los pentagramas dentro de cada sistema. Lo hacemos usando la subpropiedad `alignment-distances` de `line-break-system-details`.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 20)
               (alignment-distances . (10)))
```

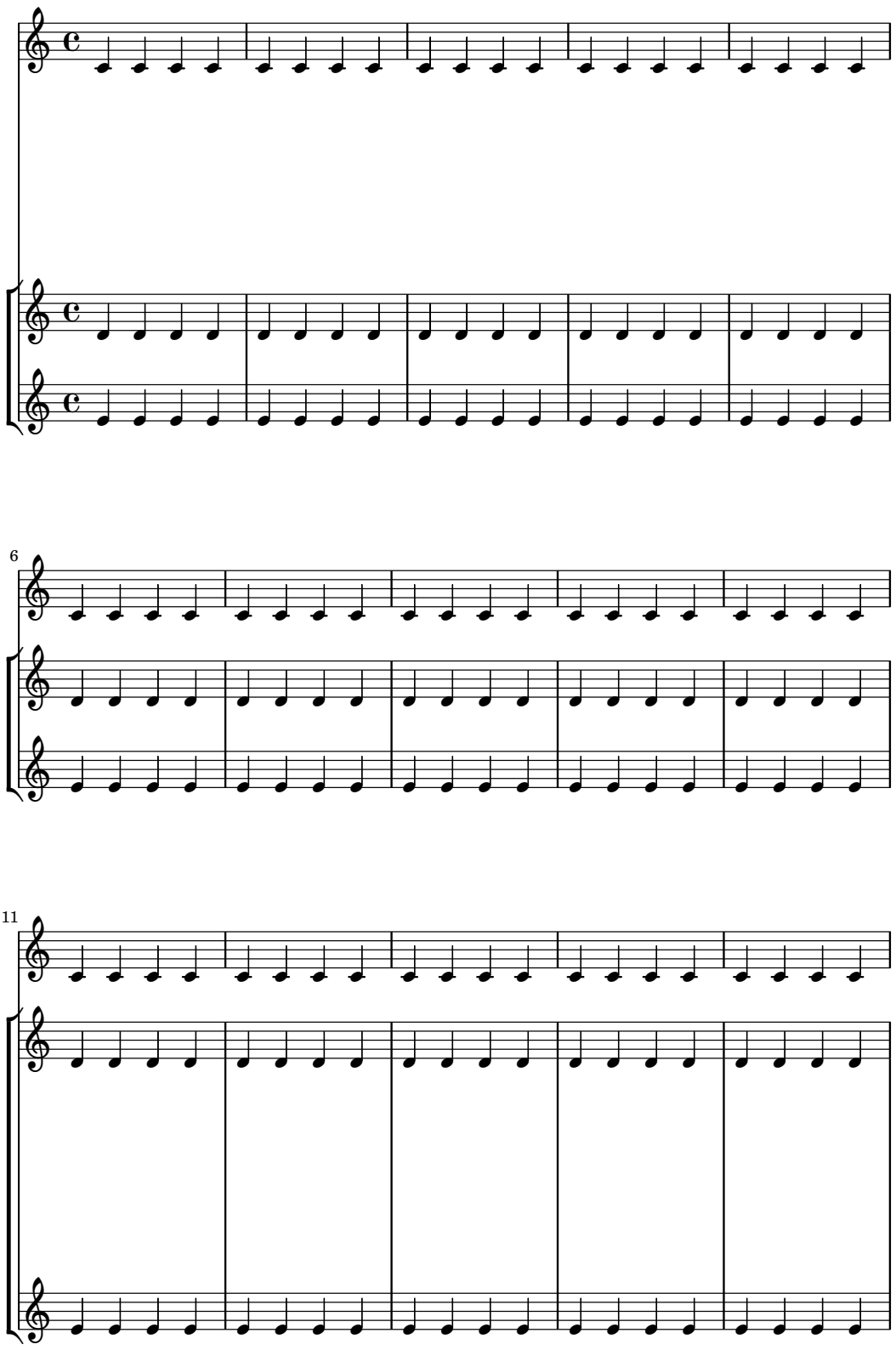
```
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 60)
               (alignment-distances . (15)))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 85)
             (alignment-distances . (20)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Observe que aquí asignamos dos valores distintos al atributo `line-break-system-details` del grob `NonMusicalPaperColumn`. Aunque el atributo alist de `line-break-system-details` acepta muchos parámetros de espaciado adicionales (entre ellos, por ejemplo, un par `X-offset` correspondiente), sólo tenemos que establecer el desplazamiento `Y-offset` y los pares `alignment-distances` para controlar el punto de origen vertical de cada sistema y pentagrama. Finalmente, observe que `alignment-distances` especifica el posicionamiento vertical de los pentagramas pero no de los grupos de pentagramas.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 0)
               (alignment-distances . (30 10)))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
```

```
            #'((Y-offset . 60)
               (alignment-distances . (10 10)))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((Y-offset . 100)
               (alignment-distances . (10 30)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new StaffGroup <<
        \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
        \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
      >>
    >>
  }
}
```



Algunos puntos que tener en cuenta:

- Al usar `alignment-distances`, la letra y otras líneas que no son pautas, no cuentan como un pentagrama.
- Las unidades de los números que se pasan a `X-offset`, `Y-offset` y `alignment-distances` se interpretan como múltiplos de la distancia entre líneas de pentagrama adyacentes. Los valores positivos mueven a los pentagramas y a la letra hacia arriba, los valores negativos mueven los pentagramas y la letra hacia abajo.
- A causa de que los ajustes a `NonMusicalPaperColumn.line-break-system-details` dados

aquí permiten el posicionamiento de pentagramas y sistemas en cualquier lugar de la página, es posible violar los márgenes o los límites del papel, o incluso imprimir pentagramas o sistemas unos encima de otros. Esto se evitaría pasando a los diferentes ajustes unos valores razonables.

Véase también

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

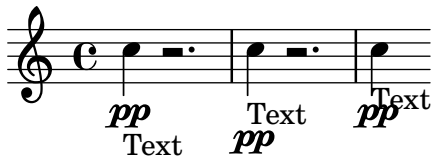
4.4.3 Evitar las colisiones verticales

Podemos decir intuitivamente que algunos objetos de la notación musical pertenecen al pentagrama y otros se sitúan fuera del pentagrama. Entre los objetos que pertenecen al exterior del pentagrama están las marcas de ensayo, las marcas textuales y las indicaciones de dinámica (en lo sucesivo, les llamaremos objetos fuera del pentagrama). La regla de LilyPond para la colocación vertical de los objetos fuera del pentagrama es ponerlos tan cerca del pentagrama como sea posible pero no tan próximos como para que choquen con otro objeto.

LilyPond utiliza la propiedad `outside-staff-priority` para determinar si un grob es un objeto fuera del pentagrama: si `outside-staff-priority` es un número, el grob es un objeto fuera del pentagrama. Además, `outside-staff-priority` informa a LilyPond en qué orden se debe situar los objetos.

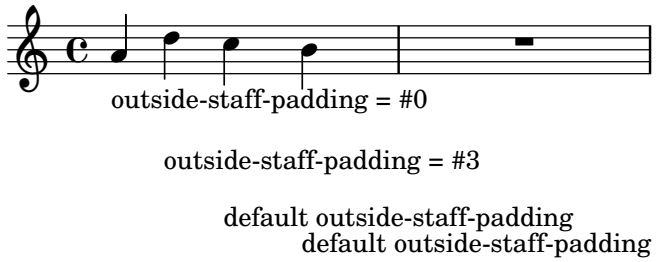
En primer lugar, LilyPond sitúa todos los objetos que no pertenecen al exterior del pentagrama. Después ordena los objetos fuera del pentagrama de acuerdo a su prioridad `outside-staff-priority` (en orden creciente). LilyPond toma los objetos fuera del pentagrama uno a uno y los coloca de forma que no choquen con ningún objeto que ya haya sido colocado. Esto es, si dos grobs fuera del pentagrama compiten por el mismo espacio, el que tiene la prioridad `outside-staff-priority` más baja se colocará más próximo al pentagrama.

```
c4_"Text"\pp
r2.
\once \override TextScript.outside-staff-priority = #1
c4_"Text"\pp % this time the text will be closer to the staff
r2.
% by setting outside-staff-priority to a non-number,
% we disable the automatic collision avoidance
\once \override TextScript.outside-staff-priority = ##f
\once \override DynamicLineSpanner.outside-staff-priority = ##f
c4_"Text"\pp % now they will collide
```



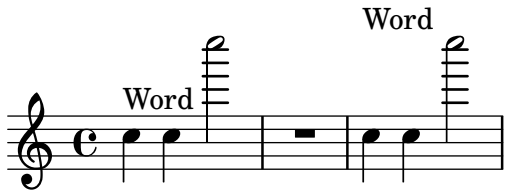
El relleno vertical que rodea a los objetos fuera del pentagrama se puede controlar con `outside-staff-padding`.

```
\once \override TextScript.outside-staff-padding = #0
a4-"outside-staff-padding = #0"
\once \override TextScript.outside-staff-padding = #3
d-"outside-staff-padding = #3"
c-"default outside-staff-padding"
b-"default outside-staff-padding"
R1
```



De forma predeterminada, los objetos fuera del pentagrama se colocan de forma que eviten una colisión horizontal con los grobs posicionados previamente. Esto puede llevar a situaciones en las que los objetos se colocan muy próximos entre sí en el sentido horizontal. Como se muestra en el ejemplo que aparece más abajo, al establecer `outside-staff-horizontal-padding` se incrementa el espacio horicontal necesario, y en este caso mueve el texto hacia arriba para evitar que quede demasiado cerca de las líneas adicionales.

```
c4^"Word" c c''2
R1
\once \override TextScript.outside-staff-horizontal-padding = #1
c,,4^"Word" c c''2
```



Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.
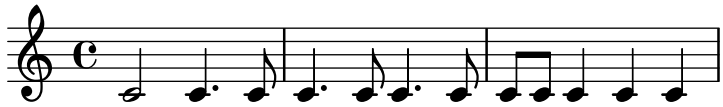
4.5 Espaciado horizontal

4.5.1 Panorámica del espaciado horizontal

El motor de espaciado traduce las diferencias en las duraciones a distancias ampliables (‘muelles’) de distintas longitudes. Las duraciones más largas reciben un espacio mayor y las duraciones más cortas reciben menos. Las duraciones más breves reciben un espacio de tamaño fijo (que se controla mediante `shortest-duration-space` en el objeto Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*). Cuanto más larga es la duración, más espacio recibe: al doblar una duración se añade un espacio de tamaño fijo (este tamaño se controla mediante `spacing-increment`) a la nota.

Por ejemplo, la pieza siguiente contiene numerosas blancas, negras y corcheas; la corchea va seguida de un ancho de cabeza de blanca (ACB). La negra va seguida de 2 ACB, la blanca por 3 ACB, etc.

```
c2 c4. c8
c4. c8 c4. c8
c8 c c4 c c
```



Normalmente, el valor de `spacing-increment` está establecido en 1.2 espacios de pentagrama, lo que es aproximadamente el ancho de una cabeza, y `shortest-duration-space` está establecido en 2.0, lo que significa que la nota más breve recibe 2.4 espacios de pentagrama (2.0

multiplicado por el `spacing-increment`) de espacio horizontal. Este espacio se cuenta a partir del borde izquierdo del símbolo, de manera que las notas más breves van seguidas generalmente por un espacio de 1 ACB.
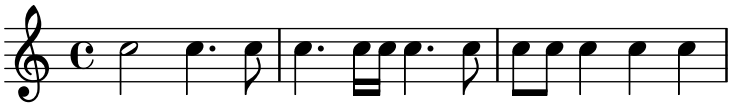
Si siguiésemos el procedimiento anterior exactamente, entonces la adición de una sola fusa a una partitura que usa corcheas y semicorcheas, aumentaría enormemente la anchura de la partitura completa. La nota más breve ya no es la semicorchea, sino la fusa, añadiendo así 1 ACB a cada una de las notas. Para evitarlo, la duración más breve a efectos de espaciado no es la nota más breve de la partitura, sino la que ocurre con más frecuencia.

La duración más breve que es más común se determina de la siguiente manera: en cada compás se determina la duración más breve. La menor duración más común se toma como base para el espaciado, habiendo estipulado que esta duración menor siempre debe ser igual o menor que una corchea. La duración más breve se imprime cuando se ejecuta `lilypond` con la opción '`--verbose`'.
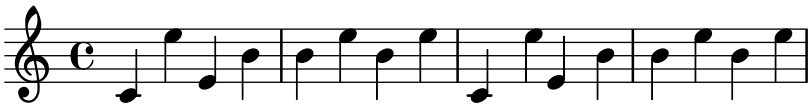
Estas duraciones también se pueden personalizar. Si establecemos la `common-shortest-duration` en Sección "SpacingSpanner" in *Referencia de Funcionamiento Interno*, entonces éste establece la duración base para el espaciado. La duración máxima para esta base (normalmente una corchea), se fija a través de `base-shortest-duration`.

Las notas que son aún más breves que la nota común más breve van seguidas por un espacio proporcional a su duración en relación con la nota común más breve. Así pues, si fuésemos a añadir tan sólo algunas semicorcheas al ejemplo anterior, irían seguidas por medio ACB:

```
c2 c4. c8 c4. c16[ c] c4. c8 c8 c8 c4 c4 c4
```



En el *Ensayo sobre grabado musical automatizado*, se explicó que las direcciones de las plicas influyen en el espaciado (véase Sección "Espaciado óptico" in *Ensayo*). Esto se controla con la propiedad `stem-spacing-correction` en el objeto Sección "NoteSpacing" in *Referencia de Funcionamiento Interno*. Estos se generan para cada uno de los contextos de Sección "Voice" in *Referencia de Funcionamiento Interno*. El objeto `StaffSpacing` (generado en el contexto de Sección "Staff" in *Referencia de Funcionamiento Interno*) contiene la misma propiedad para controlar el espaciado de las líneas de plica o divisorias. El ejemplo siguiente muestra estas correcciones, una vez con los valores predeterminados y otra con correcciones exageradas:



Está contemplada la notación proporcional; consulte Sección 4.5.5 [Notación proporcional], página 557.

Véase también

Ensayo sobre grabado musical automatizado: Sección "Espaciado óptico" in *Ensayo*.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "SpacingSpanner" in *Referencia de Funcionamiento Interno*, Sección "NoteSpacing" in *Referencia de Funcionamiento Interno*, Sección "StaffSpacing" in *Referencia de Funcionamiento Interno*, Sección "NonMusicalPaperColumn" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe ningún mecanismo conveniente para sobreescribir el espaciado manualmente. Se puede usar el siguiente rodeo para insertar espacio adicional en una partitura, ajustando el valor del relleno en la medida necesaria.

```
\override Score.NonMusicalPaperColumn.padding = #10
```
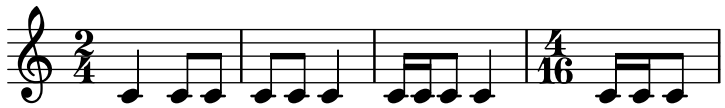
No existe ningún rodeo para disminuir la magnitud de la separación.

4.5.2 Área de espaciado nueva

Se pueden iniciar secciones nuevas con diferentes parámetros de espaciado, con `newSpacingSection`. Esto es útil cuando hay secciones que tienen distinta noción de las notas largas y cortas.
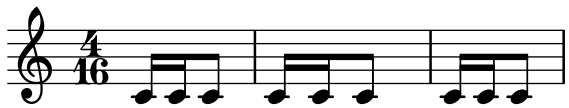
En el ejemplo siguiente, el cambio de compás introduce una sección nueva, y por ello las semicorcheas se separan automáticamente de manera ligeramente más amplia.

```
\time 2/4
c4 c8 c
c8 c c4 c16[ c c8] c4
\newSpacingSection
\time 4/16
c16[ c c8]
```



La instrucción `\newSpacingSection` crea un nuevo objeto `SpacingSpanner` en ese momento musical. Si los ajustes de espaciado automáticos no dan el espaciado deseado, pueden aplicarse sobreescrituras manuales a sus propiedades con `\override`. Las sobreescrituras se deben aplicar en el mismo momento musical que la propia instrucción `\newSpacingSection`. Entonces afectarán al espaciado de toda la música que venga a continuación hasta que las propiedades se cambien en una nueva sección de espaciado, por ejemplo:

```
\time 4/16
c16[ c c8]
\newSpacingSection
\override Score.SpacingSpanner.spacing-increment = #2
c16[ c c8]
\newSpacingSection
\revert Score.SpacingSpanner.spacing-increment
c16[ c c8]
```



Véase también
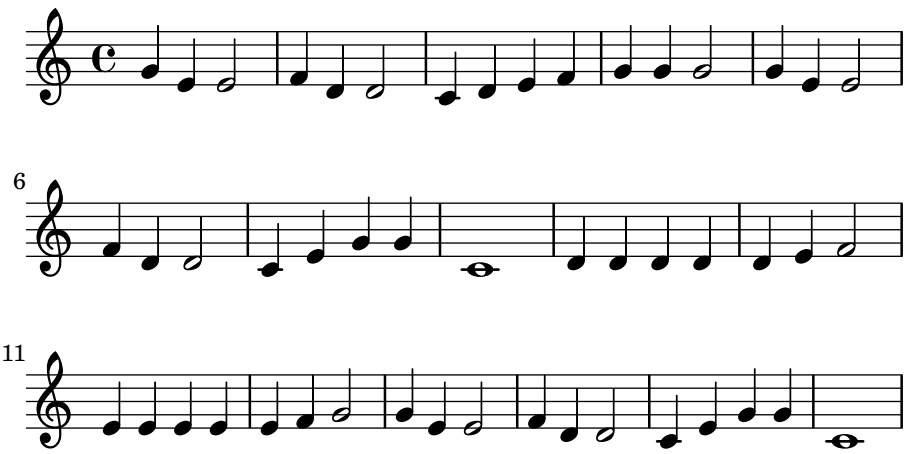
Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*.

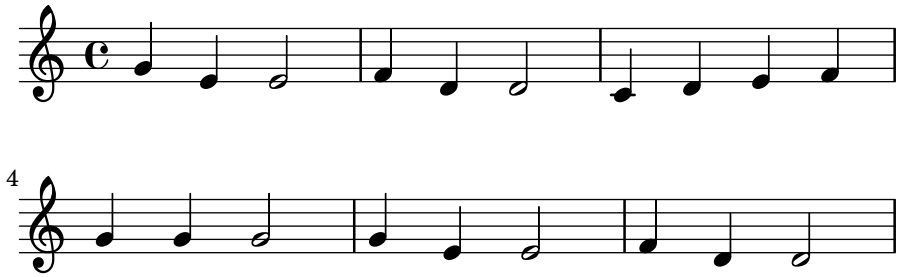
4.5.3 Cambiar el espaciado horizontal
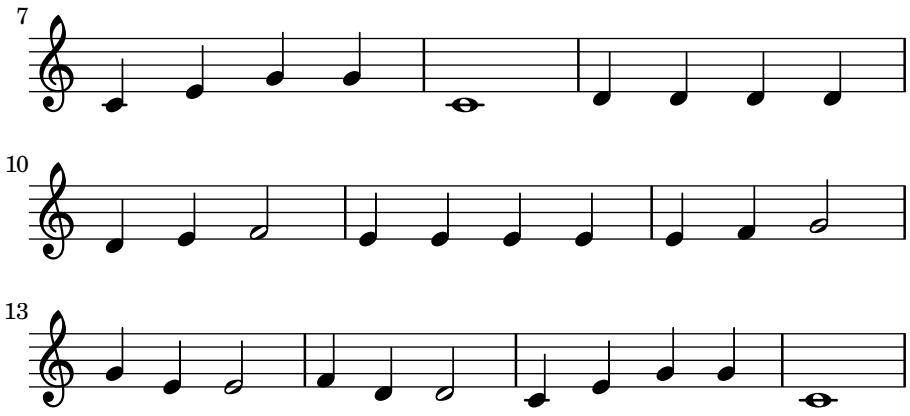
Se puede alterar el espaciado horizontal con la propiedad `base-shortest-duration`. Aquí compararemos la misma música, una vez sin alterar la propiedad, y luego alterándola. Los valores mayores de `ly:make-moment` producen música más pequeña. Observe que `ly:make-moment` construye una duración, por lo que `1 4` es una duración mayor que `1 16`.

```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```



```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/16)
    }
  }
}
```

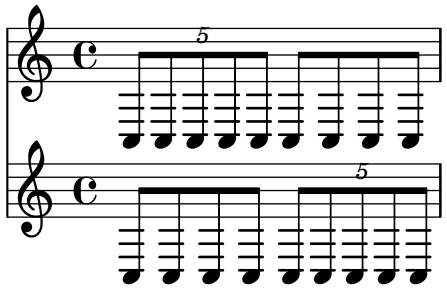




Fragmentos de código seleccionados

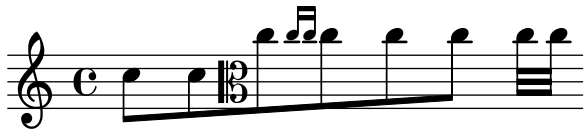
De forma predeterminada, el espaciado en los grupos de valoración especial depende de varios factores ajenos a la duración (como alteraciones, cambios de clave, etc.). Para pasar por alto estos símbolos y forzar un espaciado de duraciones iguales uniforme, use `Score.SpacingSpanner.uniform-stretching`. Esta propiedad sólo puede cambiarse al principio de la partitura:

```
\score {
  <<
    \new Staff {
      \tuplet 5/4 { c8 c c c c } c8 c c c
    }
    \new Staff {
      c8 c c c \tuplet 5/4 { c8 c c c c }
    }
  >>
  \layout {
    \context {
      \Score
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```



Cuando se establece `strict-note-spacing`, las notas se separan sin tener en cuenta las claves, líneas divisorias ni notas de adorno:

```
\override Score.SpacingSpanner.strict-note-spacing = ##t
\new Staff { c8[ c \clef alto c \grace { c16 c } c8 c c]  c32[ c] }
```



Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.5.4 Longitud de la línea

Los ajustes más básicos que influyen en el espaciado son `indent` y `line-width`. Se definen en el bloque `\layout`. Controlan el sangrado de la primera línea de música y la longitud de las líneas.

Si se fija un valor verdadero para `ragged-right` en el bloque `\layout`, entonces los sistemas terminan en su longitud horizontal natural, en lugar de repartirse horizontalmente hasta llenar toda la línea. Esto es útil para fragmentos cortos, y para comprobar qué tan apretado es el espaciado natural. El valor normal predeterminado es falso, pero si la partitura tiene un solo sistema, el valor predeterminado es verdadero.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza. No se efectúa ninguna restricción sobre dicha línea. El resultado es similar al formateo de los párrafos de texto. En un párrafo, la última línea sencillamente ocupa su longitud horizontal natural.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.5.5 Notación proporcional

LilyPond contempla la notación proporcional, un tipo de espaciado horizontal en el que cada nota consume una medida horizontal que equivale exactamente a su duración rítmica. Este tipo de espaciado proporcional es comparable al espaciado horizontal hecho sobre un papel milimetrado. Ciertas partituras del finales del s.XX y principios del s.XXI utilizan notación proporcional para clarificar relaciones rítmicas complejas o para facilitar la colocación de líneas cronométricas u otros gráficos directamente en la partitura.

LilyPond contempla cinco ajustes distintos para la notación proporcional, que se pueden usar solos o combinados:
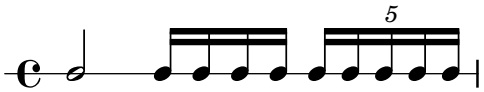
- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove "Separating_line_group_engraver"`
- `\override PaperColumn.used = ##t`

En los ejemplos que siguen, exploramos el uso de estos cinco ajustes de la notación proporcional y examinamos la forma en que interactúan.

Comenzamos con el siguiente ejemplo de un solo compás, que usa un espaciado clásico sin justificación por la derecha.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
  >>
```

```
}
```



Observe que la blanca que inicia el compás ocupa mucho menos de la mitad de todo el espacio horizontal del compás. De forma similar, las semicorcheas y el cinquillo de semicorcheas con que finaliza el compás ocupan en conjunto mucho más de la mitad de todo el espacio horizontal del compás.

En el grabado clásico, este espaciado puede ser exactamente el que deseamos porque podemos tomar prestado el espacio horizontal de la blanca y conservar el espacio horizontal a lo largo del compás como un todo.

Por otro lado, si queremos insertar una línea de tiempo graduada o algún otro gráfico encima o debajo de la partitura, necesitamos la notación proporcional. Se activa la notación proporcional con el ajuste `proportionalNotationDuration`.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
  >>
 \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
    }
  }
}
```



La blanca al principio del compás y las notas rápidas de la segunda mitad del compás ocupan ahora cantidades iguales de espacio horizontal. Podríamos colocar una línea de tiempo graduada o un gráfico encima o debajo de este ejemplo.

El ajuste `proportionalNotationDuration` es un ajuste de contexto que reside en `Score`. Recordemos que los ajustes de contexto aparecen en uno de tres posibles lugares del archivo de entrada: en un bloque `\with`, en un bloque `\context`, o directamente entre la música precedido por la instrucción `\set`. Como con todos los ajustes de contexto, el usuario puede elegir en cuál de los tres lugares diferentes prefiere establecer el valor de `proportionalNotationDuration`.

El ajuste `proportionalNotationDuration` acepta un solo argumento, que es la duración de referencia contra el que se aplica el espaciado de toda la música. La función Scheme de LilyPond `make-moment` acepta dos argumentos: un numerador y un denominador que, juntos, expresan una cierta fracción de redonda. La llamada `(ly:make-moment 1/20)`, por tanto, produce una duración de referencia de una nota de un veinteavo de redonda (semicorcheas de cinquillo). Son también posibles valores como `(ly:make-moment 1/16)`, `(ly:make-moment 1/8)` y `(ly:make-moment 3/97)`.

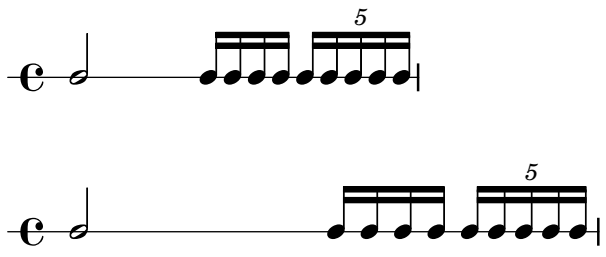
¿Cómo seleccionamos la duración correcta de referencia para pasarla a `proportionalNotationDuration`? Normalmente mediante un proceso de ensayo y error, comenzando con una duración cercana a la más rápida (o más breve) duración de

la pieza. Las duraciones de referencia más pequeñas aplican un espaciado más suelto; las duraciones de referencia más largas aplican un espaciado más apretado.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}

\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/16)
    }
  }
}

\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/32)
    }
  }
}
```



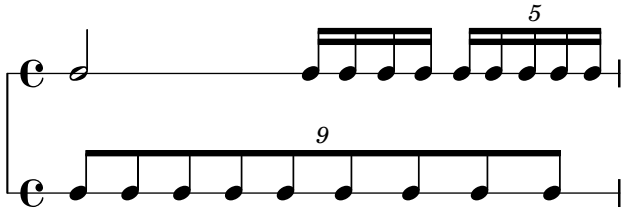


Observe que una duración de referencia demasiado grande (como la corchea, en el ejemplo de arriba) produce un espaciado excesivamente apretado y puede ser causa de colisiones entre las cabezas de nota. Observe también que la notación proporcional en general ocupa más espacio horizontal que el espaciado clásico. El espaciado proporcional aporta claridad rítmica a expensas del espacio horizontal.

Ahora veremos cómo espaciar de forma óptima grupos de valoración especial que se superponen.

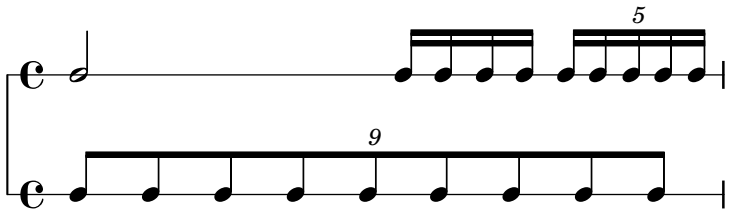
Empezamos por examinar qué le ocurre a nuestro ejemplo original, con espaciado clásico, cuando añadimos un segundo pentagrama con un tipo diferente de grupo especial.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c'8 c' c' c' c' c' c' c' c' }
    }
  >>
}
```



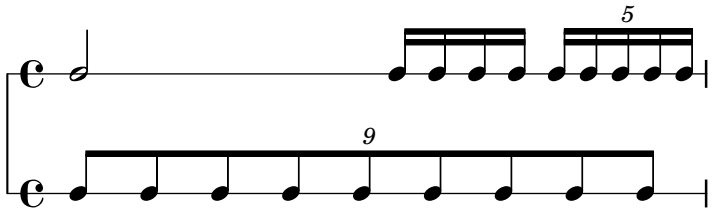
El espaciado es defectuoso porque las notas regularmente espaciadas del pentagrama inferior no se amplían de manera uniforme. Los grabados clásicos incluyen muy pocos tresillos complejos y así las reglas del grabado clásico pueden generar este tipo de resultado. El establecimiento de `proportionalNotationDuration` lo soluciona.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c'8 c' c' c' c' c' c' c' c' }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
    }
  }
}
```



Pero si observamos con mucho cuidado podremos ver que las notas de la segunda mitad del 9-illo están espaciadas de forma ligeramente más ancha que las de la primera mitad del 9-illo. Para asegurar una ampliación uniforme, activamos `uniform-stretching`, que es una propiedad de `SpacingSpanner`.

```
\score {
  <<
    \new RhythmicStaff {
      c'2 c'16 c' c' c' \tuplet 5/4 { c'16 c' c' c' c' }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c'8 c' c' c' c' c' c' c' c' }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```



Nuestro ejemplo de dos pentagramas ahora está espaciado exactamente, nuestras relaciones rítmicas son visualmente claras, y podemos incluir una línea de tiempo graduada o un gráfico, si queremos.

Observe que el paquete de notación proporcional de LilyPond espera que todas las partituras proporcionales establezcan el atributo `uniform-stretching` de `SpacingSpanner` al valor ##t. El establecimiento de `proportionalNotationDuration` sin ajustar también el atributo `uniform-stretching` de `SpacingSpanner` al valor ##t causará, por ejemplo, que los desplazamientos (`skips`) consuman una cantidad de espacio horizontal incorrecta.

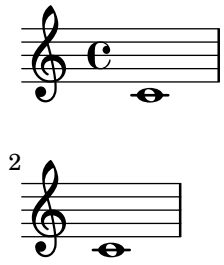
El `SpacingSpanner` es un grob abstracto que reside en el contexto de `Score`. Como con nuestros ajustes de `proportionalNotationDuration`, las sobreescrituras al `SpacingSpanner` sólo pueden ocurrir en uno de estos tres lugares dentro del archivo de entrada: en el bloque `\with`, en el bloque `\context` o directamente dentro de la escritura de notas.

De manera predeterminada, sólo hay un `SpacingSpanner` por `Score`. Esto supone que, por omisión, `uniform-stretching` está activado para la partitura completa o desactivado para la partitura completa. Sin embargo, podemos sobreescribir este comportamiento y activar distintas posibilidades de espaciado en distintos lugares de la partitura. Lo hacemos con la instrucción `\newSpacingSection`. Consulte Sección 4.5.2 [Área de espaciado nueva], página 554 para más información.

A continuación examinamos los efectos del grabador `Separating_line_group_engraver` y veremos por qué las partituras proporcionales con frecuencia eliminan este grabador. El ejemplo siguiente muestra que hay una pequeña cantidad de espacio "preliminar" justo antes de la primera nota de cada sistema.

```
\paper {
  indent = #0
}

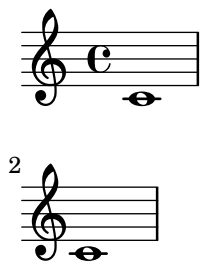
\new Staff {
  c'1
  \break
  c'1
}
```



Esta cantidad de espacio preliminar es la misma ya sea después de una indicación de compás, una armadura o una clave. El grabador `Separating_line_group_engraver` es responsable de este espacio. La eliminación de `Separating_line_group_engraver` reduce este espacio a cero.

```
\paper {
  indent = #0
}

\new Staff \with {
  \remove "Separating_line_group_engraver"
} {
  c'1
  \break
  c'1
}
```



Los elementos no musicales como la indicación de compás, la armadura, la clave y las alteraciones son problemáticos en notación proporcional. Ninguno de estos elementos tiene duración rítmica. Pero todos ellos consumen espacio horizontal. Las distintas partituras proporcionales abordan este problema de distinta manera.
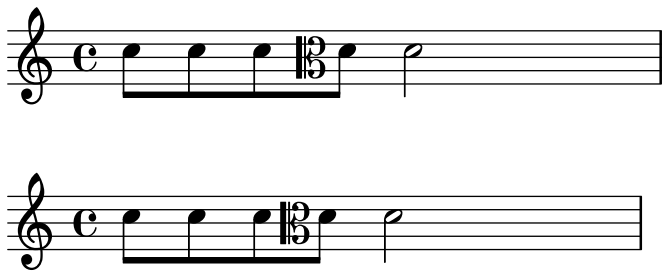
Sería posible evitar los problemas de espaciado con las armaduras, simplemente evitando tenerlas. Esta es una opción válida pues casi todas las partituras proporcionales son música

contemporánea. Lo mismo puede valer para las indicaciones de compás, especialmente para las partituras que incluyen una línea de tiempo graduada u otro gráfico. Pero estas partituras son excepcionales y casi todas las partituras proporcionales incluyen al menos unas pocas indicaciones de compás. Las claves y las alteraciones son aún más esenciales.

Así pues ¿qué estrategias existen para el espaciado de los elementos no musicales en un contexto proporcional? Una buena opción es la propiedad `strict-note-spacing` de `SpacingSpanner`. Compare las dos partituras siguientes:

```
\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
  c''8 c'' c'' \clef alto d' d'2
}

\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  c''8 c'' c'' \clef alto d' d'2
}
```



Las dos partituras son proporcionales, pero el espaciado de la primera es muy suelto a causa del cambio de clave. Sin embargo, el espaciado de la segunda partitura se mantiene estricto, porque `strict-note-spacing` está activado. La activación de `strict-note-spacing` hace que el ancho de las indicaciones de compás, armaduras, cambios de clave y alteraciones no tomen parte en el algoritmo de espaciado.

Además de los ajustes dados aquí, hay otros que aparecen con frecuencia en las partituras proporcionales. Entre ellos están:

- `\override SpacingSpanner.strict-grace-spacing = ##t`
- `\set tupletFullLength = ##t`
- `\override Beam.breakable = ##t`
- `\override Glissando.breakable = ##t`
- `\override TextSpanner.breakable = ##t`
- `\remove "Forbid_line_break_engraver" in the Voice context`

Estos ajustes dan a las notas de adorno un espaciado estricto, extienden los corchetes de grupo especial para que marquen tanto los puntos de comienzo como de final, y permiten que los elementos de extensión se dividan entre los sistemas y las páginas. Consulte las partes respectivas del manual para ver estos ajustes relacionados.

Véase también

Referencia de la notación: Sección 4.5.2 [Área de espaciado nueva], página 554.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.6 Encajar la música en menos páginas

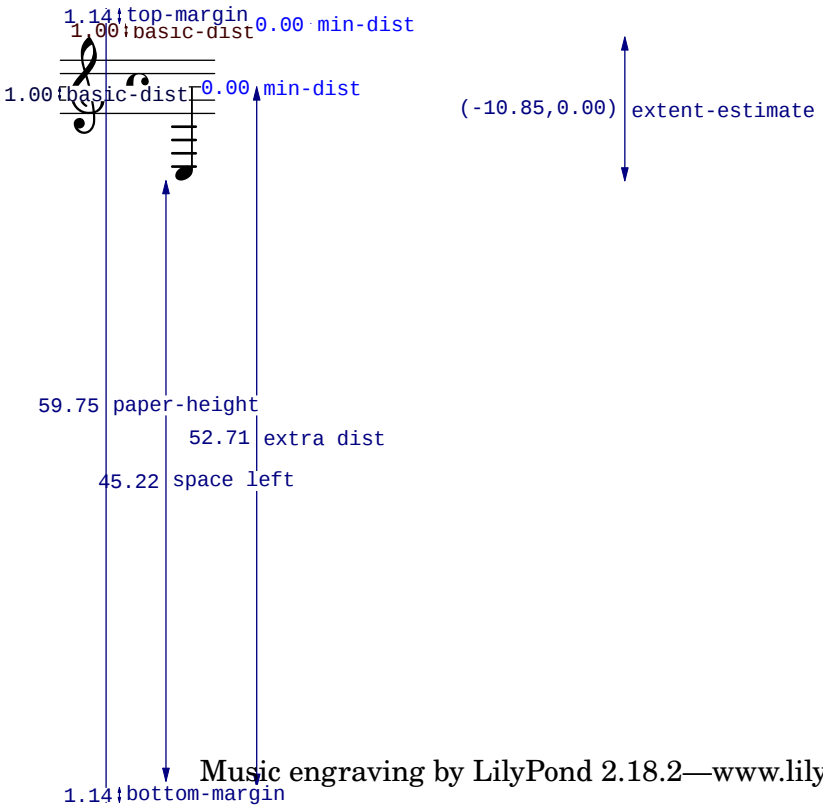
En ocasiones, podemos terminar con uno o dos pentagramas en una segunda página (o tercera, o cuarta...). Es fastidioso, especialmente cuando vemos que las páginas anteriores parecen tener espacio de sobra.

Al investigar los problemas de disposición, una herramienta de valor incalculable es `annotate-spacing`. Esta instrucción imprime los valores de un cierto número de variables de espaciado; para ver más detalles consulte la sección siguiente, Sección 4.6.1 [Mostrar el espaciado], página 564.

4.6.1 Mostrar el espaciado

Para presentar gráficamente las dimensiones de las variables de disposición vertical que pueden verse alteradas por el formato de la página, establezca `annotate-spacing` en el bloque `\paper`:

```
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}
```



Todas las dimensiones de disposición se muestran en espacios de pentagrama, independientemente de las unidades especificadas en los bloques `\paper` o `\layout`. En el ejemplo anterior, `paper-height` tiene un valor de 59.75 `staff-spaces` (espacios de pentagrama), y el `staff-size` (tamaño del pentagrama) es de 20 puntos, el valor predeterminado. Observe que:

1 punto = (25.4/72.27) mm

1 staff-space = (`staff-size`)/4 pts
= (`staff-size`)/4 * (25.4/72.27) mm

En este caso, un `staff-space` equivale aproximadamente a 1.757mm. Así, la medida `paper-height` de 59.75 `staff-spaces` equivale a 105 milímetros, la altura de una hoja `a6` en orientación apaisada. Las parejas (*a*,*b*) son intervalos, donde *a* es el límite inferior y *b* es el límite superior del intervalo.

Véase también

Referencia de la notación: Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.6.2 Cambiar el espaciado

La salida de `annotate-spacing` revela las dimensiones verticales con gran detalle. Para ver más detalles acerca de la modificación de los márgenes y otras variables de diseño de la página, consulte Sección 4.1 [Disposición de la página], página 513.

Aparte de los márgenes, hay algunas otras opciones para ahorrar espacio:

- Forzar los sistemas para que se junten lo más posible entre sí (de forma que quepan la mayor cantidad de ellos en la página) y al mismo tiempo estén espaciados de forma que no exista un espacio vacío en la parte baja del papel.

```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forzar el número de sistemas. Puede ser de utilidad de dos formas. Si tan sólo se establece un valor, incluso el mismo valor que el número de sistemas que se están trazando de forma predeterminada, ello ocasionará a veces que se traten de introducir más sistemas en cada página, pues en este caso se omite un paso en que se produce una estimación, dando así un encaje mejor en cada página. Asimismo, el forzar una reducción neta en el número de sistemas puede producir el ahorro de una página adicional. Por ejemplo, si la disposición predeterminada tiene 11 sistemas, la asignación siguiente fuerza una disposición con 10 sistemas.
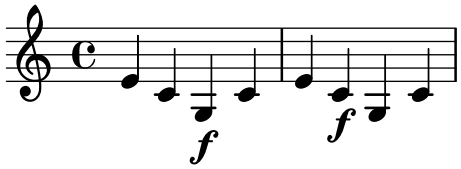
```
\paper {
  system-count = #10
}
```

- Forzar el número de páginas. Por ejemplo, la asignación siguiente fuerza una disposición con dos páginas.

```
\paper {
  page-count = #2
}
```
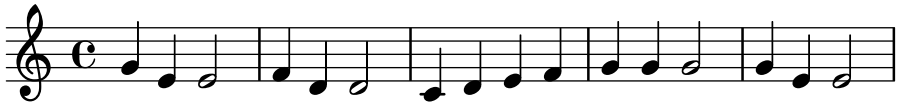
- Evitar (o reducir en número) los objetos que aumentan el tamaño vertical de un sistema. Por ejemplo, las repeticiones de primera y segunda vez (o repeticiones con finales alternativos) necesitan espacio adicional. Si estas repeticiones se reparten a lo largo de dos sistemas, ocupan más espacio que un sistema con las casillas de repetición y otro sistema sin ellas. Por ejemplo, las indicaciones dinámicas que se ‘salen’ de un sistema se pueden acercar al pentagrama:

```
e4 c g\f c
e4 c g-\tweak X-offset #-2.7 \f c
```
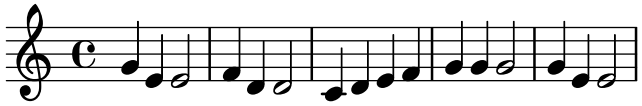


- Alterar el espaciado horizontal por medio de `SpacingSpanner`. Para ver más detalles, consulte Sección 4.5.3 [Cambiar el espaciado horizontal], página 555. El ejemplo siguiente ilustra el espaciado predeterminado:

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



El ejemplo siguiente modifica `common-shortest-duration` de un valor de `1/4` (negra) a `1/2` (blanca). La negra es la nota más común y la más breve del ejemplo, por lo que al agrandar esta duración se produce un efecto de 'apretujamiento':

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.common-shortest-duration =
        #(ly:make-moment 1/2)
    }
  }
}
```



La propiedad `common-shortest-duration` no se puede modificar de manera dinámica, por lo que se debe situar siempre dentro de un bloque `\context` de forma que se aplique a la partitura completa.

Véase también

Referencia de la notación: Sección 4.1 [Disposición de la página], página 513, Sección 4.5.3 [Cambiar el espaciado horizontal], página 555.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

5 Cambiar los valores por omisión

El objetivo del diseño de LilyPond es proporcionar la más alta calidad de los resultados, de forma predeterminada. A pesar de ello, podría tener que cambiar este resultado predeterminado. La disposición sobre el papel se controla a través de un amplio número de 'botones e interruptores' llamados en su conjunto 'propiedades'. En el Manual de aprendizaje podemos encontrar una introducción en forma de tutorial al acceso y modificación de estas propiedades, véase Sección "Trucar la salida" in *Manual de Aprendizaje*. Éste debería leerse en primer lugar. Este capítulo cubre un terreno similar, pero con un estilo más adecuado para un manual de referencia.

La descripción definitiva de los controles que están disponibles para su ajuste fino están en un documento aparte: la Sección "Referencia de funcionamiento interno" in *Referencia de Funcionamiento Interno*. Dicho manual relaciona todas las variables, funciones y opciones que se encuentran disponibles en LilyPond. Está escrito como un documento HTML, que se puede encontrar en línea, pero que también va incluido en el paquete de la documentación de LilyPond.

Internamente, LilyPond utiliza el lenguaje Scheme (un dialecto de LISP) para aportar la infraestructura. La sobreescritura de las decisiones de disposición da acceso efectivo a las interioridades del programa, lo que requiere código de Scheme como entrada. Los elementos de Scheme se inauguran dentro de un archivo '`.ly`' con el símbolo de cuadradillo `#`.[1]

5.1 Contextos de interpretación

Esta sección explica qué son los contextos y cómo modificarlos.

Véase también

Manual de aprendizaje: Sección "Contextos y grabadores" in *Manual de Aprendizaje*.

Archivos de inicio: '`ly/engraver-init.ly`', '`ly/performer-init.ly`'.

Fragmentos de código: Sección "Contexts and engravers" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Contexts" in *Referencia de Funcionamiento Interno*, Sección "Engravers and Performers" in *Referencia de Funcionamiento Interno*.

5.1.1 Explicación de los contextos

Los contextos se disponen de forma jerárquica:

Definiciones de salida - estructura de los contextos

Esta sección explica la relevancia de las definiciones de salida cuando se trabaja con los contextos. Más adelante ofrecemos ejemplos de definiciones de salida reales (véase [Cambiar todos los contextos del mismo tipo], página 578).

Aunque la música escrita en un archivo haga referencia a tipos y nombres de contexto, los contextos se crean solamente cuando la música se está realmente interpretando. LilyPond interpreta la música bajo el control de una 'definición de salida' y puede hacerlo así para varias definiciones de salida distintas, dando como resultado distintas salidas también. La definicion de salida que corresponde a la impresión de la música se especifica mediante `\layout`.

Una definición de salida mucho más sencilla que se usa para la producción de MIDI se especifica mediante `\midi`. Internamente, LilyPond utiliza varias otras definiciones de salida, como cuando se usa el combinador de particellas ([Combinación automática de las partes], página 175) o se crean fragmentos de notas guía citadas ([Citar otras voces], página 204).

Las definiciones de salida establecen la relación entre los contextos así como sus respectivos valores predeterminados. Aunque la mayor parte de los cambios se suelen hacer dentro de un

[1] Sección "Tutorial de Scheme" in *Extender* contiene un breve tutorial sobre la introducción de números, listas, cadenas y símbolos en Scheme.

bloque `\layout`, los ajustes de valores relacionados con el MIDI solamente tienen efecto cuando se hacen dentro de un bloque `\midi` block.

Algunos ajustes afectan a varias salidas: por ejemplo, si se desactiva el barrado automático, `autoBeaming`, dentro de algún contexto, las barras cuentan como melismas en lo que respecta a la correspondencia entre la música y la letra, como se describe en [Duración automática de las sílabas], página 254. Esta correspondencia se hace tanto para la salida impresa como para el MIDI. Si los cambios hechos sobre el `autoBeaming` dentro de la definición de contexto de un bloque `\layout` no se repiten dentro del correspondiente bloque `\midi`, la letra y la música dejarán de estar sincronizadas dentro del MIDI.

Véase también

Archivos instalados: '`ly/engraver-init.ly`'. '`ly/performer-init.ly`'.

Score. El contexto maestro

Este es el contexto de notación del nivel más alto. Ningún otro contexto puede contener a un contexto Score. De forma predeterminada, el contexto Score maneja la administración de las indicaciones de compás y se asegura de que ciertos elementos como claves, compases y armaduras están siempre alineados entre los distintos pentagramas.

Se crea implícitamente una instancia del contexto Score cuando se procesa un bloque `\score {...}`.

Contextos del nivel superior: contenedores de pentagramas

StaffGroup

Agrupa pentagramas y añade un corchete en la parte izquierda, formando un grupo. Las líneas divisorias de los pentagramas contenidos se conectan verticalmente. `StaffGroup` sólo consiste en una colección de pentagramas, con un corchete delante y líneas divisorias de arriba a abajo.

ChoirStaff

Idéntico a `StaffGroup` excepto que las barras de compás de los pentagramas contenidos no se conectan verticalmente.

GrandStaff

Un grupo de pentagramas, con una llave en la parte izquierda que abarca el grupo. Las barras de compás de los pentagramas contenidos se conectan verticalmente.

PianoStaff

Igual que `GrandStaff`, pero contempla la posibilidad de poner el nombre del instrumento a la izquierda del sistema.

Contextos de nivel intermedio. Pentagramas

Staff

Maneja claves, barras de compás, tonalidades y alteraciones accidentales. Puede contener contextos de `Voice`.

RhythmicStaff

Como `Staff`, pero para imprimir ritmos. Al imprimir se ignoran las alturas de las notas; las notas se imprimen sobre una línea. La salida de MIDI conserva las alturas sin modificación.

TabStaff

Contexto para generar tablaturas. De forma predeterminada dispone la expresión musical como una tablatura de guitarra, impresa sobre seis líneas.

DrumStaff

Maneja el tipografiado para instrumentos de percusión. Puede contener contextos `DrumVoice`.

VaticanaStaff

Igual que `Staff`, excepto que está pensado para tipografiar piezas en estilo gregoriano.

MensuralStaff

Igual que `Staff`, excepto que está diseñado para tipografiar piezas en estilo mensural.

Contextos del nivel más bajo. Voces

Los contextos del mismo nivel que Voice dan un valor inicial a ciertas propiedades e inician los grabadores correspondientes. Un contexto del nivel más bajo es aquel que no tiene un contexto descendiente predeterminado `defaultchild`. Aunque es posible hacer que pueda aceptar o contener subcontextos, éstos solo se pueden crear e introducir de forma explícita.

Voice

Corresponde a una voz sobre un pentagrama. este contexto maneja la conversión de las indicaciones dinámicas, plicas, barras, subíndices y superíndices, ligaduras de expresión y de unión, y silencios. Tenemos que crear instancias explícitas de este contexto si necesitamos varias voces en el mismo pentagrama.

VaticanaVoice

Lo mismo que `Voice`, excepto que está diseñado para tipografiar piezas en estilo gregoriano.

MensuralVoice

Lo mismo que `Voice`, con modificaciones para el tipografiado de piezas en estilo mensural.

Lyrics

Corresponde a una voz con letra. Maneja la impresión de una sola línea de letra.

DrumVoice

El contexto de voz utilizado en una pauta de percusión.

FiguredBass

El contexto en que los objetos `BassFigure` se crean a partir de la entrada escrita en el modo `\figuremode`.

TabVoice

El contexto de voz utilizado dentro de un contexto `TabStaff`. Se suele dejar que se cree implícitamente.

CueVoice

El contexto de voz que se utiliza para dibujar notas de tamaño reducido, con el principal objetivo de añadir notas guía de un pentagrama a otro, véase [Formateo de las notas guía], página 207. Normalmente se deja que se cree implícitamente.

ChordNames

Tipografía nombres de acordes.

5.1.2 Crear y referenciar contextos
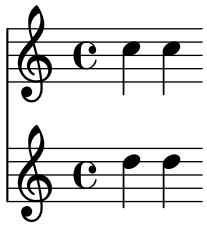
LilyPond crea automáticamente contextos de nifel inferior si se encuentra una expresión musical antes de que exista un contexto adecuado, pero normalmente esto sólo funciona bien para partituras sencillas o fragmentos musicales como los que aparecen en la documentación. Para partituras más complejas, se recomienda especificar explícitamente todos los contextos con las instrucciones `\new` o `\context`. La sintaxis de estas dos instrucciones es muy similar:

`[\new | \context]` *`Contexto`* `[ =` *`nombre`*`] [`*`expresión_musical`*`]`

donde se puede especificar `\new` o `\context`. *Contexto* es el tipo de contexto que se desea crear, *nombre* es un nombre opcional que se da al contexto concreto que se está creando, y *expresión_musical* es una sola expresión musical que será interpretada por los complementos grabadores y reproductores dentro de este contexto.
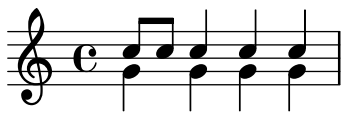
El prefijo `\new` sin ningún nombre se usa con frecuencia para crear partituras con muchos pentagramas:

```
<<
  \new Staff {
    % leave the Voice context to be created implicitly
    c4 c
  }
  \new Staff {
    d4 d
  }
>>
```



y para introducir varias voces dentro de un solo pentagrama:

```
<<
  \new Staff <<
    \new Voice {
      \voiceOne
      c8 c c4 c c
    }
    \new Voice {
      \voiceTwo
      g4 g g g
    }
  >>
>>
```
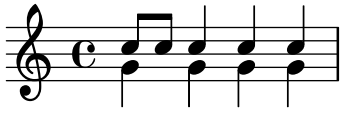


`\new` debería usarse siempre para especificar contextos sin nombre.

La diferencia entre `\new` y `\context` se encuentra en la acción que se realiza:

- `\new` con un nombre o sin él, siempre crea un contexto nuevo y distinto, incluso si ya existe un contexto con el mismo nombre:

```
<<
  \new Staff <<
    \new Voice = "A" {
      \voiceOne
      c8 c c4 c c
    }
    \new Voice = "A" {
      \voiceTwo
      g4 g g g
    }
```
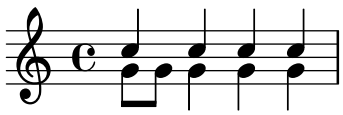
```
  >>
>>
```



- `\context` con un nombre especificado, crea un contexto nuevo solamente si no existe ya un contexto del mismo tipo y con el mismo nombre, dentro de la misma jerarquía de contextos. En caso contrario, se toma como referencia a dicho contexto creado previamente, y su expresión musical se pasa a este contexto para su interpretación.

  Una aplicación de los contextos con nombre es la separación entre la disposición de la partitura y el contenido musical. Son válidas culaquiera de las dos formas siguientes:

```
\score {
  <<
    % score layout
    \new Staff <<
      \new Voice = "one" {
        \voiceOne
      }
      \new Voice = "two" {
        \voiceTwo
      }
    >>
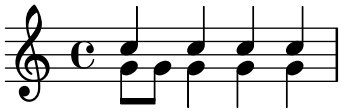
    % musical content
    \context Voice = "one" {
      \relative c'' {
        c4 c c c
      }
    }
    \context Voice = "two" {
      \relative c'' {
        g8 g g4 g g
      }
    }
  >>
}
```



```
\score {
  <<
    % score layout
    \new Staff <<
      \context Voice = "one" {
        \voiceOne
      }
      \context Voice = "two" {
        \voiceTwo
```

```
    }
  >>

  % musical content
  \context Voice = "one" {
    \relative c'' {
      c4 c c c
    }
  }
  \context Voice = "two" {
    \relative c'' {
      g8 g g4 g g
    }
  }
 >>
}
```



De manera alternativa, se pueden utilizar variables con un efecto similar. Véase Sección "Organizar las piezas mediante variables" in *Manual de Aprendizaje*.

- `\context` sin ningún nombre corresponderá con el primer contexto que se encuentre entre los creados previamente que sean del mismo tipo dentro de la misma jerarquía de contextos, incluso si tiene nombre, y su expresión musical se pasará a dicho contexto para su interpretación. Esta forma rara vez es útil. Sin embargo, `\context` sin nombre y sin expresión musical se usa para establecer el contexto en que se ejecuta un procedimiento de Scheme especificado con `\applyContext`:

```
\new Staff \relative c' {
  c1
  \context Timing
  \applyContext #(lambda (ctx)
                   (newline)
                   (display (ly:context-current-moment ctx)))
  c1
}
```

Un contexto debe tener un nombre si se va a hacer referencia a él más tarde, por ejemplo cuando se asocia la letra con la música:

```
\new Voice = "tenor" música
...
\new Lyrics \lyricsto "tenor" letra
```

Para ver más detalles sobre la asociación de letra y música, consulte [Duración automática de las sílabas], página 254.

Las propiedades de todos los contextos de un tipo en particular se pueden modificar dentro de un bloque `\layout` (con una sintaxis diferente), véase [Cambiar todos los contextos del mismo tipo], página 578.

Esta construcción también ofrece una forma de mantener las instrucciones de disposición separadas del contenido musical. Si se va a modificar un solo contexto, debe usarse un bloque `\with`, véase [Cambiar solamente un contexto determinado], página 580.

Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” in *Manual de Aprendizaje*.

Referencia de la notación: [Cambiar solamente un contexto determinado], página 580, [Duración automática de las sílabas], página 254.

5.1.3 Mantener vivos los contextos

Normalmente los contextos finalizan en el primer momento musical en que no tienen nada que hacer. Así, los contextos de `Voice` mueren tan pronto como ya no contienen ningún evento; los contextos de `Staff` mueren tan pronto como todos los contextos de `Voice` que contenían ya no contengan ningún evento; etc. Esto puede ocasionar dificultades si se tiene que hacer referencia a contextos anteriores que ya han muerto, por ejemplo, al cambiar de pentagramas con instrucciones `\change`, asociar letra con una voz mediante instrucciones `\lyricsto`, o cuando se añaden eventos musicales adicionales a un contexto anterior.

Existe una excepción a esta regla general: precisamente uno de los contextos de `Voice` que están dentro de un contexto de `Staff` o de una construcción `<<...>>` persiste siempre hasta el final de, contexto de `Staff` circundante o la construcción `<<...>>`, incluso aunque puede haber períodos en que no tiene nada que hacer. El contexto que persiste de esta forma será el primero que se encuentre en la primera construcción encerrada entre llaves `{...}`, ignorando cualquiera que se encuentre dentro de construcciones encerradas por ángulos dobles `<<...>>`.

Cualquier contexto se puede mantener vivo si nos aseguramos de que tiene algo que hacer en cualquier momento musical dado. Los contextos de `Staff` se mantienen con vida si nos aseguramos de que una de sus voces se mantiene viva. Una manera de hacerlo es añadir silencios de separación a una voz en paralelo con la música real. Éstos deben añadirse a todos y cada uno de los contextos de `Voice` que se hayan de mantener vivos. Si se van a usar esporádicamente varias voces, es más seguro mantenerlas todas vivas en lugar de tratar de confiar en las excepciones que hemos mencionado arriba.

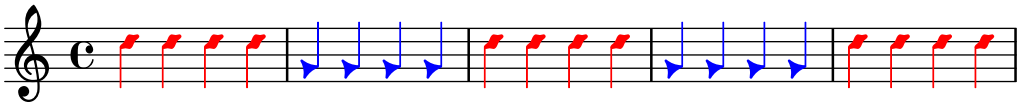
En el ejemplo siguiente, tanto la voz A como la voz B se mantienen vivas de esta manera durante la duración de la pieza:

```
musicA = \relative c'' { d4 d d d }
musicB = \relative c'' { g4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 }  % Keep Voice "A" alive for 5 bars
    \new Voice = "B" { s1*5 }  % Keep Voice "B" alive for 5 bars
  >>
}

music = {
  \context Voice = "A" {
    \voiceOneStyle
    \musicA
  }
  \context Voice = "B" {
    \voiceTwoStyle
    \musicB
  }
  \context Voice = "A" { \musicA }
  \context Voice = "B" { \musicB }
  \context Voice = "A" { \musicA }
```

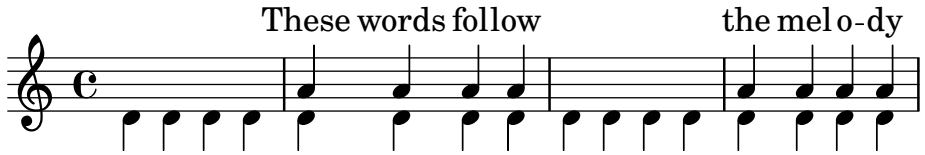
```
}

\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}
```



El ejemplo siguiente muestra cómo se puede escribir una línea melódica esporádica con letra utilizando este enfoque. Por supuesto, en una situación real la melodía y el acompañamiento consistirían en varias secciones diferentes.

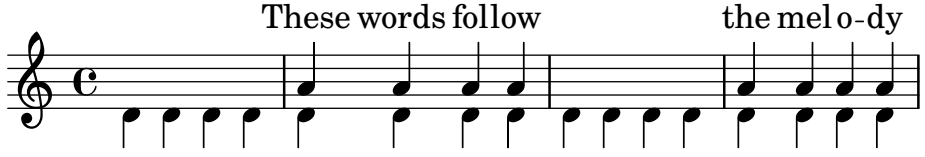
```
melody = \relative c'' { a4 a a a }
accompaniment = \relative c' { d4 d d d }
words = \lyricmode { These words fol -- low the mel -- o -- dy }
\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne
          s1*4  % Keep Voice "melody" alive for 4 bars
        }
        {
          \new Voice = "accompaniment" {
            \voiceTwo
            \accompaniment
          }
          <<
            \context Voice = "melody" { \melody }
            \context Voice = "accompaniment" { \accompaniment }
          >>
          \context Voice = "accompaniment" { \accompaniment }
          <<
            \context Voice = "melody" { \melody }
            \context Voice = "accompaniment" { \accompaniment }
          >>
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = #"music" }
    \lyricsto "melody" { \words }
  >>
}
```



Una forma alternativa, que podría resultar mejor en muchas situaciones, es mantener con vida la línea melódica simplemente incluyendo notas espaciadoras para alinearla correctamente con el acompañamiento:

```
melody = \relative c'' {
  s1  % skip a bar
  a4 a a a
  s1  % skip a bar
  a4 a a a
}
accompaniment = \relative c' {
  d4 d d d
  d4 d d d
  d4 d d d
  d4 d d d
}
words = \lyricmode { These words fol -- low the mel -- o -- dy }

\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne
          \melody
        }
        \new Voice = "accompaniment" {
          \voiceTwo
          \accompaniment
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = #"music" }
    \lyricsto "melody" { \words }
  >>
}
```



5.1.4 Modificar los complementos (plug-ins) de contexto

Los contextos de notación (como `Score` y `Staff`) no sólo almacenan propiedades, también contienen «plug-ins» o complementos llamados 'grabadores' que crean elementos de notación. Por ejemplo, el contexto `Voice` contiene un grabador `Note_heads_engraver` que crea las cabezas de nota y el contexto `Staff` contiene un grabador `Key_engraver` que crea la armadura.

Para ver una descripción completa de todos y cada uno de los complementos, consulte Referencia de funcionamiento interno ↦ Traducción ↦ Grabadores. Cada contexto que se describe
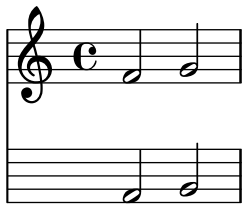
en Referencia de funcionamiento interno ↦ Traducción ↦ Contexto. relaciona los grabadores que se usan para ese contexto.

Puede ser de utilidad jugar un poco con estos complementos. Se hace iniciando un contexto nuevo con `\new` o `\context` y modificándolo:

```
\new contexto \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ...música...
}
```

donde los . . . deben ser el nombre de un grabador. Aquí tenemos un ejemplo sencillo que suprime los grabadores `Time_signature_engraver` y `Clef_engraver` de un contexto `Staff`:

```
<<
  \new Staff {
    f2 g
  }
  \new Staff \with {
     \remove "Time_signature_engraver"
     \remove "Clef_engraver"
  } {
    f2 g2
  }
>>
```

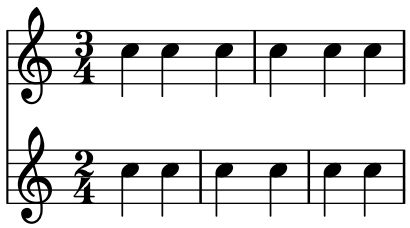


En el segundo pentagrama no hay indicación de compás ni clave. Éste es un método bastante rudimentario de hacer que desaparezcan los objetos porque afecta a todo el pentagrama. Este método también afecta al espaciado, lo que puede ser deseable o no serlo. Se muestran métodos más sofisticados para quitar objetos en Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*.

El ejemplo siguiente muestra una aplicación práctica. Normalmente las líneas divisorias y las indicaciones de compás están sincronizadas a lo largo de toda la partitura. Lo hacen los grabadores `Timing_translator` y `Default_bar_line_engraver`. Estos complementos mantienen al día la administración de las indicaciones de compás, posición dentro del compás, etc. Moviendo estos grabadores desde el contexto de `Score` al de `Staff`, podemos conseguir una partitura en la que cada pentagrama tiene su propio compás independiente.

```
\score {
  <<
    \new Staff \with {
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
```

```
    }
    \relative c'' {
        \time 3/4
        c4 c c c c c
    }
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
  \relative c'' {
      \time 2/4
      c4 c c c c c
  }
>>
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
    }
  }
}
```



Advertencias y problemas conocidos

El orden en que los grabadores se especifican es el orden en que se llaman para realizar su tarea de procesamiento. Normalmente, el orden en que se especifican los grabadores no tiene importancia, pero en algunos casos especiales sí la tiene, por ejemplo donde un grabador escribe una propiedad y otro la lee, o donde un grabador crea un groby otro debe procesarlo.

Las siguientes ordenaciones son importantes:

- el grabador de compases `Bar_engraver` debe ir normalmente en primer lugar,
- el grabador de digitaciones `New_fingering_engraver` debe ir antes del grabador `Script_column_engraver` de columnas de inscripciones,
- el `Timing_translator` debe ir antes del grabador de números de compás `Bar_number_engraver`.

Véase también

Archivos instalados: '`ly/engraver-init.ly`'.

5.1.5 Cambiar los valores por omisión de los contextos

Se pueden cambiar las propiedades de contexto y de grob con las instrucciones `\set` y `\override`, tal y como se describe en Sección 5.3 [Modificar las propiedades], página 590. Estas instrucciones crean eventos musicales, haciendo que los cambios tengan efecto en el punto temporal en que la música se está procesando.

Por contra, esta sección explica la forma de cambiarlos valores *predeterminados* de las propiedades de contexto y de grob en el momento en que se crea el contexto. Existen dos formas de hacerlo. Una modifica los valores predeterminados en todos los contextos de un tipo dado, y el otro modifica los valores predeterminados solamente en una instancia concreta de un contexto.

Cambiar todos los contextos del mismo tipo

Los ajustes de contexto predeterminados que se han de usar para la composición tipográfica dentro de `Score`, `Staff`, `Voice` y otros contextos se pueden especificar en un bloque `\context` dentro de cualquier bloque `\layout`.

Los ajustes para la salida MIDI, al contrario que para la composición tipográfica, se tendrán que especificar aparte en bloques `\midi` (véase [Definiciones de salida - estructura de los contextos], página 567).

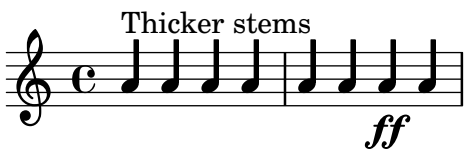
El bloque `\layout` se debe colocar dentro del bloque `\score` al que se aplica, después de la música.

```
\layout {
  \context {
    \Voice
    [ajustes de contexto para todos los contextos Voice]
  }
  \context {
    \Staff
    [ajustes de contexto para todos los contextos Staff]
  }
}
```

Se pueden especificar los siguientes tipos de ajustes:

- Una instrucción `\override`, pero omitiendo el nombre del contexto

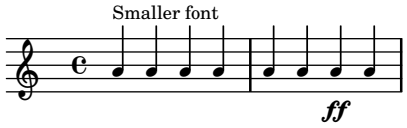
```
\score {
  \relative c'' {
    a4^"Thicker stems" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      \override Stem.thickness = #4.0
    }
  }
}
```



- Estableciendo una propiedad de contexto directamente

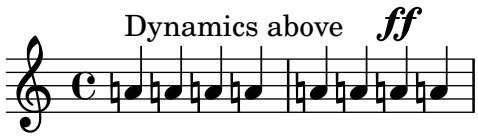
```
\score {
  \relative c'' {
    a4^"Smaller font" a a a
    a4 a a\ff a
```

```
  }
  \layout {
    \context {
      \Staff
      fontSize = #-4
    }
  }
}
```



- Una instrucción predefinida tal como `\dynamicUp` o una expresión musical como `\accidentalStyle dodecaphonic`

```
\score {
  \relative c'' {
    a4^"Dynamics above" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Voice
      \dynamicUp
    }
    \context {
      \Staff
      \accidentalStyle dodecaphonic
    }
  }
}
```

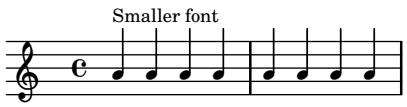


- Una variable definida por el usuario que contenga un bloque `\with`; para ver detalles acerca del bloque `\with`, consulte [Cambiar solamente un contexto determinado], página 580.

```
StaffDefaults = \with {
  fontSize = #-4
}
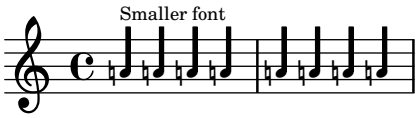
\score {
  \new Staff {
    \relative c'' {
      a4^"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \context {
```

```
        \Staff
        \StaffDefaults
      }
    }
  }
```



Las instrucciones de ajuste de propiedades se pueden disponer dentro de un bloque `\layout` sin que estén encerradas en un bloque `\context`. Tales ajustes son equivalentes a incluir las mismas instrucciones de ajuste de propiedades al comienzo de cada uno de los contextos del tipo especificado. Si no se especifica ningún contexto, *todos y cada uno* de los contextos del nivel inferior quedan afectados, véase [Contextos del nivel más bajo. Voces], página 569. La sintaxis de una instrucción de ajuste de propiedades dentro de un bloque `\layout` es la misma que si la misma instrucción estuviera escrita en el propio flujo musical.

