

LilyPond

El tipografiador de música

Manual del usuario

El equipo de desarrollo de LilyPond

Copyright © 1999–2007 por los autores

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

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.

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”.

(For LilyPond version 2.11.31)

Índice General

Preámbulo	1
1 Introducción	2
1.1 Grabado	2
1.2 Grabado automático.....	3
1.3 ¿Qué símbolos grabar?.....	5
1.4 Representación musical	6
1.5 Aplicaciones de ejemplo	8
1.6 Sobre el presente manual	9
2 Tutorial	12
2.1 Primeros pasos	12
2.1.1 Compilar un archivo.....	12
2.1.2 Notación sencilla	13
2.1.3 Trabajar sobre archivos de texto.....	17
2.1.4 Cómo leer el tutorial	18
2.2 Notación en un solo pentagrama.....	18
2.2.1 Nombres de nota relativos	18
2.2.2 Alteraciones accidentales y armaduras	18
2.2.3 Ligaduras de unión y de expresión	20
2.2.4 Articulaciones y matices dinámicos	21
2.2.5 Barras automáticas y manuales	22
2.2.6 Comandos rítmicos avanzados	23
2.3 Varias notas a la vez	24
2.3.1 Explicación de las expresiones musicales	24
2.3.2 Varios pentagramas.....	26
2.3.3 Sistemas de piano	27
2.3.4 Polifonía en un solo pentagrama	27
2.3.5 Combinar notas para formar acordes	28
2.4 Canciones	29
2.4.1 Imprimir la letra	29
2.4.2 Hojas guía de acordes	30
2.5 Retoques finales.....	31
2.5.1 Número de la versión	31
2.5.2 Añadir títulos	31
2.5.3 Nombres de nota absolutos	32
2.5.4 Organizar las piezas mediante identificadores.....	33
2.5.5 Más allá del tutorial.....	34
2.5.6 Cómo leer el manual.....	34
3 Juntándolo todo	35
3.1 Extender las plantillas.....	35
3.2 Cómo funcionan los archivos de LilyPond.....	38
3.3 La partitura es una sola expresión musical	39
3.4 Una particella orquestal	41

4	Trabajar en proyectos de LilyPond	43
4.1	Sugerencias para escribir archivos de LilyPond	43
4.1.1	Sugerencias de tipo general	43
4.1.2	Tipografiar música existente	44
4.1.3	Proyectos grandes	44
4.2	Ahorrar tecleo mediante identificadores y funciones	44
4.3	Hojas de estilo	46
4.4	Actualizar archivos antiguos	50
4.5	Resolución de problemas (tomar cada parte por separado)	50
4.6	Ejemplos mínimos	51
5	Trucar la salida	52
5.1	Mover objetos	52
5.2	Arreglar notación con superposiciones	54
5.3	Trucos comunes	55
5.4	Archivos por omisión	57
5.5	Encajar la música en menos páginas	58
5.6	Trucos avanzados con Scheme	59
5.7	Evitar los trucos con un proceso ralentizado	60
6	Notación básica	61
6.1	Alturas	61
6.1.1	Alturas normales	61
6.1.2	Alteraciones accidentales	62
6.1.3	Alteraciones de precaución	63
6.1.4	Microtonos	63
6.1.5	Nombres de las notas en otros idiomas	64
6.1.6	Octavas relativas	64
6.1.7	Comprobación de la octava	66
6.1.8	Transposición	66
6.1.9	Silencios	67
6.1.10	Desplazamientos	68
6.2	Ritmos	69
6.2.1	Duraciones	69
6.2.2	Puntillos	69
6.2.3	Grupos especiales	70
6.2.4	Escalar las duraciones	72
6.2.5	Comprobación del compás	73
6.2.6	Comprobación del número de compás	73
6.2.7	División automática de las notas	73
6.3	Polifonía	74
6.3.1	Acordes	74
6.3.2	Plicas	74
6.3.3	Polifonía básica	75
6.3.4	Voces explícitas	76
6.3.5	Resolución de las colisiones	78
6.4	Notación de los pentagramas	80
6.4.1	Clave	80
6.4.2	Armadura de la tonalidad	82
6.4.3	Indicación de compás	83
6.4.4	Compases parciales	84
6.4.5	Barras de compás	85
6.4.6	Música sin compasear	87

6.4.7	Delimitadores del comienzo de un sistema.....	87
6.4.8	El símbolo del pentagrama.....	89
6.4.9	Escribir música en paralelo.....	90
6.5	Conectar notas.....	91
6.5.1	Ligaduras de unión.....	91
6.5.2	Ligaduras de expresión.....	93
6.5.3	Ligaduras de fraseo.....	94
6.5.4	Ligaduras Laissez vibrer.....	95
6.5.5	Barras automáticas.....	95
6.5.6	Barras manuales.....	95
6.5.7	Notas de adorno.....	97
6.6	Expresiones.....	100
6.6.1	Articulaciones.....	100
6.6.2	Indicaciones de digitación.....	102
6.6.3	Matices dinámicos.....	103
6.6.4	Marcas de respiración.....	106
6.6.5	Trinos.....	106
6.6.6	Glissando.....	107
6.6.7	Arpeggio.....	107
6.6.8	Caídas y elevaciones.....	109
6.7	Repeticiones.....	109
6.7.1	Tipos de repetición.....	109
6.7.2	Sintaxis de las repeticiones.....	109
6.7.3	Las repeticiones y el MIDI.....	111
6.7.4	Comandos de repetición manual.....	112
6.7.5	Repeticiones de trémolo.....	113
6.7.6	Subdivisiones del trémolo.....	113
6.7.7	Repeticiones de compás.....	114
7	Notación específica de instrumentos.....	115
7.1	Música de piano.....	115
7.1.1	Cambios de pentagrama automáticos.....	115
7.1.2	Cambios de pentagrama manuales.....	115
7.1.3	Pedales.....	115
7.1.4	Líneas de cambio de pentagrama.....	115
7.1.5	Plicas de pentagrama cruzado.....	115
7.2	Nombres de acorde.....	115
7.2.1	Escritura de los nombres de acorde.....	115
7.2.2	Modo de acordes.....	115
7.2.3	Impresión de los nombres de acorde.....	115
7.3	Música vocal.....	115
7.3.1	Elaborar canciones sencillas.....	115
7.3.2	Introducir la letra.....	115
7.3.3	Hyphens and extenders.....	115
7.3.4	The Lyrics context.....	115
7.3.5	Melismata.....	116
7.3.6	Otra manera de introducir la letra.....	116
7.3.7	Flexibilidad en la colocación.....	116
7.3.7.1	Una sílaba de la letra sobre varias notas de un melisma.....	116
7.3.7.2	Letras en divisi.....	116
7.3.7.3	Cambiar la melodía dependiendo de la letra.....	116
7.3.7.4	Specifying melismata within the lyrics.....	116
7.3.7.5	Letra independiente de las notas.....	116
7.3.8	Espaciar las sílabas de la letra.....	116

7.3.9	Más acerca de los versos	116
7.3.9.1	Adding stanza numbers	116
7.3.9.2	Añadir expresiones dinámicas	116
7.3.9.3	Añadir el nombre de los cantantes	116
7.3.9.4	Printing stanzas at the end	116
7.3.9.5	Printing stanzas at the end in multiple columns	116
7.3.10	Tesitura	116
7.3.11	Otros temas relacionados con la música vocal	116
7.4	Música rítmica	116
7.4.1	Mostrar los ritmos de la melodía	116
7.4.2	Introducir la percusión	117
7.4.3	Pautas de percusión	117
7.4.4	Notas fantasma	117
7.5	Guitarra	117
7.5.1	Indicación de los números de cuerda	117
7.5.2	Fundamentos de las tablaturas	117
7.5.3	Tablaturas para otros instrumentos	117
7.5.4	Tablaturas de banjo	117
7.5.5	Diagramas de trastes	117
7.5.6	Digitaciones de la mano derecha	117
7.5.7	Otros temas relacionados con la música de guitarra	117
7.6	Gaita	117
7.6.1	Definiciones para la gaita	117
7.6.2	Ejemplo de música de gaita	117
7.7	Notación antigua	117
7.7.1	Cabezas de nota antiguas	117
7.7.2	Alteraciones accidentales antiguos	118
7.7.3	Silencios antiguos	118
7.7.4	Claves antiguas	118
7.7.5	Corchetes antiguos	118
7.7.6	Indicaciones antiguas de compás	118
7.7.7	Articulaciones antiguas	118
7.7.8	Custos	118
7.7.9	Divisiones	118
7.7.10	Ligaduras	118
7.7.10.1	Ligaduras mensurales blancas	118
7.7.10.2	Ligaduras de neumas cuadrados gregorianos	118
7.7.11	Contextos del Canto Gregoriano	118
7.7.12	Contextos de la música mensural	118
7.7.13	Alteraciones de la Musica Ficta	118
7.7.14	Bajo figurado	118
7.8	Notación específica de otros instrumentos	118
7.8.1	Armónicos artificiales de los instrumentos de cuerda	118
8	Notación avanzada	119
8.1	Texto	119
8.1.1	Guiones de texto	119
8.1.2	Extensiones de texto y de línea	119
8.1.3	Extensiones de texto	119
8.1.4	Indicaciones de texto	119
8.1.5	Marcado del texto	119
8.1.6	Partituras anidadas	119
8.1.7	Page wrapping text	119
8.1.8	Panorámica de los comandos de marcado de texto	119

8.1.9	Overview of text markup list commands	127
8.1.10	Selección de la tipografía	127
8.1.11	Indicaciones dinámicas contemporáneas	127
8.2	Preparación de particellas	128
8.2.1	Silencios multicompás	128
8.2.2	Indicaciones metronómicas	128
8.2.3	Llamadas de ensayo	128
8.2.4	Numeración de compases	128
8.2.5	Nombres de instrumentos	128
8.2.6	Transposición de los instrumentos	128
8.2.7	Corchetes de octava	128
8.2.8	Distintas ediciones a partir de una misma fuente	128
8.3	Música orquestal	128
8.3.1	Combinación automática de las partes	128
8.3.2	Ocultar pentagramas	128
8.3.3	Citar otras voces	128
8.3.4	Formateo de las notas de aviso	128
8.3.5	Alinear con una cadenza	128
8.4	Notación contemporánea	128
8.4.1	Notación polimétrica	129
8.4.2	Gestión del tiempo	129
8.4.3	Notación proporcional (introducción)	129
8.4.4	Racimos (clusters)	129
8.4.5	Cabezas de nota especiales	129
8.4.6	Barras punteadas	129
8.4.7	Improvisación	129
8.4.8	Seleccionar el tamaño de la tipografía para la notación	129
8.5	Utilización educativa	129
8.5.1	Globos de ayuda	129
8.5.2	Papel de música en blanco	129
8.5.3	Notas ocultas	129
8.5.4	Cabezas de notas con formas diversas	129
8.5.5	Cabezas de notas de Notación Fácil	129
8.5.6	Corchetes de análisis	129
8.5.7	Colorear objetos	129
8.5.8	Paréntesis	129
8.5.9	Líneas de rejilla	130
9	Cambiar los valores por omisión	131
9.1	Notación automática	131
9.1.1	Alteraciones accidentales automáticas	131
9.1.2	Establecer el comportamiento de las barras automáticas	131
9.2	Contextos de interpretación	131
9.2.1	Explicación de los contextos	131
9.2.2	Crear contextos	131
9.2.3	Cambiar las propiedades de un contexto al vuelo	131
9.2.4	Modificar los complementos (plug-ins) de contexto	131
9.2.5	Ajuste fino de la presentación dentro de los contextos	131
9.2.6	Cambiar los valores por omisión de los contextos	131
9.2.7	Definir contextos nuevos	131
9.2.8	Alinear los contextos	131
9.2.9	Vertical grouping of grobs	131
9.3	El comando <code>\override</code>	131
9.3.1	Construir un truco	131

9.3.2	Navegar por la referencia del programa	132
9.3.3	Interfaces de la presentación	132
9.3.4	Determinar la propiedad del grob	132
9.3.5	Objetos conectados con la entrada	132
9.3.6	Usar código de Scheme en lugar de <code>\tweak</code>	132
9.3.7	<code>\set</code> frente a <code>\override</code>	132
9.3.8	Trucos difíciles	132
10	Notación no musical	133
10.1	Archivos de entrada	133
10.1.1	Estructura del archivo (introducción)	133
10.1.2	Estructura del archivo	133
10.1.3	Una expresión musical única	133
10.1.4	Varias partituras en un libro	133
10.1.5	Extraer fragmentos de notación	133
10.1.6	Inclusión de archivos de LilyPond	133
10.1.7	Codificación del texto	133
10.2	Títulos y cabeceras	133
10.2.1	Crear títulos	133
10.2.2	Títulos personalizados	133
10.2.3	Reference to page numbers	133
10.2.4	Table of contents	133
10.3	Salida MIDI	133
10.3.1	Crear archivos MIDI	133
10.3.2	El bloque MIDI	134
10.3.3	Nombres de los instrumentos MIDI	134
10.4	Presentar la notación de LilyPond	134
10.5	Saltar la música corregida	134
11	Problemas de espaciado	135
11.1	El papel y las páginas	135
11.1.1	Tamaño del papel	135
11.1.2	Formateo de las páginas	135
11.2	Disposición de la música	135
11.2.1	Establecer el tamaño del pentagrama	135
11.2.2	Disposición de la partitura	135
11.3	Mostrar el espaciado	135
11.4	Salto	135
11.4.1	Salto de línea	135
11.4.2	Salto de página	135
11.4.3	Salto de página óptimos	135
11.4.4	Paso de página óptimo	135
11.4.5	Salto de línea explícitos	135
11.4.6	Utilizar una voz adicional para los saltos de línea	135
11.5	Espaciado vertical	136
11.5.1	Espaciado vertical dentro de un sistema	136
11.5.2	Espaciado vertical entre sistemas	136
11.5.3	Posicionamiento explícito de los pentagramas y los sistemas	136
11.5.4	Espaciado vertical en dos fases	136
11.5.5	Evitar las colisiones verticales	136
11.6	Espaciado horizontal	136
11.6.1	Panorámica del espaciado horizontal	136
11.6.2	Área de espaciado nueva	136

11.6.3	Cambiar el espaciado horizontal	136
11.6.4	Longitud de la línea	136
11.6.5	Notación proporcional	136
12	Interfaces para programadores	137
12.1	Funciones musicales	137
12.1.1	Panorámica de las funciones musicales	137
12.1.2	Funciones de sustitución sencillas	137
12.1.3	Funciones de sustitución en parejas	137
12.1.4	Matemáticas dentro de las funciones	137
12.1.5	Funciones vacías	137
12.1.6	Funciones sin argumentos	137
12.1.7	Panorámica de las funciones musicales disponibles	137
12.2	Interfaces para el programador	140
12.2.1	Variables de entrada y Scheme	140
12.2.2	Representación interna de la música	140
12.3	Construcción de funciones complejas	140
12.3.1	Presentación de expresiones musicales	140
12.3.2	Propiedades de la música	140
12.3.3	Doblar una nota con ligaduras (ejemplo)	140
12.3.4	Añadir articulación a las notas (ejemplo)	140
12.4	Interfaz de marcado para el programador	140
12.4.1	Construcción del marcado en Scheme	140
12.4.2	Cómo funciona internamente el marcado	140
12.4.3	Definición de un comando nuevo de marcado	141
12.4.4	New markup list command definition	141
12.5	Contextos para programadores	141
12.5.1	Evaluación de contextos	141
12.5.2	Ejecutar una función sobre todos los objetos de la presentación	141
12.6	Procedimientos de Scheme como propiedades	141
Apéndice A	Lista bibliográfica	142
Apéndice B	Tutorial de Scheme	143
Apéndice C	Tablas del manual sobre notación	144
C.1	Carta de nombres de acordes	144
C.2	Instrumentos MIDI	144
C.3	Lista de colores	144
C.4	La tipografía Feta	144
C.5	Estilos de cabezas de nota	144

Apéndice D	Plantillas	145
D.1	Pentagrama único	145
D.1.1	Sólo notas	145
D.1.2	Notas y letra	145
D.1.3	Notas y acordes	145
D.1.4	Notas, letra y acordes	145
D.2	Plantillas de piano	145
D.2.1	Piano solo	145
D.2.2	Piano y melodía con letra	145
D.2.3	Piano con letra centrada	145
D.2.4	Piano con dinámicas centradas	145
D.3	Cuarteto de cuerda	145
D.3.1	Cuarteto de cuerda	145
D.3.2	Particellas de cuarteto de cuerda	145
D.4	Conjuntos vocales	145
D.4.1	Partitura vocal SATB	145
D.4.2	Partitura vocal SATB y reducción para piano automática	145
D.4.3	SATB con contextos alineados	145
D.5	Plantillas para notación antigua	145
D.5.1	Transcripción de música mensural	145
D.5.2	Plantilla para transcripción de canto gregoriano	145
D.6	Combo de jazz	145
D.7	Plantillas de lilypond-book	146
D.7.1	LaTeX	146
D.7.2	Texinfo	146
Apéndice E	Hoja de referencia rápida	147
Apéndice F	Licencia de documentación libre de GNU	151
F.0.1	ADDENDUM: cómo utilizar esta licencia para sus documentos	156
Apéndice G	Índice de comandos de LilyPond	157
Apéndice H	Índice de LilyPond	159

Preámbulo

Debió ser en el transcurso de un ensayo de la EJE (Joven Orquesta de Eindhoven), allá por 1995 cuando Jan, uno de los violistas chiflados, le habló a Han-Wen, uno de los trompistas distorsionados, acerca del gran proyecto en que estaba trabajando. Era un sistema automático para imprimir música (para ser exactos se trataba de MPP, un preprocesador para MusiXTeX). Ocurrió entonces que Han-Wen quiso imprimir unas particellas sacadas de una partitura, y así empezó a echarle un vistazo al programa, y en seguida se quedó estancado. Decidieron que MPP era un callejón sin salida. Después de muchísimo filosofar y de montañas de encendidas conversaciones por correo electrónico, Han-Wen inició el proyecto LilyPond en 1996. Esta vez fue Jan quien resultó absorbido por el nuevo proyecto de Han-Wen.

En ciertos aspectos, desarrollar un programa de ordenador es como aprender a tocar un instrumento. Al principio es divertido descubrir cómo funciona, y supone un divertido reto intentar aquello que no eres capaz de hacer. Una vez pasado el entusiasmo inicial, hay que practicar más y más. Las escalas y los estudios pueden llegar a aturdir, y si no está motivado por otras personas (profesores, directores o el público) uno siempre está tentado de abandonarlo. Uno persevera y, poco a poco, tocar se convierte en parte de la vida de uno. Algunos días se acoge de forma natural, y es estupendo, y otros simplemente la cosa no funciona, pero uno sigue tocando día tras día.

Igual que hacer música, trabajar en LilyPond puede ser un trabajo muy duro y hay días en que uno se siente como pisando un hormiguero. A pesar de todo, se ha convertido en parte de nuestra vida y seguimos haciéndolo. Con toda probabilidad la motivación más importante es que nuestro programa realmente hace algo útil por las personas. Cuando navegamos por la red encontramos mucha gente que utiliza LilyPond y produce unas partituras impresionantes con él. De esta observación se desprende una sensación algo irreal, pero muy agradable.

Nuestros usuarios no sólo nos transmiten buenas vibraciones por usar el programa, también muchos de ellos nos ayudan enviando sugerencias e informes de fallo, por ello nos gustaría agradecer a todos los usuarios que nos han enviado estos informes, emitido sugerencias o contribuido a LilyPond de cualquier otra forma.

Tocar e imprimir música es algo más que una bonita analogía. Programar juntos es muy divertido, y ayudar a las personas es algo profundamente gratificante, pero en último término trabajar en LilyPond es una forma de expresar nuestro profundo amor por la música. ¡Ojalá le ayude a elaborar montañas de preciosas partituras!

Han-Wen y Jan

Utrecht/Eindhoven, Holanda, julio de 2002.

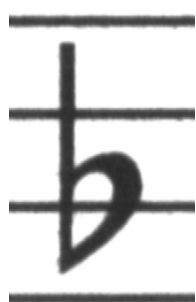
1 Introducción

1.1 Grabado

El arte de la tipografía musical se conoce como *grabado (en plancha)*. El término deriva del proceso tradicional de la impresión musical. Hace sólo unas décadas, la música impresa se hacía estampando la música sobre planchas de zinc o estaño de forma invertida como en un espejo. Después la plancha se entintaba y las depresiones causadas por los cortes y estampados retenían la tinta. Al presionar una hoja de papel sobre la plancha, se formaba una imagen. El estampado y cortado se hacía completamente a mano. Cualquier corrección era muy fastidiosa de realizar, si es que era posible hacerla siquiera, así que el grabado tenía que quedar perfecto a la primera. El grabado era una habilidad altamente especializada; un artesano necesitaba unos cinco años de preparación antes de poder ostentar el título de maestro grabador, y se necesitaban otros cinco años de experiencia para ser un auténtico experto.

Hoy en día, toda la música impresa nueva se produce con ordenadores. Esto tiene unas ventajas evidentes: las copias son más baratas de producir y el trabajo editorial se puede repartir por correo electrónico. Desgraciadamente la penetrante utilización de ordenadores también ha hecho disminuir la calidad gráfica de las partituras. Las impresiones de ordenador tienen un aspecto insulso y mecánico, lo que hace que sea desagradable tocar a partir de ellas.

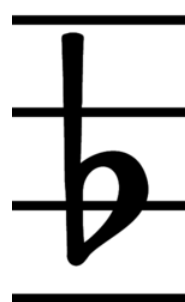
Las imágenes siguientes ilustran la diferencia entre el grabado tradicional y la salida típica de ordenador, y la tercera imagen muestra cómo LilyPond imita el aspecto tradicional. La imagen de la izquierda presenta el dibujo escaneado de un símbolo de bemol sacado de una edición publicada en el año 2000. La del centro es un símbolo procedente de una edición de Bärenreiter grabada a mano de la misma música. La de la izquierda ilustra los típicos puntos débiles de la impresión por ordenador: las líneas del pentagrama son muy delgadas, el peso del símbolo del bemol es también demasiado ligero como las líneas del pentagrama, y tiene una apariencia rectilínea con esquinas afiladas. En contraste, el bemol de Bärenreiter tiene una apariencia redonda, pesada, casi voluptuosa. Nuestro símbolo del bemol se diseñó según éste, entre otros. Es de forma redondeada y su peso está en armonía con el grosor de nuestras líneas de pentagrama, que son asimismo mucho más gruesas que las de la edición por ordenador.



Henle (2000)



Bärenreiter (1950)

Tipografía Feta de
LilyPond (2003)

Tratándose del espaciado, la distribución del espacio debe reflejar las duraciones que hay entre las notas. Sin embargo muchas partituras modernas se atañen a las duraciones con precisión matemática, lo que lleva a unos resultados bastante pobres. En el siguiente ejemplo se muestra un ejemplo dos veces: una utilizando espaciado matemáticamente exacto, y otra con ciertas correcciones. ¿Puede adivinar cuál es cuál?



Cada uno de los dos compases de este fragmento tiene solamente notas de duración constante. El espaciado debería reflejarlo. Desgraciadamente el ojo nos engaña un poco; no solamente percibe la distancia entre las cabezas de las notas, sino que tiene también en cuenta la distancia entre las plicas. Como resultado, las notas de una combinación plica arriba/plica abajo se tendrían que separar más, y las notas de una combinación plica abajo/plica arriba deberían juntarse, todo ello dependiendo de las posiciones combinadas de las notas. Los dos compases de arriba están impresos con esta corrección y los de abajo sin ella, formando grupos de notas pegadas con plica abajo/plica arriba.

Los músicos están normalmente más concentrados en tocar que en estudiar el aspecto de una partitura, y por ello las pequeñeces sobre los detalles tipográficos pueden parecer académicas. Pero no lo son. En las partituras más largas con ritmos monótonos, las correcciones de espaciado llevan a sutiles variaciones en la disposición de cada una de las líneas dándoles una especie de firma visual distintiva. Sin esta firma, todas las líneas parecerían iguales, y se convertirían en un laberinto. Si un músico aparta la mirada o tiene un lapsus de concentración, las líneas podrán perder su lugar sobre el papel.

De forma similar, la fuerza visual de unos símbolos pesados sobre gruesas líneas de pentagrama se sostiene mejor cuando el lector se aleja del papel, por ejemplo cuando está sobre un atril. Una distribución cuidadosa del espacio blanco permite disponer la música muy apretada sin que los símbolos se toquen unos a otros. El resultado reduce a un mínimo las vueltas de página, lo que es una gran ventaja.

Ésta es una característica normal del arte tipográfico. La disposición de la página tiene que ser bonita, no sólo por sí misma, sino sobre todo porque así ayuda al lector en su tarea. Para los materiales destinados a la interpretación, como las partituras, esto es de una importancia doble: los músicos tienen una capacidad de concentración limitada. Cuanta menos atención necesiten para el acto de leer, más se pueden dedicar al acto de tocar la música. Dicho de otra forma: una mejor tipografía se traduce en una mejor interpretación.

Estos ejemplos demuestran que la tipografía musical es un arte sutil y complejo, y que su elaboración requiere una experiencia considerable, que los músicos no suelen tener. LilyPond representa nuestro esfuerzo para llevar la excelencia visual de la música grabada a mano a la era de la informática, y ponerla a disposición de los músicos normales. Hemos ido afinando nuestros algoritmos, diseños de tipografía y preferencias del programa para producir una impresión cuya calidad se equipara con la de las viejas ediciones que tanto nos gusta contemplar y de las que tanto nos gusta tocar.

1.2 Grabado automático

¿Cómo nos las arreglamos para implementar la tipografía? Si un artesano necesita más de diez años para convertirse en un auténtico maestro ¿cómo vamos a poder nosotros, simples hackers, escribir un programa que les quite el trabajo?

La respuesta es: ¡no podemos! Puesto que la tipografía se fundamenta en el juicio humano sobre la apariencia, nunca se puede sustituir completamente a las personas. Sin embargo, se puede automatizar gran parte del trabajo más duro y repetitivo. Si LilyPond resuelve la mayoría de las situaciones comunes de forma correcta, esto ya será una tremenda mejoría sobre los programas existentes. El resto de los casos se podrán afinar a mano. Con el transcurso de los

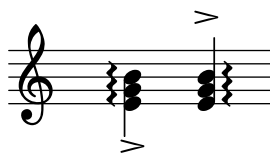
años, el software se puede refinar para que haga un mayor número de cosas de forma automática, de tal forma que los ajustes manuales tienden a ser cada vez menos necesarios.

Cuando empezamos, escribimos el programa LilyPond completamente en el lenguaje C++; la funcionalidad del programa quedaba como esculpida en piedra por los desarrolladores. Este esquema resultó no ser muy satisfactorio por una serie de motivos:

- Cuando LilyPond comete fallos, los usuarios tienen la necesidad de superar las decisiones de formateo. Por ello el usuario debe tener acceso al motor de formateo. De aquí que no podamos dejar establecidas las reglas y valores durante la compilación, sino que los usuarios deben poder acceder a ellos durante la ejecución del programa.
- El grabado de música es cosa de juicio visual y por ello es cuestión de gustos. A pesar de saber tanto como creemos saber, los usuarios pueden no estar de acuerdo con nuestras decisiones personales. Por tanto la definición del estilo tipográfico también debe estar al alcance del usuario.
- Por último, estamos continuamente refinando los algoritmos de formateo y por tanto necesitamos un enfoque flexible para las reglas. El lenguaje C++ fuerza un cierto método para agrupar las reglas que no encaja bien con la manera de funcionar de la notación musical.

Estos problemas se han solucionado integrando un intérprete del lenguaje Scheme y reescribiendo parte del código de LilyPond en Scheme. La actual arquitectura de formateo se construye alrededor del concepto de objetos gráficos, descrita por variables y funciones de Scheme. Esta arquitectura puede tratar al mismo tiempo con las reglas de formateo, el estilo tipográfico y las decisiones de formateo individuales. El usuario tiene acceso directo a la mayor parte de estos controles.

Las variables de Scheme controlan las decisiones de formateo. Por ejemplo, muchos objetos gráficos tienen una variable de dirección que codifica la elección entre arriba y abajo (o izquierda y derecha). Aquí puede ver dos acordes con acentos y signos de arpeggio. En el primer acorde los objetos gráficos tienen todas sus direcciones hacia abajo (o hacia la izquierda). El segundo acorde tiene todas las direcciones hacia arriba (o hacia la derecha).

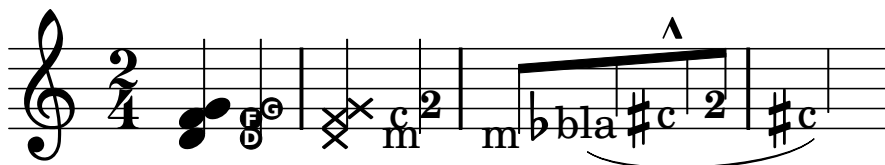


El proceso de formatear una partitura consiste en leer y escribir las variables de los objetos gráficos. Ciertas variables tienen un valor predefinido. Por ejemplo, el grosor de muchas líneas (una característica del estilo tipográfico) son variables con un valor preestablecido. Podemos alterar este valor libremente dando así a nuestra partitura una impresión tipográfica distinta.



Las reglas de formateo también son variables que están predefinidas: cada objeto tiene unas variables que contienen procedimientos. Estos procedimientos realizan el trabajo real de formateo y sustituyéndolos por otros podemos alterar el aspecto de los objetos. En el siguiente

ejemplo, la regla que define cómo se dibuja la cabeza de una nota se altera durante el transcurso del fragmento musical.



1.3 ¿Qué símbolos grabar?

El proceso de formateo toma las decisiones sobre dónde colocar los símbolos. Sin embargo esto sólo se puede hacer una vez que se ha decidido *qué* símbolos han de imprimirse, o dicho de otro modo: qué notación utilizar.

La notación musical común es un sistema de registro de música que ha venido evolucionando desde hace mil años. La forma que se usa en nuestros días data de los primeros tiempos del Renacimiento. Aunque la forma básica (es decir: puntos sobre una pauta de cinco líneas) no ha cambiado, los detalles continúan evolucionando para expresar todas las innovaciones de la notación contemporánea. Por tanto abarca unos quinientos años de música. Sus aplicaciones se extienden sobre un amplio rango que abarca desde melodías monofónicas hasta monstruosos contrapuntos para gran orquesta.

¿Cómo podemos tratar con una bestia de tantas cabezas, y obligarla a que se encierre dentro de los límites de un programa de ordenador? Nuestra solución es trocear el problema de la notación (por oposición al grabado, esto es, a la tipografía) en fragmentos digeribles y más fáciles de programar: cada tipo de símbolo se maneja por un módulo separado que recibe el nombre de plug-in. Cada plug-in es completamente modular e independiente, de forma que puede desarrollarse y mejorarse por separado. Estos plug-ins se llaman **engravers** (grabadores), por analogía con los artesanos que traducen las ideas musicales a símbolos gráficos.

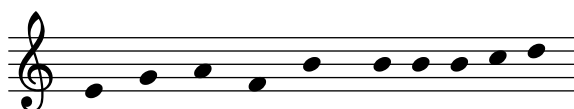
En el siguiente ejemplo vemos cómo comenzamos con un plug-in para las cabezas de las notas, el `Note_heads_engraver`.



A continuación un `Staff_symbol_engraver` (grabador del pentagrama) añade las líneas de la pauta.



El **Clef_engraver** (grabador de la clave) define un punto de referencia para la altura de las notas en el pentagrama.



y el `Stem_engraver` (grabador de las plicas) añade las plicas.



El `Stem_engraver` (grabador de plicas) recibe una notificación cuando llega una cabeza. Cada vez que se ve una cabeza (o más, si es un acorde), se crea un objeto plica y se conecta a la cabeza. Añadiendo grabadores para las barras, ligaduras, acentos, alteraciones, líneas divisorias, indicación de compás y armadura conseguimos una notación completa.



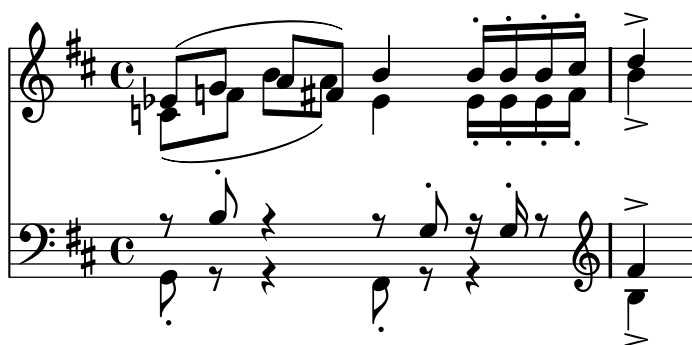
Este sistema funciona bien para la música monofónica, pero ¿y con la polifonía? En notación polifónica muchas voces pueden compartir el mismo pentagrama.



En esta situación, las alteraciones y la pauta se comparten, pero las plicas, ligaduras, barras, etc. son propias de cada voz. Por tanto los grabadores han de agruparse. Los grabadores de cabezas, plicas, ligaduras, etc. se unen en un grupo llamado ‘Contexto de voz’, mientras que los grabadores de la armadura, alteraciones, compás, etc. van a un grupo que se llama ‘Contexto de la pauta’. En el caso de la polifonía, un único Contexto de pauta contiene más de un Contexto de voz. De forma semejante, varios Contextos de pauta pueden agruparse en un único Contexto de partitura. El Contexto de partitura es el contexto de notación de más alto nivel.

