



Tesis Doctoral

**Tipografía musical para la educación
y análisis de la dinámica del software libre**

Francisco Vila Doncel

Departamento de
Didáctica de la expresión musical, plástica y corporal

2015



Tesis Doctoral

**Tipografía musical para la educación
y análisis de la dinámica del software libre**

Francisco Vila Doncel

Departamento de
Didáctica de la expresión musical, plástica y corporal

Conformidad del Director:

Fdo: *Sebastián Feu Molina*

2015

Índice general

Resumen	xvii
Agradecimientos	xix
Presentación	1
Marco teórico e histórico. Software libre, tipografía musical y educación	9
1 Compendio bibliográfico	11
1.1 Generalidades sobre investigación y metodología	13
1.2 Estadística y gráficos estadísticos	14
1.3 Control de versiones	15
1.4 Software libre y educación	15
1.5 Tipografía	16
1.6 Notación y tipografía musical	16
1.7 Otras obras sobre software de tipografía y codificación musicales	19
1.8 Discusión de resultados y prospectiva	20
1.9 Cuestiones de propiedad intelectual	21
2 Software libre y control de versiones	23
2.1 Conceptos	23
2.1.1 Programas y código fuente	23
2.1.2 Software	25
2.1.3 Libertad	25
2.1.4 El software propietario	26
2.2 El software libre	27
2.2.1 Libertad frente a gratuidad	27
2.2.2 La cláusula copyleft. La licencia GPL	29
2.3 Por qué software libre: una cuestión de libertad	33
2.4 Aspectos éticos	34
2.5 El aspecto económico	34
2.6 Seguridad, fiabilidad	36
2.7 Disponibilidad dentro de la legalidad	37
2.8 La paradoja de la masa crítica. La innovación disruptiva	37
2.9 Desarrollo cooperativo	39
2.10 Control de versiones. Git	41
2.10.1 Nociones de Git	43
2.10.2 Ramas y mezclas	45
2.11 El Software libre en Extremadura	46
2.11.1 El proyecto gnuLinEx	46
2.11.2 El CIRL	47

2.11.3	CENATIC	48
2.11.4	El proyecto UEX-Linux	48
2.12	Licencias libres en Extremadura	50
3	Tipografía musical y el proyecto LilyPond	53
3.1	Tipografía	53
3.2	Notación musical	54
3.2.1	Meta-principios de la notación musical	56
3.3	Orígenes de la notación musical	57
3.4	Tipografía musical	59
3.5	Tipografía por ordenador	61
3.6	Tipografía musical por ordenador. Orígenes de LilyPond	63
3.6.1	LilyPond como software libre	70
3.7	Interfaces de texto	72
3.7.1	Ejemplos de interfaces de texto	75
3.7.2	Entornos de desarrollo	75
3.7.3	LilyPondTool	77
3.7.4	Frescobaldi	78
3.7.5	Otros entornos de escritorio	79
3.7.6	Entornos online (SAAS)	79
3.8	LilyPond en la práctica	80
3.8.1	Descarga e instalación	81
3.8.2	Creación de una partitura sencilla	82
3.8.3	Procesado del documento	82
3.9	Aspectos del resultado visual	83
3.9.1	La metáfora de la máquina de pan	83
3.9.2	Líneas adicionales	83
3.10	MuseScore	85
4	Introducción al lenguaje LilyPond	89
4.1	Elementos de notación más frecuentes	90
4.1.1	Altura y duración de las notas	90
4.1.2	Silencios y ligaduras	91
4.1.3	Clave, tonalidad, compás	92
4.1.4	Polifonía	92
4.1.5	Música vocal	92
4.1.6	Articulaciones y matices	94
4.1.7	Acordes, anacrusas, tresillos, mordentes, calderón, doble barra	94
4.1.8	Repeticiones	94
4.1.9	Variables y transposición	95
5	Aplicaciones	97
5.1	El Liedboek	97
5.2	La integral de Oskar Fried	98
5.3	Los arreglos y orquestaciones de Aurélien Bello	100
5.4	El libro <i>Un canto navideño a nuestra tierra</i> , de Daniel López y Emilio Díaz	101
5.5	Grandes proyectos: Mutopia	103
5.6	Musipedia	105
5.7	La colección de T. L. de Victoria, de Nancho Álvarez	105