```
\score {
  \new Staff {
    \relative c'' {
      a4^"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \accidentalStyle dodecaphonic
    \set fontSize = #-4
    \override Voice.Stem.thickness = #4.0
  }
}
```



Cambiar solamente un contexto determinado

Las propiedades de contexto de una única instancia de contexto pueden cambiarse dentro de un bloque `\with`. Todas las demás instancias de contexto del mismo tipo retienen los ajustes predeterminados que LilyPond tiene programados y que se modifican por parte de cualquier bloque `\layout` que se encuentre dentro del ámbito. El bloque `\with` se debe situar inmediatamente después de las instrucciones `\new` *context-type*:

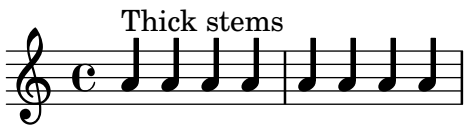
```
\new Staff \with { [ajustes de contexto para esta instancia de contexto solamente] }
{
  ...
}
```

Dado que dicha 'modificación de contexto' está especificada dentro de la música, afectará a *todas* las salidas (tipografía *y también* el MIDI), a diferencia de los cambios que se hacen dentro de una definición de salida.

Se pueden especificar los siguientes tipos de ajustes:

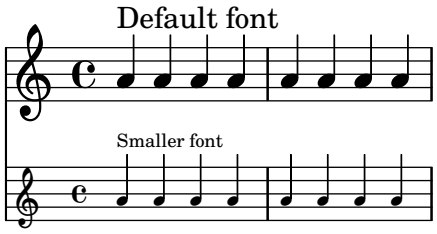
- Una instrucción `\override`, pero omitiendo el nombre del contexto

```
\score {
  \new Staff {
    \new Voice \with { \override Stem.thickness = #4.0 }
    {
      \relative c'' {
        a4^"Thick stems" a a a
        a4 a a a
      }
    }
  }
}
```



- Estableciendo una propiedad de contexto directamente

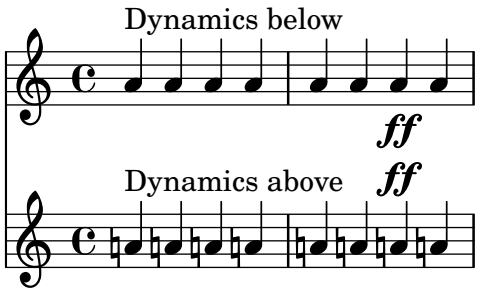
```
\score {
  <<
    \new Staff {
      \relative c'' {
        a4^"Default font" a a a
        a4 a a a
      }
    }
    \new Staff \with { fontSize = #-4 }
    {
      \relative c'' {
        a4^"Smaller font" a a a
        a4 a a a
      }
    }
  >>
}
```



- Una instrucción predefinida tal como `\dynamicUp`

```
\score {
  <<
    \new Staff {
      \new Voice {
        \relative c'' {
          a4^"Dynamics below" a a a
          a4 a a\ff a
        }
```

```
      }
    }
    \new Staff \with { \accidentalStyle dodecaphonic }
    {
      \new Voice \with { \dynamicUp }
      {
        \relative c'' {
          a4^"Dynamics above" a a a
          a4 a a\ff a
        }
      }
    }
  >>
}
```



Orden de precedencia

El valor de una propiedad que se aplica en un momento determinado se determina de la siguiente forma:

- si está en efecto una instrucción `\override` o `\set` dentro del flujo musical, se usa dicho valor,
- de lo contrario, se usa el valor predeterminado tomado de un enunciado `\with` en las instrucciones de inicio del contexto,
- de lo contrario, se usa el valor tomado del bloque `\context` más reciente que corresponda dentro de los bloques `\layout` o `\midi`,
- de lo contrario se usa el valor predeterminado que LilyPond lleva preprogramado.

Véase también

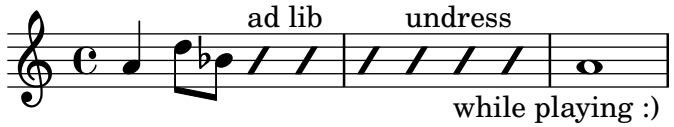
Manual de aprendizaje: Sección “Modificar las propiedades de los contextos” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.1 [Explicación de los contextos], página 567, [Contextos del nivel más bajo. Voces], página 569, Sección 5.3.2 [La instrucción set], página 590, Sección 5.3.3 [La instrucción override], página 593, Sección 4.2.1 [El bloque `\layout`], página 524.

5.1.6 Definir contextos nuevos

Los contextos específicos, como `Staff` y `Voice`, están construidos a partir de bloques sencillos. Es posible crear nuevos tipos de contextos con combinaciones distintas de añadidos grabadores.

El siguiente ejemplo muestra cómo construir un tipo diferente de contexto de `Voice` partiendo de cero. Será parecido a `Voice`, pero imprime solamente cabezas centradas en forma de barra inclinada. Se puede usar para indicar improvisación en piezas de jazz,



Estos ajustes se definen dentro de un bloque `\context` que a su vez está dentro de un bloque `\layout`,

```
\layout {
  \context {
    ...
  }
}
```

En el siguiente análisis, la entrada de ejemplo que se muestra debe ir en el lugar de los puntos suspensivos . . . del fragmento anterior.

En primer lugar es necesario definir un nombre para el nuevo contexto:

```
\name ImproVoice
```

Debido a que es parecido al contexto `Voice`, queremos órdenes que funcionen dentro de los contextos `Voice` (existentes) para que siga funcionando. Esto se consigue dando al contexto nuevo un alias de `Voice`,

```
\alias Voice
```

El contexto imprimirá notas y textos explicativos, por ello tenemos que añadir los grabadores que aportan esta funcionalidad, y además el grabador que agrupa las notas, plicas y silencios que están en el mismo momento musical en forma de columnas,

```
\consists "Note_heads_engraver"
\consists "Text_engraver"
\consists "Rhythmic_column_engraver"
```

Las cabezas de todas las notas se deben situar sobre la línea central,

```
\consists "Pitch_squash_engraver"
squashedPosition = #0
```

El grabador `Pitch_squash_engraver` modifica las cabezas de nota (creadas por el grabador `Note_heads_engraver`) y establece sus posiciones verticales al valor de `squashedPosition`, en este caso `0`, la línea central.

Las notas parecen barras inclinadas y no tienen plica:

```
\override NoteHead.style = #'slash
\hide Stem
```

Todos estos complementos o plug-ins tienen que comunicarse bajo el control del contexto. Los mecanismos con el que se comunican los contextos se establecen mediante la declaración del `\type` (tipo) del contexto. Dentro de un bloque `\layout`, casi todos los contextos serán del tipo `Engraver_group`. Algunos contextos especiales y los contextos de los bloques `\midi` usan otros tipos. La copia y la modificación de una definición de contexto existente también cumplimentan el tipo. Como este ejemplo crea una definición partiendo de cero, tiene que ser especificada explícitamente.

```
\type "Engraver_group"
```

Al juntarlo todo, obtenemos

```
\context {
  \name ImproVoice
  \type "Engraver_group"
  \consists "Note_heads_engraver"
  \consists "Text_engraver"
  \consists "Rhythmic_column_engraver"
  \consists "Pitch_squash_engraver"
```

```
  squashedPosition = #0
  \override NoteHead.style = #'slash
  \hide Stem
  \alias Voice
}
```

Los contextos dan lugar a jerarquías. Queremos poner el contexto `ImproVoice` dentro del contexto `Staff`, igual que los contextos de voz normales. Por tanto, modificamos la definición de `Staff` con la instrucción `\accepts` (acepta),

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Lo opuesto a `\accepts` (acepta) es `\denies` (deniega), lo que a veces se necesita cuando se están reutilizando definiciones de contexto existentes.

Ponemos ambos dentro de un bloque `\layout`, como

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \context {
    \Staff
    \accepts "ImproVoice"
  }
}
```

Así pues, la salida que aparece al comienzo de esta sub-sección se puede escribir como

```
\relative c'' {
  a4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"desvístete"
    c c_"mientras tocas :)"
  }
  a1
}
```

Para completar el ejemplo, los cambios que afectan a la jerarquía de contextos se deben repetir dentro de un bloque `\midi` de manera que la salida MIDI dependa de las mismas relaciones de contexto.

Véase también

Referencia de funcionamiento interno: Sección “Engraver_group” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Text_engraver” in *Referencia de Funcionamiento Interno*, Sección “Rhythmic_column_engraver” in *Referencia de Funcionamiento Interno*, Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*.

5.1.7 Orden de disposición de los contextos

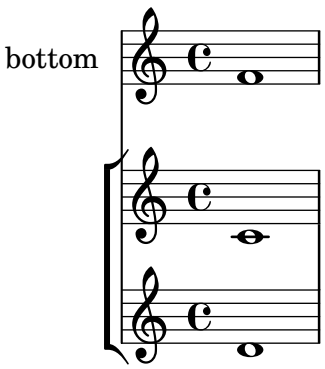
Los contextos se disponen en un sistema normalmente desde arriba hacia abajo en el orden en que se encuentran en el archivo de entrada. Cuando los contextos se anidan unos dentro de

otros, el contexto exterior incluye a los contextos anidados tal y como se especifica en el archivo de entrada, siempre y cuando los contextos interiores estén incluidos en la lista "accepts" del contexto externo. Los contextos anidados que no está nincluidos en la lista "accepts" del contexto externo se recolocan debajo del contexto externo en lugar de anidarse dentro de él.

La lista "accepts" de un contexto se puede cambiar con las instrucciones `\accepts` (acepta) o `\denies` (niega). `\accepts` añade un contexto a la lista "accepts" y `\denies` elimina un contexto de la lista.

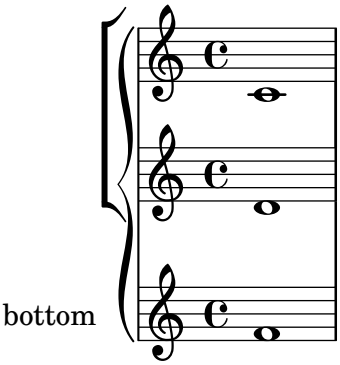
Por ejemplo, un grupo de pentagramas con un corchete cuadrado no se encuentra normalmente en el interior de un pentagrama con llave curva que tenga conectadas las líneas divisorias, y un `GrandStaff` para piano no acepta un `StaffGroup` dentro de él, de forma predeterminada.

```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = bottom f'1 }
  >>
}
```



Sin embargo, usando la instrucción `\accepts`, se puede añadir un `StaffGroup` al contexto `GrandStaff`:

```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = bottom f'1 }
  >>
  \layout {
    \context {
      \GrandStaff
      \accepts "StaffGroup"
    }
  }
}
```



`\denies` se usa principalmente cuando un contexto nuevo se está basando en otro, pero los anidamientos requeridos difieren. Por ejemplo, el contexto `VaticanaStaff` está basado en el contexto `Staff`, pero con el contexto `VaticanaVoice` sustituido por el contexto `Voice` en la lista "accepts".

Observe que discretamente se crea un contexto de forma implícita si se encuentra una instrucción donde no hay un contexto apropiado para contenerlo.

Dentro de una definición de contexto, el tipo de un subcontexto que se va a crear implícitamente se especifica usando `\defaultchild` (hijo predeterminado). Algunos eventos musicales requieren un contexto '`Bottom`' (inferior): cuando se encuentra este evento, se crean subcontextos de forma recursiva hasta que se alcanza un contexto que no tiene establecido el '`defaultchild`'.

La creación implícita de contextos puede dar lugar a pentagramas o partituras nuevos no esperados. La utilización de `\new` para crear contextos explícitamente evita esos problemas.

En ocasiones se necesita que un contexto exista durante un breve intervalo de tiempo, siendo un buen ejemplo el contexto de pentagrama de un ossia. Esto se consigue normalmente mediante la introducción de la definición del contexto en el lugar apropiado en paralelo con la sección correspondiente de la música principal. De forma predeterminada, el contexto tempral se coloca debajo de todos los contextos existentes. Para reposicionarlo por encima del contexto que tenga el nombre "principal", debería definirse de esta forma:

```
\new Staff \with { alignAboveContext = #"principal" }
```

Se presenta una situación similar cuando se posiciona un contexto temporal de letra de una canción dentro de una disposición de varios pentagramas tal como `ChoirStaff`, por ejemplo, cuando se añade una segunda estrofa a una sección que se repite. De forma predeterminada, el contexto temporal de letra se coloca debajo de los pentagramas inferiores. Mediante la definición del contexto temporal de letra con `alignBelowContext` se puede posicionar correctamente debajo del contexto de letra con nombre que contiene el texto de la primera estrofa.

En diversos lugares pueden verse ejemplos que muestran esta recolocación de contextos temporales: véase Sección "Anidado de expresiones musicales" in *Manual de Aprendizaje*, Sección 1.6.2 [Modificación de pentagramas sueltos], página 191 y Sección 2.1.2 [Técnicas específicas para la letra], página 263.

Véase también

Manual de aprendizaje: Sección "Anidado de expresiones musicales" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.6.2 [Modificación de pentagramas sueltos], página 191, Sección 2.1.2 [Técnicas específicas para la letra], página 263.

Manual de utilización del programa: Sección "Aparece un pentagrama de más" in *Utilización del Programa*.
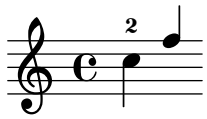
Archivos instalados: '`ly/engraver-init.ly`'.

5.2 Explicación del Manual de referencia de funcionamiento interno

5.2.1 Navegar por la referencia del programa

Supongamos que queremos mover la indicación de digitación del fragmento siguiente:

```
c-2
\stemUp
f
```



Si hace una visita a la documentación en busca de instrucciones de digitación (en [Indicaciones de digitación], página 214), encontrará:

Véase también

Referencia de funcionamiento interno: Sección “Fingering” in *Referencia de Funcionamiento Interno*.

La referencia del programador se encuentra disponible en forma de documento HTML. Se recomienda mucho que lo lea en la forma HTML, bien en línea o bien descargando los archivos de la documentación HTML. Esta sección sería mucho más difícil de entender si está utilizando el manual en formato PDF.

Siga el enlace que lleva a Sección “Fingering” in *Referencia de Funcionamiento Interno*. Al principio de la página, puede ver

Los objetos de digitación se crean por parte de: Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno* y Sección “New_fingering_engraver” in *Referencia de Funcionamiento Interno*.

Siguiendo los enlaces relacionados dentro de la referencia del programa, podemos seguir el flujo de información dentro del programa:

- Sección “Fingering” in *Referencia de Funcionamiento Interno*: los objetos Sección “Fingering” in *Referencia de Funcionamiento Interno* se crean por parte de: Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*
- Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*: Tipos de música aceptados: Sección “fingering-event” in *Referencia de Funcionamiento Interno*
- Sección “fingering-event” in *Referencia de Funcionamiento Interno*: El tipo de evento musical `fingering-event` está descrito en Expresiones musicales con el nombre de Sección “FingeringEvent” in *Referencia de Funcionamiento Interno*

Este camino se recorre en contra de la corriente de información del programa: comienza por la salida y acaba en el evento de entrada. También podríamos haber empezado por un evento de la entrada, y leído siguiendo el flujo de información terminando en su caso en el objeto (u objetos) de la salida.

La referencia del programa también se puede examinar como un documento normal. Contiene capítulos que tratan de `Music definitions` de la Sección “Translation” in *Referencia de Funcionamiento Interno*, y del Sección “Backend” in *Referencia de Funcionamiento Interno*. Cada uno de los capítulos relaciona todas las definiciones utilizadas y todas las propiedades que se pueden ajustar.

5.2.2 Interfaces de la presentación

La página HTML que pudimos ver en la sección anterior describe el objeto de presentación llamado Sección “Fingering” in *Referencia de Funcionamiento Interno*. Dicho objeto es un símbolo dentro de la partitura. Tiene propiedades que guardan números (como grosores y direcciones), pero también punteros a objetos relacionados. Un objeto de presentación también se llama un *Grob*, que es una abreviatura de Graphical Object (objeto gráfico). Para ver más detalles acerca de los objetos gráficos o Grobs, consulte Sección “grob-interface” in *Referencia de Funcionamiento Interno*.

La página dedicada a `Fingering` relaciona las definiciones del objeto `Fingering`. Por ejemplo, la página dice

`padding` (dimensión, en espacios de pentagrama):

`0.5`

lo que significa que el número se mantendrá a una distancia de al menos 0.5 de la cabeza de la nota.

Cada objeto de presentación puede tener varias funciones como elemento notacional o tipográfico. Por ejemplo, el objeto de digitación Fingering tiene los siguientes aspectos

- Su tamaño es independiente del espaciado horizontal, al contrario de las ligaduras o las barras de las figuras.
- Es un elemento de texto. Casi seguro que es un texto muy corto.
- este elemento de texto se tipografía con un tipo de letra, no como las ligaduras o las barras de las figuras.
- Horizontalmente, el centro del símbolo se debe alinear con el centro de la cabeza de la nota.
- Verticalmente, el símbolo se coloca cerca de la nota y del pentagrama.
- La posición vertical también está coordinada con otros símbolos de superíndice y de subíndice.

Cada uno de estos aspectos se capta en lo que se llaman *interfaces*, que se relacionan al final de la página dedicada a Sección “Fingering” in *Referencia de Funcionamiento Interno*

Este objeto contempla los siguientes interfaces: Sección “item-interface” in *Referencia de Funcionamiento Interno*, Sección “self-alignment-interface” in *Referencia de Funcionamiento Interno*, Sección “side-position-interface” in *Referencia de Funcionamiento Interno*, Sección “text-interface” in *Referencia de Funcionamiento Interno*, Sección “text-script-interface” in *Referencia de Funcionamiento Interno*, Sección “font-interface” in *Referencia de Funcionamiento Interno*, Sección “finger-interface” in *Referencia de Funcionamiento Interno* y Sección “grob-interface” in *Referencia de Funcionamiento Interno*.

Al pulsar sobre cualquiera de los enlaces nos desplazaremos a la página del respectivo interfaz del objeto. Cada interfaz tiene un cierto número de propiedades. Algunas de ellas no son para que el usuario las pueda ajustar (‘Propiedades internas’), pero otras sí se pueden modificar.

Hemos estado hablando de *el* objeto `Fingering`, pero realmente esto no significa mucho. El archivo de inicialización (véase Sección “Otras fuentes de información” in *Manual de Aprendizaje*) ‘`scm/define-grobs.scm`’ muestra el alma del ‘objeto’,

```
(Fingering
  . ((padding . 0.5)
     (avoid-slur . around)
     (slur-padding . 0.2)
     (staff-padding . 0.5)
     (self-alignment-X . 0)
     (self-alignment-Y . 0)
     (script-priority . 100)
```
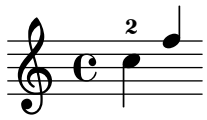
```
(stencil . ,ly:text-interface::print)
(direction . ,ly:script-interface::calc-direction)
(font-encoding . fetaText)
(font-size . -5)  ; don't overlap when next to heads.
(meta . ((class . Item)
(interfaces . (finger-interface
               font-interface
               text-script-interface
               text-interface
               side-position-interface
               self-alignment-interface
               item-interface))))))
```

Como podemos ver, el objeto `Fingering` no es más que un montón de valores de variables, y la página web de la Referencia de funcionamiento interno se genera directamente a partir de esta definición.

5.2.3 Determinar la propiedad del grob

Recordemos que queríamos cambiar la posición del **2** en

```
c-2
\stemUp
f
```



Puesto que el **2** se encuentra colocado verticalmente sobre su nota, tenemos que negociar con el interfaz asociado con esta colocación. Esto se hace usando `side-position-interface`. La página que describe este interface dice:

`side-position-interface`

Colocar un objeto víctima (este mismo) junto a otros objetos (el soporte). La propiedad `direction` significa dónde poner el objeto víctima con relación al soporte (¿a la izquierda o a la derecha, encima o debajo?)

Debajo de esta descripción, la variable `padding` (relleno) se describe como
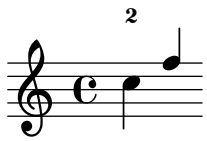
`padding` (dimensión, en espacios de pentagrama)
Añadir esta cantidad de espacio adicional entre objetos que están unos junto a otros.

Aumentando el valor de `padding`, podemos alejar la cifra de digitación de la cabeza de la nota. La siguiente orden inserta un espacio en blanco de 3 espacios de pentagrama entre la nota y la digitación:

```
\once \override Voice.Fingering.padding = #3
```

Al insertar esta instrucción antes de que se haya creado el objeto Fingering, es decir, antes del `c2`, llegamos al siguiente resultado:

```
\once \override Voice.Fingering.padding = #3
c-2
\stemUp
f
```



En este caso, el contexto de este truco es `Voice`. Este hecho se puede deducir también a partir de la referencia del programa, ya que la página dedicada al añadido Sección "Fingering_engraver" in *Referencia de Funcionamiento Interno* dice

El grabador Fingering_engraver es parte de los contextos: ... Sección "Voice" in *Referencia de Funcionamiento Interno*

5.2.4 Convenciones de nombres

Se hace necesario presentar una panorámica de las diversas convenciones de nomenclatura:

- funciones de Scheme: minúsculas-con-guiones (incluso nombres de una sola palabra)
- funciones de Scheme: ly:más-estilo-de-scheme
- eventos, clases y propiedades musicales: como-las-funciones-de-scheme
- interfaces de Grobs: estilo-scheme
- propiedades de backend: estilo-scheme (¡pero X e Y en mayúsculas!)
- contextos (y ExpresionesMusicales y grobs): Mayúsculas o MayúsculasDeCamello
- propiedades de contexto: minúsculasSeguidoDeMayúsculasDeCamello
- grabadores: Mayúsculas_seguido_de_minúsculas_y_con_barras_bajas

5.3 Modificar las propiedades

5.3.1 Panorámica de la modificación de las propiedades

Cada contexto es responsable de la creación de ciertos tipos de objetos gráficos. Los ajustes que se usan para imprimir estos objetos también se almacenan por contexto. Mediante la modificación de estos ajustes, se puede alterar la apariencia de los objetos.

Existen dos tipos diferentes de propiedades almacenadas en los contextos: las propiedades de contexto y las propiedades de grob. Las propiedades de contexto son propiedades que se aplican al contexto como un todo y controlan la forma en que el propio contexto se imprime. Por contra, las propiedades de grob se aplican a los tipos de grob específicos que se imprimirán dentro del contexto.

Las instrucciones `\set` y `\unset` se usan para cambiar los valores de las propiedades de contexto. Las instrucciones `\override` y `\revert` se usan para cambiar los valores de las propiedades de grob.

Véase también

Referencia de funcionamiento interno: Sección "Backend" in *Referencia de Funcionamiento Interno*, Sección "All layout objects" in *Referencia de Funcionamiento Interno*, Sección "OverrideProperty" in *Referencia de Funcionamiento Interno*, Sección "RevertProperty" in *Referencia de Funcionamiento Interno*, Sección "PropertySet" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El «back-end» o motor de salida no es muy estricto en la comprobación de tipos de las propiedades de objetos. Las referencias cíclicas en valores Scheme de propiedades pueden producir cuelgues o salidas abruptas, o las dos cosas.

5.3.2 La instrucción `\set`

Cada contexto puede tener distintas *propiedades*, variables contenidas dentro de ese contexto. Se pueden cambiar mientras dura el paso de interpretación. Se consigue insertando la instrucción `\set` dentro de la música:

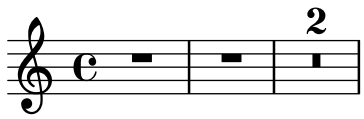
`\set` *`contexto.propiedad`* `= #`*`valor`*

valor es un objeto de Scheme, razón por la que va precedido del carácter almohadilla, `#`.

El nombre de las propiedades de contexto suele ir en minúsculas con mayúscula en medio. Controlan sobre todo la traducción de la música a la notación, p.ej. `localKeySignature` (para determinar si hay que impriir alteraciones o no), o `measurePosition` (para determinar cuándo hay que imprimir una línea divisoria). El valor de las propiedades de contexto puede modifcarse con el tiempo durante la interpretación de la música; un ejemplo obvio es `measurePosition`. Las propiedades de contexto se modifican mediante la instrucción `\set`.
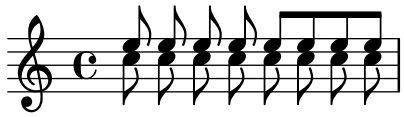
Por ejemplo, los silencios multicompás se combinan en un solo compás si el valor de la propiedad de contexto `skipBars` se establece a `#t` (verdadero):

```
R1*2
\set Score.skipBars = ##t
R1*2
```



Si se omite el argumento *context*, entonces se utiliza el contexto actual de nivel más bajo (normalmente `ChordNames`, `Voice` o `Lyrics`). En este ejemplo:
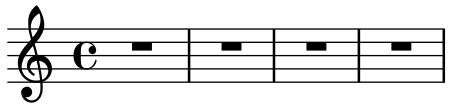
```
\set Score.autoBeaming = ##f
<<
  {
    e8 e e e
    \set autoBeaming = ##t
    e8 e e e
  } \\ {
    c8 c c c c8 c c c
  }
>>
```



El cambio se aplica ‘al vuelo’, mientras dura la música, de forma que el ajuste sólo afecta al segundo grupo de corcheas.

Observe que el contexto del nivel más bajo no siempre contiene la propiedad que querríamos modificar: por ejemplo, intentar ajustar el valor de la propiedad `skipBars` del contexto predeterminado del nivel más bajo, que en este caso es `Voice`, no tendrá ningún efecto, porque skipBars es una propiedad del contexto `Score`.

```
R1*2
\set skipBars = ##t
R1*2
```



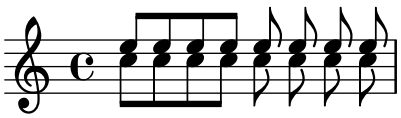
Los contextos son jerárquicos, y si se ha especificado un contexto mayor, por ejemplo `Staff`, entonces el cambio se aplicaría también a todos los contextos `Voice` dentro del pentagrama actual.

También existe una instrucción `\unset`:

`\unset` *`contexto.propiedad`*

que elimina la definición de *propiedad*. Esta instrucción elimina la definición solamente si está establecida dentro de *contexto*. Properties that have been set in enclosing contexts will not be altered by an unset in an enclosed context:

```
\set Score.autoBeaming = ##t
<<
  {
    \unset autoBeaming
    e8 e e e
    \unset Score.autoBeaming
    e8 e e e
  } \\ {
    c8 c c c c8 c c c
  }
>>
```
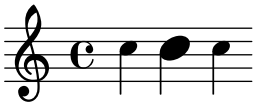


Como `\set`, el argumento *contexto* no se tiene que especificar para un contexto del nivel más bajo, por lo que los dos enunciados

```
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t
```

son equivalentes si el contexto inferior en curso es `Voice`.

Los ajustes que se aplican solamente a un único paso de tiempo se pueden escribir con `\once`, por ejemplo en

```
c4
\once \set fontSize = #4.7
c4
c4
```



En el manual de Referencia de funcionamiento interno hay una descripción completa de todas las propiedades de contexto disponibles, consulte Traducción ↦ Propiedades de contexto modificables por el usuario.

Véase también

Referencia de funcionamiento interno: Sección “Tunable context properties” in *Referencia de Funcionamiento Interno*.

5.3.3 La instrucción `\override`
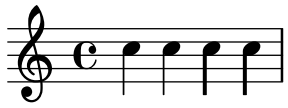
Existe un tipo especial de propiedad de contexto: la descripción de los grobs. Las decscripciones de los grobs reciben un nombre en `MayúsculasDeCamello` (empezando en mayúscula). Contienen los 'ajustes predeterminados' para un tipo particular de grob, en forma de lista asociativa. Consulte '`scm/define-grobs.scm`' para ver los ajustes de cada descripción de grob. Las descripciones de grob se modifican con `\override`.

La sintaxis de la instrucción `\override` es

`\override [`*`contexto`*`.]`*`NombreDelGrob.propiedad`* `= #`*`valor`*
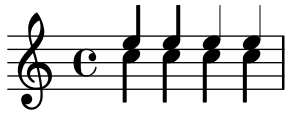
Por ejemplo, podemos aumentar el grosor de la plica de una figura sobreescribiendo la propiedad `thickness` (grosor) del objeto `Stem` (plica):

```
c4 c
\override Voice.Stem.thickness = #3.0
c4 c
```



Si no se ha especificado ningún contexto en la instrucción `\override`, se utiliza el contexto del nivel inferior:

```
{ \override Staff.Stem.thickness = #3.0
  <<
    {
      e4 e
      \override Stem.thickness = #0.5
      e4 e
    } \\ {
      c4 c c c
    }
  >>
}
```



Algunas opciones susceptibles de trucaje, se llaman 'subpropiedades' y residen dentro de las propiedades. Para efectura trujajes sobre ellas, utilice instrucciones de la forma

```
\override Stem.details.beamed-lengths = #'(4 4 3)
```

o para modificar los extremos de los objetos extensos, utilice una forma como las siguientes:

```
\override TextSpanner.bound-details.left.text = #"left text"
\override TextSpanner.bound-details.right.text = #"right text"
```

El efecto de una instrucción de sobreescritura `\override` se puede deshacer con `\revert`.
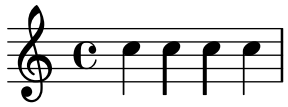
La sintaxis de la instrucción `\revert` es

`\revert [`*`Contexto`*`.]`*`NombreDelGrob.propiedad`*

Por ejemplo,
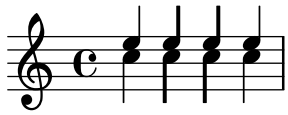
```
c4
\override Voice.Stem.thickness = #3.0
c4 c
```

```
\revert Voice.Stem.thickness
c4
```
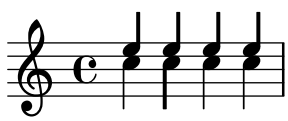


Los efectos de `\override` y `\revert` se aplican a todos los grobs del contexto afectado partiendo del momento actual y hacia adelante:

```
{
  <<
    {
      e4
      \override Staff.Stem.thickness = #3.0
      e4 e e
    } \\ {
      c4 c c
      \revert Staff.Stem.thickness
      c4
    }
  >>
}
```



Se puede usar `\once` con `\override` para afectar solamente al instante de tiempo actual:

```
{
  <<
    {
      \override Stem.thickness = #3.0
      e4 e e e
    } \\ {
      c4
      \once \override Stem.thickness = #3.0
      c4 c c
    }
  >>
}
```



Véase también

Referencia de funcionamiento interno: Sección “Backend” in *Referencia de Funcionamiento Interno*

5.3.4 La instrucción \tweak

El cambio de las propiedades de grob mediante `\override` produce la aplicación del cambio a todos los grobs dados en el contexto en el momento en que se aplica dicho cambio. Sin embargo, en ocasiones podemos desear que los cambios se apliquen a un solo grob en lugar de a todos los grobs del contexto afectado. Esto se consigue con la instrucción `\tweak`, que tiene la sintaxis siguiente:

`\tweak [`*objeto-de-presentación*`.]`*propiedad-del-grob valor*

Es opcional especificar el *objeto-de-presentación*. La instrucción `\tweak` se aplica al objeto musical que viene inmediatamente después de *valor* dentro del flujo musical.
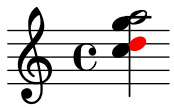
Para ver una introducción a la sintaxis y los usos de la instrucción tweak, consulte Sección “Métodos de trucaje” in *Manual de Aprendizaje*.

Si se colocan varios elementos similares en el mismo momento musical, la instrucción `\override` no se puede usar para modificar uno solo de ellos: aquí es donde se debe usar la instrucción `\tweak`. Entre los elementos que pueden aparecer más de una vez en el mismo momento musical están los siguientes:

- las cabezas de las notas de un acorde
- signos de articulación sobre la misma nota
- ligaduras de unión entre notas de un acorde
- corchetes de grupos especials que comienzan en el mismo momento

En este ejemplo se modifican el color de una cabeza y el tipo de otra, dentro del mismo acorde:

```
< c
  \tweak color #red
  d
  g
  \tweak duration-log #1
  a
> 4
```
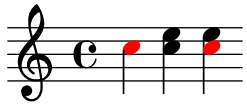


`\tweak` se puede usar para modificar ligaduras de expresión:

```
c-\tweak thickness #5 ( d e f)
```



Para que funcione la instrucción `\tweak`, debe permanecer adyacente al objeto al que se ha de aplicar después de que el código de entrada se ha convertido a un flujo musical. El trucaje de un acorde completo no hace nada porque su evento musical actúa solamente como un contenedor, y todos los objetos de presentación se crean a partir de eventos dentro del `EventChord`:
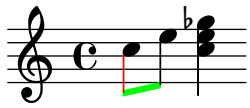
```
\tweak color #red c4
\tweak color #red <c e>4
<\tweak color #red c e>4
```



La instrucción `\tweak` sencilla no se puede usar para modificar ningún objeto que no se haya creado directamente a partir de la entrada. Concretamente, no afecta a las plicas, barras automáticas ni alteraciones, porque éstos se generan posteriormente por parte de objetos de presentación `NoteHead` más que por elementos musicales del flujo de entrada.

Tales objetos de presentación creados indirectamente se pueden trucar usando la forma de la instrucción `\tweak` en que el nombre del grob se especifica de forma explícita:
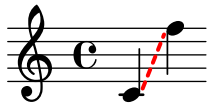
```
\tweak Stem.color #red
\tweak Beam.color #green c8 e
<c e \tweak Accidental.font-size #-3 ges>4
```



No se puede usar `\tweak` para modificar las claves o las indicaciones de compás, porque éstos se separan de cualquier instrucción `\tweak` precedente dentro del flujo de entrada merced a la inserción automática de elementos adicionales que se requieren para especificar el contexto.

Se pueden colocar varias instrucciones `\tweak` antes de un elemento de notación; todos le afectan:

```
c
-\tweak style #'dashed-line
-\tweak dash-fraction #0.2
-\tweak thickness #3
-\tweak color #red
 \glissando
f'
```



El flujo musical que se genera a partir de una sección de un archivo de entrada (incluido cualquier elemento insertado automáticamente) puede examinarse, véase Sección “Presentación de las expresiones musicales” in *Extender*. Esto puede ser de utilidad en la determinación de lo que puede modificarse por medio de una instrucción `\tweak`.

Véase también

Manual de aprendizaje: Sección “Métodos de trucaje” in *Manual de Aprendizaje*.

Manual de extensión: Sección “Presentación de las expresiones musicales” in *Extender*.

Advertencias y problemas conocidos

No se puede usar la instrucción `\tweak` para modificar los puntos de control de una sola de varias ligaduras de unión dentro de un acorde, aparte de la primera que se encuentre en el código de entrada.

5.3.5 `\set` frente a `\override`

Tanto `\set` como `\override` manipulan propiedades asociadas a contextos. En ambos casos las propiedades tienen en cuenta la jerarquía de contextos: las propiedades no establecidas en el propio contexto presentan los valores del contexto padre respectivo.

El valor y la duración en el tiempo de las propiedades de contexto son dinámicos y están disponibles solamente cuando la música se está interpretando o 'iterando'. En el momento de la creación del contexto, se inicializan las propiedades a partir de la definición de contexto correspondiente y las posibles modificaciones de contexto. Después de esto, cualquier cambio se obtiene a través de instrucciones de establecimiento de propiedades dentro de la propia música.

Ahora bien, las definiciones de grob (objetos gráficos) son una clase especial de propiedades de contexto. Dado que su estructura, mantenimiento y utilización es distinta de las propiedades de contexto ordinarias, se accede a ellas con un conjunto de instrucciones diferente, y se estudian por separado dentro de la documentación.

A diferencia de las propiedades de contexto normales, las definiciones de grob están subdivididas en propiedades de grob. Un "grob" (objeto gráfico) se crea normalmente por parte de un grabador en el momento de la interpretación de una expresión musical y recibe sus propiedades iniciales de la definición de grob en curso del contexto del grabador. El grabador (u otras partes del 'backend' de LilyPond) pueden después añadir o modificar propiedades del grob, pero ello no afecta a la definición de grob del contexto.

Lo que conocemos como 'propiedades de grob' en el contexto del trucaje a nivel de usuario son en realidad las propiedades de la definición de grob de un contexto. A diferencia de las propiedades de contexto normales, las definiciones de grob mantienen la contabilidad necesaria para seguir la pista de sus partes, las propiedades de grob individuales (e incluso subpropiedades de éstos), de forma separada, de forma que es posible definir dichas partes dentro de contextos diferentes y hacer que la definición de grob en su conjunto, en el momento de la creación del grob, se monte a partir de las piezas proporcionadas en diferentes contextos entre el contexto actual y sus contextos padre.

Las definiciones de grob se manipulan usando `\override` y `\revert` y tienen un nombre que empieza con una letra mayúscula (como '`NoteHead`') mientras que las propiedades de contexto ordinarias se manipulan utilizando `\set` y `\unset` y se nombran empezando en letra minúscula.

Las instrucciones especiales `\tweak` y `\overrideProperty` cambian las propiedades de grob pasando por encima de las propiedades de contexto completamente. En su lugar, atrapan a los grobs tan pronto se crean, y después establecen propiedades sobre ellos directamente cuando se originan a partir de un evento musical trucado o son de un tipo en particular, respectivamente.

5.3.6 Modificación de las listas-A

Ciertas propiedades configurables por parte del usuario se representan internamente como *listas-A* (listas asociativas), que almacenan duplas de *claves* y *valores*. La estructura de una lista-A es la siguiente:

```
'((clave1 . valor1)
  (clave2 . valor2)
  (clave3 . valor3)
  ...)
```

Si una lista-A es una propiedad de un grob o una variable de `\paper`, sus claves se pueden modificar individualmente sin que afecte a las otras claves.

Por ejemplo, para reducir el espacio entre pentagramas adyacentes dentro de un grupo, use la propiedad `staff-staff-spacing` del grob `StaffGrouper`. La propiedad es una lista-A con cuatro claves: `basic-distance` (distancia básica), `minimum-distance` (distancia mínima), `padding` (relleno) y `stretchability` (ampliabilidad). Los ajustes estándar para esta propiedad

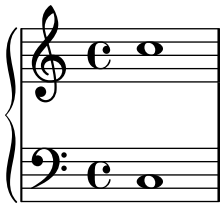
se relacionan en la sección “Backend” de la Referencia de Funcionamiento Interno (véase Sección “StaffGrouper” in *Referencia de Funcionamiento Interno*):

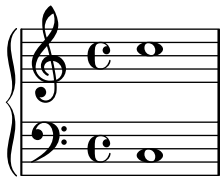
```
'((basic-distance . 9)
  (minimum-distance . 7)
  (padding . 1)
  (stretchability . 5))
```

Una forma de acercar los pentagramas entre sí es reducir el valor de la clave `basic-distance` (9) para que se corresponda con el valor de `minimum-distance` (7). Para modificar una única clave de forma individual, utilice una *declaración anidada*:

```
% default space between staves
\new PianoStaff <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>

% reduced space between staves
\new PianoStaff \with {
  % this is the nested declaration
  \override StaffGrouper.staff-staff-spacing.basic-distance = #7
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>
```
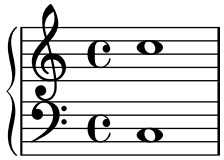




La utilización de una declaración anidada actualiza la clave especificada (como `basic-distance` en el ejemplo anterior) sin alterar ninguna de las otras claves que ya se habían establecido para la misma propiedad.

Ahora, supongamos que deseamos que los pentagramas estén tan próximos como sea posible sin que se superpongan. La manera más sencilla de hacerlo es establecer las cuatro claves de la lista-A a cero. Sin embargo, no es necesario escribir cuatro declaraciones anidadas, una por cada clave. En lugar de eso, se puede redefinir completamente la propiedad con una sola declaración, como una lista-A:

```
\new PianoStaff \with {
  \override StaffGrouper.staff-staff-spacing =
    #'((basic-distance . 0)
       (minimum-distance . 0)
       (padding . 0)
       (stretchability . 0))
```

```
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>
```



Observe que cualquier clave que no haya sido relacionada explícitamente en la definición de la lista-A, será reiniciada a sus valores *predeterminados si no se han fijado*. En el caso de `staff-staff-spacing`, el valor de cualquier clave no fijada se reiniciaría a cero (excepto `stretchability`, que toma el valor de `basic-distance` si no se fija). Así, las dos declaraciones siguientes son equivalentes:

```
\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7))

\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7)
     (minimum-distance . 0)
     (padding . 0)
     (stretchability . 7))
```

Una consecuencia de esto (posiblemente no intencionada) es la eliminación de cualquier valor estándar que se establezca en un archivo de inicio y que se carga cada vez que se compila un archivo de entrada. En el ejemplo anterior, los ajustes estándar para `padding` y `minimum-distance` (definidos en '`scm/define-grobs.scm`') se reinician a sus valores predeterminados si no se han fijado (cero para las dos claves). La definición de una propiedad o variable como una lista-A (de cualquier tamaño) siempre reinicia todos los valores de clave no establecidos a sus valores predeterminados si no se han fijado. Ano ser que este sea el resultado deseado, es más seguro actualizar los valores de clave individualmente con una declaración anidada.

Nota: Las declaraciones anidadas no funcionan para las listas-A de propiedades de contexto (como `beamExceptions`, `keySignature`, `timeSignatureSettings`, etc.). Estas propiedades sólo se pueden modificar redefiniéndolas completamente como listas-A.

5.4 Conceptos y propiedades útiles

5.4.1 Modos de entrada

La forma en que se interpreta la notación contenida dentro de un archivo de entrada, está determinada por el modo de entrada en curso.

Modo de acordes

Se activa con la instrucción `\chordmode` y produce que la entrada se interprete con al sintaxis de la notación de acordes, véase Sección 2.7 [Notación de acordes], página 399. Los acordes se imprimen como notas sobre un pentagrama.

El modo de acordes se activa también con la instrucción `\chords`. Esto crea también un contexto `ChordNames` nuevo y produce que el código que sigue se interprete con la sintaxis de la notación de acordes y se imprima como nombres de acorde dentro del contexto `ChordNames`, véase [Impresión de los nombres de acorde], página 405.

Modo de percusión

Se activa con la instrucción `\drummode` y produce que el código de entrada se interprete con la sintaxis de la notación de percusión, véase [Notación básica de percusión], página 376.

El modo de percusión también se activa con la instrucción `\drums`. También crea un contexto `DrumStaff` nuevo y hace que el código que sigue se interprete con la sintaxis de la notación de percusión y se imprima como símbolos de percusión sobre un pentagrama de percusión, véase [Notación básica de percusión], página 376.

Modo de cifras

Se activa con la instrucción `\figuremode` y hace que el código de entrada se interprete con la sintaxis del bajo cifrado, véase [Introducir el bajo cifrado], página 414.

El modo de cifras también se activa con la instrucción `\figures`. También crea un contexto de `FiguredBass` nuevo y hace que el código que viene a continuación se interprete con la sintaxis del bajo cifrado y se imprima como símbolos de bajo cifrado dentro del contexto `FiguredBass`, véase [Introducción al bajo cifrado], página 413.

Modos de traste y tablatura

No existen modos de entrada especiales para introducir símbolos de trastes y de tablatura.

Para crear diagramas de trastes, escriba las notas o acordes en el modo de notas e imprímalos dentro de un contexto `TabStaff`, véase [Tablaturas predeterminadas], página 332.

Para crear diagramas de trastes encima de un pentagrama, escríbalos como elementos de marcado encima de las notas utilizando la instrucción `\fret-diagram`, véase [Marcas de diagramas de trastes], página 348.

Modo de letra

Se activa con la instrucción `\lyricmode`, y hace que la entrada se interprete como sílabas de la letra de la canción con duraciones opcionales y modificadores de letra asociados, véase Sección 2.1 [Música vocal], página 250.

El modo de letra también se habilita con la instrucción `\addlyrics`. Esto también crea un contexto `Lyrics` nuevo y una instrucción `\lyricsto` implícita que asocia la letra que viene a continuación con la música precedente.

Modo de marcado

Se activa con la instrucción `\markup`, y hace que la entrada se interprete con la sintaxis del marcado, véase ⟨undefined⟩ [Text markup commands], página ⟨undefined⟩.

Modo de notas
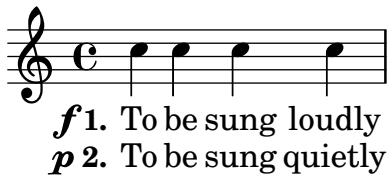
Es el modo predeterminado o se puede activar con la instrucción `\notemode`. La entrada se interpreta como alturas, duraciones, marcado, etc. y se imprime como notación musical sobre un pentagrama.

Normalmente no es necesario especificar el modo de notas de forma explícita, pero puede ser útil hacerlo en ciertas situaciones, por ejemplo si estamos en el modo de letra, en el modo de acordes o en otro modo y queremos insertar algo que solamente se puede hacer con la sintaxis del modo de notas.

Por ejemplo, para insertar indicaciones dinámicas para las estrofas de una pieza coral es necesario entrar en el modo de notas para poder interpretar dichas indicaciones:

```
{ c4 c4 c4 c4 }
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic f 1. } }
  To be sung loudly
}
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic p 2. } }
```

```
  To be sung quietly
}
```



5.4.2 Dirección y posición

Al tipografiar música, la dirección y colocación de muchos elementos es cuestión de elección. Por ejemplo, las plicas de las notas se pueden dirigir hacia arriba o hacia abajo; la letra, las indicaciones dinámicas y otras marcas expresivas se pueden colocar encima o debajo del pentagrama; el texto se pude alinear a la izquierda, a la derecha o centrado; etc. La mayoría de estas elecciones pueden dejarse que LilyPond las determine automáticamente, pero en ciertos casos puede ser deseable forzar una dirección o colocación concreta.

Indicadores de dirección de las articulaciones

De forma predeterminada algunas direcciones siempre son hacia arriba o siempre hacia abajo (p. ej. los matices o el calderón), mientras que otras cosas pueden alternar entre arriba y abajo en función de la dirección de las plicas (como las ligaduras o los acentos).

Se puede sobreescribir la acción predeterminada mediante el prefijado de la articulación por un *indicacor de dirección*. Están disponibles tres indicadores de dirección: `^` (que significa "arriba"), `_` (que significa "abajo") o `-` (que significa "usar la dirección predeterminada"). El indicador de dirección se puede normalmente omitir, en cuyo caso se supone el indicador predeterminado `-`, pero se necesita un indicador de dirección **siempre** antes de:

- las instrucciones `\tweak`
- las instrucciones `\markup`
- las instrucciones `\tag`
- los marcados de cadena, p.ej. `-"cadena"`
- las instrucciones de digitación, p.ej. `-1`
- las abreviaturas de articulación, p.ej. `-.`, `->`, `--`

Estas indicaciones afectan sólo a la nota siguiente.

```
c2( c)
c2_( c)
c2( c)
c2^( c)
```



La propiedad de dirección

La posición o dirección de muchos objetos de presentación está controlada por la propiedad `direction`.

El valor de la propiedad `direction` se puede establecer al valor `1`, con el significado de "hacia arriba" o "encima", o a `-1`, con el significado de "hacia abajo" o "debajo". Se pueden usar los símbolos `UP` y `DOWN` en sustitución de `1` y `-1` respectivamente. La dirección predeterminada se puede especificar estableciendo `direction` a `0` ó a `CENTER`. De forma alternativa, en muchos casos existen instrucciones predefinidas para especificar la dirección. Todas ellas son de la forma:
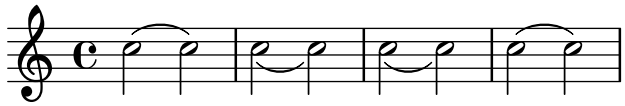
`\xxxUp, \xxxDown o \xxxNeutral`

donde `\xxxNeutral` significa "utilizar la dirección predeterminada". Véase Sección "Objetos interiores al pentagrama" in *Manual de Aprendizaje*.

En alguna que otra ocasión como en el arpegio, el valor de la propiedad `direction` puede especificar si el objeto se debe colocar a la izquierda o a la derecha del objeto padre. En este caso `-1` ó `LEFT` significan "a la izquierda" y `1` ó `RIGHT` significan "a la derecha". `0` ó `CENTER` significan "utilizar la dirección predeterminada".

Estas indicaciones afectan a todas las notas hasta que son canceladas.

```
c2( c)
\slurDown
c2( c)
c2( c)
\slurNeutral
c2( c)
```



En música polifónica, en general es mejor especificar una voz explícita que cambiar la dirección de un objeto. Para ver más información, véase Sección 1.5.2 [Varias voces], página 166.

Véase también

Manual de aprendizaje: Sección "Objetos interiores al pentagrama" in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.5.2 [Varias voces], página 166.

5.4.3 Distancias y medidas

Las distancias en LilyPond son de dos tipos: absolutas y escaladas.

Las distancias absolutas se usan para especificar márgenes, sangrados y otros detalles de diseño de página, y de forma predeterminada se especifican en milímetros. Las distancias se pueden especificar en otras unidades escribiendo después de la cifra indicativa de la cantidad, `\mm`, `\cm`, `\in` (pulgadas), o `\pt` (puntos, 1/72.27 pulgadas). Las distancias de diseño de página se pueden especificar también en unidades escalables (véase el párrafo siguiente) adjuntando `\staff-space` a la cantidad. La disposición de página se describe en detalle en Sección 4.1 [Disposición de la página], página 513.

Las distancias escaladas siempre se especifican en unidades de un espacio del pentagrama o, más raramente, medio espacio del pentagrama. El espacio de pentagrama es la distancia entre dos líneas del pentagrama adyacentes. El valor predeterminado se puede cambiar globalmente fijando el tamaño global del pentagrama, o se puede sobreescribir localmente cambiando la propiedad `staff-space` del objeto `StaffSymbol`. Las distancias escaladas se escalan automáticamente con cualquier cambio al tamaño global del pentagrama o a la propiedad `staff-space` del objeto `StaffSymbol`, pero las fuentes tipográficas se escalan solamente con los cambios efectuados al tamaño global del pentagrama. Así, el tamaño global del pentagrama posibilita la fácil variación del tamaño general de una partitura impresa. Para ver los métodos de establecimiento del tamaño global del pentagrama, véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.

Si se necesita dibujar sólo una sección de una partitura a una escala distinta, por ejemplo una sección ossia o una nota al pie, no se puede simplemente cambiar el tamaño global del pentagrama porque esto afectaría a toda la partitura. En tales casos, el cambio de tamaño se hace sobreescribiendo tanto la propiedad `staff-space` de `StaffSymbol` como el tamaño de las

fuentes tipográficas. Está a nuestra disposición una función de Scheme, `magstep`, para convertir de un cambio en el tamaño de la fuente al cambio equivalente en `staff-space`. Para ver una explicación y un ejemplo de su utilización, consulte Sección “Longitud y grosor de los objetos” in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Longitud y grosor de los objetos” in *Manual de Aprendizaje*.
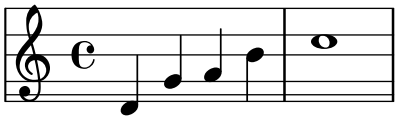
Referencia de la notación: Sección 4.1 [Disposición de la página], página 513, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 526.

5.4.4 Propiedades del símbolo del pentagrama

Se puede definir al mismo tiempo la posición vertical de las líneas de la pauta y el número de líneas de la misma. Como muestra el siguiente ejemplo, las posiciones de las notas no están influidas por las posiciones de las líneas de la pauta.
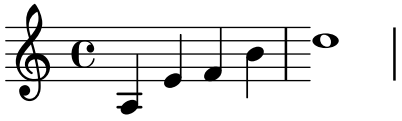
Nota: La propiedad `'line-positions` sobreescribe a la propiedad `'line-count`. El número de líneas de la pauta está definido implícitamente por el número de elementos de la lista de valores de `'line-positions`.

```
\new Staff \with {
  \override StaffSymbol.line-positions = #'(7 3 0 -4 -6 -7)
}
{ a4 e' f b | d1 }
```



Se puede modificar la anchura de la pauta. Las unidades son espacios de pentagrama. El espaciado de los objetos dentro del pentagrama no resulta afectado por este ajuste.

```
\new Staff \with {
  \override StaffSymbol.width = #23
}
{ a4 e' f b | d1 }
```



5.4.5 Objetos de extensión

Muchos objetos de notación musical abarcan varias notas o incluso varios compases. Son ejemplos los crescendi, trinos, corchetes de grupo especial y corchetes de primera y segunda vez. Estos objetos se llaman “spanners” u «objetos de extensión», y tienen propiedades especiales para controlar su apariencia y comportamiento. Algunas de estas propiedades son comunes a todos los objetos de extensión; otras se limitan a un subconjunto de los extensores.

Todos los objetos de extensión contemplan el interface `spanner-interface`. Algunos, básicamente aquellos que trazan una línea recta entre los dos objetos, contemplan también el interface `line-spanner-interface`.
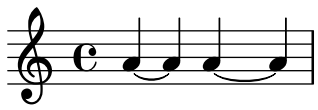
Uso del `spanner-interface`

Este interface proporciona dos propiedades que se aplican a varios extensores.
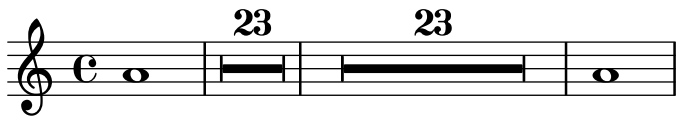
La propiedad `minimum-length` *(longitud mínima)*

La longitud mínima del objeto de extensión se especifica a través de la propiedad `minimum-length`. Su aumento suele producir el efecto necesario de aumentar el espaciado de las notas entre los dos puntos extremos. Sin embargo, esta sobreescritura no tiene ningún efecto sobre muchos extensores, pues su longitud está determinada por otras consideraciones. Más abajo se muestran algunos ejemplos de dónde es efectiva.
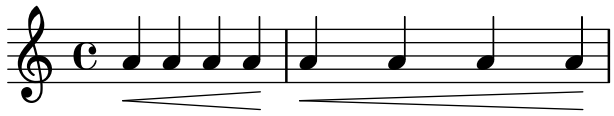
```
a~ a
a
% increase the length of the tie
-\tweak minimum-length #5
~ a
```



```
a1
\compressFullBarRests
R1*23
% increase the length of the rest bar
\once \override MultiMeasureRest.minimum-length = #20
R1*23
a1
```




```
a \< a a a \!
% increase the length of the hairpin
\override Hairpin.minimum-length = #20
a \< a a a \!
```



Esta sobreescritura se puede usar también para aumentar la longitud de las ligaduras de expresión y de fraseo:

```
a( g)
a
-\tweak minimum-length #5
( g)

a\( g\)
a
-\tweak minimum-length #5
\( g\)
```

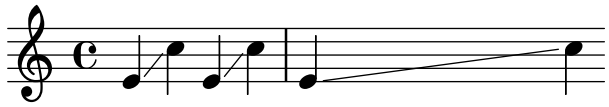


Para algunos objetos de presentación, la propiedad `minimum-length` es efectiva sólo si se llama explícitamente al procedimiento `set-spacing-rods`. Para hacerlo, se debe fijar la propiedad `springs-and-rods` al valor `ly:spanner::set-spacing-rods`. Por ejemplo, la longitud mínima de un glissando no tiene efecto a no ser que se establezca la propiedad `springs-and-rods`:

```
% default
e \glissando c'

% not effective alone
\once \override Glissando.minimum-length = #20
e, \glissando c'
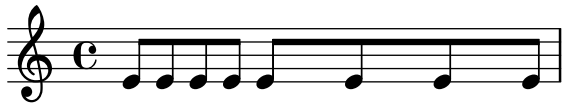
% effective only when both overrides are present
\once \override Glissando.minimum-length = #20
\once \override Glissando.springs-and-rods = #ly:spanner::set-spacing-rods
e, \glissando c'
```



Lo mismo se puede decir del objeto `Beam`:

```
% not effective alone
\once \override Beam.minimum-length = #20
e8 e e e

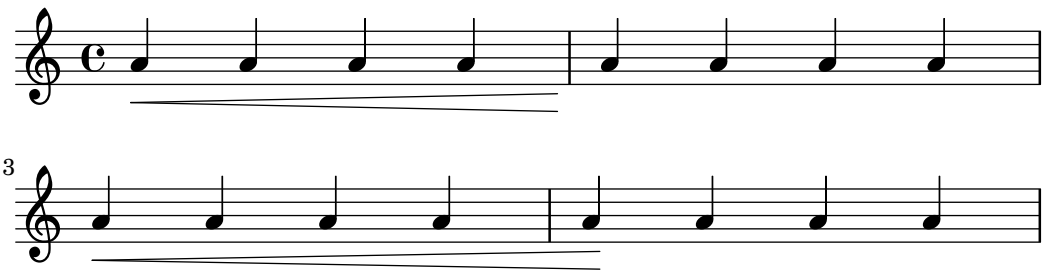
% effective only when both overrides are present
\once \override Beam.minimum-length = #20
\once \override Beam.springs-and-rods = #ly:spanner::set-spacing-rods
e8 e e e
```



La propiedad `to-barline`

La segunda propiedad útil del `spanner-interface` es `to-barline`. De forma predeterminada tiene el valor cierto, haciendo que los reguladores y otros objetos de extensión que terminan sobre la primera nota de un compás, en vez de eso terminen en la línea divisoria inmediatamente precedente. Si se establece al valor falso, el extensor llegará más allá de la barra de compás y terminará exactamente sobre la nota:

```
a \< a a a a \! a a a \break
\override Hairpin.to-barline = ##f
a \< a a a a \! a a a
```



Esta propiedad no es efectiva para todos los extensores. Por ejemplo, su establecimiento a `#t` no tienen ningún efecto sobre las ligaduras de expresión o de fraseo, o sobre otros extensores para los que terminar en la barra de compás no tendría ningún significado.
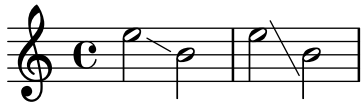
Uso del `line-spanner-interface`

Entre los objetos que contemplan el interface `line-spanner-interface` se encuentran

- `DynamicTextSpanner`
- `Glissando`
- `TextSpanner`
- `TrillSpanner`
- `VoiceFollower`

La rutina responsable de dibujar los sellos de estos extensores es `ly:line-interface::print`. esta rutina determina la localización exacta de los dos puntos extremos y traza una línea entre ellos, en el estilo solicitado. Las posiciones de los dos puntos extremos del extensor se calculan al vuelo, pero es posible sobreescribir sus coordenadas Y. Las propiedades que se deben especificar están anidadas a dos niveles de profundidad en la jerarquía de propiedades, pero la sintaxis de la instrucción `\override` es bastante sencilla:

```
e2 \glissando b
\once \override Glissando.bound-details.left.Y = #3
\once \override Glissando.bound-details.right.Y = #-2
e2 \glissando b
```
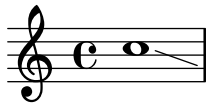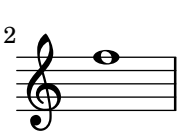


Las unidades para la propiedad `Y` son `staff-space`s, siendo el punto del cero la línea central del pentagrama. Para el glissando, esto es el valor de `Y` en la coordenada X que corresponde al punto central de cada cabeza de nota si nos imaginamos que la línea se extiende hasta allí.

Si no está fijado `Y`, su valor se calcula a partir de la posición vertical del punto de anclaje correspondiente del extensor.

En caso de salto de línea, los valores para los puntos extremos se especifican por las sub-listas `left-broken` y `right-broken` de `bound-details`. Por ejemplo:

```
\override Glissando.breakable = ##t
\override Glissando.bound-details.right-broken.Y = #-3
c1 \glissando \break
f1
```





Un número de propiedades adicionales de las sub-listas `left` y `right` de la propiedad `bound-details` se pueden especificar de la misma forma que `Y`:

`Y` Establece la coordenada Y del punto extremo, en desplazamientos de `staff-space`s desde la línea central del pentagrama. De forma predeterminada es el centro del objeto ancla, y así un glissando apunta al centro vertical de la cabeza de la nota.

Para extensores horizontales como los extensores de texto y los trinos, está inamoviblemente codificado como 0.
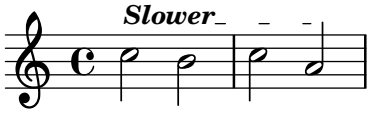
`attach-dir (dirección de anclaje)`
Determina dónde comienza y termina la línea en la dirección X, con relación al objeto ancla. Sí, un valor de `-1` (o `LEFT`, izquierda) hace que la línea comience o termine en el lado izquierdo de la cabeza de la nota a la que está anclado.

`X` Es la coordenada X absoluta del punto extremo. Se suele calcular al vuelo, y su sobreescritura no tiene un efecto útil.

`stencil (sello)`
Los extensores de línea pueden tener símbolos al comienzo o al final, lo que está contenido en esta sub-propiedad. Esto es para uso interno; se recomienda en su lugar el uso de `text`.

`text (texto)`
Es un elemento de marcado que se evalúa para dar lugar al sello. Se usa para escribir *cresc.*, *tr* y otros textos sobre los objetos de extensión horizontales.
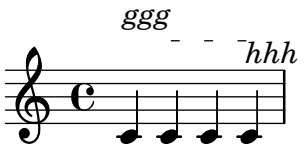
```
\override TextSpanner.bound-details.left.text
   = \markup { \small \bold Slower }
c2\startTextSpan b c a\stopTextSpan
```



`stencil-align-dir-y (alineación del sello en y)`
`stencil-offset (desplazamiento del sello)`
Si no se modifican uno u otro, el sello se coloca sencillamente en el punto extremo, centrado sobre la línea, como viene definido por las subpropiedades `X` e `Y`. Si se fijan `stencil-align-dir-y` o `stencil-offset` se moverá el símbolo del borde verticalmente con relación al extremo de la línea:

```
\override TextSpanner.bound-details.left.stencil-align-dir-y = #-2
\override TextSpanner.bound-details.right.stencil-align-dir-y = #UP

\override TextSpanner.bound-details.left.text = #"ggg"
\override TextSpanner.bound-details.right.text = #"hhh"
c4^\startTextSpan c c c \stopTextSpan
```
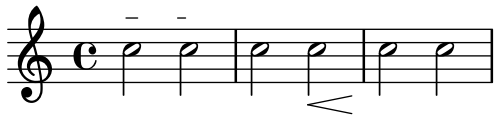


Observe que los valores negativos mueven el texto *hacia arriba*, al contrario de lo que podría esperarse, pues el valor de `-1` o `DOWN` (abajo) significa alinear el borde *inferior* del texto con la línea de extensión. Un valor de `1` o `UP` (arriba) alinea el borde superior del texto con la línea extensora.

`arrow (flecha)`
: Al establecer esta sub-propiedad a `#t` se produce una punta de flecha en el extremo de la línea.

`padding (relleno)`
: Esta sub-propiedad controla el espacio entre el punto extremo especificado de la línea y el extremo real. Sin relleno, un glissando empezaría y terminaría en el centro de la cabeza de las notas.

La función musical `\endSpanners` finaliza de forma prematura el extensor que comienza sobre la nota que sigue inmediatamente a continuación. Se termina después de una nota exactamente, o en la siguiente barra de compás si `to-barline` es verdadero y se produce una divisoria antes de la siguiente nota.

```
\endSpanners
c2 \startTextSpan c2 c2
\endSpanners
c2 \< c2 c2
```



Si se usa `\endSpanners` no es necesario cerrar \startTextSpan con \stopTextSpan, ni cerrar los reguladores con `\!`.

Véase también

Referencia de funcionamiento interno: Sección “TextSpanner” in *Referencia de Funcionamiento Interno*, Sección “Glissando” in *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” in *Referencia de Funcionamiento Interno*, Sección “TrillSpanner” in *Referencia de Funcionamiento Interno*, Sección “line-spanner-interface” in *Referencia de Funcionamiento Interno*.

5.4.6 Visibilidad de los objetos

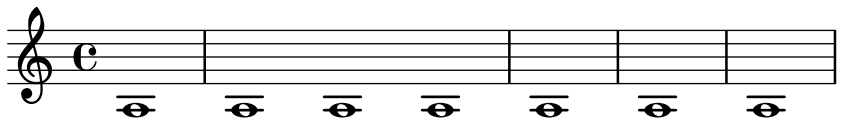
Hay cuatro formas principales en que se puede controlar la visibilidad de los objetos de presentación: se puede eliminar su sello, se pueden volver transparentes, se pueden pintar de blanco, o se puede sobreescribir su propiedad `break-visibility`. Las tres primeras se aplican a todos los objetos de presentación; la última sólo a unos pocos: los objetos *divisibles*. El Manual de aprendizaje introduce estas cuatro técnicas, véase Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*.

Hay también algunas otras técnicas que son específicas de ciertos objetos de presentación. Se estudian bajo Consideraciones especiales.

Eliminar el sello

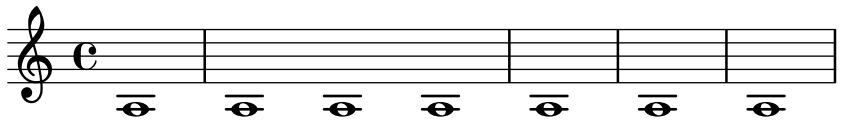
Todo objeto de presentación tiene una propiedad stencil (sello). De forma predeterminada está establecida a la función específica que dibuja ese objeto. Si se sobreescribe esta propiedad a `#f` no se llama a ninguna función y el objeto no se dibuja. La acción predeterminada se puede recuperar con `\revert`.

```
a1 a
\override Score.BarLine.stencil = ##f
a a
\revert Score.BarLine.stencil
a a a
```



Esta operación, bastante común, tiene la forma abreviada `\omit` como atajo:
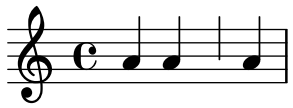
```
a1 a
\omit Score.BarLine
a a
\undo \omit Score.BarLine
a a a
```



Hacer transparentes los objetos
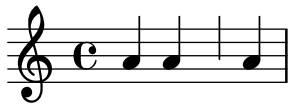
Todo objeto de presentación tiene una propiedad transparent (transparente) que de forma predeterminada está establecida a `#f`. Si se fija a `#t` el objeto aún ocupa espacio pero es invisible.

```
a4 a
\once \override NoteHead.transparent = ##t
a a
```



Esta operación, bastante común, tiene la forma abreviada `\hide` como atajo:

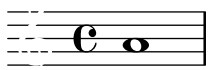
```
a4 a
\once \hide NoteHead
a a
```



Pintar los objetos de blanco

Todo objeto de presentación tiene una propiedad de color que de forma predeterminada está establecida a `black` (negro). Si se sobreescribe a `white` (blanco) el objeto será indistinguible del fondo blanco. Sin embargo, si el objeto cruza a otros objetos, el color de los puntos de cruce queda determinado por el orden en que se dibujan estos objetos, lo que puede dejar una imagen fantasma del objeto blanco, como puede verse aquí:

```
\override Staff.Clef.color = #white
a1
```



Se puede evitar esto cambiando el orden de impresión de los objetos. Todos los objetos de presentación tienen una propiedad `layer` (capa) que se debe establecer a un valor entero. Los objetos con el valor de `layer` más bajo se dibujan primero, después se dibujan los objetos con valores progresivamente mayores, y así los objetos con valores más altos se dibujan encima de los que tienen valores más bajos. De forma predeterminada, casi todos los objetos tienen asignado

un valor `layer` de `1`, aunque algunos objetos, entre ellos el pentagrama y las líneas divisorias, `StaffSymbol` y `BarLine`, tienen asignado un calor de `0`. El orden de impresión de los objetos con el mismo valor de `layer` es indeterminado.

En el ejemplo de arriba, la clave blanca, con un valor `layer` predeterminado de `1`, se dibuja después de las líneas del pentagrama (valor `layer` predeterminado de `0`), sobreimpresionándolas. Para cambiarlo, se debe dar al objeto `Clef` un valor de `layer` más bajo, digamos `-1`, para que se dibuje antes:

```
\override Staff.Clef.color = #white
\override Staff.Clef.layer = #-1
a1
```



Uso de break-visibility (visibilidad en el salto)

Casi todos los objetos de presentación se imprimen una sola vez, pero algunos como las líneas divisorias, claves, indicaciones de compás y armaduras de tonalidad, se pueden tener que imprimir dos veces cuando se produce un salto de línea : una vez al final de la línea y otra al comienzo de la siguiente. Estos objetos reciben el nombre de *divisibles*, y tienen una propiedad, `break-visibility` (visibilidad en el salto), para controlar su visibilidad en las tres posiciones en que pueden aparecer: al comienzo de una línea, dentro de la línea si se produce un cambio, y al final de la línea si el cambio se produce en ese lugar.

Por ejemplo, la indicación de compás se imprime de forma predeterminada al comienzo de la primera línea y en ningún otro lugar a no ser que cambie, en cuyo caso se imprime en el punto en que se produce el cambio. Si este cambio se produce al final de una línea, la nueva indicación de compás se imprime al principio de la línea siguiente y también al final de la línea anterior como indicación de precaución.

Este comportamiento se controla por medio de la propiedad `break-visibility`, que se explica en Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*. Esta propiedad toma un vector de tres valores booleanos que, por orden, determinan si el objeto se imprime al final, dentro, o al principio de la línea. O, para ser más exactos: antes del salto de línea, si no hay salto, o después del salto.