Véase también

Referencia del programa: `Contexts`.



1.4 Representación musical

Idealmente el formato de entrada para cualquier sistema de formateo de alto nivel es una descripción abstracta del contenido. En este caso, eso constituiría la propia música, lo que plantea un tremendo problema: ¿cómo podemos definir qué es realmente la música? En lugar de intentar hallar una respuesta, le hemos dado la vuelta a la pregunta. Escribimos un programa capaz de producir partituras y ajustamos el formato para que sea tan escueto como sea posible. Cuando el formato ya no puede reducirse más, por definición nos habremos quedado con el contenido musical propiamente dicho. Nuestro programa sirve como definición formal de un documento musical.

La sintaxis también es el interfaz de usuario de LilyPond, así que es fácil teclear

c'4 d'8

un Do1 (Do central) negra, y un Re1 (el Re por encima del Do central) corchea.



A una escala microscópica, dicha sintaxis es fácil de utilizar. A una escala mayor, la sintaxis también requiere una estructura. ¿De qué otra forma podríamos introducir piezas complejas como sinfonías u óperas? La estructura se forma mediante el concepto de expresiones musicales: al combinar pequeños fragmentos de música dentro de otros mayores, se pueden expresar ideas musicales más complejas. Por ejemplo

c4



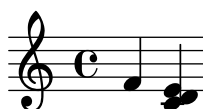
Los acordes se pueden construir encerrando las notas entre << y >>

<<c4 d4 e4>>



Esta expresión se coloca en secuencia encerrándola dentro de llaves { ... }

{ f4 <<c4 d4 e4>> }



Lo anterior, a su vez también es una expresión, y por ello se puede combinar de nuevo con otra expresión simultánea (una blanca) usando <<, \\\, y >>

<< g2 \\\ { f4 <<c4 d4 e4>> } >>



Las mencionadas estructuras recursivas se pueden especificar de forma nítida y formal dentro de una gramática independiente del contexto. El código de análisis también se genera a partir de esta gramática. En otras palabras, la sintaxis de LilyPond está definida de una forma clara y sin ambigüedades.

Los interfaces de usuario y la sintaxis son lo que la persona ve y con lo que trata principalmente. En parte, son fruto de preferencias personales y como tales están sujetas a mucha

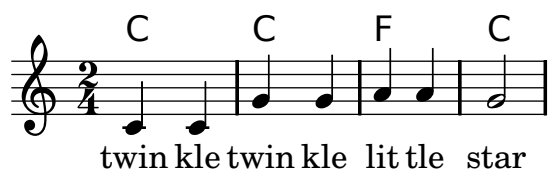
discusión. Aunque las discusiones sobre el gusto tienen su mérito, no son demasiado productivas. Dentro de la escena global de LilyPond, la sintaxis de la entrada tiene una importancia relativamente pequeña: inventarse una sintaxis elegante es fácil, pero escribir un código de formato decente es mucho más difícil. Esto también queda ilustrado por la cantidad de líneas de código de los componentes respectivos: el análisis y la representación se llevan menos del 10% del código fuente.

1.5 Aplicaciones de ejemplo

Escribimos LilyPond como un experimento de cómo condensar el arte del grabado de música dentro de un programa de ordenador. Gracias a todo este duro trabajo, el programa ahora se puede usar para hacer trabajos útiles. La aplicación más sencilla es imprimir notas.



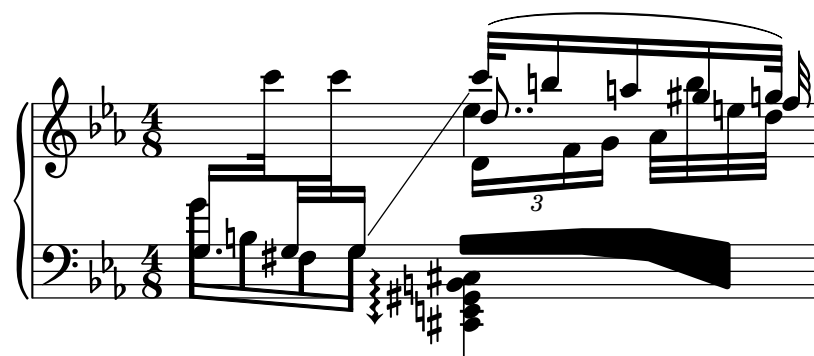
Añadiendo los nombres de acordes y la letra, obtenemos una hoja guía de acordes (lead sheet).



También se pueden imprimir notación polifónica y música para piano. El ejemplo siguiente combina algunas otras construcciones exóticas.

Screech and boink Random complex notation

Han-Wen Nienhuys



Todos los fragmentos mostrados se han escrito a mano, pero esto no es necesariamente así. Puesto que el motor de formato es casi completamente automático, puede servir como medio de salida para otros programas que manipulan música. Por ejemplo, se puede usar también para convertir bases de datos de fragmentos musicales en imágenes con destino a páginas web y presentaciones multimedia.

Este manual también es un ejemplo de aplicación: el formato de entrada es texto sencillo, y por ello se puede empotrar fácilmente dentro de otros formatos basados en texto, como \LaTeX , HTML, o en el caso concreto de este manual, Texinfo. A través de un programa especial, los fragmentos de entrada se pueden sustituir por imágenes musicales dentro de los archivos de salida PDF o HTML resultantes. Esto convierte la tarea de mezclar música y texto dentro de los documentos, en algo muy sencillo.

1.6 Sobre el presente manual

El manual se divide en cuatro libros.

Learning manual

Este libro explica cómo empezar el aprendizaje de LilyPond, así como algunos conceptos clave en términos sencillos.

- *Capítulo 2 [Tutorial], página 12* da una amable introducción a la tipografía musical. Los usuarios que se acercan por primera vez deben comenzar por aquí.
- *Capítulo 3 [Juntándolo todo], página 35* explica algunos conceptos generales sobre el formato de los archivos de entrada de LilyPond. Si no está seguro de dónde colocar un comando ¡lea este capítulo!
- *Capítulo 4 [Trabajar en proyectos de LilyPond], página 43* trata los usos prácticos de LilyPond y cómo evitar ciertos problemas bastante comunes.
- *Capítulo 5 [Trucar la salida], página 52* muestra la manera de cambiar el grabado por omisión que produce LilyPond.

Notation reference

Este libro explica todas las instrucciones de LilyPond que producen notación impresa. Da por supuesto que el lector está familiarizado con los conceptos del manual de aprendizaje.

- *Capítulo 6 [Notación básica], página 61* trata ciertos temas agrupados según las construcciones de notación. Esta sección proporciona detalles sobre notación básica que probablemente serán de utilidad en casi cualquier proyecto de notación.
- *Capítulo 7 [Notación específica de instrumentos], página 115* también trata los temas agrupados por construcciones de notación. Esta sección proporciona detalles sobre notación especial que solamente será útil para ciertos grupos de instrumentos (o voces).
- *Capítulo 8 [Notación avanzada], página 119* trata temas agrupados por construcciones de notación. Esta sección proporciona detalles acerca de cierta notación complicada o poco usual.
- *Capítulo 9 [Cambiar los valores por omisión], página 131* explica cómo hacer un ajuste fino de la presentación.
- *Capítulo 10 [Notación no musical], página 133* trata la salida no musical como los títulos, piezas de varios movimientos, y sobre cómo elegir los instrumentos MIDI.
- *Capítulo 11 [Problemas de espaciado], página 135* trata asuntos que afectan a la salida global, como la elección del tamaño del papel o la especificación de los saltos de página.
- *Capítulo 12 [Interfaces para programadores], página 137* explica cómo crear funciones musicales.

Program usage

Este libro explica la manera de ejecutar el programa y cómo integrar la notación de LilyPond con otros programas.

- *program usage manual*, **Running LilyPond** muestra la manera de ejecutar LilyPond y sus programas asistentes. Además, esta sección explica cómo actualizar los archivos de entrada de versiones de LilyPond anteriores.
- *program usage manual*, **LilyPond-book** da los detalles que se encuentran detrás de la creación de documentos con ejemplos de música insertados, como este manual.
- *program usage manual*, **Converting from other formats** explica cómo ejecutar los programas de conversión. Estos programas se proporcionan con el paquete LilyPond, y convierten una amplia variedad de formatos de música al formato `.ly`.

Appendices

Este libro contiene unos útiles cuadros de referencia.

- *Apéndice A [Lista bibliográfica], página 142* contiene un conjunto de libros de referencia muy útiles para aquellas personas que desean saber más sobre notación y grabado.
- El *Apéndice B [Tutorial de Scheme], página 143* presenta una breve introducción a Scheme, el lenguaje de programación que utilizan las funciones musicales.
- *Apéndice C [Tablas del manual sobre notación], página 144* son un conjunto de tablas que relacionan los nombres de los acordes, instrumentos MIDI, nombres de los colores y la tipografía Feta.
- *Apéndice D [Plantillas], página 145* de piezas de LilyPond. Sólo tiene que cortar y pegar una plantilla en un archivo de texto, escribir las notas y ¡ya está!
- La *Apéndice E [Hoja de referencia rápida], página 147* es una referencia manejable de los comandos de LilyPond más comunes.
- El *Apéndice G [Índice de comandos de LilyPond], página 157* es un índice de todos los `\commands` de LilyPond.
- El *Apéndice H [Índice de LilyPond], página 159* es un índice completo.

Other information

Existen otras fuentes de información que pueden resultar muy útiles.

- El glosario explica los términos musicales, e incluye traducciones a varios idiomas. Es un documento aparte, disponible en HTML y en PDF. Si no está familiarizado con la notación o la terminología musicales (sobre todo si no es un anglófono nativo), le recomendamos que consulte el glosario.
- Los fragmentos de código (snippets) son una enorme colección de pequeños ejemplos que ejemplifican toda clase de consejos, trucos y funcionalidades especiales de LilyPond. La mayor parte de estos fragmentos de código también se pueden encontrar en el **Archivo de fragmentos de código (snippets) de LilyPond**. Este sitio web también dispone de un manual de LilyPond en el que se pueden realizar búsquedas.
- La Referencia del programa es un conjunto de páginas HTML con una tupida red de enlaces cruzados, que documentan al detalle el meollo de todas y cada una de las clases, objetos y funciones de LilyPond. Se produce directamente a partir de las definiciones de formateo que se utilizan.

Casi toda la funcionalidad de formateo que se emplea internamente, se encuentra disponible para el usuario de forma directa. Por ejemplo, todas las variables que controlan los valores de grosor, distancias, etc., se pueden cambiar dentro de los archivos de entrada. Hay un enorme número de opciones de formateo, y todas ellas se describen en este documento. Cada sección del manual de notación tiene una subsección **Véase también**, que hace referencia a la documentación generada. En el documento HTML, estas subsecciones llevan enlaces que se pueden plusar.

Cuando ya sea un usuario con experiencia podrá usar el manual como referencia: hay un índice muy completo¹, pero el documento también está disponible en una sola página HTML, en la que es fácil buscar cualquier cosa utilizando la función de búsqueda de su navegador de web.

En todos los documentos HTML que tienen fragmentos de música incrustados, la entrada de LilyPond que se utilizó para producir dicha imagen se puede ver pulsando con el ratón sobre la imagen.

La localización exacta de los archivos de documentación que hemos mencionado puede variar de un sistema a otro. En ocasiones este manual hace referencia a archivos de inicialización y de ejemplo. A lo largo del manual, nos referimos a archivos de entrada por su ruta relativa respecto de directorio de nivel más alto de los archivos de código fuente. Por ejemplo, `'input/lsr/dirname/bla.ly'` puede referirse al archivo `'lilypond2.x.y/input/lsr/dirname/bla.ly'`. En los paquetes binarios para la plataforma Unix, normalmente la documentación y los ejemplos se encuentran en algún lugar dentro de `'/usr/share/doc/lilypond/'`. Los archivos de inicialización, como por ejemplo `'scm/lily.scm'`, o `'ly/engraver-init.ly'`, se encuentran normalmente en el directorio `'/usr/share/lilypond/'`.

Por último, este y el resto de los manuales están disponibles en línea tanto como archivos PDF como HTML en el sitio web, que encontrará en <http://www.lilypond.org/>.

¹ Si está buscando algo y no lo encuentra en el manual, eso se considera un bug (fallo). En este caso le rogamos que envíe un informe de fallo.

2 Tutorial

Este tutorial comienza con una introducción al lenguaje musical LilyPond y cómo producir música impresa. Después de este primer contacto, explicaremos cómo crear notación musical usual.

2.1 Primeros pasos

Esta sección le ofrece una introducción básica al trabajo con LilyPond.

2.1.1 Compilar un archivo

El primer ejemplo demuestra cómo empezar a trabajar con LilyPond. Para crear una partitura, escribimos un archivo de texto que especifica la notación. Por ejemplo, si escribimos

```
{
  c' e' g' e'
}
```

el resultado tiene este aspecto



Advertencia: Todo el código que se da como entrada a LilyPond tiene que ir entre { llaves }. Las llaves deberían también estar rodeadas por espacios a no ser que se encuentren al principio o al final de una línea, para evitar ambigüedades. Es posible que se omitan en algunos ejemplos del presente manual ¡pero no las omita en su propia música!

Además la entrada de LilyPond es **sensible a las mayúsculas**. { c d e } es una entrada válida; { C D E } producirá un mensaje de error.

Introducir música y ver la salida

En esta sección vamos a explicar qué comandos hay que ejecutar y cómo, para ver o imprimir el resultado.

MacOS X

Si hace doble click sobre **LilyPond.app**, se abrirá con un archivo de ejemplo. Guárdelo, por ejemplo, como ‘prueba.ly’ en el Escritorio, y a continuación procéselo con la orden de menú ‘**Compilar > Componer archivo**’. El PDF resultante se mostrará en la pantalla.

Advierta que la primera vez que ejecute LilyPond, tardará un minuto o dos porque todas las tipografías del sistema han de ser analizadas previamente.

Para posteriores usos de LilyPond, debería comenzar eligiendo ‘Nuevo’ o ‘Abrir’. Tiene que grabar el archivo antes de componerlo tipográficamente. Si se produce algún error durante el proceso, observe la ventana del registro.

Windows

En Windows, si hace doble click sobre el icono de LilyPond que está en el escritorio, se abrirá un sencillo editor de texto con un archivo de ejemplo. Guárdelo, por ejemplo, con el nombre ‘prueba.ly’ en el escritorio y después haga doble click sobre el icono del archivo para procesarlo (el icono tiene la forma de una nota). Transcurridos unos segundos, obtendrá un archivo ‘prueba.pdf’ en el escritorio. Haga doble click sobre este archivo PDF para ver la partitura

compuesta tipográficamente. Un método alternativo para procesar el archivo ‘prueba.ly’ es arrastrarlo y soltarlo sobre el icono de LilyPond utilizando el ratón.

Para editar un archivo ‘.ly’ existente, haga click sobre él con el botón derecho del ratón y elija “Edit source” (editar el archivo fuente). Para empezar con un archivo vacío, arranque el editor como se describe más arriba y elija “New” (nuevo) del menú “File” (archivo).

Al hacer doble click sobre el archivo no sólo se obtiene como resultado un archivo PDF, sino también un archivo ‘.log’ que contiene cierta información acerca de lo que LilyPond ha hecho con el archivo. Si se produce algún error, examine este archivo de registro.

Tenga en cuenta que hay disponibles varios editores de texto alternativos con un mejor apoyo para la realización de documentos de LilyPond, consulte program usage manual, **Editor support** para ver más información.

Unix

Comience abriendo una ventana de terminal e iniciando un editor de texto. Por ejemplo, puede abrir un xterm y ejecutar `joe`¹. En su editor de texto, introduzca la siguiente entrada y guarde el archivo como ‘prueba.ly’

```
{
  c' e' g' e'
}
```

Para procesar ‘prueba.ly’, haga lo siguiente:

```
lilypond prueba.ly
```

Verá algo parecido a:

```
lilypond prueba.ly
GNU LilyPond 2.10.0
Procesando `prueba.ly'
Analizando...
Interpretando la música... [1]
Preprocesando los objetos gráficos...
Calculando los cortes de línea... [2]
Escribiendo la página de salida en `prueba.ps'...
Convirtiendo a `prueba.pdf'...
```

El resultado es el archivo ‘prueba.pdf’ que podrá imprimir o ver con las utilidades estándar de su sistema operativo.²

2.1.2 Notación sencilla

LilyPond añadirá ciertos elementos de notación de manera automática. En el siguiente ejemplo hemos especificado solamente cuatro alturas, pero LilyPond ha añadido la clave, el compás y las duraciones.

```
{
  c' e' g' e'
}
```



¹ Existen macros para los adictos a VIM, y hay un `LilyPond-mode` (modo de LilyPond) para los adictos a Emacs. Si no han sido instalados aún, consulte el archivo ‘INSTALL.txt’. El entorno de edición más sencillo es ‘LilyPondTool’. Consulte program usage manual, **Editor support** para más información.

² Si su sistema no dispone de dichas herramientas, puede probar **Ghostscript**, un programa de libre disposición para ver e imprimir archivos PDF y PostScript.

Este comportamiento se puede modificar, pero en general estos valores automáticos son adecuados.

Alturas

La manera más sencilla de introducir las notas es mediante la utilización del modo `\relative`. En este modo se supone que el **intervalo** entre la nota anterior y la actual se encuentra dentro de una **cuarta**. Comenzamos introduciendo la pieza musical más elemental, una **escala**.

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



La nota inicial es Do **central**. Cada nota sucesiva se encuentra dentro de una cuarta de la nota previa (en otras palabras: la primera ‘c’ es el Do más cercano al Do central; a éste le sigue el Re más cercano a la nota previa, y así sucesivamente). Podemos crear melodías con intervalos mayores:

```
\relative c' {
  d f a g
  c b f d
}
```



Como puede observar, este ejemplo no comienza en el Do central. La primera nota (la ‘d’) es el Re más cercano al Do central.

Para poner intervalos mayores de una cuarta, podemos elevar la octava añadiendo una comilla simple ‘ (o apóstrofe) al nombre de la nota. También podemos bajar la octava adjuntando una coma , al nombre de la nota.

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



Para subir o bajar una nota en dos (¡o más!) octavas, utilizamos varias ‘ ‘ ó ,, (pero tenga cuidado de utilizar dos comillas simples ‘ ‘ ¡y no una comilla doble " !). El valor inicial de `\relative c'` también puede modificarse de esta forma.

Duraciones (valores rítmicos)

La **duración** de una nota se especifica mediante un número después del nombre de la nota. ‘1’ significa **redonda**, ‘2’ significa **blanca**, ‘4’ significa **negra** y así sucesivamente. Las barras de corchea se añaden automáticamente.

```
\relative c'' {
  a1
  a2 a4 a8 a
  a16 a a a a32 a a a a64 a a a a a a a2
}
```



Si no especifica una duración, se utiliza la duración previa para la nota siguiente. La figura por omisión de la primera nota es una negra.

Para crear figuras con puntillo, añade un punto ‘.’ al número de la duración.

```
\relative c'' {
  a a a4. a8
  a8. a16 a a8. a8 a4.
}
```



Silencios

Un **silencio** se introduce igual que una nota, con el nombre ‘r’:

```
\relative c'' {
  a r r2
  r8 a r4 r4. r8
}
```



Indicación de compás

La indicación de compás) se puede establecer con el comando `\time` :

```
\relative c'' {
  \time 3/4
  a4 a a
  \time 6/8
```

```

a4. a
\time 4/4
a4 a a a
}

```



Clave

La clave se puede establecer utilizando el comando `\clef` :

```

\relative c' {
  \clef treble
  c1
  \clef alto
  c1
  \clef tenor
  c1
  \clef bass
  c1
}

```



Todo junto

He aquí un pequeño ejemplo que muestra todos los elementos anteriores juntos:

```

\relative c, {
  \time 3/4
  \clef bass
  c2 e8 c' g'2.
  f4 e d c4 c, r4
}

```



Más información

Introducir alturas y duraciones

véase [Sección 6.1 \[Alturas\]](#), página 61 y [Sección 6.2.1 \[Duraciones\]](#), página 69.

Silencios véase [Sección 6.1.9 \[Silencios\]](#), página 67.

Indicaciones de compás y otros comandos de tiempo

véase [Sección 6.4.3 \[Indicación de compás\]](#), página 83.

Claves

véase [Sección 6.4.1 \[Clave\]](#), página 80.

2.1.3 Trabajar sobre archivos de texto

Los archivos de entrada de LilyPond son como los archivos fuente de muchos lenguajes de programación corrientes. Son sensibles a las mayúsculas e insensibles al número de espacios. Las expresiones se forman con llaves `{ }` y los comentarios se denotan por un signo de porcentaje (%) o por `%{ ... %}`.

Si la frase anterior no tiene sentido para usted ¡no se preocupe! A continuación explicaremos el significado de todos estos términos:

- **Sensible a las mayúsculas:** tiene importancia el hecho de que introduzca una letra en minúsculas (p.ej. `a`, `b`, `s`, `t`) o en mayúsculas (p.ej. `A`, `B`, `S`, `T`). Las notas son minúsculas: `{ c d e }` es una entrada válida; `{ C D E }` produciría un mensaje de error.
- **Insensible al número de espacios:** no importa cuántos espacios (o saltos de línea) añada. `{ c d e }` significa lo mismo que `{ c d e }` y que

```

      {
c                d
e }

```

Por supuesto, el ejemplo anterior es difícil de leer. Una regla práctica es sangrar los bloques de código con un carácter de tabulación, o bien con dos espacios:

```

{
  c d e
}

```

- **Expresiones:** Todo fragmento de código de entrada para LilyPond ha de llevar `{ llaves }` antes y después de la entrada. Estas llaves le dicen a LilyPond que la entrada es una expresión musical unitaria, igual que los paréntesis ‘()’ de las matemáticas. Las llaves deben ir rodeadas de un espacio a no ser que se encuentren al comienzo o al final de una línea, para evitar cualquier ambigüedad.

Una función (como por ejemplo `\relative { }`) también es una expresión musical unitaria.

- **Comentarios:** Un comentario es una nota para el lector humano de la entrada musical; se ignora cuando esta entrada se analiza, de manera que no tiene ningún efecto sobre la salida impresa. Existen dos tipos de comentarios. El símbolo de porcentaje ‘%’ introduce un comentario de línea; todo lo que se encuentra después de ‘%’ en esa línea se ignora. Un comentario de bloque marca una sección entera de entrada musical como comentario. Todo lo que está encerrado dentro de `%{ y %}` se ignora (pero los comentarios no pueden incluir otros comentarios). El siguiente fragmento muestra algunos posibles usos para los comentarios:

```

% a continuación van las notas de campanitas del lugar
c4 c g' g a a g2

%{
  Esta línea y las notas que aparecen más abajo
  se ignoran, por estar dentro de un
  comentario de bloque.

  g g f f e e d d c2
%}

```

Hay más trucos para elaborar archivos de entrada en [Sección 4.1 \[Sugerencias para escribir archivos de LilyPond\]](#), página 43.

2.1.4 Cómo leer el tutorial

Como vimos en [Sección 2.1.3 \[Trabajar sobre archivos de texto\]](#), página 17, la entrada de LilyPond debe estar rodeada de llaves `{ }` o de `\relative c'' { ... }`. Durante el resto del presente manual, la mayor parte de los ejemplos omitirán las llaves.

Si está leyendo la documentación HTML y quiere ver el código de LilyPond exacto que se utilizó para crear el ejemplo, sencillamente haga click sobre la imagen. Si no está leyendo la versión en HTML, podría copiar y pegar la entrada que se muestra, pero **deberá** añadir `\relative c'' { }` de la siguiente manera:

```
\relative c'' {
  ... aquí va el ejemplo...
}
```

¿Por qué omitir las llaves? Casi todos los ejemplos del presente manual se pueden insertar en medio de un fragmento mayor de música. Para estos ejemplos no tiene ningún sentido añadir `\relative c'' { }` (no debería poner un `\relative` dentro de otro `\relative`), de forma que usted no podría copiar un ejemplo pequeño procedente de la documentación y pegarlo dentro de su propia pieza.

2.2 Notación en un solo pentagrama

Esta sección es una introducción a la notación corriente que se utiliza para una voz o un pentagrama.

2.2.1 Nombres de nota relativos

Como hemos visto en [Sección 2.1.2 \[Notación sencilla\]](#), página 13, LilyPond calcula la altura de cada nota de forma relativa respecto de la nota anterior³. Si no se añade ninguna marca de octava (`'` y `,`), se supone que cada altura se encuentra dentro de una cuarta desde la nota anterior.

LilyPond examina las alturas basándose en los nombres de las notas (dicho de otra forma: una cuarta aumentada *no* es lo mismo que una quinta disminuida. Si empezamos en un Do, un Fa sostenido situado a continuación se colocará por encima del Do, mientras que un Sol bemol se colocará por debajo de este mismo Do.

```
c2 fis
c2 ges
```



Más información

Octavas relativas

ver [Sección 6.1.6 \[Octavas relativas\]](#), página 64.

Comprobación de octava

ver [Sección 6.1.7 \[Comprobación de la octava\]](#), página 66.

2.2.2 Alteraciones accidentales y armaduras

³ Existe otra forma de introducir las alturas, [Sección 2.5.3 \[Nombres de nota absolutos\]](#), página 32, pero en la práctica el modo relativo es mucho más cómodo y seguro de utilizar.

Alteraciones accidentales

Una nota **sostenido** se hace añadiendo ‘is’ al nombre, y una nota **bemol** añadiendo ‘es’. Como ha podido adivinar, un **doble sostenido** o **doble bemol** se hace añadiendo ‘isis’ o ‘eses’⁴

```
cis1 ees fisis, aeses
```



Armaduras

La armadura de la tonalidad se establece mediante el comando `\key` seguido de una nota y `\major` o `\minor`.

```
\key d \major
a1
\key c \minor
a
```



Advertencia: armaduras y alturas

Para determinar si hay que imprimir una alteración accidental, LilyPond examina las notas y la armadura de la tonalidad. La armadura solamente afecta a las alteraciones *impresas*, jno a las propias notas! Ésta es una funcionalidad que frecuentemente confunde a los que se inician con el programa, por ello permítanos explicarla en detalle.

LilyPond hace una clara distinción entre el contenido musical y la presentación. La alteración (bemol, becuadro o sostenido) de una nota es parte de la altura, y por tanto es contenido musical. Si una alteración (un signo *impreso* de bemol, becuadro o sostenido) se imprime o no delante de la nota correspondiente, es una cuestión de presentación. La presentación es algo que sigue unas reglas, así que las alteraciones accidentales se imprimen automáticamente según dichas reglas. Las alturas de las notas en su música son obras de arte, por tanto no se añadirán automáticamente, y usted deberá introducir aquello que quiera oír.

En el siguiente ejemplo

```
\key d \major
d cis fis
```

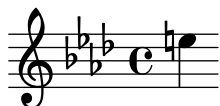


⁴ Esta sintaxis derivó de las convenciones de las lenguas nórdicas y germánicas para nombrar las notas como el alemán y el holandés. Para utilizar otros nombres para las alteraciones accidentales, vea [Sección 6.1.5 \[Nombres de las notas en otros idiomas\]](#), página 64.

ninguna nota lleva una alteración impresa, pero de todas formas usted debe añadir la ‘is’ a **cis** y a **fis**.

El texto ‘e’ no significa “imprimir una bolita negra en la primera línea del pentagrama.” Más bien significa: “hay una nota Mi natural.” En la tonalidad de La bemol mayor, *lleva* una alteración accidental:

```
\key aes \major
e
```



Poner todas las alteraciones de forma explícita puede que requiera algo más de trabajo al teclear, pero la ventaja es que la transposición es más fácil, y las alteraciones se pueden imprimir siguiendo varias convenciones distintas. Consulte [Sección 9.1.1 \[Alteraciones accidentales automáticas\]](#), [página 131](#) para ver ejemplos de cómo se pueden imprimir las alteraciones de acuerdo a reglas diferentes.

Más información

Alteraciones accidentales

ver [Sección 6.1.2 \[Alteraciones accidentales\]](#), [página 62](#) y [Sección 9.1.1 \[Alteraciones accidentales automáticas\]](#), [página 131](#).

Armadura de la tonalidad

ver [Sección 6.4.2 \[Armadura de la tonalidad\]](#), [página 82](#).

2.2.3 Ligaduras de unión y de expresión

Ligaduras de unión

Una ligadura de unión se crea adjuntando un carácter tilde ‘~’ a la primera nota ligada

```
g4~ g c2~
c4 ~ c8 a8 ~ a2
```



Ligaduras de expresión

Una ligadura de expresión es una curva que se traza abarcando varias notas. Las notas inicial y final se marcan mediante ‘(’ y ‘)’ respectivamente.

```
d4( c16) cis( d e c cis d) e( d4)
```



Ligaduras de fraseo

Las ligaduras que se utilizan para indicar fraseos más largos se pueden introducir mediante \ (y \). Puede haber al mismo tiempo ligaduras de legato y ligaduras de fraseo, pero no es posible tener legatos simultáneos o ligaduras de expresión simultáneas.

```
a8(\( ais b c) cis2 b'2 a4 cis,\)
```



Advertencias: ligaduras de expresión frente a ligaduras de unión

Una ligadura de expresión parece una **ligadura de unión**, pero tiene un significado distinto. Una ligadura (de unión) sencillamente hace que la primera nota sea más larga, y sólo se puede utilizar sobre parejas de notas iguales. Las ligaduras de expresión indican las articulaciones de las notas, y se pueden utilizar sobre grupos mayores de notas. Las ligaduras de unión y de expresión se pueden anidar unas dentro de otras.

```
c2~( c8 fis fis4 ~ fis2 g2)
```



Más información

Ligaduras de unión

ver [Sección 6.5.1 \[Ligaduras de unión\]](#), página 91.

Ligaduras de expresión

ver [Sección 6.5.2 \[Ligaduras de expresión\]](#), página 93.

Ligaduras de fraseo

ver [Sección 6.5.3 \[Ligaduras de fraseo\]](#), página 94.

2.2.4 Articulaciones y matices dinámicos

Articulaciones

Las **articulaciones** más corrientes se pueden añadir a las notas utilizando un guión ‘-’ seguido de un carácter único:

```
c-. c-- c-> c-^ c-+ c-_
```



Digitaciones

De manera similar, las indicaciones de digitación se pueden añadir a una nota utilizando un guión (‘-’) seguido del dígito deseado:

```
c-3 e-5 b-2 a-1
```



Las articulaciones y digitaciones normalmente se colocan de forma automática, pero puede especificar una dirección mediante ‘^’ (encima) o ‘_’ (debajo). También puede usar varias articulaciones sobre la misma nota. Sin embargo, casi siempre es mejor dejar que LilyPond determine la dirección de las articulaciones.

```
c_~1 d^ . f^4_2-> e^_+ 
```



Matrices dinámicas

Las expresiones de matiz o signos dinámicos se hacen añadiendo las marcas (con una barra invertida) a la nota:

```
c\ff c\mf c\p c\pp
```



Los crescendi y decrescendi comienzan con los comandos \< y \>. Una expresión terminal de matiz, como por ejemplo \f, dará por terminado el (de)crescendo, o bien se puede usar el comando \!:

```
c2\< c2\ff\> c2 c2\!
```



Más información

Articulaciones

ver [Sección 6.6.1 \[Articulaciones\]](#), página 100.

Digitaciones

ver [Sección 6.6.2 \[Indicaciones de digitación\]](#), página 102.

Matrices

ver [Sección 6.6.3 \[Matrices dinámicos\]](#), página 103.

2.2.5 Barras automáticas y manuales

Todas las barras se dibujan automáticamente:

```
a8 ais d ees r d c16 b a8
```



Si no le gustan las barras automáticas, pueden forzarse manualmente. Marque la primera nota que comprende la barra con '[' y la última con ']'.
`a8[ais] d[ees r d] a b`



Más información

Barras automáticas

ver [Sección 6.5.5 \[Barras automáticas\]](#), página 95.

Barras manuales

ver [Sección 6.5.6 \[Barras manuales\]](#), página 95.

2.2.6 Comandos rítmicos avanzados

Compás parcial

Una anacrusa (o *anacrusa*) se introduce con la palabra clave `\partial`. Va seguida de una duración: `\partial 4` es una anacrusa de negra y `\partial 8` de corchea.

`\partial 8`
`f8 c2 d`



Grupos especiales

Los grupos especiales como los tresillos se hacen con la palabra clave `\times`. Requiere dos argumentos: una fracción y un fragmento de música. La duración del fragmento de música se multiplica por la fracción. Los tresillos hacen que las notas ocupen 2/3 de su duración expresa, por tanto un tresillo lleva una fracción de 2/3

`\times 2/3 { f8 g a }`
`\times 2/3 { c r c }`
`\times 2/3 { f,8 g16[a g a] }`
`\times 2/3 { d4 a8 }`



Notas de adorno

Las notas de adorno se crean con el comando `\grace`, aunque también se pueden conseguir precediendo una expresión musical por la palabra clave `\appoggiatura` o `\acciaccatura`

`c2 \grace { a32[b] } c2`
`c2 \appoggiatura b16 c2`
`c2 \acciaccatura b16 c2`



Más información

Notas de adorno

ver [Sección 6.5.7 \[Notas de adorno\]](#), página 97,

Grupos especiales

ver [Sección 6.2.3 \[Grupos especiales\]](#), página 70,

Anacrusas ver [Sección 6.4.4 \[Compases parciales\]](#), página 84.