5.8	La <i>Ópera Libre</i> de Valentin Villenave	106
5.9	El cancionero de A. Simpson	109
5.10	El cancionero <i>Land of plenty</i> , de Jack Cooper	110
5.11	Serpent Publications, de Laura Conrad	112
5.12	Kainhofer Edition, de Reinhold Kainhofer	114
5.13	LilyPond en educación y en transcripción.	115
5.13.1	Cursos en Badajoz	115
5.13.2	Congreso de la AEC, Strasbourg 2007	117
5.13.3	Talleres en Valencia de Don Juan	119
5.13.4	Abjad, de Víctor Adán y Trevor Bača	119
5.13.5	MEAWS, de Graham Percival	120
5.13.6	La tesis de Luca Rossetto	120
5.13.7	LilyPond para personas con discapacidad visual	120
5.14	El clarinete robótico del equipo de Mark Sheahan y Peter Chubb	122
5.15	Otras aplicaciones	123
5.15.1	El premio LOMUS	125
5.15.2	Desarrollos personalizados	126
Estudio 1. Dinámica del software libre de edición de partituras		127
6 Estudio 1. Objetivos e hipótesis		129
7 Estudio 1. Metodología		131
7.1	Método	131
7.2	Variables	132
7.3	Muestra	133
7.3.1	Naturaleza de los datos	134
7.4	Puntos de interés	134
7.5	Fuente primaria	134
7.6	Disponibilidad de los datos	136
7.7	Instrumento	137
7.7.1	Software empleado	137
7.8	Procedimiento	138
7.8.1	Obtención de datos	138
7.8.2	Caracterización de las intervenciones	139
7.9	Fuentes de error	141
7.10	Vídeos de la evolución	143
8 Estudio 1. Resultados y discusión		145
8.1	Desarrolladores	145
8.1.1	Carga total de trabajo	148
8.2	Intervenciones	148
8.3	Tamaño del código	150
8.4	Tamaño de la documentación	153
8.5	Tamaño de los instalables binarios	155
8.6	Actividad media diaria, semanal y mensual	157
8.7	Evolución del trabajo sobre el código	159
8.8	Evolución del trabajo sobre la documentación	161

8.9	El “factor bus” de Percival: actividad frente a ranking y ley de Zipf	164
8.10	Modelización de la serie temporal	168
8.10.1	Descomposición de la serie temporal	169
8.10.2	Diagnósticos sobre el modelo	170
8.10.3	Valor predictivo del modelo	172
8.11	Trabajos relacionados. Vídeos de la evolución	172
8.12	Vídeos: 1. Método de Ogawa	172
8.12.1	Documentación del proceso de generación de los vídeos por el método de Ogawa	174
8.13	Vídeos: 2. Método de Caudwell	176
8.13.1	Documentación del proceso de generación de los vídeos por el método de Caudwell	177
9	Estudio 1. Conclusiones	181
9.1	Limitaciones y prospectiva	183
9.2	Aplicabilidad	184
9.2.1	Aplicabilidad de la metodología	184
9.2.2	Aplicabilidad de los resultados	184
	Estudio 2. Experimento sobre el desempeño en educación musical de una herramienta libre para producir partituras	185
10	Estudio 2. Objetivos e hipótesis	187
11	Estudio 2. Metodología	191
11.1	Método	191
11.2	Fases de la investigación	192
11.3	Población y muestra	192
11.3.1	Identificación de las variables	194
11.3.2	Diseño y validación del instrumento	194
11.3.3	Aleatorización	195
11.3.4	Anonimización	197
11.4	Instrumento	198
11.4.1	Diseño del cuestionario	198
11.4.2	Diseño del experimento	203
11.5	Preparación del material	209
11.5.1	Selección de los modelos: grupo 1	210
11.5.2	Selección de los modelos: grupo 2	216
11.5.3	Proceso de recreación: técnicas utilizadas	223
11.5.4	Recreación y validación: el problema de la calidad	226
11.5.5	Montaje del material	242
11.6	Validación del cuestionario y del material	244
11.6.1	Variables de validación	245
11.6.2	Jueces participantes en la validación	246
11.6.3	Resultados de la validación: observaciones de los jueces	247
11.6.4	Resultados de la validación: V de Aiken por pregunta	247
11.6.5	Resultados de la validación: V de Aiken por juez	249
11.6.6	Consistencia interna	250

11.6.7	Estabilidad del instrumento: fiabilidad test/retest	251
11.7	Puesta en práctica	252
11.7.1	Fase piloto	255
11.8	Análisis estadístico	256
11.8.1	Baremo para la puntuación del experimento	256
11.8.2	Estadísticos y software empleado	258
12	Estudio 2. Resultados y discusión	261
12.1	Iluminación de la escena	261
12.2	Participantes	263
12.2.1	Datos sociodemográficos	264
12.3	Uso de partituras	267
12.3.1	Valores globales	267
12.3.2	Asociaciones entre frecuencias de uso	269
12.4	Uso de software de edición de partituras y formación en el mismo	278
12.4.1	Uso de software	278
12.4.2	Formación recibida en software	279
12.5	Resultado del experimento comparativo	280
12.5.1	Recuentos global y medio	280
12.5.2	Frecuencia de las puntuaciones	282
12.5.3	Comparación de medias para los baremos Favorable y Desfavorable	284
12.5.4	Comparación del baremo con la media teórica	285
12.5.5	Efecto de la dominancia lateral en la elección Izquierda/Derecha	286
12.5.6	Efecto de la iluminación y la graduación para la vista	287
12.5.7	Impacto del sexo sobre la elección A/B	289
12.5.8	Efecto de la edad sobre la elección A/B	289
12.5.9	Otros factores	290
12.5.10	Resultado por modelos	291
12.5.11	Resultado por modelos, según sexo	293
12.5.12	Resultado por modelos, según graduación e iluminación	294
12.5.13	Resultado por modelos, según estudios	294
12.5.14	Uso de árboles de regresión para la clasificación por modelos	295
12.6	Observaciones	299
13	Estudio 2. Conclusiones	303
13.1	Conclusiones respecto de los objetivos	303
13.2	Conclusiones respecto de las hipótesis	305
13.3	Limitaciones y prospectiva	308
13.4	Repetibilidad	311
13.5	Aplicabilidad	312
13.6	Conclusiones finales	313
	Apéndices	314
	A Lista completa de los desarrolladores de LilyPond	317
	B El epitafio de Sículo	319
	C Modelos A y B de los casos del experimento	323