Como alternativa se puede especificar cualquiera de las ocho combinaciones mediante funciones predefinidas cuya definición está en ‘`scm/output-lib.scm`’, donde las tres últimas columnas indican si los objetos de presentación serán visibles en las posiciones que se muestran en el encabezamiento de cada columna:
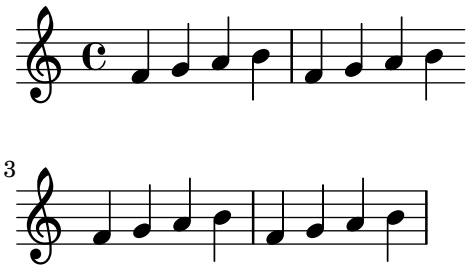
Forma de función	**Forma de vector**	**Antes del salto**	**Si no hay salto**	**Después del salto**
`all-visible`	`'#(#t #t #t)`	sí	sí	sí
`begin-of-line-visible`	`'#(#f #f #t)`	no	no	sí
`center-visible`	`'#(#f #t #f)`	no	sí	no
`end-of-line-visible`	`'#(#t #f #f)`	sí	no	no
`begin-of-line-invisible`	`'#(#t #t #f)`	sí	sí	no
`center-invisible`	`'#(#t #f #t)`	sí	no	sí
`end-of-line-invisible`	`'#(#f #t #t)`	no	sí	sí
`all-invisible`	`'#(#f #f #f)`	no	no	no

Los ajustes predeterminados de `break-visibility` dependen del objeto de presentación. La tabla siguiente muestra todos los objetos de presentación de interés que resultan afectados por `break-visibility` y el ajuste predeterminado de esta propiedad:

Objeto	**Contexto usual**	**Valor predet.**
`BarLine`	`Score`	calculado
`BarNumber`	`Score`	`begin-of-line-visible`
`BreathingSign`	`Voice`	`begin-of-line-invisible`
`Clef`	`Staff`	`begin-of-line-visible`
`Custos`	`Staff`	`end-of-line-visible`
`DoublePercentRepeat`	`Voice`	`begin-of-line-invisible`
`KeyCancellation`	`Staff`	`begin-of-line-invisible`
`KeySignature`	`Staff`	`begin-of-line-visible`
`ClefModifier`	`Staff`	`begin-of-line-visible`
`RehearsalMark`	`Score`	`end-of-line-invisible`
`TimeSignature`	`Staff`	`all-visible`

El ejemplo de abajo muestra el uso de la forma de vector para controlar la visibilidad de las líneas divisorias:

```
f4 g a b
f4 g a b
% Remove bar line at the end of the current line
\once \override Score.BarLine.break-visibility = ##(#f #t #t)
\break
f4 g a b
f4 g a b
```



Aunque los tres componentes del vector utilizado para sobreescribir `break-visibility` deben estar presentes, no todos son efectivos para todos los objetos de presentación, y algunas combinaciones pueden incluso dar errores. Son de aplicación las siguientes limitaciones:
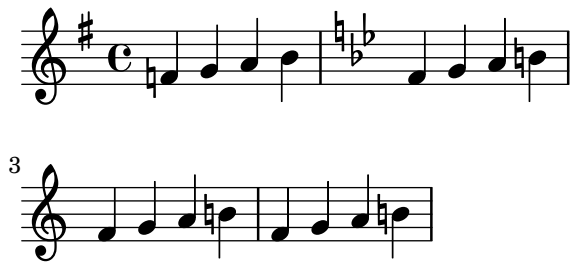
- Las líneas divisorias no se pueden imprimir al principio de la línea.
- No se puede imprimir el número de compás al principio de la primera línea a no ser que su valor establecido sea distinto de 1.
- Clave: véase más abajo
- Las repeticiones dobles de tipo porcentaje se imprimen completamente o se suprimen por completo. Utilice begin-of line-invisible para imprimirlas y all-invisible para suprimirlas.
- Armadura: véase más abajo
- ClefModifier: véase más abajo

Consideraciones especiales

Visibilidad después de un cambio explícito
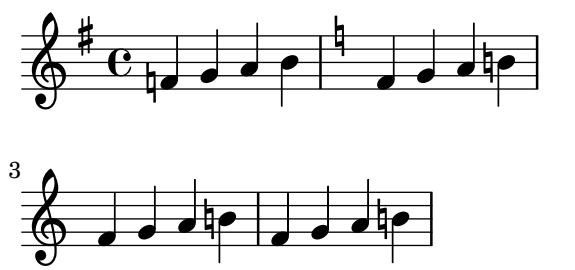
La propiedad `break-visibility` controla la visibilidad de las armaduras y cambios de clave sólo al principio de las líneas, es decir, después de un salto. No tiene ningún efecto sobre la visibilidad de la armadura o la clave después de un cambio explícito de tonalidad o de clave dentro o al final de una línea. En el ejemplo siguiente la armadura que sigue al cambio explícito de tonalidad a Si bemol mayor es visible incluso con `all-invisible` establecido.

```
\key g \major
f4 g a b
% Try to remove all key signatures
\override Staff.KeySignature.break-visibility = #all-invisible
\key bes \major
f4 g a b
\break
f4 g a b
f4 g a b
```



La visibilidad de estos cambios explícitos de tonalidad y de clave se controla por medio de las propiedades `explicitKeySignatureVisibility` y `explicitClefVisibility`. Son los equivalentes de la propiedad `break-visibility` y las dos toman un vector de tres valores booleanos o las funciones predefinidas relacionadas anteriormente, exactamente igual que `break-visibility`. Las dos son propiedades del contexto Staff, no de los propios objetos de presentación, y por tanto se establecen utilizando la instrucción `\set`. Las dos están establecidas de forma predeterminada al valor `all-visible`. Estas propiedades controlan sólo la visibilidad de las armaduras y las claves que resultan de los cambios explícitos y no afectan a las armaduras y tonalidades que están al principio de las líneas; para quitarlas, aún se debe sobreescribir la propiedad `break-visibility` en el objeto correspondiente.
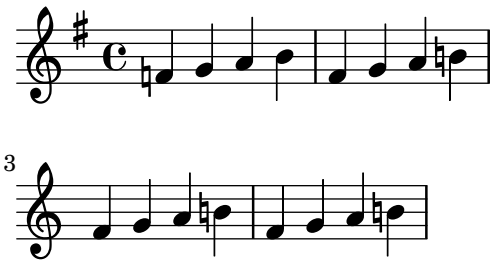
```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeySignature.break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```



Visibilidad de las alteraciones de precaución

Para eliminar las alteraciones de cancelación que se imprimen en un cambio de tonalidad explícito, establezca la propiedad `printKeyCancellation` del contexto Staff a `#f`:
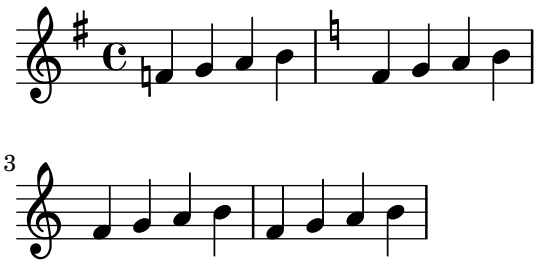
```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.printKeyCancellation = ##f
\override Staff.KeySignature.break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```



Con estas sobreescrituras solamente permanecen las alteraciones accidentales delante de las notas para indicar el cambio de tonalidad.

Observe que al modificar la tonalidad a Do mayor o La menor, las alteraciones de cancelación serían *la única* indicación del cambio de armadura. En este caso, el establecimiento de `printKeyCancellation` al valor `#f` no tiene ningún efecto:

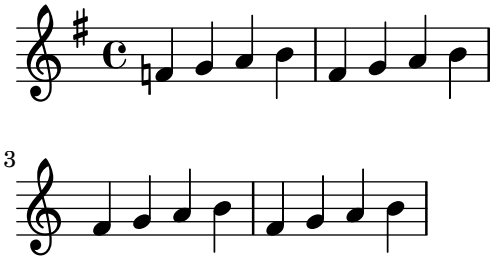
```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.printKeyCancellation = ##f
\key c \major
f4 g a b \break
f4 g a b
f4 g a b
```



En lugar de ello, para suprimir las alteraciones de cancelación aún cuando la armadura cambia a Do mayor o a La menor, sobreescriba la visibilidad del grob `KeyCancellation`:

```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeyCancellation.break-visibility = #all-invisible
\key c \major
```

```
f4 g a b \break
f4 g a b
f4 g a b
```



Líneas divisorias automáticas

Como caso especial, la impresión de las líneas divisorias también se puede inhabilitar estableciendo la propiedad `automaticBars` en el contexto Score. Si se fija a `#f`, las barras de compás no se imprimen automáticamente; se deben crear explícitamente con una instrucción `\bar`. A diferencia de la instrucción predefinida `\cadenzaOn`, los compases se siguen contando. La generación de compases continúa de acuerdo a esta cuenta si esta propiedad se establece posteriormente a `#t`. Si se fija al valor `#f`, sólo pueden producirse saltos de línea en instrucciones `\bar` explícitas.

Claves transportadas

El pequeño símbolo de transposición sobre una clave se produce por parte del objeto `ClefModifier`. Su visibilidad se hereda automáticamente de la del objeto `Clef`, así que no es necesario aplicar las sobreescrituras correspondientes `break-visibility` a los objetos `ClefModifier` para suprimir los símbolos de transposición u octavación para las claves invisibles.

Para los cambios de clave explícitos, la propiedad `explicitClefVisibility` controla tanto el símbolo de clave como el símbolo de transposición o de octava asociado.

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” in *Manual de Aprendizaje*

5.4.7 Estilos de línea

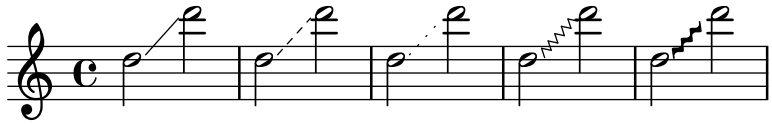
Ciertas indicaciones de ejecución, p.ej., *rallentando*, *accelerando* y los *trinos* se escriben como texto y se extienden sobre muchos compases mediante líneas, a veces punteadas u onduladas.

Todos ellos usan las mismas rutinas que el glissando para dibujar los textos y las líneas, y por ello el ajuste fino de su comportamiento se hace de la misma manera. Se hace con un spanner (un objeto de extensión), y la rutina responsable de dibujar los objetos de extensión es `ly:line-interface::print`. Esta rutina determina la colocación exacta de los dos *extremos del objeto de extensión* y dibuja una línea entre ellos, en el estilo solicitado.

He aquí un ejemplo de los distintos estilos de línea disponibles, y cómo aplicarles ajustes finos.
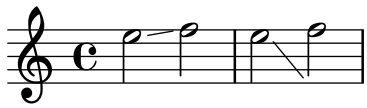
```
d2 \glissando d'2
\once \override Glissando.style = #'dashed-line
d,2 \glissando d'2
\override Glissando.style = #'dotted-line
d,2 \glissando d'2
\override Glissando.style = #'zigzag
d,2 \glissando d'2
\override Glissando.style = #'trill
```

```
d,2 \glissando d'2
```



Las posiciones de los puntos extremos del objeto de extensión se computan al vuelo para cada uno de los objetos gráficos, pero es posible sobreescribirlos:

```
e2 \glissando f
\once \override Glissando.bound-details.right.Y = #-2
e2 \glissando f
```



El valor de `Y` está establecido a `-2` para el extremo derecho. El lado izquierdo se puede ajustar de forma similar especificando `left` en vez de `right`.

Si no está establecido `Y`, el valor se computa a partir de la posición vertical de los puntos de anclaje izquierdo y derecho del extensor.

Son posibles otros ajustes de los extensores, para ver más detalles consulte Sección 5.4.5 [Objetos de extensión], página 603.

5.4.8 Rotación de objetos

Tanto los objetos de presentación como los elementos de texto de marcado se pueden girar cualquier ángulo respecto a cualquier punto, pero difiere el método de hacerlo.

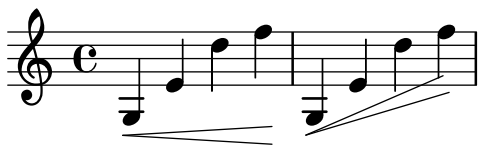
Rotación de objetos de presentación

Todos los objetos de presentación que contemplan el interface `grob-interface` se pueden rotar estableciendo su propiedad `rotation`. Acepta una lista de tres elementos: el ángulo de la rotación en sentido antihorario, y las coordenadas x e y del punto con relación al punto de referencia del objeto, alrededor del que se va a realizar la rotación. El ángulo de rotación se especifica en grados y las coordenadas en espacios de pentagrama.

El ángulo de rotación y las coordenadas del punto de rotación se deben determinar por ensayo y error.

Solamente en ciertas ocasiones es útil la rotación de objetos de presentación; el ejemplo siguiente muestra una situación en que puede serlo:
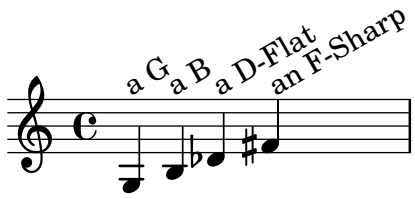
```
g4\< e' d' f\!
\override Hairpin.rotation = #'(20 -1 0)
g,,4\< e' d' f\!
```



Rotación de elementos de marcado

Todos los textos de marcado se pueden rotar para que se dispongan en cualquier ángulo precediéndolos de la instrucción `\rotate`. La instrucción acepta dos argumentos: el ángulo de rotación en grados en sentido antihorario, y el texto que rotar. Los límites que ocupa el texto no se rotan: toman su valor a partir de los extremos de las coordenadas x e y del texto rotado. En el ejemplo siguiente la propiedad `outside-staff-priority` del texto se establece a `#f` para desactivar la evitación automática de colisiones, lo que empuja al texto a una posición muy alta.

```
\override TextScript.outside-staff-priority = ##f
g4^\markup { \rotate #30 "a G" }
b^\markup { \rotate #30 "a B" }
des^\markup { \rotate #30 "a D-Flat" }
fis^\markup { \rotate #30 "an F-Sharp" }
```



5.5 Trucos avanzados

Esta sección trata sobre distintos enfoques en la realización de ajustes finos a la apariencia de la partitura impresa.

Véase también

Manual de aprendizaje: Sección “Trucar la salida” in *Manual de Aprendizaje*, Sección “Otras fuentes de información” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2 [Explicación del Manual de referencia de funcionamiento interno], página 587, Sección 5.3 [Modificar las propiedades], página 590.

Manual de extensión: Sección “Interfaces para programadores” in *Extender*.

Archivos de inicio: ‘`scm/define-grobs.scm`’.

Fragmentos de código: Sección “Tweaks and overrides” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “All layout objects” in *Referencia de Funcionamiento Interno*.

5.5.1 Alineación de objetos

Los objetos gráficos que soportan el interface `self-alignment-interface` y/o el `side-position-interface` se pueden alinear contra un objeto colocado previamente, de diversas maneras. Para ver una lista de estos objetos, consulte Sección “self-alignment-interface” in *Referencia de Funcionamiento Interno* y Sección “side-position-interface” in *Referencia de Funcionamiento Interno*.

Todos los objetos gráficos tienen un punto de referencia, una extensión horizontal y una extensión vertical. La extensión horizontal es una pareja de números que dan los desplazamientos a partir del punto de referencia de los bordes izquierdo y derecho, siendo negativos los desplazamientos hacia la izquierda. La extensión vertical es una pareja de números que dan el desplazamiento a partir del punto de referencia hasta los bordes inferior y superior, siendo negativos los desplazamientos hacia abajo.

La posición de un objeto sobre el pentagrama viene dada por los valores de las propiedades `X-offset` e `Y-offset`. El valor de `X-offset` da el desplazamiento desde la coordenada X del

punto de referencia del objeto padre, y el valor de `Y-offset` da el desplazamiento a partir de la línea central del pentagrama. Los valores de `X-offset` y `Y-offset` se pueden establecer directamente o se puede dejar que se calculen por parte de procedimientos para conseguir una alineación con el objeto padre.

> **Nota:** Muchos objetos tienen consideraciones de posicionamiento especiales que hacen que se ignore o se modifique cualquier ajuste realizado a `X-offset` o a `Y-offset`, a pesar de que el objeto contemple el interface `self-alignment-interface`. La sobreescritura de las propiedades `X-offset` ó `Y-offset` a un valor fijo hace que se descarte la respectiva propiedad `self-alignment`.

Por ejemplo, una alteración accidental se puede reposicionar verticalmente estableciendo `Y-offset` pero los cambios a `X-offset` no tienen ningún efecto.

Las letras de ensayo se pueden alinear con objetos divisibles como líneas divisorias, claves, armaduras e indicaciones de compás. Hay propiedades especiales que se encuentran en `break-aligned-interface` para el posicionamiento de las letras de ensayo sobre dichos objetos.

Véase también
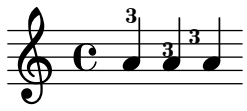
Referencia de la notación: [Uso del break-alignable-interface], página 619.

Manual de extensión: Sección “Funciones de callback” in *Extender*.

Establecer directamente `X-offset` y `Y-offset`

Se pueden dar valores numéricos a las propiedades `X-offset` y `Y-offset` de muchos objetos. El ejemplo siguiente muestra tres notas con una digitación en su posición predeterminada y con los valores `X-offset` y `Y-offset` modificados.

```
a-3
a
-\tweak X-offset #0
-\tweak Y-offset #0
-3
a
-\tweak X-offset #-1
-\tweak Y-offset #1
-3
```



Uso del `side-position-interface`

Un objeto que contempla el `side-position-interface` se puede colocar junto a su objeto padre de forma que los bordes especificados de los dos objetos se toquen. El objeto se puede situar encima, debajo, a la derecha o a la izquierda del objeto padre. El padre no se puede especificar; viene determinado por el orden de los elementos en el flujo de entrada. Casi todos los objetos tienen la cabeza de la nota asociada como padre.

Los valores de las propiedades `side-axis` y `direction` determinan dónde colocar el objeto, como sigue:

```
side-axis                direction
```

propiedad	**propiedad**	**colocación**
`0`	`-1`	izquierda
`0`	`1`	derecha
`1`	`-1`	debajo
`1`	`1`	encima

Si `side-axis` es `0`, `X-offset` se debe establecer al procedimiento `ly:side-position-interface::x-aligned-side`. Este procedimiento devuelve el calor correcto de `X-offset` para situar el objeto al lado izquierdo o derecho del padre de acuerdo con el valor de `direction`.

Si `side-axis` es `1`, `Y-offset` se debe establecer al procedimiento `ly:side-position-interface::y-aligned-side`. Este procedimiento devuelve el valor correcto de `Y-offset` para situar el objeto encima o debajo del padre de acuerdo con el valor de `direction`.

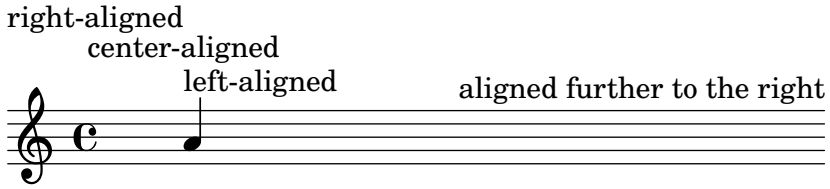
Uso del `self-alignment-interface`

Auto-alineación horizontal de los objetos

La alineación horizontal de un objeto que contempla el interface `self-alignment-interface` está controlada por el valor de la propiedad `self-alignment-X`, siempre y cuando la propiedad `X-offset` de este objeto esté establecida a `ly:self-alignment-interface::x-aligned-on-self`. Se le puede dar a `self-alignment-X` cualquier valor real, en unidades de la mitad de la extensión X total del objeto. Los valores negativos mueven el objeto a la derecha, los positivos hacia la izquierda. Un valor de `0` centra el objeto sobre el punto de referencia de su padre, un valor de `-1` alinea el borde izquierdo del objeto sobre el punto de referencia de su padre, y un valor de `1` alinea el borde derecho del objeto sobre el punto de referencia de su padre. Se pueden usar los símbolos `LEFT`, `CENTER` y `RIGHT` en sustitución de los valores `-1`, `0` y `1`, respectivamente.

Normalmente se usaría la instrucción `\override` para modificar el valor de `self-alignment-X`, pero se puede usar la instrucción `\tweak` para alinear varias anotaciones por separado sobre una sola nota:

```
a'
-\tweak self-alignment-X #-1
^"left-aligned"
-\tweak self-alignment-X #0
^"center-aligned"
-\tweak self-alignment-X #RIGHT
^"right-aligned"
-\tweak self-alignment-X #-2.5
^"aligned further to the right"
```



Auto-alineación vertical de los objetos

Los objetos se pueden alinear verticalmente en una forma análoga a la alineación horizontal si la propiedad `Y-offset` está establecida a `ly:self-alignment-interface::y-aligned-on-self`.
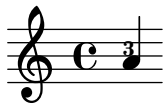
Sin embargo, a menudo se encuentran implicados otros mecanismos en la alineación vertical: el valor de `Y-offset` es tan sólo una variable que se tiene en cuenta. Esto puede hacer que ajustar el valor de ciertos objetos sea una tarea dificultosa. Las unidades son sólo la mitad de las dimensiones verticales del objeto, que suele ser bastante pequeño, por lo que pueden requerirse números bastante grandes. Un valor de `-1` alinea el borde inferior del objeto con el punto de referencia del objeto padre, un valor de `0` alinea el centro del objeto con el punto de referencia del padre, y un valor de `1` alinea el borde superior del objeto con el punto de referencia del padre. Se pueden usar los símbolos `DOWN`, `CENTER`, `UP` en sustitución de `-1`, `0` y `1`, respectivamente.

Auto-alineación de objetos en las dos direcciones

Estableciendo tanto `X-offset` como `Y-offset`, se puede alinear un objeto en las dos direcciones simultáneamente.

El ejemplo siguiente muestra cómo ajustar una digitación de forma que se acerque a la cabeza de la nota.

```
a
-\tweak self-alignment-X #0.5  % move horizontally left
-\tweak Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak self-alignment-Y #-1  % move vertically up
-3  % third finger
```
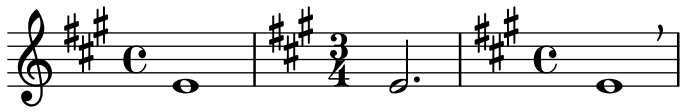


Uso del `break-alignable-interface`

Las letras de ensayo se pueden alinear con objetos de notación distintos a las barras de compás. Estos objetos son `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature` y `time-signature`.
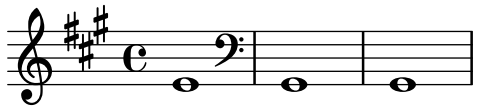
Cada tipo de objeto tiene su propio punto de referencia predeterminado, con el que se alinean las marcas de ensayo:

```
% The rehearsal mark will be aligned to the right edge of the Clef
\override Score.RehearsalMark.break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark ""
e1
% The rehearsal mark will be aligned to the left edge of the Time Signature
\override Score.RehearsalMark.break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark ""
e2.
% The rehearsal mark will be centered above the Breath Mark
\override Score.RehearsalMark.break-align-symbols = #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
e1
\breathe
\mark ""
```



Se puede especificar una lista de posibles objetos para la alineación. Si algunos de los objetos son invisibles en ese punto debido al valor de `break-visibility` o a valores de visibilidad explícitos para las armaduras y las claves, la letra de ensayo o número de compás se alinean con el primer objeto de la lista que sea visible. Si ningún objeto de la lista es visible, el objeto se alinea con la línea divisoria. Si la línea divisoria es invisible, el objeto se alinea con el punto en el que se encontraría la línea divisoria.
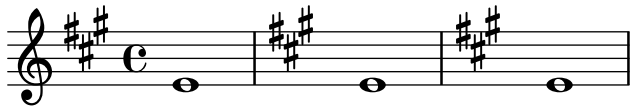
```
% The rehearsal mark will be aligned to the right edge of the Key Signature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark ""
e1
% The rehearsal mark will be aligned to the right edge of the Clef
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef bass
\mark ""
gis,,1
% The rehearsal mark will be centered above the Bar Line
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark ""
e''1
```



La alineación de la letra de ensayo con relación al objeto de notación se puede cambiar, como se ve en el ejemplo siguiente. En una partitura con varios pentagramas, este ajuste se debe hacer para todos los pentagramas.
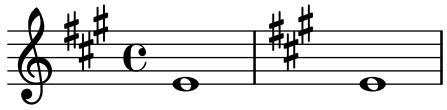
```
% The RehearsalMark will be aligned with the right edge of the Key Signature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark ""
e1
% The RehearsalMark will be centered above the Key Signature
\once \override Score.KeySignature.break-align-anchor-alignment = #CENTER
\mark ""
\key a \major
e1
% The RehearsalMark will be aligned with the left edge of the Key Signature
```

```
\once \override Score.KeySignature.break-align-anchor-alignment = #LEFT
\key a \major
\mark ""
e1
```



La letra de ensayo también se puede desplazar al borde derecho o izquierdo en una medida arbitraria. Las unidades son espacios de pentagrama:

```
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted right by 3.5 staff-spaces
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\once \override Score.KeySignature.break-align-anchor = #3.5
\key a \major
\mark ""
e1
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted left by 2 staff-spaces
\once \override Score.KeySignature.break-align-anchor = #-2
\key a \major
\mark ""
e1
```



5.5.2 Agrupación vertical de objetos gráficos

Los grobs (objetos gráficos) `VerticalAlignment` y `VerticalAxisGroup` trabajan de manera coordinada. `VerticalAxisGroup` agrupa distintos grobs como `Staff`, `Lyrics`, etc. Después, `VerticalAlignment` alinea los distintos grobs agrupados previamente por `VerticalAxisGroup`. Normalmente sólo existe un `VerticalAlignment` por cada partitura, pero cada `Staff`, `Lyrics`, etc. tiene su propio `VerticalAxisGroup`.

5.5.3 Modificación de los sellos

Todos los objetos de presentación tienen una propiedad `stencil` que es parte del `grob-interface`. De forma predeterminada, esta propiedad suele estar establecida a una función específica del objeto que está hecha a medida para disponer el símbolo que lo representa en la salida. Por ejemplo, el ajuste estándar para la propiedad `stencil` del objeto `MultiMeasureRest` es `ly:multi-measure-rest::print`.
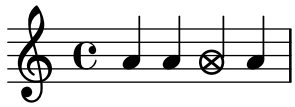
El símbolo estándar de cualquier objeto se puede sustituir modificando la propiedad `stencil` para que haga referencia a un procedimiento diferente escrito especialmente. Esto requiere un alto nivel de conocimiento del funcionamiento interno de LilyPond, pero hay una forma más fácil que a menudo puede dar resultados adecuados.

El procedimiento es establecer la propiedad `stencil` al procedimiento que imprime texto (`ly:text-interface::print`) y añadir una propiedad `text` al objeto ajustada para que contenga el texto de marcado que produce el símbolo requerido. Debido a la flexibilidad del marcado,

se pueden conseguir muchas cosas; en particular, consulte [Notación gráfica dentro de elementos de marcado], página 240.

El ejemplo siguiente muestra esto cambiando el símbolo de la cabeza de la nota a unas aspas dentro de una circunferencia.

```
XinO = {
  \once \override NoteHead.stencil = #ly:text-interface::print
  \once \override NoteHead.text = \markup {
    \combine
      \halign #-0.7 \draw-circle #0.85 #0.2 ##f
      \musicglyph #"noteheads.s2cross"
  }
}
\relative c'' {
  a a \XinO a a
}
```



Cualquiera de los glifos de la fuente tipográfica Feta se puede pasar a la instrucción de marcado `\musicglyph`: véase Sección A.8 [La tipografía Feta], página 650.

Véase también

Referencia de la notación: [Notación gráfica dentro de elementos de marcado], página 240, Sección 1.8.2 [Formatear el texto], página 233, ⟨undefined⟩ [Text markup commands], página ⟨undefined⟩, Sección A.8 [La tipografía Feta], página 650.

5.5.4 Modificación de las formas

Modificación de ligaduras de unión y de expresión

Las ligaduras de unión (`Ties`), de expresión (`Slurs`), de fraseo (`PhrasingSlurs`), de dejar vibrar (`LaissezVibrerTies`) y de repetición (`RepeatTies`) se trazan como curvas de Bézier de tercer orden. Si la forma de la ligadura calculada automáticamente no resulta óptima, se puede modificar su forma manualmente de dos formas:

a. mediante la especificación de los desplazamientos que se quieren realizar sobre los puntos de control de la curva de Bézier calculada automáticamente, o bien
b. mediante la especificación explícita de las posiciones de los cuatro puntos de control que se requieren para definir la curva deseada.

A continuación se explican ambos métodos. El primer método es más adecuado cuando solo se requieren ligeros ajustes de la curva; el segundo puede ser mejor para crear curvas que tienen relación con una única nota.

Curvas de Bézier cúbicas
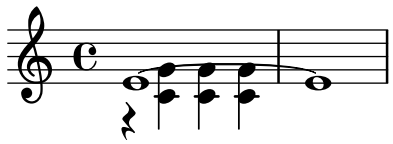
Las curvas Bézier de tercer orden o cúbicas están definidas por cuatro puntos de control. El primer y cuarto puntos de control son exactamente los puntos extremos de comienzo y de final de la curva. Los dos puntos de control intermedios definen la forma. Se pueden encontrar en la web animaciones que muestran cómo se traza la curva, pero la descripción siguiente puede ser de ayuda. La curva comienza a partir del primer punto de control dirigiéndose directamente hacia el segundo, curvándose progresivamente para dirigirse hacia el tercero y continuando la

curva hacia el cuarto, llegando a éste en viaje directo desde el tercer punto de control. La curva está contenida enteramente dentro del cuadrilátero definido por los cuatro puntos de control. Todas las traslaciones, rotaciones y escalado de los puntos de control producen exactamente las mismas operaciones sobre la curva.

Especificar desplazamientos a partir de los puntos de control actuales

En este ejemplo, la colocación automática de la ligadura no resulta óptima, y `\tieDown` no es la solución.

```
<<
  { e1~ e }
\\
  { r4 <g c,> <g c,> <g c,> }
>>
```



El ajuste de los puntos de control de la ligadura mediante la instrucción `\shape` permite evitar las colisiones.

La sintaxis de `\shape` es:

`[-]\shape` *`desplazamientos elemento`*
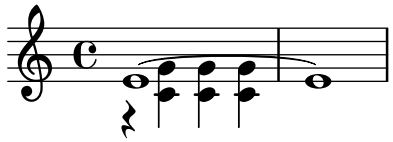
Esta instrucción recoloca los puntos de control de *elemento* en las cantidades dadas por *desplazamientos*. El argumento *desplazamientos* es una lista de parejas de números o una lista de tales listas. Cada elemento de una pareja representa el desplazamiento de una de las coordenadas de un punto de control. Si el *elemento* es una cadena de caracteres, el resultado es `\once\override` para el tipo de grob especificado. Si *elemento* es una expresión musical, el resultado es la misma expresión musical con la consiguiente modificación aplicada.

En otras palabras, la función `\shape` puede actuar como una instrucción `\once\override` o como una instrucción `\tweak` dependiendo de si el argumento *elemento* es el nombre de un grob, como "Slur", o una expresión musical, como "(". El argumento *desplazamientos* especifica los desplazamientos de los cuatro puntos de control como una lista de cuatro parejas de valores (dx . dy) en unidades de espacios de pentagrama (o una lista de tales listas si la curva tiene más de un segmento).

El guión precedente se exige si, y sólo si, se usa la forma `\tweak`.
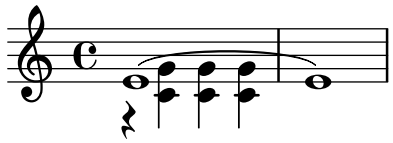
Así, usando el mismo ejemplo de arriba y la forma `\once\override` de la instrucción `\shape`, lo siguiente tiene el efecto de elevar la ligadura de unión en la mitad de un espacio de pentagrama:

```
<<
  {
    \shape #'((0 . 0.5) (0 . 0.5) (0 . 0.5) (0 . 0.5)) Tie
    e1~ e
  }
\\
  { r4 <g c,> <g c,> <g c,> }
>>
```
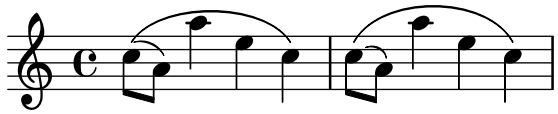


Este posicionamiento de la ligadura de unión es mejor, pero quizá debería elevarse más en la parte central. El ejemplo siguiente hace esto, esta vez usando la forma alternativa `\tweak`:

```
<<
  {
    e1-\shape #'((0 . 0.5) (0 . 1) (0 . 1) (0 . 0.5)) ~ e
  }
\\
  { r4 <g c,> <g c,> <g c,> }
>>
```
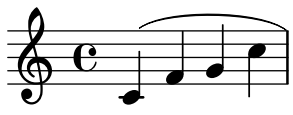


Los cambios en las posiciones horizontales de los puntos de control se pueden hacer de la misma forma, y dos curvas distintas que dan comienzo en el mismo momento musical se pueden conformar también:
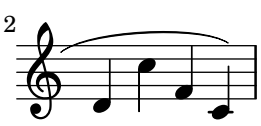
```
c8(\( a) a'4 e c\)
\shape #'((0.7 . -0.4) (0.5 . -0.4) (0.3 . -0.3) (0 . -0.2)) Slur
\shape #'((0 . 0) (0 . 0.5) (0 . 0.5) (0 . 0)) PhrasingSlur
c8(\( a) a'4 e c\)
```



La función `\shape` también puede desplazar los puntos de control de las curvas que se extienden atravesando saltos de línea. Cada pieza de la curva dividida puede recibir su propia lista de desplazamientos. Si no se necesita hacer cambios a uno de los segmentos en particular, se puede usar una lista vacía como contenedor. En este ejemplo, el salto de línea hace que la única ligadura de expresión aparezca como si fueran dos:
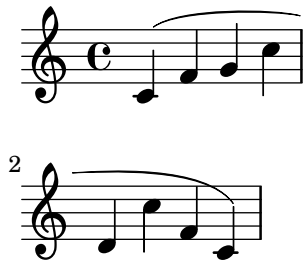
```
c4( f g c
\break
d,4 c' f, c)
```
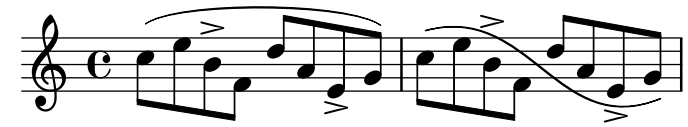




Los cambios en la forma de las dos mitades de la ligadura de expresión deja más claro que la ligadura continúa más allá del salto de línea:

```
% () may be used as a shorthand for ((0 . 0) (0 . 0) (0 . 0) (0 . 0))
% if any of the segments does not need to be changed
\shape #'(
           (( 0 . 0) (0 . 0) (0 . 0) (0 . 1))
           ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
         ) Slur
c4( f g c
\break
d,4 c' f, c)
```



En una curva en forma de «S» siempre es necesario ajustar manualmente los puntos de control: LilyPond nunca escoge tales formas automáticamente.

```
c8( e b-> f d' a e-> g)
\shape #'((0 . -1) (5.5 . -0.5) (-5.5 . -10.5) (0 . -5.5)) PhrasingSlur
c8\( e b-> f d' a e-> g\)
```



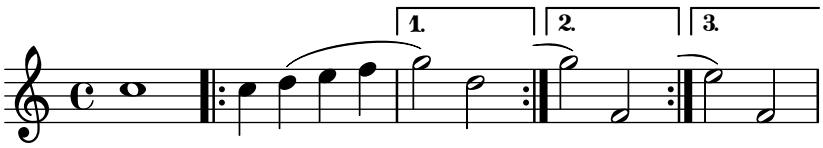
Especificar los puntos de control explícitamente

Las coordenadas de los puntos de contro de Bézier se especifican en unidades de espacios de pentagrama. La coordenada X es relativa al punto de referencia al que se adjunta la ligadura, y la coordenada Y es relativa a la línea central del pentagrama. Las coordenadas se especifican como una lista de cuatro parejas de números decimales (reales). Un enfoque consiste en estimar las coordenadas de los dos extremos, y después adivinar los dos puntos intermedios. Los valores óptimos se encuentran después por ensayo y error. Tenga en cuenta que estos valores podrían requerir un ajuste manual si se hace posteriormente cualquier cambio manual a la música o a la disposición.

Una situación en la que es preferible especificar los puntos de control explícitamente a especificar los desplazamientos es cuando se necesita especificarlos de forma relativa a una nota única. A continuación presentamos un ejemplo de esto. Muestra una forma de indicar una ligadura que se extiende hacia el interior de las casillas de primera y segunda vez de una repetición.

```
c1
\repeat volta 3 { c4 d( e f }
\alternative {
  { g2) d }
  {
    g2
    % create a slur and move it to a new position
    % the <> is just an empty chord to carry the slur termination
    -\tweak control-points #'((-2 . 3.8) (-1 . 3.9) (0 . 4) (1 . 3.4)) ( <> )
```

```
    f,
  }
  {
    e'2
    % create a slur and move it to a new position
    -\tweak control-points #'((-2 . 3) (-1 . 3.1) (0 . 3.2) (1 . 2.4)) ( <> )
    f,
  }
}
```



Advertencias y problemas conocidos

No es posible modificar la forma de las ligaduras de unión o de expresión cambiando la propiedad `control-points` si hay más de una en el mismo momento musical, ni siquiera usando la instrucción `\tweak`. Sin embargo, se puede sobreescribir la propiedad `tie-configuration` de `TieColumn` para fijar la línea de inicio y la dirección según se requiera.

Véase también
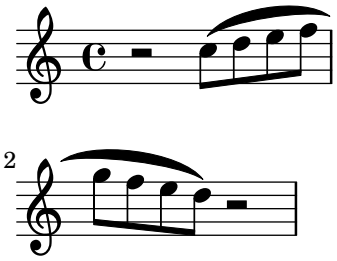
Referencia de funcionamiento interno: Sección “TieColumn” in *Referencia de Funcionamiento Interno*.

5.5.5 Modificación de objetos de extensión divididos

Uso de `\alterBroken`

Cuando un “spanner” u objeto extenso atraviesa uno o varios saltos de línea, cada fragmento hereda los atributos del objeto de extensión original. Así, el trucado ordinario de un objeto dividido aplica las mismas modificaciones a todos y cada uno de sus segmentos. En el ejemplo que aparece a continuación, la sobreescritura de `thickness` (grosor) afecta a la ligadura a ambos lados del salto de línea.

```
r2
\once\override Slur.thickness = 10
c8( d e f
\break
g8 f e d) r2
```



Es posible modificar de forma independiente el aspecto de los fragmentos individuales de un objeto de extensión dividido, con la instrucción `\alterBroken`. Esta instrucción puede producir un `\override` o bien un `\tweak` de una propiedad del objeto de extensión.

La sintaxis de `\alterBroken` es

`[-]\alterBroken` *`propiedad valores elemento`*
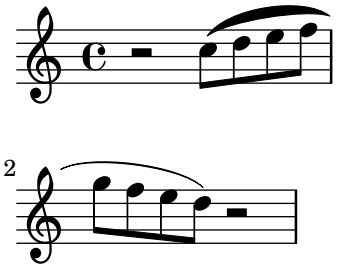
El argumento *valores* es una lista de valores, uno por cada fragmento. Si *elemento* el el nombre de un grob como `Slur` o `Staff.PianoPedalBracket`, el resultado es una instrucción de sobreescritura `\override` del tipo de grob especificado. Si *elemento* es una expresión musical como "(" o "[" el resultado es la misma expresión musical con el correspondiente truco aplicado.

El guión inicial se debe utilizar con la forma de trucaje, `\tweak`. No lo escriba cuando utilice `\alterBroken` como una operación de sobreescritura, `\override`.

En su uso como `\override`, `\alterBroken` puede ir precedido de `\once` o `\temporary` y ser revertido con `\revert` con el argumento *propiedad*.

EL código siguiente aplica un `\override` independiente a cada uno de los fragmentos de la ligadura del ejemplo anterior:

```
r2
\alterBroken thickness #'(10 1) Slur
c8( d e f
\break
g8 f e d) r2
```



La instrucción `\alterBroken` se puede usar con cualquier objeto de extensión, incluidos `Tie`, `PhrasingSlur`, `Beam` y `TextSpanner`. Por ejemplo, un editor que está preparando una edición académica podría desear indicar la ausencia de parte de una ligadura de fraseo en una fuente haciendo discontinuo solamente el segmento que se ha añadido. El siguiente ejemplo ilustra de qué manera se puede hacer esto, en este caso usando la forma `\tweak` de la instrucción:

```
% The empty list is conveniently used below, because it is the
% default setting of dash-definition, resulting in a solid curve.
c2-\alterBroken dash-definition #'(() ((0 1.0 0.4 0.75))) \(e
\break
g2 e\)
```





Es importante comprender que `\alterBroken` fija cada pieza del objeto fragmentado al valor correspondiente dentro del argumento *valores*. Cuando hay menos valores que piezas, cualquier fragmento adicional recibe la asignación de la lista vacía. Esto puede conllevar resultados no deseados si la propiedad de disposición no está establecida a la lista vacía de forma predeterminada. En estos casos, se debe asignar a cada segmento un valor adecuado.

Advertencias y problemas conocidos

Los saltos de línea pueden ocurrir en distintos lugares según cambia la disposición de la música. Los ajustes elegidos para `\alterBroken` podrían no ser adecuados para un objeto de extensión que ya no está dividido o que se fragmenta en más piezas que antes. El uso explícito (manual) de saltos de línea mediante `\break` puede prevenir situaciones como ésta.

La instrucción `\alterBroken` no es efectiva para propiedades de los objetos extensos accedidas antes del salto de línea, como `direction`.

Véase también

Extender LilyPond: Sección “Trucos difíciles” in *Extender*.

5.5.6 Contenedores unpure-pure

Los contenedores ‘unpure-pure’ (pura y no pura) son útiles para la sobreescritura de los cálculos del espaciado en el *eje Y* (concretamente `Y-offset` e `Y-extent`) con una función de Scheme en lugar de un literal, es decir, un número o una pareja de números.

Para ciertos objetos gráficos, las dimensiones `Y-extent` están basadas en la propiedad `stencil`, la sobreescritura de la propiedad de sello de éstos requiere una sobreescritura adicional de `Y-extent` con un contenedor unpure-pure. Cuando una función sobreescribe una dimensión `Y-offset` y/o `Y-extent` se supone que esto dispara los cálculos de los saltos de línea prematuramente durante la compilación. Así pues, la función no se evalúa en absoluto (devolviendo por lo general un valor de ‘`0`’ o ‘`'(0 . 0)`’) lo que puede dar lugar a colisiones. Una función ‘pura’ no afecta a las propiedades, objetos o suicidios de grobs, y por ello siempre ven sus valores relacionados con el eje Y evaluados correctamente.

Actualmente hay unas treinta funciones que ya se consideran ‘puras’ y los contenedores Unpure-pure son una manera de establecer funciones que no están en esta lista como ‘puras’. La función ‘pura’ se evalúa *antes* de cualquier salto de línea y así el espaciado horizontal se puede ajustar ‘a tiempo’. La función ‘impura’ se evalúa entonces *después* del salto de línea.

Nota: Dado que es difícil saber siempre qué funciones están en esta lista, recomendamos que cualquier función ‘pura’ que estemos creando no utilice los grobs `Beam` o `VerticalAlignment`.

Un contenedor ‘unpure-pure’ se contruye de la manera siguiente:

```
(ly:make-unpure-pure-container f0 f1)
```

donde `f0` es una función que toma *n* argumentos (*n >= 1*) y el primer argumento siempre debe ser el grob. Ésta es la función que da el resultado real. *f1* es la función que se etiqueta como ‘pura’ que toma *n + 2* argumentos. De nuevo, el primer argumento debe ser siempre el grob pero los argumentos segundo y tercero son argumentos de ‘inicio’ y de ‘final’.

inicio y *final* son, a todos los efectos, valores mudos que sólo tienen importancia para los `objetos de extensión` (o sea: `Hairpin`, regulador, o `Beam`, barra), que pueden devolver distintas estimaciones de altura basadas en una columna de inicio y una de final.

El resto son los otros argumentos para la primera función (que puede no ser ninguno si *n = 1*).

El resultado de la segunda función se usa como una aproximación del valor necesario, que se usa entonces por la primera función para obtener el valor real que se usa a continuación para el ajuste de precisión mucho más tardío durante el proceso de espaciado.
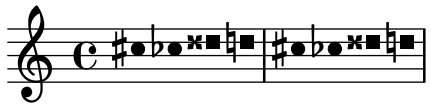
```
#(define (square-line-circle-space grob)
(let* ((pitch (ly:event-property (ly:grob-property grob 'cause) 'pitch))
       (notename (ly:pitch-notename pitch)))
 (if (= 0 (modulo notename 2))
```

```
      (make-circle-stencil 0.5 0.0 #t)
      (make-filled-box-stencil '(0 . 1.0)
                               '(-0.5 . 0.5)))))

squareLineCircleSpace = {
  \override NoteHead.stencil = #square-line-circle-space
}

smartSquareLineCircleSpace = {
  \squareLineCircleSpace
  \override NoteHead.Y-extent =
   #(ly:make-unpure-pure-container
      ly:grob::stencil-height
      (lambda (grob start end) (ly:grob::stencil-height grob)))
}

\new Voice \with { \remove "Stem_engraver" }
\relative c'' {
  \squareLineCircleSpace
  cis4 ces disis d
  \smartSquareLineCircleSpace
  cis4 ces disis d
}
```



En el primer compás, sin el contenedor *unpure-pure*, el motor de espaciado no conoce la anchura de la cabeza de la nota y permite que colisione con las alteraciones accidentales. En el segundo compás, el motor de espaciado conoce la anchura de las cabezas de las notas y evita la colisión mediante el alargamiento de la línea en la medida adecuada.

Normalmente, para cálculos simples se pueden usar funciones casi idénticas tanto para las partes 'no pura' y 'pura', simplemente cambiando el número de argumentos que se pasan a, y el ámbito de, la función.

Nota: Si una función está caracterizada como 'pura' y resulta que no lo es, el resultado puede ser inesperado.

5.6 Uso de las funciones musicales

Dondequiera que se necesite reutilizar trucos con distintas expresiones musicales, con frecuencia conviene hacer que el truco forme parte de una *función musical*. En esta sección estudiaremos solamente las funciones de *sustitución*, en las que el objeto es sustituir una variable dentro de un fragmento de código de entrada de LilyPond. Se describen otras funciones más complejas en Sección "Funciones musicales" in *Extender*.

5.6.1 Sintaxis de las funciones de sustitución

Es fácil hacer una función que sustituya una variable dentro de código de LilyPond. La forma general de estas funciones es

```
funcion =
```

```
#(define-music-function
     (parser location arg1 arg2 ...)
     (type1? type2? ...)
   #{
     ...música...
   #})
```

donde

argN	*n*-ésimo argumento
typeN?	*predicado de tipo* de Scheme para el que *argN* debe devolver `#t`.
...música...	código de entrada normal de LilyPond, que utiliza `$` (en los lugares en que sólo se permiten construcciones de Lilypond) o `#` (para usarlo como un valor de Scheme o un argumento de función musical o música dentro de listas de música) para referenciar argumentos (p.ej. '`#arg1`').

Los argumentos `parser` y `location` son necesarios, y se utilizan en algunas situaciones avanzadas como se encuentra descrito en Sección "Sintaxis de las funciones musicales" in *Extender*. Para las funciones de sustitución, tan sólo hemos de asegurarnos de incluirlos.

También es necesaria la lista de predicados de tipo. Algunos de los predicados de tipo más comunes que se utilizan en las funciones musicales son:

```
boolean?
cheap-list?  (utilizar en lugar de 'list?'
 para un procesado más rápido)
ly:duration?
ly:music?
ly:pitch?
markup?
number?
pair?
string?
symbol?
```

Para ver una lista de los predicados de tipo disponibles, consulte Sección A.20 [Predicados de tipo predefinidos], página 771. También se permiten los predicados de tipo definidos por el usuario.

Véase también

Referencia de la notación: Sección A.20 [Predicados de tipo predefinidos], página 771.

Expansión de LilyPond: Sección "Sintaxis de las funciones musicales" in *Extender*.

Archivos instalados: '`lily/music-scheme.cc`', '`scm/c++.scm`', '`scm/lily.scm`'.

5.6.2 Ejemplos de funciones de sustitución
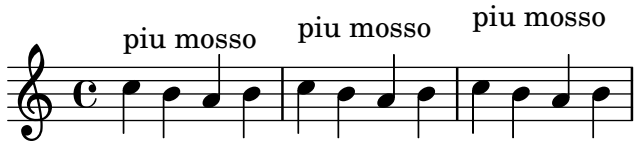
Esta sección presenta algunos ejemplos de funciones de sustitución. No pretenden ser exhaustivas, sino demostrar algunas de las posibilidades de las funciones de sustitución sencillas.

En el primer ejemplo se define una función que simplifica el ajuste del relleno de un elemento de inscripción de texto TextScript:

```
padText =
#(define-music-function
     (parser location padding)
```

```
     (number?)
   #{
     \once \override TextScript.padding = #padding
   #})
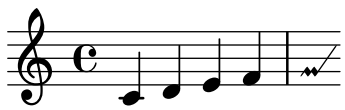
\relative c'' {
  c4^"piu mosso" b a b
  \padText #1.8
  c4^"piu mosso" b a b
  \padText #2.6
  c4^"piu mosso" b a b
}
```



Además de números, podemos usar expresiones musicales, como notas, para los argumentos de las funciones musicales:

```
custosNote =
#(define-music-function
     (parser location note)
     (ly:music?)
   #{
     \tweak NoteHead.stencil #ly:text-interface::print
     \tweak NoteHead.text
        \markup \musicglyph #"custodes.mensural.u0"
     \tweak Stem.stencil ##f
     #note
   #})

\relative c' { c4 d e f \custosNote g }
```

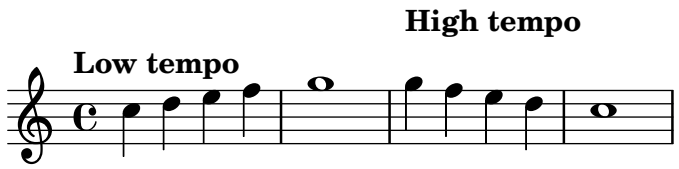


Se pueden definir funciones de sustitución con más de un argumento:

```
tempoPadded =
#(define-music-function
     (parser location padding tempotext)
     (number? markup?)
   #{
     \once \override Score.MetronomeMark.padding = #padding
     \tempo \markup { \bold #tempotext }
   #})

\relative c'' {
  \tempo \markup { "Low tempo" }
  c4 d e f g1
  \tempoPadded #4.0 "High tempo"
```

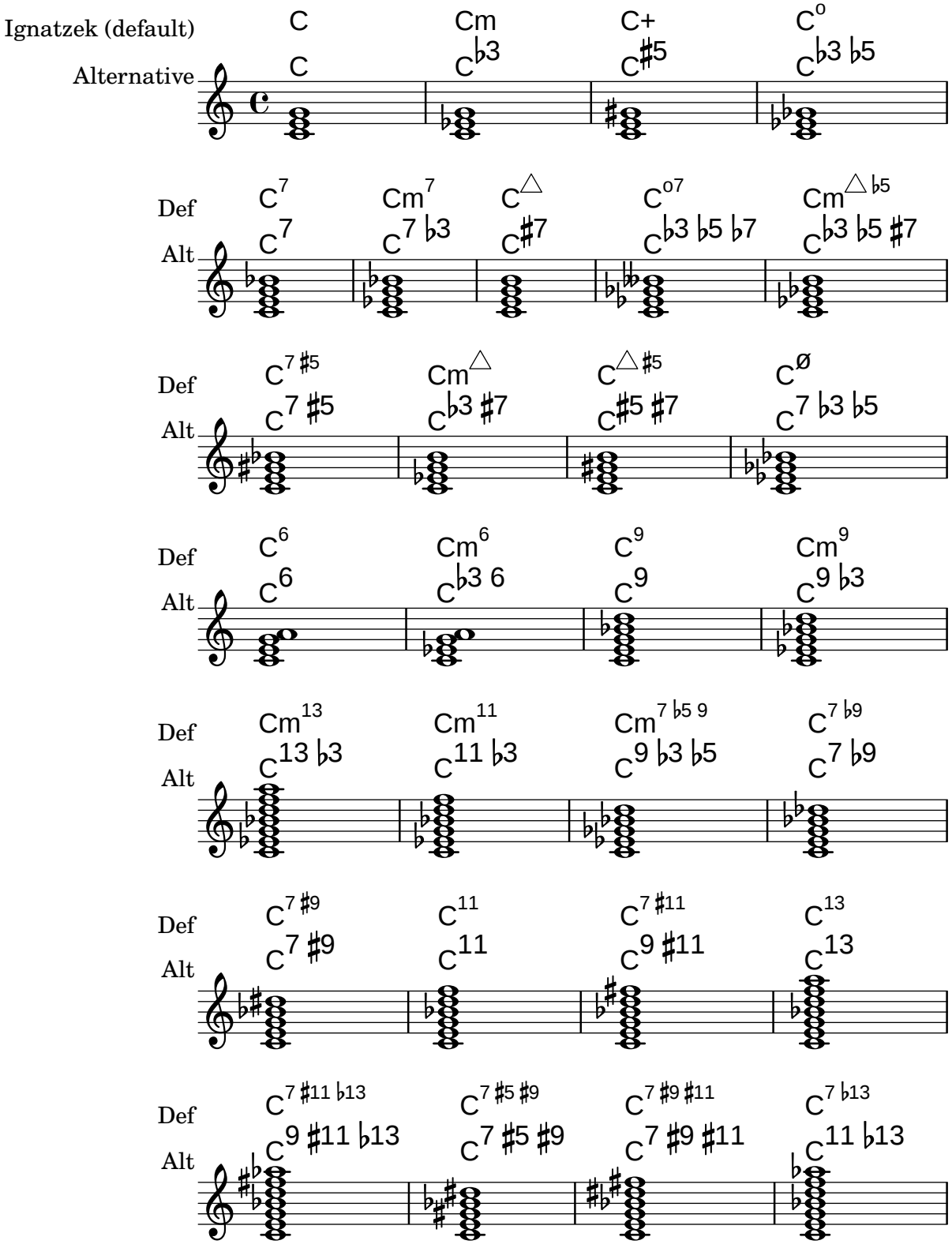
```
  g4 f e d c1
}
```

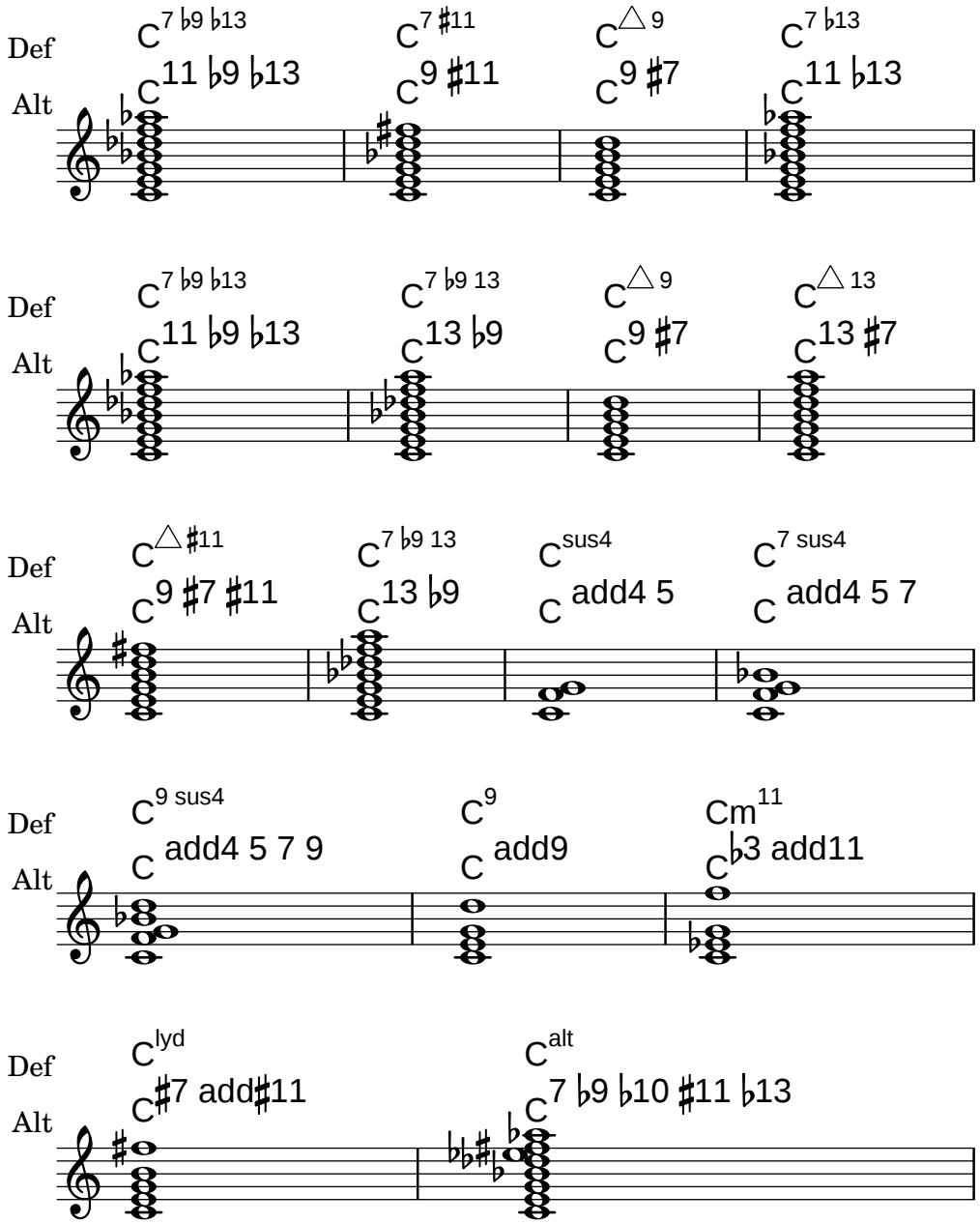


Apéndice A Tablas del manual sobre notación

A.1 Cuadro de nombres de acordes
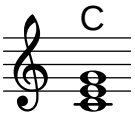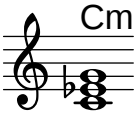
La siguiente tabla muestra dos sistemas estándar para imprimir los nombres de los acordes, junto a las notas que representan.
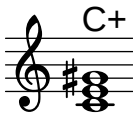
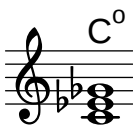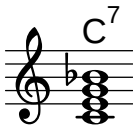
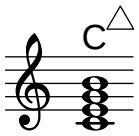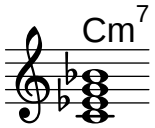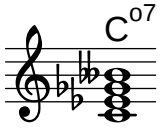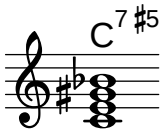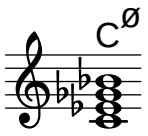
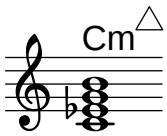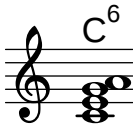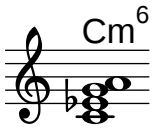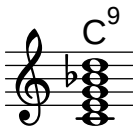


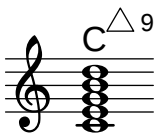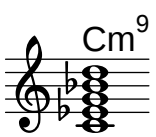
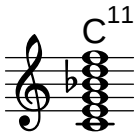
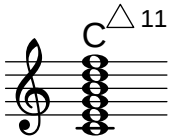
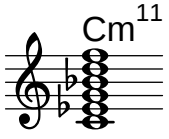


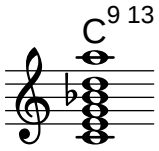
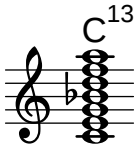
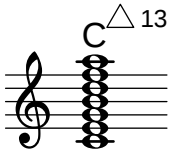
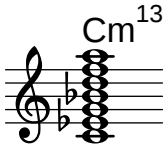
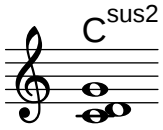
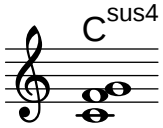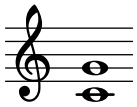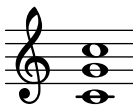
A.2 Modificadores de acorde más usuales

La tabla siguiente presenta los modificadores de acorde que se pueden usar para generar las estructuras de acordes estándar.

Tipo	**Intervalo**	**Modificador**	**Ejemplo**	**Salida**
Mayor	Tercera mayor y quinta justa	`5` o nada	`c1:5`	
Menor	Tercera menor y quinta justa	`m` o `m5`	`c1:m`	

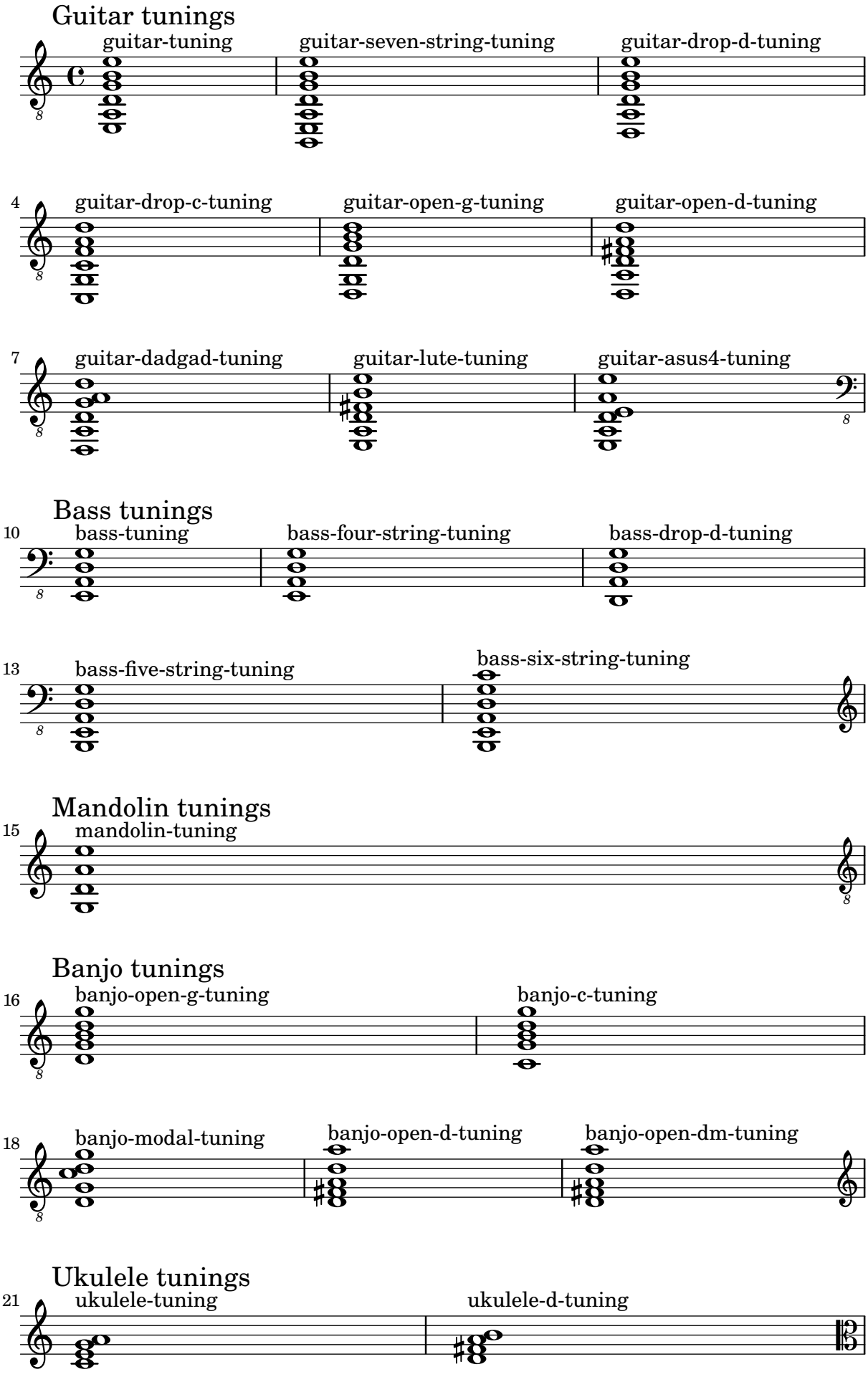
Aumentado	Tercera mayor y quinta aumentada	`aug`	`c1:aug`	
Disminuido	Tercera menor y quinta disminuida	`dim`	`c1:dim`	
Séptima dominante	Tríada mayor con séptima menor	`7`	`c1:7`	
Séptima mayor	Tríada mayor y séptima mayor	`maj7` ó `maj`	`c1:maj7`	
Séptima menor	Tríada menor y séptima menor	`m7`	`c1:m7`	
Séptima disminuida	Tríada disminuida y séptima disminuida	`dim7`	`c1:dim7`	
Aumentado con séptima	Tríada aumentada y séptima menor	`aug7`	`c1:aug`	
Séptima semi-disminuido	Tríada disminuida y séptima menor	`m7.5-`	`c1:m7.5-`	
Menor con séptima mayor	Tríada menor con séptima mayor	`m7+`	`m7+`	

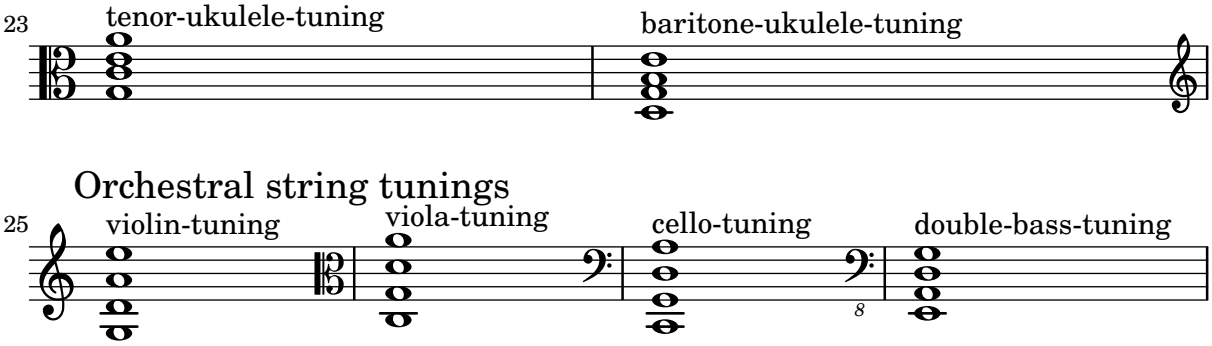
Mayor con sexta	Tríada mayor y sexta	`6`	`c1:6`	
Menor con sexta	Tríada menor y sexta	`m6`	`c1:m6`	
Novena dominante	Séptima dominante con novena mayor	`9`	`c1:9`	
Mayor con novena	Séptima mayor y novena mayor	`maj9`	`c1:maj9`	
Menor con novena	Séptima menor y novena mayor	`m9`	`c1:m9`	
Oncena dominante	Novena dominante con oncena justa	`11`	`c1:11`	
Oncena mayor	Novena mayor y oncena justa	`maj11`	`c1:maj11`	
Menor con oncena	Novena menor y oncena justa	`m11`	`c1:m11`	

Trecena dominante	Novena dominante y trecena mayor	`13`	`c1:13`	
Trecena dominante	Oncena dominante y trecena mayor	`13.11`	`c1:13.11`	
Mayor con trecena	Oncena mayor y trecena mayor	`maj13.11`	`c1:maj13.11`	
Menor con trecena	Oncena menor y trecena mayor	`m13.11`	`c1:m13.11`	
Segunda suspendida	Segunda mayor y quinta justa	`sus2`	`c1:sus2`	
Cuarta suspendida	Cuarta justa y quinta justa	`sus4`	`c1:sus4`	
Acorde de potencia o de quinta vacía (de dos notas)	Quinta justa	`1.5`	`\powerChords c1:1.5`	
Acorde de potencia o de quinta vacía (de tres notas)	Quinta justa, octava	`1.5.8`	`\powerChords c1:1.5.8`	

A.3 Afinaciones de cuerdas predefinidas

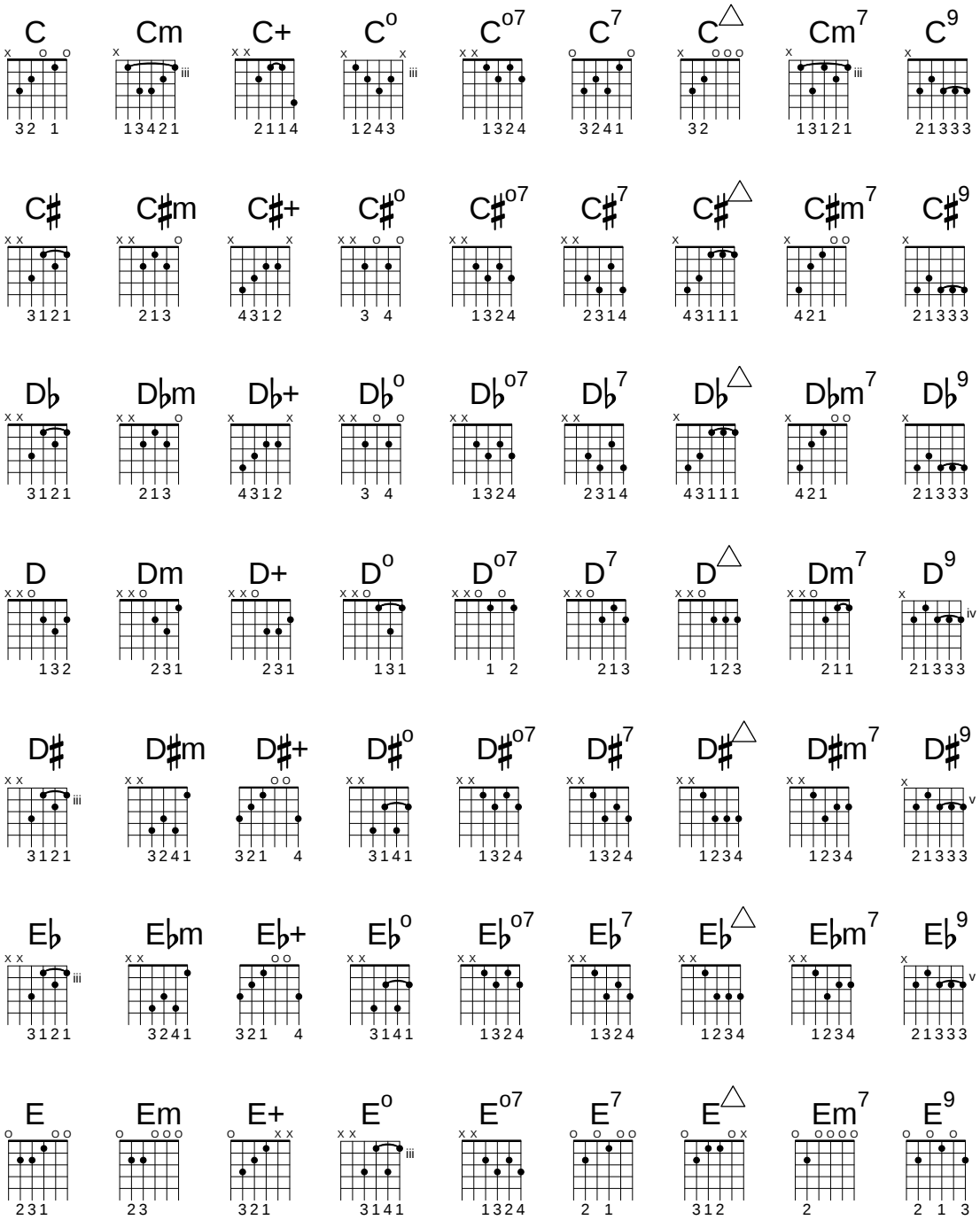
El cuadro de abajo muestra las afinaciones predefinidas de cuerdas.