2.3 Varias notas a la vez

Esta sección es una introducción a las notas simultáneas: varios instrumentos, varios pentagramas para un solo instrumento (p.ej. piano) y acordes.

La palabra ‘polifonía’ en música hace referencia al hecho de tener más de una voz en un momento determinado dentro de una pieza musical. La palabra ‘polifonía’ en LilyPond se refiere al hecho de tener más de una voz en el mismo pentagrama.

2.3.1 Explicación de las expresiones musicales

En los archivos de entrada de LilyPond, la música se representa mediante *expresiones musicales*. Una sola nota es una expresión musical, aunque no es una entrada válida por sí sola.

a4



Al encerrar un grupo de notas dentro de llaves creamos una expresión musical:

{ a4 g4 }



Si colocamos un grupo de expresiones musicales (p.ej.: notas) dentro de llaves, eso significa que se encuentran en secuencia (es decir, cada una sigue a la anterior). El resultado es otra expresión musical:

{ { a4 g } f g }



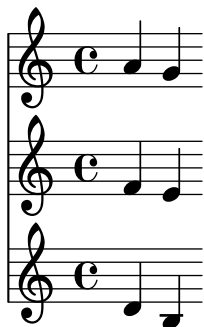
Expresiones musicales simultáneas: varios pentagramas

Esta técnica es muy útil para la música polifónica. Para introducir música con más voces o con más pentagramas, lo que hacemos es combinar varias expresiones en paralelo. Para indicar que dos voces se deben interpretar al mismo tiempo, sencillamente introduzca una combinación simultánea de expresiones musicales. Una expresión musical ‘simultánea’ se forma encerrando las expresiones dentro de << y >>. En el ejemplo que sigue, tres secuencias (cada una de las cuales contiene dos notas diferentes) se combinan de forma simultánea:

```

\relative c'' {
  <<
    { a4 g }
    { f e }
    { d b }
  >>
}

```



Tenga en cuenta que hemos sangrado cada nivel jerárquico de la entrada con un margen distinto. A LilyPond no le importa cuánto (o cuán poco) espacio haya al comienzo de una línea, pero el establecimiento de márgenes distintos dentro del código de LilyPond, de esta forma, lo hace mucho más fácil de leer por nosotros los seres humanos.

Advertencia: cada nota se entiende relativa a la nota anterior de la entrada, no relativa a la `c''` dentro del comando inicial `\relative`.

Expresiones musicales simultáneas: un solo pentagrama

Para determinar el número de pentagramas en una pieza, LilyPond examina la primera expresión. Si ésta consiste en una sola nota, hay un solo pentagrama; si hay una expresión simultánea, hay más de un pentagrama.

```

\relative c'' {
  c2 <<c e>>
  << { e f } { c <<b d>> } >>
}

```



Analogía: expresiones matemáticas

Este mecanismo es semejante a las fórmulas matemáticas: una fórmula grande se construye combinando fórmulas pequeñas. Dichas fórmulas se llaman expresiones, y su definición es recursiva de tal forma que se pueden construir expresiones de un tamaño y complejidad arbitrarios. Por ejemplo:

1

1 + 2

(1 + 2) * 3

((1 + 2) * 3) / (4 * 5)

Ésta es una secuencia de expresiones donde cada expresión se encuentra contenida dentro de la siguiente, más grande. Las expresiones más simples son números, y las mayores se hacen combinando expresiones mediante operadores (como '+', '*' y '/') y paréntesis. Del mismo modo que las expresiones matemáticas, las expresiones musicales se pueden anidar a una profundidad arbitraria, lo que se hace necesario para músicas complejas como las partituras polifónicas.

2.3.2 Varios pentagramas

Como ya hemos visto en [Sección 2.3.1 \[Explicación de las expresiones musicales\]](#), página 24, los archivos de entrada para LilyPond se construyen a base de expresiones musicales. Si la partitura comienza con expresiones musicales simultáneas, LilyPond crea varios pentagramas. Sin embargo es más fácil ver lo que ocurre si creamos cada uno de los pentagramas de forma explícita.

Para imprimir más de un pentagrama, cada fragmento de música que constituye un pentagrama se marca escribiendo `\new Staff` antes de él. Estos elementos `Staff` se combinan después en paralelo con `<<` y `>>`:

```
\relative c'' {
  <<
    \new Staff { \clef treble c }
    \new Staff { \clef bass c,, }
  >>
}
```



El comando `\new` inaugura un 'contexto de notación'. Un contexto de notación es un entorno dentro del que se interpretan los acontecimientos musicales (como las notas o los comandos `\clef`). Para piezas sencillas, tales contextos de notación se crean automáticamente. Para piezas más complicadas, es mejor marcar los contextos de forma explícita.

Existen varias clases de contextos. `Score`, `Staff` y `Voice` manejan la notación melódica, mientras que `Lyrics` se ocupa de los textos cantados y `ChordNames` imprime los nombres de los acordes.

En términos de sintaxis, la anteposición de `\new` a una expresión musical crea una expresión musical mayor. Es semejante al signo menos de las matemáticas. La fórmula $(4 + 5)$ es una expresión, por tanto $-(4 + 5)$ es una expresión más amplia.

Las indicaciones de compás escritas en un pentagrama afecta al resto, pero la armadura de la tonalidad de un pentagrama *no* afecta a los otros pentagramas⁵.

```
\relative c'' {
  <<
    \new Staff { \clef treble \time 3/4 c }
    \new Staff { \clef bass \key d \major c,, }
  >>
}
```

⁵ Este comportamiento se puede cambiar si uno lo desea; consulte [Capítulo 9 \[Cambiar los valores por omisión\]](#), página 131 para ver más detalles.



2.3.3 Sistemas de piano

La música para piano se compone tipográficamente en forma de dos pentagramas unidos mediante una llave. El aspecto impreso de este sistema de pentagramas se parece al ejemplo polifónico que aparece en [Sección 2.3.2 \[Varios pentagramas\]](#), página 26, pero en esta ocasión la expresión completa se coloca dentro de un `PianoStaff`:

```
\new PianoStaff <<
  \new Staff ...
  \new Staff ...
>>
```

He aquí un pequeño ejemplo:

```
\relative c'' {
  \new PianoStaff <<
    \new Staff { \time 2/4 c4 e g g, }
    \new Staff { \clef bass c,, c' e c }
  >>
}
```



Más información

Ver [Sección 7.1 \[Música de piano\]](#), página 115.

2.3.4 Polifonía en un solo pentagrama

Cuando distintas líneas melódicas se combinan sobre un solo pentagrama, se imprimen como voces polifónicas; cada voz lleva sus propias plicas, ligaduras y barras de corchea, y la voz superior tiene las plicas hacia arriba mientras que la voz inferior las tiene hacia abajo.

La introducción de estas partes se hace escribiendo cada voz en forma de secuencia (con `{...}`) y combinando éstas de forma simultánea, separando las voces con `\\`

```
<<
  { a4 g2 f4~ f4 } \\
  { r4 g4 f2 f4 }
>>
```



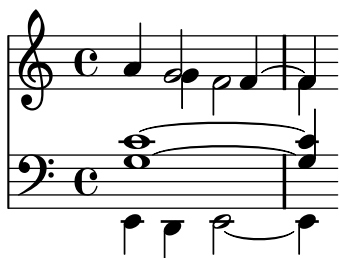
Para el tipografiado de música polifónica, puede ser conveniente la utilización de silencios separadores, o sea, silencios que no aparecen impresos. Son muy útiles para rellenar voces que temporalmente no están cantando. He aquí el mismo ejemplo con un silencio separador (`'r'`) en vez de un silencio normal (`'r'`):

```
<<
  { a4 g2 f4~ f4 } \\
  { s4 g4 f2 f4 }
>>
```



Una vez más, las expresiones de este tipo se pueden anidar de forma arbitraria.

```
<<
  \new Staff <<
    { a4 g2 f4~ f4 } \\
    { s4 g4 f2 f4 }
  >>
  \new Staff <<
    \clef bass
    { <c g>1 ~ <c g>4 } \\
    { e,,4 d e2 ~ e4 }
  >>
>>
```



Más información

Ver [Sección 6.3.3 \[Polifonía básica\]](#), página 75.

2.3.5 Combinar notas para formar acordes

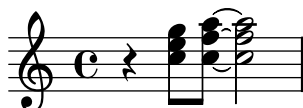
Los acordes se hacen rodeando las notas con paréntesis en ángulo simples. Los paréntesis en ángulo son los símbolos ‘<’ (menor que) y ‘>’ (mayor que).

```
r4 <c e g>4 <c f a>2
```

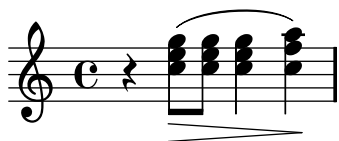


Se pueden combinar las marcas como barras y ligaduras con acordes. Se deben colocar por fuera de los paréntesis en ángulo:

```
r4 <c e g>8[ <c f a>]~ <c f a>2
```



```
r4 <c e g>8\>( <c e g> <c e g>4 <c f a>\!)
```



2.4 Canciones

Esta sección es una introducción a la música vocal y a las partituras de canciones sencillas.

2.4.1 Imprimir la letra

Consideremos una melodía sencilla:

```
\relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
```



La letra se puede asignar a esas notas, combinando ambas con la palabra clave `\addlyrics`. La letra se escribe separando cada sílaba mediante un espacio.

```
<<
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free }
>>
```



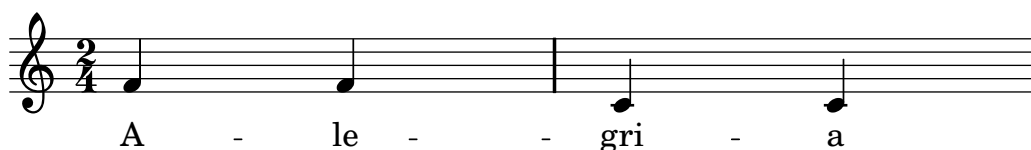
Esta melodía acaba en un *melisma*, una sílaba única ('suelta') que se canta sobre más de una nota. Esto se indica mediante una *línea de extensión*. Se escribe como dos guiones bajos `--`:

```
<<
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free -- }
>>
```



De forma parecida, los guiones entre las palabras se pueden escribir como dos guiones, dando como resultado unas líneas cortas entre cada dos sílabas:

```
<<
  \relative c' {
    \time 2/4
    f4 f c c
  }
  \addlyrics { A -- le -- gri -- a }
>>
```



Más información

Otras posibilidades como poner varios versos debajo de una melodía, se discuten en [Sección 7.3 \[Música vocal\]](#), página 115.

2.4.2 Hojas guía de acordes

En la música pop es muy común denotar el acompañamiento con los nombres de los acordes. Estos acordes se pueden escribir como notas:

```
\chordmode { c2 f4. g8 }
```



Ahora, cada altura se lee como la fundamental de un acorde en vez de como una nota. Este modo se activa con `\chordmode`. Otros acordes pueden construirse añadiendo modificadores después de un signo de dos puntos. El ejemplo que sigue muestra algunos modificadores comunes:

```
\chordmode { c2 f4:m g4:maj7 gis1:dim7 }
```



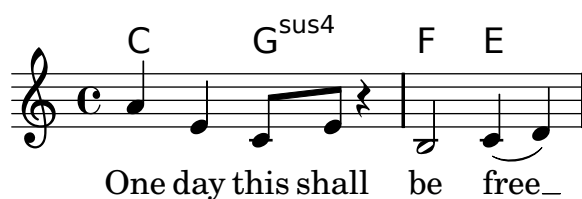
Para canciones con acordes, éstos no se imprimen sobre pentagramas, sino como nombres en su propia línea. Esto se consigue utilizando `\chords` en sustitución de `\chordmode`. Utiliza la misma sintaxis que `\chordmode`, pero pinta las notas dentro de un contexto de `ChordNames`, con el siguiente resultado:

```
\chords { c2 f4.:m g4.:maj7 gis8:dim7 }
```

C FmG[△] G[♯]07

Cuando se combinan, los nombres de acorde, la letra y la melodía forman una hoja de canción con acordes ('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 __ }
>>
```



Más información

Puede ver una lista completa de los modificadores y otras opciones de presentación en [Sección 6.3.1 \[Acordes\]](#), página 74.

2.5 Retoques finales

Éste es el último apartado del tutorial; muestra la forma de dar los toques finales a piezas sencillas, y ofrece una introducción al resto del manual.

2.5.1 Número de la versión

La indicación `\version` indica para qué versión de LilyPond se escribió el archivo. Para marcar un archivo para la versión 2.11.15, escriba

```
\version "2.11.15"
```

al principio de su archivo de LilyPond.

Estas anotaciones hacen menos problemáticas las subsiguientes actualizaciones de LilyPond. Los cambios en la sintaxis se tratan mediante un programa especial, 'convert-ly' (ver program usage manual, *Updating files with convert-ly*), y utiliza `\version` para determinar qué reglas hay que aplicar.

2.5.2 Añadir títulos

La información sobre el título, autor, número de Opus y similares se escriben en el bloque `\header`. Éste se encuentra fuera de la expresión musical principal; el bloque `\header` normalmente se sitúa por debajo del [Sección 2.5.1 \[Número de la versión\]](#), página 31.

```
\version "2.11.15"
\header {
  title = "Symphony"
  composer = "Me"
  opus = "Op. 9"
}

{
  ... music ...
}
```

Cuando se procesa el archivo, el título y el autor se imprimen por encima de la música. Puede obtener más información sobre los títulos en [Sección 10.2.1 \[Crear títulos\]](#), página 133.

2.5.3 Nombres de nota absolutos

Hasta el momento siempre hemos utilizado `\relative` para definir las alturas. Ésta es la forma más sencilla de escribir la mayor parte de la música, pero existe otra forma de definir las alturas: el modo absoluto.

Si omite el `\relative`, LilyPond tratará todas las alturas como valores absolutos. Una `c'` significará siembre un Do central, una `b` significará siempre la nota inmediatamente por debajo del Do central, y una `g`, significará siempre la nota que se coloca en la primera línea del pentagrama en clave de Fa.

```
{
  \clef bass
  c' b g, g,
  g, f, f c'
}
```



He aquí una escala que abarca cuatro octavas:

```
{
  \clef bass
  c, d, e, f,
  g, a, b, c
  d e f g
  a b c' d'
  \clef treble
  e' f' g' a'
  b' c'' d'' e''
  f'' g'' a'' b''
  c'''1
}
```



Como puede ver, escribir una melodía en clave de Sol implica escribir gran cantidad de apóstrofes '. Consideremos este fragmento de Mozart:

```
{
  \key a \major
  \time 6/8
  cis''8. d''16 cis''8 e''4 e''8
```

```
b'8. cis''16 b'8 d''4 d''8
}
```



Todos estos apóstrofes hacen casi ilegible el código de entrada y será origen de numerosos errores. Con `\relative`, el ejemplo anterior es mucho más fácil de leer:

```
\relative c'' {
  \key a \major
  \time 6/8
  cis8. d16 cis8 e4 e8
  b8. cis16 b8 d4 d8
}
```



Si comete un error con una marca de octava (' o ,) mientras trabaja en el modo `\relative`, será muy obvio (muchas notas estarán en la octava equivocada). Mientras trabaja en el modo absoluto, un solo fallo no será tan visible, y tampoco será tan fácil de localizar.

Sin embargo, el modo absoluto es útil para escribir música que contenga intervalos grandes, y será extremadamente útil para hacer archivos de LilyPond generados por ordenador.

2.5.4 Organizar las piezas mediante identificadores

Cuando los elementos que hemos discutido anteriormente se combinan para producir archivos mayores, las expresiones musicales se hacen enormes. En música polifónica con muchos pentagramas, los archivos de entrada pueden volverse muy propensos a la confusión. Podemos reducir esta confusión utilizando los *identificadores*.

Con los identificadores (también conocidos como variables o macros), podemos trocear las expresiones musicales complejas. Un identificador se asigna de la manera siguiente:

```
musicaConNombre = { ... }
```

El contenido de la expresión musical `musicaConNombre` se puede usar posteriormente colocando una barra invertida delante del nombre (`\musicaConNombre`, igual que un comando normal de LilyPond). Los identificadores se deben definir *antes* de la expresión musical principal.

```
violin = \new Staff { \relative c'' {
  a4 b c b
}}
cello = \new Staff { \relative c {
  \clef bass
  e2 d
}}
{
  <<
    \violin
    \cello
  >>
}
```

}



El nombre de un identificador debe consistir enteramente en caracteres alfabéticos, es decir sin números, guiones ni guiones bajos.

Es posible utilizar variables para otras muchas clases de objetos en el código de entrada. Por ejemplo:

```
width = 4.5\cm
name = "Wendy"
aFivePaper = \paper { paperheight = 21.0 \cm }
```

Dependiendo de su contenido, el identificador se puede usar en distintos lugares. El siguiente ejemplo utiliza las variables anteriores:

```
\paper {
  \aFivePaper
  line-width = \width
}
{ c4^\name }
```

2.5.5 Más allá del tutorial

Después de terminar el tutorial, quizá debería probar a escribir una o dos piezas. Comience con una de las [Apéndice D \[Plantillas\]](#), [página 145](#) y añada algunas notas. Si necesita un tipo de notación que no ha sido tratada en el tutorial, eche un vistazo a la Referencia de Notación, empezando por [Capítulo 6 \[Notación básica\]](#), [página 61](#). Si quiere escribir música para un conjunto instrumental que no está cubierto por ninguna plantilla, consulte [Sección 3.1 \[Extender las plantillas\]](#), [página 35](#).

Una vez que ha escrito algunas piezas cortas, lea el resto del Manual de aprendizaje (capítulos 3 al 5). ¡Por supuesto, no pasa nada por leerlo ahora mismo! Sin embargo, el resto del Manual de Aprendizaje da por sentado que está familiarizado con la entrada de LilyPond. Puede saltarse estos capítulos ahora y volver a ellos cuando haya adquirido más experiencia.

2.5.6 Cómo leer el manual

Como ya vimos en [Sección 2.1.4 \[Cómo leer el tutorial\]](#), [página 18](#), muchos ejemplos del tutorial omitían el `\relative c' { ... }` alrededor del ejemplo mostrado.

Durante el resto del manual seremos mucho menos estrictos respecto de los ejemplos impresos: a veces pueden omitir el `\relative c' { ... }`, pero otras veces se puede usar una nota inicial diferente (como `c'` o `c,,`), y en ocasiones ¡todo el ejemplo estará en el modo de notas absoluto! Sin embargo, las ambigüedades de esta clase existen solamente allí donde las alturas no son lo más importante. En aquellos ejemplos en los que las alturas tienen importancia, hemos escrito explícitamente `\relative` o bien el `{ }` en modo absoluto.

Si aún tiene dudas respecto de la entrada de LilyPond exacta que se ha utilizado en un ejemplo, lea la versión HTML (si no lo está haciendo ya) y pulse con el ratón sobre la imagen de la música. De esta forma se mostrará la entrada exacta que LilyPond usó para generar este manual.

3 Juntándolo todo

Este capítulo trata de los conceptos generales de LilyPond y de cómo elaborar bloques `\score`.

3.1 Extender las plantillas

Ha leído el tutorial y ahora sabe escribir música. Pero ¿cómo puede poner los pentagramas que quiere? Las plantillas están muy bien, pero ¿qué ocurre si quiere algo que no está en una de ellas?

Para empezar, tome la plantilla que le parezca más parecida a aquello que quiere conseguir. Digamos que quiere escribir algo para soprano y cello. En este caso comenzaríamos con la plantilla ‘Notas y letra’ (para la parte de soprano).

```
\version "2.11.15"
melodia = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

texto = \lyricmode {
  Aaa Bee Cee Dee
}

\score{
  <<
    \new Voice = "uno" {
      \autoBeamOff
      \melodia
    }
    \new Lyrics \lyricsto "uno" \text
  >>
  \layout { }
  \midi { }
}
```

Ahora queremos añadir una parte de violoncello. Veamos el ejemplo ‘Sólo notas’:

```
\version "2.11.15"
melodia = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
  \new Staff \melody
  \layout { }
  \midi { }
}
```

No necesitamos dos comandos `\version`. Vamos a necesitar la sección `melodia`. No queremos dos secciones `\score` (si tuviésemos dos `\scores`, acabaríamos con las dos particellas por separado. Queremos las dos juntas, como un dúo. Dentro de la sección `\score`, no nos hacen falta dos `\layout` ni dos `\midi`.

Si nos limitásemos a copiar y pegar la sección `melodia`, acabaríamos con dos secciones `melodia` separadas, así que vamos a cambiarles el nombre. Llamaremos `musicaSoprano` a la sección de la soprano y `musicaCello` a la sección del violoncello. Al mismo tiempo cambiaremos el nombre de `texto` a `letraSoprano`. Recuerde cambiar el nombre a las dos apariciones de todos estos nombres – tanto la definición inicial (la parte `melodia = relative c' { }`) – como el uso de ese nombre (en la sección `\score`).

También aprovecharemos para cambiar el pentagrama de la parte del cello (los violoncellos se escriben normalmente en clave de Fa). Asimismo, cambiaremos algunas notas del cello.

```
\version "2.11.15"
musicaSoprano = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

letraSoprano = \lyricmode {
  Aaa Bee Cee Dee
}

musicaCello = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  d4 g fis8 e d4
}

\score{
  <<
    \new Voice = "uno" {
      \autoBeamOff
      \sopranoMusic
    }
    \new Lyrics \lyricsto "uno" \letraSoprano
  >>
  \layout { }
  \midi { }
}
```

Esto tiene una pinta prometedora, pero la parte del cello no sale en la partitura (no la hemos puesto en la sección `\score`). Si queremos que la parte del cello aparezca debajo de la de soprano, tenemos que añadir

```
\new Staff \musicaCello
```

justo debajo de todo lo de la soprano. También tenemos que poner `<< y >>` antes y después de la música – lo que indica a LilyPond que hay más de una cosa (en este caso, `Staff`) sucediendo al mismo tiempo –. La `\score` se parecerá ahora a esto

```

\score{
  <<
    <<
      \new Voice = "uno" {
        \autoBeamOff
        \sopranoMusic
      }
      \new Lyrics \lyricsto "uno" \letraSoprano
    >>
    \new Staff \musicaCello
  >>
  \layout { }
  \midi { }
}

```

Esto parece un poco enrevesado; los márgenes están descuadrados. Esto tiene fácil solución. Presentamos aquí la plantilla completa para soprano y cello.

```

\version "2.11.15"
sopranoMusic = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

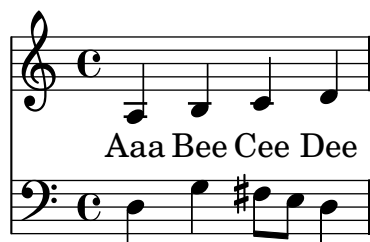
sopranoLyrics = \lyricmode {
  Aaa Bee Cee Dee
}

celloMusic = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  d4 g fis8 e d4
}

\score{
  <<
    <<
      \new Voice = "one" {
        \autoBeamOff
        \sopranoMusic
      }
      \new Lyrics \lyricsto "one" \sopranoLyrics
    >>
    \new Staff \celloMusic
  >>
  \layout { }
  \midi { }
}

```



3.2 Cómo funcionan los archivos de LilyPond

El formato de entrada de LilyPond es bastante libre en su forma y concede a los usuarios con experiencia mucha flexibilidad para estructurar sus archivos de la forma que deseen. Sin embargo, toda esta flexibilidad puede hacer que las cosas se vuelvan confusas para los nuevos usuarios. Esta sección le va a explicar parte de esta estructura, pero puede obviar ciertos detalles en aras de la simplicidad. Para ver una descripción completa del formato de entrada, consulte [Sección 10.1.2 \[Estructura del archivo\], página 133](#).

Casi todos los ejemplos del presente manual con pequeños fragmentos de código. Por ejemplo

```
c4 a b c
```

Como (esperamos) ya se habrá dado cuenta, esto no se puede compilar tal cual está. Estos ejemplos son sólo resúmenes de los ejemplos completos. Todos ellos necesitan, cuando menos, un par de llaves antes y después para que se puedan compilar

```
{
  c4 a b c
}
```

Casi todos los ejemplos también usan el comando `\relative c'` (o `c''`). Esto no es necesario para conseguir que los ejemplos simplemente se puedan compilar, pero casi siempre la salida tendrá un aspecto muy extraño si omite el `\relative c'`.

```
\relative c'' {
  c4 a b c
}
```



Ahora nos encontramos con el único tropiezo de verdad: la entrada de LilyPond en esta forma es realmente *otra* abreviatura. Aunque se puede compilar y presenta la salida correcta, es una abreviatura de

```
\score {
  \relative c'' {
    c4 a b c
  }
}
```

Un `\score` debe comenzar con una única expresión musical. Recuerde que una expresión musical podía ser cualquier cosa desde una sola nota hasta todo un

```
{
  \new GrandStaff <<
    introduzca aquí la partitura completa de una ópera de Wagner
  >>
}
```

Ya que todo está dentro de `{ ... }`, cuenta como una sola expresión musical.

La `\score` puede contener otras cosas como

```

\score {
  { c'4 a b c' }
  \layout { }
  \midi { }
  \header { }
}

```

Hay personas que ponen algunos de estos comandos fuera del bloque `\score` (por ejemplo, `\header` se sitúa con frecuencia por encima del `\score`. Es tan sólo otra abreviatura que LilyPond acepta.

Otro atajo genial es la posibilidad de definir variables. Todas las plantillas emplean lo siguiente:

```

melodia = \relative c' {
  c4 a b c
}

\score {
  { \melodia }
}

```

Cuando LilyPond examina este archivo, toma el valor de `melodia` (todo lo que está después del signo igual) y lo inserta dondequiera que ve `\melodia`. No se requiere un cuidado especial con los nombres – puede ser `melodia`, `global`, `manoderechadelpiano`, o `fulanomengano` –. Puede usar el nombre de variable que desee. Para ver más detalles, consulte [Sección 4.2 \[Ahorrar tecleo mediante identificadores y funciones\]](#), página 44.

Para ver una definición completa del formato de entrada, consulte [Sección 10.1.2 \[Estructura del archivo\]](#), página 133.

3.3 La partitura es una sola expresión musical

En la sección anterior, [Sección 3.2 \[Cómo funcionan los archivos de LilyPond\]](#), página 38, pudimos ver la organización general de los archivos de entrada de LilyPond. Pero parece que nos saltamos la parte más importante: ¿cómo averiguamos qué escribir después de `\score`?

No nos hemos saltado nada en absoluto. El gran misterio es, sencillamente, que no hay *ningún* misterio. La siguiente línea lo explica todo:

Una \score debe comenzar con una única expresión musical.

Quizá encuentre útil dar un repaso a [Sección 2.3.1 \[Explicación de las expresiones musicales\]](#), página 24. En esta sección, vimos cómo elaborar grandes expresiones musicales a partir de pequeñas piezas – comenzábamos con notas, luego acordes, etc. –. Ahora partiremos de una gran expresión musical y recorreremos el camino inverso hacia abajo.

```

\score {
  { % esta llave da inicio a la expresión musical completa
    \new GrandStaff <<
      introduzca aquí la partitura completa de una ópera de Wagner
    >>
  } % esta llave da por terminada la expresión musical completa
  \layout { }
}

```

Una ópera de Wagner completa puede ser fácilmente el doble de larga que este manual, por tanto vamos a hacer sólo un cantante y un piano. No necesitamos un `GrandStaff` para este conjunto, así que lo retiramos. Sin embargo, sí que *necesitamos* un cantante y un piano.

```

\score {
  {
    <<
      \new Staff = "cantante" <<
      >>
      \new PianoStaff = "piano" <<
      >>
    >>
  }
  \layout { }
}

```

Recuerde que usamos << y >> para presentar música simultánea. Y desde luego ¡queremos presentar las partes vocal y del piano al mismo tiempo!

```

\score {
  {
    <<
      \new Staff = "cantante" <<
      \new Voice = "vocal" { }
      >>
      \new Lyrics \lyricsto vocal \new Lyrics { }
      \new PianoStaff = "piano" <<
      \new Staff = "superior" { }
      \new Staff = "inferior" { }
      >>
    >>
  }
  \layout { }
}

```

Ahora tenemos muchos más detalles. Tenemos la pauta del cantante: contiene una **Voice** o voz (en LilyPond, este término hace referencia a un conjunto de notas, no necesariamente notas vocales – por ejemplo, un violín generalmente toca una voz –) y el texto de la canción. También tenemos una pauta de piano: contiene un pentagrama superior (mano derecha) y un pentagrama inferior (mano izquierda).

En este momento podríamos comenzar a meter las notas. Dentro de las llaves que siguen a `\new Voice = vocal`, podríamos empezar escribiendo

```

\relative c'' {
  a4 b c d
}

```

Pero si lo hiciéramos, la sección `\score` se haría bastante larga y sería más difícil comprender lo que ocurre. En lugar de esto utilizaremos identificadores o variables.