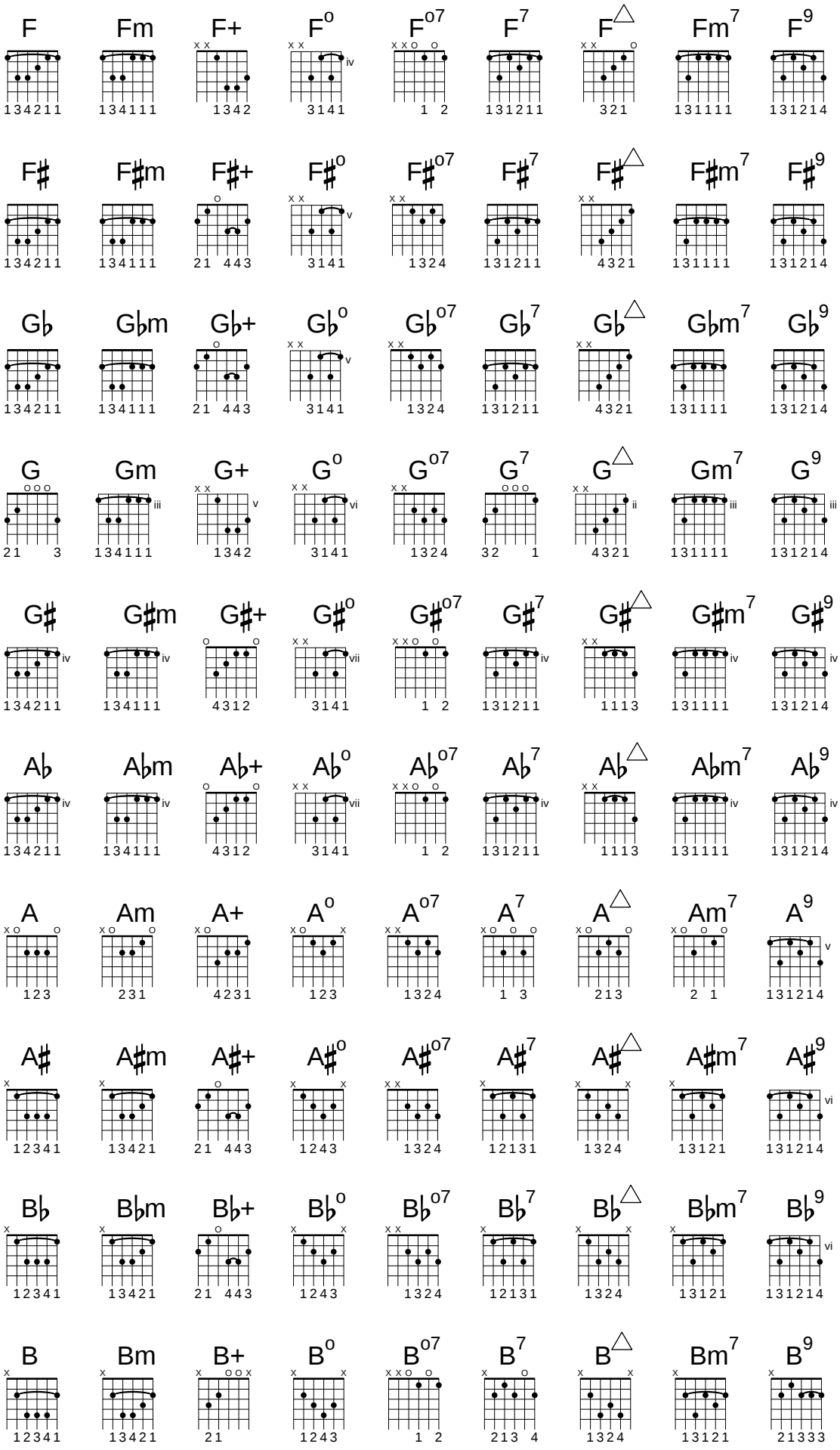

Guitar tunings
guitar-tuning
guitar-seven-string-tuning
guitar-drop-d-tuning
4
guitar-drop-c-tuning
guitar-open-g-tuning
guitar-open-d-tuning
7
guitar-dadgad-tuning
guitar-lute-tuning
guitar-asus4-tuning
Bass tunings
10
bass-tuning
bass-four-string-tuning
bass-drop-d-tuning
13
bass-five-string-tuning
bass-six-string-tuning
Mandolin tunings
15
mandolin-tuning
Banjo tunings
16
banjo-open-g-tuning
banjo-c-tuning
18
banjo-modal-tuning
banjo-open-d-tuning
banjo-open-dm-tuning
Ukulele tunings
21
ukulele-tuning
ukulele-d-tuning



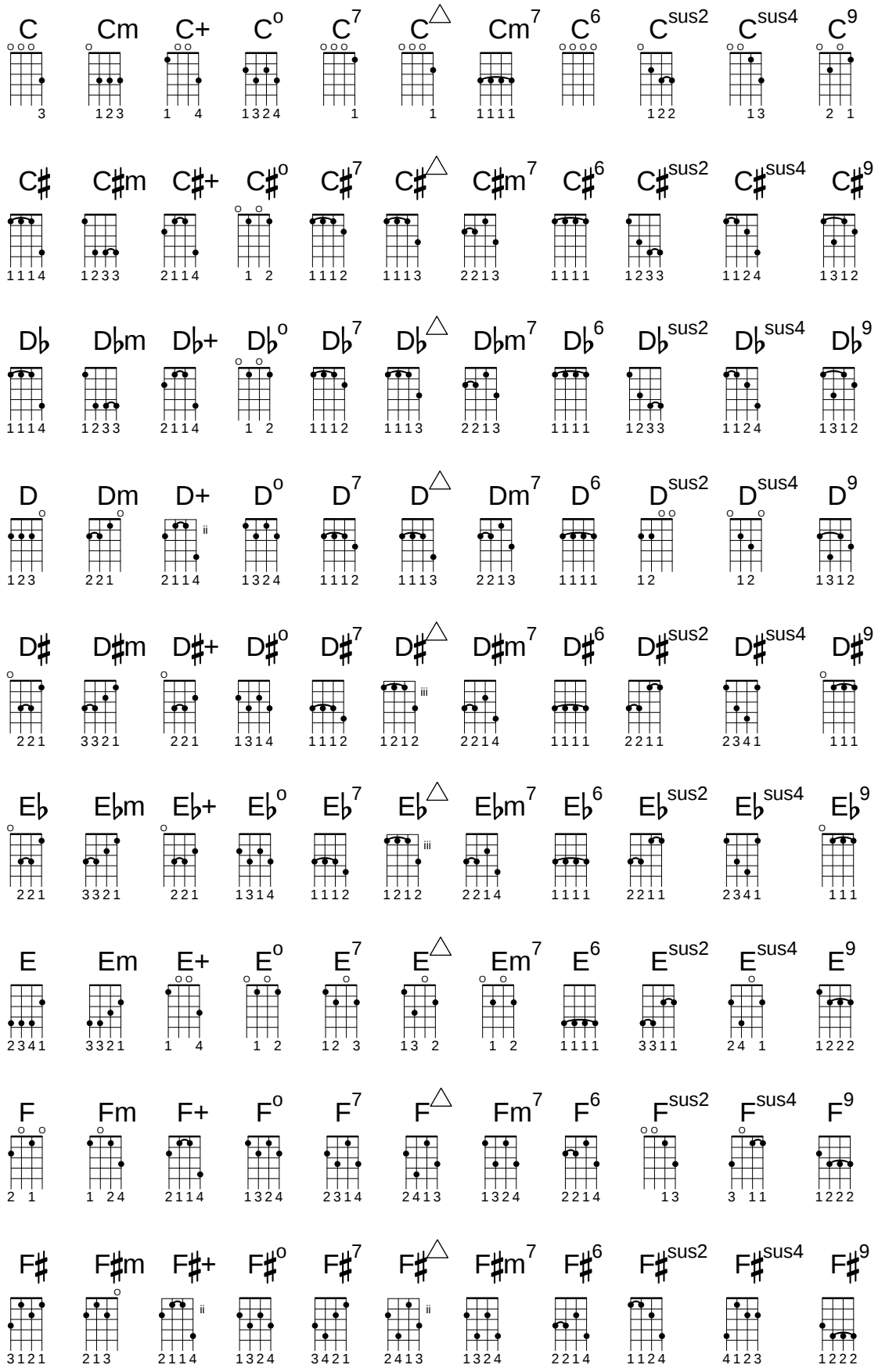
A.4 Diagramas predefinidos de posiciones

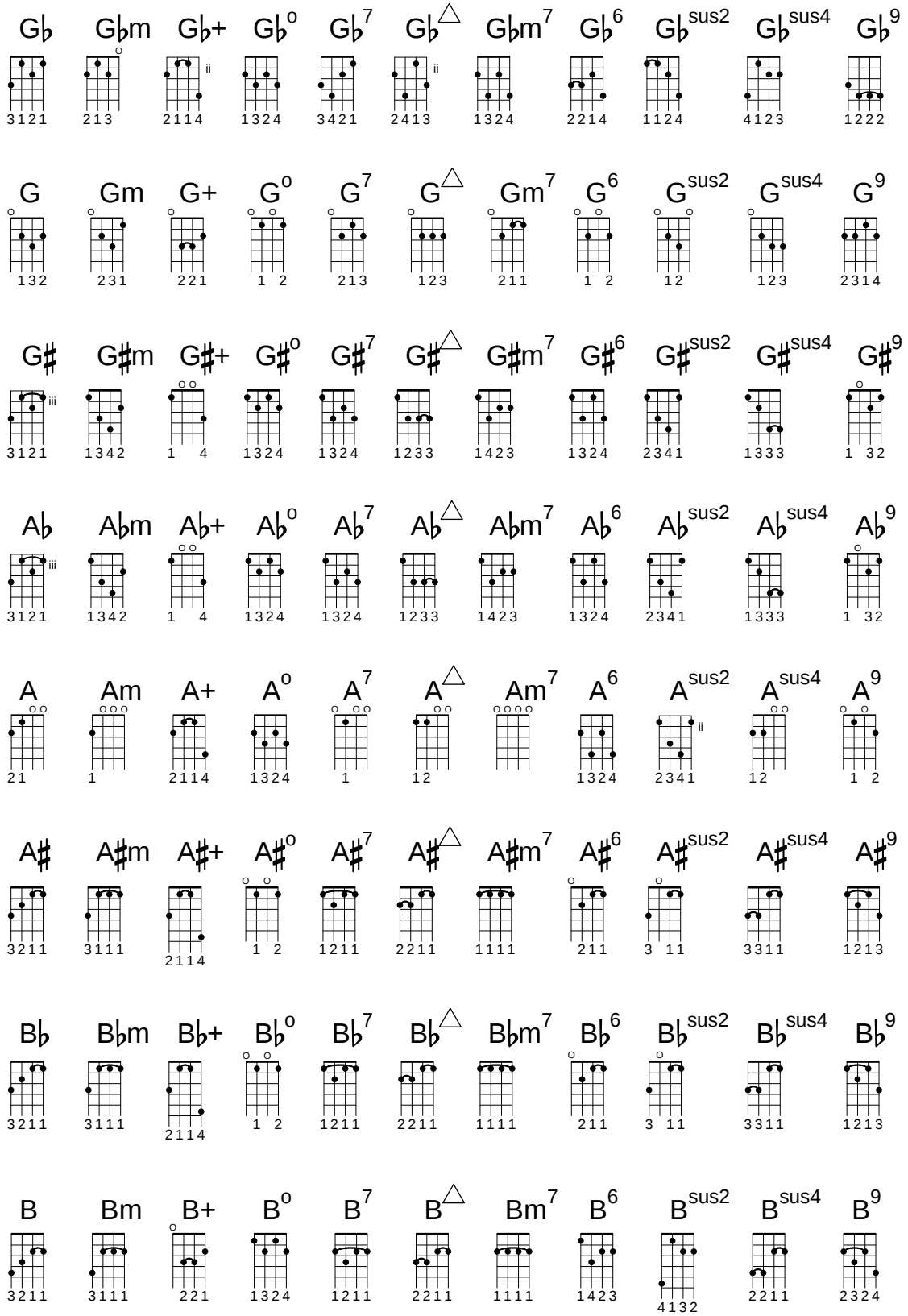
Diagramas para guitarra



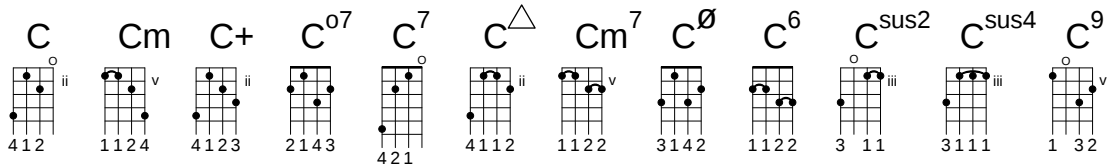

F Fm F+ F° F°7 F7 F△ Fm7 F9
F♯ F♯m F♯+ F♯° F♯°7 F♯7 F♯△ F♯m7 F♯9
G♭ G♭m G♭+ G♭° G♭°7 G♭7 G♭△ G♭m7 G♭9
G Gm G+ G° G°7 G7 G△ Gm7 G9
G♯ G♯m G♯+ G♯° G♯°7 G♯7 G♯△ G♯m7 G♯9
A♭ A♭m A♭+ A♭° A♭°7 A♭7 A♭△ A♭m7 A♭9
A Am A+ A° A°7 A7 A△ Am7 A9
A♯ A♯m A♯+ A♯° A♯°7 A♯7 A♯△ A♯m7 A♯9
B♭ B♭m B♭+ B♭° B♭°7 B♭7 B♭△ B♭m7 B♭9
B Bm B+ B° B°7 B7 B△ Bm7 B9

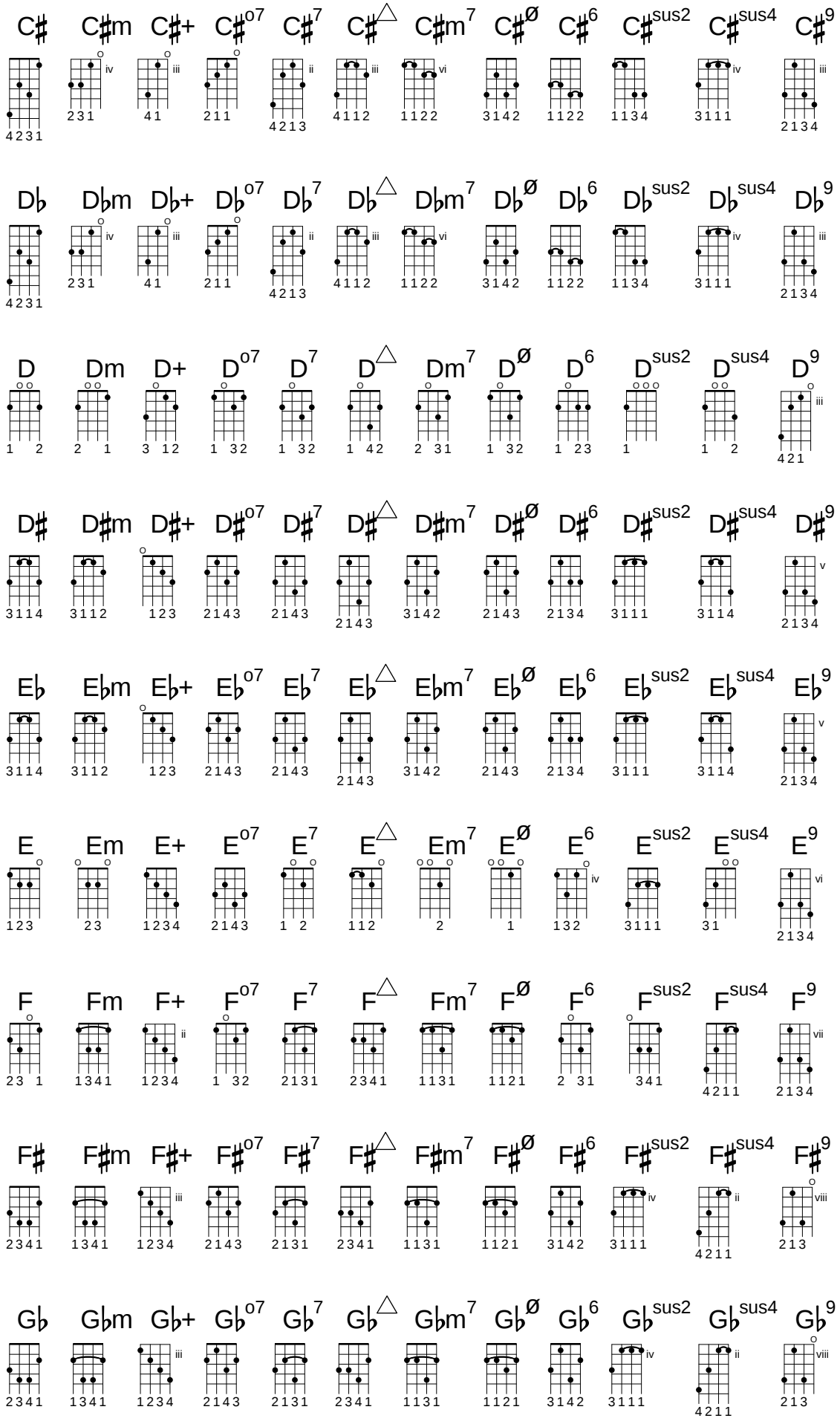
Diagramas para ukelele

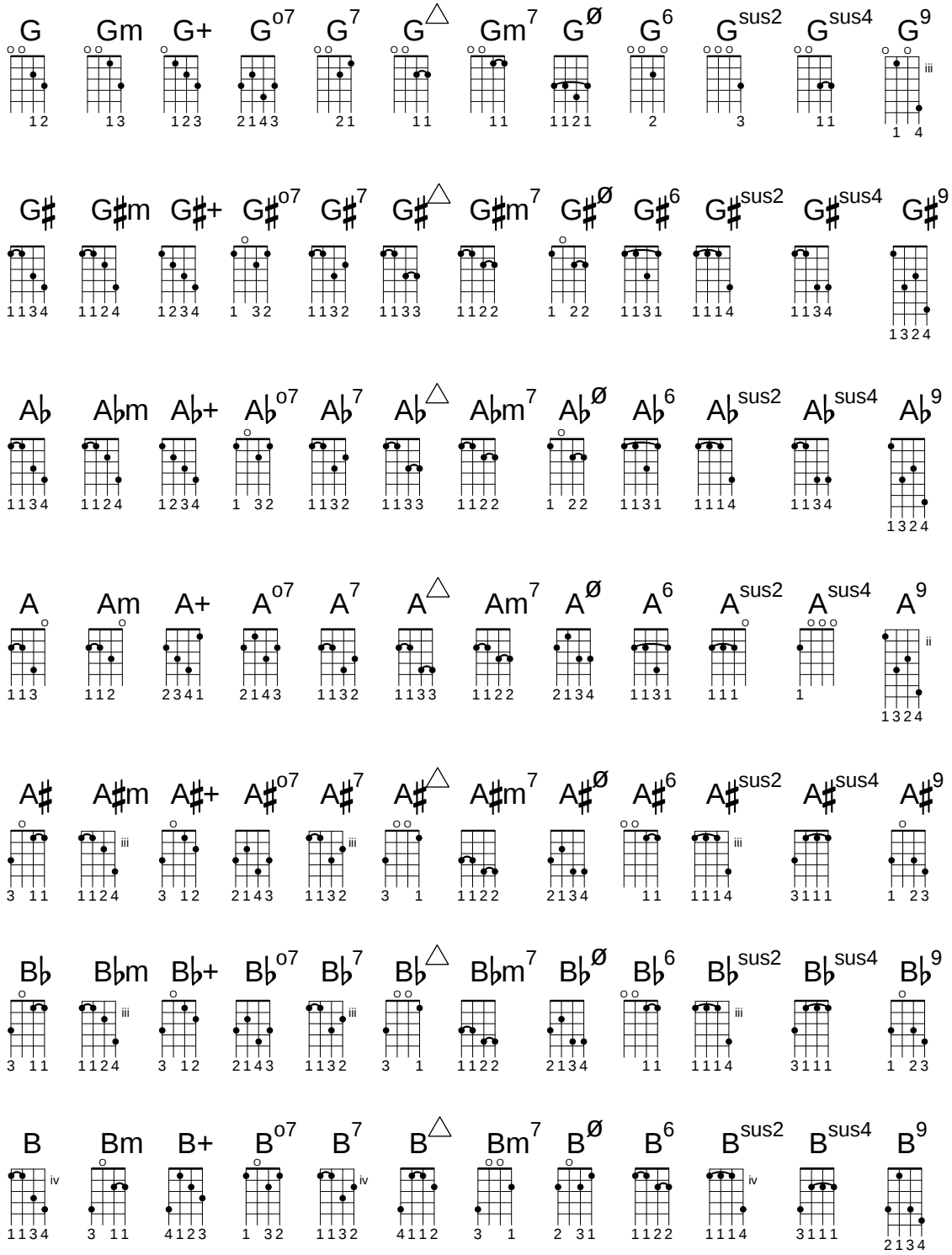




Diagramas para mandolina




C♯ C♯m C♯+ C♯o7 C♯7 C♯△ C♯m7 C♯ø C♯6 C♯sus2 C♯sus4 C♯9
4231 231 41 211 4213 4112 1122 3142 1122 1134 3111 2134
D♭ D♭m D♭+ D♭o7 D♭7 D♭△ D♭m7 D♭ø D♭6 D♭sus2 D♭sus4 D♭9
4231 231 41 211 4213 4112 1122 3142 1122 1134 3111 2134
D Dm D+ Do7 D7 D△ Dm7 Dø D6 Dsus2 Dsus4 D9
1 2 2 1 3 12 1 32 1 32 1 42 2 31 1 32 1 23 1 1 2 421
D♯ D♯m D♯+ D♯o7 D♯7 D♯△ D♯m7 D♯ø D♯6 D♯sus2 D♯sus4 D♯9
3114 3112 123 2143 2143 2143 3142 2143 2134 3111 3114 2134
E♭ E♭m E♭+ E♭o7 E♭7 E♭△ E♭m7 E♭ø E♭6 E♭sus2 E♭sus4 E♭9
3114 3112 123 2143 2143 2143 3142 2143 2134 3111 3114 2134
E Em E+ Eo7 E7 E△ Em7 Eø E6 Esus2 Esus4 E9
123 23 1234 2143 1 2 112 2 1 132 3111 31 2134
F Fm F+ Fo7 F7 F△ Fm7 Fø F6 Fsus2 Fsus4 F9
23 1 1341 1234 1 32 2131 2341 1131 1121 2 31 341 4211 2134
F♯ F♯m F♯+ F♯o7 F♯7 F♯△ F♯m7 F♯ø F♯6 F♯sus2 F♯sus4 F♯9
2341 1341 1234 2143 2131 2341 1131 1121 3142 3111 4211 213
G♭ G♭m G♭+ G♭o7 G♭7 G♭△ G♭m7 G♭ø G♭6 G♭sus2 G♭sus4 G♭9
2341 1341 1234 2143 2131 2341 1131 1121 3142 3111 4211 213



A.5 Tamaños de página predefinidos

Los tamaños de página se definen en el archivo '`scm/paper.scm`'

La serie A de "ISO 216"

`"a10"` (26 x 37 mm)

`"a9"` (37 x 52 mm)

`"a8"` (52 x 74 mm)

`"a7"` (74 x 105 mm)

`"a6"` (105 x 148 mm)

`"a5"` (148 x 210 mm)

`"a4"` (210 x 297 mm)

`"a3"` (297 x 420 mm)

`"a2"` (420 x 594 mm)

`"a1"` (594 x 841 mm)

`"a0"` (841 x 1189 mm)

La serie B de "ISO 216"

`"b10"` (31 x 44 mm)

`"b9"` (44 x 62 mm)

`"b8"` (62 x 88 mm)

`"b7"` (88 x 125 mm)

`"b6"` (125 x 176 mm)

`"b5"` (176 x 250 mm)

`"b4"` (250 x 353 mm)

`"b3"` (353 x 500 mm)

`"b2"` (500 x 707 mm)

`"b1"` (707 x 1000 mm)

`"b0"` (1000 x 1414 mm)

Dos tamaños extendidos según se definen en "DIN 476"

`"4a0"` (1682 x 2378 mm)

`"2a0"` (1189 x 1682 mm)

Serie C estándar de "ISO 269"

`"c10"` (28 x 40 mm)

`"c9"` (40 x 57 mm)

`"c8"` (57 x 81 mm)

`"c7"` (81 x 114 mm)

`"c6"` (114 x 162 mm)

`"c5"` (162 x 229 mm)

`"c4"` (229 x 324 mm)

`"c3"` (324 x 458 mm)

`"c2"` (458 x 648 mm)

`"c1"` (648 x 917 mm)

`"c0"` (917 x 1297 mm)

Tamaños de página norteamiericanos

`"junior-legal"` (8.0 x 5.0 in)

`"legal"` (8.5 x 14.0 in)

`"ledger"` (17.0 x 11.0 in)

`"letter"` (8.5 x 11.0 in)

`"tabloid"`
(11.0 x 17.0 in)

`"11x17"` (11.0 x 17.0 in)

`"17x11"` (17.0 x 11.0 in)

Tamaño carta gubernamental del IEEE Printer Working Group, para escritura de niños

`"government-letter"`
(8 x 10.5 in)

`"government-legal"`
(8.5 x 13.0 in)

`"philippine-legal"`
(8.5 x 13.0 in)

Tamaños de ANSI

`"ansi a"` (8.5 x 11.0 in)

`"ansi b"` (17.0 x 11.0 in)

`"ansi c"` (17.0 x 22.0 in)

`"ansi d"` (22.0 x 34.0 in)

`"ansi e"` (34.0 x 44.0 in)

`"engineering f"`
(28.0 x 40.0 in)

Tamaños arquitectóncos norteamericanos

`"arch a"` (9.0 x 12.0 in)

`"arch b"` (12.0 x 18.0 in)

`"arch c"` (18.0 x 24.0 in)

`"arch d"` (24.0 x 36.0 in)

`"arch e"` (36.0 x 48.0 in)

`"arch e1"` (30.0 x 42.0 in)

Tamaños antiguos que aún se utilizan en el Reino Unido

`"statement"`
(5.5 x 8.5 in)

`"half letter"`
(5.5 x 8.5 in)

`"quarto"` (8.0 x 10.0 in)

`"octavo"` (6.75 x 10.5 in)

`"executive"`
(7.25 x 10.5 in)

`"monarch"`
(7.25 x 10.5 in)

`"foolscap"`
(8.27 x 13.0 in)

`"folio"` (8.27 x 13.0 in)

`"super-b"`
(13.0 x 19.0 in)

`"post"` (15.5 x 19.5 in)

`"crown"` (15.0 x 20.0 in)

`"large post"`
(16.5 x 21.0 in)

`"demy"` (17.5 x 22.5 in)

`"medium"` (18.0 x 23.0 in)

`"broadsheet"`
(18.0 x 24.0 in)

`"royal"` (20.0 x 25.0 in)

`"elephant"`
(23.0 x 28.0 in)

`"double demy"`
(22.5 x 35.0 in)

`"quad demy"`
(35.0 x 45.0 in)

`"atlas"` (26.0 x 34.0 in)

`"imperial"`
(22.0 x 30.0 in)

`"antiquarian"`
(31.0 x 53.0 in)

Tamaños basados en PA4

`"pa0"` (840 x 1120 mm)

`"pa1"` (560 x 840 mm)

`"pa2"` (420 x 560 mm)

`"pa3"` (280 x 420 mm)

`"pa4"` (210 x 280 mm)

`"pa5"` (140 x 210 mm)

`"pa6"` (105 x 140 mm)

`"pa7"` (70 x 105 mm)

`"pa8"` (52 x 70 mm)

`"pa9"` (35 x 52 mm)

`"pa10"` (26 x 35 mm)

Usado en el Asia suroriental y en Australia

`"f4"` (210 x 330 mm)

Usado para los diminutos ejemplos de `@lilypond` de la documentación; basado en a8 apaisado.

`"a8landscape"`
(74 x 52 mm)

A.6 instrumentos MIDI

A continuación figura una lista con los nombres que se pueden utilizar para la propiedad `midiInstrument` (instrumento MIDI). El orden de los instrumentos que aparece aquí, de arriba a abajo empezando por la columna de la izquierda, corresponde a los 128 números de programa del estándar General MIDI.

```
acoustic grand
bright acoustic
electric grand
honky-tonk
electric piano 1
electric piano 2
harpsichord
clav
celesta
glockenspiel
music box
vibraphone
marimba
xylophone
tubular bells
dulcimer
drawbar organ
percussive organ
rock organ
church organ
reed organ
accordion
harmonica
concertina
acoustic guitar (nylon)
acoustic guitar (steel)
electric guitar (jazz)
electric guitar (clean)
electric guitar (muted)
overdriven guitar
distorted guitar
guitar harmonics
acoustic bass
electric bass (finger)
electric bass (pick)
fretless bass
slap bass 1
slap bass 2
synth bass 1
synth bass 2
violin
viola
cello
contrabass
tremolo strings
pizzicato strings
orchestral harp
timpani
string ensemble 1
string ensemble 2
synthstrings 1
synthstrings 2
choir aahs
voice oohs
synth voice
orchestra hit
trumpet
trombone
tuba
muted trumpet
french horn
brass section
synthbrass 1
synthbrass 2
soprano sax
alto sax
tenor sax
baritone sax
oboe
english horn
bassoon
clarinet
piccolo
flute
recorder
pan flute
blown bottle
shakuhachi
whistle
ocarina
lead 1 (square)
lead 2 (sawtooth)
lead 3 (calliope)
lead 4 (chiff)
lead 5 (charang)
lead 6 (voice)
lead 7 (fifths)
lead 8 (bass+lead)
pad 1 (new age)
pad 2 (warm)
pad 3 (polysynth)
pad 4 (choir)
pad 5 (bowed)
pad 6 (metallic)
pad 7 (halo)
pad 8 (sweep)
fx 1 (rain)
fx 2 (soundtrack)
fx 3 (crystal)
fx 4 (atmosphere)
fx 5 (brightness)
fx 6 (goblins)
fx 7 (echoes)
fx 8 (sci-fi)
sitar
banjo
shamisen
koto
kalimba
bagpipe
fiddle
shanai
tinkle bell
agogo
steel drums
woodblock
taiko drum
melodic tom
synth drum
reverse cymbal
guitar fret noise
breath noise
seashore
bird tweet
telephone ring
helicopter
applause
gunshot
```

A.7 Lista de colores

Colores normales

La sintaxis para el uso de colores se encuentra detallada en [Colorear objetos], página 217.

```
black       white          red         green
blue        cyan           magenta     yellow
grey        darkred        darkgreen   darkblue
darkcyan    darkmagenta    darkyellow
```

Nombres de los colores de X

Los nombres de los colores de X admiten algunas variantes:

Un nombre que se escribe como una palabra única con mayúsculas intercaladas (p.ej. 'LightSlateBlue') se puede escribir también como palabras separadas por espacios y sin mayúsculas (p.ej. 'light slate blue').

La palabra 'grey' siempre se puede escribir como 'gray' (p.ej. 'DarkSlateGray').

Algunos nombres admiten un sufijo numérico (p.ej. 'LightSalmon4').

Nombres de los colores sin sufijo numérico:

```
snow              GhostWhite       WhiteSmoke       gainsboro        FloralWhite
OldLace           linen            AntiqueWhite     PapayaWhip       BlanchedAlmond
bisque            PeachPuff        NavajoWhite      moccasin         cornsilk
ivory             LemonChiffon     seashell         honeydew         MintCream
azure             AliceBlue        lavender         LavenderBlush    MistyRose
white             black            DarkSlateGrey    DimGrey          SlateGrey
LightSlateGrey    grey             LightGrey        MidnightBlue     navy
NavyBlue          CornflowerBlue   DarkSlateBlue    SlateBlue        MediumSlateBlue
LightSlateBlue    MediumBlue       RoyalBlue        blue             DodgerBlue
DeepSkyBlue       SkyBlue          LightSkyBlue     SteelBlue        LightSteelBlue
LightBlue         PowderBlue       PaleTurquoise    DarkTurquoise    MediumTurquoise
turquoise         cyan             LightCyan        CadetBlue        MediumAquamarine
aquamarine        DarkGreen        DarkOliveGreen   DarkSeaGreen     SeaGreen
MediumSeaGreen    LightSeaGreen    PaleGreen        SpringGreen      LawnGreen
green             chartreuse       MediumSpringGreen       GreenYellow      LimeGreen
YellowGreen       ForestGreen      OliveDrab        DarkKhaki        khaki
PaleGoldenrod     LightGoldenrodYellow     LightYellow      yellow   gold
LightGoldenrod    goldenrod        DarkGoldenrod    RosyBrown        IndianRed
SaddleBrown       sienna           peru             burlywood        beige
wheat             SandyBrown       tan              chocolate        firebrick
brown             DarkSalmon       salmon           LightSalmon      orange
DarkOrange        coral            LightCoral       tomato           OrangeRed
red               HotPink          DeepPink         pink             LightPink
PaleVioletRed     maroon           MediumVioletRed  VioletRed        magenta
violet            plum             orchid           MediumOrchid     DarkOrchid
DarkViolet        BlueViolet       purple           MediumPurple     thistle
DarkGrey          DarkBlue         DarkCyan         DarkMagenta      DarkRed
LightGreen
```

Nombres de los colores con sufijo numérico

En los nombres siguientes, el sufijo N puede ser un número del 1 al 4:

```
snowN             seashellN        AntiqueWhiteN    bisqueN          PeachPuffN
NavajoWhiteN      LemonChiffonN    cornsilkN        ivoryN           honeydewN
LavenderBlushN    MistyRoseN       azureN           SlateBlueN       RoyalBlueN
```

blueN	DodgerBlueN	SteelBlueN	DeepSkyBlueN	SkyBlueN
LightSkyBlueN	LightSteelBlueN	LightBlueN	LightCyanN	PaleTurquoiseN
CadetBlueN	turquoiseN	cyanN	aquamarineN	DarkSeaGreenN
SeaGreenN	PaleGreenN	SpringGreenN	greenN	chartreuseN
OliveDrabN	DarkOliveGreenN	khakiN	LightGoldenrodN	LightYellowN
yellowN	goldN	goldenrodN	DarkGoldenrodN	RosyBrownN
IndianRedN	siennaN	burlywoodN	wheatN	tanN
chocolateN	firebrickN	brownN	salmonN	LightSalmonN
orangeN	DarkOrangeN	coralN	tomatoN	OrangeRedN
redN	DeepPinkN	HotPinkN	pinkN	LightPinkN
PaleVioletRedN	maroonN	VioletRedN	magentaN	orchidN
plumN	MediumOrchidN	DarkOrchidN	purpleN	MediumPurpleN
thistleN				

Escala de grises

Se puede obtener una escala de grises utilizando:

greyN

donde N es un número entre 0 y 100.

A.8 La tipografía Feta

Los símbolos siguientes están disponibles en la tipografía Emmentaler y es posible acceder a ellos directamente utilizando elementos de marcado textual con el nombre del glifo tal y como se muestra en las tablas de abajo, por ejemplo `g^\markup {\musicglyph #"scripts.segno" }` ó `\markup {\musicglyph #"five"}`. Para ver más información, consulte Sección 1.8.2 [Formatear el texto], página 233.

Glifos de clave

clefs.C		clefs.C_change	
clefs.F		clefs.F_change	
clefs.G		clefs.G_change	
clefs.percussion		clefs.percussion_change	
clefs.tab		clefs.tab_change	

Glifos de indicación de compás

timesig.C44		timesig.C22	

Glifos de cifras

plus	+	comma	,
hyphen	-	period	.
zero	0	one	1
two	2	three	3
four	4	five	5
six	6	seven	7
eight	8	nine	9

Glifos de alteraciones

accidentals.sharp		accidentals .sharp.arrowup	
accidentals .sharp.arrowdown		accidentals .sharp.arrowboth	
accidentals.sharp .slashslash.stem		accidentals.sharp .slashslashslash.stemstem	
accidentals.sharp .slashslashslash.stem		accidentals.sharp .slashslash.stemstemstem	
accidentals.natural		accidentals .natural.arrowup	
accidentals .natural.arrowdown		accidentals .natural.arrowboth	
accidentals.flat		accidentals.flat.arrowup	

accidentals .flat.arrowdown		accidentals .flat.arrowboth	
accidentals.flat.slash		accidentals.flat .slashslash	
accidentals .mirroredflat.flat		accidentals.mirroredflat	
accidentals .mirroredflat.backslash		accidentals.flatflat	
accidentals .flatflat.slash		accidentals.doublesharp	
accidentals.rightparen		accidentals.leftparen	

Glifos de las cabezas de nota predeterminadas

noteheads.uM2		noteheads.dM2	
noteheads.sM1		noteheads.s0	
noteheads.s1		noteheads.s2	

Glifos de las cabezas de nota especiales

noteheads.sM1double		noteheads.s0diamond	
noteheads.s1diamond		noteheads.s2diamond	
noteheads.s0triangle		noteheads.d1triangle	
noteheads.u1triangle		noteheads.u2triangle	
noteheads.d2triangle		noteheads.s0slash	

noteheads.s1slash		noteheads.s2slash	
noteheads.s0cross		noteheads.s1cross	
noteheads.s2cross		noteheads.s2xcircle	
noteheads.s0harmonic		noteheads.s2harmonic	

Glifos de las cabezas de nota con formas

noteheads.s0do		noteheads.d1do	
noteheads.u1do		noteheads.d2do	
noteheads.u2do		noteheads.s0doThin	
noteheads.d1doThin		noteheads.u1doThin	
noteheads.d2doThin		noteheads.u2doThin	
noteheads.s0re		noteheads.u1re	
noteheads.d1re		noteheads.u2re	
noteheads.d2re		noteheads.s0reThin	
noteheads.u1reThin		noteheads.d1reThin	
noteheads.u2reThin		noteheads.d2reThin	
noteheads.s0mi		noteheads.s1mi	

noteheads.s2mi

noteheads.s0miMirror

noteheads.s1miMirror

noteheads.s2miMirror

noteheads.s0miThin

noteheads.s1miThin

noteheads.s2miThin

noteheads.u0fa

noteheads.d0fa

noteheads.u1fa

noteheads.d1fa

noteheads.u2fa

noteheads.d2fa

noteheads.u0faThin

noteheads.d0faThin

noteheads.u1faThin

noteheads.d1faThin

noteheads.u2faThin 

noteheads.d2faThin 

noteheads.s0sol 

noteheads.s1sol

noteheads.s2sol

noteheads.s0la

noteheads.s1la

noteheads.s2la

noteheads.s0laThin

noteheads.s1laThin

noteheads.s2laThin

noteheads.s0ti

noteheads.u1ti

noteheads.d1ti

noteheads.u2ti

noteheads.d2ti

noteheads.s0tiThin

noteheads.u1tiThin

noteheads.d1tiThin

noteheads.u2tiThin

noteheads.d2tiThin

noteheads.u0doFunk

noteheads.d0doFunk

noteheads.u1doFunk

noteheads.d1doFunk

noteheads.u2doFunk

noteheads.d2doFunk

noteheads.u0reFunk

noteheads.d0reFunk

noteheads.u1reFunk

noteheads.d1reFunk

noteheads.u2reFunk

noteheads.d2reFunk

noteheads.u0miFunk

noteheads.d0miFunk

noteheads.u1miFunk

noteheads.d1miFunk

noteheads.s2miFunk

noteheads.u0faFunk

noteheads.d0faFunk

noteheads.u1faFunk

noteheads.d1faFunk

noteheads.u2faFunk

noteheads.d2faFunk

noteheads.s0solFunk

noteheads.s1solFunk

noteheads.s2solFunk

noteheads.s0laFunk

noteheads.s1laFunk

noteheads.s2laFunk

noteheads.u0tiFunk

noteheads.d0tiFunk

noteheads.u1tiFunk

noteheads.d1tiFunk

noteheads.u2tiFunk

noteheads.d2tiFunk

noteheads.s0doWalker

noteheads.u1doWalker

noteheads.d1doWalker

noteheads.u2doWalker

noteheads.d2doWalker

noteheads.s0reWalker

noteheads.u1reWalker

noteheads.d1reWalker

noteheads.u2reWalker

noteheads.d2reWalker

noteheads.s0miWalker

noteheads.s1miWalker

noteheads.s2miWalker

noteheads.s0faWalker

noteheads.u1faWalker

noteheads.d1faWalker

noteheads.u2faWalker

noteheads.d2faWalker		noteheads.s0laWalker	
noteheads.s1laWalker		noteheads.s2laWalker	
noteheads.s0tiWalker		noteheads.u1tiWalker	
noteheads.d1tiWalker		noteheads.u2tiWalker	
noteheads.d2tiWalker			

Glifos de silencios

rests.0		rests.1	
rests.0o		rests.1o	
rests.M3		rests.M2	
rests.M1		rests.M1o	
rests.2		rests.2classical	
rests.3		rests.4	
rests.5		rests.6	
rests.7			

Glifos de corchetes

flags.u3		flags.u4	
flags.u5		flags.u6	
flags.u7		flags.d3	
flags.d4		flags.d5	
flags.d6		flags.d7	
flags.ugrace		flags.dgrace	

Glifos de puntillos

dots.dot	

Glifos de matices dinámicos

space		f	
m		p	
r		s	
z			

Grifos de inscripciones

scripts.ufermata		scripts.dfermata	
scripts.ushortfermata		scripts.dshortfermata	
scripts.ulongfermata		scripts.dlongfermata	
scripts.uverylongfermata		scripts.dverylongfermata	
scripts.thumb		scripts.sforzato	
scripts.espr		scripts.staccato	
scripts.ustaccatissimo		scripts.dstaccatissimo	
scripts.tenuto		scripts.uportato	
scripts.dportato		scripts.umarcato	
scripts.dmarcato		scripts.open	
scripts.halfopen		scripts.halfopenvertical	
scripts.stopped		scripts.upbow	
scripts.downbow		scripts.reverseturn	
scripts.turn		scripts.trill	

scripts.upedalheel		scripts.dpedalheel	
scripts.upedaltoe		scripts.dpedaltoe	
scripts.flageolet		scripts.segno	
scripts.varsegno		scripts.coda	
scripts.varcoda		scripts.rcomma	
scripts.lcomma		scripts.rvarcomma	
scripts.lvarcomma		scripts.arpeggio	
scripts.trill_element		scripts.arpeggio .arrow.M1	
scripts.arpeggio.arrow.1		scripts.trilelement	
scripts.prall		scripts.mordent	
scripts.prallprall		scripts.prallmordent	
scripts.upprall		scripts.upmordent	
scripts.pralldown		scripts.downprall	
scripts.downmordent		scripts.prallup	
scripts.lineprall		scripts.caesura.curved	

scripts.caesura.straight		scripts.tickmark	
scripts.snappizzicato		scripts.ictus	
scripts.uaccentus		scripts.daccentus	
scripts.usemicirculus		scripts.dsemicirculus	
scripts.circulus		scripts.augmentum	
scripts .usignumcongruentiae		scripts .dsignumcongruentiae	

Glifos de flechas

arrowheads.open.01		arrowheads.open.0M1	
arrowheads.open.11		arrowheads.open.1M1	
arrowheads.close.01		arrowheads.close.0M1	
arrowheads.close.11		arrowheads.close.1M1	

Glifos de puntas de corchete

brackettips.up		brackettips.down	

Glifos de pedal

pedal.*		pedal.M	
pedal..		pedal.P	
pedal.d		pedal.e	
pedal.Ped			

Glifos de acordeón

accordion.discant 

accordion.dot

accordion.freebass 

accordion.stdbass 

accordion.bayanbass 

accordion.oldEE 

accordion.push 

accordion.pull 

Glifos de ligadura

ties.lyric.short

ties.lyric.default

Glifos de vaticana

clefs.vaticana.do

clefs.vaticana.do_change

clefs.vaticana.fa 

clefs.vaticana.fa_change 

custodes.vaticana.u0

custodes.vaticana.u1

custodes.vaticana.u2

custodes.vaticana.d0

custodes.vaticana.d1

custodes.vaticana.d2

accidentals.vaticanaM1

accidentals.vaticana0

dots.dotvaticana

noteheads .svaticana.punctum

noteheads.svaticana .punctum.cavum

noteheads.svaticana .linea.punctum

noteheads.svaticana.linea.punctum.cavum

noteheads.svaticana.inclinatum

noteheads.svaticana.lpes

noteheads.svaticana.vlpes

noteheads.svaticana.upes

noteheads.svaticana.vupes

noteheads.svaticana.plica

noteheads.svaticana.vplica

noteheads.svaticana.epiphonus

noteheads.svaticana.vepiphonus

noteheads.svaticana.reverse.plica

noteheads.svaticana.reverse.vplica

noteheads.svaticana.inner.cephalicus

noteheads.svaticana.cephalicus

noteheads.svaticana.quilisma

Glifos de medicea

clefs.medicaea.do

clefs.medicaea.do_change

clefs.medicaea.fa

clefs.medicaea.fa_change

custodes.medicaea.u0

custodes.medicaea.u1

custodes.medicaea.u2

custodes.medicaea.d0

custodes.medicaea.d1

custodes.medicaea.d2

accidentals.medicaeaM1

noteheads.smedicaea.inclinatum

noteheads.smedicaea.punctum

noteheads.smedicaea.rvirga

noteheads.smedicaea.virga

Glifos de Hufnagel

clefs.hufnagel.do 

clefs.hufnagel.do_change 

clefs.hufnagel.fa 

clefs.hufnagel.fa_change 

clefs.hufnagel.do.fa 

clefs.hufnagel .do.fa_change 

custodes.hufnagel.u0 

custodes.hufnagel.u1 

custodes.hufnagel.u2 

custodes.hufnagel.d0 

custodes.hufnagel.d1 

custodes.hufnagel.d2 

accidentals.hufnagelM1 

noteheads .shufnagel.punctum

noteheads .shufnagel.virga 

noteheads.shufnagel.lpes

Glifos de mensural

rests.M3mensural

rests.M2mensural

rests.M1mensural

rests.0mensural

rests.1mensural

rests.2mensural

rests.3mensural

rests.4mensural

clefs.mensural.c 

clefs.mensural.c_change 

clefs.blackmensural.c 

clefs.blackmensural .c_change 

clefs.mensural.f		clefs.mensural.f_change	
clefs.mensural.g		clefs.mensural.g_change	
custodes.mensural.u0		custodes.mensural.u1	
custodes.mensural.u2		custodes.mensural.d0	
custodes.mensural.d1		custodes.mensural.d2	
accidentals.mensural1		accidentals.mensuralM1	
flags.mensuralu03		flags.mensuralu13	
flags.mensuralu23		flags.mensurald03	
flags.mensurald13		flags.mensurald23	
flags.mensuralu04		flags.mensuralu14	
flags.mensuralu24		flags.mensurald04	
flags.mensurald14		flags.mensurald24	
flags.mensuralu05		flags.mensuralu15	
flags.mensuralu25		flags.mensurald05	
flags.mensurald15		flags.mensurald25	

flags.mensuralu06 

flags.mensuralu16

flags.mensuralu26 

flags.mensurald06

flags.mensurald16

flags.mensurald26

timesig.mensural44 

timesig.mensural22 

timesig.mensural32 

timesig.mensural64 

timesig.mensural94 

timesig.mensural34 

timesig.mensural68 

timesig.mensural98 

timesig.mensural48 

timesig.mensural68alt 

timesig.mensural24 

noteheads.uM3mensural 

noteheads.dM3mensural 

noteheads.sM3ligmensural 

noteheads.uM2mensural 

noteheads.dM2mensural 

noteheads.sM2ligmensural 

noteheads.sM1mensural 

noteheads.urM3mensural 

noteheads.drM3mensural 

noteheads
.srM3ligmensural 

noteheads.urM2mensural 

noteheads.drM2mensural 

noteheads
.srM2ligmensural 

noteheads.srM1mensural 

noteheads .uM3semimensural 

noteheads .dM3semimensural 

noteheads .sM3semiligmensural 

noteheads .uM2semimensural 

noteheads .dM2semimensural 

noteheads .sM2semiligmensural 

noteheads .sM1semimensural 

noteheads .urM3semimensural 

noteheads .drM3semimensural 

noteheads .srM3semiligmensural 

noteheads .urM2semimensural 

noteheads .drM2semimensural 

noteheads .srM2semiligmensural 

noteheads .srM1semimensural 

noteheads .uM3blackmensural 

noteheads .dM3blackmensural 

noteheads .sM3blackligmensural 

noteheads .uM2blackmensural 

noteheads .dM2blackmensural 

noteheads .sM2blackligmensural 

noteheads .sM1blackmensural 

noteheads.s0mensural

noteheads.s1mensural

noteheads.s2mensural

noteheads .s0blackmensural

Glifos de neomensural

rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural		rests.0neomensural	
rests.1neomensural		rests.2neomensural	
rests.3neomensural		rests.4neomensural	
clefs.neomensural.c		clefs.neomensural .c_change	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	
timesig.neomensural68		timesig.neomensural98	
timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		noteheads.uM3neomensural	
noteheads.dM3neomensural		noteheads.uM2neomensural	
noteheads.dM2neomensural		noteheads.sM1neomensural	
noteheads .urM3neomensural		noteheads .drM3neomensural	

noteheads .urM2neomensural		noteheads .drM2neomensural	
noteheads .srM1neomensural		noteheads.s0neomensural	
noteheads.s1neomensural		noteheads.s2neomensural	

Glifos de Petrucci

clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	
clefs.petrucci.c5		clefs.petrucci.c5_change	
clefs.petrucci.f		clefs.petrucci.f_change	
clefs.petrucci.g		clefs.petrucci.g_change	
noteheads.s0petrucci		noteheads.s1petrucci	
noteheads.s2petrucci		noteheads .s0blackpetrucci	
noteheads .s1blackpetrucci		noteheads .s2blackpetrucci	

Glifos de Solesmes

noteheads.ssolesmes.incl.parvum

noteheads.ssolesmes.auct.asc

noteheads.ssolesmes.auct.desc

noteheads.ssolesmes.incl.auctum

noteheads.ssolesmes.stropha

noteheads.ssolesmes.stropha.aucta

noteheads.ssolesmes.oriscus

Glifos de la notación del canto kievano

clefs.kievan.do

clefs.kievan.do_change

accidentals.kievan1

accidentals.kievanM1

scripts.barline.kievan

dots.dotkievan

noteheads.sM2kievan

noteheads.sM1kievan

noteheads.s0kievan

noteheads.d2kievan

noteheads.u2kievan

noteheads.s1kievan

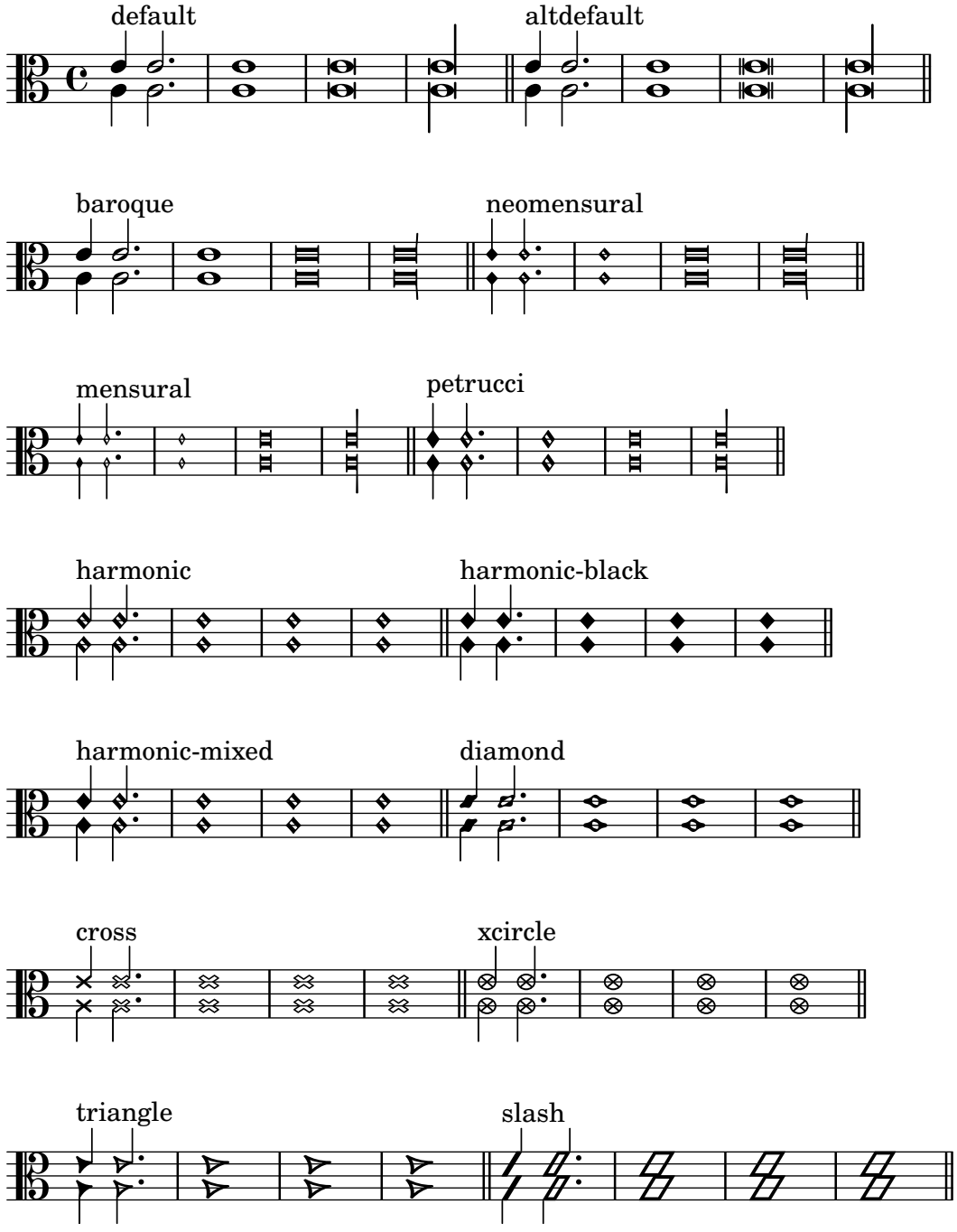
noteheads.sr1kievan

noteheads.d3kievan

noteheads.u3kievan

A.9 Estilos de cabezas de nota

Se pueden usar los siguientes estilos para las cabezas de las notas.



A.10 Instrucciones de marcado de texto

The following commands can all be used inside `\markup { }`.

A.10.1 Font

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size to display *arg*. Adjusts `baseline-skip` and `word-space` accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

default text font size text font size 16 text font size 12

`\bold` *arg* (markup)
Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```

default **bold**

`\box` *arg* (markup)
Draw a box round *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
  \line { V. S. }
}
```

V. S.

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps` *arg* (markup)
Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

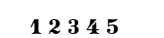
`\dynamic` *arg* (markup)
Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like 'più **f**', the normal words (like 'più') should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```



`\finger` *arg* (markup)
Set *arg* as small numbers.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```



`\fontCaps` *arg* (markup)
Set `font-shape` to `caps`

Note: `\fontCaps` requires the installation and selection of fonts which support the `caps` font shape.

`\fontsize` *increment* (number) *arg* (markup)
Add *increment* to the font-size. Adjusts `baseline-skip` accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
}
```


Used properties:

- `baseline-skip` (2)
- `word-space` (1)
- `font-size` (0)

`\huge` *arg* (markup)
Set font size to +2.

```
\markup {
  default
  \hspace #2
  \huge
  huge
}
```



`\italic` *arg* (markup)
Use italic `font-shape` for *arg*.

```
\markup {
  default
  \hspace #2
  \italic
  italic
}
```

default *italic*

`\large` *arg* (markup)
Set font size to +1.

```
\markup {
  default
  \hspace #2
  \large
  large
}
```

default large

`\larger` *arg* (markup)
Increase the font size relative to the current setting.

```
\markup {
  default
  \hspace #2
  \larger
  larger
}
```

default larger

`\magnify` *sz* (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```
\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}
```

default 50% larger

`\medium` *arg* (markup)
Switch to medium font-series (in contrast to bold).

```
\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}
```

some bold text medium font series **bold again**

`\normal-size-sub` *arg* (markup)
Set *arg* in subscript with a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

default $_{\text{subscript in standard size}}$

Used properties:

- `baseline-skip`

`\normal-size-super` *arg* (markup)
Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

default $^{\text{superscript in standard size}}$

Used properties:

- `baseline-skip`

`\normal-text` *arg* (markup)
Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \caps {
    huge bold sans caps
    \hspace #2
    \normal-text {
      huge normal
    }
```

```
    \hspace #2
    as before
  }
}
```

HUGE BOLD SANS CAPS huge normal **AS BEFORE**

`\normalsize` *arg* (markup)
Set font size to default.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small normal size teeny again

`\number` *arg* (markup)
Set font family to `number`, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

0123456789.,

`\replace` *replacements* (list) *arg* (markup)
Used to automatically replace a string by another in the markup *arg*. Each pair of the alist *replacements* specifies what should be replaced. The `key` is the string to be replaced by the `value` string.

```
\markup \replace #'(("thx" . "Thanks!")) thx
```

Thanks!

`\roman` *arg* (markup)
Set font family to `roman`.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \roman {
      text in roman font family
    }
```

```
    \hspace #2
    return to sans
  }
}
```

sans serif, bold text in roman font family return to sans

`\sans` *arg* (markup)
Switch to the sans serif font family.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

default sans serif

`\simple` *str* (string)
A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

Note: for creating standard text markup or defining new markup commands, the use of `\simple` is unnecessary.

```
\markup {
  \simple #"simple"
  \simple #"text"
  \simple #"strings"
}
```

simple text strings

`\small` *arg* (markup)
Set font size to -1.

```
\markup {
  default
  \hspace #2
  \small
  small
}
```

default small

`\smallCaps` *arg* (markup)
Emit *arg* as small caps.

Note: `\smallCaps` does not support accented characters.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
```

```
  }
}
```

default TEXT IN SMALL CAPS

`\smaller` *arg* (markup)
Decrease the font size relative to the current setting.

```
\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
    \hspace #2
    more large text
  }
}
```

some large text a bit smaller more large text

`\sub` *arg* (markup)
Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
    O
  }
}
```

H_2O

Used properties:

- `baseline-skip`
- `font-size` (0)

`\super` *arg* (markup)
Set *arg* in superscript.

```
\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}
```

$E = mc^2$

Used properties:

- `baseline-skip`
- `font-size` (0)

`\teeny` *arg* (markup)
Set font size to -3.

```
\markup {
  default
  \hspace #2
  \teeny
  teeny
}
```

default teeny

`\text` *arg* (markup)
Use a text font instead of music symbol or music alphabet font.

```
\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
    5
  }
}
```

1, 2, three, four, **5**

`\tiny` *arg* (markup)
Set font size to -2.

```
\markup {
  default
  \hspace #2
  \tiny
  tiny
}
```

default tiny

`\typewriter` *arg* (markup)
Use `font-family` typewriter for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

default `typewriter`

`\underline` *arg* (markup)
Underline *arg*. Looks at `thickness` to determine line thickness, and `offset` to determine line y-offset.

```
\markup \fill-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
}
```

underlined underlined underlined

Used properties:

- `offset` (2)
- `thickness` (1)

`\upright` *arg* (markup)
Set `font-shape` to `upright`. This is the opposite of `italic`.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

A.10.2 Align

`\center-align` *arg* (markup)
Align `arg` to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}
```

one
two
three

`\center-column` *args* (markup list)
Put `args` in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\column` *args* (markup list)

Stack the markups in *args* vertically. The property `baseline-skip` determines the space between markups in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)

Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```
\combine { a list }
```

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (markup list)

Concatenate *args* in a horizontal line, without spaces in between. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.

```
\markup {
  \concat {
    one
    two
    three
  }
}
```

onetwothree

`\dir-column` *args* (markup list)
Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```
\markup {
  \override #`(direction . ,UP) {
    \dir-column {
      going up
    }
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #'(direction . 1) {
    \dir-column {
      going up
    }
  }
}
```

up
going
going
down
up
going

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *args* (markup list)
Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```
\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
```

```
      }
      \line { across the page }
    }
  }
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Used properties:

- `line-width` (#f)
- `word-space` (0.6)
- `text-direction` (1)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* in a horizontal line of width `line-width` with a line of markups *pattern* in between. Patterns are spaced apart by *space*. Patterns are aligned to the *dir* markup.

```
\markup \column {
  "right-aligned :"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned :"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned :"
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left first
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left second
}
```

right-aligned :
first . . . . . right
second . . . . . right

center-aligned :
left - right

left-aligned :
left : : : : : : : : : : : : : : : first
left : : : : : : : : : : : : : : : second

Used properties:

- `line-width`
- `word-space`

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)
Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
    three
    \null
    one
    \general-align #X #CENTER
    two
    three
    \null
    \line {
      one
      \general-align #Y #UP
      two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2
      two
      three
    }
  }
}
```

one
two
three

one
two
three

one two three

one three
two

`\halign` *dir* (number) *arg* (markup)
Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values in between interpolate alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT
```

```
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}
```

one
two
three

one
two
three
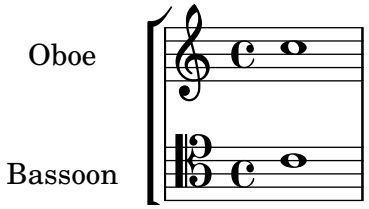
one
two
three

one
two
three

`\hcenter-in` *length* (number) *arg* (markup)
Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
```

```
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)

Create an invisible object taking up horizontal space *amount*.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```

one two three

`\justify-field` *symbol* (symbol)

Justify the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:myText
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify` *args* (markup list)

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; X is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)
Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
    three
  }
}
```

one
two
three

`\left-column` *args* (markup list)
Put `args` in a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\line` *args* (markup list)
Put *args* in a horizontal line. The property `word-space` determines the space between markups in *args*.

```
\markup {
  \line {
    one two three
  }
}
```

one two three

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)
Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

```
\markup {
  one
  \lower #3
  two
  three
}
```

one three
two

`\pad-around` *amount* (number) *arg* (markup)
Add padding *amount* all around *arg*.

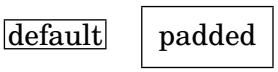
```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```

default padded

`\pad-markup` *amount* (number) *arg* (markup)
Add space around a markup object. Identical to `pad-around`.

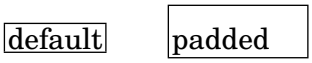
```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
```

```
    }
  }
}
```



`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)
Make *arg* take at least *x-ext*, *y-ext* space.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}
```



`\pad-x` *amount* (number) *arg* (markup)
Add padding *amount* around *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}
```



`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)
Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)
Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

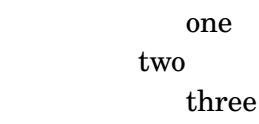
If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```
\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}
```



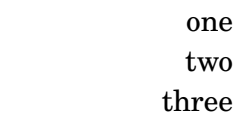
`\right-align` *arg* (markup)
Align *arg* on its right edge.

```
\markup {
  \column {
    one
    \right-align
    two
    three
  }
}
```



`\right-column` *args* (markup list)
Put `args` in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

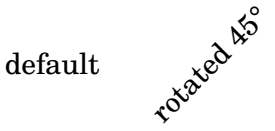


Used properties:

- `baseline-skip`

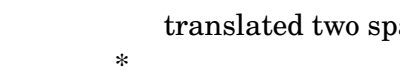
`\rotate` *ang* (number) *arg* (markup)
Rotate object with *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}
```



`\translate` *offset* (pair of numbers) *arg* (markup)
Translate *arg* relative to its surroundings. *offset* is a pair of numbers representing the displacement in the X and Y axis.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}
```



`\translate-scaled` *offset* (pair of numbers) *arg* (markup)
Translate *arg* by *offset*, scaling the offset by the `font-size`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```

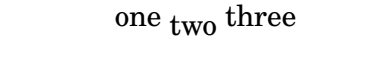


Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)
Align `arg` to its Y center.

```
\markup {
  one
  \vcenter
  two
  three
}
```



`\vspace` *amount* (number)
Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
    \center-column {
    one
    \vspace #2
    two
    \vspace #5
    three
  }
```

```
}
```

one

two

three

`\wordwrap-field` *symbol* (symbol)
Wordwrap the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua.  Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:myText
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\wordwrap` *args* (markup list)
Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where X is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string` *arg* (string)
Wordwrap a string. Paragraphs may be separated with double newlines.

```
\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et
dolore magna aliqua.
Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

Used properties:

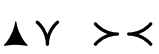
- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

A.10.3 Graphic

`\arrow-head` *axis* (integer) *dir* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```
\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}
```



`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

```
\markup {
  \beam #5 #1 #2
}
```



`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

```
\markup {
  \bracket {
    \note #"2." #UP
  }
}
```



`\circle` *arg* (markup)

Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
```

```
}
```



Used properties:

- `circle-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\draw-circle` *radius* (number) *thickness* (number) *filled* (boolean)
A circle of radius *radius* and thickness *thickness*, optionally filled.

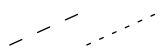
```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-dashed-line` *dest* (pair of numbers)
A dashed line.

If `full-length` is set to #t (default) the dashed-line extends to the whole length given by *dest*, without white space at beginning or end. `off` will then be altered to fit. To insist on the given (or default) values of `on`, `off` use `\override #'(full-length . #f)` Manual settings for `on`,`off` and `phase` are possible.

```
\markup {
  \draw-dashed-line #'(5.1 . 2.3)
  \override #'(on . 0.3)
  \override #'(off . 0.5)
  \draw-dashed-line #'(5.1 . 2.3)
}
```



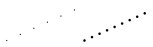
Used properties:

- `full-length` (#t)
- `phase` (0)
- `off` (1)
- `on` (1)
- `thickness` (1)

`\draw-dotted-line` *dest* (pair of numbers)
A dotted line.

The dotted-line always extends to the whole length given by *dest*, without white space at beginning or end. Manual settings for `off` are possible to get larger or smaller space between the dots. The given (or default) value of `off` will be altered to fit the line-length.

```
\markup {
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'(thickness . 2)
  \override #'(off . 0.2)
  \draw-dotted-line #'(5.1 . 2.3)
}
```



Used properties:

- `phase` (0)
- `off` (1)
- `thickness` (1)

`\draw-hline`

Draws a line across a page, where the property `span-factor` controls what fraction of the page is taken up.

```
\markup {
  \column {
    \draw-hline
    \override #'(span-factor . 1/3)
    \draw-hline
  }
}
```



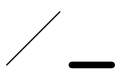
Used properties:

- `span-factor` (1)
- `line-width`
- `draw-line-markup`

`\draw-line` *dest* (pair of numbers)

A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



Used properties:

- `thickness` (1)

`\ellipse` *arg* (markup)

Draw an ellipse around *arg*. Use `thickness`, `x-padding`, `y-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \ellipse {
```

```
    Hi
  }
}
```

Hi

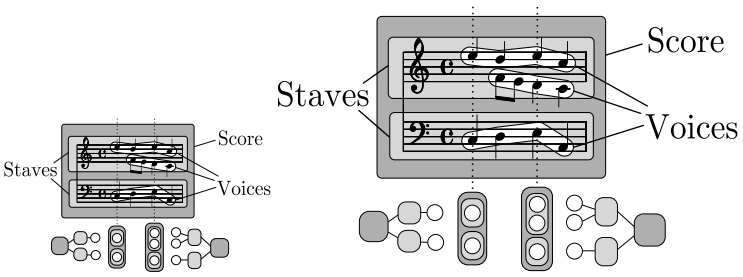
Used properties:

- `y-padding` (0.2)
- `x-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\epsfile` *axis* (number) *size* (number) *file-name* (string)

Inline an EPS image. The image is scaled along *axis* to *size*.

```
\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
    \epsfile #Y #20 #"context-example.eps"
  }
}
```



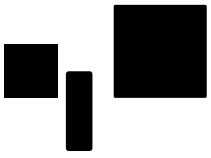
`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

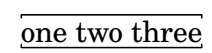
```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}
```



`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```



`\oval` *arg* (markup)
: Draw an oval around *arg*. Use `thickness`, `x-padding`, `x-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \oval {
    Hi
  }
}
```

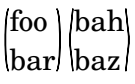


Used properties:

- `y-padding` (0.75)
- `x-padding` (0.75)
- `font-size` (0)
- `thickness` (1)

`\parenthesize` *arg* (markup)
: Draw parentheses around *arg*. This is useful for parenthesizing a column containing several lines of text.

```
\markup {
  \line {
    \parenthesize {
      \column {
        foo
        bar
      }
    }
    \override #'(angularity . 2) {
      \parenthesize {
        \column {
          bah
          baz
        }
      }
    }
  }
}
```



Used properties:

- `width` (0.25)
- `thickness` (1)
- `size` (1)
- `padding`
- `angularity` (0)

`\path` *thickness* (number) *commands* (list)

Draws a path with line *thickness* according to the directions given in *commands*. *commands* is a list of lists where the `car` of each sublist is a drawing command and the `cdr` comprises the associated arguments for each command.

There are seven commands available to use in the list `commands`: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with *r* are the relative variants of the other three commands.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments; they are the X and Y coordinates for the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Note that a sequence of commands *must* begin with a `moveto` or `rmoveto` to work with the SVG output.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `'butt`, `'round`, and `'square`. Available line-join styles are `'miter`, `'round`, and `'bevel`.

The property `filled` specifies whether or not the path is filled with color.

```
samplePath =
  #'((moveto 0 0)
     (lineto -1 1)
     (lineto 1 1)
     (lineto 1 -1)
     (curveto -5 -5 -5 5 -1 0)
     (closepath))

\markup {
  \path #0.25 #samplePath

  \override #'(line-join-style . miter) \path #0.25 #samplePath

  \override #'(filled . #t) \path #0.25 #samplePath
}
```



Used properties:

- `filled` (#f)

- `line-join-style` (round)
- `line-cap-style` (round)
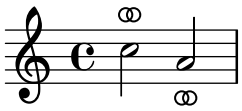
`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string.

```
ringsps = #"
  0.15 setlinewidth
  0.9 0.6 moveto
  0.4 0.6 0.5 0 361 arc
  stroke
  1.0 0.6 0.5 0 361 arc
  stroke
  "

rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}
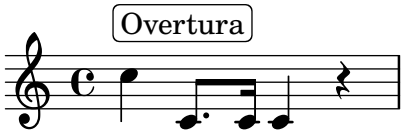
\relative c'' {
  c2^\rings
  a2_\rings
}
```



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup; the `corner-radius` property makes it possible to define another shape for the corners (default is 1).

```
c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



Used properties:

- `box-padding` (0.5)
- `font-size` (0)
- `corner-radius` (1)
- `thickness` (1)

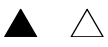
`\scale` *factor-pair* (pair of numbers) *arg* (markup)
Scale *arg*. *factor-pair* is a pair of numbers representing the scaling-factor in the X and Y axes. Negative values may be used to produce mirror images.

```
\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}
```



`\triangle` *filled* (boolean)
A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```



Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)
Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"http://lilypond.org/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

A.10.4 Music

`\customTabClef` *num-strings* (integer) *staff-space* (number)
Draw a tab clef sans-serif style.

`\doubleflat`
Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`

Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```

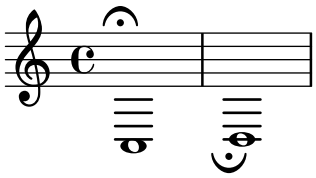


`\fermata`

Create a fermata glyph. When *direction* is `DOWN`, use an inverted glyph. Note that within music, one would usually use the `\fermata` articulation instead of a markup.

```
{ c1^\markup \fermata d1_\markup \fermata }

\markup { \fermata \override #`(direction . ,DOWN) \fermata }
```





Used properties:

- `direction` (1)

`\flat`

Draw a flat symbol.

```
\markup {
  \flat
}
```



`\musicglyph` *glyph-name* (string)

glyph-name is converted to a musical symbol; for example, `\musicglyph #"accidentals.natural"` selects the natural sign from the music font. See Sección "The Feta font" in *Referencia de la Notación* for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #"f"
  \musicglyph #"rests.2"
  \musicglyph #"clefs.G_change"
}
```



`\natural`

Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)
: Construct a note symbol, with stem and flag. By using fractional values for *dir*, longer or shorter stems can be obtained. Supports all note-head-styles. Supported flag-styles are `default`, `old-straight-flag`, `modern-straight-flag` and `flat-flag`.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```



Used properties:

- `style` ('())
- `flag-style` ('())
- `font-size` (0)

`\note` *duration* (string) *dir* (number)
: This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #"4." #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross) {
    \note #"4.." #UP
  }
  \hspace #2
  \note #"breve" #0
}
```



Used properties:

- `style` ('())
- `flag-style` ('())
- `font-size` (0)

`\rest-by-number` *log* (number) *dot-count* (number)
: A rest or multi-measure-rest symbol.

```
\markup {
  \rest-by-number #3 #2
  \hspace #2
  \rest-by-number #0 #1
```

```
  \hspace #2
  \override #'(multi-measure-rest . #t)
  \rest-by-number #0 #0
}
```



Used properties:

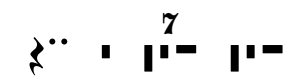
- `multi-measure-rest` (#f)
- `style` ('())
- `font-size` (0)

`\rest` *duration* (string)

This produces a rest, with the *duration* for the rest type and augmentation dots. `"breve"`, `"longa"` and `"maxima"` are valid input-strings.

Printing MultiMeasureRests could be enabled with `\override #'(multi-measure-rest . #t)` If MultiMeasureRests are taken, the MultiMeasureRestNumber is printed above. This is enabled for all styles using default-glyphs. Could be disabled with `\override #'(multi-measure-rest-number . #f)`

```
\markup {
  \rest #"4.."
  \hspace #2
  \rest #"breve"
  \hspace #2
  \override #'(multi-measure-rest . #t)
  {
  \rest #"7"
  \hspace #2
  \override #'(multi-measure-rest-number . #f)
  \rest #"7"
  }
}
```



Used properties:

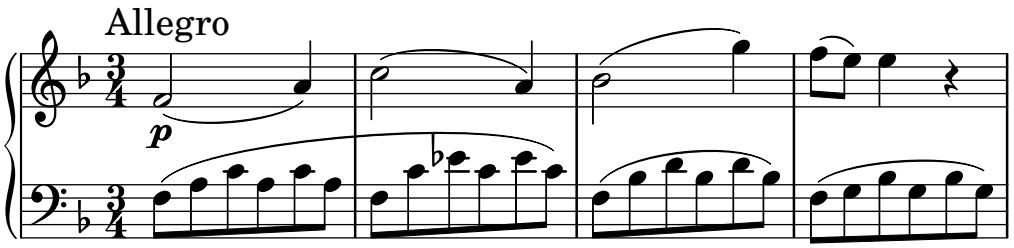
- `word-space` (0.6)
- `multi-measure-rest-number` (#t)
- `multi-measure-rest` (#f)
- `style` ('())

`\score` *score* (score)

Inline an image of music. The reference point (usually the middle staff line) of the lowest staff in the top system is placed on the baseline.

```
\markup {
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
```

```
        \mark \markup { Allegro }
        f2\p( a4)
        c2( a4)
        bes2( g'4)
        f8( e) e4 r
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
        f c' es c es c)
        f,( bes d bes d bes)
        f( g bes g bes g)
      }
    >>
    \layout {
      indent = 0.0\cm
      \context {
        \Score
        \override RehearsalMark
          #'break-align-symbols = #'(time-signature key-signature)
        \override RehearsalMark
          #'self-alignment-X = #LEFT
      }
      \context {
        \Staff
        \override TimeSignature
          #'break-align-anchor-alignment = #LEFT
      }
    }
  }
}
```



Used properties:

- `baseline-skip`

`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```



`\semisharp`

Draw a semisharp symbol.

```
\markup {
  \semisharp
}
```



`\sesquiflat`

Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
}
```



`\sesquisharp`

Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```



`\sharp`

Draw a sharp symbol.

```
\markup {
  \sharp
}
```



`\tied-lyric` *str* (string)

Like simple-markup, but use tie characters for '~' tilde symbols.

```
\markup \column {
  \tied-lyric #"Siam navi~all'onde~algenti Lasciate~in abbandono"
  \tied-lyric #"Impetuosi venti I nostri~affetti sono"
  \tied-lyric #"Ogni diletto~e scoglio Tutta la vita~e~un mar."
}
```

Siam navi‿all'onde‿algenti Lasciate‿in abbandono
Impetuosi venti I nostri‿affetti sono
Ogni diletto‿e scoglio Tutta la vita‿e‿un mar.

Used properties:

- `word-space`

A.10.5 Instrument Specific Markup

`\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s:`*number* – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t:`*number* – Set the line thickness (relative to normal line thickness). Default: 0.5.
  - `h:`*number* – Set the height of the diagram in frets. Default: 4.
  - `w:`*number* – Set the width of the diagram in strings. Default: 6.
  - `f:`*number* – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
  - `d:`*number* – Set radius of dot, in terms of fret spacing. Default: 0.25.
  - `p:`*number* – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
  - `c:`*string1*`-`*string2*`-`*fret* – Include a barre mark from *string1* to *string2* on *fret*.
  - *string*`-`*fret* – Place a dot on *string* at *fret*. If *fret* is '`o`', *string* is identified as open. If *fret* is '`x`', *string* is identified as muted.
  - *string*`-`*fret*`-`*fingering* – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by '`x`'.
- Open strings are indicated by '`o`'.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.

- Fingerings are given by following the fret number with a `-`, followed by the finger indicator, e.g. '`3-2`' for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with `-(` to start a barre and `-)` to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
  #'((mute 6) (mute 5) (open 4)
     (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute` *string-number*`)`
: Place a small 'x' at the top of string *string-number*.

`(open` *string-number*`)`
: Place a small 'o' at the top of string *string-number*.

`(barre` *start-string end-string fret-number*`)`
: Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(capo` *fret-number*`)`
: Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

`(place-fret` *string-number fret-number* `[`*finger-value* `[`*color-modifier*`]])`
: Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, and an optional color modifier *color-modifier*. By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of the variable *dot-color*. Setting *color-modifier* to `inverted` inverts the dot color for a specific fingering. If the *finger* part of the `place-fret` element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\harp-pedal` *definition-string* (string)

Make a harp pedal diagram.

Possible elements in *definition-string*:

`^`	pedal is up
`-`	pedal is neutral
`v`	pedal is down
`|`	vertical divider line
`o`	the following pedal should be circled (indicating a change)

The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you'll have to live with the warnings.

The appearance of the diagram can be tweaked inter alia using the size property of the TextScript grob (`\override Voice.TextScript #'size = #0.3`) for the overall, the thickness property (`\override Voice.TextScript #'thickness = #3`) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the harp-pedal-details list of properties (`\override Voice.TextScript #'harp-pedal-details #'box-width = #1`). It contains the following settings: `box-offset` (vertical shift of the box center for up/down pedals), `box-width`, `box-height`, `space-before-divider` (the spacing between two boxes before the divider) and `space-after-divider` (box spacing after the divider).

```
\markup \harp-pedal #"^-v|--ov^"
```



Used properties:

- `thickness` (0.5)
- `harp-pedal-details` ('())
- `size` (1.2)

`\woodwind-diagram` *instrument* (symbol) *user-draw-commands* (list)

Make a woodwind-instrument diagram. For example, say

```
\markup \woodwind-diagram
  #'oboe #'((lh . (d ees)) (cc . (five3qT1q)) (rh . (gis)))
```

for an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed while the five-hole of the central column effectuates a trill between 1/4 and 3/4 closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function `(print-keys 'instrument)` in your .ly file, where instrument is the instrument whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

- 1q (1/4 covered)
- 1h (1/2 covered)
- 3q (3/4 covered)
- R (ring depressed)
- F (fully covered; the default if no state put)

Additionally, these configurations can be used in trills. So, for example, `three3qTR` effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, `threeRT` effectuates a trill between R and open, whereas `threeTR` effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke `(print-keys-verbose 'instrument)`.

Lastly, substituting an empty list for the pressed-key alist will result in a diagram with all of the keys drawn but none filled, for example:

```
\markup \woodwind-diagram #'oboe #'()
```

Used properties:

- `graphical` (#t)
- `thickness` (0.1)
- `size` (1)

A.10.6 Accordion Registers

`\discant` *name* (string)

`\discant` ***name*** generates a discant accordion register symbol.

To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

The register names in the default `\discant` register set have modeled after numeric Swiss notation like depicted in `http://de.wikipedia.org/wiki/Register_%28Akkordeon%29`, omitting the slashes and dropping leading zeros.

The string *name* is basically a three-digit number with the lowest digit specifying the number of 16' reeds, the tens the number of 8' reeds, and the hundreds specifying the number of 4' reeds. Without modification, the specified number of reeds in 8' is centered in the symbol. Newer instruments may have registrations where 8' can be used either within or without a tone chamber, 'cassotto'. Notationally, the central dot then indicates use of cassotto. One can suffix the tens' digits '`1`' and '`2`' with '`+`' or '`-`' to indicate clustering the dots at the right or left respectively rather than centered.

Some examples are



\discant #"1" \discant #"1+0"



\discant #"120" \discant #"131"

Used properties:

- `font-size` (0)

`\freeBass` *name* (string)

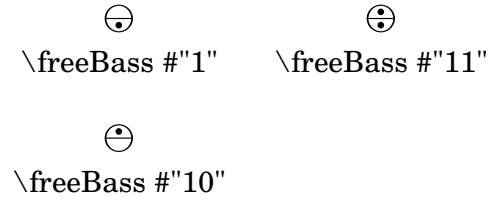
`\freeBass` ***name*** generates a free bass/converter accordion register symbol for the usual two-reed layout.

To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

Available registrations are



Used properties:

- `font-size` (0)

`\stdBass` *name* (string)

`\stdBass` ***name*** generates a standard bass accordion register symbol.

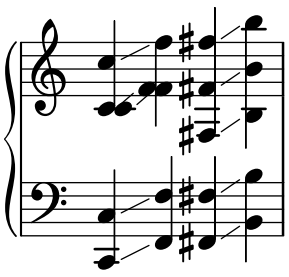
To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

The default bass register definitions have been modeled after the article http://www.accordions.com/index/art/stradella.shtml originally appearing in Accord Magazine.

The underlying register model is



This kind of overlapping arrangement is common for Italian instruments though the exact location of the octave breaks differ.

When not composing for a particular target instrument, using the five reed definitions makes more sense than using a four reed layout: in that manner, the '`Master`' register is unambiguous. This is rather the rule in literature bothering about bass registrations at all.

Available registrations are



\stdBass #"Soprano"



\stdBass #"Soft Bass"



\stdBass #"Alto"



\stdBass #"Soft Tenor"



\stdBass #"Tenor"



\stdBass #"Bass/Alto"



\stdBass #"Master"

Used properties:

- `font-size` (0)

`\stdBassIV` *name* (string)
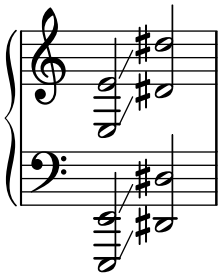
`\stdBassIV` ***name*** generates a standard bass accordion register symbol.

To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

The main use is for four-reed standard bass instruments with reedbank layout



Notable instruments are Morino models with MIII (the others are five-reed instead) and the Atlantic IV. Most of those models have three register switches. Some newer Morinos with MIII might have five or even seven.

The prevalent three-register layout uses the middle three switches '`Tenor`', '`Master`', '`Soft Bass`'. Note that the sound is quite darker than the same registrations of '`c,`'-based instruments.

Available registrations are



\stdBassIV #"Soprano"



\stdBassIV #"Soft Bass"



\stdBassIV #"Alto"



\stdBassIV #"Bass/Alto"



\stdBassIV #"Tenor"



\stdBassIV #"Soft Bass/Alto"



\stdBassIV #"Master"



\stdBassIV #"Soft Tenor"

Used properties:

- `font-size` (0)

`\stdBassV` *name* (string)
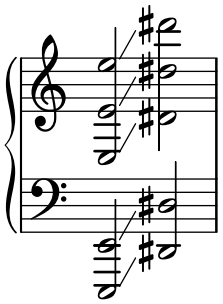
`\stdBassV` *`name`* generates a standard bass accordion register symbol.

To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

The main use is for five-reed standard bass instruments with reedbank layout



This tends to be the bass layout for Hohner's Morino series without convertor or MIII manual.

With the exception of the rather new 7-register layout, the highest two chord reeds are usually sounded together. The Older instruments offer 5 or 3 bass registers. The Tango VM offers an additional '`Solo Bass`' setting that mutes the chord reeds. The symbol on the register buttons of the Tango VM would actually match the physical five-octave layout reflected here, but it is not used in literature.

Composers should likely prefer the five-reed versions of these symbols. The mismatch of a four-reed instrument with five-reed symbols is easier to resolve for the player than the other way round.

Available registrations are



\stdBassV #"Bass/Alto"



\stdBassV #"Soft Bass"



\stdBassV #"Soft Bass/Alto"



\stdBassV #"Soft Tenor"



\stdBassV #"Alto"



\stdBassV #"Soprano"



\stdBassV #"Tenor"



\stdBassV #"Sopranos"



\stdBassV #"Master"



\stdBassV #"Solo Bass"

Used properties:

- `font-size` (0)

`\stdBassVI` *name* (string)
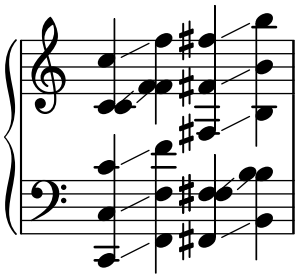
`\stdBassVI` ***name*** generates a standard bass accordion register symbol for six reed basses.

To make it available,

```
#(use-modules (scm accreg))
```

is required near the top of your input file.

This is primarily the register layout for the Hohner “Gola” model. The layout is



The registers are effectively quite similar to that of `\stdBass`. An additional bass reed at alto pitch is omitted for esthetical reasons from the ‘`Master`’ setting, so the symbols are almost the same except for the ‘`Alto/Soprano`’ register with bass notes at Alto pitch and chords at Soprano pitch.

Available registrations are



\stdBassVI #"Soprano"



\stdBassVI #"Alto/Soprano"



\stdBassVI #"Alto"



\stdBassVI #"Bass/Alto"



\stdBassVI #"Soft Tenor"



\stdBassVI #"Soft Bass"



\stdBassVI #"Master"

Used properties:

- `font-size` (0)

A.10.7 Other

`\auto-footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a c

The footnote will be annotated automatically.

Used properties:

- `padding` (0.0)
- `raise` (0.5)

`\backslashed-digit` *num* (integer)

A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```


Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\char` *num* (integer)

Produce a single character. Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```

A ©

`\eyeglasses`
: Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```



`\footnote` *mkup* (markup) *note* (markup)
: Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a c

The footnote will not be annotated automatically.

`\fraction` *arg1* (markup) *arg2* (markup)
: Make a fraction of two markups.

```
\markup {

  \fraction 355 113
}
```

$\pi \approx \frac{355}{113}$

Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)
: Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
  \null
}
```

from *myTitle*

`\left-brace` *size* (number)

A feta brace in point size *size*.

```
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}
```



`\lookup` *glyph-name* (string)

Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```



`\markalphabet` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

H AA

`\markletter` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

J AB

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (procedure) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list. Properties may be any property supported by Sección "font-interface" in *Referencia de Funcionamiento Interno*, Sección "text-interface" in *Referencia de Funcionamiento Interno* and Sección "instrument-specific-markup-interface" in *Referencia de Funcionamiento Interno*.

```
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
      \column {
        increased
        baseline-skip
      }
    }
  }
}
```

default
baseline-skip

increased
baseline-skip

`\page-link` *page-number* (number) *arg* (markup)

Add a link to the page *page-number* around *arg*. This only works in the PDF backend.

```
\markup {
  \page-link #2  { \italic { This links to page 2... } }
}
```

This links to page 2...

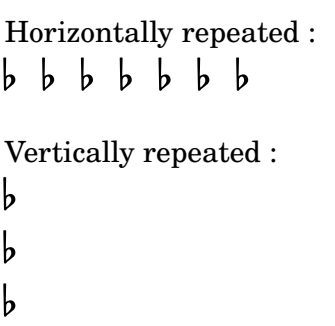
`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\pattern` *count* (integer) *axis* (integer) *space* (number) *pattern* (markup)

Prints *count* times a *pattern* markup. Patterns are spaced apart by *space*. Patterns are distributed on *axis*.

```
\markup \column {
  "Horizontally repeated :"
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated :"
  \pattern #3 #Y #0.5 \flat
}
```



`\property-recursive` *symbol* (symbol)

Print out a warning when a header field markup contains some recursive markup definition.

`\right-brace` *size* (number)

A feta brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```



`\slashed-digit` *num* (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```



Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil` *stil* (stencil)
: Use a stencil as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`
: Create a box of the same height as the space in the current font.

`\transparent` *arg* (markup)
: Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

`\verbatim-file` *name* (string)
: Read the contents of file *name*, and include it verbatim.

```
\markup {
  \verbatim-file #"simple.ly"
}
```

```
%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly.
%% May be omitted.
\version "2.16.0"
```

`\whiteout` *arg* (markup)
: Provide a white background for *arg*.

```
\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}
```



`\with-color` *color* (color) *arg* (markup)
: Draw *arg* in color specified by *color*.

```
\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}
```

red green blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.

`\with-link` *label* (symbol) *arg* (markup)
Add a link to the page holding label *label* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-link #'label {
    \italic { This links to the page containing the label... }
  }
}
```

This links to the page containing the label...

A.11 Instrucciones de listas de marcado de texto

Se puede usar cualquiera de las siguientes instrucciones con `\markuplist`:

`\column-lines` *args* (markup list)
Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (markup list)
Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\map-markup-commands` *compose* (procedure) *args* (markup list)
This applies the function *compose* to every markup in *args* (including elements of markup list command calls) in order to produce a new markup list. Since the return value from a markup list command call is not a markup list but rather a list of stencils, this requires passing those stencils off as the results of individual markup calls. That way, the results should work out as long as no markups rely on side effects.

`\override-lines` *new-prop* (pair) *args* (markup list)
: Like `\override`, for markup lists.

`\table-of-contents`

`\wordwrap-internal` *justify* (boolean) *args* (markup list)
: Internal markup list command used to define `\justify` and `\wordwrap`.

  Used properties:

  - `text-direction` (1)
  - `word-space`
  - `line-width` (#f)

`\wordwrap-lines` *args* (markup list)
: Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

  Used properties:

  - `text-direction` (1)
  - `word-space`
  - `line-width` (#f)
  - `baseline-skip`

`\wordwrap-string-internal` *justify* (boolean) *arg* (string)
: Internal markup list command used to define `\justify-string` and `\wordwrap-string`.

  Used properties:

  - `text-direction` (1)
  - `word-space`
  - `line-width`

A.12 Lista de caracteres especiales

Se pueden usar las siguientes referencias a carcteres especiales; para ver más detalles, consulte [Alias de ASCII], página 499.

Se usa la sintaxis del HTML y casi todas estas referencias son las mismas que en el HTML. El resto están inspiradas en LaTeX.

Los caracteres están enmarcados en un rectángulo de forma que pueda verse el tamaño que tienen. Se ha añadido un pequeño relleno de separación entre el carácter y el rectángulo para mejorar la legibilidad.

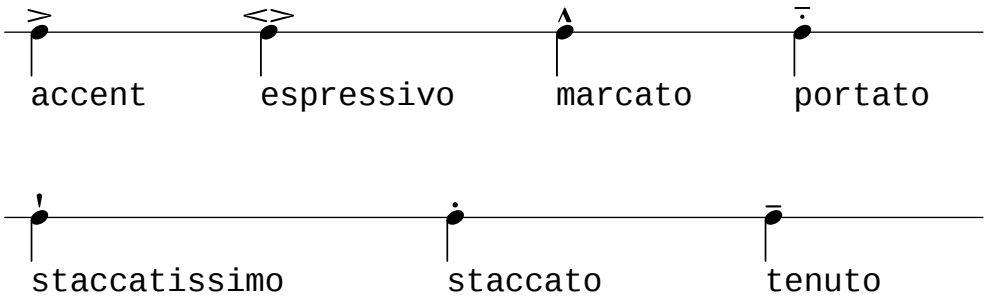
…	…	–	–	—	—	¡	¡		
¿	¿	&solidus;	⁄	&flq;	‹	&frq;	›		
&flqq;	«	&frqq;	»	&glq;	‚	&grq;	‘		
&glqq;	„	&grqq;	“	&elq;	‘	&erq;	’		
&elqq;	“	&erqq;	”						

				&nnbsp;		‍			
‌		·	·	•	•	&copyright;	©		
&registered;	®	&trademark;	™	†	†	‡	‡		
№	№	ª	ª	º	º	¶	¶		
§	§	°	°	№	№	‰	‰		
¦	¦	´	´	&acutedbl;	˝	`	`		
˘	˘	ˇ	ˇ	&cedilla;	¸	&circumflex;	^		
&diaeresis;	¨	&macron;	¯	&aa;	å	&AA;	Å		
&ae;	æ	&AE;	Æ	&dh;	ð	&DH;	Ð		
&dj;	đ	&DJ;	Đ	&l;	ł	&L;	Ł		
&ng;	ŋ	&NG;	Ŋ	&o;	ø	&O;	Ø		
&oe;	œ	&OE;	Œ	&s;	ß	&ss;	ß		
&th;	þ	&TH;	Þ	+	+	−	−		
×	×	÷	÷	¹	¹	²	²		
³	³	&sqrt;	√	&increment;	Δ	&infty;	∞		
∑	∑	±	±	&bulletop;	∙	&partial;	∂		
&neg;	¬	&currency;	¤	$	$	€	€		
&pounds;	£	¥	¥	¢	¢				

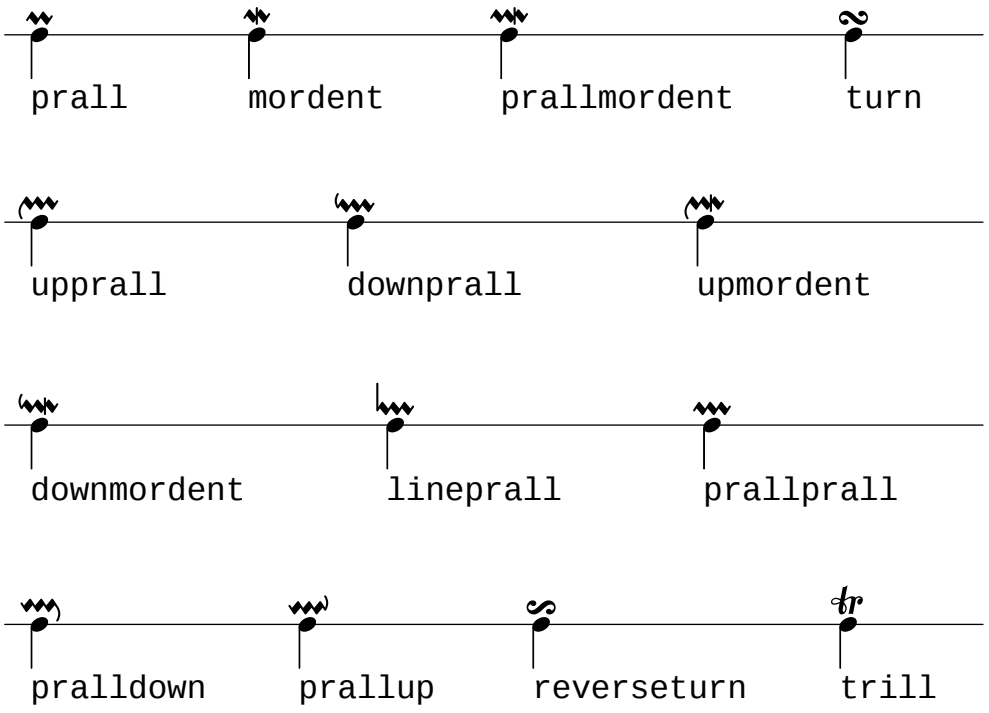
A.13 Lista de articulaciones

Las siguientes inscripciones están disponibles en la tipografía Feta y se pueden adjuntar a las notas (p.ej. '`c\accent`').

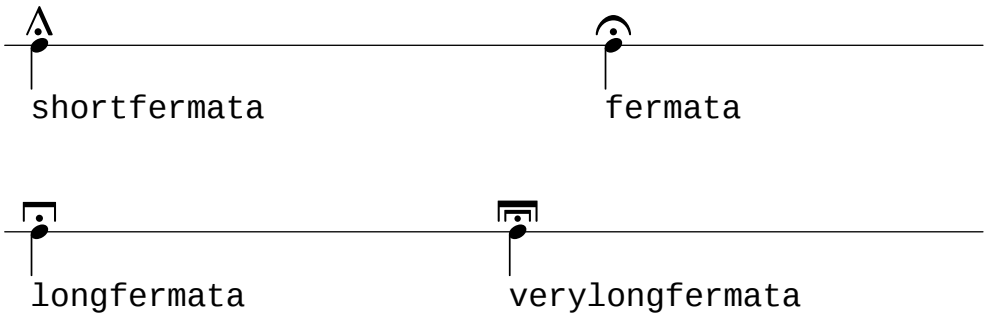
Indicaciones de articulación



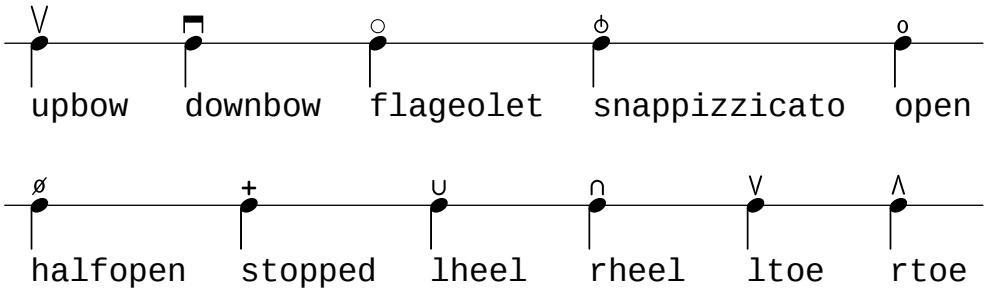
Indicaciones de adornos



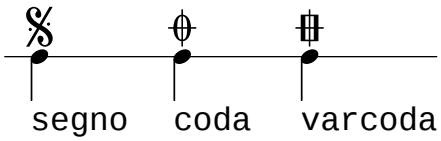
Indicaciones de calderón



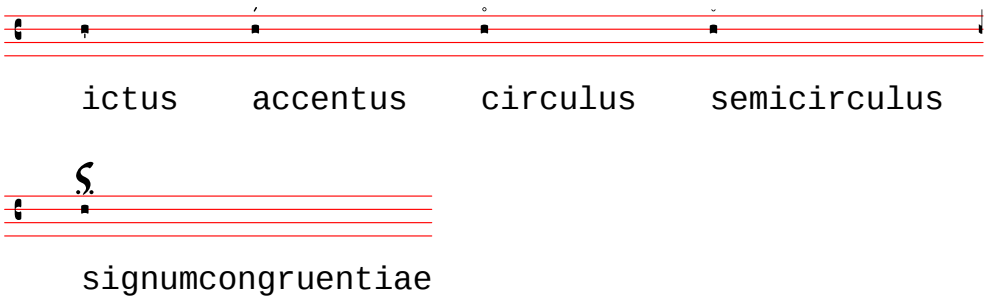
Indicaciones específicas de ciertos instrumentos



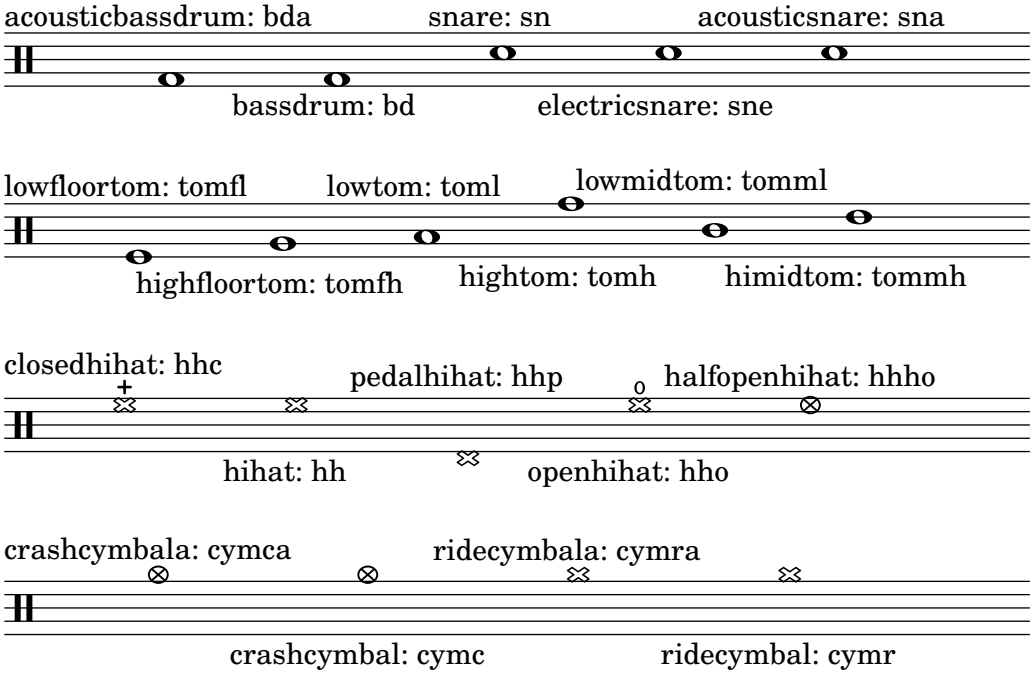
Indicaciones de repetición

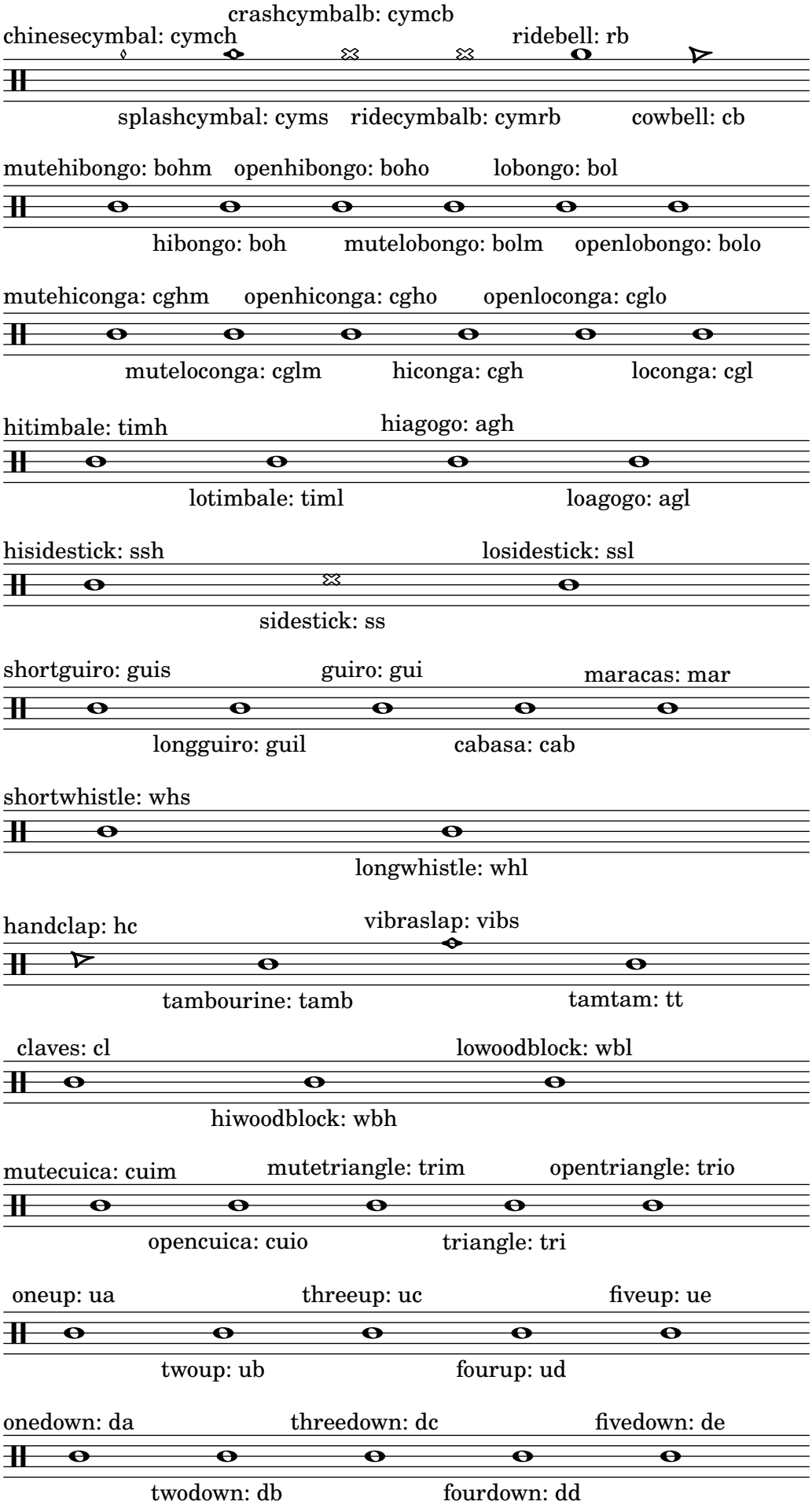


Indicaciones antiguas



A.14 Notas de percusión




crashcymbalb: cymcb
chinesecymbal: cymch
ridebell: rb
splashcymbal: cyms
ridecymbalb: cymrb
cowbell: cb
mutehibongo: bohm
openhibongo: boho
lobongo: bol
hibongo: boh
mutelobongo: bolm
openlobongo: bolo
mutehiconga: cghm
openhiconga: cgho
openloconga: cglo
muteloconga: cglm
hiconga: cgh
loconga: cgl
hitimbale: timh
hiagogo: agh
lotimbale: timl
loagogo: agl
hisidestick: ssh
losidestick: ssl
sidestick: ss
shortguiro: guis
guiro: gui
maracas: mar
longguiro: guil
cabasa: cab
shortwhistle: whs
longwhistle: whl
handclap: hc
vibraslap: vibs
tambourine: tamb
tamtam: tt
claves: cl
lowoodblock: wbl
hiwoodblock: wbh
mutecuica: cuim
mutetriangle: trim
opentriangle: trio
opencuica: cuio
triangle: tri
oneup: ua
threeup: uc
fiveup: ue
twoup: ub
fourup: ud
onedown: da
threedown: dc
fivedown: de
twodown: db
fourdown: dd

A.15 Glosario técnico

Glosario de los términos técnicos y conceptos que se utilizan internamente en LilyPond. Estos términos pueden aparecer en los manuales, en las listas de distribución de correo o en el código fuente.

alist (lista-A)

Una lista asociativa o abreviadamente una **lista-A** (alist en inglés) es una pareja de Scheme que asocia un valor con una clave: `(clave . valor)`. Por ejemplo, en '`scm/lily.scm`', la lista-A "type-p-name-alist" asocia ciertos predicadps de tipo (p.ej. `ly:music?`) con nombres (p.ej. "music") de forma que se pueda informar de los fallos de comprobación de tipo con un mensaje de consola que incluye el nombre del predicado de tipo esperado.

callback

Una **callback** es una rutina, función o método cuya referencia se pasa como argumento en una llamada a otra rutina, permitiendo así que la runtina llamada invoque a aquélla. La técnica permite que una capa de software de nivel más bajo llame a una función definida en una capa de nivel más alto. Las funciones de callback se usan ampliamente en LilyPond para permitir al código de Scheme del nivel de usuario definir cuántas acciones de bajo nivel se llevan a cabo.

closure (cerradura)

En Scheme, se crea una **cerradura** cuando una función, por lo general una expresión lambda, se pasa como variable. La cerradura contiene el codigo de la función y referencias a las ligaduras léxicas de las variables libres de la función (es decir, las variables que se usan en la expresión pero se definen fuera de ella). Cuando más tarde se aplica esta función a diferentes argumentos, las ligaduras de variables libres que se capturaron dentro de la cerradura se utilizan para obtener los valores de las variables libres que se usarán en el cálculo. Una propiedad útil de las cerraduras es la retención de los valores internos de las variables de una invocación a otra, permitiendo así que se pueda mantener un estado.

Una **cerradura simple** es una cerradura cuya expresión no tiene variables libres y por ello no tiene ligaduras de variables libres.

Una cerradura simple se representa en LilyPond mediante un 'smob' que contiene la expresión y un método para aplicar la expresión a la lista de argumentos que se le pasa.

glifo

Un **glifo** es una representación gráfica particular de un carácter tipográfico, o una combinación de dos caracteres que forman una ligadura. Un conjunto de glifos con un estilo y forma uniformes forman una fuente tipográfica, y un conjunto de fuentes tipográficas que abarcan varios estilos forman un tipo.

Véase también

Referencia de la notación: Sección 1.8.3 [Tipografías], página 246, Sección 3.3.3 [Caracteres especiales], página 497.

grob (objeto gráfico)

Los objetos de LilyPond que representan elementos de la notación en la salida impresa tales como la cabeza y la plica de las notas, ligaduras de unión y de expresión, digitaciones, claves, et. se denominan 'objetos de presentación', a menudo conocidos como 'OBjetos GRáficos', o abreviadamente **grobs**. Se representan mediante instancias de la clase `Grob`.

Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” in *Manual de Aprendizaje*, Sección “Convenciones de nombres de objetos y propiedades” in *Manual de Aprendizaje*, Sección “Propiedades de los objetos de presentación” in *Manual de Aprendizaje*.

Referencia de funcionamiento interno: Sección “grob-interface” in *Referencia de Funcionamiento Interno*, Sección “All layout objects” in *Referencia de Funcionamiento Interno*.

inmutable

Un objeto **inmutable** es aquél cuyo estado no se puede modificar después de su creación, en contraste con los objetos mutables, que se pueden modificar después de su creación.

En LilyPond, las propiedades inmutables o compartidas definen el estilo y comportamiento predeterminados de los grobs. Se comparten por parte de muchos objetos. En aparente contradicción con su nombre, se pueden cambiar utilizando `\override` y `\revert`.

Véase también

Referencia de la notación: [mutable], página 729.

interfaz

Las acciones y propiedades comunes a un conjunto de grobs se agrupan en un objeto denominado `interfaz de grob (grob-inerface)`, o abreviadamente ‘interfaz’.

Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” in *Manual de Aprendizaje*, Sección “Convenciones de nombres de objetos y propiedades” in *Manual de Aprendizaje*, Sección “Propiedades de los interfaces” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2.2 [Interfaces de la presentación], página 588.

Referencia de funcionamiento interno: Sección “Graphical Object Interfaces” in *Referencia de Funcionamiento Interno*.

lexer (analizador léxico)

Un **lexer** o analizador léxico es un programa que convierte una secuencia de caracteres en una secuencia de elementos o tokens, en un proceso que se llama análisis léxico. El analizador léxico de LilyPond convierte el flujo obtenido a partir de un archivo de entrada ‘`.ly`’ en un flujo descompuesto en tokens más apto para la siguiente fase del procesado: el análisis sintáctico, véase [parser (analizador sintáctico)], página 730. El analizador léxico de LilyPond lexer está construido con la herramienta Flex a partir del archivo de lexer ‘`lily/lexer.ll`’ que contiene las reglas léxicas. Este archivo es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

mutable

Un objeto **mutable** es aquél cuyo estado se puede modificar después de su creación, en contraste con un objeto inmutable, cuyo estado se fija en el momento de la creación.

En LilyPond, las propiedades mutables contienen valores específicos de un grob. Por lo general, las listas de otros objetos o los resultados de los cálculos se almacenan en propiedades mutables.

Véase también

Referencia de la notación: [inmutable], página 729.

output-def (definición de salida)

Una instancia de la clase `Output-def` contiene los métodos y estructuras de datos asociados con un bloque de salida. Se crean instancias parra los bloques midi, layout y paper.

parser (analizador sintáctico)

Un **parser** o analizador sintáctico analiza la secuencia de tokens o elementos léxicos producida por un analizador léxico para determinar su estructura gramatical, agrupando los elementos léxicos en conjuntos mayores según las reglas de la gramática. Si la secuencia de elementos léxicos es válida, el producto final es un árbol de tokens cuya raíz es el símbolo inicial de la gramática. Si no se puede conseguir esto, el archivo es inválido y se produce un mensaje de error adecuado. Las agrupaciones sintácticas y las reglas para construir estas agrupaciones a partir de sus elementos constituyentes para la sintaxis de LilyPond están definidas en '`lily/parser.yy`' y se muestran en la forma normal de Backus (BNF) dentro de Sección "LilyPond grammar" in *Guía del colaborador*. Este archivo se usa para construir el analizador sintáctico durante la compilación del programa por parte del generador de analizadores sintácticos, Bison. Es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

variable del analizador sintáctico

Son variables definidas directamente dentro de Scheme. Su uso directo por parte de los usuarios está fuertemente desaconsejado, porque su semántica de ámbito puede ser confusa.

Cuando el valor de una de estas variables se modifica dentro de un archivo '`.ly`', el cambio es global, y a no ser que se revierta explícitamente, el nuevo valor persistirá hasta el final del archivo, afectando a todos los bloques `\score` así como a los archivos externos añadidos con la instrucción `\include`. Esto puede conducir a consecuencias imprevistas y en proyectos de composición tipográfica complejos puede ser difícil de rastrear.

LilyPond utiliza las siguientes variables del analizador sintáctico:

- afterGraceFraction
- musicQuotes
- mode
- output-count
- output-suffix
- partCombineListener
- pitchnames
- toplevel-bookparts
- toplevel-scores
- showLastLength
- showFirstLength

prob (objeto de propiedades)

Los OBjetos de PRopiedades, o abreviadamente **probs**, son instancias de la clase `Prob`, que es una sencilla clase básica que tiene listas-A de propiedades mutables e inmutables y los métodos para manipularlas. Las clases `Music` y `Stream_event` derivan de `Prob`. También se crean instancias de la clase `Prob` para almacenar el contenido formateado de los grobs del sistema y los bloques de títulos durante el proceso de disposición de la página.

cerradura simple

Véase [closure (cerradura)], página 728.

smob (objeto de Scheme)

Los **Smobs**, u OBjetos de ScheMe, forman parte del mecanismo utilizado por Guile para exportar objetos de C y de C++ al código de Scheme. En LilyPond, se crean smobs a partir de objetos de C++ por medio de macros. Hay dos tipos de objetos smob: los smobs simples, orientados a objetos inmutables simples como números, y los smobs complejos, usados para objetos con identidades. Si tiene acceso a las fuentes de LilyPond sources, encontrará más información en '`lily/includes/smob.hh`'.

stencil (sello)

Las instancias de la clase **stencil** contienen la información necesaria para imprimir un objeto tipográfico. Es un smob simple que contiene una caja de confinamiento, que a su vez define las dimensiones vertical y horizontal del objeto, y una expresión de Scheme que imprime el objeto cuendo se evalúa. Los stencils o sellos se pueden combinar para formar sellos más complejos definidos por un árbol de expresiones de Scheme formado a partir de las expresiones de Scheme de los sellos que lo componen.

La propiedad `stencil`, que conecta a un grob con su sello, se define dentro del interfaz `grob-interface`.

Véase también

Referencia de funcionamiento interno: Sección "grob-interface" in *Referencia de Funcionamiento Interno*.

A.16 Todas las propiedades de contexto

`accidentalGrouping` (symbol)
: If set to `'voice`, accidentals on the same note in different octaves may be horizontally staggered if in different voices.

`additionalPitchPrefix` (string)
: Text with which to prefix additional pitches within a chord name.

`aDueText` (markup)
: Text to print at a unisono passage.

`alignAboveContext` (string)
: Where to insert newly created context in vertical alignment.

`alignBassFigureAccidentals` (boolean)
: If true, then the accidentals are aligned in bass figure context.

`alignBelowContext` (string)
: Where to insert newly created context in vertical alignment.

`alternativeNumberingStyle` (symbol)
: The style of an alternative's bar numbers. Can be `numbers` for going back to the same number or `numbers-with-letters` for going back to the same number with letter suffixes. No setting will not go back in measure-number time.

`associatedVoice` (string)
: Name of the `Voice` that has the melody for this `Lyrics` line.

`autoAccidentals` (list)
: List of different ways to typeset an accidental.

  For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

  Each entry in the list is either a symbol or a procedure.

symbol The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is Sección "Score" in *Referencia de Funcionamiento Interno* then all staves share accidentals, and if *context* is Sección "Staff" in *Referencia de Funcionamiento Interno* then all voices in the same staff share accidentals, but staves do not.

procedure The procedure represents an accidental rule to be applied to the previously specified context.

The procedure takes the following arguments:

`context` The current context to which the rule should be applied.

`pitch` The pitch of the note to be evaluated.

`barnum` The current bar number.

`measurepos`
The current measure position.

The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. `(#t . #f)` does not make sense.

`autoBeamCheck` (procedure)
A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-`#f` return value starts or stops the auto beam.

`autoBeaming` (boolean)
If set to true then beams are generated automatically.

`autoCautionaries` (list)
List similar to `autoAccidentals`, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

`automaticBars` (boolean)
If set to false then bar lines will not be printed automatically; they must be explicitly created with a `\bar` command. Unlike the `\cadenzaOn` keyword, measures are still counted. Bar line generation will resume according to that count if this property is unset.

`barAlways` (boolean)
If set to true a bar line is drawn after each note.

`barCheckSynchronize` (boolean)
If true then reset `measurePosition` when finding a bar check.

`barNumberFormatter` (procedure)
A procedure that takes a bar number, measure position, and alternative number and returns a markup of the bar number to print.

`barNumberVisibility` (procedure)
A procedure that takes a bar number and a measure position and returns whether the corresponding bar number should be printed. Note that the actual print-out of bar numbers is controlled with the `break-visibility` property.

The following procedures are predefined:

`all-bar-numbers-visible`
Enable bar numbers for all bars, including the first one and broken bars (which get bar numbers in parentheses).

`first-bar-number-invisible`
: Enable bar numbers for all bars (including broken bars) except the first one. If the first bar is broken, it doesn't get a bar number either.

`first-bar-number-invisible-save-broken-bars`
: Enable bar numbers for all bars (including broken bars) except the first one. A broken first bar gets a bar number.

`first-bar-number-invisible-and-no-parenthesized-bar-numbers`
: Enable bar numbers for all bars except the first bar and broken bars. This is the default.

`(every-nth-bar-number-visible` *n*`)`
: Assuming *n* is value 2, for example, this enables bar numbers for bars 2, 4, 6, etc.

`(modulo-bar-number-visible` *n m*`)`
: If bar numbers 1, 4, 7, etc., should be enabled, *n* (the modulo) must be set to 3 and *m* (the division remainder) to 1.

`baseMoment` (moment)
: Smallest unit of time that will stand on its own as a subdivided section.

`bassFigureFormatFunction` (procedure)
: A procedure that is called to produce the formatting for a `BassFigure` grob. It takes a list of `BassFigureEvent`s, a context, and the grob to format.

`bassStaffProperties` (list)
: An alist of property settings to apply for the down staff of `PianoStaff`. Used by `\autochange`.

`beamExceptions` (list)
: An alist of exceptions to autobeam rules that normally end on beats.

`beamHalfMeasure` (boolean)
: Whether to allow a beam to begin halfway through the measure in triple time, which could look like 6/8.

`beatStructure` (list)
: List of `baseMoment`s that are combined to make beats.

`chordChanges` (boolean)
: Only show changes in chords scheme?

`chordNameExceptions` (list)
: An alist of chord exceptions. Contains `(`*chord* `.` *markup*`)` entries.

`chordNameExceptionsFull` (list)
: An alist of full chord exceptions. Contains `(`*chord* `.` *markup*`)` entries.

`chordNameExceptionsPartial` (list)
: An alist of partial chord exceptions. Contains `(`*chord* `. (`*prefix-markup suffix-markup*`))` entries.

`chordNameFunction` (procedure)
: The function that converts lists of pitches to chord names.

`chordNameLowercaseMinor` (boolean)
: Downcase roots of minor chords?

`chordNameSeparator` (markup)
: The markup object used to separate parts of a chord name.

`chordNoteNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for single pitches.

`chordPrefixSpacer` (number)
: The space added between the root symbol and the prefix of a chord name.

`chordRootNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for chords.

`clefGlyph` (string)
: Name of the symbol within the music font.

`clefPosition` (number)
: Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`clefTransposition` (integer)
: Add this much extra transposition. Values of 7 and -7 are common.

`clefTranspositionFormatter` (procedure)
: A procedure that takes the Transposition number as a string and the style as a symbol and returns a markup.

`clefTranspositionStyle` (symbol)
: Determines the way the ClefModifier grob is displayed. Possible values are '`default`', '`parenthesized`' and '`bracketed`'.

`completionBusy` (boolean)
: Whether a completion-note head is playing.

`completionUnit` (moment)
: Sub-bar unit of completion.

`connectArpeggios` (boolean)
: If set, connect arpeggios across piano staff.

`countPercentRepeats` (boolean)
: If set, produce counters for percent repeats.

`createKeyOnClefChange` (boolean)
: Print a key signature whenever the clef is changed.

`createSpacing` (boolean)
: Create `StaffSpacing` objects? Should be set for staves.

`crescendoSpanner` (symbol)
: The type of spanner to be used for crescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin crescendo is used.

`crescendoText` (markup)
: The text to print at start of non-hairpin crescendo, i.e., '`cresc.`'.

`cueClefGlyph` (string)
: Name of the symbol within the music font.

`cueClefPosition` (number)
: Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`cueClefTransposition` (integer)
: Add this much extra transposition. Values of 7 and -7 are common.

`cueClefTranspositionFormatter` (procedure)
: A procedure that takes the Transposition number as a string and the style as a symbol and returns a markup.

`cueClefTranspositionStyle` (symbol)
: Determines the way the ClefModifier grob is displayed. Possible values are '`default`', '`parenthesized`' and '`bracketed`'.

`currentBarNumber` (integer)
: Contains the current barnumber. This property is incremented at every bar line.

`decrescendoSpanner` (symbol)
: The type of spanner to be used for decrescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin decrescendo is used.

`decrescendoText` (markup)
: The text to print at start of non-hairpin decrescendo, i.e., '`dim.`'.

`defaultBarType` (string)
: Set the default type of bar line. See `whichBar` for information on available bar types.

  This variable is read by Sección "Timing_translator" in *Referencia de Funcionamiento Interno* at Sección "Score" in *Referencia de Funcionamiento Interno* level.

`defaultStrings` (list)
: A list of strings to use in calculating frets for tablatures and fretboards if no strings are provided in the notes for the current moment.

`doubleRepeatSegnoType` (string)
: Set the default bar line for the combinations double repeat with segno. Default is '`:|.S.|:`'.

`doubleRepeatType` (string)
: Set the default bar line for double repeats.

`doubleSlurs` (boolean)
: If set, two slurs are created for every slurred note, one above and one below the chord.

`drumPitchTable` (hash table)
: A table mapping percussion instruments (symbols) to pitches.

`drumStyleTable` (hash table)
: A hash table which maps drums to layout settings. Predefined values: '`drums-style`', '`timbales-style`', '`congas-style`', '`bongos-style`', and '`percussion-style`'.

  The layout style is a hash table, containing the drum-pitches (e.g., the symbol '`hihat`') as keys, and a list `(notehead-style script vertical-position)` as values.

`endRepeatSegnoType` (string)
: Set the default bar line for the combinations ending of repeat with segno. Default is '`:|.S`'.

`endRepeatType` (string)
: Set the default bar line for the ending of repeats.

`explicitClefVisibility` (vector)
: '`break-visibility`' function for clef changes.

`explicitCueClefVisibility` (vector)
: '`break-visibility`' function for cue clef changes.

`explicitKeySignatureVisibility` (vector)
: '`break-visibility`' function for explicit key changes. '`\override`' of the `break-visibility` property will set the visibility for normal (i.e., at the start of the line) key signatures.

`extendersOverRests` (boolean)
: Whether to continue extenders as they cross a rest.

`extraNatural` (boolean)
: Whether to typeset an extra natural sign before accidentals that reduce the effect of a previous alteration.

`figuredBassAlterationDirection` (direction)
: Where to put alterations relative to the main figure.

`figuredBassCenterContinuations` (boolean)
: Whether to vertically center pairs of extender lines. This does not work with three or more lines.

`figuredBassFormatter` (procedure)
: A routine generating a markup for a bass figure.

`figuredBassPlusDirection` (direction)
: Where to put plus signs relative to the main figure.

`fingeringOrientations` (list)
: A list of symbols, containing '`left`', '`right`', '`up`' and/or '`down`'. This list determines where fingerings are put relative to the chord being fingered.

`firstClef` (boolean)
: If true, create a new clef when starting a staff.

`followVoice` (boolean)
: If set, note heads are tracked across staff switches by a thin line.

`fontSize` (number)
: The relative size of all grobs in a context.

`forbidBreak` (boolean)
: If set to `#t`, prevent a line break at this point.

`forceClef` (boolean)
: Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.

`fretLabels` (list)
: A list of strings or Scheme-formatted markups containing, in the correct order, the labels to be used for lettered frets in tablature.

`glissandoMap` (list)
: A map in the form of '((source1 . target1) (source2 . target2) (sourcen . targetn)) showing the glissandi to be drawn for note columns. The value '() will default to '((0 . 0) (1 . 1) (n . n)), where n is the minimal number of note-heads in the two note columns between which the glissandi occur.

`gridInterval` (moment)
: Interval for which to generate `GridPoint`s.

`handleNegativeFrets` (symbol)
: How the automatic fret calculator should handle calculated negative frets. Values include `'ignore`, to leave them out of the diagram completely, `'include`, to include

them as calculated, and `'recalculate`, to ignore the specified string and find a string where they will fit with a positive fret number.

`harmonicAccidentals` (boolean)
: If set, harmonic notes in chords get accidentals.

`harmonicDots` (boolean)
: If set, harmonic notes in dotted chords get dots.

`highStringOne` (boolean)
: Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.

`ignoreBarChecks` (boolean)
: Ignore bar checks.

`ignoreFiguredBassRest` (boolean)
: Don't swallow rest events.

`ignoreMelismata` (boolean)
: Ignore melismata for this Sección "Lyrics" in *Referencia de Funcionamiento Interno* line.

`implicitBassFigures` (list)
: A list of bass figures that are not printed as numbers, but only as extender lines.

`implicitTimeSignatureVisibility` (vector)
: break visibility for the default time signature.

`includeGraceNotes` (boolean)
: Do not ignore grace notes for Sección "Lyrics" in *Referencia de Funcionamiento Interno*.

`instrumentCueName` (markup)
: The name to print if another instrument is to be taken.

`instrumentEqualizer` (procedure)
: A function taking a string (instrument name), and returning a `(min . max)` pair of numbers for the loudness range of the instrument.

`instrumentName` (markup)
: The name to print left of a staff. The `instrumentName` property labels the staff in the first system, and the `shortInstrumentName` property labels following lines.

`instrumentTransposition` (pitch)
: Define the transposition of the instrument. Its value is the pitch that sounds when the instrument plays written middle C. This is used to transpose the MIDI output, and `\quote`s.

`internalBarNumber` (integer)
: Contains the current barnumber. This property is used for internal timekeeping, among others by the `Accidental_engraver`.

`keepAliveInterfaces` (list)
: A list of symbols, signifying grob interfaces that are worth keeping a staff with `remove-empty` set around for.

`keyAlterationOrder` (list)
: An alist that defines in what order alterations should be printed. The format is `(step . alter)`, where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

`keySignature` (list)
: The current key signature. This is an alist containing `(step . alter)` or `((octave . step) . alter)`, where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. `keySignature = #`((6 . ,FLAT))`.

`lyricMelismaAlignment` (number)
: Alignment to use for a melisma syllable.

`majorSevenSymbol` (markup)
: How should the major 7th be formatted in a chord name?

`markFormatter` (procedure)
: A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.

`maximumFretStretch` (number)
: Don't allocate frets further than this from specified frets.

`measureLength` (moment)
: Length of one measure in the current time signature.

`measurePosition` (moment)
: How much of the current measure have we had. This can be set manually to create incomplete measures.

`melismaBusyProperties` (list)
: A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to `'(melismaBusy beamMelismaBusy)`, only manual melismata and manual beams are considered. Possible values include `melismaBusy`, `slurMelismaBusy`, `tieMelismaBusy`, and `beamMelismaBusy`.

`metronomeMarkFormatter` (procedure)
: How to produce a metronome markup. Called with two arguments: a `TempoChangeEvent` and context.

`middleCClefPosition` (number)
: The position of the middle C, as determined only by the clef. This can be calculated by looking at `clefPosition` and `clefGlyph`.

`middleCCuePosition` (number)
: The position of the middle C, as determined only by the clef of the cue notes. This can be calculated by looking at `cueClefPosition` and `cueClefGlyph`.

`middleCOffset` (number)
: The offset of middle C from the position given by `middleCClefPosition` This is used for ottava brackets.

`middleCPosition` (number)
: The place of the middle C, measured in half staff-spaces. Usually determined by looking at `middleCClefPosition` and `middleCOffset`.

`midiBalance` (number)
: Stereo balance for the MIDI channel associated with the current context. Ranges from -1 to 1, where the values -1 (`#LEFT`), 0 (`#CENTER`) and 1 (`#RIGHT`) correspond to leftmost emphasis, center balance, and rightmost emphasis, respectively.

`midiChannelMapping` (symbol)
: How to map MIDI channels: per `staff` (default), `instrument` or `voice`.

`midiChorusLevel` (number)
: Chorus effect level for the MIDI channel associated with the current context. Ranges from 0 to 1 (0=off, 1=full effect).

`midiInstrument` (string)
: Name of the MIDI instrument to use.

`midiMaximumVolume` (number)
: Analogous to `midiMinimumVolume`.

`midiMergeUnisons` (boolean)
: If true, output only one MIDI note-on event when notes with the same pitch, in the same MIDI-file track, overlap.

`midiMinimumVolume` (number)
: Set the minimum loudness for MIDI. Ranges from 0 to 1.

`midiPanPosition` (number)
: Pan position for the MIDI channel associated with the current context. Ranges from -1 to 1, where the values -1 (`#LEFT`), 0 (`#CENTER`) and 1 (`#RIGHT`) correspond to hard left, center, and hard right, respectively.

`midiReverbLevel` (number)
: Reverb effect level for the MIDI channel associated with the current context. Ranges from 0 to 1 (0=off, 1=full effect).

`minimumFret` (number)
: The tablature auto string-selecting mechanism selects the highest string with a fret at least `minimumFret`.

`minimumPageTurnLength` (moment)
: Minimum length of a rest for a page turn to be allowed.

`minimumRepeatLengthForPageTurn` (moment)
: Minimum length of a repeated section for a page turn to be allowed within that section.

`minorChordModifier` (markup)
: Markup displayed following the root for a minor chord

`noChordSymbol` (markup)
: Markup to be displayed for rests in a ChordNames context.

`noteToFretFunction` (procedure)
: Convert list of notes and list of defined strings to full list of strings and fret numbers. Parameters: The context, a list of note events, a list of tabstring events, and the fretboard grob if a fretboard is desired.

`ottavation` (markup)
: If set, the text for an ottava spanner. Changing this creates a new text spanner.

`output` (music output)
: The output produced by a score-level translator during music interpretation.

`partCombineTextsOnNote` (boolean)
: Print part-combine texts only on the next note rather than immediately on rests or skips.

`pedalSostenutoStrings` (list)
: See `pedalSustainStrings`.

`pedalSostenutoStyle` (symbol)
: See `pedalSustainStyle`.

`pedalSustainStrings` (list)
: A list of strings to print for sustain-pedal. Format is `(up updown down)`, where each of the three is the string to print when this is done with the pedal.

`pedalSustainStyle` (symbol)
: A symbol that indicates how to print sustain pedals: `text`, `bracket` or `mixed` (both).

`pedalUnaCordaStrings` (list)
: See `pedalSustainStrings`.

`pedalUnaCordaStyle` (symbol)
: See `pedalSustainStyle`.

`predefinedDiagramTable` (hash table)
: The hash table of predefined fret diagrams to use in FretBoards.

`printKeyCancellation` (boolean)
: Print restoration alterations before a key signature change.

`printOctaveNames` (boolean)
: Print octave marks for the `NoteNames` context.

`printPartCombineTexts` (boolean)
: Set 'Solo' and 'A due' texts in the part combiner?

`proportionalNotationDuration` (moment)
: Global override for shortest-playing duration. This is used for switching on proportional notation.

`rehearsalMark` (integer)
: The last rehearsal mark printed.

`repeatCommands` (list)
: This property is a list of commands of the form `(list 'volta x)`, where x is a string or `#f`. `'end-repeat` is also accepted as a command.

`repeatCountVisibility` (procedure)
: A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when `countPercentRepeats` is set.

`restCompletionBusy` (boolean)
: Signal whether a completion-rest is active.

`restNumberThreshold` (number)
: If a multimeasure rest has more measures than this, a number is printed.

`restrainOpenStrings` (boolean)
: Exclude open strings from the automatic fret calculator.

`searchForVoice` (boolean)
: Signal whether a search should be made of all contexts in the context hierarchy for a voice to provide rhythms for the lyrics.

`segnoType` (string)
: Set the default bar line for a requested segno. Default is '`S`'.

`shapeNoteStyles` (vector)
: Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

`shortInstrumentName` (markup)
: See `instrumentName`.

`shortVocalName` (markup)
: Name of a vocal line, short version.