```

melodia = { }
texto = { }
superior = { }
inferior = { }
\score {
  {
    <<
      \new Staff = "cantante" <<
      \new Voice = "vocal" { \melodia }
      >>

```

```

\new Lyrics \lyricsto vocal \new Lyrics { \texto }
\new PianoStaff = "piano" <<
  \new Staff = "superior" { \superior }
  \new Staff = "inferior" { \inferior }
>>
>>
}
\layout { }
}

```

Recuerde que puede usar casi cualquier nombre que se le antoje. Las limitaciones impuestas a los nombres de identificador se detallan en [Sección 10.1.2 \[Estructura del archivo\]](#), página 133.

Cuando escriba una sección `\score` o cuando la esté leyendo, hágalo despacio y con cuidado. Comience por la capa exterior y luego trabaje sobre cada una de las capas interiores. Esto también sirve para ser estricto con los márgenes – ¡asegúrese de que en su editor de texto cada elemento de la misma capa comienza en la misma posición horizontal! –.

3.4 Una particella orquestal

En música orquestal, todas las notas se imprimen dos veces. Una vez en las particellas para los músicos, y otra para la partitura del director. Los identificadores se pueden usar para evitar la duplicación del trabajo. La música se escribe una vez y se almacena en una variable. El contenido de dicha variable se usa después para generar tanto la particella como la partitura del director.

Es muy conveniente definir las notas en un archivo especial. Por ejemplo, supongamos que el archivo `'trompa.ly'` contiene la siguiente parte de un dúo para trompa y fagot:

```

notasTrompa = \relative c {
  \time 2/4
  r4 f8 a cis4 f e d
}

```

Luego se hace una particella escribiendo en un archivo lo siguiente

```

\include "trompa.ly"
\header {
  instrument = "Trompa en Fa"
}

{
  \transpose f c' \notasTrompa
}

```

La línea

```
\include "trompa.ly"
```

sustituye el contenido de `'trompa.ly'` en esta posición dentro del archivo, así que `notasTrompa` se define con posterioridad. El comando `\transpose f c'` indica que el argumento constituido por `\notasTrompa` se debe transponer una quinta hacia arriba. Lo que suena como `'f'` se escribe como `c'`, lo que corresponde con el tono de afinación de una trompa normal en Fa. La transposición se puede ver en la siguiente salida



En piezas para conjunto, con frecuencia una de las voces no suena durante muchos compases. Esto queda denotado por un silencio especial, el silencio multicomás. Se introduce con una ‘R’ mayúscula seguida de una duración (1 en el caso de la redonda, 2 en el caso de una blanca, etc.). Multiplicando la duración se pueden construir silencios más largos. Por ejemplo, este silencio ocupa 3 compases de 2/4

R2*3

Cuando se imprime la particella tienen que comprimirse los silencios multicomás. Esto se hace estableciendo una variable en tiempo de ejecución

```
\set Score.skipBars = ##t
```

Este comando establece el valor de la propiedad `skipBars` en el contexto de `Score` a verdadero (`##t`). Anteponiendo el silencio y esta opción a la música anterior, llegamos al siguiente resultado

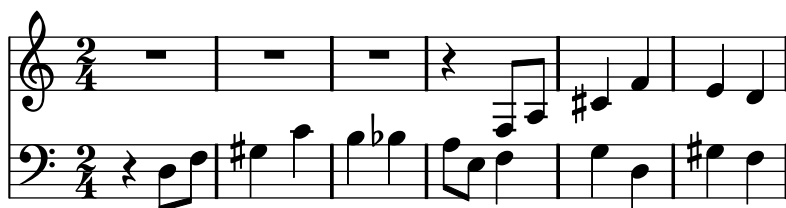


Esta partitura se hace combinando toda la música junta. Suponiendo que la otra voz se encuentra dentro de `notasFagot` en el archivo ‘`fagot.ly`’, la partitura se hace con

```
\include "fagot.ly"
\include "trompa.ly"

<<
  \new Staff \notasTrompa
  \new Staff \notasFagot
>>
```

lo que nos lleva a



Se puede encontrar una información más profunda sobre cómo preparar particellas y partituras en el manual de notación; consulte [Sección 8.3 \[Música orquestal\]](#), página 128.

El establecimiento de variables en tiempo de ejecución (‘propiedades’) se trata en [Sección 9.2.3 \[Cambiar las propiedades de un contexto al vuelo\]](#), página 131.

4 Trabajar en proyectos de LilyPond

Esta sección explica cómo resolver o evitar ciertos problemas comunes. Si tiene experiencia en programación muchos de estos consejos pueden parecer obvios, pero aún así le recomendamos que lea este capítulo.

4.1 Sugerencias para escribir archivos de LilyPond

En este momento está preparado para comenzar a escribir archivos de LilyPond más grandes – no sólo los pequeños ejemplos que aparecen en el tutorial, sino piezas completas –. Pero ¿cómo debe proceder para hacerlo?

En la medida en que LilyPond entienda sus archivos y produzca la salida que usted pretendía, realmente no importa mucho qué aspecto tengan sus archivos. Sin embargo existen algunas otras cosas a tener en cuenta cuando se escriben archivos de LilyPond.

- ¿Qué ocurre si comete un fallo? La estructura de un archivo lilypond puede hacer que ciertos errores se hagan más fáciles (o más difíciles) de encontrar.
- ¿Qué ocurre si quiere compartir sus archivos con otras personas? De hecho, ¿y si quiere alterar sus propios archivos después de algunos años? Algunos archivos de lilypond se comprenden a primera vista; otros pueden tenerle rascándose la cabeza durante una hora.
- ¿Qué ocurre si quiere actualizar su archivo de lilypond para poderlo usar con una versión más reciente del programa? La sintaxis de la entrada se modifica de forma ocasional según lilypond se va perfeccionando. Casi todos los cambios se pueden hacer de forma automática con `convert-ly`, pero algunos podrían necesitar de una ayuda manual. Los archivos de LilyPond se pueden estructurar para que sean más fáciles (o más difíciles) de actualizar.

4.1.1 Sugerencias de tipo general

Presentamos algunas sugerencias que pueden serle de ayuda para evitar o corregir problemas:

- **Incluya los números de `\version` en todos los archivos.** Dése cuenta de que todas las plantillas contienen una cadena como `\version "2.11.15"`. Le recomendamos mucho que siempre incluya la `\version`, sin importar cuán pequeño pueda ser su archivo. Desde la experiencia personal podemos decirle que es bastante frustrante intentar recordar el número de versión de LilyPond que estaba usando hace unos años. `convert-ly` requiere que declare qué versión de LilyPond utilizó.
- **Incluya comprobaciones:** Sección 6.2.5 [Comprobación del compás], página 73, Sección 6.1.7 [Comprobación de la octava], página 66 y `\numberofmeasures` [Comprobación del número de compás], página `\numberofmeasures`. Si incluye comprobaciones de vez en cuando, en caso de que cometa un error podrá localizarlo mucho más rápidamente. ¿Con qué frecuencia es ‘de vez en cuando’? Depende de la complejidad de la música. Para una música muy sencilla, quizá tan sólo una o dos veces. Para una música muy compleja, quizá a cada compás.
- **Un compás por cada línea de texto.** Si hay algo muy complicado, ya sea en la propia música o en la salida que desea producir, a menudo conviene escribir un solo compás por cada línea. El ahorro en espacio de pantalla que se obtiene al amontonar ocho compases por línea no merece la pena si luego tiene que ‘depurar’ los archivos.
- **Comente los archivos.** Utilice o números de compás (de vez en cuando) o referencias a temas musicales (‘segundo tema de los violines,’ ‘cuarta variación,’ etc.). Puede que no necesite comentarios cuando introduce una pieza por vez primera, pero si quiere volver a ella o modificar algo al cabo de dos o tres años, y también si le pasa la fuente a un amigo, será todo un desafío determinar sus intenciones o de qué manera estaba estructurado el archivo si no le añadió los comentarios.
- **Aplique márgenes a las llaves.** Muchos problemas están causados por una falta de equilibrio en el número de `{` y `}`.

- **Escriba las duraciones explícitamente** al comienzo de las secciones e identificadores. Si especifica `c4 d e` al principio de una frase (en lugar de sólo `c d e`) se puede ahorrar problemas si reelabora la música más tarde.
- **Separe los trucos** de las definiciones musicales. Consulte [Sección 4.2 \[Ahorrar tecleo mediante identificadores y funciones\]](#), página 44 y [Sección 4.3 \[Hojas de estilo\]](#), página 46.

4.1.2 Tipografiar música existente

Si está introduciendo música a partir de una partitura existente (es decir, tipografiando una hoja de música ya impresa),

- Introduzca un sistema del manuscrito (la copia física) cada vez (pero mantenga la práctica de escribir un compás por línea de texto), y compruebe cada sistema cuando lo haya terminado. Puede usar el comando `showLastLength` para acelerar el proceso – ver [Sección 10.5 \[Saltar la música corregida\]](#), página 134 –.
- Defina `mBreak = { \break }` e inserte `\mBreak` dentro del archivo de entrada donde el manuscrito tenga un saldo de línea. De esta forma le resultará mucho más fácil comparar la música de LilyPond con la original. Cuando haya terminado de revisar su partitura podrá definir `mBreak = { }` para quitar todos esos saltos de línea. Así permitirá a LilyPond colocar los saltos donde éste lo estime más oportuno.

4.1.3 Proyectos grandes

Al trabajar en proyecto grande se hace esencial tener una estructura clara en los archivos de lilypond.

- **Utilice un identificador para cada voz**, con un mínimo de estructura dentro de la definición. La estructura de la sección `\score` es la que cambiará con mayor probabilidad; por contra, es extremadamente improbable que cambie la definición de `violin` en versiones nuevas de LilyPond.

```
violin = \relative c'' {
  g4 c'8. e16
}
...
\score {
  \new GrandStaff {
    \new Staff {
      \violin
    }
  }
}
```

- **Separe los trucos de las definiciones musicales.** Ya se mencionó en [Sección 4.1.1 \[Sugerencias de tipo general\]](#), página 43, pero para proyectos grandes es vital. Quizá tengamos que cambiar la definición de `fthenp`, pero en ese caso sólo lo tendremos que hacer una vez, y aún podremos evitar tocar nada dentro de `violin`.

```
fthenp = _\markup{
  \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p }
violin = \relative c'' {
  g4\fthenp c'8. e16
}
```

4.2 Ahorrar tecleo mediante identificadores y funciones

Llegado a este punto, usted ha visto cosas de este tipo:

```

hornNotes = \relative c'' { c4 b dis c }
\score {
  {
    \hornNotes
  }
}

```



Incluso se dará cuenta de que esto puede ser útil en música minimalista:

```

fragA = \relative c'' { a4 a8. b16 }
fragB = \relative c'' { a8. gis16 ees4 }
violin = \new Staff { \fragA \fragA \fragB \fragA }
\score {
  {
    \violin
  }
}

```



Sin embargo también puede usar estos identificadores (que también se conocen como variables, macros o comandos definidos por el usuario) para hacer trucos:

```

dolce = \markup{ \italic \bold dolce }
padText = { \once \override TextScript #'padding = #5.0 }
fthenp=_markup{ \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p }
violin = \relative c'' {
  \repeat volta 2 {
    c4._\dolce b8 a8 g a b |
    \padText
    c4.^"hi there!" d8 e' f g d |
    c,4.\fthenp b8 c4 c-. |
  }
}
\score {
  {
    \violin
  }
}
\layout{ragged-right=##t}
}

```



Obviamente estos identificadores son útiles para ahorrar tecleo. Pero son dignos de tener en cuenta incluso si se van a utilizar una sola vez: reducen la complejidad. Examinemos el ejemplo anterior reescrito sin ningún identificador. Encontrará que es mucho más difícil de leer, sobre todo la última línea.

```
violin = \relative c'' {
  \repeat volta 2 {
    c4._\markup{ \italic \bold dolce } b8 a8 g a b |
    \once \override TextScript #'padding = #5.0
    c4.^"hi there!" d8 e' f g d |
    c,4.\markup{ \dynamic f \italic \small { 2nd }
      \hspace #0.1 \dynamic p } b8 c4 c-. |
  }
}
```

Hasta ahora hemos contemplado la sustitución estática: cuando LilyPond se encuentra con `\padText`, lo sustituye con aquello que hemos definido que sea (es decir, todo lo que está a la derecha de `padtext=`).

LilyPond también puede manejar sustituciones no estáticas (piense en ellas como en funciones).

```
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" d e f
  \padText #2.6
  c4^"piu mosso" fis a g
}
```



La utilización de identificadores también es una buena forma de reducir el trabajo si la sintaxis de entrada de LilyPond cambia (ver [Sección 4.4 \[Actualizar archivos antiguos\]](#), página 50). Si tiene una sola definición (como p.ej. `\dolce`) para todos sus archivos (ver [Sección 4.3 \[Hojas de estilo\]](#), página 46), y después la sintaxis se modifica, sólo tendrá que actualizar su definición `\dolce` única, en lugar de tener que hacer cambios en cada uno de los archivos `.ly`.

4.3 Hojas de estilo

La salida que produce LilyPond se puede modificar profundamente; consulte [Capítulo 5 \[Trucar la salida\]](#), página 52 para leer detalles sobre este asunto. Pero ¿qué ocurre si tiene muchos archivos a los que les quiere aplicar sus propios trucos? O ¿qué ocurre si, sencillamente, quiere separar los trucos de la propia música? Todo esto es bastante fácil de conseguir.

Veamos un ejemplo. No se preocupe si no entiende las partes que tienen todos los `#()`. Esto se explicará en [Sección 5.6 \[Trucos avanzados con Scheme\]](#), página 59.

```
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line(:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
#{
  \once \override Score . RehearsalMark #'self-alignment-X = #left
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $markp }
#})

\relative c'' {
  \tempo 4=50
  a4.\mpdolce d8 cis4--\glissando a | b4 bes a2
  \tempoMark "Poco piu mosso"
  cis4.\< d8 e4 fis | g8(\! fis)-. e( d)-. cis2
}
```



Existen varios problemas con la salida que se superpone; los arreglaremos utilizando las técnicas descritas en [Sección 5.1 \[Mover objetos\]](#), página 52. Pero también haremos algo respecto a las definiciones `mpdolce` y `tempoMark`. Estas producen la salida que deseamos, pero quizá las querríamos utilizar en otra pieza. Podríamos simplemente copiarlas y pegarlas al principio de cada archivo, pero sería bastante molesto. También hace que se queden las definiciones a la vista dentro de nuestros archivos de música, y yo personalmente encuentro todos los `#()` bastante poco estéticos. Los vamos a esconder dentro de otro archivo:

```
%%% guardar esto en un archivo de nombre "definiciones.ly"
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line(:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
#{
  \once \override Score . RehearsalMark #'self-alignment-X = #left
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $markp }
#})
```

Ahora modificaremos la música (guardemos este archivo como `"musica.ly"`).

```
\include "definiciones.ly"

\relative c'' {
  \tempo 4=50
  a4.\mpdolce d8 cis4--\glissando a | b4 bes a2
  \once \override Score.RehearsalMark #'padding = #2.0
  \tempoMark "Poco piu mosso"
  cis4.\< d8 e4 fis | g8(\! fis)-. e( d)-. cis2
}
```



Eso tiene mejor aspecto, pero haremos algunos cambios más. El glissando es difícil de ver, así que lo haremos más grueso y lo acercaremos a las cabezas de las notas. Pondremos la indicación metronómica encima de la clave, en lugar de ir encima de la primera nota. Y por último, mi profesor de composición odia las indicaciones de compás 'C', así que la convertiremos en '4/4'.

Sin embargo, no debe cambiar el archivo 'musica.ly'. Sustituya nuestro archivo 'definiciones.ly' con éste:

```
%%% definiciones.ly
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line( #:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
#{
  \once \override Score . RehearsalMark #'self-alignment-X = #left
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $markp }
#})

\layout{
  \context { \Score
    \override MetronomeMark #'extra-offset = #'(-9 . 0)
    \override MetronomeMark #'padding = #'3
  }
  \context { \Staff
    \override TimeSignature #'style = #'numbered
  }
  \context { \Voice
    \override Glissando #'thickness = #3
    \override Glissando #'gap = #0.1
  }
}
```



¡Eso tiene un aspecto mucho mejor! Ahora suponga que quiere publicar esta pieza. A mi profesor de composición no le gustan las indicaciones de compás 'C', pero yo les tengo cierto cariño. Copiaremos el archivo actual 'definiciones.ly' a 'publicar-web.ly' y modificaremos éste. Como el propósito de esta música es producir un PDF que va a mostrarse en la pantalla, también vamos a aumentar el tamaño general de la salida.

```
%%% definiciones.ly
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line( #:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
#{
```

```

\once \override Score . RehearsalMark #'self-alignment-X = #left
\once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
\mark \markup { \bold $markp }
#})

#(set-global-staff-size 23)
\layout{
  \context { \Score
    \override MetronomeMark #'extra-offset = #'(-9 . 0)
    \override MetronomeMark #'padding = #'3
  }
  \context { \Staff
  }
  \context { \Voice
    \override Glissando #'thickness = #3
    \override Glissando #'gap = #0.1
  }
}

```



Ahora, en la música, simplemente sustituyo `\include "definiciones.ly"` por `\include "publicar-web.ly"`. Por supuesto, podríamos hacer esto aún más práctico. Podríamos hacer un archivo `'definiciones.ly'` que contuviera solamente las definiciones de `mpdolce` y `tempoMark`, un archivo `'web-publish.ly'` que contuviera solamente la sección `\layout` que se mostró en el ejemplo, y un archivo `'universidad.ly'` que contendría solamente los trucos para producir la salida que le gusta a mi profesor. La parte más alta de `'musica.ly'` tendría entonces este aspecto:

```

\include "definiciones.ly"

%%% ¡Quitar el comentario de una sola de estas líneas!
\include "publicar-web.ly"
%\include "universidad.ly"

```

Este enfoque puede ser útil incluso si va a producir sólo un conjunto de particellas. Yo utilizo media docena de archivos de 'hojas de estilo' para mis proyectos. Comienzo todos los archivos de música con `\include "../global.ly"`, que contiene

```

%%% global.ly
\version "2.11.15"
#(ly:set-option 'point-and-click #f)
\include "../iniciar/iniciar-definiciones.ly"

```

```
\include "../iniciar/iniciar-disposicion.ly"
\include "../iniciar/iniciar-cabeceras.ly"
\include "../iniciar/iniciar-papel.ly"
```

4.4 Actualizar archivos antiguos

La sintaxis de la entrada de LilyPond cambia de manera ocasional. A medida que el propio LilyPond mejora, la sintaxis (el lenguaje de la entrada) se modifica en consonancia. A veces estos cambios se hacen para conseguir que la entrada sea más fácil de leer y escribir, y otras veces estos cambios son para dar cabida a nuevas funcionalidades de LilyPond.

LilyPond lleva incorporado un archivo que facilita esta actualización: `convert-ly`. Para ver detalles sobre cómo ejecutar este programa, consulte program usage manual, *Updating files with convert-ly*.

Desgraciadamente `convert-ly` no puede tratar todos los cambios en la entrada. Se ocupa de los cambios sencillos de búsqueda y sustitución (como `raggedright` que se convierte en `ragged-right`), pero algunos cambios son demasiado complicados. Los cambios de sintaxis que `convert-ly` es incapaz de manejar se relacionan en program usage manual, *Updating files with convert-ly*.

Por ejemplo, en la versión 2.4 y anteriores de LilyPond, los acentos y las letras no inglesas se introducían utilizando LaTeX: por ejemplo, ‘No\`e1’ (que significa ‘Navidad’ en francés). En LilyPond 2.6 y siguientes, el carácter especial ‘ë’ debe introducirse directamente en el archivo de LilyPond como un carácter UTF-8. `convert-ly` no puede cambiar todos los caracteres especiales de LaTeX a caracteres de UTF-8; tendrá que actualizar manualmente sus archivos de LilyPond antiguos.

4.5 Resolución de problemas (tomar cada parte por separado)

Antes o después escribirá un archivo que LilyPond no podrá compilar. Los mensajes que LilyPond proporciona pueden ayudarle a encontrar el error, pero en muchos casos tendrá que llevar a cabo algún tipo de investigación para determinar el origen del problema.

Las herramientas más poderosas para este cometido son el comentario de una sola línea (indicado por `%`) y el comentario de bloque (indicado por `%{ ... %}`). Si no sabe dónde está el problema, comience cubriendo grandes secciones del archivo de entrada en un comentario. Después de eliminar una sección convirtiéndola en un comentario, pruebe a compilar el archivo otra vez. Si funciona, entonces el problema debía estar en la porción que había eliminado. Si no funciona, continúe eliminando material (transformándolo en comentarios) hasta que tenga algo que funcione.

En un caso extremo podría terminar con sólo

```
\score {
  <<
    % \melodia
    % \harmonia
    % \bajo
  >>
  \layout{ }
}
```

(en otras palabras: un archivo sin música)

Si ocurre esto, no abandone. Descomente un trozo pequeño – digamos la parte del bajo – y observe si funciona. Si no es así, transforme en comentarios toda la música del bajo (pero deje el `\bajo` de la sección `\score` no comentado).

```

bajo = \relative c' {
%{
  c4 c c c
  d d d d
%}
}

```

Ahora empiece poco a poco descomentando cada vez más fracciones de la parte del `bajo` hasta que encuentre la línea del problema.

Otra técnica de depuración muy útil es la construcción de [Sección 4.6 \[Ejemplos mínimos\]](#), [página 51](#).

4.6 Ejemplos mínimos

Un ejemplo mínimo es un ejemplo tan pequeño como sea posible. Estos ejemplos son mucho más fáciles de comprender que los ejemplos largos. Los ejemplos mínimos se utilizan para

- Informes de fallo
- Solicitudes de ayuda a las listas de correo
- Añadir ejemplos al [Repositorio de Fragmentos de Código de LilyPond](#)

Para construir un ejemplo que sea lo más pequeño posible, la regla es bastante simple: quite todo lo que no sea necesario. Al tratar de quitar partes innecesarias de un archivo, es una buena idea convertir líneas en comentarios en vez de borrarlas. De esta forma, si descubre que en realidad sí *necesitaba* algunas de estas líneas, podrá descomentarlas y no tendrá que teclearlas de nuevo partiendo de cero.

Existen dos excepciones a la regla del “lo más pequeño posible”:

- Incluya el número de `\version`.
- Si puede, ponga `\paper{ ragged-right=##t }` al principio del ejemplo.

En resumen, el objetivo de un ejemplo mínimo es que sea fácil de leer:

- Evite usar notas, tonalidades o compases demasiado complicados, a no ser que quiera demostrar algo sobre el comportamiento de estos elementos precisamente.
- No use comandos `\override` a no ser que ése sea el propósito del ejemplo.

5 Trucar la salida

Este capítulo trata de cómo modificar la salida. LilyPond es extremadamente configurable; prácticamente todos los fragmentos de la salida se pueden cambiar.

5.1 Mover objetos

Aunque pueda sorprenderle, LilyPond no es perfecto. Ciertos elementos de notación se pueden superponer, lo que es una lástima, pero en casi todos los casos se resuelve fácilmente.

HACER: con las nuevas funcionalidades de espaciado en la versión 2.12, estos ejemplos específicos ya no son de relevancia. Sin embargo siguen demostrando las poderosas funcionalidades de lilypond, así que quedan aquí hasta que alguien elabore unos ejemplos mejores.

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
e4~\markup{ \italic ritenuto } g b e
```



La solución más fácil es aumentar la distancia entre el objeto (texto en este caso, pero muy bien podrían ser digitaciones o dinámicas) y la nota. En LilyPond, esto se llama la propiedad `padding` (relleno); se mide en espacios de pentagrama. Para la mayor parte de los objetos, este valor ronda la cantidad de 1.0 o menos (varía dependiendo del objeto). Queremos aumentarlo, así que probaremos el valor 1.5

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'padding = #1.5
e4~\markup{ \italic ritenuto } g b e
```



Esto tiene un mejor aspecto, pero no es suficiente. Después de probar con algunos valores, creemos que 2.3 es el mejor número en este caso. Sin embargo esta cantidad es el mero resultado del ensayo y error y de mi gusto personal acerca de la notación. Pruebe el ejemplo anterior con 2.3... pero también con otros valores mayores (y menores). ¿Cuál cree que queda mejor?

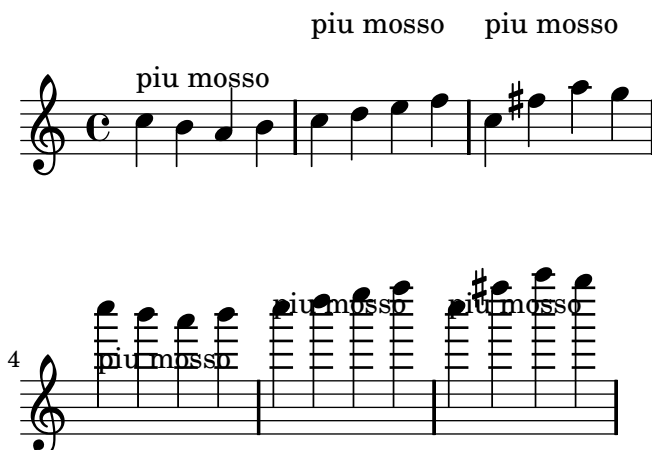
La propiedad `staff-padding` (relleno de pentagrama) está estrechamente relacionada. `padding` controla la cantidad de espacio mínima entre un objeto y el objeto más cercano (generalmente la nota o las líneas del pentagrama); `staff-padding` controla la cantidad mínima de espacio entre un objeto y el pentagrama. Ello supone una sutil diferencia, pero podrá observar el comportamiento a continuación.

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
c4~"piu mosso" b a b
\once \override TextScript #'padding = #4.6
c4~"piu mosso" d e f
\once \override TextScript #'staff-padding = #4.6
```

```

c4^"piu mosso" fis a g
\break
c'4^"piu mosso" b a b
\once \override TextScript #'padding = #4.6
c4^"piu mosso" d e f
\once \override TextScript #'staff-padding = #4.6
c4^"piu mosso" fis a g

```



Otra solución nos proporciona un control absoluto sobre la situación del objeto: podemos moverlo horizontal o verticalmente. Se hace con la propiedad `extra-offset` (desplazamiento adicional). Es ligeramente más complicado y puede causar otros problemas. Cuando movemos objetos con `extra-offset`, el movimiento se hace después de que LilyPond haya colocado todos los demás objetos. Esto significa que el resultado podría entrar en conflicto con otros objetos.

```

% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'extra-offset = #'( 1.0 . -1.0 )
e4^\markup{ \italic ritenuto } g b e

```



Con `extra-offset`, el primer número controla el movimiento horizontal (negativo hacia la izquierda); el segundo número controla el movimiento vertical (positivo hacia arriba). Después de algunos ensayos, hemos decidido que los siguientes valores son apropiados

```

% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'extra-offset = #'( -1.6 . 1.0 )
e4^\markup{ \italic ritenuto } g b e

```



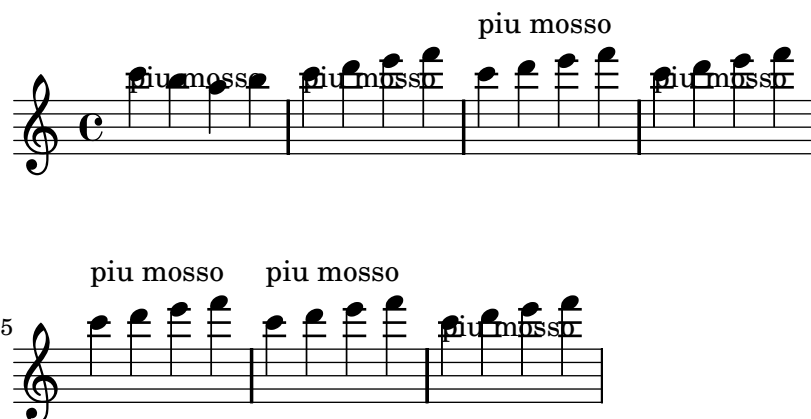
Una vez más, estos números son simplemente el resultado de algunos experimentos y de observar la salida. Quizá prefiera que el texto se encuentre algo más arriba, o a la izquierda, o en cualquier dirección. ¡Pruébalo y observe el resultado!

Una advertencia final: en esta sección hemos usado

```
\once \override TextScript ...
```

Esto altera la presentación del texto para la nota siguiente. Si la nota no tiene ningún texto, este truco no hace nada (y **no** se queda esperando al siguiente fragmento de texto). Para cambiar el comportamiento permanentemente a partir del comando, omita el `\once`. Para detener este truco, use `\revert` (revertir). Todo esto se explica en profundidad en [Sección 9.3 \[El comando `\override`\]](#), página 131.

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
c4^"piu mosso" b
\once \override TextScript #'padding = #4.6
a4 b
c4^"piu mosso" d e f
\once \override TextScript #'padding = #4.6
c4^"piu mosso" d e f
c4^"piu mosso" d e f
\break
\override TextScript #'padding = #4.6
c4^"piu mosso" d e f
c4^"piu mosso" d e f
\revert TextScript #'padding
c4^"piu mosso" d e f
```



Véase también

En el presente manual: [Sección 9.3 \[El comando `\override`\]](#), página 131, [Sección 5.3 \[Trucos comunes\]](#), página 55.

5.2 Arreglar notación con superposiciones

En [Sección 5.1 \[Mover objetos\]](#), página 52, pudimos ver cómo mover un objeto `TextScript`. El mismo mecanismo se puede usar para mover otros tipos de objetos; simplemente sustituya `TextScript` con el nombre de otro objeto.

Para encontrar el nombre del objeto, consulte la sección ‘**véase también**’ al final de la página relevante dentro de la documentación. Por ejemplo, al final de [Sección 6.6.3 \[Matices dinámicos\]](#), [página 103](#), vemos

Véase también

Referencia del programa: `DynamicText`, `Hairpin`. La posición vertical de estos símbolos se maneja por medio de `DynamicLineSpanner`.

Así que para mover expresiones dinámicas verticalmente, usamos

```
\override DynamicLineSpanner #'padding = #2.0
```

No podemos listar todos y cada uno de los objetos, pero presentamos a continuación una lista de los objetos más comunes.

Tipo de objeto		Nombre del objeto
Expresiones (verticalmente)	dinámicas	<code>DynamicLineSpanner</code>
Expresiones (horizontalmente)	dinámicas	<code>DynamicText</code>
Ligaduras de unión		<code>Tie</code>
Ligaduras de expresión		<code>Slur</code>
Articulaciones		<code>Script</code>
Digitaciones		<code>Fingering</code>
Texto, p.ej. <code>^"text"</code>		<code>TextScript</code>
Llamadas de ensayo o marcas de texto		<code>RehearsalMark</code>

5.3 Trucos comunes

Algunas sustituciones son tan comunes que se proporcionan comandos preestablecidos como atajos, como `\slurUp` (ligadura hacia arriba) y `\stemDown` (plica hacia abajo). Estos comandos se describen dentro de la Referencia de Notación bajo las secciones correspondientes.

La lista completa de modificaciones disponibles para cada tipo de objeto (como ligaduras o barras de corchea) están documentadas en la Referencia del Programa. Sin embargo, muchos objetos de la presentación comparten propiedades que se pueden usar para aplicar trucos genéricos.

- La propiedad `padding` (relleno) se puede establecer de forma que incremente (o disminuya) la distancia entre símbolos que se imprimen encima o debajo de las notas. Se aplica a todos los objetos con `side-position-interface`.

```
c2\fermata
\override Script #'padding = #3
b2\fermata
```

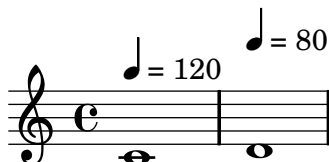


```
% This will not work, see below:
\override MetronomeMark #'padding = #3
\tempo 4=120
```

```

c1
% This works:
\override Score.MetronomeMark #'padding = #3
\tempo 4=80
d1

```



Observe en el segundo ejemplo cuán importante es determinar qué contexto maneja un objeto determinado. Debido a que el objeto `MetronomeMark` (indicación metronómica) se maneja en el contexto `Score`, los cambios de propiedades dentro del contexto `Voice` no se tendrán en cuenta. Para ver más detalles, consulte [Sección 9.3.1 \[Construir un truco\]](#), [página 131](#).

- La propiedad `extra-offset` mueve objetos en la salida; requiere una pareja de números. El primer número controla el movimiento horizontal, un número positivo moverá el objeto hacia la derecha. El segundo número controla el movimiento vertical; un número positivo lo desplazará hacia arriba. La propiedad `extra-offset` es una funcionalidad de bajo nivel: el motor de formateo es completamente olvidadizo respecto de estos desplazamientos.

En el ejemplo siguiente, la segunda digitación se desplaza un poco hacia la izquierda y 1.8 espacios de pentagrama hacia abajo:

```

\stemUp
f-5
\once \override Fingering
    #'extra-offset = #'(-0.3 . -1.8)
f-5

```



- El establecimiento de la propiedad `transparent` provocará que un objeto se imprima con ‘tinta invisible’: el objeto no se imprime, pero se conserva todo el resto de su comportamiento. El objeto aún ocupa un espacio, toma parte en las colisiones, y se le pueden adjuntar ligaduras de unión o de expresión y barras de corchea.

El ejemplo siguientes demuestra cómo conectar distintas voces utilizando ligaduras. Normalmente las ligaduras sólo unen dos notas de la misma voz. Al introducir una ligadura en una voz distinta,



y suprimiendo la primera plica hacia arriba en dicha voz, la ligadura parece cruzarse de una voz a otra:

```
<< {
  \once \override Stem #'transparent = ##t
  b8~ b8\noBeam
} \\\ {
  b[ g8]
} >>
```



Para asegurarse de que la plica que hemos suprimido no aprieta demasiado a la ligadura, también alargamos la plica, estableciendo su `length` (longitud) a 8,

```
<< {
  \once \override Stem #'transparent = ##t
  \once \override Stem #'length = #8
  b8~ b8\noBeam
} \\\ {
  b[ g8]
} >>
```



Las distancias en LilyPond se miden en espacios de pentagrama, mientras que las propiedades de grosor se miden en grosores de líneas de pentagrama. Algunas propiedades son diferentes; por ejemplo, el grosor de las barras de corchea se mide en espacios de pentagrama. Para más información, consulte la porción correspondiente de la referencia del programa.

5.4 Archivos por omisión

La documentación de la Referencia del Programa contiene una gran cantidad de información sobre LilyPond, pero más información aún se puede obtener a partir de la observación de los archivos internos de LilyPond.

Algunos ajustes por omisión como las definiciones de las `\header{}`s (encabezamientos) están almacenados en archivos `.ly`. Otros ajustes como las definiciones de los comandos de marcado se almacenan como archivos `.scm` (de Scheme). Cae fuera del ámbito de presente manual cualquier explicación más profunda; los usuarios están advertidos de que se necesita una considerable cantidad de conocimientos técnicos para comprender estos archivos.

- Linux: `'directorio_de_instalación/lilypond/usr/share/lilypond/current/'`
- OSX: `'carpeta_de_instalación/LilyPond.app/Contents/Resources/share/lilypond/current/'`. Para llegar aquí, o bien entre con `cd` en este directorio desde el Terminal, o haga control-click sobre la aplicación LilyPond y elija 'Mostrar el Contenido del Paquete'.
- Windows: `'carpeta_de_instalación/LilyPond/usr/share/lilypond/current/'`

Los directorios `'ly/'` y `'scm/'` son de especial interés. Archivos como `'ly/property-init.ly'` y `'ly/declarations-init.ly'` definen todos los trucos comunes.

5.5 Encajar la música en menos páginas

A veces puede acabar con uno o dos pentagramas en una segunda página (o tercera, o cuarta...). Es fastidioso sobre todo si observa las páginas anteriores y parece haber espacio suficiente en ellas.

Al investigar asuntos relacionados con la presentación, la herramienta `annotate-spacing` (anotar el espaciado) no tiene precio. Este comando imprime los valores de algunos comandos de espaciado; consulte [Sección 11.3 \[Mostrar el espaciado\]](#), página 135 para ver más detalles. A partir de la salida de `annotate-spacing`, podemos ver qué márgenes podríamos desear alterar.

Aparte de los márgenes, existen otras opciones para ahorrar espacio:

- Puede indicarle a LilyPond que coloque los sistemas tan juntos como sea posible (para que quepan tantos sistemas como sea posible sobre una página), pero luego separar estos sistemas para que no haya ningún espacio vacío al final de la página.

```
\paper {
  between-system-padding = #0.1
  between-system-space = #0.1
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Puede forzar el número de sistemas (es decir, si LilyPond quiere tipografiar la música con 11 sistemas, puede forzarlo para que use 10).

```
\paper {
  system-count = #10
}
```

- Evite (o reduzca) el uso de objetos que aumenten el tamaño vertical de un sistema. Por ejemplo, las repeticiones con primera y segunda vez necesitan espacio adicional. Si estas repeticiones abarcan dos sistemas, ocuparán más espacio que un solo sistema con las repeticiones y otro sistema sin ellas.

Otro ejemplo es desplazar las expresiones dinámicas que se ‘asoman por encima’ de un sistema.

```
\relative c' {
  e4 c g\ff c
  \override DynamicLineSpanner #'padding = #-1.8
  \override DynamicText #'extra-offset = #'(-2.1 . 0)
  e4 c g\ff c
}
```



- Altere el espaciado horizontal por medio de `SpacingSpanner`. Consulte [Sección 11.6.3 \[Cambiar el espaciado horizontal\]](#), página 136 para ver más detalles.

```
\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 4)
      }
    }
  }
}

```



5.6 Trucos avanzados con Scheme

Hemos visto cómo la salida de LilyPond se puede modificar de manera muy profunda utilizando comandos como `\override TextScript #'extra-offset = (1 . -1)`. Pero tenemos un potencial incluso mayor si utilizamos Scheme. Para ver una explicación completa de esto, consulte el [Apéndice B \[Tutorial de Scheme\]](#), página 143 e [Capítulo 12 \[Interfaces para programadores\]](#), página 137.

Podemos usar Scheme simplemente para sobreponer (`\override`) comandos,

```

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" d e f
  \padText #2.6
  c4^"piu mosso" fis a g
}

```



Podemos usarlo para crear comandos nuevos,

```

tempoMark = #(define-music-function (parser location padding marktext)
                    (number? string?)
  #{
    \once \override Score . RehearsalMark #'padding = $padding
    \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
    \mark \markup { \bold $marktext }
  #})

\relative c'' {
  c2 e
  \tempoMark #3.0 #"Allegro"
  g c
}

```



E incluso se le pueden pasar expresiones musicales.

```

pattern = #(define-music-function (parser location x y) (ly:music? ly:music?)
  #{
    $x e8 a b $y b a e
  #})

\relative c''{
  \pattern c8 c8\f
  \pattern {d16 dis} { ais16-> b\p }
}

```



5.7 Evitar los trucos con un proceso ralentizado

LilyPond puede llevar a cabo comprobaciones adicionales al tiempo que procesa los archivos. Estos comandos consumen tiempo, pero el resultado puede necesitar menos trucos manuales.

```

%% asegura que las marcas de texto y letras de las canciones se encuentran dentro de l
\override Score.PaperColumn #'keep-inside-line = ##t

```

6 Notación básica

Este capítulo explica cómo usar las funcionalidades de notación básica.

6.1 Alturas

Esta sección trata de cómo especificar la altura de las notas.

6.1.1 Alturas normales

El nombre de una nota se especifica usando las letras minúsculas de la **a** a la **g**. Una escala ascendente de Do mayor se tipografía con

```
\clef bass
c d e f g a b c'
```



El nombre de la nota **c** se tipografía una octava por debajo del Do central.

```
\clef treble
c1
\clef bass
c1
```



La especificación opcional de la octava tiene la forma de una serie de comillas simples o apóstrofes (‘’) o bien una serie de comas (‘,’). Cada ‘’ eleva la altura una octava; cada ‘,’ baja la altura una octava.

```
\clef treble
c' c'' e' g d'' d' d c
\clef bass
c, c,, e, g d,, d, d c
```



Se puede usar un método alternativo para declarar en qué octava grabar una nota; este método no requiere tantas especificaciones de octava (‘ y ,). Consulte [Sección 6.1.6 \[Octavas relativas\]](#), página 64.

6.1.2 Alteraciones accidentales

Un sostenido se forma añadiendo **-is** al final del nombre de la nota y un bemol se forma añadiendo **-es**. Los dobles sostenidos y dobles bemoles se obtienen añadiendo **-isis** o **-eses** al nombre de la nota.

```
a2 ais a aes
a2 aisis a aeses
```



Estos son los nombres de las notas en holandés. En holandés, **aes** se contrae en **as**, pero las dos formas se aceptan. De forma similar, se aceptan tanto **es** como **ees**

```
a2 as e es
```



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
```



La entrada **d e f** se interpreta como ‘imprimir un Re natural, Mi natural y Fa natural’, independientemente de la armadura de la tonalidad. Para ver más información acerca de la distinción entre el contenido musical y la presentación de ese contenido, consulte [Sección 2.2.2 \[Alteraciones accidentales y armaduras\]](#), página 18.

```
\key d \major
d e f g
d e fis g
```



Propiedades trucadas frecuentemente

De acuerdo con las reglas estándar del tipografiado, se imprime un signo de becuadro antes de un sostenido o un bemol siempre que una alteración anterior tiene que cancelarse. Para modificar este comportamiento, utilice `\set Staff.extraNatural = ##f`

```
ceses4 ces cis c
\set Staff.extraNatural = ##f
ceses4 ces cis c
```



Véase también

Referencia del programa: `LedgerLineSpanner`, `NoteHead`.

6.1.3 Alteraciones de precaución

Normalmente las alteraciones accidentales se imprimen automáticamente, pero también puede imprimirlas manualmente. Una alteración recordatoria se puede forzar añadiendo un signo de admiración `!` después de la altura de la nota. Una alteración de precaución (o sea, una alteración entre paréntesis) se puede obtener 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
```



Véase también

La producción automática de alteraciones accidentales se puede ajustar de muchas formas. Para ver más información, consulte [Sección 9.1.1 \[Alteraciones accidentales automáticas\]](#), página 131.

6.1.4 Microtonos

Los medios bemoles y los medios sostenidos se forman añadiendo `-eh` y `-ih`; a continuación presentamos una serie de DOs cada vez más agudos

```
\set Staff.extraNatural = ##f
ceseh ceh cih cisih
```



Los microtonos también se exportan al archivo MIDI.

Fallos

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.

6.1.5 Nombres de las notas en otros idiomas

Existen conjuntos predefinidos de nombres de notas para algunos idiomas aparte del inglés. Para usarlos, incluya el archivo de inicio específico del idioma. Por ejemplo, añada `\include "english.ly"` al principio del archivo de entrada. Los archivos de idioma disponibles y los nombres de las notas que definen son los siguientes:

	Nombres de las notas								sostenido	bemol	doble sostenido	doble bemol
nederlands.ly	c	d	e	f	g	a	bes	b	-is	-es	-isis	-es
english.ly	c	d	e	f	g	a	bf	b	-s/-sharp	-f/-flat	-ss/-x/ -sharpsharp	-ff/ -fla
deutsch.ly	c	d	e	f	g	a	b	h	-is	-es	-isis	-es
norsk.ly	c	d	e	f	g	a	b	h	-iss/-is	-ess/-es	-ississ/-isis	-ess
svenska.ly	c	d	e	f	g	a	b	h	-iss	-ess	-ississ	-ess
italiano.ly	do	re	mi	fa	sol	la	sib	si	-d	-b	-dd	-bb
catalan.ly	do	re	mi	fa	sol	la	sib	si	-d/-s	-b	-dd/-ss	-bb
espanol.ly	do	re	mi	fa	sol	la	sib	si	-s	-b	-ss	-bb

Tenga en cuenta que en holandés, alemán, noruego y sueco, los bemoles del ‘La’ como por ejemplo `aes` y `aeses` se contraen normalmente como `as` y `ases` (o más frecuentemente como `asas`). Algunas veces se definen sólo estas contracciones en los archivos de idioma correspondientes (esto también se aplica a los sufijos para cuartos de tono que se ven más abajo).

Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol ‘normales’. Los nombres de nota para cuartos de tono que se definen en los distintos archivos de idioma se encuentran listados en la tabla siguiente. Aquí los prefijos ‘semi-’ y ‘sesqui-’ significan ‘medio’ y ‘uno y medio’ respectivamente. Para el noruego, sueco, catalán y español no se ha definido aún ningún nombre especial.

	Nombres de las notas								semi-sostenido	semi-bemol	sesqui-sostenido	sesqui-bemol
nederlands.ly	c	d	e	f	g	a	bes	b	-ih	-eh	-isih	-eseh
english.ly	c	d	e	f	g	a	bf	b	-qs	-qf	-tqs	-tqf
deutsch.ly	c	d	e	f	g	a	b	h	-ih	-eh	-isih	-eseh
norsk.ly	c	d	e	f	g	a	b	h				
svenska.ly	c	d	e	f	g	a	b	h				
italiano.ly	do	re	mi	fa	sol	la	sib	si	-sd	-sb	-dsd	-bsb
catalan.ly	do	re	mi	fa	sol	la	sib	si				
espanol.ly	do	re	mi	fa	sol	la	sib	si				

6.1.6 Octavas relativas

Las octavas se especifican añadiendo ‘`'` y ‘`,`’ a los nombres de las notas. Cuando copie música existente, es fácil poner por accidente una nota en la octava equivocada y luego es difícil encontrar dicho error. La octava relativa previene estos errores haciendo que éstos tengan consecuencias mucho mayores: un solo error transporta una octava el resto de la pieza

```
\relative altura_inicial expresión_musical
```

o bien

`\relative expresión_musical`

`c'` se usa por defecto si no se define una altura inicial.

La octava de las notas que aparecen dentro de *expresión_musical* se calculan como sigue: si no se usa ninguna marca de cambio de octava, el intervalo básico entre la nota actual y la anterior se toma siempre como si fuera una cuarta o menos. Esta distancia se determina independientemente de las alteraciones; un **fisis** (Fa doble sostenido) después de un **ceses** (Do doble bemol) se colocará por encima del **ceses**. En otras palabras, una cuarta doble aumentada se considera un intervalo menor que una quinta disminuida, incluso aunque la cuarta doble aumentada tiene siete semitonos mientras que la quinta disminuida sólo tiene seis semitonos.

Las marcas de cambio de octava ' y , se pueden añadir para elevar o bajar la altura en una octava más. Al entrar en el modo relativo, se puede especificar una altura de inicio absoluta que se comportará como la nota precedente a la primera nota de la *expresión_musical*. Si no se especifica ninguna altura inicial, se utilizará el Do central como comienzo.

Aquí podemos ver el modo relativo en acción:

```
\relative c'' {
  b c d c b c bes a
}
```



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

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



Si el elemento anterior es un acorde, la primera nota del acorde se utiliza para determinar la primera nota del siguiente acorde

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



La altura después del `\relative` contiene un nombre de nota.

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` o `\relative` en su argumento. Para usar el modo relativo dentro de música transportada, se debe escribir un `\relative` adicional dentro de `\transpose`.

6.1.7 Comprobación de la octava

Las comprobaciones de octava hacen que los errores de octava sean más fáciles de corregir: una nota puede venir seguida de *=comillas* lo que indica cuál debería ser su octava absoluta. En el siguiente ejemplo,

```
\relative c'' { c='' b=' d,='' }
```

la *d* generará una advertencia, porque se esperaba un *d''* (a causa de que el intervalo de *b'* a *d''* es de una tercera solamente), pero aparece un *d'*. En la salida, la octava se corrige para que sea un *d''* y la siguiente nota se calcula de forma relativa a *d''* en lugar de a *d'*.

También existe una comprobación de octava que no produce ninguna salida visible. La sintaxis

```
\octave altura
```

comprueba que *altura* (sin las comillas) produce *altura* (con las comillas) en modo `\relative` comparado con la nota que se da en el comando `\relative`. Si no es así, se imprime una advertencia, y la octava se corrige. La *altura* no se imprime como una nota.

En el ejemplo que aparece a continuación, la primera comprobación se aprueba sin más problema, porque el *e* (en modo `relative`) está dentro de una quinta desde *a'*. Sin embargo la segunda comprobación produce una advertencia porque el *e* no está dentro de una quinta del *b'*. Se imprime el mensaje de advertencia y la octava se ajusta de modo que las notas siguientes están en la octava correcta de nuevo.

```
\relative c' {
  e
  \octave a'
  \octave b'
}
```

La octava de una nota que sigue a una comprobación de octava se determina con respecto a la nota que la precede. En el siguiente fragmento, la última nota es una *a'*, por encima del Do central. Esto significa que la comprobación de `\octave` se aprueba satisfactoriamente, así que la comprobación se puede borrar sin que cambie la salida de la pieza.

```
\relative c' {
  e
  \octave b
  a
}
```



6.1.8 Transposición

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

```
\transpose desde hacia expresión_musical
```

Esto significa que la *expresión_musical* se transporta el intervalo que hay entre las notas *desde* y *hacia*: cualquier nota con la altura *desde* se cambia por *hacia*.

Por ejemplo, tomemos una pieza escrita en la tonalidad de Re mayor. Si esta pieza es demasiado grave para el ejecutante, se puede transportar hacia arriba a Mi mayor con

```
\transpose d e ...
```

Suponga una pieza escrita para el violín (que es un instrumento en Do). Si esta parte se tiene que 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), el siguiente transporte producirá la particella correcta:

```
\transpose a c ...
```

`\transpose` distingue entre notas enarmónicas: tanto `\transpose c cis` como `\transpose c des` transportarán un semitono hacia arriba. La primera versión imprimirá sostenidos y la segunda imprimirá bemoles

```
mus = { \key d \major cis d fis g }
\new Staff {
  \clef "F" \mus
  \clef "G"
  \transpose c g' \mus
  \transpose c f' \mus
}
```



`\transpose` también se puede usar para introducir notas escritas para un instrumento transpositor. Las alturas se introducen normalmente en LilyPond en Do (o en ‘afinación de concierto’), pero se pueden introducir en otra tonalidad. Por ejemplo, al introducir música para una trompeta en Si bemol que comienza en un Re de concierto, se puede escribir

```
\transpose c bes { e4 ... }
```

Para imprimir esta música en Si bemol de nuevo (o sea, para producir una particella de trompeta en vez de una partitura para el director en afinación de concierto), puede envolver la música existente con otro `transpose`

```
\transpose bes c { \transpose c bes { e4 ... } }
```

Véase también

Referencia del programa: `TransposedMusic`.

Ejemplo: ‘`scheme/transpose-pitches-with-minimum-accidentals.ly`’.

Fallos

Si quiere usar al mismo tiempo `\transpose` y `\relative`, debe escribir `\transpose` por fuera de `\relative`, porque `\relative` no tendrá ningún efecto sobre la música que aparezca dentro de un `\transpose`.

6.1.9 Silencios

Los silencios se introducen como notas con el nombre `r`

```
r1 r2 r4 r8
```



Los silencios de un compás centrados en medio del compás, se deben hacer con silencios multicompas. Se pueden usar para un solo compás así como para muchos compases, y se tratan en [Sección 8.2.1 \[Silencios multicompas\]](#), página 128.

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:

```
a'4\rest d'4\rest
```



Esto hace mucho más sencillo el formateo de la música polifónica, porque el formateador automático de colisiones de silencios se olvidará de ellos.

Véase también

Referencia del programa: `Rest`.

6.1.10 Desplazamientos

Un silencio invisible – también conocido como ‘skip’ (desplazamiento) – se puede introducir como una nota con el nombre ‘s’ o con `\skip duración`

```
a4 a4 s4 a4 \skip 1 a4
```



La sintaxis de `s` sólo está disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, debe usar el comando `\skip`

```
<<
  \relative { a'2 a2 }
  \new Lyrics \lyricmode { \skip 2 bla2 }
>>
```



El comando `skip` es un mero contenedor musical vacío. No produce ninguna salida, ni siquiera una salida transparente.

El comando de desplazamiento `s` crea una **Staff** y una **Voice** cuando es necesario, de forma similar a los comandos de nota y de silencio. Por ejemplo, el fragmento siguiente produce un pentagrama vacío.

```
{ s4 }
```



El fragmento `{ \skip 4 }` produciría una página vacía.

Véase también

Referencia del programa: `SkipMusic`.

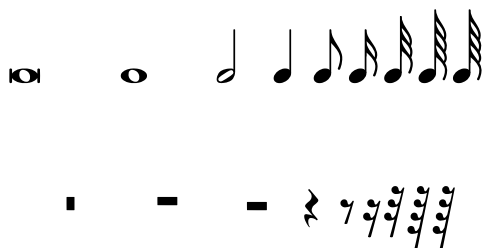
6.2 Ritmos

Esta sección trata los ritmos, duraciones y compases.

6.2.1 Duraciones

En los modos de Nota, Acorde y Letra, 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 los comandos `\longa` y `\breve`

```
c'\breve
c'1 c'2 c'4 c'8 c'16 c'32 c'64 c'64
r\longa r\breve
r1 r2 r4 r8 r16 r32 r64 r64
```



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 }
```



6.2.2 Puntillos

Para obtener duraciones de notas con puntillo, simplemente escriba un punto (‘.’) después del número. Las notas con doble puntillo se consiguen de manera similar.

```
a'4 b' c'4. b'8 a'4. b'4.. c'8.
```



Comandos predefinidos

Normalmente los puntillos se desplazan hacia arriba para evitar las líneas del pentagrama, excepto en situaciones de polifonía. Se pueden utilizar los siguientes comandos para forzar manualmente una dirección en particular

`\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Véase también

Referencia del programa: `Dots` y `DotColumn`.

6.2.3 Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical multiplicando todas las duraciones por una fracción:

`\times fracción expresión_musical`

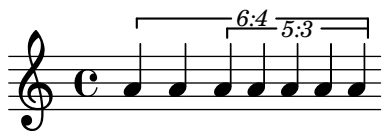
La duración de la *expresión_musical* se multiplicará por la fracción. El denominador de la fracción se imprimirá sobre las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor de 2, por tanto las notas duran $\frac{2}{3}$ de su valor escrito

`g'4 \times 2/3 {c'4 c' c'} d'4 d'4`



Los grupos pueden anidarse unos dentro de otros, por ejemplo

```
\override TupletNumber #'text = #tuplet-number::calc-fraction-text
\times 4/6 {
  a4 a
  \times 3/5 { a a a a a }
}
```



Comandos predefinidos

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

Propiedades trucadas frecuentemente

La propiedad `tupletSpannerDuration` (duración del selector de grupo) especifica la duración de cada corchete. Con esto puede hacer muchos grupos escribiendo `\times` una sola vez, ahorrándose mucho trabajo de teclado. En el siguiente ejemplo se muestran dos tresillos, aunque `\times` sólo se ha utilizado una vez

```
\set tupletSpannerDuration = #(ly:make-moment 1 4)
\times 2/3 { c8 c c c c c }
```



Para ver más información sobre `make-moment`, consulte [Sección 8.4.2 \[Gestión del tiempo\]](#), [página 129](#).

El formato del número se determina por medio de la propiedad `text` en `TupletNumber`. Por defecto se imprime sólo el denominador, pero si su valor se establece a la función `tuplet-number::calc-fraction-text`, se imprimirá como *numerador:denominador*.

Para que no se impriman los números de los grupos, emplee

```
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
\override TupletNumber #'transparent = ##t
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
```



Utilice la función `\tweak` para sobrescribir los grupos anidados que comienzan en el mismo instante musical. En este ejemplo, `\tweak` especifica un texto de fracción para el `TupletNumber` exterior y un texto de denominador para el `TupletNumber` del primero de los tres grupos interiores.

```
\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \times 4/3 {
    \tweak #'text #tuplet-number::calc-denominator-text
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}
```



Aquí `\tweak` y `\override` trabajan conjuntamente para especificar la dirección del `TupletBracket` (el corchete del grupo). El primer `\tweak` coloca el `TupletBracket` del grupo exterior por encima del pentagrama. El segundo `\tweak` coloca el `TupletBracket` del primero de los tres grupos interiores por debajo del pentagrama. Tenga en cuenta que este par de funciones `\tweak` afecta solamente al grupo exterior y al primero de los tres grupos interiores porque sólo esos dos grupos comienzan en el mismo instante musical. Utilizamos `\override` de la manera usual para colocar los corchetes `TupletBrackets` del segundo y tercer grupos interiores por debajo del pentagrama.

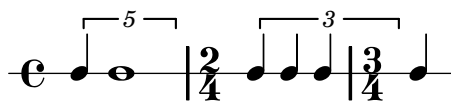
```

\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \tweak #'direction #up
  \times 4/3 {
    \tweak #'direction #down
    \times 2/3 { c'8[ c'8 c'8] }
    \override TupletBracket #'direction = #down
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}

```



Se puede hacer que los corchetes de grupo abarquen tanto como su duración completa indique o que lleguen hasta justo antes de la nota siguiente



Véase también

Referencia del programa: `TupletBracket`, `TupletNumber` y `TimeScaledMusic`.

6.2.4 Escalar las duraciones

Es posible alterar la duración de las figuras en un factor N/M añadiendo ‘ $*N/M$ ’ (o ‘ $*N$ ’ si $M=1$). Esto no afectará a la apariencia de las notas o silencios que se producen. Se pueden combinar en la forma ‘ $*M*N$ ’.

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

```

\time 2/4
a4*2/3 gis4*2/3 a4*2/3
a4 a4 a4*2
b16*4 c4

```



Véase también

En este manual: [Sección 6.2.3 \[Grupos especiales\]](#), página 70.

6.2.5 Comprobación del 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, '|'. Cuando se encuentra durante la interpretación, debe caer en el límite entre dos compases. Si no es así, se muestra una advertencia. 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, por ejemplo

```
\lyricmode {
  \time 2/4
  Bri -- lla | Bri -- lla
}
```

Las comprobaciones de compás fallidas tienen su origen en la introducción incorrecta de las duraciones. 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.

También es posible redefinir el significado de |. Se hace asignando una expresión musical a pipeSymbol,

```
pipeSymbol = \bar "||"

{ c'2 c' | c'2 c' }
```



6.2.6 Comprobación del número de compás

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
```

imprimirá una advertencia si el currentBarNumber (número del compás actual) no es 123 cuando se está procesando.

6.2.7 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. En los siguientes ejemplos, las notas que atraviesan la barra de compás se dividen y se unen mediante una ligadura.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
} {
  c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2
}
```





Este grabador divide todas las notas largas en la barra de compás e inserta ligaduras. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

Si quiere permitir el salto de línea en las barras de compás en las que `Completion_heads_engraver` divide las notas, debe eliminar también `Forbid_line_break_engraver`.

Fallos

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

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

Véase también

Referencia del programa: `Completion_heads_engraver`.

6.3 Polifonía

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.

6.3.1 Acordes

Un acorde se forma encerrando un conjunto de notas entre `<` y `>`. Un acorde puede ir seguido de una duración o un conjunto de articulaciones, como si fueran simples notas.

`<c e g>4 <c>8`



Para ver más información sobre los acordes, consulte [Sección 7.2 \[Nombres de acorde\]](#), [página 115](#).

6.3.2 Plicas

Cuando se encuentra con una nota, se crea un objeto `Stem` (plica) automáticamente. Para las redondas y los silencios, también se crean pero se hacen invisibles.

Comandos predefinidos

`\stemUp`, `\stemDown`, `\stemNeutral`.

Propiedades trucadas frecuentemente

Para modificar la dirección de las plicas en medio de un pentagrama, utilice

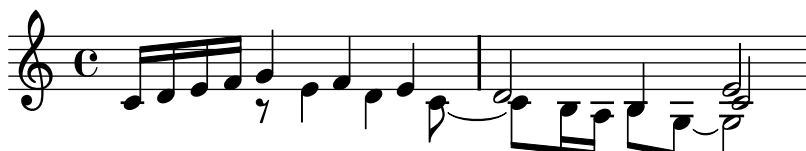
```
a4 b c b
\override Stem #'neutral-direction = #up
a4 b c b
\override Stem #'neutral-direction = #down
a4 b c b
```



6.3.3 Polifonía básica

La manera más fácil de escribir fragmentos con más de una voz sobre el mismo pentagrama es introducir cada voz como una secuencia (con `{...}`), y combinarlas simultáneamente, separando las voces con `\`

```
\new Staff \relative c' {
  c16 d e f
  <<
    { g4 f e | d2 e2 } \
    { r8 e4 d c8 ~ | c b16 a b8 g ~ g2 } \
    { s2. | s4 b4 c2 }
  >>
}
```



El separador hace que se produzca un nuevo contexto de `Voice`¹. Llevan los nombres "1", "2", etc. En cada uno de estos contextos, la dirección vertical de las ligaduras, plicas, etc., se establece de la manera que corresponda.

Todas estas voces están separadas de la voz que contiene las notas justo por fuera de la construcción `<< \ >>`. Se debe tener esto en cuenta cuando haga algún cambio en una voz. También significa que las ligaduras de unión y de expresión no pueden salir de (ni entrar en) una construcción `<< \ >>`. En cambio las voces paralelas que están en construcciones `<< \ >>` separadas dentro del mismo pentagrama, son la misma voz. Presentamos a continuación el mismo ejemplo, con cabezas distintas para cada voz. Dése cuenta de que el cambio en el estilo de cabezas de nota en la voz principal no afecta a las que están dentro de la construcción `<< \ >>`. Asimismo, el cambio a la segunda voz en la primera construcción `<< \ >>` es efectiva en la segunda `<< \ >>`, y la voz se liga de una construcción a otra.

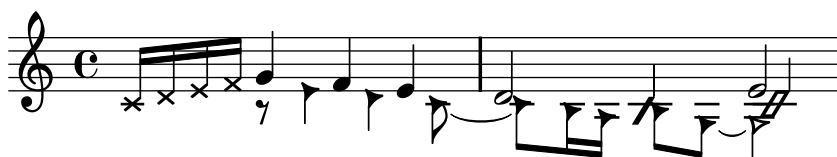
```
\new Staff \relative c' {
  \override NoteHead #'style = #'cross
  c16 d e f
  <<
    { g4 f e } \
```

¹ Las voces polifónicas se conocen a veces como ‘capas’ en otros programas de edición de partituras.

```

    { \override NoteHead #'style = #'triangle
      r8 e4 d c8 ~ }
  >> |
  <<
    { d2 e2 } \\\
    { c8 b16 a b8 g ~ g2 } \\\
    { \override NoteHead #'style = #'slash s4 b4 c2 }
  >>
}

```



La polifonía no cambia la relación entre las notas que están dentro de un bloque `\relative { }`. Cada una de las notas se calcula relativa a la nota inmediatamente anterior.

```
\relative { notaA << notaB \\\ notaC >> notaD }
```

`notaC` es relativa a `notaB`, no a `notaA`; `notaD` es relativa a `notaC`, no a `notaB` ni a `notaA`.

6.3.4 Voces explícitas

Los contextos de `Voice` también se pueden instanciar manualmente dentro de un bloque `<< >>` para crear música polifónica, usando `\voiceOne`, hasta `\voiceFour` para asignar las direcciones de las plicas y un desplazamiento horizontal para cada parte.

Concretamente:

```
<< \superior \\\ \inferior >>
```

equivale a

```

<<
  \new Voice = "1" { \voiceOne \superior }
  \new Voice = "2" { \voiceTwo \inferior }
>>

```

Los comandos `\voiceXXX` establecen la dirección de las plicas, ligaduras de expresión y de unión, articulaciones, anotaciones de texto, puntillos y digitaciones. `\voiceOne` y `\voiceThree` hacen a los mencionados objetos apuntar hacia arriba, mientras que `\voiceTwo` y `\voiceFour` los hacen apuntar hacia abajo. El comando `\oneVoice` hace volver a los ajustes normales.

Una expresión que aparece directamente dentro de un `<< >>` pertenece a la voz principal. Es útil cuando aparecen voces adicionales al tiempo que está cantando la voz principal. A continuación podemos ver una representación mejor del ejemplo del apartado anterior. Las cabezas en forma de cruz demuestran que la melodía principal ahora está en un contexto de una voz única.

```

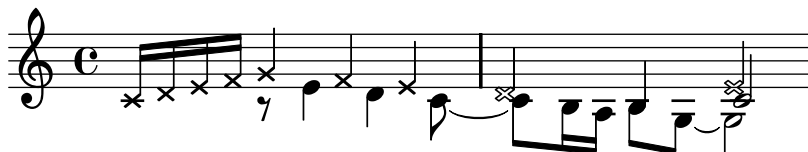
\new Staff \relative c' {
  \override NoteHead #'style = #'cross
  c16 d e f
  \voiceOne
  <<
    { g4 f e | d2 e2 }
    \new Voice="1" { \voiceTwo
      r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
    }
    \oneVoice
  >>
}

```

```

    }
    \new Voice { \voiceThree
      s2. | s4 b4 c2
      \oneVoice
    }
  >>
  \oneVoice
}

```

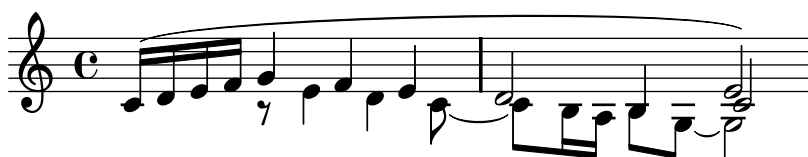


La definición correcta de las voces permite que la melodía se pueda marcar con ligaduras de expresión.

```

\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
    { g4 f e | d2 e2) }
    \context Voice="1" { \voiceTwo
      r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
      \oneVoice
    }
    \new Voice { \voiceThree
      s2. s4 b4 c2
      \oneVoice
    }
  >>
  \oneVoice
}

```



Al evitar el separador `\\` posibilitamos la escritura de construcciones polifónicas anidadas, lo que en algún caso podría constituir una forma más natural de tipografiar la música.

```

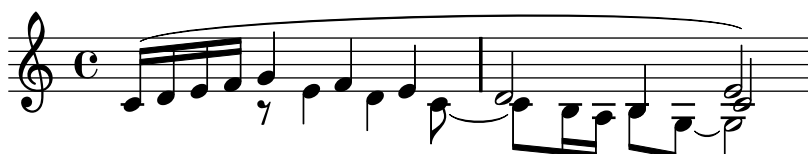
\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
    { g4 f e | d2 e2) }
    \context Voice="1" { \voiceTwo
      r8 e4 d c8 ~ |
      <<
        {c8 b16 a b8 g ~ g2}
        \new Voice { \voiceThree

```

```

        s4 b4 c2
        \oneVoice
    }
    >>
    \oneVoice
    }
    >>
    \oneVoice
    }

```



En ciertos casos de música polifónica compleja podría ser necesario utilizar voces adicionales para evitar colisiones entre las notas. Las voces adicionales se añaden definiendo un identificador como se muestra a continuación:

```

voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c''' <<
  { \voiceOne g4 ~ \stemDown g32[ f( es d c b a b64 )g] } \\
  { \voiceThree b4} \\
  { \voiceFive d,} \\
  { \voiceTwo g,}
>>

```



6.3.5 Resolución de las colisiones

Normalmente las cabezas de nota con un número de puntillos diferente no se mezclan en una sola, pero cuando se le da un valor verdadero a la propiedad `merge-differently-dotted` en el objeto `NoteCollision`, se juntan en una sola:

```

\new Voice << {
  g8 g8
  \override Staff.NoteCollision
    #'merge-differently-dotted = ##t
  g8 g8
} \\ { g8.[ f16] g8.[ f16] } >>

```



De forma similar puede mezclar cabezas de blanca con corcheas, ajustando `merge-differently-headed`:

```
\new Voice << {
  c8 c4.
  \override Staff.NoteCollision
    #'merge-differently-headed = ##t
  c8 c4. } \ { c2 c2 } >>
```



`merge-differently-headed` y `merge-differently-dotted` se aplican solamente a direcciones de plica opuestas (es decir, a las voces 1 y 2).

LilyPond también desplaza verticalmente los silencios que están en el lado contrario de una plica, por ejemplo

```
\new Voice << c''4 \ r4 >>
```



Si tres o más notas coinciden verticalmente en la misma columna, `merge-differently-headed` no puede completar satisfactoriamente la mezcla de las dos notas que deberían mezclarse. Para hacer posible que la mezcla funcione, aplique un `\shift` (desplazamiento) a la nota que no debe mezclarse. En el primer compás del ejemplo siguiente, `merge-differently-headed` no funciona (la cabeza de la blanca está rellena) En el segundo compás, se aplica `\shiftOn` para mover horizontalmente el g (Sol) superior, y `merge-differently-headed` funciona correctamente.

```
\override Staff.NoteCollision #'merge-differently-headed = ##t
<<
  { d=''2 g2 } \
  { \oneVoice d=''8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
<<
  { d=''2 \shiftOn g2 } \
  { \oneVoice d=''8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
```



Comandos predefinidos

`\oneVoice`, `\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`: estos comandos especifican el grado en que se deben desplazar los acordes de la voz en curso. Las voces exteriores (normalmente: la voz uno y la voz dos) tienen `\shiftOff` (desplazamiento desactivado), mientras que las voces interiores (tres y cuatro) tienen `\shiftOn` (desplazamiento activado). `\shiftOnn` y `\shiftOnnn` definen niveles de desplazamiento más grandes.

Cuando LilyPond no puede con todo, se pueden usar la propiedad `force-hshift` (forzar desplazamiento) del objeto `NoteColumn` y los silencios con altura para sobrescribir las decisiones de tipografía.

```
\relative <<
{
  <d g>
  <d g>
} \ {
  <b f'>
  \once \override NoteColumn #'force-hshift = #1.7
  <b f'>
} >>
```



Véase también

Referencia del programa: los objetos responsables de resolver las colisiones son `NoteCollision` y `RestCollision`.

Fallos

Cuando se emplea `merge-differently-headed` con una corchea (o una nota más corta) con la plica hacia arriba, y una blanca con la plica hacia abajo, la corchea no tiene el desplazamiento correcto.

No hay soporte para racimos (clusters) donde la misma nota se presenta con diferentes alteraciones accidentales dentro del mismo acorde. En este caso se recomienda usar la transcripción enarmónica, o usar la notación especial de racimos (ver [Sección 8.4.4 \[Racimos \(clusters\)\]](#), página 129).

6.4 Notación de los pentagramas

Esta sección describe la notación musical que se produce en el nivel del pentagrama, como las armaduras de tonalidad, claves e indicaciones de compás.

6.4.1 Clave

La clave indica qué líneas del pentagrama corresponden a qué notas. La clave se establece con el comando `\clef`

```
{ c''2 \clef alto g'2 }
```



Las claves soportadas son

Clave	Posición
treble, violin, G, G2	clave de Sol en 2ª
alto, C	clave de Do en 3ª
tenor	clave de Do en 4ª
bass, F	clave de Fa en 4ª
french	clave de Sol en 1ª, conocida también como clave de violín francesa
soprano	clave de Do en 1ª
mezzosoprano	clave de Do en 2ª
baritone	clave de Do en 5ª
varbaritone	clave de Fa en 3ª
subbass	clave de Fa en 5ª
percussion	clave de percusión
tab	clave de tablatura

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. El argumento *nombre_de_clave* se debe encerrar entre comillas cuando contiene guiones bajos o dígitos. Por ejemplo:

```
\clef "G_8" c4
```



Propiedades trucadas frecuentemente

El comando `\clef "treble_8"` equivale a establecer `clefGlyph`, `clefPosition` (que controla la posición Y de la clave), `middleCPosition` y `clefOctavation`. Se imprime una clave cuando se cambia cualquiera de estas propiedades. El siguiente ejemplo muestra las posibilidades que se ofrecen estableciendo manualmente las propiedades.