`skipBars` (boolean)
: If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.

```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```

`skipTypesetting` (boolean)
: If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.

`slashChordSeparator` (markup)
: The markup object used to separate a chord name from its root note in case of inversions or slash chords.

`soloIIText` (markup)
: The text for the start of a solo for voice ‘two’ when part-combining.

`soloText` (markup)
: The text for the start of a solo when part-combining.

`squashedPosition` (integer)
: Vertical position of squashing for Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*.

`staffLineLayoutFunction` (procedure)
: Layout of staff lines, `traditional`, or `semitone`.

`stanza` (markup)
: Stanza ‘number’ to print before the start of a verse. Use in `Lyrics` context.

`startRepeatSegnoType` (string)
: Set the default bar line for the combinations beginning of repeat with segno. Default is ‘`S.|:`’.

`startRepeatType` (string)
: Set the default bar line for the beginning of repeats.

`stemLeftBeamCount` (integer)
: Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.

`stemRightBeamCount` (integer)
: See `stemLeftBeamCount`.

`strictBeatBeaming` (boolean)
: Should partial beams reflect the beat structure even if it causes flags to hang out?

`stringNumberOrientations` (list)
: See `fingeringOrientations`.

`stringOneTopmost` (boolean)
: Whether the first string is printed on the top line of the tablature.

`stringTunings` (list)
: The tablature strings tuning. It is a list of the pitches of each string (starting with the lowest numbered one).

`strokeFingerOrientations` (list)
: See `fingeringOrientations`.

`subdivideBeams` (boolean)
: If set, multiple beams will be subdivided at `baseMoment` positions by only drawing one beam over the beat.

`suggestAccidentals` (boolean)
: If set, accidentals are typeset as cautionary suggestions over the note.

`systemStartDelimiter` (symbol)
: Which grob to make for the start of the system/staff? Set to `SystemStartBrace`, `SystemStartBracket` or `SystemStartBar`.

`systemStartDelimiterHierarchy` (pair)
: A nested list, indicating the nesting of a start delimiters.

`tablatureFormat` (procedure)
: A function formatting a tablature note head. Called with three arguments: context, string number and, fret number. It returns the text as a markup.

`tabStaffLineLayoutFunction` (procedure)
: A function determining the staff position of a tablature note head. Called with two arguments: the context and the string.

`tempoHideNote` (boolean)
: Hide the note = count in tempo marks.

`tempoWholesPerMinute` (moment)
: The tempo in whole notes per minute.

`tieWaitForNote` (boolean)
: If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.

`timeSignatureFraction` (fraction, as pair)
: A pair of numbers, signifying the time signature. For example, `'(4 . 4)` is a 4/4 time signature.

`timeSignatureSettings` (list)
: A nested alist of settings for time signatures. Contains elements for various time signatures. The element for each time signature contains entries for `baseMoment`, `beatStructure`, and `beamExceptions`.

`timing` (boolean)
: Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

`tonic` (pitch)
: The tonic of the current scale.

`topLevelAlignment` (boolean)
: If true, the *Vertical_align_engraver* will create a *VerticalAlignment*; otherwise, it will create a *StaffGrouper*

`trebleStaffProperties` (list)
: An alist of property settings to apply for the up staff of `PianoStaff`. Used by `\autochange`.

`tremoloFlags` (integer)
: The number of tremolo flags to add if no number is specified.

`tupletFullLength` (boolean)
: If set, the tuplet is printed up to the start of the next note.

`tupletFullLengthNote` (boolean)
: If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

`tupletSpannerDuration` (moment)
: Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```

`useBassFigureExtenders` (boolean)
: Whether to use extender lines for repeated bass figures.

`vocalName` (markup)
: Name of a vocal line.

`voltaSpannerDuration` (moment)
: This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

`whichBar` (string)
: This property is read to determine what type of bar line to create.

  Example:

  ```
  \set Staff.whichBar = ".|:"
  ```

  This will create a start-repeat bar in this staff only. Valid values are described in '`scm/bar-line.scm`'.

A.17 Propiedades de disposición

`add-stem-support` (boolean)
: If set, the `Stem` object is included in this script's support.

`after-line-breaking` (boolean)
: Dummy property, used to trigger callback for `after-line-breaking`.

`align-dir` (direction)
: Which side to align? `-1`: left side, `0`: around center of width, `1`: right side.

`allow-loose-spacing` (boolean)
: If set, column can be detached from main spacing.

`allow-span-bar` (boolean)
: If false, no inter-staff bar line will be created below this bar line.

`alteration` (number)
: Alteration numbers for accidental.

`alteration-alist` (list)
: List of `(`*`pitch . accidental`*`)` pairs for key signature.

`annotation` (string)
: Annotate a grob for debug purposes.

`annotation-balloon` (boolean)
: Print the balloon around an annotation.

`annotation-line` (boolean)
: Print the line from an annotation to the grob that it annotates.

`arpeggio-direction` (direction)
: If set, put an arrow on the arpeggio squiggly line.

`arrow-length` (number)
: Arrow length.

`arrow-width` (number)
: Arrow width.

`auto-knee-gap` (dimension, in staff space)
: If a gap is found between note heads where a horizontal beam fits that is larger than this number, make a kneed beam.

`automatically-numbered` (boolean)
: Should a footnote be automatically numbered?

`average-spacing-wishes` (boolean)
: If set, the spacing wishes are averaged over staves.

`avoid-note-head` (boolean)
: If set, the stem of a chord does not pass through all note heads, but starts at the last note head.

`avoid-scripts` (boolean)
: If set, a tuplet bracket avoids the scripts associated with the note heads it encompasses.

`avoid-slur` (symbol)
: Method of handling slur collisions. Choices are `inside`, `outside`, `around`, and `ignore`. `inside` adjusts the slur if needed to keep the grob inside the slur. `outside` moves the grob vertically to the outside of the slur. `around` moves the grob vertically to the outside of the slur only if there is a collision. `ignore` does not move either. In grobs whose notational significance depends on vertical position (such as accidentals, clefs, etc.), `outside` and `around` behave like `ignore`.

`axes` (list)
: List of axis numbers. In the case of alignment grobs, this should contain only one number.

`bar-extent` (pair of numbers)
: The Y-extent of the actual bar line. This may differ from `Y-extent` because it does not include the dots in a repeat bar line.

`base-shortest-duration` (moment)
: Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.

`baseline-skip` (dimension, in staff space)
: Distance between base lines of multiple lines of text.

`beam-thickness` (dimension, in staff space)
: Beam thickness, measured in `staff-space` units.

`beam-width` (dimension, in staff space)
: Width of the tremolo sign.

`beamed-stem-shorten` (list)
: How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.

`beaming` (pair)
: Pair of number lists. Each number list specifies which beams to make. `0` is the central beam, `1` is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.

`beamlet-default-length` (pair)
: A pair of numbers. The first number specifies the default length of a beamlet that sticks out of the left hand side of this stem; the second number specifies the default length of the beamlet to the right. The actual length of a beamlet is determined by taking either the default length or the length specified by `beamlet-max-length-proportion`, whichever is smaller.

`beamlet-max-length-proportion` (pair)
: The maximum length of a beamlet, as a proportion of the distance between two adjacent stems.

`before-line-breaking` (boolean)
: Dummy property, used to trigger a callback function.

`between-cols` (pair)
: Where to attach a loose column to.

`bound-details` (list)
: An alist of properties for determining attachments of spanners to edges.

`bound-padding` (number)
: The amount of padding to insert around spanner bounds.

`bracket-flare` (pair of numbers)
: A pair of numbers specifying how much edges of brackets should slant outward. Value `0.0` means straight edges.

`bracket-visibility` (boolean or symbol)
: This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to `if-no-beam` makes it print only if there is no beam associated with this tuplet bracket.

`break-align-anchor` (number)
: Grobs aligned to this break-align grob will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.

`break-align-anchor-alignment` (number)
: Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent.

`break-align-orders` (vector)
: Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.

  For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

`break-align-symbol` (symbol)
: This key is used for aligning and spacing breakable items.

`break-align-symbols` (list)
: A list of symbols that determine which break-aligned grobs to align this to. If the grob selected by the first symbol in the list is invisible due to break-visibility, we will align to the next grob (and so on). Choices are `left-edge`, `ambitus`, `breathing-sign`, `clef`, `staff-bar`, `key-cancellation`, `key-signature`, `time-signature`, and `custos`.

`break-overshoot` (pair of numbers)
: How much does a broken spanner stick out of its bounds?

`break-visibility` (vector)
: A vector of 3 booleans, `#(`*`end-of-line unbroken begin-of-line`*`)`. `#t` means visible, `#f` means killed.

`breakable` (boolean)
: Allow breaks here.

`broken-bound-padding` (number)
: The amount of padding to insert when a spanner is broken at a line break.

`circled-tip` (boolean)
: Put a circle at start/end of hairpins (al/del niente).

`clip-edges` (boolean)
: Allow outward pointing beamlets at the edges of beams?

`collapse-height` (dimension, in staff space)
: Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

`collision-bias` (number)
: Number determining how much to favor the left (negative) or right (positive). Larger absolute values in either direction will push a collision in this direction.

`collision-interfaces` (list)
: A list of interfaces for which automatic beam-collision resolution is run.

`collision-padding` (number)
: Amount of padding to apply after a collision is detected via the self-alignment-interface.

`collision-voice-only` (boolean)
: Does automatic beam collsion apply only to the voice in which the beam was created?

`color` (color)
: The color of this grob.

`common-shortest-duration` (moment)
: The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

`concaveness` (number)
: A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

`connect-to-neighbor` (pair)
: Pair of booleans, indicating whether this grob looks as a continued break.

`control-points` (list)
: List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

`count-from` (integer)
: The first measure in a measure count receives this number. The following measures are numbered in increments from this initial value.

`damping` (number)
: Amount of beam slope damping.

`dash-definition` (pair)
: List of `dash-elements` defining the dash structure. Each `dash-element` has a starting t value, an ending t-value, a `dash-fraction`, and a `dash-period`.

`dash-fraction` (number)
: Size of the dashes, relative to `dash-period`. Should be between `0.0` (no line) and `1.0` (continuous line).

`dash-period` (number)
: The length of one dash together with whitespace. If negative, no line is drawn at all.

`default-direction` (direction)
: Direction determined by note head positions.

`default-staff-staff-spacing` (list)
: The settings to use for `staff-staff-spacing` when it is unset, for ungrouped staves and for grouped staves that do not have the relevant `StaffGrouper` property set (`staff-staff-spacing` or `staffgroup-staff-spacing`).

`details` (list)
: Alist of parameters for detailed grob behavior. More information on the allowed parameters for a grob can be found by looking at the top of the Internals Reference page for each interface having a `details` property.

`digit-names` (vector)
: Names for string finger digits.

`direction` (direction)
: If `side-axis` is `0` (or `X`), then this property determines whether the object is placed `LEFT`, `CENTER` or `RIGHT` with respect to the other object. Otherwise, it determines whether the object is placed `UP`, `CENTER` or `DOWN`. Numerical values may also be used: `UP`=`1`, `DOWN`=`-1`, `LEFT`=`-1`, `RIGHT`=`1`, `CENTER`=`0`.

`dot-count` (integer)
: The number of dots.

`dot-negative-kern` (number)
: The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.

`dot-placement-list` (list)
: List consisting of `(`*`description string-number fret-number finger-number`*`)` entries used to define fret diagrams.

`duration-log` (integer)
: The 2-log of the note head duration, i.e., `0` = whole note, `1` = half note, etc.

`eccentricity` (number)
: How asymmetrical to make a slur. Positive means move the center to the right.

`edge-height` (pair)
: A pair of numbers specifying the heights of the vertical edges: `(`*`left-height . right-height`*`)`.

`edge-text` (pair)
: A pair specifying the texts to be set at the edges: `(`*`left-text . right-text`*`)`.

`expand-limit` (integer)
: Maximum number of measures expanded in church rests.

`extra-dy` (number)
: Slope glissandi this much extra.

`extra-offset` (pair of numbers)
: A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in `staff-space` units of the staff's `StaffSymbol`.

`extra-spacing-height` (pair of numbers)
: In the horizontal spacing problem, we increase the height of each item by this amount (by adding the 'car' to the bottom of the item and adding the 'cdr' to the top of the item). In order to make a grob infinitely high (to prevent the horizontal spacing problem from placing any other grobs above or below this grob), set this to `(-inf.0 . +inf.0)`.

`extra-spacing-width` (pair of numbers)
: In the horizontal spacing problem, we pad each item by this amount (by adding the 'car' on the left side of the item and adding the 'cdr' on the right side of the item). In order to make a grob take up no horizontal space at all, set this to `(+inf.0 . -inf.0)`.

`flag-count` (number)
: The number of tremolo beams.

`flat-positions` (list)
: Flats in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: `(alto treble tenor soprano baritone mezzosoprano bass)`. If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.

`font-encoding` (symbol)
: The font encoding is the broadest category for selecting a font. Currently, only lilypond's system fonts (Emmentaler) are using this property. Available values are `fetaMusic` (Emmentaler), `fetaBraces`, `fetaText` (Emmentaler).

`font-family` (symbol)
: The font family is the broadest category for selecting text fonts. Options include: `sans`, `roman`.

`font-name` (string)
: Specifies a file name (without extension) of the font to load. This setting overrides selection using `font-family`, `font-series` and `font-shape`.

`font-series` (symbol)
: Select the series of a font. Choices include `medium`, `bold`, `bold-narrow`, etc.

`font-shape` (symbol)
: Select the shape of a font. Choices include `upright`, `italic`, `caps`.

`font-size` (number)
: The font size, compared to the 'normal' size. `0` is style-sheet's normal size, `-1` is smaller, `+1` is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. Fractional values are allowed.

`footnote` (boolean)
: Should this be a footnote or in-note?

`footnote-music` (music)
: Music creating a footnote.

`footnote-text` (markup)
: A footnote for the grob.

`force-hshift` (number)
: This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by Sección "note-collision-interface" in *Referencia de Funcionamiento Interno*.

`forced-spacing` (number)
: Spacing forced between grobs, used in various ligature engravers.

`fraction` (fraction, as pair)
: Numerator and denominator of a time signature object.

`french-beaming` (boolean)
: Use French beaming style for this stem. The stem stops at the innermost beams.

`fret-diagram-details` (list)
: An alist of detailed grob properties for fret diagrams. Each alist entry consists of a `(`*`property`*` . `*`value`*`)` pair. The properties which can be included in `fret-diagram-details` include the following:

  - `barre-type` – Type of barre indication used. Choices include `curved`, `straight`, and `none`. Default `curved`.
  - `capo-thickness` – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.
  - `dot-color` – Color of dots. Options include `black` and `white`. Default `black`.
  - `dot-label-font-mag` – Magnification for font used to label fret dots. Default value 1.
  - `dot-position` – Location of dot in fret space. Default 0.6 for dots without labels, 0.95-`dot-radius` for dots with labels.
  - `dot-radius` – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
  - `finger-code` – Code for the type of fingering indication used. Options include `none`, `in-dot`, and `below-string`. Default `none` for markup fret diagrams, `below-string` for `FretBoards` fret diagrams.

- `fret-count` – The number of frets. Default 4.
- `fret-label-custom-format` – The format string to be used label the lowest fret number, when `number-type` equals to `custom`. Default `"~a"`.
- `fret-label-font-mag` – The magnification of the font used to label the lowest fret number. Default 0.5.
- `fret-label-vertical-offset` – The offset of the fret label from the center of the fret in direction parallel to strings. Default 0.
- `label-dir` – Side to which the fret label is attached. `-1`, `LEFT`, or `DOWN` for left or down; `1`, `RIGHT`, or `UP` for right or up. Default `RIGHT`.
- `mute-string` – Character string to be used to indicate muted string. Default `"x"`.
- `number-type` – Type of numbers to use in fret label. Choices include `roman-lower`, `roman-upper`, `arabic` and `custom`. In the later case, the format string is supplied by the `fret-label-custom-format` property. Default `roman-lower`.
- `open-string` – Character string to be used to indicate open string. Default `"o"`.
- `orientation` – Orientation of fret-diagram. Options include `normal`, `landscape`, and `opposing-landscape`. Default `normal`.
- `string-count` – The number of strings. Default 6.
- `string-label-font-mag` – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6 for `normal` orientation, 0.5 for `landscape` and `opposing-landscape`.
- `string-thickness-factor` – Factor for changing thickness of each string in the fret diagram. Thickness of string k is given by `thickness` * (1+`string-thickness-factor`) ^ (k-1). Default 0.
- `top-fret-thickness` – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.
- `xo-font-magnification` – Magnification used for mute and open string indicators. Default value 0.5.
- `xo-padding` – Padding for open and mute indicators from top fret. Default value 0.25.

`full-length-padding` (number)
: How much padding to use at the right side of a full-length tuplet bracket.

`full-length-to-extent` (boolean)
: Run to the extent of the column for a full-length tuplet bracket.

`full-measure-extra-space` (number)
: Extra space that is allocated at the beginning of a measure with only one note. This property is read from the NonMusicalPaperColumn that begins the measure.

`full-size-change` (boolean)
: Don't make a change clef smaller.

`gap` (dimension, in staff space)
: Size of a gap in a variable symbol.

`gap-count` (integer)
: Number of gapped beams for tremolo.

`glissando-skip` (boolean)
: Should this `NoteHead` be skipped by glissandi?

`glyph` (string)
: A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

  In combination with (span) bar lines, it is a string resembling the bar line appearance in ASCII form.

`glyph-name` (string)
: The glyph name within the font.

  In the context of (span) bar lines, *glyph-name* represents a processed form of `glyph`, where decisions about line breaking etc. are already taken.

`glyph-name-alist` (list)
: An alist of key-string pairs.

`graphical` (boolean)
: Display in graphical (vs. text) form.

`grow-direction` (direction)
: Crescendo or decrescendo?

`hair-thickness` (number)
: Thickness of the thin line in a bar line.

`harp-pedal-details` (list)
: An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a `(`*`property . value`*`)` pair. The properties which can be included in harp-pedal-details include the following:

  - `box-offset` – Vertical shift of the center of flat/sharp pedal boxes above/below the horizontal line. Default value 0.8.
  - `box-width` – Width of each pedal box. Default value 0.4.
  - `box-height` – Height of each pedal box. Default value 1.0.
  - `space-before-divider` – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
  - `space-after-divider` – Space between boxes after the first divider. Default value 0.8.
  - `circle-thickness` – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
  - `circle-x-padding` – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
  - `circle-y-padding` – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.

`head-direction` (direction)
: Are the note heads left or right in a semitie?

`height` (dimension, in staff space)
: Height of an object in `staff-space` units.

`height-limit` (dimension, in staff space)
: Maximum slur height: The longer the slur, the closer it is to this height.

`hide-tied-accidental-after-break` (boolean)
: If set, an accidental that appears on a tied note after a line break will not be displayed.

`horizon-padding` (number)
: The amount to pad the axis along which a `Skyline` is built for the `side-position-interface`.

`horizontal-shift` (integer)
: An integer that identifies ranking of `NoteColumn`s for horizontal shifting. This is used by Sección "note-collision-interface" in *Referencia de Funcionamiento Interno*.

`horizontal-skylines` (pair of skylines)
: Two skylines, one to the left and one to the right of this grob.

`id` (string)
: An id string for the grob. Depending on the typestting backend being used, this id will be assigned to a group containing all of the stencils that comprise a given grob. For example, in the svg backend, the string will be assigned to the `id` attribute of a group (<g>) that encloses the stencils that comprise the grob. In the Postscript backend, as there is no way to group items, the setting of the id property will have no effect.

`ignore-collision` (boolean)
: If set, don't do note collision resolution on this `NoteColumn`.

`implicit` (boolean)
: Is this an implicit bass figure?

`inspect-index` (integer)
: If debugging is set, set beam and slur configuration to this index, and print the respective scores.

`inspect-quants` (pair of numbers)
: If debugging is set, set beam and slur quants to this position, and print the respective scores.

`keep-inside-line` (boolean)
: If set, this column cannot have objects sticking into the margin.

`kern` (dimension, in staff space)
: Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

`knee` (boolean)
: Is this beam kneed?

`knee-spacing-correction` (number)
: Factor for the optical correction amount for kneed beams. Set between `0` for no correction and `1` for full correction.

`labels` (list)
: List of labels (symbols) placed on a column.

`layer` (integer)
: An integer which determines the order of printing objects. Objects with the lowest value of layer are drawn first, then objects with progressively higher values are drawn, so objects with higher values overwrite objects with lower values. By default most objects are assigned a layer value of 1.

`ledger-extra` (dimension, in staff space)
: Extra distance from staff line to draw ledger lines for.

`ledger-line-thickness` (pair of numbers)
: The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

`ledger-positions` (list)
: Repeating pattern for the vertical positions of ledger lines. Bracketed groups are always shown together.

`left-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`left-padding` (dimension, in staff space)
: The amount of space that is put left to an object (e.g., a lyric extender).

`length` (dimension, in staff space)
: User override for the stem length of unbeamed stems.

`length-fraction` (number)
: Multiplier for lengths. Used for determining ledger lines and stem lengths.

`line-break-penalty` (number)
: Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

`line-break-permission` (symbol)
: Instructs the line breaker on whether to put a line break at this column. Can be `force` or `allow`.

`line-break-system-details` (list)
: An alist of properties to use if this column is the start of a system.

`line-count` (integer)
: The number of staff lines.

`line-positions` (list)
: Vertical positions of staff lines.

`line-thickness` (number)
: The thickness of the tie or slur contour.

`long-text` (markup)
: Text markup. See Sección “Formatting text” in *Referencia de la Notación*.

`max-beam-connect` (integer)
: Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.

`max-stretch` (number)
: The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).

`maximum-gap` (number)
: Maximum value allowed for `gap` property.

`measure-count` (integer)
: The number of measures for a multi-measure rest.

`measure-length` (moment)
: Length of a measure. Used in some spacing situations.

`merge-differently-dotted` (boolean)
: Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.

  `merge-differently-dotted` only applies to opposing stem directions (i.e., voice 1 & 2).

`merge-differently-headed` (boolean)
: Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by Sección "note-collision-interface" in *Referencia de Funcionamiento Interno*.

  `merge-differently-headed` only applies to opposing stem directions (i.e., voice 1 & 2).

`minimum-distance` (dimension, in staff space)
: Minimum distance between rest and notes or beam.

`minimum-length` (dimension, in staff space)
: Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance between noteheads.

`minimum-length-fraction` (number)
: Minimum length of ledger line as fraction of note head size.

`minimum-space` (dimension, in staff space)
: Minimum distance that the victim should move (after padding).

`minimum-X-extent` (pair of numbers)
: Minimum size of an object in X dimension, measured in `staff-space` units.

`minimum-Y-extent` (pair of numbers)
: Minimum size of an object in Y dimension, measured in `staff-space` units.

`neutral-direction` (direction)
: Which direction to take in the center of the staff.

`neutral-position` (number)
: Position (in half staff spaces) where to flip the direction of custos stem.

`next` (graphical (layout) object)
: Object that is next relation (e.g., the lyric syllable following an extender).

`no-alignment` (boolean)
: If set, don't place this grob in a `VerticalAlignment`; rather, place it using its own `Y-offset` callback.

`no-ledgers` (boolean)
: If set, don't draw ledger lines on this object.

`no-stem-extend` (boolean)
: If set, notes with ledger lines do not get stems extending to the middle staff line.

`non-break-align-symbols` (list)
: A list of symbols that determine which NON-break-aligned interfaces to align this to.

`non-default` (boolean)
: Set for manually specified clefs.

`non-musical` (boolean)
: True if the grob belongs to a `NonMusicalPaperColumn`.

`nonstaff-nonstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the next non-staff line in the direction of `staff-affinity`, if both are on the same side of the related staff, and `staff-affinity` is either `UP` or `DOWN`. See `staff-staff-spacing` for a description of the alist structure.

`nonstaff-relatedstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the nearest staff in the direction of `staff-affinity`, if there are no non-staff lines between the two, and `staff-affinity` is either `UP` or `DOWN`. If `staff-affinity` is `CENTER`, then `nonstaff-relatedstaff-spacing` is used for the nearest staves on *both* sides, even if other non-staff lines appear between the current one and either of the staves. See `staff-staff-spacing` for a description of the alist structure.

`nonstaff-unrelatedstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the nearest staff in the opposite direction from `staff-affinity`, if there are no other non-staff lines between the two, and `staff-affinity` is either `UP` or `DOWN`. See `staff-staff-spacing` for a description of the alist structure.

`normalized-endpoints` (pair)
: Represents left and right placement over the total spanner, where the width of the spanner is normalized between 0 and 1.

`note-names` (vector)
: Vector of strings containing names for easy-notation note heads.

`outside-staff-horizontal-padding` (number)
: By default, an outside-staff-object can be placed so that is it very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.

`outside-staff-padding` (number)
: The padding to place between grobs when spacing according to `outside-staff-priority`. Two grobs with different `outside-staff-padding` values have the larger value of padding between them.

`outside-staff-placement-directive` (symbol)
: One of four directives telling how outside staff objects should be placed.

  - `left-to-right-greedy` – Place each successive grob from left to right.
  - `left-to-right-polite` – Place a grob from left to right only if it does not potentially overlap with another grob that has been placed on a pass through a grob array. If there is overlap, do another pass to determine placement.
  - `right-to-left-greedy` – Same as `left-to-right-greedy`, but from right to left.
  - `right-to-left-polite` – Same as `left-to-right-polite`, but from right to left.

`outside-staff-priority` (number)
: If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller `outside-staff-priority` is closer to the staff.

`packed-spacing` (boolean)
: If set, the notes are spaced as tightly as possible.

`padding` (dimension, in staff space)
: Add this much extra space between objects that are next to each other.

`padding-pairs` (list)
: An alist mapping `(`*`name . name`*`)` to distances.

`page-break-penalty` (number)
: Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.

`page-break-permission` (symbol)
: Instructs the page breaker on whether to put a page break at this column. Can be `force` or `allow`.

`page-turn-penalty` (number)
: Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

`page-turn-permission` (symbol)
: Instructs the page breaker on whether to put a page turn at this column. Can be `force` or `allow`.

`parenthesized` (boolean)
: Parenthesize this grob.

`positions` (pair of numbers)
: Pair of staff coordinates `(`*`left . right`*`)`, where both *left* and *right* are in `staff-space` units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.

`prefer-dotted-right` (boolean)
: For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.

`protrusion` (number)
: In an arpeggio bracket, the length of the horizontal edges.

`ratio` (number)
: Parameter for slur shape. The higher this number, the quicker the slur attains its `height-limit`.

`remove-empty` (boolean)
: If set, remove group if it contains no interesting items.

`remove-first` (boolean)
: Remove the first staff of an orchestral score?

`replacement-alist` (list)
: Alist of strings. The key is a string of the pattern to be replaced. The value is a string of what should be displayed. Useful for ligatures.

`restore-first` (boolean)
: Print a natural before the accidental.

`rhythmic-location` (rhythmic location)
: Where (bar number, measure position) in the score.

`right-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`right-padding` (dimension, in staff space)
: Space to insert on the right side of an object (e.g., between note and its accidentals).

`rotation` (list)
: Number of degrees to rotate this object, and what point to rotate around. For example, `'(45 0 0)` rotates by 45 degrees around the center of this object.

`round-up-exceptions` (list)
: A list of pairs where car is the numerator and cdr the denominator of a moment. Each pair in this list means that the multi-measure rests of the corresponding length will be rounded up to the longer rest. See *round-up-to-longer-rest*.

`round-up-to-longer-rest` (boolean)
: Displays the longer multi-measure rest when the length of a measure is between two values of `usable-duration-logs`. For example, displays a breve instead of a whole in a 3/2 measure.

`rounded` (boolean)
: Decide whether lines should be drawn rounded or not.

`same-direction-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.

`script-priority` (number)
: A key for determining the order of scripts in a stack, by being added to the position of the script in the user input, the sum being the overall priority. Smaller means closer to the head.

`self-alignment-X` (number)
: Specify alignment of an object. The value `-1` means left aligned, `0` centered, and `1` right-aligned in X direction. Other numerical values may also be specified.

`self-alignment-Y` (number)
: Like `self-alignment-X` but for the Y axis.

`sharp-positions` (list)
: Sharps in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: `(alto treble tenor soprano baritone mezzosoprano bass)`. If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.

`shorten-pair` (pair of numbers)
: The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.

`shortest-duration-space` (dimension, in staff space)
: Start with this much space for the shortest duration. This is expressed in `spacing-increment` as unit. See also Sección “spacing-spanner-interface” in *Referencia de Funcionamiento Interno*.

`shortest-playing-duration` (moment)
: The duration of the shortest note playing here.

`shortest-starter-duration` (moment)
: The duration of the shortest note that starts here.

`side-axis` (number)
: If the value is `X` (or equivalently `0`), the object is placed horizontally next to the other object. If the value is `Y` or `1`, it is placed vertically.

`side-relative-direction` (direction)
: Multiply direction of `direction-source` with this to get the direction of this object.

`simple-Y` (boolean)
: Should the Y placement of a spanner disregard changes in system heights?

`size` (number)
: Size of object, relative to standard size.

`skip-quanting` (boolean)
: Should beam quanting be skipped?

`skyline-horizontal-padding` (number)
: For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

`skyline-vertical-padding` (number)
: The amount by which the left and right skylines of a column are padded vertically, beyond the `Y-extent`s and `extra-spacing-height`s of the constituent grobs in the column. Increase this to prevent interleaving of grobs from adjacent columns.

`slash-negative-kern` (number)
: The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

`slope` (number)
: The slope of this object.

`slur-padding` (number)
: Extra distance between slur and script.

`snap-radius` (number)
: The maximum distance between two objects that will cause them to snap to alignment along an axis.

`space-alist` (list)
: A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: `(`*`break-align-symbol type . distance`*`)`, where *type* can be the symbols `minimum-space` or `extra-space`.

`space-to-barline` (boolean)
: If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

`spacing-increment` (number)
: Add this much space for a doubled duration. Typically, the width of a note head. See also Sección “spacing-spanner-interface” in *Referencia de Funcionamiento Interno*.

`spacing-pair` (pair)
: A pair of alignment symbols which set an object's spacing relative to its left and right `BreakAlignment`s.

  For example, a `MultiMeasureRest` will ignore prefatory items at its bounds (i.e., clefs, key signatures and time signatures) using the following override:

  ```
  \override MultiMeasureRest
    #'spacing-pair = #'(staff-bar . staff-bar)
  ```

`spanner-id` (string)
: An identifier to distinguish concurrent spanners.

`springs-and-rods` (boolean)
: Dummy variable for triggering spacing routines.

`stacking-dir` (direction)
: Stack objects in which direction?

`staff-affinity` (direction)
: The direction of the staff to use for spacing the current non-staff line. Choices are `UP`, `DOWN`, and `CENTER`. If `CENTER`, the non-staff line will be placed equidistant between the two nearest staves on either side, unless collisions or other spacing constraints prevent this. Setting `staff-affinity` for a staff causes it to be treated as a non-staff line. Setting `staff-affinity` to `#f` causes a non-staff line to be treated as a staff.

`staff-padding` (dimension, in staff space)
: Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

`staff-position` (number)
: Vertical position, measured in half staff spaces, counted from the middle line.

`staff-space` (dimension, in staff space)
: Amount of space between staff lines, expressed in global `staff-space`.

`staff-staff-spacing` (list)
: When applied to a staff-group's `StaffGrouper` grob, this spacing alist controls the distance between consecutive staves within the staff-group. When applied to a staff's `VerticalAxisGroup` grob, it controls the distance between the staff and the nearest staff below it in the same system, replacing any settings inherited from the `StaffGrouper` grob of the containing staff-group, if there is one. This property remains in effect even when non-staff lines appear between staves. The alist can contain the following keys:

  - `basic-distance` – the vertical distance, measured in staff-spaces, between the reference points of the two items when no collisions would result, and no stretching or compressing is in effect.
  - `minimum-distance` – the smallest allowable vertical distance, measured in staff-spaces, between the reference points of the two items, when compressing is in effect.
  - `padding` – the minimum required amount of unobstructed vertical whitespace between the bounding boxes (or skylines) of the two items, measured in staff-spaces.
  - `stretchability` – a unitless measure of the dimension's relative propensity to stretch. If zero, the distance will not stretch (unless collisions would result).

`staffgroup-staff-spacing` (list)
: The spacing alist controlling the distance between the last staff of the current staff-group and the staff just below it in the same system, even if one or more non-staff lines exist between the two staves. If the `staff-staff-spacing` property of the staff's `VerticalAxisGroup` grob is set, that is used instead. See `staff-staff-spacing` for a description of the alist structure.

`stem-attachment` (pair of numbers)
: An `(x . y)` pair where the stem attaches to the notehead.

`stem-begin-position` (number)
: User override for the begin position of a stem.

`stem-spacing-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

`stemlet-length` (number)
: How long should be a stem over a rest?

`stencil` (stencil)
: The symbol to print.

`stencils` (list)
: Multiple stencils, used as intermediate value.

`strict-grace-spacing` (boolean)
: If set, main notes are spaced normally, then grace notes are put left of the musical columns for the main notes.

`strict-note-spacing` (boolean)
: If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

`stroke-style` (string)
: Set to `"grace"` to turn stroke through flag on.

`style` (symbol)
: This setting determines in what style a grob is typeset. Valid choices depend on the `stencil` callback reading this property.

`text` (markup)
: Text markup. See Sección "Formatting text" in *Referencia de la Notación*.

`text-direction` (direction)
: This controls the ordering of the words. The default `RIGHT` is for roman text. Arabic or Hebrew should use `LEFT`.

`thick-thickness` (number)
: Bar line thickness, measured in `line-thickness`.

`thickness` (number)
: Line thickness, generally measured in `line-thickness`.

`thin-kern` (number)
: The space after a hair-line in a bar line.

`tie-configuration` (list)
: List of `(`*`position . dir`*`)` pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the

direction of the tie (`1`=>up, `-1`=>down, `0`=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

`to-barline` (boolean)
: If true, the spanner will stop at the bar line just before it would otherwise stop.

`toward-stem-shift` (number)
: Amount by which scripts are shifted toward the stem if their direction coincides with the stem direction. `0.0` means keep the default position (centered on the note head), `1.0` means centered on the stem. Interpolated values are possible.

`transparent` (boolean)
: This makes the grob invisible.

`uniform-stretching` (boolean)
: If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.

`usable-duration-logs` (list)
: List of `duration-log`s that can be used in typesetting the grob.

`use-skylines` (boolean)
: Should skylines be used for side positioning?

`used` (boolean)
: If set, this spacing column is kept in the spacing problem.

`vertical-skylines` (pair of skylines)
: Two skylines, one above and one below this grob.

`when` (moment)
: Global time step associated with this column happen?

`whiteout` (boolean)
: If true, the grob is printed over a white background to white-out underlying material, if the grob is visible. Usually #f by default.

`width` (dimension, in staff space)
: The width of a grob measured in staff space.

`word-space` (dimension, in staff space)
: Space to insert between words in texts.

`X-extent` (pair of numbers)
: Extent (size) in the X direction, measured in staff-space units, relative to object's reference point.

`X-offset` (number)
: The horizontal amount that this object is moved relative to its X-parent.

`X-positions` (pair of numbers)
: Pair of X staff coordinates of a spanner in the form `(`*`left`*` . `*`right`*`)`, where both *left* and *right* are in `staff-space` units of the current staff.

`Y-extent` (pair of numbers)
: Extent (size) in the Y direction, measured in staff-space units, relative to object's reference point.

`Y-offset` (number)
: The vertical amount that this object is moved relative to its Y-parent.

`zigzag-length` (dimension, in staff space)
: The length of the lines of a zigzag, relative to `zigzag-width`. A value of `1` gives 60-degree zigzags.

`zigzag-width` (dimension, in staff space)
: The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

A.18 Funciones musicales disponibles

`absolute` [music] - *music* (music)
: Make *music* absolute. This does not actually change the music itself but rather hides it from surrounding `\relative` commands.

`acciaccatura` [music] - *music* (music)
: Create an acciaccatura from the following music expression

`accidentalStyle` [music] - *style* (symbol list)
: Set accidental style to symbol list *style* in the form '`piano-cautionary`'. If *style* has a form like '`Staff.piano-cautionary`', the settings are applied to that context. Otherwise, the context defaults to '`Staff`', except for piano styles, which use '`GrandStaff`' as a context.

`addChordShape` [void] - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (string or pair)
: Add chord shape *shape-definition* to the *chord-shape-table* hash with the key `(cons key-symbol tuning)`.

`addInstrumentDefinition` [void] - *name* (string) *lst* (list)
: Create instrument *name* with properties *list*.

`addQuote` [void] - *name* (string) *music* (music)
: Define *music* as a quotable music expression named *name*

`afterGrace` [music] - *main* (music) *grace* (music)
: Create *grace* note(s) after a *main* music expression.

`allowPageTurn` [music]
: Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`allowVoltaHook` [void] - *bar* (string)
: (undocumented; fixme)

`alterBroken` [music] - *property* (symbol list or symbol) *arg* (list) *item* (symbol list or music)
: Override *property* for pieces of broken spanner *item* with values *arg*. *item* may either be music in the form of a starting spanner event, or a symbol list in the form '`Context.Grob`' or just '`Grob`'. Iff *item* is in the form of a spanner event, *property* may also have the form '`Grob.property`' for specifying a directed tweak.

`appendToTag` [music] - *tag* (symbol) *more* (music) *music* (music)
: Append *more* to the `elements` of all music expressions in *music* that are tagged with *tag*.

`applyContext` [music] - *proc* (procedure)
: Modify context properties with Scheme procedure *proc*.

`applyMusic` [music] - *func* (procedure) *music* (music)
: Apply procedure *func* to *music*.

`applyOutput` [music] - *ctx* (symbol) *proc* (procedure)
: Apply function `proc` to every layout object in context `ctx`

`appoggiatura` [music] - *music* (music)
Create an appoggiatura from *music*

`assertBeamQuant` [music] - *l* (pair) *r* (pair)
Testing function: check whether the beam quants *l* and *r* are correct

`assertBeamSlope` [music] - *comp* (procedure)
Testing function: check whether the slope of the beam is the same as `comp`

`autochange` [music] - *music* (music)
Make voices that switch between staves automatically

`balloonGrobText` [music] - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
Attach *text* to *grob-name* at offset *offset* (use like `\once`)

`balloonText` [post event] - *offset* (pair of numbers) *text* (markup)
Attach *text* at *offset* (use like `\tweak`)

`bar` [music] - *type* (string)
Insert a bar line of type *type*

`barNumberCheck` [music] - *n* (integer)
Print a warning if the current bar number is not *n*.

`bendAfter` [post event] - *delta* (real number)
Create a fall or doit of pitch interval *delta*.

`bookOutputName` [void] - *newfilename* (string)
Direct output for the current book block to *newfilename*.

`bookOutputSuffix` [void] - *newsuffix* (string)
Set the output filename suffix for the current book block to *newsuffix*.

`breathe` [music]
Insert a breath mark.

`chordRepeats` [music] - *event-types* [list] *music* (music)
Walk through *music* putting the notes of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(string-number-event)`.

`clef` [music] - *type* (string)
Set the current clef to *type*.

`compoundMeter` [music] - *args* (pair)
Create compound time signatures. The argument is a Scheme list of lists. Each list describes one fraction, with the last entry being the denominator, while the first entries describe the summands in the enumerator. If the time signature consists of just one fraction, the list can be given directly, i.e. not as a list containing a single list. For example, a time signature of (3+1)/8 + 2/4 would be created as `\compoundMeter #'((3 1 8) (2 4))`, and a time signature of (3+2)/8 as `\compoundMeter #'((3 2 8))` or shorter `\compoundMeter #'(3 2 8)`.

`crossStaff` [music] - *notes* (music)
Create cross-staff stems

`cueClef` [music] - *type* (string)
Set the current cue clef to *type*.

`cueClefUnset` [music]
Unset the current cue clef.

`cueDuring` [music] - *what* (string) *dir* (direction) *main-music* (music)
: Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.

`cueDuringWithClef` [music] - *what* (string) *dir* (direction) *clef* (string) *main-music* (music)
: Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.

`deadNote` [music] - *note* (music)
: Print *note* with a cross-shaped note head.

`defaultNoteHeads` [music]
: Revert to the default note head style.

`defineBarLine` [void] - *bar* (string) *glyph-list* (list)
: Define bar line settings for bar line *bar*. The list *glyph-list* must have three entries which define the appearance at the end of line, at the beginning of the next line, and the span bar, respectively.

`displayLilyMusic` [music] - *music* (music)
: Display the LilyPond input representation of *music* to the console.

`displayMusic` [music] - *music* (music)
: Display the internal representation of *music* to the console.

`displayScheme` (any type) - *expr* (any type)
: Display the internal representation of *expr* to the console.

`endSpanners` [music] - *music* (music)
: Terminate the next spanner prematurely after exactly one note without the need of a specific end spanner.

`eventChords` [music] - *music* (music)
: Compatibility function wrapping `EventChord` around isolated rhythmic events occuring since version 2.15.28, after expanding repeat chords '`q`'.

`featherDurations` [music] - *factor* (moment) *argument* (music)
: Adjust durations of music in *argument* by rational *factor*.

`finger` [post event] - *finger* (number or markup)
: Apply *finger* as a fingering indication.

`footnote` [music] - *mark* [markup] *offset* (pair of numbers) *footnote* (markup) *item* (symbol list or music)
: Make the markup *footnote* a footnote on *item*. The footnote is marked with a markup *mark* moved by *offset* with respect to the marked music.

  If *mark* is not given or specified as `\default`, it is replaced by an automatically generated sequence number. If *item* is a symbol list of form '`Grob`' or '`Context.Grob`', then grobs of that type will be marked at the current time step in the given context (default `Bottom`).

  If *item* is music, the music will get a footnote attached to a grob immediately attached to the event, like `\tweak` does. For attaching a footnote to an *indirectly* caused grob, write `\single\footnote`, use *item* to specify the grob, and follow it with the music to annotate.

  Like with `\tweak`, if you use a footnote on a following post-event, the `\footnote` command itself needs to be attached to the preceding note or rest as a post-event with `-`.

`grace` [music] - *music* (music)
: Insert *music* as grace notes.

`grobdescriptions` (any type) - *descriptions* (list)
: Create a context modification from *descriptions*, a list in the format of `all-grob-descriptions`.

`harmonicByFret` [music] - *fret* (number) *music* (music)
: Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at *fret*.

`harmonicByRatio` [music] - *ratio* (number) *music* (music)
: Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through *ratio*.

`harmonicNote` [music] - *note* (music)
: Print *note* with a diamond-shaped note head.

`harmonicsOn` [music]
: Set the default note head style to a diamond-shaped style.

`hide` [music] - *item* (symbol list or music)
: Set *item*'s '`transparent`' property to `#t`, making it invisible while still retaining its dimensions.

  If *item* is a symbol list of form `GrobName` or `Context.GrobName`, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

`inStaffSegno` [music]
: Put the segno variant 'varsegno' at this position into the staff, compatible with the repeat command.

`instrumentSwitch` [music] - *name* (string)
: Switch instrument to *name*, which must be predefined with `\addInstrumentDefinition`.

`inversion` [music] - *around* (pitch) *to* (pitch) *music* (music)
: Invert *music* about *around* and transpose from *around* to *to*.

`keepWithTag` [music] - *tag* (symbol list or symbol) *music* (music)
: Include only elements of *music* that are either untagged or tagged with one of the tags in *tag*. *tag* may be either a single symbol or a list of symbols.

`key` [music] - *tonic* [pitch] *pitch-alist* [list]
: Set key to *tonic* and scale *pitch-alist*. If both are null, just generate `KeyChangeEvent`.

`killCues` [music] - *music* (music)
: Remove cue notes from *music*.

`label` [music] - *label* (symbol)
: Create *label* as a bookmarking label.

`language` [void] - *language* (string)
: Set note names for language *language*.

`languageRestore` [void]
: Restore a previously-saved pitchnames alist.

`languageSaveAndChange` [void] - *language* (string)
: Store the previous pitchnames alist, and set a new one.

`makeClusters` [music] - *arg* (music)
Display chords in *arg* as clusters.

`makeDefaultStringTuning` [void] - *symbol* (symbol) *pitches* (list)
This defines a string tuning *symbol* via a list of *pitches*. The *symbol* also gets registered in `defaultStringTunings` for documentation purposes.

`mark` [music] - *label* [any type]
Make the music for the \mark command.

`modalInversion` [music] - *around* (pitch) *to* (pitch) *scale* (music) *music* (music)
Invert *music* about *around* using *scale* and transpose from *around* to *to*.

`modalTranspose` [music] - *from* (pitch) *to* (pitch) *scale* (music) *music* (music)
Transpose *music* from pitch *from* to pitch *to* using *scale*.

`musicMap` [music] - *proc* (procedure) *mus* (music)
Apply *proc* to *mus* and all of the music it contains.

`noPageBreak` [music]
Forbid a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`noPageTurn` [music]
Forbid a page turn. May be used at toplevel (i.e., between scores or markups), or inside a score.

`octaveCheck` [music] - *pitch* (pitch)
Octave check.

`offset` [music] - *property* (symbol list or symbol) *offsets* (any type) *item* (symbol list or music)
Offset the default value of *property* of *item* by *offsets*. If *item* is a string, the result is `\override` for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`omit` [music] - *item* (symbol list or music)
Set *item*'s '`stencil`' property to `#f`, effectively omitting it without taking up space.
If *item* is a symbol list of form `GrobName` or `Context.GrobName`, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

`once` [music] - *music* (music)
Set `once` to `#t` on all layout instruction events in *music*. This will complain about music with an actual duration. As a special exception, if *music* contains '`tweaks`' it will be silently ignored in order to allow for `\once \tweak` to work as both one-time override and proper tweak.

`ottava` [music] - *octave* (integer)
Set the octavation.

`overrideProperty` [music] - *grob-property-path* (symbol list) *value* (any type)
Set the grob property specified by *grob-property-path* to *value*. *grob-property-path* is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well.

`overrideTimeSignatureSettings` [music] - *time-signature* (fraction, as pair) *base-moment* (fraction, as pair) *beat-structure* (list) *beam-exceptions* (list)
Override `timeSignatureSettings` for time signatures of *time-signature* to have settings of *base-moment*, *beat-structure*, and *beam-exceptions*.

`pageBreak` [music]
: Force a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`pageTurn` [music]
: Force a page turn between two scores or top-level markups.

`palmMute` [music] - *note* (music)
: Print *note* with a triangle-shaped note head.

`palmMuteOn` [music]
: Set the default note head style to a triangle-shaped style.

`parallelMusic` [void] - *voice-ids* (list) *music* (music)
: Define parallel music sequences, separated by '`|`' (bar check signs), and assign them to the identifiers provided in *voice-ids*.

  voice-ids: a list of music identifiers (symbols containing only letters)

  music: a music sequence, containing BarChecks as limiting expressions.

  Example:

```
\parallelMusic #'(A B C) {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
  A = { c c | d d | }
  B = { d d | e e | }
  C = { e e | f f | }
```

`parenthesize` [music] - *arg* (music)
: Tag *arg* to be parenthesized.

`partcombine` [music] - *part1* (music) *part2* (music)
: Take the music in *part1* and *part2* and typeset so that they share a staff.

`partcombineDown` [music] - *part1* (music) *part2* (music)
: Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed downward.

`partcombineForce` [music] - *type* (symbol-or-boolean) *once* (boolean)
: Override the part-combiner.

`partcombineUp` [music] - *part1* (music) *part2* (music)
: Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed upward.

`partial` [music] - *dur* (duration)
: Make a partial measure.

`phrasingSlurDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)
: Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for phrasing slurs.

`pitchedTrill` [music] - *main-note* (music) *secondary-note* (music)
: Print a trill with *main-note* as the main note of the trill and print *secondary-note* as a stemless note head in parentheses.

`pointAndClickOff` [void]
: Suppress generating extra code in final-format (e.g. pdf) files to point back to the lilypond source statement.

`pointAndClickOn` [void]
: Enable generation of code in final-format (e.g. pdf) files to reference the originating lilypond source statement; this is helpful when developing a score but generates bigger final-format files.

`pointAndClickTypes` [void] - *types* (symbol list or symbol)
: Set a type or list of types (such as `#'note-event`) for which point-and-click info is generated.

`pushToTag` [music] - *tag* (symbol) *more* (music) *music* (music)
: Add *more* to the front of `elements` of all music expressions in *music* that are tagged with *tag*.

`quoteDuring` [music] - *what* (string) *main-music* (music)
: Indicate a section of music to be quoted. *what* indicates the name of the quoted voice, as specified in an `\addQuote` command. *main-music* is used to indicate the length of music to be quoted; usually contains spacers or multi-measure rests.

`relative` [music] - *pitch* [pitch] *music* (music)
: Make *music* relative to *pitch*. If *pitch* is omitted, the first note in *music* is given in absolute pitch.

`removeWithTag` [music] - *tag* (symbol list or symbol) *music* (music)
: Remove elements of *music* that are tagged with one of the tags in *tag*. *tag* may be either a single symbol or a list of symbols.

`resetRelativeOctave` [music] - *pitch* (pitch)
: Set the octave inside a \relative section.

`retrograde` [music] - *music* (music)
: Return *music* in reverse order.

`revertTimeSignatureSettings` [music] - *time-signature* (pair)
: Revert `timeSignatureSettings` for time signatures of *time-signature*.

`rightHandFinger` [post event] - *finger* (number or markup)
: Apply *finger* as a fingering indication.

`scaleDurations` [music] - *fraction* (fraction, as pair) *music* (music)
: Multiply the duration of events in *music* by *fraction*.

`settingsFrom` (any type) - *ctx* [symbol] *music* (music)
: Take the layout instruction events from *music*, optionally restricted to those applying to context type *ctx*, and return a context modification duplicating their effect.

`shape` [music] - *offsets* (list) *item* (symbol list or music)
: Offset control-points of *item* by *offsets*. The argument is a list of number pairs or list of such lists. Each element of a pair represents an offset to one of the coordinates of a control-point. If *item* is a string, the result is `\once\override` for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`shiftDurations` [music] - *dur* (integer) *dots* (integer) *arg* (music)
: Change the duration of *arg* by adding *dur* to the `durlog` of *arg* and *dots* to the `dots` of *arg*.

`single` [music] - *overrides* (music) *music* (music)
: Convert *overrides* to tweaks and apply them to *music*. This does not convert `\revert`, `\set` or `\unset`.

`skip` [music] - *dur* (duration)
: Skip forward by *dur*.

`slashedGrace` [music] - *music* (music)
: Create slashed graces (slashes through stems, but no slur) from the following music expression

`slurDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)
: Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for slurs.

`spacingTweaks` [music] - *parameters* (list)
: Set the system stretch, by reading the 'system-stretch property of the 'parameters' assoc list.

`storePredefinedDiagram` [void] - *fretboard-table* (hash table) *chord* (music) *tuning* (pair) *diagram-definition* (string or pair)
: Add predefined fret diagram defined by *diagram-definition* for the chord pitches *chord* and the stringTuning *tuning*.

`stringTuning` (any type) - *chord* (music)
: Convert *chord* to a string tuning. *chord* must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.

`styledNoteHeads` [music] - *style* (symbol) *heads* (symbol list or symbol) *music* (music)
: Set *heads* in *music* to *style*.

`tabChordRepeats` [music] - *event-types* [list] *music* (music)
: Walk through *music* putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(articulation-event)`.

`tabChordRepetition` [void]
: Include the string and fingering information in a chord repetition. This function is deprecated; try using `\tabChordRepeats` instead.

`tag` [music] - *tag* (symbol list or symbol) *music* (music)
: Tag the following *music* with *tag* and return the result, by adding the single symbol or symbol list *tag* to the `tags` property of *music*.

`temporary` [music] - *music* (music)
: Make any `\override` in *music* replace an existing grob property value only temporarily, restoring the old value when a corresponding `\revert` is executed. This is achieved by clearing the '`pop-first`' property normally set on `\overrides`.

  An `\override`/`\revert` sequence created by using `\temporary` and `\undo` on the same music containing overrides will cancel out perfectly or cause a warning.

  Non-property-related music is ignored, warnings are generated for any property-changing music that isn't an `\override`.

`tieDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)
: Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for ties.

`time` [music] - *beat-structure* [number list] *fraction* (fraction, as pair)
: Set *fraction* as time signature, with optional number list *beat-structure* before it.

`times` [music] - *fraction* (fraction, as pair) *music* (music)
: Scale *music* in time by *fraction*.

`tocItem` [music] - *text* (markup)
: Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transpose` [music] - *from* (pitch) *to* (pitch) *music* (music)
: Transpose *music* from pitch *from* to pitch *to*.

`transposedCueDuring` [music] - *what* (string) *dir* (direction) *pitch* (pitch) *main-music* (music)
: Insert notes from the part *what* into a voice called `cue`, using the transposition defined by *pitch*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`transposition` [music] - *pitch* (pitch)
: Set instrument transposition

`tuplet` [music] - *ratio* (fraction, as pair) *tuplet-span* [duration] *music* (music)
: Scale the given *music* to tuplets. *ratio* is a fraction that specifies how many notes are played in place of the nominal value: it will be '`3/2`' for triplets, namely three notes being played in place of two. If the optional duration *tuplet-span* is specified, it is used instead of `tupletSpannerDuration` for grouping the tuplets. For example,

  ```
  \tuplet 3/2 4 { c8 c c c c c }
  ```

  will result in two groups of three tuplets, each group lasting for a quarter note.

`tupletSpan` [music] - *tuplet-span* [duration]
: Set `tupletSpannerDuration`, the length into which `\tuplet` without an explicit '`tuplet-span`' argument of its own will group its tuplets, to the duration *tuplet-span*. To revert to the default of not subdividing the contents of a `\tuplet` command without explicit '`tuplet-span`', use

  ```
  \tupletSpan \default
  ```

`tweak` [music] - *prop* (symbol list or symbol) *value* (any type) *item* (symbol list or music)
: Add a tweak to the following *item*, usually music. Layout objects created by *item* get their property *prop* set to *value*. If *prop* has the form '`Grob.property`', like with