```
{
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'4
  \set Staff.clefGlyph = #"clefs.G"
  c'4
  \set Staff.clefGlyph = #"clefs.C"
  c'4
  \set Staff.clefOctavation = #7
  c'4
  \set Staff.clefOctavation = #0
}
```

```

\set Staff.clefPosition = #0
c'4
\clef "bass"
c'4
\set Staff.middleCPosition = #4
c'4
}

```



Véase también

En el presente manual: [Sección 6.5.7 \[Notas de adorno\]](#), página 97.

Referencia del programa: `Clef`.

6.4.2 Armadura de la tonalidad

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 el comando `\key`

```
\key nota tipo
```

Aquí, *tipo* 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), `\locrian` (locrio), `\aeolian` (eolio), `\mixolydian` (mixolidio), `\lydian` (lidio), `\phrygian` (frigio) y `\dorian` (dórico).

Este comando establece la propiedad de contexto `Staff.keySignature`. Se pueden especificar armaduras no estándar estableciendo esta propiedad directamente.

Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios, porque las notas sin alteración pueden llevar signos de becuadro en función de la armadura. Para ver más información, consulte [Sección 6.1.2 \[Alteraciones accidentales\]](#), página 62 o [Sección 2.2.2 \[Alteraciones accidentales y armaduras\]](#), página 18.

```

\key g \major
f1
fis

```



Propiedades trucadas frecuentemente

Se imprime un símbolo de becuadro para cancelar las alteraciones anteriores. Se puede suprimir esto estableciendo la propiedad `Staff.printKeyCancellation`.

```

\key d \major
a b cis d
\key g \minor
a bes c d
\set Staff.printKeyCancellation = ##f
\key d \major
a b cis d
\key g \minor
a bes c d

```



Véase también

Referencia del programa: `KeyCancellation`, `KeySignature`.

6.4.3 Indicación de compás

La indicación de compás indica el metro de la pieza: su patrón regular de partes fuertes y débiles. Se denota por una fracción al comienzo del pentagrama.

La indicación de compás se establece mediante el comando `\time`

```
\time 2/4 c'2 \time 3/4 c'2.
```



Propiedades trucadas frecuentemente

Se puede personalizar el símbolo que se imprime mediante la propiedad `style`. Un valor de `#'()` produce un estilo de fracción para los compases de 4/4 y de 2/2,

```

\time 4/4 c'1
\time 2/2 c'1
\override Staff.TimeSignature #'style = #'()
\time 4/4 c'1
\time 2/2 c'1

```

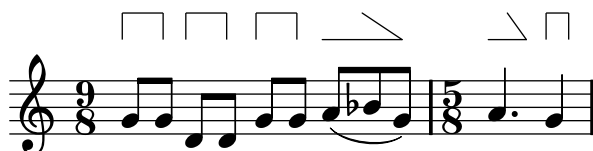


Existen otras muchas opciones para su presentación. Para ver más ejemplos, consulte [Sección 7.7.6 \[Indicaciones antiguas de compás\]](#), página 118.

`\time` establece las propiedades `timeSignatureFraction`, `beatLength` y `measureLength` en el contexto `Timing`, cuyo nombre alternativo normalmente es `Score`. La propiedad `measureLength` determina el lugar en que se deben insertar las barras de compás y cómo se generan las barras de corchea automáticas. La modificación del valor de `timeSignatureFraction` también provoca que se imprima el símbolo.

Están disponibles más opciones a través de la función de Scheme `set-time-signature`. En combinación con el grabador `Measure_grouping_engraver` (agrupación de compases), creará signos de `MeasureGrouping`. Estos signos facilitan la lectura de música moderna rítmicamente compleja. En el siguiente ejemplo el compás de 9/8 se subdivide en 2, 2, 2 y 3. Esto se pasa a `set-time-signature` como un tercer argumento con el valor (2 2 2 3)

```
\score {
  \relative c'' {
    #(set-time-signature 9 8 '(2 2 2 3))
    g8[ g] d[ d] g[ g] a8[( bes g)] |
    #(set-time-signature 5 8 '(3 2))
    a4. g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Véase también

Referencia del programa: `TimeSignature` y `Timing_translator`.

Ejemplos: `'contemporary/compound-time-signature.ly'`.

Fallos

EL barrado automático no utiliza la agrupación de compases especificada con `set-time-signature`.

6.4.4 Compases parciales

Los compases parciales como las anacrusas o partes al alzar se escriben usando la construcción

```
\partial 16*5 c16 cis d dis e | a2. c,4 | b2
```





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.

Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice

`\bar ""`

De esta manera se insertará una barra invisible de compás y se hará posible el salto de línea en este punto (sin incrementar el contador de los números de compás).

En las partituras con muchos pentagramas, un comando `\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 {
      e'4 d'
      \bar "||"
      f' e'
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



Propiedades trucadas frecuentemente

El comando `\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.

Una línea divisoria se crea cuando se establece la propiedad `whichBar`. Al comienzo del compás su valor se toma de `Timing.defaultBarType`. Los contenidos de los `repeatCommands` se utilizan para sobrescribir a las barras de compás por omisión.

Le recomendamos que utilice siempre `\repeat` para hacer repeticiones. Consulte [Sección 6.7 \[Repeticiones\]](#), página 109.

Véase también

En este manual: [Sección 6.7 \[Repeticiones\]](#), página 109, [Sección 6.4.7 \[Delimitadores del comienzo de un sistema\]](#), página 87.

Referencia del programa: `BarLine` (creada al nivel de `Staff` (pentagrama)), `SpanBar` (a través de los pentagramas).

6.4.6 Música sin compasear

Las líneas divisorias y los números de compás se calculan automáticamente. Para música sin compasear (por ejemplo cadencias), esto no es deseable. Para desactivar las barras y los números de compás automáticos, utilice los comandos `\cadenzaOn` y `\cadenzaOff`.

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



Fallos

LilyPond insertará saltos de línea y de página solamente en las barras de compás. A menos que la música sin compasear acabe antes del final de la línea del pentagrama, tendrá que insertar barras de compás invisibles

```
\bar ""
```

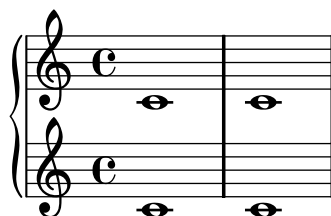
para indicar dónde se pueden producir los saltos.

6.4.7 Delimitadores del comienzo de un sistema

Muchas partituras constan de más de un pentagrama. Estos pentagramas se pueden unir de cuatro formas distintas:

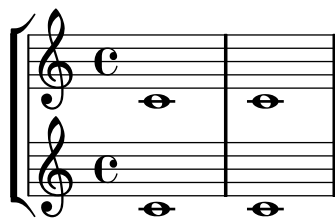
- El grupo comienza con una llave a la izquierda y las líneas de compás están conectadas. Esto se hace con el contexto `GrandStaff`.

```
\new GrandStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



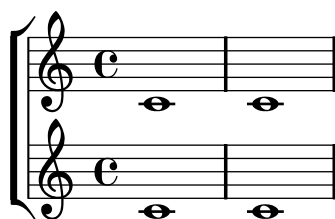
- El grupo comienza con un corchete recto y las líneas de compás están conectadas. Esto se hace con el contexto `StaffGroup`

```
\new StaffGroup
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- El grupo comienza con un corchete recto pero las líneas de compás no están conectadas. Esto se hace con el contexto **ChoirStaff**.

```
\new ChoirStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- El grupo comienza con una línea vertical. Las líneas de compás no están conectadas. Ésta es la opción por defecto para la partitura.

```
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



Véase también

Las barras de compás al principio de cada sistema son **SystemStartBar**, **SystemStartBrace** y **SystemStartBracket**. En cada contexto solamente se crea uno de estos tipos, y ese tipo está determinado por la propiedad **systemStartDelimiter**.

Propiedades trucadas frecuentemente

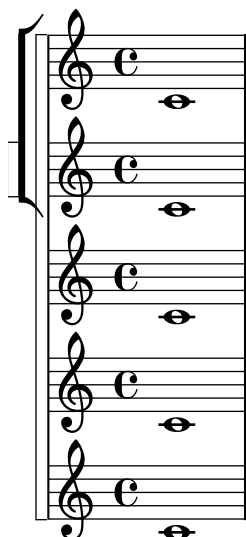
Los delimitadores del comienzo del sistema se pueden anidar muchas veces:

```
\new StaffGroup
\relative <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBracket a (SystemStartSquare b)) d)
  \new Staff { c1 }
  \new Staff { c1 }
```

```

\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
>>

```



6.4.8 El símbolo del pentagrama

Las notas, signos dinámicos, etc. están agrupados con un conjunto de líneas horizontales que reciben el nombre de pentagrama o pauta ('staff', en plural 'staves'). En LilyPond, estas líneas se dibujan utilizando un objeto de presentación separado que se llama el **staff symbol** (el símbolo del pentagrama).

El símbolo del pentagrama se puede ajustar en el número, grosor y separación de las líneas, usando las propiedades. Esto se demuestra en los archivos de ejemplo 'staff/changing-the-number-of-lines-in-a-staff.ly' y 'staff/changing-the-staff-size.ly'.

Además, los pentagramas se pueden iniciar e interrumpir libremente. Esto se hace con `\startStaff` y `\stopStaff`.

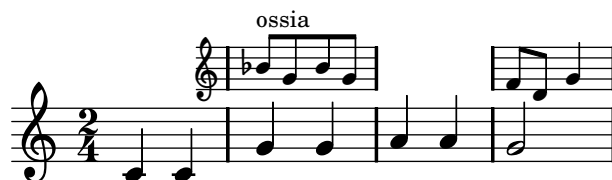
```

b4 b
\override Staff.StaffSymbol #'line-count = 2
\stopStaff \startStaff
b b
\revert Staff.StaffSymbol #'line-count
\stopStaff \startStaff
b b

```



En combinación con los pentagramas ocultos o franceses, esto se puede usar para componer tipográficamente secciones de ossia. He aquí un ejemplo:



Véase también

Referencia del programa: `StaffSymbol`.

Ejemplos: `'staff'`

6.4.9 Escribir música en paralelo

La música para varias partes se puede intercalar

```
\parallelMusic #'(voiceA voiceB) {
  r8      g'16[ c'' ] e''[ g' c'' e'' ] r8      g'16[ c'' ] e''[ g' c'' e'' ] |
  c'2                                           c'2 |
  r8      a'16[ d'' ] f''[ a' d'' f'' ] r8      a'16[ d'' ] f''[ a' d'' f'' ] |
  c'2                                           c'2 |
}
\new StaffGroup <<
  \new Staff \new Voice \voiceA
  \new Staff \new Voice \voiceB
>>
```



Esto funciona bastante bien para música de piano

```
music = {
  \key c \major
  \time 4/4
  \parallelMusic #'(voiceA voiceB voiceC voiceD) {
    % Bar 1
    r8 g'16[ c'' ] e''[ g' c'' e'' ] r8 g'16[ c'' ] e''[ g' c''
e'' ] |
    c'2 |
    r8 a16[ d' ] f'[ a d' f'] r8 a16[ d' ] f'[ a d' f'] |
    c2 |
  }
```

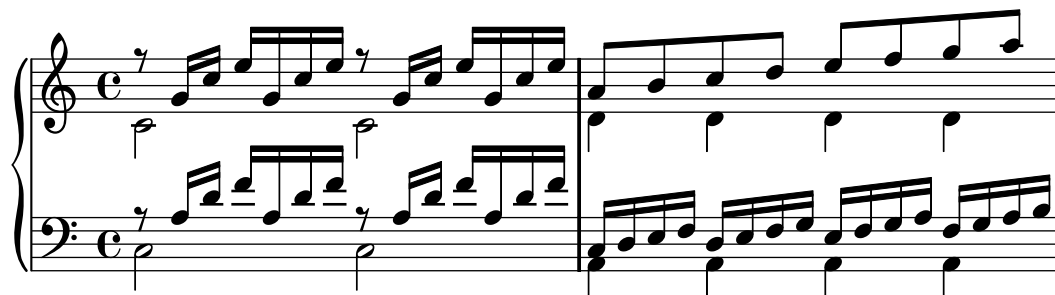
```

% Bar 2
a'8 b'      c'' d''      e'' f''      g'' a'' |
d'4         d'         d'         d' |
c16 d e f    d e f g    e f g a    f g a b |
a,4         a,4         a,4         a,4 |

% Bar 3 ...
}
}

\score {
  \new PianoStaff <<
    \music
    \new Staff <<
      \voiceA \
      \voiceB
    >>
    \new Staff {
      \clef bass
      <<
        \voiceC \
        \voiceD
      >>
    }
  >>
}

```



6.5 Conectar notas

Esta sección trata de la notación que afecta a grupos de notas.

6.5.1 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. 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. Se introduce una ligadura de unión usando el símbolo de la tilde curva ‘~’

$e' \sim e' <c' e' g'> \sim <c' e' g'>$



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 la ligadura dentro del acorde,

`<c~ e g~ b> <c e g b>`



Una ligadura es simplemente una forma de extender la duración de una nota, parecida al puntillo. El ejemplo siguiente muestra dos formas de escribir exactamente el mismo concepto:



Se usan ligaduras de expresión bien cuando la nota atraviesa la barra de compás o bien cuando no se pueden usar puntillos para denotar el ritmo. Cuando se usan ligaduras, las notas de mayor duración se deben alinear con las subdivisiones del compás, como



Si necesita unir muchas notas a través de las líneas divisorias, puede resultar más fácil utilizar la división automática de notas (vea [Sección 6.2.7 \[División automática de las notas\]](#), página 73). Este mecanismo divide las notas largas de forma automática y las liga a través de las barras de compás.

Cuando la segunda vez de una repetición comienza con una nota ligada, es necesario repetir la ligadura. Esto se puede conseguir mediante `\repeatTie`,



Propiedades trucadas frecuentemente

A veces se usan ligaduras para escribir arpeggios. En este caso, dos notas ligadas no necesitan ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` (ligadura - esperar nota) al valor verdadero. La misma funcionalidad puede resultar útil, por ejemplo, para ligar un trémolo a un acorde. Por ejemplo,

```
\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
```



Las ligaduras de pueden grabar manualmente modificando la propiedad `tie-configuration`. El primer número indica la distancia desde el centro del pentagrama en espacios de pentagrama, y el segundo número indica la dirección (1 = hacia arriba, -1 = hacia abajo).

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



Comandos predefinidos

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieSolid`.

Véase también

En el presente manual: [Sección 6.2.7 \[División automática de las notas\]](#), página 73.

Referencia del programa: `Tie`.

Fallos

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.

6.5.2 Ligaduras de expresión

Una ligadura de expresión indica que las notas se deben tocar unidas o *legato*. Se escriben utilizando paréntesis

```
f( g a) a8 b( a4 g2 f4)
<c e>2( <b d>2)
```



La dirección de una ligadura de expresión se puede especificar con `\slurDIRECCIÓN`, donde *DIRECCIÓN* es o bien `Up` (arriba), `Down` (abajo), o `Neutral` (seleccionada automáticamente).

Sin embargo, hay una forma abreviada muy conveniente para forzar las direcciones de las ligaduras de expresión. Escribiendo `_` o `^` antes de los paréntesis de apertura, también se establece la dirección. Por ejemplo,

```
c4_( c) c^( c)
```

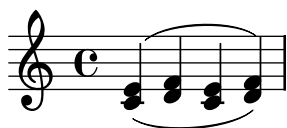


Se puede imprimir una sola ligadura cada vez. Si necesita imprimir una ligadura larga por encima de varias ligaduras más cortas, consulte [Sección 6.5.3 \[Ligaduras de fraseo\]](#), página 94.

Propiedades trucadas frecuentemente

Algunos compositores escriben dos ligaduras cuando quieren acordes legato. Esto se consigue en LilyPond mediante el establecimiento de `doubleSlurs`,

```
\set doubleSlurs = ##t
<c e>4 ( <d f> <c e> <d f> )
```



Comandos predefinidos

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurSolid`.

Véase también

Referencia del programa: `internals document`, `Slur`.

6.5.3 Ligaduras de fraseo

Una ligadura de fraseo (o marca de fraseo) conecta las notas y se utiliza para indicar una frase musical. Se escribe usando `\(` y `\)` respectivamente

```
\time 6/4 c'\( d( e) f( e) d\)
```



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; utilice en su lugar `\phrasingSlurUp`, `\phrasingSlurDown`, y `\phrasingSlurNeutral`.

No se pueden tener varias ligaduras de fraseo simultáneas.

Comandos predefinidos

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`.

Véase también

Referencia del programa: `PhrasingSlur`.

6.5.4 Ligaduras Laissez vibrer

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 escribiendo `\laissezVibrer`,

```
<c f g>\laissezVibrer
```



Véase también

Referencia del programa: `LaissezVibrerTie LaissezVibrerTieColumn`

Archivos de ejemplo: ‘`connecting/laissez-vibrer-ties.ly`’

6.5.5 Barras automáticas

LilyPond inserta las barras de corchea automáticamente

```
\time 2/4 c8 c c c \time 6/8 c c c c8. c16 c8
```



Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita. También es posible definir patrones de barrado que difieran de los ajustes por defecto. Para ver más detalles, consulte [Sección 9.1.2 \[Establecer el comportamiento de las barras automáticas\]](#), página 131.

Las notas individuales se pueden marcar con `\noBeam` para evitar que lleven una barra

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



Véase también

Referencia del programa: `Beam`.

6.5.6 Barras manuales

En ciertos casos puede ser preciso sobrescribir el algoritmo de barrado automático. Por ejemplo, el barrador automático no escribe barras por encima de los silencios o las líneas divisorias. Tales barras se especifican manualmente marcando los puntos de comienzo y final con `[y]`

```
{
  r4 r8[ g' a r8] r8 g[ | a] r8
}
```



Propiedades trucadas frecuentemente

Normalmente los patrones de barrado dentro de una barra se determinan automáticamente. Si es necesario se pueden usar las propiedades `stemLeftBeamCount` y `stemRightBeamCount` para sobrescribir los ajustes por defecto. Si se establece el valor de una propiedad cualquiera, este valor se utilizará una sola vez, y después se borrará.

```
{
  f8[ r16
    f g a]
  f8[ r16
    \set stemLeftBeamCount = #1
    f g a]
}
```



Se puede establecer la propiedad `subdivideBeams` para subdividir todas las barras de semi-corchea y de valores inferiores en las posiciones de las fracciones o partes, tal y como está definido por parte de la propiedad `beatLength`.

```
c16[ c c c c c c c]
\set subdivideBeams = ##t
c16[ c c c c c c c]
\set Score.beatLength = #(ly:make-moment 1 8)
c16[ c c c c c c c]
```



Para ver más información sobre `make-moment`, consulte [Sección 8.4.2 \[Gestión del tiempo\]](#), página 129.

Los saltos de línea normalmente están prohibidos cuando las barras cruzan las líneas divisorias. Este comportamiento se puede modificar estableciendo `breakable`.

Se insertan automáticamente barras de corchea en ángulo cuando se detecta un desnivel muy grande entre las cabezas de las notas. Este comportamiento se puede ajustar a través del objeto `auto-knee-gap`.

Fallos

Las barras que cruzan pentagramas y reciben un ángulo automático no se pueden usar conjuntamente con los pentagramas ocultos. Consulte [Sección 8.3.2 \[Ocultar pentagramas\]](#), página 128.

Las barras de corchea no evitan las colisiones con los símbolos que rodean las notas, como los textos y las alteraciones.

6.5.7 Notas de adorno

Los mordentes y notas de adorno son ornamentos escritos. El más común es la acciaccatura o mordente de una nota, que se debe tocar muy corta. Se denota mediante una corchea pequeña, ligada, con un corchete tachado o atravesado por una barrita. La appoggiatura es una nota de adorno que toma una fracción fija de la nota principal, y se denota como una nota pequeña, ligada, sin tachar. Ambas se introducen con los comandos `\acciaccatura` y `\appoggiatura`, como se muestra en el ejemplo siguiente

```
b4 \acciaccatura d8 c4 \appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```



Ambas son formas especiales del comando `\grace`. Al anteponer esta palabra a una expresión musical se crea una nueva expresión, que se imprime en un tipo más pequeño y no ocupa ningún tiempo lógico en el compás.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2 c4
```



Al contrario que `\acciaccatura` y `\appoggiatura`, el comando `\grace` no inicia una ligadura.

Internamente, la cuenta del tiempo para las notas de adorno se hace utilizando un segundo contador, el contador ‘de adorno’. Cada instante en el tiempo comprende dos números racionales: uno denota el tiempo lógico, y el otro denota el tiempo de adorno. El ejemplo anterior se muestra a continuación con vectores de tiempo



La colocación de las 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 { e4 \grace { c16[ d e f] } e4 }
\new Staff { c4 \grace { g8[ b] } c4 } >>
```



Si quiere acabar una nota con un adorno, use el comando `\afterGrace`. Toma dos argumentos: la nota principal y las notas de adorno que siguen a la nota principal.

```
c1 \afterGrace d1 { c16[ d] } c4
```



Esto pondrá las notas de adorno después de un ‘espacio’ que dura $\frac{3}{4}$ de la longitud de la nota principal. La fracción $\frac{3}{4}$ se puede cambiar estableciendo `afterGraceFraction`, por ejemplo

```
#(define afterGraceFraction (cons 7 8))
```

pondrá la nota de adorno a $\frac{7}{8}$ de la nota principal.

Se puede conseguir el mismo efecto de forma manual haciendo

```
\new Voice {
  << { d1^\trill_( }
    { s2 \grace { c16[ d] } } >>
  c4)
}
```



Ajustando la duración de la nota de desplazamiento (aquí es una blanca), se ajusta el espacio entre la nota principal y la de adorno.

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, por ejemplo:

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4
}
```



Las sobreescrituras (overrides) también se deben deshacer dentro de la expresión de adorno.

La disposición de las expresiones de adorno se pueden cambiar a lo largo de la música usando la función `add-grace-property`. El siguiente ejemplo suprime la definición de la dirección de la Stem (plica) para este adorno, y así las plicas no siempre apuntan hacia arriba.

```
\new Staff {
  #(add-grace-property 'Voice 'Stem 'direction '())
  ...
}
```

Otra opción es modificar las variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` y `stopAppoggiaturaMusic`. Hay más información en el archivo `ly/grace-init.ly`.

La barrita que cruza el corchete de las acciaccaturas se puede obtener en otras situaciones mediante `\override Stem #'stroke-style = #"grace"`.

Propiedades trucadas frecuentemente

Se puede forzar que las notas de adorno utilicen espaciado flotante,



Véase también

Referencia del programa: `GraceMusic`.

Fallos

Una partitura que comienza con una expresión `\grace` requiere una declaración explícita `\new Voice` pues en caso contrario la nota principal y la de adorno acabarán en pentagramas distintos.

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 d4 }
    \new Staff { c4 \bar "|:" d4 } >>
```



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

```
\new Staff { c4 \bar "|:" \grace s16 d4 }
```

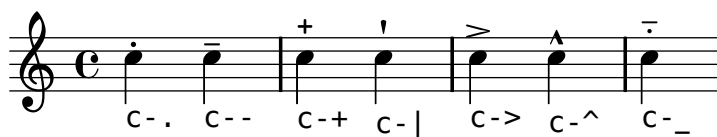
Las secciones de adorno sólo se deben usar dentro de expresiones de música secuenciales. No están soportados el anidado ni la yuxtaposición de secciones de adorno, y podría producir caídas u otros errores.

6.6 Expresiones

Las marcas de expresión ayudan al músico a sacar de la música algo más que simples notas y ritmos.

6.6.1 Articulaciones

Un amplio abanico de símbolos pueden aparecer encima o debajo de las notas para indicar distintas características de la ejecución. Se adjuntan a una nota escribiendo un guión y el carácter que significa la articulación. Se muestran a continuación:



Los significados de estas abreviaturas se pueden modificar. Consulte `'ly/script-init.ly'` para ver ejemplos.

La indicación se coloca automáticamente, pero la dirección (arriba o abajo) se puede también forzar. Como otros fragmentos de código de LilyPond, `_` los situará por debajo del pentagrama, y `^` los colocará por encima.

```
c''4^^ c''4_^
```

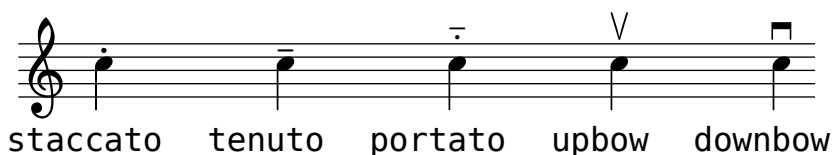
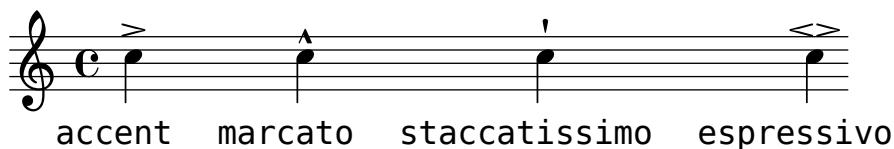


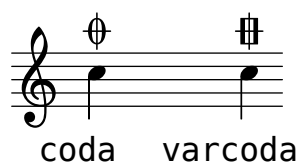
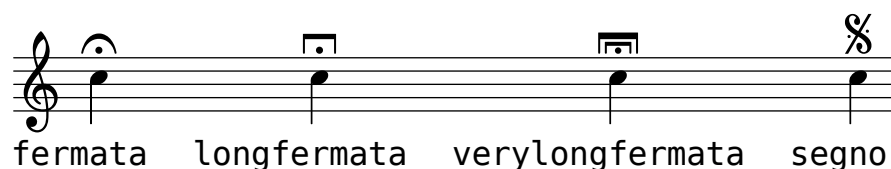
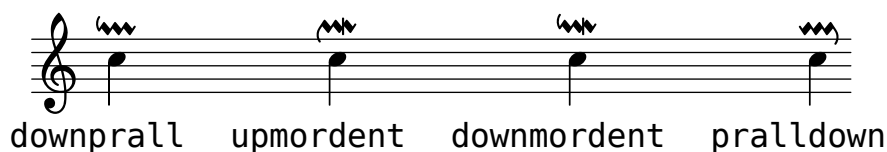
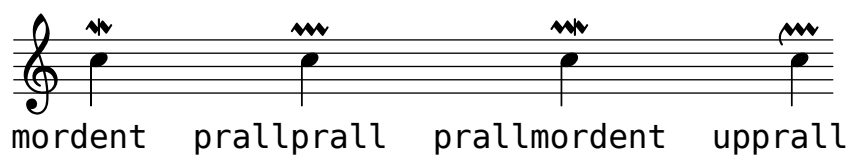
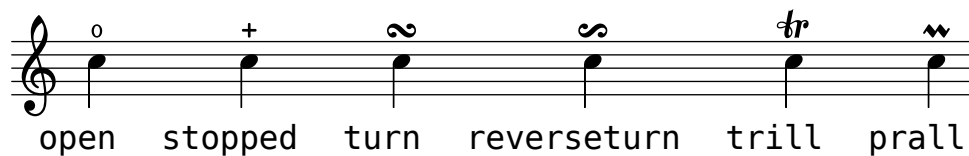
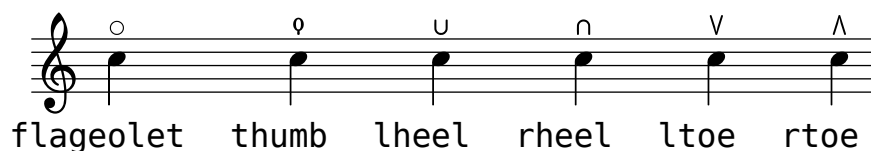
Se pueden añadir otros símbolos usando la sintaxis `nota\nombre`. Una vez más, se puede forzar que aparezcan por encima o por debajo usando `^` y `_`, por ejemplo

```
c\fermata c^\fermata c_\fermata
```



He aquí un cuadro que muestra todas las articulaciones disponibles:





Propiedades trucadas frecuentemente

El orden vertical de las articulaciones se controla con la propiedad `script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo el `TextScript` (el símbolo de sostenido) tiene primero la prioridad más baja, así que se coloca en la parte más baja en el primer ejemplo. En el segundo, el semitrino (el `Script`) tiene la prioridad más baja, por tanto se coloca en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se escriben decide cuál de ellos va primero.

```
\once \override TextScript #'script-priority = #-100
a4^\prall^\markup { \sharp }
```

```
\once \override Script #'script-priority = #-100
a4^\prall^\markup { \sharp }
```



Véase también

Referencia del programa: `Script`.

Fallos

Estos signos aparecen en la salida impresa pero no tienen ningún efecto sobre la representación de MIDI de la música.

6.6.2 Indicaciones de digitación

Las instrucciones de digitación se pueden introducir usando

nota-dígito

Para los cambios de dedo, utilice marcas de texto.

```
c4-1 c-2 c-3 c-4
```

```
c^\markup { \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 cello)

```
<a_\thumb a'-3>8 <b_\thumb b'-3>
```



Las digitaciones para los acordes también se pueden añadir a las notas individuales del acorde escribiéndolas después de las alturas

```
< c-1 e-2 g-3 b-5 >4
```



Propiedades trucadas frecuentemente

Puede ejercer un mayor control sobre la digitación de los acordes estableciendo `fingeringOrientations`

```
\set fingeringOrientations = #'(left down)
<c-1 es-2 g-4 bes-5 > 4
\set fingeringOrientations = #'(up right down)
<c-1 es-2 g-4 bes-5 > 4
```



Usando esta funcionalidad, también es posible poner instrucciones de digitación muy cerca de las cabezas de nota en música monofónica,

```
\set fingeringOrientations = #'(right)
<es'-2>4
```



Véase también

Referencia del programa: `Fingering`.

Ejemplos: `'expressive/fingering-chords.ly'`

6.6.3 Matrices dinámicas

Las marcas dinámicas de matiz absoluto se especifican usando un comando después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz` y `\rfz`.

```
c\ppp c\pp c \p c\mp c\mf c\f c\ff c\fff
c2\fp c\s f c\sff c\sp c\spp c\s f z c\rfz
```



Un regulador de crescendo se comienza con `\<` y se termina con `\!` o con un matiz absoluto. Un regulador decrescendo comienza con `\>` y se termina también con `\!` o con un matiz dinámico absoluto. Se pueden usar `\cr` y `\decr` en lugar de `\<` y `\>`. A causa de que estas marcas se unen a notas, se deben usar notas espaciadoras si se necesitan varias marcas durante una nota.

```
c\< c\! d\> e\!
<< f1 { s4 s4\< s4\! \> s4\! } >>
```



Un regulador comienza normalmente en el borde izquierdo de la nota inicial y acaba en el borde derecho de la nota final. Si la nota final cae sobre el principio de un compás, el regulador finaliza en la línea divisoria inmediatamente anterior. Esto se puede modificar estableciendo la propiedad `hairpinToBarline`,

```
\set hairpinToBarline = ##f
c4\< c2. c4\!
```



En algunas situaciones, la marca de articulación `\espressivo` puede ser apropiada para indicar un crescendo y decrescendo sobre una nota,

```
c2 b4 a g1\espressivo
```



Esto puede dar lugar a reguladores muy cortos. Utilice `minimum-length` dentro de `Voice.Hairpin` para alargarlos, por ejemplo

```
\override Voice.Hairpin #'minimum-length = #5
```

Los reguladores se pueden imprimir con la punta en círculo (notación al niente) estableciendo la propiedad `circled-tip`,

```
\override Hairpin #'circled-tip = ##t
c2\< c\!
c4\> c\< c2\!
```



También puede usar texto que diga *cresc.* en vez de reguladores

```
\setTextCresc
c\< d e f\!
\setHairpinCresc
e\> d c b\!
\setTextDecresc
c\> d e f\!
\setTextDim
```

```
e\> d c b\!
```



Asimismo puede aportar sus propios textos

```
\set crescendoText = \markup { \italic "cresc. poco" }
\set crescendoSpanner = #'dashed-line
a'2\< a a a\!\mf
```



Para crear nuevas indicaciones dinámicas o texto que se deba alinear con los matices, consulte [Sección 8.1.11 \[Indicaciones dinámicas contemporáneas\]](#), página 127.

El posicionamiento vertical de las expresiones dinámicas se gestiona a través de `DynamicLineSpanner`.

Propiedades trucadas frecuentemente

Las expresiones dinámicas que suceden en, comienzan sobre, o acaban en, la misma nota se alinearán verticalmente. Si quiere asegurar que las expresiones dinámicas se alinean cuando no suceden sobre la misma nota, puede aumentar el valor de la propiedad `staff-padding`.

```
\override DynamicLineSpanner #'staff-padding = #4
```

También puede utilizar esta propiedad si las expresiones dinámicas colisionan con otros elementos de la notación.

Los crescendi y decrescendi que acaban sobre la misma nota de una línea nueva no se imprimen. Para cambiar este comportamiento, utilice

```
\override Score.Hairpin #'after-line-breaking = ##t
```

Los cambios dinámicos de tipo texto (como *cresc.* y *dim.*) se imprimen con una línea de puntos que muestra su alcance. Para evitar que se imprima esta línea, use

```
\override DynamicTextSpanner #'dash-period = #-1.0
```

Comandos predefinidos

```
\dynamicUp, \dynamicDown, \dynamicNeutral.
```

Véase también

Referencia del programa: `DynamicText`, `Hairpin`. El posicionamiento vertical de estos símbolos se maneja por medio de `DynamicLineSpanner`.

6.6.4 Marcas de respiración

Las marcas de respiración se introducen usando `\breathe`

```
c'4 \breathe d4
```



Propiedades trucadas frecuentemente

El glifo de la marca de respiración se puede ajustar por medio de la sobreescritura de la propiedad `text` del objeto de presentación `BreathingSign` con cualquier texto de marcado. Por ejemplo,

```
c'4
\override BreathingSign #'text
= #(make-musicglyph-markup "scripts.rvarcomma")
\breathe
d4
```



Véase también

Referencia del programa: `BreathingSign`.

Ejemplos: `'expressive/breathing-sign.ly'`

6.6.5 Trinos

Los trinos cortos se imprimen como una articulación normal; consulte [Sección 6.6.1 \[Articulaciones\]](#), página 100.

Los trinos largos mantenidos se hacen con `\startTrillSpan` y `\stopTrillSpan`,

```
\new Voice {
  << { c1 \startTrillSpan }
    { s2. \grace { d16[\stopTrillSpan e] } } >>
  c4 }
```



Los trinos que se tienen que ejecutar sobre notas especificadas explícitamente se pueden tipografiar con el comando `pitchedTrill`,

```
\pitchedTrill c4\startTrillSpan fis
f\stopTrillSpan
```



El primer argumento es la nota principal. La altura de la segunda se imprime como una cabeza de nota sin plica entre paréntesis.

Comandos predefinidos

`\startTrillSpan`, `\stopTrillSpan`.

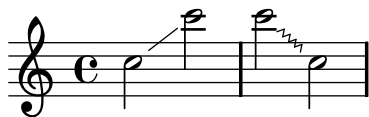
Véase también

Referencia del programa: `TrillSpanner`.

6.6.6 Glissando

Un glissando es un cambio gradual en la altura. Se denota por medio de una línea o una línea ondulada entre dos notas. Se llama adjuntando `\glissando` a una nota

```
c2\glissando c'  
\override Glissando #'style = #'zigzag  
c2\glissando c,
```



Véase también

Referencia del programa: `Glissando`.

Archivos de ejemplo: `'expressive/glissando.ly'`, `'expressive/line-styles.ly'`

Fallos

La impresión de texto sobre la línea (como *gliss.*) no está soportada.

6.6.7 Arpeggio

Se puede especificar un signo de arpeggio (conocido también como acorde quebrado) sobre un acorde adjuntando un `\arpeggio` a un acorde

```
<c e g c>\arpeggio
```



Un corchete recto a la izquierda indica que el intérprete no tiene que arpeggiar el acorde

```
\arpeggioBracket  
<c' e g c>\arpeggio
```



La dirección del arpeggio se denota a veces mediante la adición de una punta de flecha a la línea ondulada

```
\new Voice {
  \arpeggioUp
  <c e g c>\arpeggio
  \arpeggioDown
  <c e g c>\arpeggio
}
```



Propiedades trucadas frecuentemente

Cuando un arpeggio cruza uno o varios pentagramas, puede adjuntar un arpeggio a los acordes en los dos pentagramas y establecer `PianoStaff.connectArpeggios`

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff { <c' e g c>\arpeggio }
  \new Staff { \clef bass <c,, e g>\arpeggio }
>>
```



Comandos predefinidos

`\arpeggio`, `\arpeggioUp`, `\arpeggioDown`, `\arpeggioNeutral`, `\arpeggioBracket`.

Véase también

Manual de notación: [Sección 6.5.1 \[Ligaduras de unión\]](#), [página 91](#), para escribir arpeggios explícitamente.

Referencia del programa: `Arpeggio`.

Fallos

No es posible mezclar arpeggios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

6.6.8 Caídas y elevaciones

Se pueden expresar caídas y subidas de tono (falls y doits) añadidas a las notas mediante el comando `\bendAfter`,



6.7 Repeticiones

La repetición es un concepto fundamental en música, y existen varios tipos de notación para las repeticiones.

6.7.1 Tipos de repetición

Están soportados los siguientes tipos de repetición

desplegada

Las música repetida está completamente escrita (e interpretada). Es útil cuando se introduce música repetitiva. Éste es el único tipo de repetición que se incluye en la salida MIDI.

volta (primera y segunda vez)

Las repeticiones no están desarrolladas, pero se imprimen finales alternativos (volte) de izquierda a derecha con corchetes. Ésta es la notación estándar para las repeticiones con alternativas. Por omisión no se reproducen en la salida MIDI.

trémolo Hacen barras de trémolo. Por omisión no se reproducen en la salida MIDI.

porcentaje

Hacen repeticiones de compases o parte de ellos. Tienen un aspecto semejante a un signo de porcentaje. Por omisión no se reproducen en la salida MIDI. Las repeticiones de porcentaje se deben declarar dentro de un contexto *Voice*.

6.7.2 Sintaxis de las repeticiones

LilyPond tiene una construcción sintáctica para especificar distintos tipos de repetición. La sintaxis es

```
\repeat variante número_de_repeticiones cuerpo_de_la_repetición
```

Si tiene finales alternativos, puede añadir

```
\alternative {
  alternativa1
  alternativa2
  alternativa3
  ...
}
```

donde cada *alternativa* es una expresión musical. Si no da un número de alternativas suficiente para todas las repeticiones, se supondrá que la primera alternativa se interpreta más de una vez.

Las repeticiones estándar se usan de esta forma:

```
c1
\repeat volta 2 { c4 d e f }
\repeat volta 2 { f e d c }
```



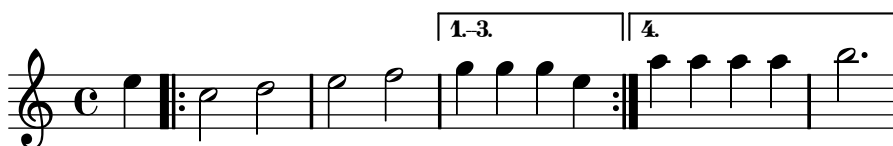
Con finales alternativos

```
c1
\repeat volta 2 {c4 d e f}
\alternative { {d2 d} {f f,} }
```



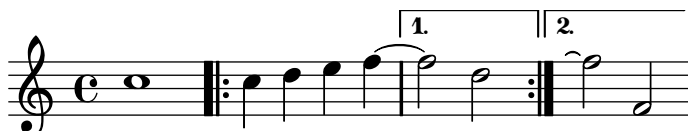
Se pueden crear repeticiones con anacrusa.

```
\new Staff {
  \partial 4 e |
  \repeat volta 4 { c2 d2 | e2 f2 | }
  \alternative { { g4 g g e } { a a a a | b2. } }
}
```



Se pueden añadir ligaduras de unión al segundo final,

```
c1
\repeat volta 2 {c4 d e f ~ }
\alternative { {f2 d} {f\repeatTie f,} }
```



Es posible acortar los corchetes de primera y segunda vez estableciendo `voltaSpannerDuration`. En el ejemplo siguiente, el corchete sólo dura un compás, lo que supone una duración de 3/4.

```
\relative c''{
  \time 3/4
  c c c
  \set Staff.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat "volta" 5 { d d d }
  \alternative { { e e e f f f }
    { g g g } }
}
```



Si quiere comenzar una repetición al principio de una línea y tiene una doble barra al final de la línea anterior, utilice

```
... \bar "||:" \break
\repeat volta 2 { ...
```

consulte [Sección 6.4.5 \[Barras de compás\]](#), página 85 para ver más información.

Véase también

Ejemplos:

Normalmente los corchetes para la repetición sólo se imprimen sobre el pentagrama superior. Esto se puede ajustar moviendo `Volta_engraver` al contexto `Staff` donde quiera que aparezcan los corchetes; consulte [Sección 9.2.4 \[Modificar los complementos \(plug-ins\) de contexto\]](#), página 131 y

```
'repeats/volta-multi-staff.ly'.
```

Fallos

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.

La información del contador de tiempo no se recuerda al comienzo de una alternativa, por ello después de una repetición la información de la cuenta de tiempo se debe reajustar manualmente; por ejemplo, estableciendo el valor de `Score.measurePosition` o introduciendo `\partial`. De forma similar, las ligaduras de expresión o de unión tampoco se repiten.

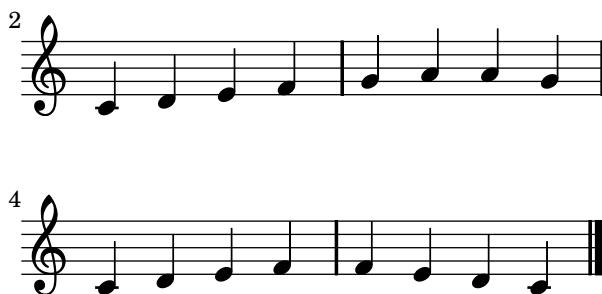
Los corchetes de primera y segunda vez no se alinean verticalmente.

6.7.3 Las repeticiones y el MIDI

Con un poco de trucaje, todos los tipos de repeticiones pueden estar presentes en la salida MIDI. Se consigue por medio de la aplicación de la función musical `\unfoldRepeats`. Esta función transforma todas las repeticiones en repeticiones desarrolladas o 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 "||."
```





Cuando se elabora un archivo de partitura que utiliza `\unfoldRepeats` para el MIDI, es necesario hacer dos bloques `\score`: uno para el MIDI (con las repeticiones desplegadas) y otro para la notación (con primera y segunda vez, trémolo y repeticiones de porcentaje). Por ejemplo,

```
\score {
  ..música..
  \layout { .. }
}
\score {
  \unfoldRepeats ..música..
  \midi { .. }
}
```

6.7.4 Comandos de repetición manual

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme de comandos de repetición.

`start-repeat`

Imprimir una barra de compás |:

`end-repeat`

Imprimir una barra de compás :|

`(volta texto)`

Imprimir un corchete de volta que diga *texto*: El texto se puede especificar como una cadena de texto o como un texto de marcado, consulte [Sección 8.1.5 \[Marcado del texto\]](#), [página 119](#). No olvide cambiar la fuente tipográfica, porque el tipo por defecto para los números no contiene caracteres alfabéticos;

`(volta #f)`

Detener un corchete de volta mantenido.

```
c4
  \set Score.repeatCommands = #'((volta "93") end-repeat)
c4 c4
  \set Score.repeatCommands = #'((volta #f))
c4 c4
```



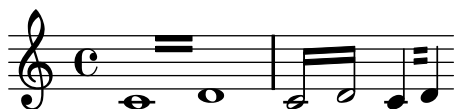
Véase también

Referencia del programa: `VoltaBracket`, `RepeatedMusic`, `VoltaRepeatedMusic` y `UnfoldedRepeatedMusic`.

6.7.5 Repeticiones de trémolo

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo`

```
\new Voice \relative c' {
  \repeat "tremolo" 8 { c16 d16 }
  \repeat "tremolo" 4 { c16 d16 }
  \repeat "tremolo" 2 { c16 d16 }
}
```



Las marcas de trémolo también se pueden poner sobre una nota única. En este caso, la nota no debe estar encerrada entre llaves.

```
\repeat "tremolo" 4 c'16
```



Se obtiene una salida similar usando la subdivisión del trémolo, que se describe en [Sección 6.7.6 \[Subdivisiones del trémolo\]](#), página 113.

Véase también

En el presente manual: [Sección 6.7.6 \[Subdivisiones del trémolo\]](#), página 113, [Sección 6.7 \[Repeticiones\]](#), página 109.

Referencia del programa: `Beam`, `StemTremolo`.

6.7.6 Subdivisiones del trémolo

Las marcas de trémolo se pueden imprimir sobre una nota única escribiendo ‘:*[número]*’ después de la nota. El número indica la duración de la subdivisión, y debe ser al menos 8. Un valor de *longitud* de 8 produce una línea tachando la plica de la nota. Si se omite la longitud, se utiliza el último valor (almacenado en `tremoloFlags`)

```
c'2:8 c':32 | c': c': |
```



Fallos

Los trémolos escritos de esta forma no se trasladan a la salida MIDI.

Véase también

En el presente manual: [Sección 6.7.5 \[Repeticiones de trémolo\]](#), página 113.

En otros lugares: `StemTremolo`.

6.7.7 Repeticiones de compás

En el estilo `percent` (porcentaje), se puede hacer que se repita un patrón de notas. Se imprime una sola vez, y luego el patrón se sustituye con un símbolo especial. Los patrones de uno y dos compases se sustituyen por símbolos parecidos a un porcentaje, y los patrones que dividen a la duración de un compás se sustituyen por barras inclinadas. Las repeticiones de porcentaje se deben declarar dentro de un contexto `Voice`.

```
\new Voice \relative c' {
  \repeat "percent" 4 { c4 }
  \repeat "percent" 2 { c2 es2 f4 fis4 g4 c4 }
}
```



Las repeticiones de compás de más de 2 compases llevan un contador, si se activa la propiedad `countPercentRepeats`,

```
\new Voice {
  \set countPercentRepeats = ##t
  \repeat "percent" 4 { c1 }
}
```



También se pueden imprimir porcentajes aislados. Esto se hace mediante la colocación de un silencio multicompañ con una función de impresión distinta,

```
\override MultiMeasureRest #'stencil
= #ly:multi-measure-rest::percent
R1
```



Véase también

Referencia del programa: `RepeatSlash`, `PercentRepeat`, `DoublePercentRepeat`, `DoublePercentRepeatCounter`, `PercentRepeatCounter`, `PercentRepeatedMusic`.

7 Notación específica de instrumentos

This section has not been translated yet; please refer to the manual in English.

7.1 Música de piano

This section has not been translated yet; please refer to the manual in English.

7.1.1 Cambios de pentagrama automáticos

This section has not been translated yet; please refer to the manual in English.

7.1.2 Cambios de pentagrama manuales

This section has not been translated yet; please refer to the manual in English.

7.1.3 Pedales

This section has not been translated yet; please refer to the manual in English.

7.1.4 Líneas de cambio de pentagrama

This section has not been translated yet; please refer to the manual in English.

7.1.5 Plicas de pentagrama cruzado

This section has not been translated yet; please refer to the manual in English.

7.2 Nombres de acorde

This section has not been translated yet; please refer to the manual in English.

7.2.1 Escritura de los nombres de acorde

This section has not been translated yet; please refer to the manual in English.

7.2.2 Modo de acordes

This section has not been translated yet; please refer to the manual in English.

7.2.3 Impresión de los nombres de acorde

This section has not been translated yet; please refer to the manual in English.

7.3 Música vocal

This section has not been translated yet; please refer to the manual in English.

7.3.1 Elaborar canciones sencillas

This section has not been translated yet; please refer to the manual in English.

7.3.2 Introducir la letra

This section has not been translated yet; please refer to the manual in English.

7.3.3 Hyphens and extenders

This section has not been translated yet; please refer to the manual in English.

7.3.4 The Lyrics context

This section has not been translated yet; please refer to the manual in English.

7.3.5 Melismata

This section has not been translated yet; please refer to the manual in English.

7.3.6 Otra manera de introducir la letra

This section has not been translated yet; please refer to the manual in English.

7.3.7 Flexibilidad en la colocación

This section has not been translated yet; please refer to the manual in English.

7.3.7.1 Una sílaba de la letra sobre varias notas de un melisma

This section has not been translated yet; please refer to the manual in English.

7.3.7.2 Letras en divisi

This section has not been translated yet; please refer to the manual in English.

7.3.7.3 Cambiar la melodía dependiendo de la letra

This section has not been translated yet; please refer to the manual in English.

7.3.7.4 Specifying melismata within the lyrics

This section has not been translated yet; please refer to the manual in English.

7.3.7.5 Letra independiente de las notas

This section has not been translated yet; please refer to the manual in English.

7.3.8 Espaciar las sílabas de la letra

This section has not been translated yet; please refer to the manual in English.

7.3.9 Más acerca de los versos

This section has not been translated yet; please refer to the manual in English.

7.3.9.1 Adding stanza numbers

7.3.9.2 Añadir expresiones dinámicas

7.3.9.3 Añadir el nombre de los cantantes

7.3.9.4 Printing stanzas at the end

7.3.9.5 Printing stanzas at the end in multiple columns

7.3.10 Tesitura

This section has not been translated yet; please refer to the manual in English.

7.3.11 Otros temas relacionados con la música vocal

This section has not been translated yet; please refer to the manual in English.

7.4 Música rítmica

This section has not been translated yet; please refer to the manual in English.

7.4.1 Mostrar los ritmos de la melodía

This section has not been translated yet; please refer to the manual in English.

7.4.2 Introducir la percusión

This section has not been translated yet; please refer to the manual in English.

7.4.3 Pautas de percusión

This section has not been translated yet; please refer to the manual in English.

7.4.4 Notas fantasma

This section has not been translated yet; please refer to the manual in English.

7.5 Guitarra

This section has not been translated yet; please refer to the manual in English.

7.5.1 Indicación de los números de cuerda

This section has not been translated yet; please refer to the manual in English.

7.5.2 Fundamentos de las tablaturas

This section has not been translated yet; please refer to the manual in English.

7.5.3 Tablaturas para otros instrumentos

This section has not been translated yet; please refer to the manual in English.

7.5.4 Tablaturas de banjo

This section has not been translated yet; please refer to the manual in English.

7.5.5 Diagramas de trastes

This section has not been translated yet; please refer to the manual in English.

7.5.6 Digitaciones de la mano derecha

This section has not been translated yet; please refer to the manual in English.

7.5.7 Otros temas relacionados con la música de guitarra

This section has not been translated yet; please refer to the manual in English.

7.6 Gaita

This section has not been translated yet; please refer to the manual in English.

7.6.1 Definiciones para la gaita

This section has not been translated yet; please refer to the manual in English.

7.6.2 Ejemplo de música de gaita

This section has not been translated yet; please refer to the manual in English.

7.7 Notación antigua

This section has not been translated yet; please refer to the manual in English.

7.7.1 Cabezas de nota antiguas

This section has not been translated yet; please refer to the manual in English.

7.7.2 Alteraciones accidentales antiguos

This section has not been translated yet; please refer to the manual in English.

7.7.3 Silencios antiguos

This section has not been translated yet; please refer to the manual in English.

7.7.4 Claves antiguas

This section has not been translated yet; please refer to the manual in English.

7.7.5 Corchetes antiguos

This section has not been translated yet; please refer to the manual in English.

7.7.6 Indicaciones antiguas de compás

This section has not been translated yet; please refer to the manual in English.

7.7.7 Articulaciones antiguas

This section has not been translated yet; please refer to the manual in English.

7.7.8 Custos

This section has not been translated yet; please refer to the manual in English.

7.7.9 Divisiones

This section has not been translated yet; please refer to the manual in English.

7.7.10 Ligaduras

This section has not been translated yet; please refer to the manual in English.

7.7.10.1 Ligaduras mensurales blancas

This section has not been translated yet; please refer to the manual in English.

7.7.10.2 Ligaduras de neumas cuadrados gregorianos

This section has not been translated yet; please refer to the manual in English.

7.7.11 Contextos del Canto Gregoriano

This section has not been translated yet; please refer to the manual in English.

7.7.12 Contextos de la música mensural

This section has not been translated yet; please refer to the manual in English.

7.7.13 Alteraciones de la Musica Ficta

This section has not been translated yet; please refer to the manual in English.

7.7.14 Bajo figurado

This section has not been translated yet; please refer to the manual in English.

7.8 Notación específica de otros instrumentos

This section has not been translated yet; please refer to the manual in English.

7.8.1 Armónicos artificiales de los instrumentos de cuerda

This section has not been translated yet; please refer to the manual in English.

8 Notación avanzada

This section has not been translated yet; please refer to the manual in English.

8.1 Texto

This section has not been translated yet; please refer to the manual in English.

8.1.1 Guiones de texto

This section has not been translated yet; please refer to the manual in English.

8.1.2 Extensiones de texto y de línea

This section has not been translated yet; please refer to the manual in English.

8.1.3 Extensiones de texto

This section has not been translated yet; please refer to the manual in English.

8.1.4 Indicaciones de texto

This section has not been translated yet; please refer to the manual in English.

8.1.5 Marcado del texto

This section has not been translated yet; please refer to the manual in English.

8.1.6 Partituras anidadas

This section has not been translated yet; please refer to the manual in English.

8.1.7 Page wrapping text

This section has not been translated yet; please refer to the manual in English.

8.1.8 Panorámica de los comandos de marcado de texto

This section has not been translated yet; please refer to the manual in English.

`\arrow-head` *axis* (integer) *direction* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

`\bigger` *arg* (markup)

Increase the font size relative to current setting.

`\bold` *arg* (markup)

Switch to bold font-series.

`\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.

`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

`\caps` *arg* (markup)

Emit *arg* as small caps.

- `\center-align` *args* (list of markups)
Put *args* in a centered column.
- `\char` *num* (integer)
Produce a single character. For example, `\char #65` produces the letter ‘A’.
- `\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.
- `\column` *args* (list of markups)
Stack the markups in *args* vertically. The property `baseline-skip` determines the space between each markup in *args*.
- `\combine` *m1* (markup) *m2* (markup)
Print two markups on top of each other.
- `\concat` *args* (list of markups)
Concatenate *args* in a horizontal line, without spaces inbetween. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.
- `\dir-column` *args* (list of markups)
Make a column of *args*, going up or down, depending on the setting of the `#'direction` layout property.
- `\doubleflat`
Draw a double flat symbol.
- `\doublesharp`
Draw a double sharp symbol.
- `\draw-circle` *radius* (number) *thickness* (number) *fill* (boolean)
A circle of radius *radius*, thickness *thickness* and optionally filled.
- `\draw-line` *dest* (pair of numbers)
A simple line. Uses the `thickness` property.
- `\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.
- `\epsfile` *axis* (number) *size* (number) *file-name* (string)
Inline an EPS image. The image is scaled along *axis* to *size*.
- `\fill-line` *markups* (list of markups)
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.
- `\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).
- `\finger` *arg* (markup)
Set the argument as small numbers.
- `\flat`
Draw a flat symbol.

`\fontCaps` *arg* (markup)

Set `font-shape` to `caps`.

`\fontsize` *increment* (number) *arg* (markup)

Add *increment* to the font-size. Adjust baseline skip accordingly.

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of two markups.

`\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 (in staff spaces). Default: 0.05.
 - `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.

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here 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.

`\fret-diagram-verbose` *marking-list* (list)

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*.

`(place-fret string-number fret-number finger-value)`

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*. By default, the fret playing indicator is a solid dot. This can be changed by setting the value of the variable *dot-color*. 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.

`\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.

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)

Align *arg* in *axis* direction to the *dir* side.

`\halign` *dir* (number) *arg* (markup)

Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values inbetween interpolate alignment accordingly.

`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

`\hcenter-in` *length* (number) *arg* (markup)

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

`\hcenter` *arg* (markup)

Align *arg* to its X center.

`\hspace` *amount* (number)

This produces an invisible object taking horizontal space. For example,

```
\markup { A \hspace #2.0 B }
```

puts extra space between A and B, on top of the space that is normally inserted before elements on a line.

`\huge` *arg* (markup)

Set font size to `+2`.

- `\italic arg` (markup)
Use italic **font-shape** for *arg*.
- `\justify-field symbol` (symbol)
Justify the data which has been assigned to *symbol*.
- `\justify args` (list of markups)
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.
- `\justify-string arg` (string)
Justify a string. Paragraphs may be separated with double newlines
- `\large arg` (markup)
Set font size to +1.
- `\larger arg` (markup)
Copy of the bigger-markup command.
- `\left-align arg` (markup)
Align *arg* on its left edge.
- `\line args` (list of markups)
Put *args* in a horizontal line. The property **word-space** determines the space between each markup in *args*.
- `\lookup glyph-name` (string)
Lookup a glyph by name.
- `\lower amount` (number) *arg* (markup)
Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.
- `\magnify sz` (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

$$A \text{ \magnify \#1.1 } \{ A \} A$$
Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.
- `\markalphabet num` (integer)
Make a markup letter for *num*. The letters start with A to Z and continue with double letters.
- `\markletter num` (integer)
Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.
- `\medium arg` (markup)
Switch to medium font series (in contrast to bold).
- `\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 user manual, **The Feta font** for a complete listing of the possible glyphs.
- `\natural`
Draw a natural symbol.
- `\normal-size-sub arg` (markup)
Set *arg* in subscript, in a normal font size.

`\normal-size-super` *arg* (markup)

Set *arg* in superscript with a normal font size.

`\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.

`\normalsize` *arg* (markup)

Set font size to default.

`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Construct a note symbol, with stem. By using fractional values for *dir*, you can obtain longer or shorter stems.

`\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.

`\null`

An empty markup with extents of a single point.

`\number` *arg* (markup)

Set font family to **number**, which yields the font used for time signatures and fingerings. This font only contains numbers and some punctuation. It doesn't have any letters.

`\on-the-fly` *procedure* (symbol) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the first argument in to the property list. Properties may be any sort of property supported by **font-interface** and **text-interface**, for example

```
\override #'(font-family . married) "bla"
```

`\pad-around` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

`\pad-markup` *padding* (number) *arg* (markup)

Add space around a markup object.

`\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.

`\pad-x` *amount* (number) *arg* (markup)

Add padding *amount* around *arg* in the X direction.

`\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.

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string. Due to technicalities of the output backends, different scales should be used for the T_EX and PostScript backend, selected with `-f`.

For the T_EX backend, the following string prints a rotated text

```
0 0 moveto /ecrm10 findfont
1.75 scalefont setfont 90 rotate (hello) show
```

The magical constant 1.75 scales from LilyPond units (staff spaces) to T_EX dimensions.

For the postscript backend, use the following

```
gsave /ecrm10 findfont
10.0 output-scale div
scalefont setfont 90 rotate (hello) show grestore
```

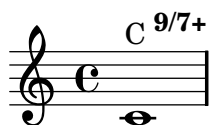
`\put-adjacent arg1 (markup) axis (integer) dir (direction) arg2 (markup)`

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`.

```
c1^\markup { C \small \raise #1.0 \bold { "9/7+" } }
```



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.

`\right-align arg (markup)`

Align *arg* on its right edge.

`\roman arg (markup)`

Set font family to `roman`.

`\rotate ang (number) arg (markup)`

Rotate object with *ang* degrees around its center.

`\sans arg (markup)`

Switch to the sans serif family.

`\score score (unknown)`

Inline an image of music.

`\semiflat`

Draw a semiflat.

`\semisharp`

Draw a semi sharp symbol.

`\sesquiflat`

Draw a 3/2 flat symbol.

`\sesquisharp`

Draw a 3/2 sharp symbol.

`\sharp`

Draw a sharp symbol.

`\simple str` (string)

A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

`\slashed-digit num` (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

`\small arg` (markup)

Set font size to -1.

`\smallCaps text` (markup)

Turn `text`, which should be a string, to small caps.

`\markup \smallCaps "Text between double quotes"`

`\smaller arg` (markup)

Decrease the font size relative to current setting.

`\stencil stil` (unknown)

Use a stencil as markup.

`\strut`

Create a box of the same height as the space in the current font.

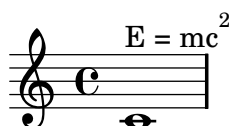
`\sub arg` (markup)

Set `arg` in subscript.

`\super arg` (markup)

Raising and lowering texts can be done with `\super` and `\sub`:

`c1~\markup { E "=" \concat { "mc" \super "2" } }`



`\teeny arg` (markup)

Set font size to -3.

`\text arg` (markup)

Use a text font instead of music symbol or music alphabet font.

`\tied-lyric str` (string)

Like simple-markup, but use tie characters for ‘~’ tilde symbols.

`\tiny arg` (markup)

Set font size to -2.

`\translate offset` (pair of numbers) `arg` (markup)

This translates an object. Its first argument is a cons of numbers.

`A \translate #(cons 2 -3) { B C } D`

This moves ‘B C’ 2 spaces to the right, and 3 down, relative to its surroundings. This command cannot be used to move isolated scripts vertically, for the same reason that `\raise` cannot be used for that.

`\translate-scaled offset` (pair of numbers) `arg` (markup)

Translate `arg` by `offset`, scaling the offset by the `font-size`.

- `\transparent` *arg* (markup)
Make the argument transparent.
- `\triangle` *filled* (boolean)
A triangle, either filled or empty.
- `\typewriter` *arg* (markup)
Use `font-family typewriter` for *arg*.
- `\upright` *arg* (markup)
Set font shape to `upright`. This is the opposite of `italic`.
- `\vcenter` *arg* (markup)
Align *arg* to its Y center.
- `\verbatim-file` *name* (string)
Read the contents of a file, and include it verbatim.
- `\whiteout` *arg* (markup)
Provide a white underground for *arg*.
- `\with-color` *color* (list) *arg* (markup)
Draw *arg* in color specified by *color*.
- `\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.
- `\with-url` *url* (string) *arg* (markup)
Add a link to URL *url* around *arg*. This only works in the PDF backend.
- `\wordwrap-field` *symbol* (symbol)
Wordwrap the data which has been assigned to *symbol*.
- `\wordwrap` *args* (list of markups)
Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.
- `\wordwrap-string` *arg* (string)
Wordwrap a string. Paragraphs may be separated with double newlines.

8.1.9 Overview of text markup list commands

This section has not been translated yet; please refer to the manual in English.

- `\column-lines` *args* (list of markups)
Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.
- `\justified-lines` *args* (list of markups)
Like `\justify`, but return a list of lines instead of a single markup. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.
- `\override-lines` *new-prop* (pair) *args* (list of markups)
Like `\override`, for markup lists.
- `\wordwrap-lines` *args* (list of markups)
Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

8.1.10 Selección de la tipografía

This section has not been translated yet; please refer to the manual in English.

8.1.11 Indicaciones dinámicas contemporáneas

This section has not been translated yet; please refer to the manual in English.

8.2 Preparación de particellas

This section has not been translated yet; please refer to the manual in English.

8.2.1 Silencios multicompás

This section has not been translated yet; please refer to the manual in English.

8.2.2 Indicaciones metronómicas

This section has not been translated yet; please refer to the manual in English.

8.2.3 Llamadas de ensayo

This section has not been translated yet; please refer to the manual in English.

8.2.4 Numeración de compases

This section has not been translated yet; please refer to the manual in English.

8.2.5 Nombres de instrumentos

This section has not been translated yet; please refer to the manual in English.

8.2.6 Transposición de los instrumentos

This section has not been translated yet; please refer to the manual in English.

8.2.7 Corchetes de octava

This section has not been translated yet; please refer to the manual in English.

8.2.8 Distintas ediciones a partir de una misma fuente

This section has not been translated yet; please refer to the manual in English.

8.3 Música orquestal

This section has not been translated yet; please refer to the manual in English.

8.3.1 Combinación automática de las partes

This section has not been translated yet; please refer to the manual in English.

8.3.2 Ocultar pentagramas

This section has not been translated yet; please refer to the manual in English.

8.3.3 Citar otras voces

This section has not been translated yet; please refer to the manual in English.

8.3.4 Formateo de las notas de aviso

This section has not been translated yet; please refer to the manual in English.

8.3.5 Alinear con una cadenza

This section has not been translated yet; please refer to the manual in English.

8.4 Notación contemporánea

This section has not been translated yet; please refer to the manual in English.

8.4.1 Notación polimétrica

This section has not been translated yet; please refer to the manual in English.

8.4.2 Gestión del tiempo

This section has not been translated yet; please refer to the manual in English.

8.4.3 Notación proporcional (introducción)

This section has not been translated yet; please refer to the manual in English.

8.4.4 Racimos (clusters)

This section has not been translated yet; please refer to the manual in English.

8.4.5 Cabezas de nota especiales

This section has not been translated yet; please refer to the manual in English.

8.4.6 Barras punteadas

This section has not been translated yet; please refer to the manual in English.

8.4.7 Improvisación

This section has not been translated yet; please refer to the manual in English.

8.4.8 Seleccionar el tamaño de la tipografía para la notación

This section has not been translated yet; please refer to the manual in English.

8.5 Utilización educativa

This section has not been translated yet; please refer to the manual in English.

8.5.1 Globos de ayuda

This section has not been translated yet; please refer to the manual in English.

8.5.2 Papel de música en blanco

This section has not been translated yet; please refer to the manual in English.

8.5.3 Notas ocultas

This section has not been translated yet; please refer to the manual in English.

8.5.4 Cabezas de notas con formas diversas

This section has not been translated yet; please refer to the manual in English.

8.5.5 Cabezas de notas de Notación Fácil

This section has not been translated yet; please refer to the manual in English.

8.5.6 Corchetes de análisis

This section has not been translated yet; please refer to the manual in English.

8.5.7 Colorear objetos

This section has not been translated yet; please refer to the manual in English.

8.5.8 Paréntesis

This section has not been translated yet; please refer to the manual in English.

8.5.9 Líneas de rejilla

This section has not been translated yet; please refer to the manual in English.

9 Cambiar los valores por omisión

This section has not been translated yet; please refer to the manual in English.

9.1 Notación automática

This section has not been translated yet; please refer to the manual in English.

9.1.1 Alteraciones accidentales automáticas

This section has not been translated yet; please refer to the manual in English.

9.1.2 Establecer el comportamiento de las barras automáticas

This section has not been translated yet; please refer to the manual in English.

9.2 Contextos de interpretación

This section has not been translated yet; please refer to the manual in English.