  ```
  \tweak Accidental.color #red cis'
  ```

  an indirectly created grob ('`Accidental`' is caused by '`NoteHead`') can be tweaked; otherwise only directly created grobs are affected.

  As a special case, *item* may be a symbol list specifying a grob path, in which case `\override` is called on it instead of creating tweaked music. This is mainly useful when using `\tweak` as as a component for building other functions.

  If this use case would call for `\once \override` rather than a plain `\override`, writing `\once \tweak ...` can be convenient.

  prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

`undo` [music] - *music* (music)
: Convert `\override` and `\set` in *music* to `\revert` and `\unset`, respectively. Any reverts and unsets already in *music* cause a warning. Non-property-related music is ignored.

`unfoldRepeats` [music] - *music* (music)
: Force any `\repeat volta`, `\repeat tremolo` or `\repeat percent` commands in *music* to be interpreted as `\repeat unfold`.

`void` [void] - *arg* (any type)
: Accept a scheme argument, return a void expression. Use this if you want to have a scheme expression evaluated because of its side-effects, but its value ignored.

`withMusicProperty` [music] - *sym* (symbol) *val* (any type) *music* (music)
: Set *sym* to *val* in *music*.

`xNote` [music] - *note* (music)
: Print *note* with a cross-shaped note head.

`xNotesOn` [music]
: Set the default note head style to a cross-shaped style.

A.19 Identificadores de modificación de contextos

Se definen las siguientes instrucciones para su uso como modificaciones de contexto dentro de un bloque `\layout` o `\with`.

`RemoveEmptyStaves`
: Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`.
  - Sets grob property `remove-empty` in `Sección ‘‘VerticalAxisGroup’’ in` *Referencia de Funcionamiento Interno* to `#t`.

A.20 Predicados de tipo predefinidos

R5RS primary predicates

Type predicate	**Description**
`boolean?`	boolean
`char?`	character
`number?`	number
`pair?`	pair
`port?`	port
`procedure?`	procedure
`string?`	string
`symbol?`	symbol
`vector?`	vector

R5RS secondary predicates

Type predicate	**Description**
`char-alphabetic?`	alphabetic character
`char-lower-case?`	lower-case character
`char-numeric?`	numeric character
`char-upper-case?`	upper-case character
`char-whitespace?`	whitespace character
`complex?`	complex number
`eof-object?`	end-of-file object
`even?`	even number
`exact?`	exact number
`inexact?`	inexact number
`input-port?`	input port
`integer?`	integer
`list?`	list *(use* `cheap-list?` *for faster processing)*
`negative?`	negative number
`null?`	null

`odd?`	odd number
`output-port?`	output port
`positive?`	positive number
`rational?`	rational number
`real?`	real number
`zero?`	zero

Guile predicates

Type predicate	**Description**
`hash-table?`	hash table

LilyPond scheme predicates

Type predicate	**Description**
`boolean-or-symbol?`	boolean or symbol
`cheap-list?`	list *(use this instead of* `list?` *for faster processing)*
`color?`	color
`fraction?`	fraction, as pair
`grob-list?`	list of grobs
`index?`	non-negative integer
`markup?`	markup
`markup-command-list?`	markup command list
`markup-list?`	markup list
`moment-pair?`	pair of moment objects
`number-list?`	number list
`number-or-grob?`	number or grob
`number-or-markup?`	number or markup
`number-or-pair?`	number or pair
`number-or-string?`	number or string
`number-pair?`	pair of numbers
`number-pair-list?`	list of number pairs
`rhythmic-location?`	rhythmic location
`scheme?`	any type
`string-or-music?`	string or music
`string-or-pair?`	string or pair
`string-or-symbol?`	string or symbol
`symbol-list?`	symbol list
`symbol-list-or-music?`	symbol list or music
`symbol-list-or-symbol?`	symbol list or symbol
`void?`	void

LilyPond exported predicates

Type predicate	**Description**
`ly:book?`	book
`ly:box?`	box
`ly:context?`	context
`ly:context-def?`	context definition
`ly:context-mod?`	context modification
`ly:dimension?`	dimension, in staff space

`ly:dir?`	direction
`ly:dispatcher?`	dispatcher
`ly:duration?`	duration
`ly:event?`	post event
`ly:font-metric?`	font metric
`ly:grob?`	graphical (layout) object
`ly:grob-array?`	array of grobs
`ly:input-location?`	input location
`ly:item?`	item
`ly:iterator?`	iterator
`ly:lily-lexer?`	lily-lexer
`ly:lily-parser?`	lily-parser
`ly:listener?`	listener
`ly:moment?`	moment
`ly:music?`	music
`ly:music-function?`	music function
`ly:music-list?`	list of music objects
`ly:music-output?`	music output
`ly:otf-font?`	OpenType font
`ly:output-def?`	output definition
`ly:page-marker?`	page marker
`ly:pango-font?`	pango font
`ly:paper-book?`	paper book
`ly:paper-system?`	paper-system Prob
`ly:pitch?`	pitch
`ly:prob?`	property object
`ly:score?`	score
`ly:simple-closure?`	simple closure
`ly:skyline?`	skyline
`ly:skyline-pair?`	pair of skylines
`ly:source-file?`	source file
`ly:spanner?`	spanner
`ly:spring?`	spring
`ly:stencil?`	stencil
`ly:stream-event?`	stream event
`ly:translator?`	translator
`ly:translator-group?`	translator group
`ly:unpure-pure-container?`	unpure/pure container

A.21 Funciones de Scheme

`ly:add-context-mod` *contextmods modification* [Función]
Adds the given context *modification* to the list *contextmods* of context modifications.

`ly:add-file-name-alist` *alist* [Función]
Add mappings for error messages from *alist*.

`ly:add-interface` *iface desc props* [Función]
Add a new grob interface. *iface* is the interface name, *desc* is the interface description, and *props* is the list of user-settable properties for the interface.

`ly:add-listener` *list disp cl* [Función]
Add the listener *list* to the dispatcher *disp*. Whenever *disp* hears an event of class *cl*, it is forwarded to *list*.

`ly:add-option` *sym val description* [Función]
Add a program option *sym*. *val* is the default value and *description* is a string description.

`ly:all-grob-interfaces` [Función]
Return the hash table with all grob interface descriptions.

`ly:all-options` [Función]
Get all option settings in an alist.

`ly:all-stencil-expressions` [Función]
Return all symbols recognized as stencil expressions.

`ly:assoc-get` *key alist default-value strict-checking* [Función]
Return value if *key* in *alist*, else *default-value* (or `#f` if not specified). If *strict-checking* is set to `#t` and *key* is not in *alist*, a programming_error is output.

`ly:axis-group-interface::add-element` *grob grob-element* [Función]
Set *grob* the parent of *grob-element* on all axes of *grob*.

`ly:basic-progress` *str rest* [Función]
A Scheme callable function to issue a basic progress message *str*. The message is formatted with `format` and *rest*.

`ly:beam-score-count` [Función]
count number of beam scores.

`ly:book?` *x* [Función]
Is *x* a `Book` object?

`ly:book-add-bookpart!` *book-smob book-part* [Función]
Add *book-part* to *book-smob* book part list.

`ly:book-add-score!` *book-smob score* [Función]
Add *score* to *book-smob* score list.

`ly:book-book-parts` *book* [Función]
Return book parts in *book*.

`ly:book-header` *book* [Función]
Return header in *book*.

`ly:book-paper` *book* [Función]
Return paper in *book*.

`ly:book-process` *book-smob default-paper default-layout output* [Función]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:book-process-to-systems` *book-smob default-paper default-layout output* [Función]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:book-scores` *book* [Función]
Return scores in *book*.

`ly:book-set-header!` *book module* [Función]
Set the book header.

`ly:box?` *x* [Función]
Is *x* a `Box` object?

`ly:bp` *num* [Función]
num bigpoints (1/72th inch).

`ly:bracket` *a iv t p* [Función]
Make a bracket in direction *a*. The extent of the bracket is given by *iv*. The wings protrude by an amount of *p*, which may be negative. The thickness is given by *t*.

`ly:broadcast` *disp ev* [Función]
Send the stream event *ev* to the dispatcher *disp*.

`ly:camel-case->lisp-identifier` *name-sym* [Función]
Convert `FooBar_Bla` to `foo-bar-bla` style symbol.

`ly:chain-assoc-get` *key achain default-value strict-checking* [Función]
Return value for *key* from a list of alists *achain*. If no entry is found, return *default-value* or `#f` if *default-value* is not specified. With *strict-checking* set to `#t`, a programming_error is output in such cases.

`ly:check-expected-warnings` [Función]
Check whether all expected warnings have really been triggered.

`ly:cm` *num* [Función]
num cm.

`ly:command-line-code` [Función]
The Scheme code specified on command-line with '`-e`'.

`ly:command-line-options` [Función]
The Scheme options specified on command-line with '`-d`'.

`ly:connect-dispatchers` *to from* [Función]
Make the dispatcher *to* listen to events from *from*.

`ly:context?` *x* [Función]
Is *x* a `Context` object?

`ly:context-current-moment` *context* [Función]
Return the current moment of *context*.

`ly:context-def?` *x* [Función]
Is *x* a `Context_def` object?

`ly:context-def-lookup` *def sym val* [Función]
Return the value of *sym* in context definition *def* (e.g., `\Voice`). If no value is found, return *val* or `'()` if *val* is undefined. *sym* can be any of '`default-child`', '`consists`', '`description`', '`aliases`', '`accepts`', '`property-ops`', '`context-name`', '`group-type`'.

`ly:context-def-modify` *def mod* [Función]
Return the result of applying the context-mod *mod* to the context definition *def*. Does not change *def*.

`ly:context-event-source` *context* [Función]
Return `event-source` of context *context*.

`ly:context-events-below` *context* [Función]
Return a `stream-distributor` that distributes all events from *context* and all its subcontexts.

`ly:context-find` *context name* [Función]
Find a parent of *context* that has name or alias *name*. Return `#f` if not found.

`ly:context-grob-definition` *context name* [Función]
Return the definition of *name* (a symbol) within *context* as an alist.

`ly:context-id` *context* [Función]
Return the ID string of *context*, i.e., for `\context Voice = "one" ...` return the string `one`.

`ly:context-mod?` *x* [Función]
Is *x* a `Context_mod` object?

`ly:context-mod-apply!` *context mod* [Función]
Apply the context modification *mod* to *context*.

`ly:context-name` *context* [Función]
Return the name of *context*, i.e., for `\context Voice = "one" ...` return the symbol `Voice`.

`ly:context-now` *context* [Función]
Return `now-moment` of context *context*.

`ly:context-parent` *context* [Función]
Return the parent of *context*, `#f` if none.

`ly:context-property` *context sym def* [Función]
Return the value for property *sym* in *context*. If *def* is given, and property value is `'()`, return *def*.

`ly:context-property-where-defined` *context name* [Función]
Return the context above *context* where *name* is defined.

`ly:context-pushpop-property` *context grob eltprop val* [Función]
Do a single `\override` or `\revert` operation in *context*. The grob definition *grob* is extended with *eltprop* (if *val* is specified) or reverted (if unspecified).

`ly:context-set-property!` *context name val* [Función]
Set value of property *name* in context *context* to *val*.

`ly:context-unset-property` *context name* [Función]
Unset value of property *name* in context *context*.

`ly:debug` *str rest* [Función]
A Scheme callable function to issue a debug message *str*. The message is formatted with `format` and *rest*.

`ly:default-scale` [Función]
Get the global default scale.

`ly:dimension?` *d* [Función]
Return *d* as a number. Used to distinguish length variables from normal numbers.

`ly:dir?` *s* [Función]
Is *s* a direction? Valid directions are `-1`, `0`, or `1`, where `-1` represents left or down, `1` represents right or up, and `0` represents a neutral direction.

`ly:dispatcher?` *x* [Función]
Is *x* a `Dispatcher` object?

`ly:duration?` *x* [Función]
Is *x* a `Duration` object?

`ly:duration<?` *p1 p2* [Función]
Is *p1* shorter than *p2*?

`ly:duration->string` *dur* [Función]
Convert *dur* to a string.

`ly:duration-dot-count` *dur* [Función]
Extract the dot count from *dur*.

`ly:duration-factor` *dur* [Función]
Extract the compression factor from *dur*. Return it as a pair.

`ly:duration-length` *dur* [Función]
The length of the duration as a `moment`.

`ly:duration-log` *dur* [Función]
Extract the duration log from *dur*.

`ly:duration-scale` *dur* [Función]
Extract the compression factor from *dur*. Return it as a rational.

`ly:effective-prefix` [Función]
Return effective prefix.

`ly:encode-string-for-pdf` *str* [Función]
Encode the given string to either Latin1 (which is a subset of the PDFDocEncoding) or if that's not possible to full UTF-16BE with Byte-Order-Mark (BOM).

`ly:engraver-announce-end-grob` *engraver grob cause* [Función]
Announce the end of a grob (i.e., the end of a spanner) originating from given *engraver* instance, with *grob* being a grob. *cause* should either be another grob or a music event.

`ly:engraver-make-grob` *engraver grob-name cause* [Función]
Create a grob originating from given *engraver* instance, with given *grob-name*, a symbol. *cause* should either be another grob or a music event.

`ly:error` *str rest* [Función]
A Scheme callable function to issue the error *str*. The error is formatted with `format` and *rest*.

`ly:eval-simple-closure` *delayed closure scm-start scm-end* [Función]
Evaluate a simple *closure* with the given *delayed* argument. If *scm-start* and *scm-end* are defined, evaluate it purely with those start and end points.

`ly:event?` *obj* [Función]
Is *obj* a proper (non-rhythmic) event object?

`ly:event-deep-copy` *m* [Función]
Copy *m* and all sub expressions of *m*.

`ly:event-property` *sev sym val* [Función]
Get the property *sym* of stream event *sev*. If *sym* is undefined, return *val* or `'()` if *val* is not specified.

`ly:event-set-property!` *ev sym val* [Función]
Set property *sym* in event *ev* to *val*.

`ly:expand-environment` *str* [Función]
Expand `$VAR` and `${VAR}` in *str*.

`ly:expect-warning` *str rest* [Función]
A Scheme callable function to register a warning to be expected and subsequently suppressed. If the warning is not encountered, a warning about the missing warning will be shown. The message should be translated with `(_ ...)` and changing parameters given after the format string.

`ly:find-file` *name* [Función]
Return the absolute file name of *name*, or `#f` if not found.

`ly:font-config-add-directory` *dir* [Función]
Add directory *dir* to FontConfig.

`ly:font-config-add-font` *font* [Función]
Add font *font* to FontConfig.

`ly:font-config-display-fonts` [Función]
Dump a list of all fonts visible to FontConfig.

`ly:font-config-get-font-file` *name* [Función]
Get the file for font *name*.

`ly:font-design-size` *font* [Función]
Given the font metric *font*, return the design size, relative to the current output-scale.

`ly:font-file-name` *font* [Función]
Given the font metric *font*, return the corresponding file name.

`ly:font-get-glyph` *font name* [Función]
Return a stencil from *font* for the glyph named *name*. If the glyph is not available, return an empty stencil.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-charcode` *font name* [Función]
Return the character code for glyph *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-index` *font name* [Función]
Return the index for *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-index-to-charcode` *font index* [Función]
Return the character code for *index* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-magnification` *font* [Función]
Given the font metric *font*, return the magnification, relative to the current output-scale.

`ly:font-metric?` *x* [Función]
Is *x* a `Font_metric` object?

`ly:font-name` *font* [Función]
Given the font metric *font*, return the corresponding name.

`ly:font-sub-fonts` *font* [Función]
Given the font metric *font* of an OpenType font, return the names of the subfonts within *font*.

`ly:format` *str rest* [Función]
LilyPond specific format, supporting `~a` and `~[0-9]f`. Basic support for `~s` is also provided.

`ly:format-output` *context* [Función]
Given a global context in its final state, process it and return the `Music_output` object in its final state.

`ly:get-all-function-documentation` [Función]
Get a hash table with all LilyPond Scheme extension functions.

`ly:get-all-translators` [Función]
Return a list of all translator objects that may be instantiated.

`ly:get-context-mods` *contextmod* [Función]
Returns the list of context modifications stored in *contextmod*.

`ly:get-option` *var* [Función]
Get a global option setting.

`ly:get-spacing-spec` *from-scm to-scm* [Función]
Return the spacing spec going between the two given grobs, *from_scm* and *to_scm*.

`ly:get-undead` *undead* [Función]
Get back object from *undead*.

`ly:gettext` *original* [Función]
A Scheme wrapper function for `gettext`.

`ly:grob?` *x* [Función]
Is *x* a `Grob` object?

`ly:grob-alist-chain` *grob global* [Función]
Get an alist chain for grob *grob*, with *global* as the global default. If unspecified, `font-defaults` from the layout block is taken.

`ly:grob-array?` *x* [Función]
Is *x* a `Grob_array` object?

`ly:grob-array->list` *grob-arr* [Función]
Return the elements of *grob-arr* as a Scheme list.

`ly:grob-array-length` *grob-arr* [Función]
Return the length of *grob-arr*.

`ly:grob-array-ref` *grob-arr index* [Función]
Retrieve the *index*th element of *grob-arr*.

`ly:grob-basic-properties` *grob* [Función]
Get the immutable properties of *grob*.

`ly:grob-chain-callback` *grob proc sym* [Función]
Find the callback that is stored as property *sym* of grob *grob* and chain *proc* to the head of this, meaning that it is called using *grob* and the previous callback's result.

`ly:grob-common-refpoint` *grob other axis* [Función]
Find the common refpoint of *grob* and *other* for *axis*.

`ly:grob-common-refpoint-of-array` *grob others axis* [Función]
Find the common refpoint of *grob* and *others* (a grob-array) for *axis*.

`ly:grob-default-font` *grob* [Función]
Return the default font for grob *grob*.

`ly:grob-extent` *grob refp axis* [Función]
Get the extent in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-get-vertical-axis-group-index` *grob* [Función]
Get the index of the vertical axis group the grob *grob* belongs to; return `-1` if none is found.

`ly:grob-interfaces` *grob* [Función]
Return the interfaces list of grob *grob*.

`ly:grob-layout` *grob* [Función]
Get `\layout` definition from grob *grob*.

`ly:grob-object` *grob sym* [Función]
Return the value of a pointer in grob *grob* of property *sym*. It returns `'()` (end-of-list) if *sym* is undefined in *grob*.

`ly:grob-original` *grob* [Función]
Return the unbroken original grob of *grob*.

`ly:grob-parent` *grob axis* [Función]
Get the parent of *grob*. *axis* is 0 for the X-axis, 1 for the Y-axis.

`ly:grob-pq<?` *a b* [Función]
Compare two grob priority queue entries. This is an internal function.

`ly:grob-properties` *grob* [Función]
Get the mutable properties of *grob*.

`ly:grob-property` *grob sym val* [Función]
Return the value for property *sym* of *grob*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:grob-property-data` *grob sym* [Función]
Return the value for property *sym* of *grob*, but do not process callbacks.

`ly:grob-pure-height` *grob refp beg end val* [Función]
Return the pure height of *grob* given refpoint *refp*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:grob-pure-property` *grob sym beg end val* [Función]
Return the pure value for property *sym* of *grob*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:grob-relative-coordinate` *grob refp axis* [Función]
Get the coordinate in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-robust-relative-extent` *grob refp axis* [Función]
Get the extent in *axis* direction of *grob* relative to the grob *refp*, or `(0,0)` if empty.

`ly:grob-script-priority-less` *a b* [Función]
Compare two grobs by script priority. For internal use.

`ly:grob-set-nested-property!` *grob symlist val* [Función]
Set nested property *symlist* in grob *grob* to value *val*.

`ly:grob-set-object!` *grob sym val* [Función]
Set *sym* in grob *grob* to value *val*.

`ly:grob-set-parent!` *grob axis parent-grob* [Función]
Set *parent-grob* the parent of grob *grob* in axis *axis*.

`ly:grob-set-property!` *grob sym val* [Función]
Set *sym* in grob *grob* to value *val*.

`ly:grob-staff-position` *sg* [Función]
Return the Y-position of *sg* relative to the staff.

`ly:grob-suicide!` *grob* [Función]
Kill *grob*.

`ly:grob-system` *grob* [Función]
Return the system grob of *grob*.

`ly:grob-translate-axis!` *grob d a* [Función]
Translate *grob* on axis *a* over distance *d*.

`ly:grob-vertical<?` *a b* [Función]
Does *a* lie above *b* on the page?

`ly:gulp-file` *name size* [Función]
Read *size* characters from the file *name*, and return its contents in a string. If *size* is undefined, the entire file is read. The file is looked up using the search path.

`ly:hash-table-keys` *tab* [Función]
Return a list of keys in *tab*.

`ly:inch` *num* [Función]
num inches.

`ly:input-both-locations` *sip* [Función]
Return input location in *sip* as `(file-name first-line first-column last-line last-column)`.

`ly:input-file-line-char-column` *sip* [Función]
Return input location in *sip* as `(file-name line char column)`.

`ly:input-location?` *x* [Función]
Is *x* an `input-location`?

`ly:input-message` *sip msg rest* [Función]
Print *msg* as a GNU compliant error message, pointing to the location in *sip*. *msg* is interpreted similar to `format`'s argument, using *rest*.

`ly:input-warning` *sip msg rest* [Función]
Print *msg* as a GNU compliant warning message, pointing to the location in *sip*. *msg* is interpreted similar to `format`'s argument, using *rest*.

`ly:interpret-music-expression` *mus ctx* [Función]
Interpret the music expression *mus* in the global context *ctx*. The context is returned in its final state.

`ly:interpret-stencil-expression` *expr func arg1 offset* [Función]
Parse *expr*, feed bits to *func* with first arg *arg1* having offset *offset*.

`ly:intlog2` *d* [Función]
The 2-logarithm of $1/d$.

`ly:item?` *g* [Función]
Is *g* an `Item` object?

`ly:item-break-dir` *it* [Función]
The break status direction of item *it*. `-1` means end of line, `0` unbroken, and `1` beginning of line.

`ly:iterator?` *x* [Función]
Is *x* a `Music_iterator` object?

`ly:lexer-keywords` *lexer* [Función]
Return a list of `(KEY . CODE)` pairs, signifying the LilyPond reserved words list.

`ly:lily-lexer?` *x* [Función]
Is *x* a `Lily_lexer` object?

`ly:lily-parser?` *x* [Función]
Is *x* a `Lily_parser` object?

`ly:listened-event-class?` *disp cl* [Función]
Does *disp* listen to any event type in the list *cl*?

`ly:listened-event-types` *disp* [Función]
Return a list of all event types that *disp* listens to.

`ly:listener?` *x* [Función]
Is *x* a `Listener` object?

`ly:make-book` *paper header scores* [Función]
Make a `\book` of *paper* and *header* (which may be `#f` as well) containing `\scores`.

`ly:make-book-part` *scores* [Función]
Make a `\bookpart` containing `\scores`.

`ly:make-context-mod` *mod-list* [Función]
Creates a context modification, optionally initialized via the list of modifications *mod-list*.

`ly:make-dispatcher` [Función]
Return a newly created dispatcher.

`ly:make-duration` *length dotcount num den* [Función]
length is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument *dotcount*.

The duration factor is optionally given by integers *num* and *den*, alternatively by a single rational number.

A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.

`ly:make-global-context` *output-def* [Función]
Set up a global interpretation context, using the output block *output-def*. The context is returned.

`ly:make-global-translator` *global* [Función]
Create a translator group and connect it to the global context *global*. The translator group is returned.

`ly:make-listener` *callback* [Función]
Create a listener. Any time the listener hears an object, it will call *callback* with that object.

callback should take exactly one argument.

`ly:make-moment` *m g gn gd* [Función]
Create the moment with rational main timing *m*, and optional grace timing *g*.

A *moment* is a point in musical time. It consists of a pair of rationals (*m*, *g*), where *m* is the timing for the main notes, and *g* the timing for grace notes. In absence of grace notes, *g* is zero.

For compatibility reasons, it is possible to write two numbers specifying numerator and denominator instead of the rationals. These forms cannot be mixed, and the two-argument form is disambiguated by the sign of the second argument: if it is positive, it can only be a denominator and not a grace timing.

`ly:make-music` *props* [Función]
Make a C++ `Music` object and initialize it with *props*.

This function is for internal use and is only called by `make-music`, which is the preferred interface for creating music objects.

`ly:make-music-function` *signature func* [Función]
Make a function to process music, to be used for the parser. *func* is the function, and *signature* describes its arguments. *signature*'s cdr is a list containing either `ly:music?` predicates or other type predicates. Its car is the syntax function to call.

`ly:make-music-relative!` *music pitch* [Función]
Make *music* relative to *pitch*, return final pitch.

`ly:make-output-def` [Función]
Make an output definition.

`ly:make-page-label-marker` *label* [Función]
Return page marker with label *label*.

`ly:make-page-permission-marker` *symbol permission* [Función]
Return page marker with page breaking and turning permissions.

`ly:make-pango-description-string` *chain size* [Función]
Make a `PangoFontDescription` string for the property alist *chain* at size *size*.

`ly:make-paper-outputter` *port format* [Función]
Create an outputter that evaluates within `output-`*format*, writing to *port*.

`ly:make-pitch` *octave note alter* [Función]
octave is specified by an integer, zero for the octave containing middle C. *note* is a number indexing the global default scale, with 0 corresponding to pitch C and 6 usually corresponding to pitch B. Optional *alter* is a rational number of 200-cent whole tones for alteration.

`ly:make-prob` *type init rest* [Función]
Create a `Prob` object.

`ly:make-scale` *steps* [Función]
Create a scale. The argument is a vector of rational numbers, each of which represents the number of 200 cent tones of a pitch above the tonic.

`ly:make-score` *music* [Función]
Return score with *music* encapsulated in it.

`ly:make-simple-closure` *expr* [Función]
Make a simple closure. *expr* should be form of (*`func a1 a2 ...`*), and will be invoked as (*`func delayed-arg a1 a2 ...`*).

`ly:make-spring` *ideal min-dist* [Función]
Make a spring. *ideal* is the ideal distance of the spring, and *min-dist* is the minimum distance.

`ly:make-stencil` *expr xext yext* [Función]
Stencils are device independent output expressions. They carry two pieces of information:

1. A specification of how to print this object. This specification is processed by the output backends, for example '`scm/output-ps.scm`'.
2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use `empty-interval` as its value), it is taken to be empty.

`ly:make-stream-event` *cl proplist* [Función]
Create a stream event of class *cl* with the given mutable property list.

`ly:make-undead` *object* [Función]
This packages *object* in a manner that keeps it from triggering "Parsed object should be dead" messages.

`ly:make-unpure-pure-container` *unpure pure* [Función]
Make an unpure-pure container. *unpure* should be an unpure expression, and *pure* should be a pure expression. If *pure* is omitted, the value of *unpure* will be used twice, except that a callback is given two extra arguments that are ignored for the sake of pure calculations.

`ly:message` *str rest* [Función]
A Scheme callable function to issue the message *str*. The message is formatted with `format` and *rest*.

`ly:minimal-breaking` *pb* [Función]
Break (pages and lines) the `Paper_book` object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.

`ly:mm` *num* [Función]
num mm.

`ly:module->alist` *mod* [Función]
Dump the contents of module *mod* as an alist.

`ly:module-copy` *dest src* [Función]
Copy all bindings from module *src* into *dest*.

`ly:modules-lookup` *modules sym def* [Función]
Look up *sym* in the list *modules*, returning the first occurence. If not found, return *def* or `#f` if *def* isn't specified.

`ly:moment?` *x* [Función]
Is *x* a `Moment` object?

`ly:moment<?` *a b* [Función]
Compare two moments.

`ly:moment-add` *a b* [Función]
Add two moments.

`ly:moment-div` *a b* [Función]
Divide two moments.

`ly:moment-grace` *mom* [Función]
Extract grace timing as a rational number from *mom*.

`ly:moment-grace-denominator` *mom* [Función]
Extract denominator from grace timing.

`ly:moment-grace-numerator` *mom* [Función]
Extract numerator from grace timing.

`ly:moment-main` *mom* [Función]
Extract main timing as a rational number from *mom*.

`ly:moment-main-denominator` *mom* [Función]
Extract denominator from main timing.

`ly:moment-main-numerator` *mom* [Función]
Extract numerator from main timing.

`ly:moment-mod` *a b* [Función]
Modulo of two moments.

`ly:moment-mul` *a b* [Función]
Multiply two moments.

`ly:moment-sub` *a b* [Función]
Subtract two moments.

`ly:music?` *obj* [Función]
Is *obj* a music object?

`ly:music-compress` *m factor* [Función]
Compress music object *m* by moment *factor*.

`ly:music-deep-copy` *m* [Función]
Copy *m* and all sub expressions of *m*. *m* may be an arbitrary type; cons cells and music are copied recursively.

`ly:music-duration-compress` *mus fact* [Función]
Compress *mus* by factor *fact*, which is a `Moment`.

`ly:music-duration-length` *mus* [Función]
Extract the duration field from *mus* and return the length.

`ly:music-function?` *x* [Función]
Is *x* a `music-function`?

`ly:music-function-extract` *x* [Función]
Return the Scheme function inside *x*.

`ly:music-function-signature` *x* [Función]
Return the function signature inside *x*.

`ly:music-length` *mus* [Función]
Get the length of music expression *mus* and return it as a `Moment` object.

`ly:music-list?` *lst* [Función]
Is *lst* a list of music objects?

`ly:music-mutable-properties` *mus* [Función]
Return an alist containing the mutable properties of *mus*. The immutable properties are not available, since they are constant and initialized by the `make-music` function.

`ly:music-output?` *x* [Función]
Is *x* a `Music_output` object?

`ly:music-property` *mus sym val* [Función]
Return the value for property *sym* of music expression *mus*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:music-set-property!` *mus sym val* [Función]
Set property *sym* in music expression *mus* to *val*.

`ly:music-transpose` *m p* [Función]
Transpose *m* such that central C is mapped to *p*. Return *m*.

`ly:note-column-accidentals` *note-column* [Función]
Return the `AccidentalPlacement` grob from *note-column* if any, or `SCM_EOL` otherwise.

`ly:note-column-dot-column` *note-column* [Función]
Return the `DotColumn` grob from *note-column* if any, or `SCM_EOL` otherwise.

`ly:note-head::stem-attachment` *font-metric glyph-name* [Función]
Get attachment in *font-metric* for attaching a stem to notehead *glyph-name*.

`ly:number->string` *s* [Función]
Convert *s* to a string without generating many decimals.

`ly:one-line-breaking` *pb* [Función]
Put each score on a single line, and put each line on its own page. The paper-width setting will be modified so that every page will be wider than the widest line.

`ly:optimal-breaking` *pb* [Función]
Optimally break (pages and lines) the `Paper_book` object *pb* to minimize badness in bother vertical and horizontal spacing.

`ly:option-usage` *port* [Función]
Print `ly:set-option` usage. Optional *port* argumentfor the destination defaults to current output port.

`ly:otf->cff` *otf-file-name* [Función]
Convert the contents of an OTF file to a CFF file, returning it as a string.

`ly:otf-font?` *font* [Función]
Is *font* an OpenType font?

`ly:otf-font-glyph-info` *font glyph* [Función]
Given the font metric *font* of an OpenType font, return the information about named glyph *glyph* (a string).

`ly:otf-font-table-data` *font tag* [Función]
Extract a table *tag* from *font*. Return empty string for non-existent *tag*.

`ly:otf-glyph-count` *font* [Función]
Return the number of glyphs in *font*.

`ly:otf-glyph-list` *font* [Función]
Return a list of glyph names for *font*.

`ly:output-def?` *def* [Función]
Is *def* an output definition?

`ly:output-def-clone` *def* [Función]
Clone output definition *def*.

`ly:output-def-lookup` *def sym val* [Función]
Return the value of *sym* in output definition *def* (e.g., `\paper`). If no value is found, return *val* or `'()` if *val* is undefined.

`ly:output-def-parent` *def* [Función]
Return the parent output definition of *def*.

`ly:output-def-scope` *def* [Función]
Return the variable scope inside *def*.

`ly:output-def-set-variable!` *def sym val* [Función]
Set an output definition *def* variable *sym* to *val*.

`ly:output-description` *output-def* [Función]
Return the description of translators in *output-def*.

`ly:output-find-context-def` *output-def context-name* [Función]
Return an alist of all context defs (matching *context-name*if given) in *output-def*.

`ly:output-formats` [Función]
Formats passed to '`--format`' as a list of strings, used for the output.

`ly:outputter-close` *outputter* [Función]
Close port of *outputter*.

`ly:outputter-dump-stencil` *outputter stencil* [Función]
Dump stencil *expr* onto *outputter*.

`ly:outputter-dump-string` *outputter str* [Función]
Dump *str* onto *outputter*.

`ly:outputter-module` *outputter* [Función]
Return output module of *outputter*.

`ly:outputter-output-scheme` *outputter expr* [Función]
Eval *expr* in module of *outputter*.

`ly:outputter-port` *outputter* [Función]
Return output port for *outputter*.

`ly:page-marker?` *x* [Función]
Is *x* a `Page_marker` object?

`ly:page-turn-breaking` *pb* [Función]
Optimally break (pages and lines) the `Paper_book` object *pb* such that page turns only happen in specified places, returning its pages.

`ly:pango-font?` *f* [Función]
Is *f* a pango font?

`ly:pango-font-physical-fonts` *f* [Función]
Return alist of `(ps-name file-name font-index)` lists for Pango font *f*.

`ly:paper-book?` *x* [Función]
Is *x* a `Paper_book` object?

`ly:paper-book-header` *pb* [Función]
Return the header definition (`\header`) in `Paper_book` object *pb*.

`ly:paper-book-pages` *pb* [Función]
Return pages in `Paper_book` object *pb*.

`ly:paper-book-paper` *pb* [Función]
Return the paper output definition (`\paper`) in `Paper_book` object *pb*.

`ly:paper-book-performances` *pb* [Función]
Return performances in `Paper_book` object *pb*.

`ly:paper-book-scopes` *pb* [Función]
Return scopes in `Paper_book` object *pb*.

`ly:paper-book-systems` *pb* [Función]
Return systems in `Paper_book` object *pb*.

`ly:paper-fonts` *def* [Función]
Return a list containing the fonts from output definition *def* (e.g., `\paper`).

`ly:paper-get-font` *def chain* [Función]
Find a font metric in output definition *def* satisfying the font-qualifiers in alist chain *chain*, and return it. (An alist chain is a list of alists, containing grob properties.)

`ly:paper-get-number` *def sym* [Función]
Return the value of variable *sym* in output definition *def* as a double.

`ly:paper-outputscale` *def* [Función]
Return the output-scale for output definition *def*.

`ly:paper-score-paper-systems` *paper-score* [Función]
Return vector of `paper_system` objects from *paper-score*.

`ly:paper-system?` *obj* [Función]
Is *obj* a C++ `Prob` object of type `paper-system`?

`ly:paper-system-minimum-distance` *sys1 sys2* [Función]
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.

`ly:parse-file` *name* [Función]
Parse a single `.ly` file. Upon failure, throw `ly-file-failed` key.

`ly:parse-string-expression` *parser-smob ly-code filename line* [Función]
Parse the string *ly-code* with *parser-smob*. Return the contained music expression. *filename* and *line* are optional source indicators.

`ly:parsed-undead-list!` [Función]
Return the list of objects that have been found live that should have been dead, and clear that list.

`ly:parser-clear-error` *parser* [Función]
Clear the error flag for the parser.

`ly:parser-clone` *parser-smob closures location* [Función]
Return a clone of *parser-smob*. An association list of port positions to closures can be specified in *closures* in order to have `$` and `#` interpreted in their original lexical environment. If *location* is a valid location, it becomes the source of all music expressions inside.

`ly:parser-define!` *parser-smob symbol val* [Función]
Bind *symbol* to *val* in *parser-smob*'s module.

`ly:parser-error` *parser msg input* [Función]
Display an error message and make the parser fail.

`ly:parser-has-error?` *parser* [Función]
Does *parser* have an error flag?

`ly:parser-include-string` *parser-smob ly-code* [Función]
Include the string *ly-code* into the input stream for *parser-smob*. Can only be used in immediate Scheme expressions (`$` instead of `#`).

`ly:parser-lexer` *parser-smob* [Función]
Return the lexer for *parser-smob*.

`ly:parser-lookup` *parser-smob symbol* [Función]
Look up *symbol* in *parser-smob*'s module. Return `'()` if not defined.

`ly:parser-output-name` *parser* [Función]
Return the base name of the output file.

`ly:parser-parse-string` *parser-smob ly-code* [Función]
Parse the string *ly-code* with *parser-smob*. Upon failure, throw `ly-file-failed` key.

`ly:parser-set-note-names` *parser names* [Función]
Replace current note names in *parser*. *names* is an alist of symbols. This only has effect if the current mode is notes.

`ly:performance-write` *performance filename* [Función]
Write *performance* to *filename*.

`ly:pfb->pfa` *pfb-file-name* [Función]
Convert the contents of a Type 1 font in PFB format to PFA format.

`ly:pitch?` *x* [Función]
Is *x* a `Pitch` object?

`ly:pitch<?` *p1 p2* [Función]
Is *p1* lexicographically smaller than *p2*?

`ly:pitch-alteration` *pp* [Función]
Extract the alteration from pitch *pp*.

`ly:pitch-diff` *pitch root* [Función]
Return pitch *delta* such that *pitch* transposed by *delta* equals *root*.

`ly:pitch-negate` *p* [Función]
Negate *p*.

`ly:pitch-notename` *pp* [Función]
Extract the note name from pitch *pp*.

`ly:pitch-octave` *pp* [Función]
Extract the octave from pitch *pp*.

`ly:pitch-quartertones` *pp* [Función]
Calculate the number of quarter tones of *pp* from middle C.

`ly:pitch-semitones` *pp* [Función]
Calculate the number of semitones of *pp* from middle C.

`ly:pitch-steps` *p* [Función]
Number of steps counted from middle C of the pitch *p*.

`ly:pitch-tones` *pp* [Función]
Calculate the number of tones of *pp* from middle C as a rational number.

`ly:pitch-transpose` *p delta* [Función]
Transpose *p* by the amount *delta*, where *delta* is relative to middle C.

`ly:pointer-group-interface::add-grob` *grob sym grob-element* [Función]
Add *grob-element* to *grob*'s *sym* grob array.

`ly:position-on-line?` *sg spos* [Función]
Return whether *spos* is on a line of the staff associated with the grob *sg* (even on an extender line).

`ly:prob?` *x* [Función]
Is *x* a `Prob` object?

`ly:prob-immutable-properties` *prob* [Función]
Retrieve an alist of immutable properties.

`ly:prob-mutable-properties` *prob* [Función]
Retrieve an alist of mutable properties.

`ly:prob-property` *prob sym val* [Función]
Return the value for property *sym* of Prob object *prob*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:prob-property?` *obj sym* [Función]
Is boolean prop *sym* of *sym* set?

`ly:prob-set-property!` *obj sym value* [Función]
Set property *sym* of *obj* to *value*.

`ly:prob-type?` *obj type* [Función]
Is *obj* the specified prob-type?

`ly:programming-error` *str rest* [Función]
A Scheme callable function to issue the internal warning *str*. The message is formatted with `format` and *rest*.

`ly:progress` *str rest* [Función]
A Scheme callable function to print progress *str*. The message is formatted with `format` and *rest*.

`ly:property-lookup-stats` *sym* [Función]
Return hash table with a property access corresponding to *sym*. Choices are `prob`, `grob`, and `context`.

`ly:protects` [Función]
Return hash of protected objects.

`ly:pt` *num* [Función]
num printer points.

`ly:register-stencil-expression` *symbol* [Función]
Add *symbol* as head of a stencil expression.

`ly:relative-group-extent` *elements common axis* [Función]
Determine the extent of *elements* relative to *common* in the *axis* direction.

`ly:reset-all-fonts` [Función]
Forget all about previously loaded fonts.

`ly:round-filled-box` *xext yext blot* [Función]
Make a `Stencil` object that prints a black box of dimensions *xext*, *yext* and roundness *blot*.

`ly:round-filled-polygon` *points blot* [Función]
Make a `Stencil` object that prints a black polygon with corners at the points defined by *points* (list of coordinate pairs) and roundness *blot*.

`ly:run-translator` *mus output-def* [Función]
Process *mus* according to *output-def*. An interpretation context is set up, and *mus* is interpreted with it. The context is returned in its final state.

Optionally, this routine takes an object-key to to uniquely identify the score block containing it.

`ly:score?` *x* [Función]
Is *x* a `Score` object?

`ly:score-add-output-def!` *score def* [Función]
Add an output definition *def* to *score*.

`ly:score-embedded-format` *score layout* [Función]
Run *score* through *layout* (an output definition) scaled to correct output-scale already, returning a list of layout-lines.

`ly:score-error?` *score* [Función]
Was there an error in the score?

`ly:score-header` *score* [Función]
Return score header.

`ly:score-music` *score* [Función]
Return score music.

`ly:score-output-defs` *score* [Función]
All output definitions in a score.

`ly:score-set-header!` *score module* [Función]
Set the score header.

`ly:set-default-scale` *scale* [Función]
Set the global default scale. This determines the tuning of pitches with no accidentals or key signatures. The first pitch is C. Alterations are calculated relative to this scale. The number of pitches in this scale determines the number of scale steps that make up an octave. Usually the 7-note major scale.

`ly:set-grob-modification-callback` *cb* [Función]
Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++ file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property.

`ly:set-middle-C!` *context* [Función]
Set the `middleCPosition` variable in *context* based on the variables `middleCClefPosition` and `middleCOffset`.

`ly:set-option` *var val* [Función]
Set a program option.

`ly:set-property-cache-callback` *cb* [Función]
Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property.

`ly:simple-closure?` *clos* [Función]
Is *clos* a simple closure?

`ly:skyline?` *x* [Función]
Is *x* a `Skyline` object?

`ly:skyline-empty?` *sky* [Función]
Return whether *sky* is empty.

`ly:skyline-pair?` *x* [Función]
Is *x* a `Skyline_pair` object?

`ly:slur-score-count` [Función]
count number of slur scores.

`ly:smob-protects` [Función]
Return LilyPond's internal smob protection list.

`ly:solve-spring-rod-problem` *springs rods length ragged* [Función]
Solve a spring and rod problem for *count* objects, that are connected by *count*-1 *springs*, and an arbitrary number of *rods*. *count* is implicitly given by *springs* and *rods*. The *springs* argument has the format `(ideal, inverse_hook)` and *rods* is of the form `(idx1, idx2, distance)`.

length is a number, *ragged* a boolean.

The function returns a list containing the force (positive for stretching, negative for compressing and `#f` for non-satisfied constraints) followed by *spring-count*+1 positions of the objects.

`ly:source-file?` *x* [Función]
Is *x* a `Source_file` object?

`ly:spanner?` *g* [Función]
Is *g* a spanner object?

`ly:spanner-bound` *spanner dir* [Función]
Get one of the bounds of *spanner*. *dir* is `-1` for left, and `1` for right.

`ly:spanner-broken-into` *spanner* [Función]
Return broken-into list for *spanner*.

`ly:spanner-set-bound!` *spanner dir item* [Función]
Set grob *item* as bound in direction *dir* for *spanner*.

`ly:spawn` *command rest* [Función]
Simple interface to g_spawn_sync *str*. The error is formatted with `format` and *rest*.

`ly:spring?` *x* [Función]
Is *x* a `Spring` object?

`ly:spring-set-inverse-compress-strength!` *spring strength* [Función]
Set the inverse compress *strength* of *spring*.

`ly:spring-set-inverse-stretch-strength!` *spring strength* [Función]
Set the inverse stretch *strength* of *spring*.

`ly:staff-symbol-line-thickness` *grob* [Función]
Returns the `line-thickness` of the staff associated with *grob*.

`ly:staff-symbol-staff-radius` *grob* [Función]
Returns the radius of the staff associated with *grob*.

`ly:staff-symbol-staff-space` *grob* [Función]
Returns the `staff-space` of the staff associated with *grob*.

`ly:start-environment` [Función]
Return the environment (a list of strings) that was in effect at program start.

`ly:stderr-redirect` *file-name mode* [Función]
Redirect stderr to *file-name*, opened with *mode*.

`ly:stencil?` *x* [Función]
Is *x* a `Stencil` object?

`ly:stencil-add` *args* [Función]
Combine stencils. Takes any number of arguments.

`ly:stencil-aligned-to` *stil axis dir* [Función]
Align *stil* using its own extents. *dir* is a number. `-1` and `1` are left and right, respectively. Other values are interpolated (so `0` means the center).

`ly:stencil-combine-at-edge` *first axis direction second padding* [Función]
Construct a stencil by putting *second* next to *first*. *axis* can be 0 (x-axis) or 1 (y-axis). *direction* can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with *padding* as extra space. *first* and *second* may also be `'()` or `#f`.

`ly:stencil-empty?` *stil axis* [Función]
Return whether *stil* is empty. If an optional *axis* is supplied, the emptiness check is restricted to that axis.

`ly:stencil-expr` *stil* [Función]
Return the expression of *stil*.

`ly:stencil-extent` *stil axis* [Función]
Return a pair of numbers signifying the extent of *stil* in *axis* direction (`0` or `1` for x and y axis, respectively).

`ly:stencil-fonts` *s* [Función]
Analyze *s*, and return a list of fonts used in *s*.

`ly:stencil-in-color` *stc r g b* [Función]
Put *stc* in a different color.

`ly:stencil-rotate` *stil angle x y* [Función]
Return a stencil *stil* rotated *angle* degrees around the relative offset (*x*, *y*). E.g., an offset of (-1, 1) will rotate the stencil around the left upper corner.

`ly:stencil-rotate-absolute` *stil angle x y* [Función]
Return a stencil *stil* rotated *angle* degrees around point (*x*, *y*), given in absolute coordinates.

`ly:stencil-scale` *stil x y* [Función]
Scale *stil* using the horizontal and vertical scaling factors *x* and *y*.

`ly:stencil-stack` *first axis direction second padding mindist* [Función]
Construct a stencil by stacking *second* next to *first*. *axis* can be 0 (x-axis) or 1 (y-axis). *direction* can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with *padding* as extra space. *first* and *second* may also be `'()` or `#f`. As opposed to `ly:stencil-combine-at-edge`, metrics are suited for successively accumulating lines of stencils. Also, *second* stencil is drawn last.

If *mindist* is specified, reference points are placed apart at least by this distance. If either of the stencils is spacing, *padding* and *mindist* do not apply.

`ly:stencil-translate` *stil offset* [Función]
Return a *stil*, but translated by *offset* (a pair of numbers).

`ly:stencil-translate-axis` *stil amount axis* [Función]
Return a copy of *stil* but translated by *amount* in *axis* direction.

`ly:stream-event?` *obj* [Función]
Is *obj* a `Stream_event` object?

`ly:string-percent-encode` *str* [Función]
Encode all characters in string *str* with hexadecimal percent escape sequences, with the following exceptions: characters `-`, `.`, `/`, and `_`; and characters in ranges `0-9`, `A-Z`, and `a-z`.

`ly:string-substitute` *a b s* [Función]
Replace string *a* by string *b* in string *s*.

`ly:system-font-load` *name* [Función]
Load the OpenType system font '*name*`.otf`'. Fonts loaded with this command must contain three additional SFNT font tables called `LILC`, `LILF`, and `LILY`, needed for typesetting musical elements. Currently, only the Emmentaler and the Emmentaler-Brace fonts fulfill these requirements.

Note that only `ly:font-get-glyph` and derived code (like `\lookup`) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.

`ly:text-interface::interpret-markup` [Función]
Convert a text markup into a stencil. Takes three arguments, *layout*, *props*, and *markup*.

layout is a `\layout` block; it may be obtained from a grob with `ly:grob-layout`. *props* is an alist chain, i.e. a list of alists. This is typically obtained with `(ly:grob-alist-chain grob (ly:output-def-lookup layout 'text-font-defaults))`. *markup* is the markup text to be processed.

`ly:translate-cpp-warning-scheme` *str* [Función]
Translates a string in C++ printf format and modifies it to use it for scheme formatting.

`ly:translator?` *x* [Función]
Is *x* a `Translator` object?

`ly:translator-context` *trans* [Función]
Return the context of the translator object *trans*.

`ly:translator-description` *me* [Función]
Return an alist of properties of translator *me*.

`ly:translator-group?` *x* [Función]
Is *x* a `Translator_group` object?

`ly:translator-name` *trans* [Función]
Return the type name of the translator object *trans*. The name is a symbol.

`ly:transpose-key-alist` *l pit* [Función]
Make a new key alist of *l* transposed by pitch *pit*.

`ly:truncate-list!` *lst i* [Función]
Take at most the first *i* of list *lst*.

`ly:ttf->pfa` *ttf-file-name idx* [Función]
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:ttf-ps-name` *ttf-file-name idx* [Función]
Extract the PostScript name from a TrueType font. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:undead?` *x* [Función]
Is *x* a `Undead` object?

`ly:unit` [Función]
Return the unit used for lengths as a string.

`ly:unpure-pure-container?` *clos* [Función]
Is *clos* an unpure pure container?

`ly:unpure-pure-container-pure-part` *pc* [Función]
Return the pure part of *pc*.

`ly:unpure-pure-container-unpure-part` *pc* [Función]
Return the unpure part of *pc*.

`ly:usage` [Función]
Print usage message.

`ly:verbose-output?` [Función]
Was verbose output requested, i.e. loglevel at least `DEBUG`?
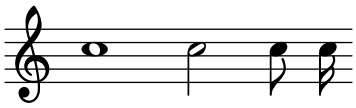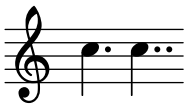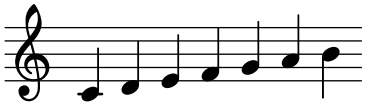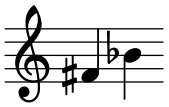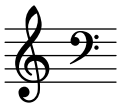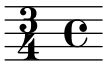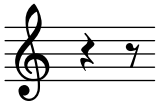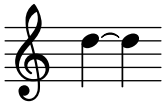
`ly:version` [Función]
Return the current lilypond version as a list, e.g., `(1 3 127 uu1)`.

`ly:warning` *str rest* [Función]
A Scheme callable function to issue the warning *str*. The message is formatted with `format` and *rest*.
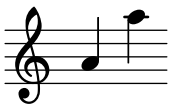
`ly:warning-located` *location str rest* [Función]
A Scheme callable function to issue the warning *str* at the specified location in an input file. The message is formatted with `format` and *rest*.

`ly:wide-char->utf-8` *wc* [Función]
Encode the Unicode codepoint *wc*, an integer, as UTF-8.
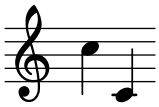
Apéndice B Hoja de referencia rápida

Sintaxis	Descripción	Ejemplo
`1 2 8 16`	duraciones	
`c4. c4..`	puntillos	
`c d e f g a b`	escala	
`fis bes`	alteración	
`\clef treble \clef bass`	claves	
`\time 3/4 \time 4/4`	compás	
`r4 r8`	silencio	
`d ~ d`	ligadura de unión	
`\key es \major`	armadura	
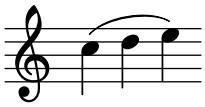
note`'` subir una octava
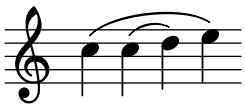


note`,` bajar una octava
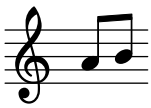


`c( d e)` ligadura de expresión



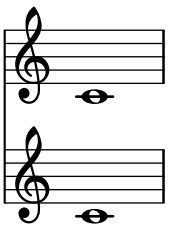
`c\( c( d) e\)` ligadura de fraseo
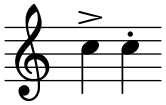


`a8[ b]` barra
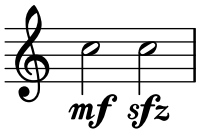


`<< \new Staff ... >>` más pentagramas



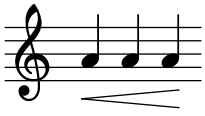
`c-> c-.` articulaciones
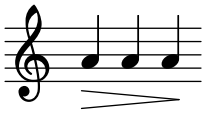


`c2\mf c\sfz` matices dinámicos
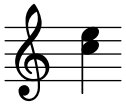


`a\< a a\!` crescendo
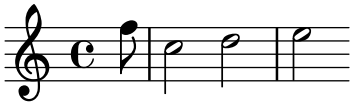


`a\> a a\!` decrescendo
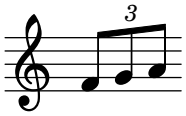


`< >` acorde
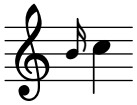


`\partial 8` anacrusa o compás incompleto



`\tuplet 3/2 {f g a}` tresillos



`\grace` mordentes

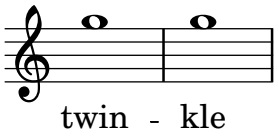


`\lyricmode { twinkle }` escribir la letra twinkle
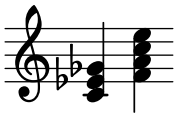
`\new Lyrics` imprimir la letra

twinkle

`twin -- kle` separador de sílabas
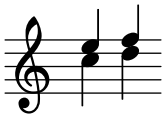


`\chordmode { c:dim f:maj7 }` acordes



`\context ChordNames` imprimir los nombres de los acordes

C° F△

`<<{e f} \\ {c d}>>` polifonía



`s4` `s8` `s16`	silencios de separación

Apéndice C GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

   The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Apéndice D Índice de instrucciones de LilyPond

Este índice relaciona todas las instrucciones y palabras clave de LilyPond, con enlaces a aquellas secciones del manual que describen o se ocupan de su uso. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece la instrucción o palabra clave; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que aparece la instrucción o palabra clave.

!

! 6

"

"|" 106

'

' 1

,

, ... 1

-

- 116

.

. 43

/

/ 404
/+ 404

:

: 159

<

< 161
<...> 161

=

= 9

>

> 161

?

? 6

[

[..... 90

]

] 90

^

^ 403

_

_ 258

\

\! 119
\(..... 130
\) 130
\< 119
\> 119
\abs-fontsize 235, 671
\absolute 2
\accent 116
\accepts 582, 584
\acciaccatura 108
\accidentalStyle 26
\addChordShape 361
\addInstrumentDefinition 203, 211
\addlyrics 253, 254, 255
\addQuote 204
\aeolian 20
\afterGrace 109
\aikenHeads 39
\aikenHeadsMinor 39
\alias 582
\allowPageTurn 532
\alterBroken 626
\alternative 144
\appendToTag 494
\appoggiatura 108
\arpeggio 138
\arpeggioArrowDown 138
\arpeggioArrowUp 138
\arpeggioBracket 139
\arpeggioNormal 138
\arpeggioParenthesis 139
\arpeggioParenthesisDashed 139
\arrow-head 242, 695
\ascendens 437, 443
\auctum 437, 443
\augmentum 443
\auto-footnote 716
\autoBeamOff 79
\autoBeamOff 320
\autoBeamOn 79
\autochange 318

\backslashed-digit ... 716
\balloonGrobText ... 221
\balloonLengthOff ... 221
\balloonLengthOn ... 221
\balloonText ... 221
\bar ... 94, 100
\barNumberCheck ... 106
\beam ... 695
\bendAfter ... 133
\bold ... 234, 672
\book ... 460, 463
\bookOutputName ... 462
\bookOutputSuffix ... 462
\bookpart ... 461, 463, 530
\box ... 240, 672
\bracket ... 125, 240, 695
\break ... 529
\breathe ... 131
\breve ... 43, 54
\cadenzaOff ... 71
\cadenzaOn ... 71
\caesura ... 435
\caps ... 672
\cavum ... 437, 443
\center-align ... 237, 680
\center-column ... 239, 680
\change ... 317
\char ... 716
\chordmode ... 5, 13, 359
\chordRepeats ... 334
\chords ... 406
\circle ... 240, 695
\clef ... 16
\cm ... 602
\coda ... 116
\column ... 239, 681
\column-lines ... 722
\combine ... 242, 681
\compoundMeter ... 74
\compressFullBarRests ... 59
\compressFullBarRests ... 60
\concat ... 681
\consists ... 582
\context ... 569, 578
\cr ... 119
\cresc ... 120
\crescHairpin ... 120
\crescTextCresc ... 120
\crossStaff ... 320
\cueClef ... 207
\cueDuring ... 207
\cueDuringWithClef ... 207
\customTabClef ... 702
\decr ... 119
\decresc ... 120
\defaultchild ... 586
\defaultTimeSignature ... 63
\defineBarLine ... 98
\deminutum ... 437, 443
\denies ... 582, 584
\descendens ... 437, 443
\dim ... 120
\dimHairpin ... 120
\dimTextDecr ... 120
\dimTextDecresc ... 120
\dimTextDim ... 120
\dir-column ... 682
\discant ... 711
\displayLilyMusic ... 511
\divisioMaior ... 435
\divisioMaxima ... 435
\divisioMinima ... 435
\dorian ... 20
\dotsDown ... 44
\dotsNeutral ... 44
\dotsUp ... 44
\doubleflat ... 702
\doublesharp ... 703
\downbow ... 116, 327
\downmordent ... 116
\downprall ... 116
\draw-circle ... 242, 696
\draw-dashed-line ... 696
\draw-dotted-line ... 696
\draw-hline ... 697
\draw-line ... 242, 697
\drummode ... 183
\dynamic ... 125, 672
\dynamicDown ... 121
\dynamicNeutral ... 121
\dynamicUp ... 121
\easyHeadsOff ... 37
\easyHeadsOn ... 37
\ellipse ... 697
\epsfile ... 242, 698
\espressivo ... 116, 120
\expandFullBarRests ... 59
\expandFullBarRests ... 60
\eyeglasses ... 717
\f ... 119
\featherDurations ... 93
\fermata ... 116, 703
\fermataMarkup ... 59
\fermataMarkup ... 60, 116
\ff ... 119
\fff ... 119
\ffff ... 119
\fffff ... 119
\fill-line ... 239, 682
\fill-with-pattern ... 683
\filled-box ... 242, 698
\finalis ... 435
\finger ... 214, 673
\flageolet ... 116
\flat ... 703
\flexa ... 443
\fontCaps ... 673
\fontsize ... 235, 673
\footnote ... 478, 717
\fp ... 119
\fraction ... 717
\freeBass ... 712
\frenchChords ... 410
\fret-diagram ... 349, 708
\fret-diagram-terse ... 351, 708
\fret-diagram-verbose ... 353, 709
\fromproperty ... 717
\funkHeads ... 39
\funkHeadsMinor ... 39
\general-align ... 238, 684

\germanChords ... 410
\glissando ... 134
\grace ... 108
\halfopen ... 116
\halign ... 237, 684
\harmonic ... 327, 336
\harmonicByFret ... 336
\harmonicByRatio ... 336
\harmonicsOff ... 327
\harmonicsOn ... 327
\harp-pedal ... 709
\hbracket ... 240, 698
\hcenter-in ... 685
\header ... 463
\hide ... 609
\hideKeySignature ... 389
\hideNotes ... 216
\hideSplitTiedTabNotes ... 335
\hideStaffSwitch ... 320
\hspace ... 686
\huge ... 213, 236, 673
\improvisationOff ... 41, 77
\improvisationOn ... 41, 77
\in ... 602
\inclinatum ... 437, 443
\include ... 490
\inStaffSegno ... 147
\instrumentSwitch ... 203
\inversion ... 13
\ionian ... 20
\italianChords ... 410
\italic ... 234, 673
\justified-lines ... 245, 722
\justify ... 239, 687
\justify-field ... 686
\justify-string ... 687
\keepWithTag ... 494
\key ... 20, 39
\killCues ... 211
\label ... 488
\laissezVibrer ... 52
\large ... 213, 236, 674
\larger ... 235, 236, 674
\layout ... 463, 524, 567, 578
\left-align ... 237, 688
\left-brace ... 718
\left-column ... 688
\lheel ... 116
\line ... 688
\linea ... 437, 443
\lineprall ... 116
\locrian ... 20
\longa ... 43, 54
\longfermata ... 116
\lookup ... 718
\lower ... 237, 689
\ltoe ... 116
\lydian ... 20
\lyricmode ... 251, 253
\lyricsto ... 253, 254, 255
\magnify ... 235, 674
\major ... 20
\makeClusters ... 166
\map-markup-commands ... 722
\marcato ... 116
\mark ... 106, 229
\markalphabet ... 718
\markLengthOff ... 68, 230
\markLengthOn ... 68, 230
\markletter ... 718
\markup ... 229, 231, 232, 233
\markuplist ... 232, 245, 246
\maxima ... 43, 54
\medium ... 674
\melisma ... 259
\melismaEnd ... 259
\mergeDifferentlyDottedOff ... 170
\mergeDifferentlyDottedOn ... 170
\mergeDifferentlyHeadedOff ... 170
\mergeDifferentlyHeadedOn ... 170
\mf ... 119
\midi ... 463, 567
\minor ... 20
\mixolydian ... 20
\mm ... 602
\modalInversion ... 15
\modalTranspose ... 14
\mordent ... 116
\mp ... 119
\musicglyph ... 108, 703
\name ... 582
\natural ... 703
\new ... 569
\newSpacingSection ... 554
\noBeam ... 90
\noBreak ... 529
\noPageBreak ... 530
\noPageTurn ... 532
\normal-size-sub ... 675
\normal-size-super ... 235, 675
\normal-text ... 675
\normalsize ... 213, 236, 676
\note ... 704
\note-by-number ... 704
\null ... 237, 719
\number ... 676
\numericTimeSignature ... 63
\octaveCheck ... 9
\omit ... 608
\on-the-fly ... 477, 719
\once ... 594
\oneVoice ... 166
\open ... 116, 327
\oriscus ... 437, 443
\ottava ... 23
\oval ... 699
\override ... 593, 597, 719
\override-lines ... 723
\overrideProperty ... 597
\overrideTimeSignatureSettings ... 63
\p ... 119
\pad-around ... 241, 689
\pad-markup ... 241, 689
\pad-to-box ... 241, 690
\pad-x ... 241, 690
\page-link ... 719
\page-ref ... 488, 719
\pageBreak ... 530
\pageTurn ... 532
\paper ... 463, 514

\parallelMusic ... 180
\parenthesize ... 219, 699
\partcombine ... 175, 280
\partcombineApart ... 176
\partcombineAutomatic ... 176
\partcombineChords ... 176
\partcombineSoloI ... 176
\partcombineSoloII ... 176
\partcombineUnisono ... 176
\partial ... 70, 144, 146
\path ... 700
\pattern ... 720
\pes ... 443
\phrasingSlurDashed ... 130
\phrasingSlurDashPattern ... 130
\phrasingSlurDotted ... 130
\phrasingSlurDown ... 130
\phrasingSlurHalfDashed ... 130
\phrasingSlurHalfSolid ... 130
\phrasingSlurNeutral ... 130
\phrasingSlurSolid ... 130
\phrasingSlurUp ... 130
\phrygian ... 20
\pitchedTrill ... 142
\portato ... 116
\postscript ... 242, 701
\powerChords ... 374
\pp ... 119
\ppp ... 119
\pppp ... 119
\prall ... 116
\pralldown ... 116
\prallmordent ... 116
\prallprall ... 116
\prallup ... 116
\predefinedFretboardsOff ... 368
\predefinedFretboardsOn ... 368
\property-recursive ... 720
\pt ... 602
\pushToTag ... 494
\put-adjacent ... 690
\quilisma ... 437, 443
\quoteDuring ... 204, 207
\raise ... 237, 690
\relative ... 2, 5, 13, 319
\RemoveEmptyStaves ... 198, 199
\removeWithTag ... 494
\repeat ... 144
\repeat percent ... 156
\repeat tremolo ... 159
\repeatTie ... 52, 147, 276
\replace ... 676
\rest ... 54, 705
\rest-by-number ... 704
\retrograde ... 14
\reverseturn ... 116
\revert ... 593
\revertTimeSignatureSettings ... 64
\rfz ... 119
\rheel ... 116
\right-align ... 237, 691
\right-brace ... 720
\right-column ... 691
\rightHandFinger ... 370
\roman ... 676
\rotate ... 691
\rounded-box ... 240, 701
\rtoe ... 116
\sacredHarpHeads ... 39
\sacredHarpHeadsMinor ... 39
\sans ... 677
\scale ... 702
\scaleDurations ... 50, 72
\score ... 459, 463, 705
\segno ... 116
\semiflat ... 706
\semiGermanChords ... 410
\semisharp ... 707
\sesquiflat ... 707
\sesquisharp ... 707
\set ... 81
\set ... 590, 597
\sf ... 119
\sff ... 119
\sfz ... 119
\shape ... 623
\sharp ... 707
\shiftOff ... 170
\shiftOn ... 170
\shiftOnn ... 170
\shiftOnnn ... 170
\shortfermata ... 116
\showKeySignature ... 389
\showStaffSwitch ... 320
\signumcongruentiae ... 116
\simple ... 677
\skip ... 56
\skip ... 275
\slashed-digit ... 720
\slashedGrace ... 108
\slurDashed ... 127
\slurDashPattern ... 128
\slurDotted ... 127
\slurDown ... 127
\slurHalfDashed ... 128
\slurHalfSolid ... 128
\slurNeutral ... 127
\slurSolid ... 127
\slurUp ... 128
\small ... 213, 236, 677
\smallCaps ... 677
\smaller ... 235, 236, 678
\snappizzicato ... 116
\sostenutoOff ... 322
\sostenutoOn ... 322
\southernHarmonyHeads ... 39
\southernHarmonyHeadsMinor ... 39
\sp ... 119
\spp ... 119
\staccatissimo ... 116
\staccato ... 116
\startGroup ... 224
\startStaff ... 191, 194
\startTrillSpan ... 141
\stdBass ... 712
\stdBassIV ... 713
\stdBassV ... 714
\stdBassVI ... 715
\stemDown ... 219
\stemNeutral ... 219

\stemUp 219
\stencil 721
\stopGroup 224
\stopped 116
\stopStaff 191, 194, 198
\stopTrillSpan 141
\storePredefinedDiagram 361
\stringTuning 345
\stropha 437, 443
\strut 721
\sub 235, 678
\super 235, 678
\sustainOff 322
\sustainOn 322
\tabChordRepeats 334
\tabFullNotation 332
\table-of-contents 490, 723
\tag 494
\taor 389
\teeny 213, 236, 679
\tempo 66
\tenuto 116
\text 679
\textLengthOff 60, 226
\textLengthOn 60, 226
\textSpannerDown 227
\textSpannerNeutral 227
\textSpannerUp 227
\thumb 116, 214
\tied-lyric 707
\tieDashed 52
\tieDotted 52
\tieDown 52
\tieNeutral 52
\tieSolid 52
\tieUp 52
\time 62
\time 81
\tiny 213, 236, 679
\tocItem 490
\translate 238, 692
\translate-scaled 238, 692
\transparent 721
\transpose 5, 10, 13
\transposedCueDuring 210
\transposition 24, 204
\treCorde 322
\triangle 242, 702
\trill 116, 141
\tuplet 45, 72
\tupletDown 46
\tupletNeutral 46
\tupletUp 46
\turn 116
\tweak 595, 597
\type 582
\typewriter 679
\unaCorda 322
\underline 234, 679
\unfoldRepeats 505
\unHideNotes 216
\unset 592
\upbow 116, 327
\upmordent 116
\upprall 116
\upright 680
\varcoda 116
\vcenter 692
\verbatim-file 721
\verylongfermata 116
\virga 437, 443
\virgula 435
\voiceFourStyle 169
\voiceNeutralStyle 169
\voiceOne 166
\voiceOne ... \voiceFour 166
\voiceOneStyle 169
\voiceThreeStyle 169
\voiceTwoStyle 169
\vspace 692
\walkerHeads 39
\walkerHeadsMinor 39
\whiteout 721
\with 576, 580
\with-color 217, 721
\with-dimensions 722
\with-link 722
\with-url 702
\woodwind-diagram 710
\wordwrap 239, 693
\wordwrap-field 693
\wordwrap-internal 723
\wordwrap-lines 245, 723
\wordwrap-string 694
\wordwrap-string-internal 723

|

| 105, 106

~

~ 51

A

absolute 2, 762
accepts 582
acciaccatura 762
accidentalStyle 762
addChordShape 361, 762
addInstrumentDefinition 203, 211, 762
additionalPitchPrefix 408
addQuote 204, 762
aeolian 20
afterGrace 109, 762
aikenHeads 39
aikenHeadsMinor 39
alias 582
alignAboveContext 586
alignBelowContext 274, 586
allowPageTurn 762
allowVoltaHook 762
alterBroken 762
annotate-spacing 564
appendToTag 762
applyContext 762
applyMusic 762
applyOutput 762

appoggiatura ... 763
arpeggio ... 138
arpeggioArrowDown ... 138
arpeggioArrowUp ... 138
arpeggioBracket ... 139
arpeggioNormal ... 138
arpeggioParenthesis ... 139
arpeggioParenthesisDashed ... 139
arrow-head ... 242
assertBeamQuant ... 763
assertBeamSlope ... 763
aug ... 401
auto-first-page-number ... 523
autoBeaming ... 81
autoBeaming ... 568
autoBeamOff ... 79
autoBeamOn ... 79
autochange ... 318, 763

B

Balloon_engraver ... 221
balloonGrobText ... 221, 763
balloonLengthOff ... 221
balloonLengthOn ... 221
balloonText ... 221, 763
banjo-c-tuning ... 376
banjo-modal-tuning ... 376
banjo-open-d-tuning ... 376
banjo-open-dm-tuning ... 376
bar ... 94, 100, 763
barCheckSynchronize ... 105
BarNumber ... 101
barNumberCheck ... 106, 763
barNumberVisibility ... 101
bartype ... 100
base-shortest-duration ... 553
baseMoment ... 81
beamExceptions ... 81
beatStructure ... 81
bendAfter ... 133, 763
binding-offset ... 520
blank-after-score-page-penalty ... 523
blank-last-page-penalty ... 522
blank-page-penalty ... 522
bold ... 234
bookOutputName ... 763
bookOutputSuffix ... 763
bookTitleMarkup ... 473
bottom-margin ... 516
box ... 240
bracket ... 125, 240
bracket ... 323
breakable ... 80
breathe ... 131
breathe ... 763
breve ... 43, 54

C

cadenzaOff ... 71
cadenzaOn ... 71
center-align ... 237
center-column ... 239
change ... 317
check-consistency ... 520
chordChanges ... 406
chordmode ... 5, 13, 359
chordNameExceptions ... 409
chordNameLowercaseMinor ... 408
ChordNames ... 359
chordNameSeparator ... 408
chordNoteNamer ... 408
chordPrefixSpacer ... 410
chordRepeats ... 763
chordRootNamer ... 408
circle ... 240
clef ... 16, 763
color ... 217
column ... 239
combine ... 242
common-shortest-duration ... 553
Completion_heads_engraver ... 75
Completion_rest_engraver ... 75
compoundMeter ... 763
compressFullBarRests ... 59
compressFullBarRests ... 60
consists ... 582
controlpitch ... 9
cr ... 119
cresc ... 120
crescHairpin ... 120
crescTextCresc ... 120
cross ... 35
crossStaff ... 763
cueClef ... 207, 763
cueClefUnset ... 763
cueDuring ... 207, 764
cueDuringWithClef ... 207, 764
currentBarNumber ... 101, 114

D

deadNote ... 764
decr ... 119
decresc ... 120
default ... 26, 28
default-staff-staff-spacing ... 536
defaultBarType ... 100
defaultNoteHeads ... 764
defaultTimeSignature ... 63
defineBarLine ... 98, 764
denies ... 582
dim ... 120
dim ... 401
dimHairpin ... 120
dimTextDecr ... 120
dimTextDecresc ... 120
dimTextDim ... 120
displayLilyMusic ... 764
displayMusic ... 764
displayScheme ... 764
disposición, archivo de ... 526
dodecaphonic ... 31
dorian ... 20
dotsDown ... 44
dotsNeutral ... 44
dotsUp ... 44

draw-circle 242
draw-line 242
drummode 183
DrumStaff 183
dynamic 125
dynamicDown 121
DynamicLineSpanner 121
dynamicNeutral 121
dynamicUp 121

E

easyHeadsOff 37
easyHeadsOn 37
endSpanners 764
epsfile 242
espressivo 120
eventChords 764
expandFullBarRests 59
expandFullBarRests 60
extra-offset 536

F

f 119
featherDurations 93
featherDurations 764
fermataMarkup 59
fermataMarkup 60
ff 119
fff 119
ffff 119
fffff 119
fill-line 239
filled-box 242
finger 214, 764
first-page-number 523
followVoice 320
font-interface 214, 246
font-size 213, 214
fontsize 235
fontSize 213
footnote 764
forget 31
four-string-banjo 376
fp 119
fret-diagram 349
fret-diagram-interface 354
fret-diagram-terse 351
fret-diagram-verbose 353
FretBoards 357
funkHeads 39
funkHeadsMinor 39

G

general-align 238
glissando 134
grace 765
GregorianTranscriptionStaff 183
Grid_line_span_engraver 222
Grid_point_engraver 222
gridInterval 222
grobdescriptions 765
grow-direction 93

H

halign 237
harmonicByFret 765
harmonicByRatio 765
harmonicNote 765
harmonicsOn 765
hbracket 240
hide 765
hideKeySignature 389
hideNotes 216
hideStaffSwitch 320
horizontal-shift 521
Horizontal_bracket_engraver 224
huge 213, 236

I

improvisationOff 41, 77
improvisationOn 41, 77
indent 202, 521, 557
inner-margin 520
inStaffSegno 765
instrumentSwitch 203, 765
inversion 765
ionian 20
italic 234

J

justified-lines 245
justify 239

K

keepWithTag 765
key 20, 39, 765
killCues 211, 765

L

label 765
laissezVibrer 52
language 765
languageRestore 765
languageSaveAndChange 765
large 213, 236
larger 235, 236
last-bottom-spacing 518
left-align 237
left-margin 519
line-width 519, 557
locrian 20
longa 43, 54
lower 237
ly:minimal-breaking 532
ly:one-line-breaking 532
ly:optimal-breaking 531
ly:page-turn-breaking 531
lydian 20

M

m ... 401
magnify ... 235
magstep ... 213, 602
maj ... 401
major ... 20
major seven symbols ... 410
majorSevenSymbol ... 408
make-dynamic-script ... 125
make-pango-font-tree ... 248
makeClusters ... 166, 766
makeDefaultStringTuning ... 766
mark ... 106, 229, 766
markLengthOff ... 68, 230
markLengthOn ... 68, 230
markup ... 229, 231, 232, 233
markup-markup-spacing ... 518
markup-system-spacing ... 518
markuplist ... 232, 245, 246
max-systems-per-page ... 521
maxima ... 43, 54
measureLength ... 81
measureLength ... 114
measurePosition ... 70, 114
MensuralStaff ... 183
mergeDifferentlyDottedOff ... 170
mergeDifferentlyDottedOn ... 170
mergeDifferentlyHeadedOff ... 170
mergeDifferentlyHeadedOn ... 170
mf ... 119
min-systems-per-page ... 521
minimum-Y-extent ... 536
minimumFret ... 333, 369
minimumPageTurnLength ... 531
minimumRepeatLengthForPageTurn ... 531
minor ... 20
minorChordModifier ... 409
mixed ... 323
mixolydian ... 20
modalInversion ... 15, 766
modalTranspose ... 14, 766
modern ... 28
modern-cautionary ... 29
modern-voice ... 29
modern-voice-cautionary ... 29
mp ... 119
MultiMeasureRestText ... 59
musicglyph ... 108
musicMap ... 766

N

name ... 582
neo-modern ... 30
neo-modern-cautionary ... 30
neo-modern-voice ... 30
neo-modern-voice-cautionary ... 31
no-reset ... 31
noBeam ... 90
nonstaff-nonstaff-spacing ... 536
nonstaff-relatedstaff-spacing ... 536
nonstaff-unrelatedstaff-spacing ... 536
noPageBreak ... 766
noPageTurn ... 766
normal-size-super ... 235
normalsize ... 213, 236
Note_heads_engraver ... 75
null ... 237
numericTimeSignature ... 63

O

octaveCheck ... 9, 766
offset ... 766
omit ... 766
once ... 766
oneVoice ... 166
ottava ... 23, 766
outer-margin ... 520
outside-staff-horizontal-padding ... 551
outside-staff-padding ... 551
outside-staff-priority ... 551
overrideProperty ... 766
overrideTimeSignatureSettings ... 766

P

p ... 119
pad-around ... 241
pad-markup ... 241
pad-to-box ... 241
pad-x ... 241
page-breaking ... 522
page-breaking-system-system-spacing ... 522
page-count ... 522
page-spacing-weight ... 523
pageBreak ... 767
pageTurn ... 767
palmMute ... 767
palmMuteOn ... 767
paper-height ... 516
paper-width ... 519
parallelMusic ... 180, 767
parenthesize ... 219, 767
partcombine ... 175, 767
partcombineApart ... 176
partcombineAutomatic ... 176
partcombineChords ... 176
partcombineDown ... 767
partcombineForce ... 767
partcombineSoloI ... 176
partcombineSoloII ... 176
partcombineUnisono ... 176
partcombineUp ... 767
partial ... 70, 767
pedalSustainStyle ... 323
percent ... 156
phrasingSlurDashed ... 130
phrasingSlurDashPattern ... 130, 767
phrasingSlurDotted ... 130
phrasingSlurDown ... 130
phrasingSlurHalfDashed ... 130
phrasingSlurHalfSolid ... 130
phrasingSlurNeutral ... 130
phrasingSlurSolid ... 130
phrasingSlurUp ... 130
phrygian ... 20
piano ... 29

piano-cautionary 30
PianoStaff 316, 318
Pitch_squash_engraver 77
pitchedTrill 142, 767
pointAndClickOff 767
pointAndClickOn 768
pointAndClickTypes 768
postscript 242
powerChords 374
pp 119
ppp 119
pppp 119
predefinedFretboardsOff 368
predefinedFretboardsOn 368
print-all-headers 523
print-first-page-number 523
print-page-number 523
pushToTag 768

Q

quotedCueEventTypes 206
quotedEventTypes 206
quoteDuring 204, 207, 768

R

r 54
R 58
ragged-bottom 516
ragged-last 520, 557
ragged-last-bottom 516
ragged-right 520, 557
raise 237
relative 2, 5, 13, 319, 768
RemoveEmptyStaves 198, 199
removeWithTag 768
repeatCommands 152
repeatTie 52
resetRelativeOctave 768
rest 54
restrainOpenStrings 333
retrograde 14, 768
revertTimeSignatureSettings 768
rfz 119
rgb-color 218
RhythmicStaff 183
right-align 237
right-margin 519
rightHandFinger 370, 768
rounded-box 240

S

s 56
sacredHarpHeads 39
sacredHarpHeadsMinor 39
scaleDurations 50, 72, 768
score-markup-spacing 518
score-system-spacing 518
scoreTitleMarkup 473
self-alignment-X 536
set 81
set-octavation 23
settingsFrom 768
sf 119
sff 119
sfz 119
shape 768
shiftDurations 768
shiftOff 170
shiftOn 170
shiftOnn 170
shiftOnnn 170
short-indent 202, 521
show-available-fonts 248
showFirstLength 500
showKeySignature 389
showLastLength 500
showStaffSwitch 320
single 768
skip 56, 769
skipTypesetting 500
slashChordSeparator 409
slashedGrace 769
slurDashed 127
slurDashPattern 128, 769
slurDotted 127
slurDown 127
slurHalfDashed 128
slurHalfSolid 128
slurNeutral 127
slurSolid 127
slurUp 128
small 213, 236
smaller 235, 236
sostenutoOff 322
sostenutoOn 322
southernHarmonyHeads 39
southernHarmonyHeadsMinor 39
sp 119
spacing 553
spacingTweaks 769
Span_stem_engraver 320
spp 119
staff-affinity 536
staff-staff-spacing 536
Staff.midiInstrument 504
Staff_symbol_engraver 198
staffgroup-staff-spacing 536
start-repeat 152
startGroup 224
startStaff 191, 194
startTrillSpan 141
Stem 320
stem-spacing-correction 553
stemDown 219
stemLeftBeamCount 91
stemNeutral 219
stemRightBeamCount 91
stemUp 219
stopGroup 224
stopStaff 191, 194, 198
stopTrillSpan 141
storePredefinedDiagram 361, 769
stringTuning 345, 769
stringTunings 345, 357
styledNoteHeads 769
sub 235

suggestAccidentals 430
super 235
sus 403
sustainOff 322
sustainOn 322
system-count 521
system-separator-markup 524
system-system-spacing 518
systems-per-page 521

T

tabChordRepeats 769
tabChordRepetition 769
TabStaff 183, 332
TabVoice 332
tag 769
taor 389
teaching 31
teeny 213, 236
tempo 66
temporary 769
text 323
textLengthOff 60, 226
textLengthOn 60, 226
textSpannerDown 227
textSpannerNeutral 227
textSpannerUp 227
thumb 214
tieDashed 52
tieDashPattern 769
tieDotted 52
tieDown 52
tieNeutral 52
tieSolid 52
tieUp 52
time 62
time 81
time 769
times 769
timeSignatureFraction 72
tiny 213, 236
tocItem 770
top-margin 516
top-markup-spacing 519
top-system-spacing 518
translate 238
translate-scaled 238
transpose 5, 10, 13, 770
transposedCueDuring 210, 770
transposition 24, 204, 770
treCorde 322
tremolo 159
tremoloFlags 159
triangle 242
trill 141
tuplet 45, 72, 770
tupletDown 46
tupletNeutral 46
TupletNumber 46
tupletNumberFormatFunction 46
tupletSpan 770
tupletSpannerDuration 46
tupletUp 46
tweak 770
two-sided 520
type 582

U

unaCorda 322
underline 234
undo 770
unfold 154
unfoldRepeats 770
unHideNotes 216

V

VaticanaStaff 183
VerticalAxisGroup 536
voice 26, 28
Voice 166
voiceOne 166
void 770

W

walkerHeads 39
walkerHeadsMinor 39
whichBar 100
with-color 217
withMusicProperty 771
wordwrap 239
wordwrap-lines 245

X

X-offset 536
x11-color 217, 218
xNote 771
xNotesOn 771

Apéndice E Índice de LilyPond

Además de todas las instrucciones y palabras clave de LilyPond, este índice es una lista de términos musicales y las palabras que tienen relación con cada uno de ellos, con enlaces a aquellas secciones del manual que describen o se ocupan de dicho término. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece el término; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que se discute dicho término.

!

! 6

"

"|" 106

'

' 1

,

, ... 1

-

- 116

.

. 43

/

/ 404
/+ 404

:

: 159

<

< 161
<...> 161

=

= 9

>

> 161

?

? 6

[

[..... 90

]

] 90

^

^ 403

_

_ 258

\

\! 119
\(..... 130
\) 130
\< 119
\> 119
\abs-fontsize 235, 671
\absolute 2
\accent 116
\accepts 582, 584
\acciaccatura 108
\accidentalStyle 26
\addChordShape 361
\addInstrumentDefinition 203, 211
\addlyrics 253, 254, 255
\addQuote 204
\aeolian 20
\afterGrace 109
\aikenHeads 39
\aikenHeadsMinor 39
\alias 582
\allowPageTurn 532
\alterBroken 626
\alternative 144
\appendToTag 494
\appoggiatura 108
\arpeggio 138
\arpeggioArrowDown 138
\arpeggioArrowUp 138
\arpeggioBracket 139
\arpeggioNormal 138
\arpeggioParenthesis 139
\arpeggioParenthesisDashed 139
\arrow-head 242, 695
\ascendens 437, 443
\auctum 437, 443
\augmentum 443
\auto-footnote 716
\autoBeamOff 79
\autoBeamOff 320
\autoBeamOn 79

\autochange ... 318
\backslashed-digit ... 716
\balloonGrobText ... 221
\balloonLengthOff ... 221
\balloonLengthOn ... 221
\balloonText ... 221
\bar ... 94, 100
\barNumberCheck ... 106
\beam ... 695
\bendAfter ... 133
\bold ... 234, 672
\book ... 460, 463
\bookOutputName ... 462
\bookOutputSuffix ... 462
\bookpart ... 461, 463, 530
\box ... 240, 672
\bracket ... 125, 240, 695
\break ... 529
\breathe ... 131
\breve ... 43, 54
\cadenzaOff ... 71
\cadenzaOn ... 71
\caesura ... 435
\caps ... 672
\cavum ... 437, 443
\center-align ... 237, 680
\center-column ... 239, 680
\change ... 317
\char ... 716
\chordmode ... 5, 13, 359
\chordRepeats ... 334
\chords ... 406
\circle ... 240, 695
\clef ... 16
\cm ... 602
\coda ... 116
\column ... 239, 681
\column-lines ... 722
\combine ... 242, 681
\compoundMeter ... 74
\compressFullBarRests ... 59
\compressFullBarRests ... 60
\concat ... 681
\consists ... 582
\context ... 569, 578
\context dentro de un bloque \layout ... 578
\cr ... 119
\cresc ... 120
\crescHairpin ... 120
\crescTextCresc ... 120
\crossStaff ... 320
\cueClef ... 207
\cueDuring ... 207
\cueDuringWithClef ... 207
\customTabClef ... 702
\decr ... 119
\decresc ... 120
\defaultchild ... 586
\defaultTimeSignature ... 63
\defineBarLine ... 98
\deminutum ... 437, 443
\denies ... 582, 584
\descendens ... 437, 443
\dim ... 120
\dimHairpin ... 120
\dimTextDecr ... 120
\dimTextDecresc ... 120
\dimTextDim ... 120
\dir-column ... 682
\discant ... 711
\displayLilyMusic ... 511
\divisioMaior ... 435
\divisioMaxima ... 435
\divisioMinima ... 435
\dorian ... 20
\dotsDown ... 44
\dotsNeutral ... 44
\dotsUp ... 44
\doubleflat ... 702
\doublesharp ... 703
\downbow ... 116, 327
\downmordent ... 116
\downprall ... 116
\draw-circle ... 242, 696
\draw-dashed-line ... 696
\draw-dotted-line ... 696
\draw-hline ... 697
\draw-line ... 242, 697
\drummode ... 183
\dynamic ... 125, 672
\dynamicDown ... 121
\dynamicNeutral ... 121
\dynamicUp ... 121
\easyHeadsOff ... 37
\easyHeadsOn ... 37
\ellipse ... 697
\epsfile ... 242, 698
\espressivo ... 116, 120
\expandFullBarRests ... 59
\expandFullBarRests ... 60
\eyeglasses ... 717
\f ... 119
\featherDurations ... 93
\fermata ... 116, 703
\fermataMarkup ... 59
\fermataMarkup ... 60, 116
\ff ... 119
\fff ... 119
\ffff ... 119
\fffff ... 119
\fill-line ... 239, 682
\fill-with-pattern ... 683
\filled-box ... 242, 698
\finalis ... 435
\finger ... 214, 673
\flageolet ... 116
\flat ... 703
\flexa ... 443
\fontCaps ... 673
\fontsize ... 235, 673
\footnote ... 478, 717
\fp ... 119
\fraction ... 717
\freeBass ... 712
\frenchChords ... 410
\fret-diagram ... 349, 708
\fret-diagram-terse ... 351, 708
\fret-diagram-verbose ... 353, 709
\fromproperty ... 717
\funkHeads ... 39

\funkHeadsMinor 39
\general-align 238, 684
\germanChords 410
\glissando 134
\grace 108
\halfopen 116
\halign 237, 684
\harmonic 327, 336
\harmonicByFret 336
\harmonicByRatio 336
\harmonicsOff 327
\harmonicsOn 327
\harp-pedal 709
\hbracket 240, 698
\hcenter-in 685
\header 463
\hide 609
\hideKeySignature 389
\hideNotes 216
\hideSplitTiedTabNotes 335
\hideStaffSwitch 320
\hspace 686
\huge 213, 236, 673
\improvisationOff 41, 77
\improvisationOn 41, 77
\in 602
\inclinatum 437, 443
\include 490
\inStaffSegno 147
\instrumentSwitch 203
\inversion 13
\ionian 20
\italianChords 410
\italic 234, 673
\justified-lines 245, 722
\justify 239, 687
\justify-field 686
\justify-string 687
\keepWithTag 494
\key 20, 39
\killCues 211
\label 488
\laissezVibrer 52
\large 213, 236, 674
\larger 235, 236, 674
\layout 463, 524, 567, 578
\left-align 237, 688
\left-brace 718
\left-column 688
\lheel 116
\line 688
\linea 437, 443
\lineprall 116
\locrian 20
\longa 43, 54
\longfermata 116
\lookup 718
\lower 237, 689
\ltoe 116
\lydian 20
\lyricmode 251, 253
\lyricsto 253, 254, 255
\magnify 235, 674
\major 20
\makeClusters 166
\map-markup-commands 722
\marcato 116
\mark 106, 229
\markalphabet 718
\markLengthOff 68, 230
\markLengthOn 68, 230
\markletter 718
\markup 229, 231, 232, 233
\markuplist 232, 245, 246
\maxima 43, 54
\medium 674
\melisma 259
\melismaEnd 259
\mergeDifferentlyDottedOff 170
\mergeDifferentlyDottedOn 170
\mergeDifferentlyHeadedOff 170
\mergeDifferentlyHeadedOn 170
\mf 119
\midi 463, 567
\minor 20
\mixolydian 20
\mm 602
\modalInversion 15
\modalTranspose 14
\mordent 116
\mp 119
\musicglyph 108, 703
\name 582
\natural 703
\new 569
\newSpacingSection 554
\noBeam 90
\noBreak 529
\noPageBreak 530
\noPageTurn 532
\normal-size-sub 675
\normal-size-super 235, 675
\normal-text 675
\normalsize 213, 236, 676
\note 704
\note-by-number 704
\null 237, 719
\number 676
\numericTimeSignature 63
\octaveCheck 9
\omit 608
\on-the-fly 477, 719
\once 592
\once 594
\oneVoice 166
\open 116, 327
\oriscus 437, 443
\ottava 23
\oval 699
\override 593, 597, 719
\override-lines 723
\overrideProperty 597
\overrideTimeSignatureSettings 63
\p 119
\pad-around 241, 689
\pad-markup 241, 689
\pad-to-box 241, 690
\pad-x 241, 690
\page-link 719
\page-ref 488, 719

\pageBreak 530
\pageTurn 532
\paper 463, 514
\parallelMusic 180
\parenthesize 219, 699
\partcombine 175, 280
\partcombine y letra 177, 280
\partcombineApart 176
\partcombineAutomatic 176
\partcombineChords 176
\partcombineSoloI 176
\partcombineSoloII 176
\partcombineUnisono 176
\partial 70, 144, 146
\path 700
\pattern 720
\pes 443
\phrasingSlurDashed 130
\phrasingSlurDashPattern 130
\phrasingSlurDotted 130
\phrasingSlurDown 130
\phrasingSlurHalfDashed 130
\phrasingSlurHalfSolid 130
\phrasingSlurNeutral 130
\phrasingSlurSolid 130
\phrasingSlurUp 130
\phrygian 20
\pitchedTrill 142
\portato 116
\postscript 242, 701
\powerChords 374
\pp 119
\ppp 119
\pppp 119
\prall 116
\pralldown 116
\prallmordent 116
\prallprall 116
\prallup 116
\predefinedFretboardsOff 368
\predefinedFretboardsOn 368
\property-recursive 720
\pt 602
\pushToTag 494
\put-adjacent 690
\quilisma 437, 443
\quoteDuring 204, 207
\raise 237, 690
\relative 2, 5, 13, 319
\RemoveEmptyStaves 198, 199
\removeWithTag 494
\repeat 144
\repeat percent 156
\repeat tremolo 159
\repeatTie 52, 147, 276
\replace 676
\rest 54, 705
\rest-by-number 704
\retrograde 14
\reverseturn 116
\revert 593
\revertTimeSignatureSettings 64
\rfz 119
\rheel 116
\right-align 237, 691
\right-brace 720
\right-column 691
\rightHandFinger 370
\roman 676
\rotate 691
\rounded-box 240, 701
\rtoe 116
\sacredHarpHeads 39
\sacredHarpHeadsMinor 39
\sans 677
\scale 702
\scaleDurations 50, 72
\score 459, 463, 705
\segno 116
\semiflat 706
\semiGermanChords 410
\semisharp 707
\sesquiflat 707
\sesquisharp 707
\set 81
\set 590, 597
\sf 119
\sff 119
\sfz 119
\shape 623
\sharp 707
\shiftOff 170
\shiftOn 170
\shiftOnn 170
\shiftOnnn 170
\shortfermata 116
\showKeySignature 389
\showStaffSwitch 320
\signumcongruentiae 116
\simple 677
\skip 56
\skip 275
\slashed-digit 720
\slashedGrace 108
\slurDashed 127
\slurDashPattern 128
\slurDotted 127
\slurDown 127
\slurHalfDashed 128
\slurHalfSolid 128
\slurNeutral 127
\slurSolid 127
\slurUp 128
\small 213, 236, 677
\smallCaps 677
\smaller 235, 236, 678
\snappizzicato 116
\sostenutoOff 322
\sostenutoOn 322
\southernHarmonyHeads 39
\southernHarmonyHeadsMinor 39
\sp 119
\spp 119
\staccatissimo 116
\staccato 116
\startGroup 224
\startStaff 191, 194
\startTrillSpan 141
\stdBass 712
\stdBassIV 713

\stdBassV ... 714
\stdBassVI ... 715
\stemDown ... 219
\stemNeutral ... 219
\stemUp ... 219
\stencil ... 721
\stopGroup ... 224
\stopped ... 116
\stopStaff ... 191, 194, 198
\stopTrillSpan ... 141
\storePredefinedDiagram ... 361
\stringTuning ... 345
\stropha ... 437, 443
\strut ... 721
\sub ... 235, 678
\super ... 235, 678
\sustainOff ... 322
\sustainOn ... 322
\tabChordRepeats ... 334
\tabFullNotation ... 332
\table-of-contents ... 490, 723
\tag ... 494
\taor ... 389
\teeny ... 213, 236, 679
\tempo ... 66
\tenuto ... 116
\text ... 679
\textLengthOff ... 60, 226
\textLengthOn ... 60, 226
\textSpannerDown ... 227
\textSpannerNeutral ... 227
\textSpannerUp ... 227
\thumb ... 116, 214
\tied-lyric ... 707
\tieDashed ... 52
\tieDotted ... 52
\tieDown ... 52
\tieNeutral ... 52
\tieSolid ... 52
\tieUp ... 52
\time ... 62
\time ... 81
\tiny ... 213, 236, 679
\tocItem ... 490
\translate ... 238, 692
\translate-scaled ... 238, 692
\transparent ... 721
\transpose ... 5, 10, 13
\transposedCueDuring ... 210
\transposition ... 24, 204
\treCorde ... 322
\triangle ... 242, 702
\trill ... 116, 141
\tuplet ... 45, 72
\tupletDown ... 46
\tupletNeutral ... 46
\tupletUp ... 46
\turn ... 116
\tweak ... 595, 597
\type ... 582
\typewriter ... 679
\unaCorda ... 322
\underline ... 234, 679
\unfoldRepeats ... 505
\unHideNotes ... 216
\unset ... 592
\upbow ... 116, 327
\upmordent ... 116
\upprall ... 116
\upright ... 680
\varcoda ... 116
\vcenter ... 692
\verbatim-file ... 721
\verylongfermata ... 116
\virga ... 437, 443
\virgula ... 435
\voiceFourStyle ... 169
\voiceNeutralStyle ... 169
\voiceOne ... 166
\voiceOne ... \voiceFour ... 166
\voiceOneStyle ... 169
\voiceThreeStyle ... 169
\voiceTwoStyle ... 169
\vspace ... 692
\walkerHeads ... 39
\walkerHeadsMinor ... 39
\whiteout ... 721
\with ... 576, 580
\with-color ... 217, 721
\with-dimensions ... 722
\with-link ... 722
\with-url ... 702
\woodwind-diagram ... 710
\wordwrap ... 239, 693
\wordwrap-field ... 693
\wordwrap-internal ... 723
\wordwrap-lines ... 245, 723
\wordwrap-string ... 694
\wordwrap-string-internal ... 723

|

| ... 105, 106

~

~ ... 51

1

15ma ... 23

8

8va ... 23

A

a due, partes ... 175
abajo, arco ... 327
absoluta ... 1
absoluta, especificación, de la octava ... 1
absoluta, introducción, de la octava ... 1
absolute ... 2, 762
absolutos, matices ... 119
acampanados, reguladores ... 122
accelerando en MIDI ... 505
accepts ... 582
acciaccatura ... 108

acciaccatura ... 762
acciaccatura de varias notas ... 112
accidental ... 5
accidentals ... 434
accidentalStyle ... 762
acento ... 116, 725
acorde arpegiado ... 138
acorde quebrado ... 138
acorde, modificación de una nota ... 595
acordeón ... 323
acordeón, cambios en el ... 323
acordeón, signos de cambio en el ... 323
acordeón, signos de discanto ... 323
acordes ... 161, 405
acordes alterados ... 402
acordes de jazz ... 399, 407
acordes de pentagrama cruzado ... 320
acordes de potencia ... 374
acordes de quinta vacía ... 374
acordes e introducción relativa de la octava ... 4
acordes extendidos ... 402
acordes vacíos ... 162
acordes y altura relativa ... 162
acordes y ligaduras ... 51
acordes, alteración de notas ... 403
acordes, alteraciones dentro de ... 32
acordes, bajo de ... 404
acordes, diagramas automáticos de ... 367
acordes, diagramas de ... 348
acordes, diagramas de trastes con nombres de ... 359
acordes, digitación de ... 215
acordes, división entre pentagramas con \autochange ... 320
acordes, eliminar repetidos ... 406
acordes, formas de, en instrumentos de trastes ... 361
acordes, inversiones ... 404
acordes, modo de ... 399
acordes, nombres de ... 399, 405
acordes, nombres de, en MIDI ... 505
acordes, repetición de ... 163, 334
acordes, supresiones en ... 403
acordes, variantes de ... 401
addChordShape ... 361, 762
adding a white background to text ... 721
addInstrumentDefinition ... 203, 211, 762
additionalPitchPrefix ... 408
addQuote ... 204, 762
adicionales internas, líneas ... 191
adicionales, líneas ... 191
adicionales, modificar líneas ... 191
adiciones en acordes ... 403
adorno, letra y notas de ... 284
adorno, notas de, dentro de corchetes de grupo especial ... 49
adorno, seguimiento de notas de ... 109
adorno, sincronización de notas de ... 112
aeolian ... 20
afinación no occidental ... 452
afinaciones del banjo ... 376
afterGrace ... 109, 762
afterGraceFraction ... 730
agrupación de grupes especiales ... 45
agrupar pulsos ... 87
Aiken, cabezas de nota con forma ... 39
aikenHeads ... 39
aikenHeadsMinor ... 39
ajustado a palabras, texto ... 239
ajuste de línea, texto con ... 239
ajuste del símbolo del pentagrama ... 603
al aire, indicación de cuerda ... 327
al niente ... 122
alias ... 582
alignAboveContext ... 586
alignBelowContext ... 274, 586
alineación a objetos ... 619
alineación a una cadenza (cadencia) ... 114
alineación, instrucciones de texto ... 240
alinear elementos de marcado ... 237
alinear texto ... 237
alinear texto de marcado ... 237
alist ... 728
allowPageTurn ... 762
allowVoltaHook ... 762
alteración con paréntesis ... 6
alteración de cuartos de tono ... 7
alteración de precaución ... 6
alteración de voz de precaución moderno, estilo de ... 29
alteración en nota ligada ... 6
alteración recordatoria ... 6
alteración, estilo de voz de ... 28
alteración, estilo de voz de precaución moderno ... 29
alteración, estilo de, olvidar ... 31
alteración, estilo moderno de ... 29
alteración, moderno de precaución, voz, estilo de ... 29
alteración, trino con ... 143
alteraciones ... 430
alteraciones accidentales ... 26, 430, 445
alteraciones automáticas ... 26
alteraciones de la musica ficta ... 430
alteraciones de piano ... 29
alteraciones en acordes ... 32
alteraciones estilo moderno ... 28
alteraciones modernas ... 29
alteraciones no restablecer, estilo de ... 31
alteraciones y notas simultáneas ... 32
alteraciones, cadencias ... 71
alteraciones, en varias voces ... 29
alteraciones, estilo de las ... 26
alteraciones, estilo de, neo-moderno ... 30
alteraciones, estilo moderno ... 28
alteraciones, estilo moderno de precaución ... 29
alteraciones, estilo predeterminado ... 28
alteraciones, estilo predeterminado de las ... 26
alteraciones, estilo teaching (enseñanza) ... 31
alteraciones, estilo, dodecafónico ... 31
alteraciones, estilo, moderno-precaución ... 28
alteraciones, estilo, neo-moderno de precaución ... 30
alteraciones, estilo, neo-moderno de voz ... 30
alteraciones, música sin metro ... 71
alteraciones, piano de precaución ... 30
alteraciones, piano de precaución, estilo de ... 30
alteraciones,estilo, neo-modern-voice-cautionary ... 31
alterBroken ... 762
alternativos, finales, en repeticiones explícitas ... 154
alto, clave de ... 16
alturas ... 1
alturas en MIDI ... 505
alturas, nombres de, otros idiomas ... 7
alturas, transporte de ... 10

Amazing Grace, ejemplo de gaita ... 389
ambitus ... 33
anacrusa ... 70
anacrusa en repeticiones ... 146
análisis musicológico ... 224
analizador léxico ... 729
analizador sintáctico ... 730
ángulo, reguladores en ... 615
anidada, repetición ... 151
anidados, corchetes de pentagrama ... 188
anidamiento de pentagramas ... 188
`annotate-spacing` ... 564
anthems ... 290
antiguas, claves ... 16
apagado ... 725
`appendToTag` ... 762
`applyContext` ... 762
`applyMusic` ... 762
`applyOutput` ... 762
appoggiatura ... 108
`appoggiatura` ... 763
árabe, ejemplo de música ... 456
árabe, música ... 452
árabe, plantilla de música ... 456
árabes, armaduras ... 456
Arabic key signatures ... 454
Arabic note names ... 453
Arabic semi-flat symbol ... 453
arco abajo ... 116, 725
arco abajo, indicación ... 327
arco arriba ... 116, 725
arco arriba, indicación ... 327
arco, indicaciones de ... 327
área de espaciado nueva ... 554
armadura ... 430
Armadura de la tonalidad ... 20
armadura, visibilidad después de un cambio explícito ... 612
armónicos artificiales ... 328
armónicos en instrumentos de trastes ... 372
armónicos en tablatura ... 336
armónicos naturales ... 327
armónicos, cabezas de nota ... 35
arpa sacra, cabezas de nota con forma ... 39
arpa, diagramas de pedal ... 325
arpa, pedales ... 325
`arpeggio` ... 138
`arpeggioArrowDown` ... 138
`arpeggioArrowUp` ... 138
`arpeggioBracket` ... 139
`arpeggioNormal` ... 138
`arpeggioParenthesis` ... 139
`arpeggioParenthesisDashed` ... 139
arpegiado, acorde, símbolos especiales ... 139
arpegio ... 138
arpegio con paréntesis de pentagrama cruzado ... 141
arpegio de pentagrama cruzado con paréntesis ... 141
arriba, arco ... 327
`arrow-head` ... 242
Articlulate, scripts ... 505
articulaciones ... 116, 435
articulaciones en MIDI ... 505
articulation-event ... 206
artificiales, armónicos ... 328
aspas, cabezas de nota ... 35
`assertBeamQuant` ... 763
`assertBeamSlope` ... 763
`associatedVoice` ... 253
associatedVoice ... 254, 285
`aug` ... 401
`auto-first-page-number` ... 523
autobarrado, propiedades para indicaciones de compás ... 63
`autoBeaming` ... 81
`autoBeaming` ... 568
`autoBeamOff` ... 79
`autoBeamOn` ... 79
autocambiador y el modo relativo ... 319
`autochange` ... 318, 763
automática, combinación de partes ... 175
automáticas, alteraciones ... 26
automaticBars ... 614
automático, cambio de pentagrama ... 318
automáticos, diagramas de acordes ... 367
automáticos, diagramas de traste ... 367
ayuda, globos de ... 221

B

backslashed digits ... 716
bajo cifrado, alineación ... 418
bajo cifrado, líneas de extensión ... 416
bajo de los acordes ... 404
bajo, clave de ... 16
`Balloon_engraver` ... 221
`balloonGrobText` ... 221, 763
`balloonLengthOff` ... 221
`balloonLengthOn` ... 221
`balloonText` ... 221, 763
banjo, afinaciones del ... 376
banjo, tablatura de ... 329
banjo, tablaturas de ... 375
`banjo-c-tuning` ... 376
`banjo-modal-tuning` ... 376
`banjo-open-d-tuning` ... 376
`banjo-open-dm-tuning` ... 376
Banter ... 407
`bar` ... 94, 100, 763
`barCheckSynchronize` ... 105
barítono, clave de ... 16
`BarNumber` ... 101
`barNumberCheck` ... 106, 763
`barNumberVisibility` ... 101
barra doble ... 94
barra final ... 94
barra, finales con varias voces ... 88
barra, finales en la partitura ... 88
barradas, cabezas de nota ... 41
barrado, propiedades predefinidas para el compás ... 63
barras de compás ... 94
barras de compás manuales ... 94
barras de compás, música sin metro ... 71
barras de pentagrama cruzado ... 317
barras de trémolo ... 159
barras en ángulo ... 80
barras en música polimétrica ... 73
barras en saltos de línea ... 80
barras manuales ... 79, 90

barras manuales, abreviatura de dirección 90
barras manuales, notas de adorno 90
barras progresivas 93
barras y letra 81
barras y melismas 79
barras, \partcombine con \autoBeamOff 80
barras, cadencias 71
barras, música sin metro 71
barras, orientación de, secundarias 87
barras, reglas personalizadas 79
barras, subdivisión 86
Bartók pizzicato 328
`bartype` 100
`base-shortest-duration` 553
`baseMoment` 81
Bass, figured 413
Bass, thorough 413
Basso continuo 413
batería 376, 378
`beamExceptions` 81
`beatStructure` 81
becuadro 5
bemol 5
bemol, doble 5
`bendAfter` 133, 763
Bézier, curvas de, puntos de control 622
`binding-offset` 520
bisbiglando 325
Bison 730
`blank-after-score-page-penalty` 523
`blank-last-page-penalty` 522
`blank-page-penalty` 522
BNF 730
`bold` 234
`bookOutputName` 763
`bookOutputSuffix` 763
`bookTitleMarkup` 473
`bottom-margin` 516
`box` 240
`bracket` 125, 240
`bracket` 323
break-align-symbols 619
break-visibility 610
`breakable` 80
breakbefore 471
`breathe` 131
`breathe` 763
`breve` 43, 54
breve, silencio de 54
buscar fuentes tipográficas disponibles 248

C

cabezas de nota antiguas 427, 445
cadencia 71
cadencia, alteraciones 71
cadencia, barras 71
cadencia, barras de compás 71
cadencia, números de compás 71
cadencias, saltos de línea en 72
cadencias, saltos de página en 72
cadenza (cadencia) 114
cadenza (cadencia), alinear a 114
`cadenzaOff` 71
`cadenzaOn` 71
caesura 132
caídas de tono (falls) 133
calderón 108, 116, 725
calderón sobre la línea divisoria 229
calderón sobre un silencio multicompás 59
callback 728
cambiar propiedades 590
cambio automático de pentagrama 318
cambio de dedo 214
cambio de instrumento 203
cambio de la fuente tipográfica 234
cambio manual de pentagrama 317
cantante, nombre del 283
cánticos 302
capas 609
capo 353
caracteres especiales 497
caracteres especiales en modo de marcado 233
caracteres no ASCII 497
cejilla, indicación de 348
`center-align` 237
`center-column` 239
centering a column of text 680
centrados, matices, en música de piano 316
centrar texto en la página 239
cerradura 728
cerradura simple 728
cesura 132
`change` 317
changing direction of text columns 682
`check-consistency` 520
`chordChanges` 406
`chordmode` 5, 13, 359
`chordNameExceptions` 409
`chordNameLowercaseMinor` 408
`ChordNames` 359
`chordNameSeparator` 408
`chordNoteNamer` 408
`chordPrefixSpacer` 410
`chordRepeats` 763
`chordRootNamer` 408
Christian Harmony, cabezas de nota con forma ... 39
cifrado americano 399
`circle` 240
circling text 695
circulus 725
citar otras voces 204, 207
clave 16
clave antigua 16
clave de alto 16
clave de bajo 16
clave de barítono 16
clave de contrabajo 16
clave de Do 16
clave de Fa 16
clave de mezzosoprano 16
clave de percusión 377
clave de Sol 16
clave de soprano 16
clave de tablatura 347
clave de tenor 16
clave de violín 16
clave francesa 16
clave moderntab 347

clave transpositora ... 17
clave, visibilidad después de un cambio explícito ... 612
claves ... 433
claves, visibilidad de la transposición ... 614
clef ... 5
`clef` ... 16, 763
clefs ... 425, 444
cluster ... 166
coda ... 108, 116, 725
coda sobre la línea divisoria ... 229
colisión del número de compás ... 105
colisiones ... 170
colisiones de pentagrama cruzado ... 317
colisiones, columnas que chocan ... 165
colisiones, ignorar ... 165, 174
colisiones, voces de pentagrama cruzado ... 317
colocación de los silencios multicompás ... 60
`color` ... 217
color en acordes ... 218
color rgb ... 218
coloreadas, notas ... 217
coloreadas, notas de acorde ... 218
coloreado de objetos ... 609
coloreados, objetos ... 217
colorear notas ... 217
colorear objetos ... 217
colores ... 217
colores de x11 ... 217
colores, lista de ... 649
coloring text ... 721
coloring voices ... 169
`column` ... 239
columnas, texto ... 239
coma, intervalos de ... 458
`combine` ... 242
comienzo de repetición ... 152
comienzo del sistema ... 184
comillas en la letra ... 251, 258
`common-shortest-duration` ... 553
compás completo, silencios de ... 58
compás compuesto, indicación de ... 74
compás de anacrusa ... 70
compás parcial ... 70
compás, agrupación ... 87
compás, barras invisibles de ... 94
compás, comprobación ... 105
compás, comprobación de ... 105
compás, definir líneas de ... 98
compás, doble ... 72
compás, estilo de la indicación de ... 63
compás, indicación de ... 426
compás, indicación de, ajustes predeterminados ... 63
compás, número de ... 115
compás, número de, formato ... 102
compás, número, alineación ... 104
compás, número, comprobación ... 105
compás, número, con letras ... 103
compás, número, con repeticiones ... 103
compás, números de ... 101
compás, propiedades, restaurar los valores predeterminados ... 64
compás, repetición de ... 156
compás, repeticiones y números de ... 151
compás, visibilidad de la indicación de ... 62
compases distintos al mismo tiempo ... 576
compases polimétricos ... 72
compases, sub-agrupar ... 87
`Completion_heads_engraver` ... 75
`Completion_rest_engraver` ... 75
`compoundMeter` ... 763
compresión de música ... 50
`compressFullBarRests` ... 59
`compressFullBarRests` ... 60
compuesto, indicaciones de compás ... 74
concatenating text ... 681
condensar silencios ... 62
conformar ligaduras ... 623
`consists` ... 582
constante, reguladores de tipo ... 122
contexto, cambiar propiedades predeterminadas de ... 577
contextos implícitos ... 586
contextos, creación y referenciación de ... 569
contextos, definición de nuevos ... 582
contextos, mantener vivos ... 573
contextos, orden de disposición ... 584
contextos, tiempo de vida ... 573
continua, ligadura ... 127
contrabajo, clave de ... 16
control, alturas de ... 9
control, trucar puntos de ... 596
controlling general text alignment ... 684
`controlpitch` ... 9
copyright, símbolo de ... 498
coral, clave de tenor ... 17
corchete de casilla de repetición con texto ... 153
corchete de primera y segunda vez ... 152
corchete horizontal ... 224
corchete vertical ... 184
corchetes ... 219, 224
corchetes anidados ... 188
corchetes de fraseo ... 224
coro, sistema de ... 184
`cr` ... 119
creating empty text objects ... 719
creating horizontal spaces in text ... 686
creating text fractions ... 717
creating vertical spaces in text ... 692, 721
`cresc` ... 120
crescendo ... 119
crescendo-event ... 206
`crescHairpin` ... 120
`crescTextCresc` ... 120
`cross` ... 35
`crossStaff` ... 763
cruce de pentagramas ... 320
cruzado, acordes de pentagrama ... 320
cruzado, línea de pentagrama ... 320
cruzado, notas de pentagrama ... 320
cruzado, pentagrama ... 320
cruzado, pentagrama, barras de ... 317
cruzado, pentagrama, notas de ... 317
cruzado, plicas de pentagrama ... 320
cuartos de tono ... 6
cuartos de tono en MIDI ... 505
cuartos de tono, alteraciones ... 7
cue notes, removing ... 211
`cueClef` ... 207, 763
`cueClefUnset` ... 763

`cueDuring` ... 207, 764
`cueDuringWithClef` ... 207, 764
cuerda al aire, indicación de ... 327
cuerda, número de la ... 330
cuerdas orquestales ... 326
cuerdas, escribir música para ... 326
CueVoice ... 207
`currentBarNumber` ... 101, 114
custodes ... 424
custos ... 424

D

D.S. al Fine ... 108
`deadNote` ... 764
decorar texto ... 240
`decr` ... 119
`decresc` ... 120
decrescendo ... 119
`default` ... 26, 28
`default-staff-staff-spacing` ... 536
`defaultBarType` ... 100
`defaultNoteHeads` ... 764
`defaultTimeSignature` ... 63
`defineBarLine` ... 98, 764
definición de salida ... 730
definir líneas de compás ... 98
`denies` ... 582
derecha, mano, digitaciones para instrumentos de traste ... 370
derecho, pedal ... 322
deslizamientos en notación de tablatura ... 342
desplazamiento ... 56
desplazamiento de silencios, automático ... 170
desplazamiento de voces ... 170
desplazamiento, silencio automático de ... 170
desplazar nota ... 170
desplegar repetición ... 154
diagrama de trastes personalizado ... 354
diagramas de acordes ... 357
diagramas de acordes para instrumentos de trastes ... 348
diagramas de posiciones ... 357
diagramas de posiciones de mandolina ... 357
diagramas de posiciones de ukelele ... 357
diagramas de traste, transporte de los ... 359
diagramas de trastes ... 348
diagramas de trastes personalizados, añadir ... 360
dibujar el símbolo del pentagrama ... 603
dibujar objetos gráficos ... 240
digitación ... 214
digitación de acordes ... 215
digitación frente a números de cuerda ... 330
digitación, instrucciones de, para acordes ... 215
digitaciones de la mano derecha para instrumentos de trastes ... 370
digitaciones y silencios multicompás ... 62
digitaciones, añadir, a diagramas de trastes ... 369
`dim` ... 120
`dim` ... 401
`dimHairpin` ... 120
diminuendo ... 119
`dimTextDecr` ... 120
`dimTextDecresc` ... 120
`dimTextDim` ... 120
dinámica ... 119
dinámica, posición vertical ... 121
dinámica, varias indicaciones sobre una nota ... 120
dinámicas indicaciones editoriales ... 125
dinámicas, indicaciones centradas en música de tecla ... 316
dinámicas, indicaciones entre paréntesis ... 125
dinámicas, indicaciones, nuevas ... 125
discanto, signos de, del acordeón ... 323
discontinua, ligadura ... 127
discontinuas, ligaduras ... 52
discontinuas, ligaduras de fraseo ... 130
diseño «mensurstriche» ... 187
diseño de página ... 557
`displayLilyMusic` ... 764
`displayMusic` ... 764
`displayScheme` ... 764
disponibles, listar las fuentes tipográficas ... 248
`disposición, archivo de` ... 526
distancia entre pentagramas ... 536
distancia entre pentagramas en música de piano ... 320
distancias absolutas ... 602
distancias escaladas ... 602
dividida, letra ... 279
divididos, modificación de objetos de extensión ... 626
divisio ... 435
división de notas ... 75
división de silencios ... 75
división en música etiquetada ... 494
divisiones ... 435
Do, clave de ... 16
dobladas, barras ... 80
doble barra ... 94
doble bemol ... 5
doble compás ... 72
doble puntillo, notas con ... 43
doble sostenido ... 5
dodecafónico, estilo de alteraciones ... 31
`dodecaphonic` ... 31
doits (elevaciones de tono) ... 133
`dorian` ... 20
dórico, modo ... 20
`dotsDown` ... 44
`dotsNeutral` ... 44
`dotsUp` ... 44
downbow ... 725
downmordent ... 725
downprall ... 725
`draw-circle` ... 242
`draw-line` ... 242
drawing a line across a page ... 697
drawing beams within text ... 695
drawing boxes with rounded corners ... 698
drawing boxes with rounded corners around text ... 701
drawing circles within text ... 696
drawing dashed lines within text ... 696
drawing dotted lines within text ... 696
drawing ellipse around text ... 697
drawing lines within text ... 697
drawing oval around text ... 699
drawing paths ... 700
drawing solid boxes within text ... 698
drawing triangles within text ... 702

`drummode` ... 183
`DrumStaff` ... 183
duración de las notas ... 43
duración predeterminada ... 43
duraciones de la melodía, mostrar ... 76
duraciones en MIDI ... 505
duraciones, escalado de ... 49
`dynamic` ... 125
dynamic-event ... 206
`dynamicDown` ... 121
`DynamicLineSpanner` ... 121
`dynamicNeutral` ... 121
`dynamicUp` ... 121

E

`easyHeadsOff` ... 37
`easyHeadsOn` ... 37
eclesiásticos, silencios ... 60
editoriales, indicaciones dinámicas ... 125
elevaciones de tono (doits) ... 133
enclosing text in a box with rounded corners ... 701
enclosing text within a box ... 672
encuadernación ... 520
`endSpanners` ... 764
enmarcar texto ... 240
ensayo, letra de, estilo ... 107
ensayo, letra de, formato ... 107
ensayo, letra de, manual ... 107
ensayo, letras de ... 106
eolio, modo ... 20
eonio, modo ... 20
EPS, salida ... 501
`epsfile` ... 242
escalado de pentagramas ... 194
escalar las duraciones ... 49
escribir música en paralelo ... 180
espaciado de la letra ... 269
espaciado horizontal ... 552
espaciado vertical ... 536
espaciado, nueva área de ... 554
espaciado, presentación del ... 564
espacio dentro de los sistemas ... 536
espacio entre pentagramas ... 536
espacios en blanco ... 465
espacios en la letra ... 251, 258
especial, símbolos de arpegio ... 139
especiales, cabezas de figura ... 35
espressivo ... 116
`espressivo` ... 120
espressivo ... 725
espressivo, articulación ... 120
estándar, tamaño de la tipografía (de notación) ... 214
estilo de la letra de ensayo ... 107
estilo de la ligadura ... 127
estilos de cabeza de nota ... 35
etiqueta ... 494
`eventChords` ... 764
eventos contenedores ... 162
excepciones, nombres de acorde ... 410
`expandFullBarRests` ... 59
`expandFullBarRests` ... 60
expansión de música ... 50
explícita, trinos de nota ... 142
explícitas, repeticiones ... 154
explicitClefVisibility ... 612
explicitKeySignatureVisibility ... 612
expresiones de marcado ... 233
extensión, modificación de objetos de ... 626
extensora, línea ... 262
`extra-offset` ... 536

F

`f` ... 119
Fa, clave de ... 16
fácil, notación, cabezas de nota de ... 37
falls (caídas de tono) ... 133
fantasma, notas ... 219
`featherDurations` ... 93
`featherDurations` ... 764
fermata ... 725
`fermataMarkup` ... 59
`fermataMarkup` ... 60
Ferneyhough, reguladores de ... 122
Feta, tipografía ... 650
`ff` ... 119
`fff` ... 119
`ffff` ... 119
`fffff` ... 119
figura, cabeza de la ... 35
Figured bass ... 413
`fill-line` ... 239
`filled-box` ... 242
final de repetición ... 152
final, barra ... 94
finales alternativos ... 144
finales alternativos y letra ... 275
finalis ... 435
`finger` ... 214, 764
`first-page-number` ... 523
flageolet ... 116, 725
flags ... 428
Flex ... 729
`followVoice` ... 320
`font-interface` ... 214, 246
`font-size` ... 213, 214
`fontsize` ... 235
`fontSize` ... 213
`footnote` ... 764
`forget` ... 31
formas, notas con ... 39
formateo de la letra ... 251
formato de la letra de ensayo ... 107
formato de los elementos de extensión de texto ... 227
forte, pedal ... 322
`four-string-banjo` ... 376
`fp` ... 119
fragmentos ... 207
fragmentos, citar ... 204
francesa, clave ... 16
francesa, partituras a la ... 198
francesa, pentagrama a la ... 198
francesa, pentagramas a la ... 194
fraseo, corchetes de ... 224
fraseo, en la letra ... 259
fraseo, ligadura de ... 127
fraseo, ligadura de, definir patrones de discontinuidad ... 130

fraseo, ligadura de, mitad continua y mitad discontinua ... 130
fraseo, ligaduras de ... 130
fraseo, ligaduras de, discontinuas ... 130
fraseo, ligaduras de, punteadas ... 130
fraseo, ligaduras de, simultáneas ... 130
fraseo, ligaduras de, varias ... 130
fraseo, marcas de ... 130
`fret-diagram` ... 349
fret-diagram, marcado con ... 349
`fret-diagram-interface` ... 354
`fret-diagram-terse` ... 351
fret-diagram-terse, marcado con ... 351
`fret-diagram-verbose` ... 353
fret-diagram-verbose, marcado con ... 353
`FretBoards` ... 357
frigio, modo ... 20
fuente tipográfica ... 728
fuente tipográfica (de notación), tamaño ... 213
fuente tipográfica (de notación), tamaño estándar de ... 214
fuente tipográfica, cambiar ... 234
fuente tipográfica, familias de ... 236
fuente tipográfica, tamaño de ... 235
fuente, establecer la familia de ... 248
fuentes tipográficas de símbolos en marcado ... 246
fuentes tipográficas, buscar, disponibles ... 248
fuentes tipográficas, cambiar para todo el documento ... 248
fuentes tipográficas, explicación ... 246
fundamental de un acorde ... 401
Funk, cabezas de nota con forma ... 39
`funkHeads` ... 39
`funkHeadsMinor` ... 39

G

gaita ... 389
gaita escocesa de las tierras altas ... 389
gaita, ejemplo de ... 389
`general-align` ... 238
glifo ... 728
glifos musicales ... 108
glissandi y repeticiones ... 151
`glissando` ... 134
glissandos en acordes ... 343
globo ... 221
globo de ayuda ... 221
grabadores, incluir en contextos ... 582
`grace` ... 765
gráfica, notación ... 242
gráficos, dibujar objetos ... 240
gráficos, insertar ... 240, 242
gráficos, insertar objetos ... 240
gramática de LilyPond ... 730
Gregorian square neumes ligatures ... 437
gregoriano, canto, pauta de transcripción de ... 183
`GregorianTranscriptionStaff` ... 183
`Grid_line_span_engraver` ... 222
`Grid_point_engraver` ... 222
`gridInterval` ... 222
grob ... 588, 728
grob, propiedades de ... 593
`grobdescriptions` ... 765
grobs, sobreescritura de ... 609
grobs, visibilidad de ... 608
`grow-direction` ... 93
grupeto ... 116
grupeto circular ... 725
grupeto circular invertido ... 725
grupetos en MIDI ... 505
grupos de valoración especial, colocación del corchete ... 46
grupos de valoración especial, modificación del número ... 46
grupos especiales ... 45
grupos especiales, formato de ... 46
guía, formateo de las notas ... 207
guía, notas ... 207
guía, notas, formato de ... 207
guía, voces ... 204
guías musicales ... 296
guiones ... 116, 263
guitarra, cabezas de nota ... 35
guitarra, mostrar ritmos rasgueados ... 77
guitarra, tablas de acordes ... 77
guitarra, tablatura de ... 329

H

Hal Leonard ... 37
halfopen ... 725
`halign` ... 237
hammer on ... 343
Harmonica Sacra, cabezas de nota con forma ... 39
`harmonicByFret` ... 765
`harmonicByRatio` ... 765
`harmonicNote` ... 765
`harmonicsOn` ... 765
harps ... 325
`hbracket` ... 240
`hide` ... 765
`hideKeySignature` ... 389
`hideNotes` ... 216
`hideStaffSwitch` ... 320
himnos ... 302
horizontal, alineación de texto ... 237
horizontal, corchete ... 224
horizontal, espaciado ... 552
horizontal, sobreescribir el espaciado ... 628
`horizontal-shift` ... 521
`Horizontal_bracket_engraver` ... 224
horizontally centering text ... 680
hufnagel ... 421, 422
`huge` ... 213, 236

I

ictus ... 725
idioma, alturas en otros ... 7
idioma, nombres de nota en otros ... 7
imágenes, insertar ... 242
implícitos, contextos ... 586
importing stencils into text ... 721
impresión de caracteres especiales ... 233
impresión de los nombres de acorde ... 405
impresión, orden de ... 609
improvisación ... 41
`improvisationOff` ... 41, 77

`improvisationOn` ... 41, 77
include-settings ... 497
including files ... 490
`indent` ... 202, 521, 557
independiente, texto ... 231
Indicación de compás ... 62
indicación de pulgar ... 116, 725
indicaciones de pedal ... 116
indicaciones de pedalero de órgano ... 116
indicaciones de tempo dentro de corchetes de grupo especial ... 49
indicaciones de texto ... 229
indicaciones dinámicas nuevas ... 125
indicar No Chord (sin acorde) en ChordNames ... 405
inlining an Encapsulated PostScript image ... 698
`inner-margin` ... 520
inscripción sobre un silencio multicompás ... 59
insertar gráficos ... 242
insertar objetos gráficos ... 240
inserting music into text ... 705
inserting PostScript directly into text ... 701
inserting URL links into text ... 702
`inStaffSegno` ... 765
instrument names, centering ... 201
instrumento, cambio de ... 203
instrumento, nombre abreviado ... 201
instrumento, nombre del ... 201
instrumento, nombres de ... 504
instrumentos transpositores ... 11
instrumentos, nombres complejos de ... 201
`instrumentSwitch` ... 203, 765
interfaces de objetos gráficos ... 729
interfaz ... 729
interfaz de grob ... 729
interfaz de la presentación ... 588
introducir la letra ... 251
inversión ... 13
`inversion` ... 765
inversión modal ... 15
invisible, plica ... 219
invisible, silencio ... 56
invisibles, notas ... 216
`ionian` ... 20
`italic` ... 234

J

jazz, acordes de ... 399, 407
justificado, texto ... 239
`justified-lines` ... 245
`justify` ... 239
justifying lines of text ... 722
justifying text ... 687

K

`keepWithTag` ... 765
`key` ... 20, 39, 765
key signature ... 5, 434
KievanStaffContext ... 444
KievanVoiceContext ... 444
`killCues` ... 211, 765
kirchenpausen ... 60

L

`label` ... 765
laissez vibrer ... 52
`laissezVibrer` ... 52
`language` ... 765
`languageRestore` ... 765
`languageSaveAndChange` ... 765
`large` ... 213, 236
`larger` ... 235, 236
`last-bottom-spacing` ... 518
left aligning text ... 688
`left-align` ... 237
`left-margin` ... 519
letra ... 251
letra asignada a una voz ... 166
letra de ensayo, estilo ... 107
letra de ensayo, formato ... 107
letra dividida ... 279
letra sobre notas de adorno ... 284
letra y barras ... 81
letra y melodías ... 254
letra y notas ligadas ... 276
letra, alineación a una melodía ... 253
letra, alineación con melodía esporádica ... 574
letra, aumentar la separación ... 269
letra, compartida entre voces ... 280
letra, desplazamiento de ... 56
letra, formateado ... 251
letra, introducir ... 251
letra, mantener dentro del margen ... 226
letra, posicionamiento ... 264
letra, puntuación en la ... 251
letra, repeticiones ... 271
letra, repeticiones con finales alternativos ... 275
letra, repeticiones con primera y segunda vez ... 275
letra, saltar notas ... 275
letras de ensayo ... 106
lexer ... 729
lheel ... 725
lidio, modo ... 20
ligadura continua ... 127
ligadura de expresión, definir patrones de discontinuidad ... 128
ligadura de expresión, estilo ... 127
ligadura de expresión, mitad continua y mitad discontinua ... 128
ligadura de fraseo ... 127
ligadura de fraseo mitad continua y mitad discontinua ... 130
ligadura de fraseo, definir patrones de discontinuidad ... 130
ligadura de puntos ... 127
ligadura de unión ... 51
ligadura discontinua ... 127
ligadura, alteraciones y ... 6
ligaduras ... 446
ligaduras de expresión ... 127
ligaduras de expresión simultáneas ... 127
ligaduras de expresión y repeticiones ... 151
ligaduras de expresión, debajo de las notas ... 127
ligaduras de expresión, encima de las notas ... 127
ligaduras de expresión, modificar ... 622
ligaduras de expresión, posicionamiento manual ... 127
ligaduras de expresión, varias ... 127

ligaduras de fraseo ... 130
ligaduras de fraseo discontinuas ... 130
ligaduras de fraseo punteadas ... 130
ligaduras de fraseo simultáneas ... 130
ligaduras de fraseo, varias ... 130
ligaduras de puntos ... 52
ligaduras de unión y corchetes de primera y segunda vez ... 52
ligaduras de unión, modificar ... 622
ligaduras de unión, repeticiones y ... 52
ligaduras discontinuas ... 52
ligaduras en la letra ... 258
ligaduras laissez vibrer ... 52
ligaduras y acordes ... 51
ligaduras, apariencia ... 52
ligaduras, colocación ... 52
ligaduras, en primera y segunda vez ... 147
ligaduras, en repeticiones ... 147
Ligatures ... 423
ligatures in text ... 681
`line-width` ... 519, 557
línea de cambio de pentagrama ... 320
línea de extensión ... 262
línea de seguimiento de pentagrama ... 320
línea divisoria predeterminada, cambiar ... 100
línea vertical entre pentagramas ... 222
línea, barras en saltos de ... 80
línea, saltos ... 94
línea, saltos de ... 527
líneas adicionales ... 191
líneas de rejilla ... 222
líneas del pentagrama ... 191
líneas divisorias ... 94
líneas divisorias invisibles ... 94
líneas divisorias manuales ... 94
líneas divisorias, eliminación ... 614
líneas divisorias, símbolos sobre las ... 229
lineprall ... 725
lista de asociación ... 728
lista de colores ... 649
lista-A ... 728
listar fuentes tipográficas disponibles ... 248
llave vertical ... 184
llaves de distintos tamaños ... 246
llaves de varios tamaños ... 246
llaves, anidado de ... 188
`locrian` ... 20
locrio, modo ... 20
`longa` ... 43, 54
longa, silencio de ... 54
longfermata ... 725
longitud de las notas ... 43
`lower` ... 237
lowering text ... 689
ltoe ... 725
`ly:minimal-breaking` ... 532
`ly:one-line-breaking` ... 532
`ly:optimal-breaking` ... 531
`ly:page-turn-breaking` ... 531
`lydian` ... 20
lyrics, using variables ... 263

M

`m` ... 401
`magnify` ... 235
magnifying text ... 674
`magstep` ... 213, 602
`maj` ... 401
`major` ... 20
`major seven symbols` ... 410
`majorSevenSymbol` ... 408
makam ... 458
makamlar ... 452, 458
`make-dynamic-script` ... 125
`make-pango-font-tree` ... 248
`makeClusters` ... 166, 766
`makeDefaultStringTuning` ... 766
mantener música etiquetada ... 494
mantenimiento, pedal, estilos de ... 323
manual, cambio de pentagrama ... 317
manual, marca de ensayo ... 107
manual, marca de repetición ... 152
`Manuales` ... 1
manuales, barras ... 79, 90
manuales, barras de compás ... 94
manuales, líneas divisorias ... 94
maqam ... 452
maqams ... 452
marca de ensayo manual ... 107
marca de ensayo, estilo ... 107
marca de ensayo, formato ... 107
marca de pulso ... 132
marcado ... 233
marcado condicional ... 477
marcado de texto ... 233
marcado de varias líneas ... 239
marcado de varias páginas ... 245
marcado sobre silencios multicompás ... 59
marcado, caracteres especiales en modo de ... 233
marcado, centrar en la página ... 239
marcado, elementos, alinear ... 237
marcado, expresiones de ... 233
marcado, notación musical dentro de ... 243
marcado, partitura dentro de ... 245
marcado, relleno de textos de ... 241
marcado, sintaxis ... 233
marcado, texto de, varias páginas ... 245
marcado, texto, ajustado a palabras ... 239
marcado, texto, alinear ... 237
marcado, texto, decorar ... 240
marcado, texto, enmarcar ... 240
marcado, texto, instrucciones de alineación ... 240
marcas de fraseo ... 130
marcas de pie ... 725
marcato ... 116, 725
margen de encuadernación ... 520
margen, texto fuera del ... 226
`mark` ... 106, 229, 766
`markLengthOff` ... 68, 230
`markLengthOn` ... 68, 230
`markup` ... 229, 231, 232, 233
`markup-markup-spacing` ... 518
`markup-system-spacing` ... 518
`markuplist` ... 232, 245, 246
matices ... 119
matices absolutos ... 119

matices, posición vertical ... 121
matices, varios sobre una nota ... 120
max-systems-per-page ... 521
maxima ... 43, 54
maxima, silencio de ... 54
mayor, modo ... 20
measureLength ... 81
measureLength ... 114
measurePosition ... 70, 114
Medicaea, Editio ... 421, 422
medida ... 62
medida, música sin ... 114
medios, intervalos ... 452
melisma ... 259, 262
melismas, barras en ... 79
melismata ... 259
melodía alternativa, cambio a una ... 285
melodía, mostrar las duraciones ... 76
menor, modo ... 20
mensuración, símbolo de ... 426
mensural ... 421, 422
Mensural ligatures ... 431
mensural, música, transcripción de ... 187
MensuralStaff ... 183
MensuralStaffContext ... 424
MensuralVoiceContext ... 424
mensurstriche, diseño ... 187
mergeDifferentlyDottedOff ... 170
mergeDifferentlyDottedOn ... 170
mergeDifferentlyHeadedOff ... 170
mergeDifferentlyHeadedOn ... 170
merging text ... 681
metro polimétrico ... 72
metro, estilo de ... 63
metrónomo, indicación de ... 66
metrónomo, indicación de, con texto ... 66
mezclar notas ... 170
mezzosoprano, clave de ... 16
mf ... 119
microtonos ... 8
microtonos en MIDI ... 505
MIDI ... 24, 502
MIDI, alturas ... 505
MIDI, articulaciones ... 505
MIDI, bloque ... 502
MIDI, cuartos de tono ... 505
MIDI, definiciones de contexto ... 503
MIDI, duraciones ... 505
MIDI, microtonos ... 505
MIDI, nombres de ... 505
MIDI, transposición ... 24
min-systems-per-page ... 521
minimum-Y-extent ... 536
minimumFret ... 333, 369
minimumPageTurnLength ... 531
minimumRepeatLengthForPageTurn ... 531
minor ... 20
minorChordModifier ... 409
mirroring markup ... 702
mixed ... 323
mixolidio, modo ... 20
mixolydian ... 20
modal, inversion ... 15
modal, transposición ... 14
modales, transformaciones ... 14
modalInversion ... 15, 766
modalTranspose ... 14, 766
modern ... 28
modern-cautionary ... 29
modern-voice ... 29
modern-voice-cautionary ... 29
moderno de precaución, alteraciones de estilo ... 29
moderno, alteraciones de estilo ... 28, 29
moderno, estilo de alteraciones ... 28
moderno-precaución, estilo de alteraciones ... 28
moderntab, clave ... 347
modificadores de acorde ... 401
modo ... 730
modo de acordes ... 399
modos ... 20
modos eclesiásticos ... 20
mordent ... 725
mordente ... 116
mordente ascendente ... 116
mordente circular ... 116
mordente circular, inverso ... 116
mordente descendente ... 116
mordentes ... 108, 389
movimientos, varios ... 460
mp ... 119
multicompás, posicionamiento de los silencios ... 60
multicompás, silencio, adjuntar texto ... 59
multicompás, silencio, aplicar un calderón a ... 59
multicompás, silencio, con marcados ... 59
multicompás, silencios ... 58
multicompás, silencios, contracción de ... 59
multicompás, silencios, digitaciones y ... 62
multicompás, silencios, expansión de ... 59
multicompás, silencios, inscripción en ... 59
MultiMeasureRestText ... 59
música de cuatro compases por línea ... 529
música dentro de marcado ... 243
música desplegada con finales alternativos ... 154
musica ficta, alteraciones ... 430
música intercalada ... 180
música para principiantes ... 37
música religiosa ... 302
música sin medida ... 114
música sin metro ... 71
música sin metro, alteraciones ... 71
música sin metro, barras ... 71
música sin metro, barras de compás ... 71
música sin metro, números de compás ... 71
musicales, glifos ... 108
musicales, guías ... 296
musicglyph ... 108
musicMap ... 766
musicológico, análisis ... 224
musicQuotes ... 730

N

N.C., símbolo ... 405
name ... 582
natural, nota ... 5
naturales, armónicos ... 327
neo-modern ... 30
neo-modern-cautionary ... 30
neo-modern-voice ... 30

neo-modern-voice, estilo de alteración 30
`neo-modern-voice-cautionary` 31
neo-modern-voice-cautionary, estilo de alteraciones 31
neo-moderno de precaución, estilo de alteraciones 30
neo-moderno, estilo de alteraciones 30
neomensural 422
niente, al 122
nivel superior, texto del 231
no musicales, símbolos 242
no restablecer, estilo de alteraciones 31
no vacíos, textos 226
no-puros: contenedores de Scheme 628
`no-reset` 31
`noBeam` 90
nombre del cantante 283
nombres de instrumento, añadir a otros contextos 203
nombres de instrumento, cambiar 203
nombres de las notas 1
nombres de personajes 294
`nonstaff-nonstaff-spacing` 536
`nonstaff-relatedstaff-spacing` 536
`nonstaff-unrelatedstaff-spacing` 536
`noPageBreak` 766
`noPageTurn` 766
`normal-size-super` 235
`normalsize` 213, 236
nota abierta 116
nota de espaciado 56
nota tapada 116
nota, cabeza de, armónico 35
nota, cabeza de, en aspas 35
nota, cabeza de, en parlato 35
nota, cabeza de, en rombo 35
nota, cabeza de, estilos 35
nota, cabeza de, guitarra 35
nota, cabeza especial de 35
nota, duración predeterminada 43
nota, estilos de cabeza 671
notación dentro de marcado 243
notación gráfica 242
notación, explicación de 221
notación, tamaño de la fuente 213
notas al pie 478
notas al pie dentro de expresiones musicales 478
notas al pie en texto independiente 484
Notas al pie, basadas en eventos 479
notas al pie, basadas en tiempo 481
notas coloreadas 217
notas coloreadas en acordes 218
notas de adorno 389
notas de adorno, cambiar los ajustes de disposición 110
notas de adorno, trucar 110
notas de pentagrama cruzado 320
notas entre paréntesis 219
notas fantasma 219
notas guía 204, 207
notas invisibles 216
notas ocultas 216
notas pequeñas 207
notas transparentes 216
notas, cabeza de, forma de rombo 327
notas, cabezas de 213
notas, cabezas de, Aiken 39
notas, cabezas de, arpa sacra 39
notas, cabezas de, barradas 41
notas, cabezas de, Christian Harmony 39
notas, cabezas de, con formas 39
notas, cabezas de, estudio 37
notas, cabezas de, Funk 39
notas, cabezas de, Harmonica Sacra 39
notas, cabezas de, improvisación 41
notas, cabezas de, notación simplificada 37
notas, cabezas de, para practicar 37
notas, cabezas de, Southern Harmony 39
notas, cabezas de, Walker 39
notas, colisiones de 170
notas, con doble puntillo 43
notas, con puntillo 43
notas, corchetes de agrupación de 224
notas, división de 75
notas, duración de 43
notas, espaciar horizontalmente 554
notas, longitud de 43
notas, nombres de, en otros idiomas 7
notas, nombres holandeses de 5
notas, nombres predeterminados 5
notas, transporte de 10
note-event 206
`Note_heads_engraver` 75
notes de pentagrama cruzado 317
notes within text by log and dot-count 704
notes within text by string 704
nuevo pentagrama 183
nuevos contextos 569
`null` 237
NullVoice 280
`numericTimeSignature` 63
número de compás 115
número de compás, colisión 105
número de compás, comprobación de 105
número de compases, formato de 102
números de compás 101
números de compás y repeticiones 151
números de compás, cadencias 71
números de compás, espaciado regular 101
números de compás, música sin metro 71
números de cuerda frente a digitaciones 330
números de página, automáticos 523
números de página, especificar el primero 523
números de página, suprimir 523

O

objects, rotating 615
objeto de propiedades 730
objeto de Scheme 731
objetos de presentación 728
objetos gráficos 728
objetos inmutables 729
objetos mutables 729
objetos, coloreado de 609
objetos, colorear 217
objetos, sobreescritura de 609
objetos, visibilidad de 608
octava, comprobación de 9

octava, corrección de ... 9
octava, especificación absoluta ... 1
octava, especificación relativa ... 2
octava, introducción absoluta ... 1
octava, introducción relativa ... 2
octava, marca de cambio de ... 1
octava, transposición ... 17
octava, transposición, opcional ... 17
octavación ... 23
octaveCheck ... 9, 766
ocultación de pentagramas ... 198
ocultar pentagramas ... 198
ocultar pentagramas de música antigua ... 199
ocultar pentagramas rítmicos ... 199
ocultas, notas ... 216
offset ... 766
olvidar, estilo de alteraciones ... 31
omit ... 766
on-the-fly ... 477
once ... 766
oneVoice ... 166
open ... 725
operación de inversión ... 13
operación de inversión modal ... 15
operación de retrogradación ... 14
operación de transposición ... 14
operaciones modales ... 14
oratorios ... 290
órgano, indicaciones de pedal de ... 725
ornamentación ... 116
ornamentos ... 108, 116
orquestales, cuerdas ... 326
ossia ... 194, 199
Otomana, música ... 458
ottava ... 23, 766
outer-margin ... 520
output-count ... 730
output-def ... 730
output-suffix ... 730
outside-staff-horizontal-padding ... 551
outside-staff-padding ... 551
outside-staff-priority ... 551
overrideProperty ... 766
overrideTimeSignatureSettings ... 766
overriding properties within text markup ... 719

P

p ... 119
pad-around ... 241
pad-markup ... 241
pad-to-box ... 241
pad-x ... 241
padding (relleno) ... 589
padding text ... 689
padding text horizontally ... 690
page-breaking ... 522
page-breaking-system-system-spacing ... 522
page-count ... 522
page-spacing-weight ... 523
pageBreak ... 767
pageTurn ... 767
página, disposición de la ... 557
página, orientación de la ... 515
página, saltos de ... 557
páginas, marcado de varias ... 245
palmMute ... 767
palmMuteOn ... 767
Pango ... 246
papel, tamaño, apaisado ... 515
papel, tamaño, orientación ... 515
paper-height ... 516
paper-width ... 519
paralelo, música en ... 180
parallelMusic ... 180, 767
parcial, compás ... 70
paréntesis ... 219
paréntesis en ángulo ... 161
paréntesis, alteración con ... 6
parenthesize ... 219, 767
parlato ... 300
parlato, cabezas de nota ... 35
parser ... 730
partcombine ... 175, 767
partcombineApart ... 176
partcombineAutomatic ... 176
partcombineChords ... 176
partcombineDown ... 767
partcombineForce ... 767
partCombineListener ... 730
partcombineSoloI ... 176
partcombineSoloII ... 176
partcombineUnisono ... 176
partcombineUp ... 767
parte al alzar ... 70
partes cantadas ... 290
partes, combinación automática ... 175
partes, combinador de ... 175
partial ... 70, 767
partitura dentro marcado ... 245
paths, drawing ... 700
pausa, marcas de ... 131
pauta ... 183
pauta de coro ... 184
pauta de percusión ... 183
pauta, escalado de ... 194
pedal de mantenimiento, estilo de ... 323
pedal de órgano ... 116
pedal derecho ... 322
pedal forte ... 322
pedal sostenuto ... 322
pedal, estilos de indicación ... 323
pedal, indicación mixta ... 323
pedal, indicación por corchete ... 323
pedal, indicación textual ... 323
pedal, indicaciones de ... 725
pedales de piano ... 322
pedales del arpa ... 325
pedales del arpa, diagramas ... 325
pedalSustainStyle ... 323
pentagrama a la francesa ... 194
pentagrama cruzado, arpegios con paréntesis ... 141
pentagrama cruzado, plica de ... 320
pentagrama cruzado, trémolo de ... 160
pentagrama de percusión ... 183
pentagrama único ... 183
pentagrama único, polifonía de ... 166
pentagrama vacío ... 198
pentagrama, cambio ... 320

pentagrama, cambio automático de ... 318
pentagrama, cambio de ... 320
pentagrama, cambios manuales de ... 317
pentagrama, detener e inicial líneas del ... 191
pentagrama, distancia entre los ... 536
pentagrama, establecer el símbolo del ... 603
pentagrama, fijar el tamaño de ... 526
pentagrama, iniciar ... 183
pentagrama, instanciar ... 183
pentagrama, línea de cambio de ... 320
pentagrama, mofidicar las líneas del ... 191
pentagrama, nuevo ... 183
pentagrama, ocultación de ... 198
pentagrama, símbolo del ... 191
pentagramas anidados ... 188
pentagramas de instrumentos de teclado ... 316
pentagramas de piano ... 184, 316
pentagramas, grupo de ... 184
pentagramas, varios ... 184
pequeñas, notas ... 207
`percent` ... 156
percusión ... 376, 378
percusión, clave ... 377
percusión, pauta de ... 183
personajes, nombres ... 294
personalización de los nombres de acorde ... 407
personalizada, marca de ensayo ... 107
personalizadas, afinaciones de cuerdas ... 345
personalizados, añadir diagramas de traste ... 360
personalizados, diagramas de trastes ... 348, 354
Petrucci ... 421, 422
`phrasingSlurDashed` ... 130
`phrasingSlurDashPattern` ... 130, 767
`phrasingSlurDotted` ... 130
`phrasingSlurDown` ... 130
`phrasingSlurHalfDashed` ... 130
`phrasingSlurHalfSolid` ... 130
`phrasingSlurNeutral` ... 130
`phrasingSlurSolid` ... 130
`phrasingSlurUp` ... 130
`phrygian` ... 20
`piano` ... 29
piano de precaución, alteraciones ... 30
piano de precaución, estilo de alteraciones ... 30
piano, alteraciones de ... 29
piano, estilo de alteraciones de ... 29
piano, música de, matices centrados ... 316
piano, pedales de ... 322
piano, pentagramas de ... 184, 316
piano, sistema de ... 184
`piano-cautionary` ... 30
`PianoStaff` ... 316, 318
picado ... 116
`Pitch_squash_engraver` ... 77
`pitchedTrill` ... 142, 767
pitchnames ... 730
pizzicato, Bartók ... 328
pizzicato, snap ... 328
placing horizontal brackets around text ... 698
placing parentheses around text ... 699
placing vertical brackets around text ... 695
plantilla música árabe ... 456
plica ... 219
plica abajo ... 219
plica arriba ... 219
plica cruzada ... 110
plica de pentagrama cruzado ... 320
plica invisible ... 219
plica neutral ... 219
plica, dirección ... 219
plicas de pentagrama cruzado ... 320
`pointAndClickOff` ... 767
`pointAndClickOn` ... 768
`pointAndClickTypes` ... 768
polifonía en un solo pentagrama ... 166
polifonía, letra compartida ... 280
polifónica, música ... 170
polimétricas, partituras ... 576
polimétrico, metro, barras en ... 73
polimétricos, compases ... 72
porcentaje, repetición de ... 156
portato ... 116, 725
posición y cejilla, indicación para instrumentos de trastes ... 372
posicionamiento de la letra ... 264
`postscript` ... 242
PostScript ... 242
PostScript encapsulado, salida de ... 501
potencia, acordes de ... 374
`powerChords` ... 374
`pp` ... 119
`ppp` ... 119
`pppp` ... 119
prall ... 725
pralldown ... 725
prallmordent ... 116, 725
prallprall ... 116, 725
prallup ... 725
precaución, alteración de ... 6
precaución, alteraciones de piano de ... 30
precaución, estilo de alteraciones, piano ... 30
`predefinedFretboardsOff` ... 368
`predefinedFretboardsOn` ... 368
predefinidas, afinaciones, para instrumentos de traste ... 345
predeterminada, duración de la nota ... 43
predeterminada, línea divisoria, cambiar ... 100
predeterminadas, cambio de propiedades de contexto ... 577
predeterminado, estilo de las alteraciones ... 26
predeterminados, nombres de nota ... 5
presentación, interfaz de la ... 588
prima volta ... 144
primera y segunda vez ... 144
primera y segunda vez y letra ... 275
primera y segunda vez, con ligaduras ... 147
primera y segunda vez, corchetes y ligaduras en ... 52
primera y segunda vez, repeticiones ... 154
principiantes, música para ... 37
`print-all-headers` ... 523
`print-first-page-number` ... 523
`print-page-number` ... 523
prob ... 730
progresivas, barras ... 93
propiedades ... 590
propiedades compartidas ... 729
propiedades de grob ... 593
propiedades inmutables ... 729
propiedades mutables ... 729
pulgar, indicación de ... 725

pull off ... 343
pulsos por minuto ... 66
pulsos, agrupación ... 87
punteadas, ligaduras de fraseo ... 130
puntillo, notas con ... 43
puntos de control en curvas de Bézier ... 622
puntos, ligadura de ... 127
puntos, ligaduras de ... 52
puntuación ... 251
puntuación en la letra ... 251
puros, contenedores de Scheme ... 628
`pushToTag` ... 768
putting space around text ... 689

Q

quebrado, acorde ... 138
quinta vacía, acordes de ... 374
quitar música etiquetada ... 494
`quotedCueEventTypes` ... 206
`quotedEventTypes` ... 206
`quoteDuring` ... 204, 207, 768

R

`r` ... 54
`R` ... 58
racimo (cluster) ... 166
`ragged-bottom` ... 516
`ragged-last` ... 520, 557
`ragged-last-bottom` ... 516
`ragged-right` ... 520, 557
`raise` ... 237
raising text ... 690
rallentando en MIDI ... 505
rango de alturas ... 33
rasgueado, mostrar ritmos ... 77
Ratisbona, Editio ... 422
recordatoria, alteración ... 6
redonda, silencios de, para un compás completo ... 58
Referencia de funcionamiento interno ... 567
referenciar contextos ... 569
referencing page labels in text ... 722
referencing page numbers in text ... 719
regulador ... 119
reguladores acampanados ... 122
reguladores de Ferneyhough ... 122
reguladores de tipo constante ... 122
reguladores en ángulo ... 615
rejilla, líneas de ... 222
relativa, especificación de la octava ... 2
relativa, introducción de la octava, y acordes ... 4
relativas, especificación de octavas ... 2
`relative` ... 2, 5, 13, 319, 768
relativo ... 2
relativo, acordes en altura ... 162
relativo, modo, autocambiador y ... 319
relativo, modo, transposición y ... 5
relleno ... 589
relleno alrededor de texto ... 241
`RemoveEmptyStaves` ... 198, 199
`RemoveEmptyStaves` ... 771
`removeWithTag` ... 768
removing cue notes ... 211
renacimiento, música del ... 187
`repeatCommands` ... 152
`repeatTie` ... 52
repetición ambigua ... 151
repetición anidada ... 151
repetición con anacrusa ... 146
repetición con finales alternativos ... 144
repetición con `q` ... 334
repetición corta ... 156
repetición de compás ... 156
repetición de la letra con finales alternativos ... 275
repetición de la letra con primera y segunda vez ... 275
repetición de porcentaje ... 156
repetición de trémolo ... 159
repetición manual ... 152
repetición y números de compás ... 151
repetición, barras de ... 94
repetición, cambiar finales alternativos ... 152
repetición, cambiar número de ... 152
repetición, comienzo de ... 152
repetición, con `q` ... 163
repetición, contador de tiempo de la ... 151
repetición, final de ... 152
repetición, ligaduras de unión en ... 52
repeticiones ... 97
repeticiones alternativas ... 154
repeticiones con primera y segunda vez ... 154
repeticiones desplegadas ... 154
repeticiones desplegadas con finales alternativos ... 154
repeticiones explícitas ... 154
repeticiones normales ... 144
repeticiones y glissandi ... 151
repeticiones y letra ... 271
repeticiones y ligadura de expresión ... 151
repeticiones, con ligaduras ... 147
repeticiones, con segno ... 147
repeticiones, expandir ... 505
repeticiones, letras y números de compás ... 150
repeticiones, números de compás de primera y segunda vez ... 150
reservados, impresión de caracteres ... 233
`resetRelativeOctave` ... 768
respiraciones ... 131
`rest` ... 54
rest-event ... 206
restaurar las propiedades predeterminadas del tipo de compás ... 64
`restrainOpenStrings` ... 333
rests or multi-measure-rests within text by log and dot-count ... 704
rests or multi-measure-rests within text by string ... 705
rests, ancient ... 429
retrogradación, transformación de ... 14
`retrograde` ... 14, 768
reverseturn ... 725
reversión de sobreescrituras ... 593
`revertTimeSignatureSettings` ... 768
`rfz` ... 119
rgb, color ... 218
`rgb-color` ... 218
rheel ... 725
`RhythmicStaff` ... 183
right aligning text ... 691

right-align ... 237
right-margin ... 519
rightHandFinger ... 370, 768
rítmica, pauta ... 183
rombo, cabezas de nota ... 35
rombo, cabezas en forma de ... 327
rotating objects ... 615
rotating text ... 691
rounded-box ... 240
rtoe ... 725

S

s ... 56
sacredHarpHeads ... 39
sacredHarpHeadsMinor ... 39
salida de gráficos de vector escalables ... 501
salida de PostScript encapsulado ... 501
salida, definiciones de ... 567
salmos ... 302
saltar notas en la letra ... 275
saltos de línea ... 527
saltos de línea normales ... 529
saltos de línea, cadencias ... 72
saltos de línea, música sin medida ... 72
saltos de página ... 557
saltos de página, cadencias ... 72
saltos de página, música sin medida ... 72
saltos, en música sin medida ... 72
SATB ... 290
scaleDurations ... 50, 72, 768
scaling markup ... 702
scaling text ... 692
Scheme, contenedores no-puros ... 628
Scheme, contenedores puros ... 628
score-markup-spacing ... 518
score-system-spacing ... 518
scoreTitleMarkup ... 473
seconda volta ... 144
segno ... 96, 108, 116, 725
segno sobre la línea divisoria ... 229
segno, con repeticiones ... 147
seguimiento de voz ... 320
selección del tamaño de la fuente tipográfica (de notación) ... 213
self-alignment-X ... 536
sello ... 731
sello, eliminar ... 608
Semai, forma ... 456
semi-bemol ... 8
semi-bemoles ... 6
Semi-flat symbol appearance ... 453
semi-sostenido ... 8
semi-sostenidos ... 6
semicirculus ... 725
semitrino ... 116, 725
semitrino ascendente ... 116
semitrino descendente ... 116, 725
separación de sistemas, marca de ... 190
separado, texto ... 231
séptima, acordes de ... 401
sesqui-bemol ... 8
sesqui-sostenido ... 8
set ... 81
set-octavation ... 23
setting extent of text objects ... 722
setting horizontal text alignment ... 684
setting subscript in standard font size ... 675
setting superscript in standard font size ... 675
settingsFrom ... 768
sf ... 119
sff ... 119
sfz ... 119
shape ... 768
shiftDurations ... 768
shiftOff ... 170
shiftOn ... 170
shiftOnn ... 170
shiftOnnn ... 170
short-indent ... 202, 521
shortfermata ... 725
show-available-fonts ... 248
showFirstLength ... 500
showFirstLength ... 730
showKeySignature ... 389
showLastLength ... 500
showLastLength ... 730
showStaffSwitch ... 320
signumcongruentiae ... 725
sílabas, duración automática de las ... 254
silencio ... 54
silencio de separación ... 56
silencio invisible ... 56
silencio normal, condensar ... 62
silencio, colisiones de ... 62
silencio, especificar la posición vertical ... 55
silencios de compás completo ... 55, 58
silencios de redonda para un compás completo ... 58
silencios eclesiásticos ... 60
silencios multicompás ... 55, 58
silencios, división de ... 75
silencios, indicar duraciones ... 54
símbolos no musicales ... 242
símbolos, fuentes tipográficas en el marcado ... 246
simple text strings ... 677
simple text strings with tie characters ... 707
simplificada, notación ... 37
simultáneas, ligaduras de expresión ... 127
simultáneas, ligaduras de fraseo ... 130
simultáneas, notas y alteraciones ... 32
sin acorde, símbolo ... 405
sin medida, música, saltos de línea en ... 72
sin medida, música, saltos de página en ... 72
single ... 768
sintaxis del marcado ... 233
sistema ... 184
sistema de coro ... 184
sistema de piano ... 184
sistema, delimitador de comienzo ... 184
sistema, delimitadores de inicio anidados ... 188
skip ... 56, 769
skipTypesetting ... 500
slashChordSeparator ... 409
slashed digits ... 720
slashedGrace ... 769
slur-event ... 206
slurDashed ... 127
slurDashPattern ... 128, 769
slurDotted ... 127
slurDown ... 127

slurHalfDashed ... 128
slurHalfSolid ... 128
slurNeutral ... 127
slurSolid ... 127
slurUp ... 128
small ... 213, 236
smaller ... 235, 236
smob ... 731
snap pizzicato ... 328
snappizzicato ... 725
sobreescritura de objetos ... 609
sobreescritura para un solo momento ... 594
sobreescrituras, reversión de ... 593
Sol, clave de ... 16
Solesmes ... 422
solo, partes ... 175
sonido ... 502
soprano, clave de ... 16
sos. ... 322
sostenido ... 5
sostenido, doble ... 5
sostenuto, pedal ... 322
sostenutoOff ... 322
sostenutoOn ... 322
Southern Harmony, cabezas de nota con forma ... 39
southernHarmonyHeads ... 39
southernHarmonyHeadsMinor ... 39
sp ... 119
spacing ... 553
spacingTweaks ... 769
Span_stem_engraver ... 320
spp ... 119
Sprechgesang ... 300
Square neumes ligatures ... 437
staccatissimo ... 116, 725
staccato ... 116, 725
stacking text in a column ... 681
staff-affinity ... 536
staff-staff-spacing ... 536
Staff.midiInstrument ... 504
Staff_symbol_engraver ... 198
staffgroup-staff-spacing ... 536
start-repeat ... 152
startGroup ... 224
startStaff ... 191, 194
startTrillSpan ... 141
Stem ... 320
stem-spacing-correction ... 553
stemDown ... 219
stemLeftBeamCount ... 91
stemNeutral ... 219
stemRightBeamCount ... 91
stemUp ... 219
stencil ... 731
stopGroup ... 224
stopStaff ... 191, 194, 198
stopTrillSpan ... 141
storePredefinedDiagram ... 361, 769
stringTuning ... 345, 769
stringTunings ... 345, 357
styledNoteHeads ... 769
styles, voice ... 169
sub ... 235
subíndice ... 235
subscript text ... 678
suggestAccidentals ... 430
super ... 235
superíndice ... 235
superscript text ... 678
sus ... 403
sustainOff ... 322
sustainOn ... 322
SVG, salida ... 501
system-count ... 521
system-separator-markup ... 524
system-system-spacing ... 518
systems-per-page ... 521

T

tabChordRepeats ... 769
tabChordRepetition ... 769
tablatura ... 329
tablatura de bajo ... 345
tablatura de banjo ... 329, 345, 375
tablatura de contrabajo ... 345
tablatura de guitarra ... 345
tablatura de mandolina ... 345
tablatura de ukelele ... 345
tablatura de viola ... 345
tablatura de violin ... 345
tablatura de violoncello ... 345
tablatura predeterminada ... 332
tablatura y armónicos ... 336
tablatura y deslizamientos ... 342
tablatura, afinaciones predefinidas de ... 345
tablatura, clave de ... 347
tablatura, pauta de ... 183
tablatura, principios básicos ... 332
tablaturas personalizadas ... 345
tablaturas, afinaciones personalizadas ... 345
TabStaff ... 183, 332
TabVoice ... 332
tag ... 769
tamaño de la página ... 514
tamaño del papel ... 514
taor ... 389
tapadas, notas, en instrumentos de trastes ... 372
taqasim ... 456
teaching ... 31
teaching (enseñanza), estilo de alteraciones ... 31
tecla, pentagramas para instrumentos de ... 316
teclado, pentagramas para instrumentos de ... 316
teclados, matices centrados ... 316
teeny ... 213, 236
tempo ... 66
temporary ... 769
tenor, clave de ... 16
tenor, clave de, coral ... 17
tenuto ... 116, 725
tesitura ... 33
text ... 323
text columns, left-aligned ... 688
text columns, right-aligned ... 691
textLengthOff ... 60, 226
textLengthOn ... 60, 226
texto ajustado a palabras ... 239
texto de marcado ajustado a palabras ... 239
texto de varias líneas ... 239

texto del nivel superior ... 231
texto destacado ... 226
texto en casilla de repetición ... 153
texto en columnas ... 239
texto entrecomillado en modo de marcado ... 233
texto fuera del margen ... 226
texto independiente ... 231
texto justificado ... 239
texto que ocupa varias páginas ... 245
texto separado ... 231
texto sobre la barra de compás ... 229
texto sobre un silencio multicompás ... 59
texto, alineación de ... 237
texto, alineación horizontal ... 237
texto, alineación vertical ... 237
texto, centrar en la página ... 239
texto, decorar ... 240
texto, elementos de extensión ... 227
texto, elementos no vacíos de ... 226
texto, en otros idiomas ... 225
texto, enmarcar ... 240
texto, formateado de los elementos de extensión ... 227
texto, indicaciones de ... 229
texto, inscripciones de ... 226
texto, instrucciones de alineación ... 240
texto, mantener dentro del margen ... 226
texto, marcado de ... 233
texto, relleno de ... 241
texto, tamaño de ... 235
`textSpannerDown` ... 227
`textSpannerNeutral` ... 227
`textSpannerUp` ... 227
Thorough bass ... 413
`thumb` ... 214
thumb-script ... 214
`tieDashed` ... 52
`tieDashPattern` ... 769
`tieDotted` ... 52
`tieDown` ... 52
tiempo, administración ... 114
tiempo, control del (dentro de la partitura) ... 114
tiempo, repeticiones y contador de ... 151
`tieNeutral` ... 52
`tieSolid` ... 52
`tieUp` ... 52
`time` ... 62
`time` ... 81
`time` ... 769
`times` ... 769
`timeSignatureFraction` ... 72
`tiny` ... 213, 236
tipografía ... 728
tipografía Feta ... 650
tipografía, fijar el tamaño de ... 526
tipografiar texto ... 233
`tocItem` ... 770
`Top` ... 1
`top-margin` ... 516
`top-markup-spacing` ... 519
`top-system-spacing` ... 518
toplevel-bookparts ... 730
toplevel-scores ... 730
transcripción de música mensural ... 187
transformación de retrogradación ... 14
transformaciones modales ... 14
`translate` ... 238
`translate-scaled` ... 238
translating text ... 692
transparentes, hacer los objetos ... 609
transparentes, notas ... 216
transportadas, visibilidad de las claves ... 614
transporte ... 10, 14
transporte de diagramas de trastes ... 359
transporte modal ... 14
`transpose` ... 5, 10, 13, 770
`transposedCueDuring` ... 210, 770
transposición ... 10
transposición de alturas ... 10
transposición de MIDI ... 24
transposición de notas ... 10
transposición instrumental ... 24
transposición modal ... 14
transposición y modo relativo ... 5
transposición, de octava, opcional ... 17
`transposition` ... 24, 204, 770
transpositor, instrumento ... 24
transpositoras, claves ... 17
transpositores, instrumentos ... 11
traste ... 333
trastes personalizados, añadir diagramas de ... 360
trastes, añadir digitaciones a diagramas de ... 369
trastes, armónicos en instrumentos de ... 372
trastes, diagramas automáticos de ... 367
trastes, diagramas de ... 348
trastes, diagramas de, con nombres de acorde ... 359
trastes, diagramas de, personalizados ... 348
trastes, diagramas de, transporte ... 359
trastes, diagramas personalizados de ... 354
trastes, instrumentos con, afinaciones predefinidas ... 345
trastes, instrumentos de, digitaciones de la derecha ... 370
trastes, instrumentos de, formas de acordes ... 361
trastes, instrumentos de, indicación de la posición y cejilla ... 372
trastes, notas tapadas en instrumentos de ... 372
tre corde ... 322
`treCorde` ... 322
`tremolo` ... 159
trémolo, indicaciones de ... 159
`tremoloFlags` ... 159
trémolos de pentagrama cruzado ... 160
tresillos ... 45
tresillos, formato de ... 46
tríadas ... 401
`triangle` ... 242
`trill` ... 141
trill ... 725
trino ... 116, 725
trino con nota y alteración ... 143
trinos ... 141
trinos de nota explícita ... 142
trinos en MIDI ... 505
trucaje de las notas de adorno ... 110
trucar ... 595
trucar puntos de control ... 596
`tuplet` ... 45, 72, 770
`tupletDown` ... 46
`tupletNeutral` ... 46
`TupletNumber` ... 46

`tupletNumberFormatFunction` ... 46
`tupletSpan` ... 770
`tupletSpannerDuration` ... 46
`tupletUp` ... 46
Turquía, música de ... 458
Turquía, nombres de notas en ... 458
`tweak` ... 770
tweak, relación con `\override` ... 597
`two-sided` ... 520
`type` ... 582

U

U.C. ... 322
ukelele ... 349
una corda ... 322
`unaCorda` ... 322
`underline` ... 234
underlining text ... 679
`undo` ... 770
`unfold` ... 154
`unfoldRepeats` ... 770
`unHideNotes` ... 216
Unicode ... 498
upmordent ... 725
upprall ... 725
UTF-8 ... 497

V

vacía, acordes de quinta ... 374
vacío, pentagrama ... 198
varcoda ... 116, 725
variable de Scheme ... 730
variable del analizador sintáctico ... 730
variable global ... 730
variables ... 464
variables, uso de las ... 492
varias ligaduras de expresión ... 127
varias ligaduras de fraseo ... 130
varias líneas, marcado ... 239
varias líneas, texto ... 239
varias voces ... 170
varios matices sobre una nota ... 120
Vaticana, Editio ... 421, 422
`VaticanaStaff` ... 183
VaticanaStaffContext ... 433
VaticanaVoiceContext ... 433
verso, número de ... 282
vertical, alineación de texto ... 237
vertical, espaciado ... 536, 557
vertical, línea, entre pentagramas ... 222
vertical, posición, de las indicaciones dinámicas ... 121
`VerticalAxisGroup` ... 536
vertically centering text ... 692
verylongfermata ... 725
viento, instrumentos ... 385
violín, clave de ... 16
visibilidad de las claves transportadas ... 614
visibilidad de los objetos ... 608
voces divididas ... 293
voces, \partcombine con \autoBeamOff ... 80
voces, alteraciones en varias ... 29
voces, citar ... 207
voces, varias ... 170
`voice` ... 26, 28
`Voice` ... 166
voice styles ... 169
`voiceOne` ... 166
`void` ... 770
volta ... 144
volta, prima ... 144
volta, seconda ... 144
voz ... 166
voz, estilo de alteraciones ... 28
voz, seguimiento de ... 320

W

Walker, cabezas de nota con forma ... 39
`walkerHeads` ... 39
`walkerHeadsMinor` ... 39
`whichBar` ... 100
White mensural ligatures ... 431
`with-color` ... 217
`withMusicProperty` ... 771
`wordwrap` ... 239
`wordwrap-lines` ... 245

X

`X-offset` ... 536
x11, color de ... 217, 218
`x11-color` ... 217, 218
`xNote` ... 771
`xNotesOn` ... 771