9.2.1 Explicación de los contextos

This section has not been translated yet; please refer to the manual in English.

9.2.2 Crear contextos

This section has not been translated yet; please refer to the manual in English.

9.2.3 Cambiar las propiedades de un contexto al vuelo

This section has not been translated yet; please refer to the manual in English.

9.2.4 Modificar los complementos (plug-ins) de contexto

This section has not been translated yet; please refer to the manual in English.

9.2.5 Ajuste fino de la presentación dentro de los contextos

This section has not been translated yet; please refer to the manual in English.

9.2.6 Cambiar los valores por omisión de los contextos

This section has not been translated yet; please refer to the manual in English.

9.2.7 Definir contextos nuevos

This section has not been translated yet; please refer to the manual in English.

9.2.8 Alinear los contextos

This section has not been translated yet; please refer to the manual in English.

9.2.9 Vertical grouping of grobs

This section has not been translated yet; please refer to the manual in English.

9.3 El comando `\override`

This section has not been translated yet; please refer to the manual in English.

9.3.1 Construir un truco

This section has not been translated yet; please refer to the manual in English.

9.3.2 Navegar por la referencia del programa

This section has not been translated yet; please refer to the manual in English.

9.3.3 Interfaces de la presentación

This section has not been translated yet; please refer to the manual in English.

9.3.4 Determinar la propiedad del grob

This section has not been translated yet; please refer to the manual in English.

9.3.5 Objetos conectados con la entrada

This section has not been translated yet; please refer to the manual in English.

9.3.6 Usar código de Scheme en lugar de `\tweak`

This section has not been translated yet; please refer to the manual in English.

9.3.7 `\set` frente a `\override`

This section has not been translated yet; please refer to the manual in English.

9.3.8 Trucos difíciles

This section has not been translated yet; please refer to the manual in English.

10 Notación no musical

This section has not been translated yet; please refer to the manual in English.

10.1 Archivos de entrada

This section has not been translated yet; please refer to the manual in English.

10.1.1 Estructura del archivo (introducción)

This section has not been translated yet; please refer to the manual in English.

10.1.2 Estructura del archivo

This section has not been translated yet; please refer to the manual in English.

10.1.3 Una expresión musical única

This section has not been translated yet; please refer to the manual in English.

10.1.4 Varias partituras en un libro

This section has not been translated yet; please refer to the manual in English.

10.1.5 Extraer fragmentos de notación

This section has not been translated yet; please refer to the manual in English.

10.1.6 Inclusión de archivos de LilyPond

This section has not been translated yet; please refer to the manual in English.

10.1.7 Codificación del texto

This section has not been translated yet; please refer to the manual in English.

10.2 Títulos y cabeceras

This section has not been translated yet; please refer to the manual in English.

10.2.1 Crear títulos

This section has not been translated yet; please refer to the manual in English.

10.2.2 Títulos personalizados

This section has not been translated yet; please refer to the manual in English.

10.2.3 Reference to page numbers

This section has not been translated yet; please refer to the manual in English.

10.2.4 Table of contents

This section has not been translated yet; please refer to the manual in English.

10.3 Salida MIDI

This section has not been translated yet; please refer to the manual in English.

10.3.1 Crear archivos MIDI

This section has not been translated yet; please refer to the manual in English.

10.3.2 El bloque MIDI

This section has not been translated yet; please refer to the manual in English.

10.3.3 Nombres de los instrumentos MIDI

This section has not been translated yet; please refer to the manual in English.

10.4 Presentar la notación de LilyPond

This section has not been translated yet; please refer to the manual in English.

10.5 Saltar la música corregida

This section has not been translated yet; please refer to the manual in English.

11 Problemas de espaciado

This section has not been translated yet; please refer to the manual in English.

11.1 El papel y las páginas

This section has not been translated yet; please refer to the manual in English.

11.1.1 Tamaño del papel

This section has not been translated yet; please refer to the manual in English.

11.1.2 Formateo de las páginas

This section has not been translated yet; please refer to the manual in English.

11.2 Disposición de la música

This section has not been translated yet; please refer to the manual in English.

11.2.1 Establecer el tamaño del pentagrama

This section has not been translated yet; please refer to the manual in English.

11.2.2 Disposición de la partitura

This section has not been translated yet; please refer to the manual in English.

11.3 Mostrar el espaciado

This section has not been translated yet; please refer to the manual in English.

11.4 Saltos

This section has not been translated yet; please refer to the manual in English.

11.4.1 Saltos de línea

This section has not been translated yet; please refer to the manual in English.

11.4.2 Saltos de página

This section has not been translated yet; please refer to the manual in English.

11.4.3 Saltos de página óptimos

This section has not been translated yet; please refer to the manual in English.

11.4.4 Paso de página óptimo

This section has not been translated yet; please refer to the manual in English.

11.4.5 Saltos de línea explícitos

This section has not been translated yet; please refer to the manual in English.

11.4.6 Utilizar una voz adicional para los saltos de línea

This section has not been translated yet; please refer to the manual in English.

11.5 Espaciado vertical

This section has not been translated yet; please refer to the manual in English.

11.5.1 Especiado vertical dentro de un sistema

This section has not been translated yet; please refer to the manual in English.

11.5.2 Espaciado vertical entre sistemas

This section has not been translated yet; please refer to the manual in English.

11.5.3 Posicionamiento explícito de los pentagramas y los sistemas

This section has not been translated yet; please refer to the manual in English.

11.5.4 Espaciado vertical en dos fases

This section has not been translated yet; please refer to the manual in English.

11.5.5 Evitar las colisiones verticales

This section has not been translated yet; please refer to the manual in English.

11.6 Espaciado horizontal

This section has not been translated yet; please refer to the manual in English.

11.6.1 Panorámica del espaciado horizontal

This section has not been translated yet; please refer to the manual in English.

11.6.2 Área de espaciado nueva

This section has not been translated yet; please refer to the manual in English.

11.6.3 Cambiar el espaciado horizontal

This section has not been translated yet; please refer to the manual in English.

11.6.4 Longitud de la línea

This section has not been translated yet; please refer to the manual in English.

11.6.5 Notación proporcional

This section has not been translated yet; please refer to the manual in English.

12 Interfaces para programadores

This section has not been translated yet; please refer to the manual in English.

12.1 Funciones musicales

This section has not been translated yet; please refer to the manual in English.

12.1.1 Panorámica de las funciones musicales

This section has not been translated yet; please refer to the manual in English.

12.1.2 Funciones de sustitución sencillas

This section has not been translated yet; please refer to the manual in English.

12.1.3 Funciones de sustitución en parejas

This section has not been translated yet; please refer to the manual in English.

12.1.4 Matemáticas dentro de las funciones

This section has not been translated yet; please refer to the manual in English.

12.1.5 Funciones vacías

This section has not been translated yet; please refer to the manual in English.

12.1.6 Funciones sin argumentos

This section has not been translated yet; please refer to the manual in English.

12.1.7 Panorámica de las funciones musicales disponibles

This section has not been translated yet; please refer to the manual in English.

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)
(undocumented; fixme)

`displayLilyMusic` - *music* (music)
(undocumented; fixme)

`tocItem` - *text* (markup)
Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`appoggiatura` - *music* (music)
(undocumented; fixme)

`cueDuring` - *what* (string) *dir* (direction) *main-music* (music)
(undocumented; fixme)

`removeWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`breathe` -
(undocumented; fixme)

`clef` - *type* (string)
(undocumented; fixme)

`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)
(undocumented; fixme)

withMusicProperty - *sym* (symbol) *val* (any type) *music* (music)
 (undocumented; fixme)

assertBeamSlope - *comp* (procedure)
 (undocumented; fixme)

acciaccatura - *music* (music)
 (undocumented; fixme)

pitchedTrill - *main-note* (music) *secondary-note* (music)
 (undocumented; fixme)

applyContext - *proc* (procedure)
 (undocumented; fixme)

allowPageTurn -
 (undocumented; fixme)

assertBeamQuant - *l* (pair) *r* (pair)
 (undocumented; fixme)

includePageLayoutFile -
 (undocumented; fixme)

transposition - *pitch-note* (music)
 (undocumented; fixme)

applyOutput - *ctx* (symbol) *proc* (procedure)
 (undocumented; fixme)

afterGrace - *main* (music) *grace* (music)
 (undocumented; fixme)

label - *label* (symbol)
 (undocumented; fixme)

oldaddyrics - *music* (music) *lyrics* (music)
 (undocumented; fixme)

pageTurn -
 (undocumented; fixme)

compressMusic - *fraction* (pair of numbers) *music* (music)
 (undocumented; fixme)

featherDurations - *factor* (moment) *argument* (music)
 (undocumented; fixme)

displayMusic - *music* (music)
 (undocumented; fixme)

parallelMusic - *voice-ids* (list) *music* (music)
 (undocumented; fixme)

resetRelativeOctave - *reference-note* (music)
 (undocumented; fixme)

addQuote - *name* (string) *music* (music)
 (undocumented; fixme)

octave - *pitch-note* (music)
 (undocumented; fixme)

`parenthesize` - *arg* (music)
 (undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
 (undocumented; fixme)

`instrumentSwitch` - *name* (string)
 (undocumented; fixme)

`makeClusters` - *arg* (music)
 (undocumented; fixme)

`spacingTweaks` - *parameters* (list)
 (undocumented; fixme)

`tag` - *tag* (symbol) *arg* (music)
 (undocumented; fixme)

`noPageTurn` -
 (undocumented; fixme)

`bendAfter` - *delta* (integer)
 (undocumented; fixme)

`partcombine` - *part1* (music) *part2* (music)
 (undocumented; fixme)

`grace` - *music* (music)
 (undocumented; fixme)

`noPageBreak` -
 (undocumented; fixme)

`pageBreak` -
 (undocumented; fixme)

`bar` - *type* (string)
 (undocumented; fixme)

`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)
 (undocumented; fixme)

`unfoldRepeats` - *music* (music)
 (undocumented; fixme)

`balloonText` - *offset* (pair of numbers) *text* (markup)
 (undocumented; fixme)

`quoteDuring` - *what* (string) *main-music* (music)
 (undocumented; fixme)

`barNumberCheck` - *n* (integer)
 (undocumented; fixme)

`addInstrumentDefinition` - *name* (string) *lst* (list)
 (undocumented; fixme)

`scoreTweak` - *name* (string)
 (undocumented; fixme)

`autochange` - *music* (music)
 (undocumented; fixme)

`rightHandFinger` - *finger* (number or string)
(undocumented; fixme)

`endSpanners` - *music* (music)
(undocumented; fixme)

`musicMap` - *proc* (procedure) *mus* (music)
(undocumented; fixme)

`applyMusic` - *func* (procedure) *music* (music)
(undocumented; fixme)

`killCues` - *music* (music)
(undocumented; fixme)

`keepWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)
(undocumented; fixme)

12.2 Interfaces para el programador

This section has not been translated yet; please refer to the manual in English.

12.2.1 Variables de entrada y Scheme

This section has not been translated yet; please refer to the manual in English.

12.2.2 Representación interna de la música

This section has not been translated yet; please refer to the manual in English.

12.3 Construcción de funciones complejas

This section has not been translated yet; please refer to the manual in English.

12.3.1 Presentación de expresiones musicales

This section has not been translated yet; please refer to the manual in English.

12.3.2 Propiedades de la música

This section has not been translated yet; please refer to the manual in English.

12.3.3 Doblar una nota con ligaduras (ejemplo)

This section has not been translated yet; please refer to the manual in English.

12.3.4 Añadir articulación a las notas (ejemplo)

This section has not been translated yet; please refer to the manual in English.

12.4 Interfaz de marcado para el programador

This section has not been translated yet; please refer to the manual in English.

12.4.1 Construcción del marcado en Scheme

This section has not been translated yet; please refer to the manual in English.

12.4.2 Cómo funciona internamente el marcado

This section has not been translated yet; please refer to the manual in English.

12.4.3 Definición de un comando nuevo de marcado

This section has not been translated yet; please refer to the manual in English.

12.4.4 New markup list command definition

This section has not been translated yet; please refer to the manual in English.

12.5 Contextos para programadores

This section has not been translated yet; please refer to the manual in English.

12.5.1 Evaluación de contextos

This section has not been translated yet; please refer to the manual in English.

12.5.2 Ejecutar una función sobre todos los objetos de la presentación

This section has not been translated yet; please refer to the manual in English.

12.6 Procedimientos de Scheme como propiedades

This section has not been translated yet; please refer to the manual in English.

Apéndice A Lista bibliográfica

This section has not been translated yet; please refer to the manual in English.

Apéndice B Tutorial de Scheme

This section has not been translated yet; please refer to the manual in English.

Apéndice C Tablas del manual sobre notación

This section has not been translated yet; please refer to the manual in English.

C.1 Carta de nombres de acordes

This section has not been translated yet; please refer to the manual in English.

C.2 Instrumentos MIDI

This section has not been translated yet; please refer to the manual in English.

C.3 Lista de colores

This section has not been translated yet; please refer to the manual in English.

Colores normales

Nombres de los colores de X

Nombres de los colores sin sufijo numérico

Nombres de los colores con sufijo numérico

Escala de grises

C.4 La tipografía Feta

This section has not been translated yet; please refer to the manual in English.

C.5 Estilos de cabezas de nota

This section has not been translated yet; please refer to the manual in English.

Apéndice D Plantillas

This section has not been translated yet; please refer to the manual in English.

D.1 Pentagrama único

This section has not been translated yet; please refer to the manual in English.

D.1.1 Sólo notas

D.1.2 Notas y letra

D.1.3 Notas y acordes

D.1.4 Notas, letra y acordes

D.2 Plantillas de piano

This section has not been translated yet; please refer to the manual in English.

D.2.1 Piano solo

D.2.2 Piano y melodía con letra

D.2.3 Piano con letra centrada

D.2.4 Piano con dinámicas centradas

D.3 Cuarteto de cuerda

This section has not been translated yet; please refer to the manual in English.

D.3.1 Cuarteto de cuerda

D.3.2 Particellas de cuarteto de cuerda

D.4 Conjuntos vocales

This section has not been translated yet; please refer to the manual in English.

D.4.1 Partitura vocal SATB

D.4.2 Partitura vocal SATB y reducción para piano automática

D.4.3 SATB con contextos alineados

D.5 Plantillas para notación antigua

This section has not been translated yet; please refer to the manual in English.

D.5.1 Transcripción de música mensural

D.5.2 Plantilla para transcripción de canto gregoriano

D.6 Combo de jazz

This section has not been translated yet; please refer to the manual in English.




D.7 Plantillas de lilypond-book

This section has not been translated yet; please refer to the manual in English.

D.7.1 LaTeX

D.7.2 Texinfo

Apéndice E Hoja de referencia rápida

Sintaxis	Descripción	Ejemplo
<code>1 2 8 16</code>	duraciones	
<code>c4. c4..</code>	puntillos	
<code>c d e f g a b</code>	escala	
<code>fis bes</code>	alteración	
<code>\clef treble \clef bass</code>	claves	
<code>\time 3/4 \time 4/4</code>	compás	
<code>r4 r8</code>	silencio	
<code>d ~ d</code>	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\s fz`

dinámica



`a\< a a\!`

crescendo



`a\> a a\!`

decrescendo



`< >`

acorde



`\partial 8`

anacrusa



`\times 2/3 {f g a}`

tresillos



`\grace`

mordentes



`\lyricmode { twinkle }`

escribir la letra

twinkle

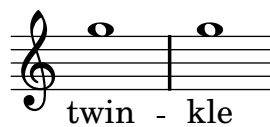
`\new Lyrics`

imprimir la letra

twinkle

`twin -- kle`

separador de sílabas



`\chordmode { c:dim f:maj7 }`

acordes



`\context ChordNames`

imprimir los nombres de los acordes

$C^{\circ} F^{\triangle}$

`<<\{e f\} \\\{c d\}>>`

polifonía



`s4 s8 s16`

silencios de separación

Apéndice F Licencia de documentación libre de GNU

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

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 written 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. Secondly, 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‘Document’, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‘you’.

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. (For example, 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.

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 ‘Transparent’ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

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 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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 less than five).
- 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', and 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. In any section entitled 'Acknowledgments' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments 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 as 'Endorsements' or to conflict in title with any Invariant Section.

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.

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 'Acknowledgments', 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an 'aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

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.

F.0.1 ADDENDUM: cómo utilizar esta licencia para sus documentos

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 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled `GNU
Free Documentation License'
```

If you have no Invariant Sections, write ‘with no Invariant Sections’ instead of saying which ones are invariant. If you have no Front-Cover Texts, write ‘no Front-Cover Texts’ instead of ‘Front-Cover Texts being *list*’; likewise for Back-Cover Texts.

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.

<code>\voiceFour</code>	80
<code>\voiceOne</code>	80
<code>\voiceThree</code>	80
<code>\voiceTwo</code>	80

<code> </code>	73
----------------------	----

<code>~</code>	91
----------------------	----

B

<code>barCheckSynchronize</code>	73
<code>breakable</code>	96

D

<code>defaultBarType</code>	86
-----------------------------------	----

P

<code>pipeSymbol</code>	73
-------------------------------	----

R

<code>r</code>	67
<code>repeatCommands</code>	86, 112

S

<code>s</code>	68
<code>stemLeftBeamCount</code>	96
<code>stemRightBeamCount</code>	96
<code>subdivideBeams</code>	96

T

<code>tremoloFlags</code>	113
<code>tupletNumberFormatFunction</code>	70

W

<code>whichBar</code>	86
-----------------------------	----

<code>\unfoldRepeats</code>	111
<code>\voiceFour</code>	80
<code>\voiceOne</code>	80
<code>\voiceThree</code>	80
<code>\voiceTwo</code>	80

.....	73
-------	----

~	91
---------	----

A

<code>acciaccatura</code>	23, 97
<code>acciaccatura</code>	138
<code>acento</code>	100
<code>acentos</code>	21
<code>acorde quebrado</code>	107
<code>acordes</code>	28, 30
<code>Acordes</code>	74
<code>acortar compases</code>	84
<code>addInstrumentDefinition</code>	139
<code>addQuote</code>	138
<code>afterGrace</code>	138
<code>ajustar la salida</code>	11
<code>ajuste del símbolo del pentagrama</code>	89
<code>al niente</code>	104
<code>allowPageTurn</code>	138
<code>alteración con paréntesis</code>	63
<code>alteración de precaución</code>	63
<code>alteración recordatoria</code>	63
<code>alturas</code>	61
<code>anacrusa</code>	23
<code>anacrusa</code>	84
<code>ángulo, barras en</code>	96
<code>apagado</code>	100
<code>applyContext</code>	138
<code>applyMusic</code>	140
<code>applyOutput</code>	138
<code>appoggiatura</code>	23, 97
<code>appoggiatura</code>	137
<code>Archivo PDF</code>	13
<code>arco abajo</code>	100
<code>arco arriba</code>	100
<code>Armadura de la tonalidad</code>	82
<code>Arpeggio</code>	107
<code>Arpeggio</code>	108
<code>arrow-head</code>	119
<code>arrow-head-markup</code>	119
<code>articulación</code>	21
<code>articulaciones</code>	21
<code>Articulaciones</code>	100
<code>assertBeamQuant</code>	138
<code>assertBeamSlope</code>	138
<code>auto-knee-gap</code>	96
<code>autochange</code>	139

B

<code>balloonGrobText</code>	139
<code>balloonText</code>	139

<code>bar</code>	139
<code>barCheckSynchronize</code>	73
<code>BarLine</code>	86
<code>barNumberCheck</code>	139
<code>barra</code>	22
<code>Barras de compás</code>	85
<code>barras de corchea y saltos de línea</code>	96
<code>barras de trémolo</code>	113
<code>barras en ángulo</code>	96
<code>barras manuales</code>	95
<code>barras, establecimiento manual</code>	22
<code>beam</code>	119
<code>Beam</code>	95
<code>Beam</code>	113
<code>beam-markup</code>	119
<code>bemol</code>	19
<code>bendAfter</code>	139
<code>bigger</code>	119
<code>bigger-markup</code>	119
<code>blanca</code>	15
<code>bold</code>	119
<code>bold-markup</code>	119
<code>box</code>	119
<code>box-markup</code>	119
<code>bracket</code>	119
<code>bracket-markup</code>	119
<code>breakable</code>	96
<code>breathe</code>	137
<code>BreathingSign</code>	106

C

<code>cabezas de nota, estilos de</code>	75
<code>cadencia</code>	87
<code>calderón</code>	100
<code>cambio de dedo</code>	102
<code>Canciones</code>	29
<code>capas</code>	75
<code>caps</code>	119
<code>caps-markup</code>	119
<code>center-align</code>	120
<code>center-align-markup</code>	120
<code>char</code>	120
<code>char-markup</code>	120
<code>ChoirStaff</code>	88
<code>circle</code>	120
<code>circle-markup</code>	120
<code>clave</code>	16
<code>clave de Do en cuarta (clave de tenor)</code>	81
<code>clave de Do en primera (clave de soprano)</code>	81
<code>clave de Do en quinta (clave de barítono)</code>	81
<code>clave de Do en segunda (clave de mezzosoprano)</code> ..	81
<code>clave de Do en tercera (clave de viola)</code>	81
<code>clave de Fa en cuarta (clave de bajo)</code>	81
<code>clave de Fa en quinta (clave de subbajo)</code>	81
<code>clave de Fa en tercera (clave de varbarítono)</code>	81
<code>clave de Sol</code>	81
<code>clave de Sol en primera (clave de violín francesa)</code> ..	81
<code>clave de tenor coral</code>	81
<code>clave de violín</code>	81
<code>clef</code>	137
<code>Clef</code>	82
<code>coda</code>	100
<code>column</code>	120

column-lines.....	127
column-lines-markup-list.....	127
column-markup.....	120
combine.....	120
combine-markup.....	120
comentario de bloque.....	17
comentario de línea.....	17
comentarios.....	17
comienzo del sistema.....	87
compás parcial.....	23, 84
Compás, comprobación de.....	73
compás, repeticiones de.....	114
Completion_heads_engraver.....	73, 74
compressMusic.....	138
Comprobación de octava.....	66
concat.....	120
concat-markup.....	120
Contexts.....	6
Converting from other formats.....	10
corchete vertical.....	87
corchetes de repetición de primera y segunda vez y ligaduras.....	92
crescendo.....	22, 104
cuarta.....	14
cuartos de tono.....	63
cueDuring.....	137

D

decrecendo.....	22, 104
defaultBarType.....	86
Desplazamiento.....	68
desplazamiento adicional.....	53
digitación.....	102
digitaciones.....	21
diminuendo.....	104
dir-column.....	120
dir-column-markup.....	120
displayLilyMusic.....	137
displayMusic.....	138
Distancias.....	57
Do central.....	14
doble bemol.....	19
doble sostenido.....	19
documentación interna.....	11
DotColumn.....	70
Dots.....	70
doubleflat.....	120
doubleflat-markup.....	120
DoublePercentRepeat.....	114
DoublePercentRepeatCounter.....	114
doublesharp.....	120
doublesharp-markup.....	120
draw-circle.....	120
draw-circle-markup.....	120
draw-line.....	120
draw-line-markup.....	120
duración.....	15
duración.....	69
dynamic.....	120
dynamic-markup.....	120
DynamicLineSpanner.....	55, 105
Dynamics.....	103
DynamicText.....	55, 105

E

Editor support.....	13
endSpanners.....	140
epsfile.....	120
epsfile-markup.....	120
equilibrio.....	2
escala.....	14
Escribir música en paralelo.....	90
espaciado óptico.....	2
espaciado regular.....	3
espressivo.....	100
expansión de repeticiones.....	111
expresión.....	25
expresión musical.....	25
expresiones dinámicas.....	22
extender lilypond.....	11
extra-offset (desplazamiento adicional).....	56

F

FDL, GNU Free Documentation License.....	151
featherDurations.....	138
fermata.....	100
figuras con puntillo.....	15
fill-line.....	120
fill-line-markup.....	120
filled-box.....	120
filled-box-markup.....	120
finger.....	120
finger-markup.....	120
Fingering.....	103
flageolet.....	100
flat.....	120
flat-markup.....	120
font-interface.....	124
fontCaps.....	121
fontCaps-markup.....	121
fontsize.....	121
fontsize-markup.....	121
Forbid_line_break_engraver.....	74
formateo de los grupos especiales.....	70
fraction.....	121
fraction-markup.....	121
frase idiomática.....	10
fraseo, ligaduras de.....	20
fret-diagram.....	121
fret-diagram-markup.....	121
fret-diagram-terse.....	121
fret-diagram-terse-markup.....	121
fret-diagram-verbose.....	122
fret-diagram-verbose-markup.....	122
fromproperty.....	122
fromproperty-markup.....	122

G

general-align.....	122
general-align-markup.....	122
Glissando.....	107
grabado.....	5
grace.....	139
GraceMusic.....	99
GrandStaff.....	87
grosor de las líneas del pentagrama, establecer....	90

grupeto circular	100
grupeto circular invertido	100
grupos especiales	23, 70
guiones	100

H

Hairpin	55, 104, 105
halign	122
halign-markup	122
hbracket	122
hbracket-markup	122
hcenter	122
hcenter-in	122
hcenter-in-markup	122
hcenter-markup	122
hoja guía de acordes	31
Hojas guía de acordes	30
hspace	122
hspace-markup	122
huge	122
huge-markup	122

I

identificadores	39, 44
idioma	10
idiomas extranjeros	10
idiomas, nombres de las notas en otros	64
includePageLayoutFile	138
indicación de compás	15
Indicación de compás	83
indicación de la versión	31
índice	11
instrumentSwitch	139
interioridades de lilypond	11
intervalo	14
italic	123
italic-markup	123

J

jerga	10
justified-lines	127
justified-lines-markup-list	127
justify	123
justify-field	123
justify-field-markup	123
justify-markup	123
justify-string	123
justify-string-markup	123

K

keepWithTag	140
KeyCancellation	83
KeySignature	83
killCues	140

L

label	138
Laissez vibrer	95
LaissezVibrerTie	95

LaissezVibrerTieColumn	95
large	123
large-markup	123
larger	123
larger-markup	123
LedgerLineSpanner	63
left-align	123
left-align-markup	123
Letra	29
ligadura de expresión	20
ligadura de unión	20, 21
ligadura de unión	91
ligaduras de expresión	20
Ligaduras de expresión	93
ligaduras de expresión frente a ligaduras de unión	21
ligaduras de fraseo	20
ligaduras de repetición	92
ligaduras de unión	20
Ligaduras laissez vibrer	95
ligaduras de fraseo	94
LilyPond-book	10
line	123
line-markup	123
línea de extensión de sílabas	29
líneas divisorias	85
llave vertical	87
lookup	123
lookup-markup	123
lower	123
lower-markup	123
lowering text	126
LSR	10

M

magnify	123
magnify-markup	123
makeClusters	139
marcas de fraseo	94
marcas de trémolo	113
marcato	100
markalphabet	123
markalphabet-markup	123
markletter	123
markletter-markup	123
Measure_grouping_engraver	84
MeasureGrouping	84
medium	123
medium-markup	123
melisma	29
metro, medida	83
modos eclesiásticos	82
mordentes	97
moving text	126
Música intercalada	90
musicglyph	123
musicglyph-markup	123
musicMap	140

N

natural	123
natural-markup	123

negra	15
negrura	2
niente, al	104
Nombres de las notas	61
nombres de las notas, holandés	62
nombres de las notas, por omisión	62
nombres de los acordes	30
noPageBreak	139
noPageTurn	139
normal-size-sub	123
normal-size-sub-markup	123
normal-size-super	124
normal-size-super-markup	124
normal-text	124
normal-text-markup	124
normalsize	124
normalsize-markup	124
Nota de espaciado	68
Notación de los pentagramas	80
notas de adorno	23
note	124
note-by-number	124
note-by-number-markup	124
note-markup	124
Note_heads_engraver	73
NoteCollision	78, 80
NoteColumn	80
NoteHead	63
null	124
null-markup	124
number	124
number-markup	124
número de líneas del pentagrama, establecer	90

O

objetos invisibles	56
objetos transparentes	56
octave	138
ocultar objetos	56
oldaddyrics	138
on-the-fly	124
on-the-fly-markup	124
open	100
órgano, indicaciones de pedal de	100
ornamentos	97, 100
ossia	89
override	124
override-lines	127
override-lines-markup-list	127
override-markup	124
overrideProperty	137

P

pad-around	124
pad-around-markup	124
pad-markup	124
pad-markup-markup	124
pad-to-box	124
pad-to-box-markup	124
pad-x	124
pad-x-markup	124
padding	52

page-ref	124
page-ref-markup	124
pageBreak	139
pageTurn	138
parallelMusic	138
parcial, compás	84
paréntesis, alteración con	63
parenthesize	139
partcombine	139
parte al alzar	84
pauta de coro	87
pedal, indicaciones de	100
pentagrama, establecer el grosor de las líneas del	90
pentagrama, establecer el número de líneas del	90
pentagramas de piano	87
pentagramas, grupo de	87
Pentagramas, varios	87
PercentRepeat	114
PercentRepeatCounter	114
PercentRepeatedMusic	114
PhrasingSlur	94
PianoStaff	108
pipeSymbol	73
pitchedTrill	138
polifonía	27
polyphony	75
porcentaje, repeticiones de	114
portato	100
postscript	124
postscript-markup	124
prall, down	100
prall, up	100
prallmordent	100
prallprall	100
precaución, alteración de	63
prima volta	109
propiedades	11
put-adjacent	125
put-adjacent-markup	125

Q

quitar objetos	56
quoteDuring	139

R

r	67
raise	125
raise-markup	125
raising text	126
recordatoria, alteración	63
redonda	15
Relativas, especificación de octavas	64
Relativo	64
relleno	55
removeWithTag	137
repeatCommands	86, 112
RepeatedMusic	112
RepeatSlash	114
repetición ambigua	111
repetición, barras de	85
repeticiones	109
resetRelativeOctave	138

Rest	68
RestCollision	80
right-align	125
right-align-markup	125
rightHandFinger	140
ritmos regulares	3
roman	125
roman-markup	125
rotate	125
rotate-markup	125
rotated text	124
Running LilyPond	10

S

s	68
salto de pentagrama, establecimiento manual	27
sans	125
sans-markup	125
Scheme	11
score	125
Score	84
score-markup	125
scoreTweak	139
Script	101
Script	102
seconda volta	109
segno	100
semi-bemoles, semi-sostenidos	63
semiflat	125
semiflat-markup	125
semisharp	125
semisharp-markup	125
semitrino	100
semitrino descendente	100
Sensible a las mayúsculas	12, 17
sesquiflat	125
sesquiflat-markup	125
sesquisharp	125
sesquisharp-markup	125
sharp	126
sharp-markup	126
shiftDurations	139
silencio	15
Silencio invisible	68
Silencios	67
símbolos musicales	2
simple	126
simple-markup	126
SkipMusic	69
slashed-digit	126
slashed-digit-markup	126
Slur	94
small	126
small-markup	126
smallCaps	126
smallCaps-markup	126
smaller	126
smaller-markup	126
snippets	10
sostenido	19
spacingTweaks	139
SpanBar	86
staccatissimo	100

staccato	21, 100
Staff	68
Staff	86
StaffGroup	87
StaffSymbol	90
Stem	74
stemLeftBeamCount	96
stemRightBeamCount	96
StemTremolo	113
stencil	126
stencil-markup	126
strut	126
strut-markup	126
sub	126
sub-markup	126
subdivideBeams	96
super	126
super-markup	126
SystemStartBar	88
SystemStartBrace	88
SystemStartBracket	88
systemStartDelimiter	88

T

tag	139
teeny	126
teeny-markup	126
tenuto	100
terminología	10
text	126
text-interface	124
text-markup	126
TextScript	101
The Feta font	123
thumb marking	100
Tie	93
tied-lyric	126
tied-lyric-markup	126
TimeScaledMusic	72
TimeSignature	84
timing, internal	97
Timing_translator	84
tiny	126
tiny-markup	126
tipografía	3, 5
tipografías	2
tocItem	137
tonalidad, armadura de la, establecer	19
translate	126
translate-markup	126
translate-scaled	126
translate-scaled-markup	126
translating text	126
transparent	127
transparent-markup	127
transposedCueDuring	137
TransposedMusic	67
Transposición	66
Transposición de alturas	66
transposition	138
tremoloFlags	113
tresillos	23, 70
triangle	127

triangle-markup	127
TrillSpanner	107
trino	100
Trinos con altura	106
Trucos, distancias	57
TupletBracket	72
TupletNumber	72
tupletNumberFormatFunction	70
tweak	140
typewriter	127
typewriter-markup	127

U

UnfoldedRepeatedMusic	112
unfoldRepeats	139
Updating files with convert-ly	31, 50
upright	127
upright-markup	127

V

varcoda	100
variables	11, 39, 44
varias voces	27
vcenter	127
vcenter-markup	127
Ver la música	13
verbatim-file	127

verbatim-file-markup	127
voces, más (en un solo pentagrama)	27
Voice	68
Voice	75, 76, 104
volta	109
VoltaBracket	112
VoltaRepeatedMusic	112
voz de pentagrama cruzado, establecimiento manual	27

W

whichBar	86
whiteout	127
whiteout-markup	127
with-color	127
with-color-markup	127
with-dimensions	127
with-dimensions-markup	127
with-url	127
with-url-markup	127
withMusicProperty	138
wordwrap	127
wordwrap-field	127
wordwrap-field-markup	127
wordwrap-lines	127
wordwrap-lines-markup-list	127
wordwrap-markup	127
wordwrap-string	127
wordwrap-string-markup	127