D Modelo del cuestionario del estudio 2	335
E Cuestionario de validación del estudio 2	339
F Cálculo del coeficiente de validez V de Aiken	345
G Glosario de términos técnicos y musicales	347
H Software utilizado	353
Bibliografía	355

Índice de figuras

1	Bibliografía utilizada, recuento por tipos	12
2	Año de publicación de las referencias bibliográficas posteriores a 1980	13
3	Las categorías del software respecto del concepto de libertad, según Hung	28
4	Esquema de una ramificación y mezcla en el historial de versiones.	45
5	Mapa conceptual que relaciona a Git, LilyPond, GNU y Linux	46
6	Categorías de la tipografía	53
7	Notación musical ecfonética bizantina	58
8	Fragmento de un «Alleluia», ejemplo de notación neumática <i>in campo aperto</i>	58
9	Doble página del «Harmonice Musices Odhecaton» de O. Petrucci	59
10	Fragmento de música etiquetado con los nombres de los elementos de notación	61
11	Situación de GNU LilyPond dentro de las categorías de software relacionadas con el tema de la investigación.	63
12	Mapa cronológico de los principales acontecimientos en el desarrollo de la informática y la tipografía musical por ordenador.	65
13	Situación de LilyPond frente a otros programas de tipografía musical	71
14	Función del compilador respecto del código fuente.	73
15	Esquema de bloques que relaciona LilyPond con su entrada y su salida	74
16	Superposición de la capa del interfaz de usuario a la capa del motor	76
17	El entorno Frescobaldi, de Wilbert Berendsen, en una captura de pantalla	78
18	Código de entrada propuesto para un ejemplo	82
19	Resultado correspondiente al código de ejemplo de la figura 18	83
20	Aspecto del ajuste tipográfico de las líneas adicionales en presencia de una alteración, por parte de LilyPond	84
21	Comparativa de los ajustes de las líneas adicionales en presencia de una alteración, en tres publicaciones comerciales	85
22	Fragmento de las notas críticas de la integral de los <i>lieder</i> de O. Fried	99
23	Muestra de uno de los villancicos de “Un canto navideño a nuestra tierra”	102
24	Primera página del prólogo de la <i>ópera libre</i>	107
25	Muestra extraída de la edición digital del cancionero de A. Simpson	110
26	Muestra extraída del cancionero <i>Land of Plenty</i> de Jack Cooper	111
27	Muestra de una <i>pavana</i> de Anthony Holborne editada por Laura Conrad para el catálogo de Serpent Publications.	113
28	Hoja de referencia para LilyPond, por R. Kainhofer	115
29	Presentación sobre LilyPond que se hizo en el congreso anual de la AEC	118
30	Fragmento de <i>Violent Dance for orchestra</i> , del compositor chino, ciego de nacimiento, Hu Haipeng.	121
31	Efecto de aplicar la extensión “Articulate” a un fragmento de música	123
32	Ejemplo de canto kievano en notación cuadrada.	126
33	Elementos más relevantes del proyecto LilyPond	135

34	Ejemplo de una intervención de código pura, en la que un desarrollador modifica el número de versión del programa	140
35	Ejemplo de identidades diversas que corresponden al mismo autor.	146
36	Número de desarrolladores activos distintos por mes desde octubre de 1996 hasta junio de 2015	147
37	Número de desarrolladores activos distintos por año completo desde el inicio del proyecto en 1996 hasta 2014	147
38	Número de intervenciones individuales en el código fuente de LilyPond por mes desde octubre de 1996 hasta junio de 2015.	149
39	Número de intervenciones individuales en el código fuente de LilyPond por año completo desde el inicio del proyecto en 1996 hasta 2014	149
40	Evolución de la proporción entre el número de intervenciones y el número de desarrolladores por año.	150
41	Evolución del número total de archivos del proyecto y del número total de líneas de código	152
42	Tamaño del código fuente de cada una de las traducciones de la documentación de LilyPond	154
43	Evolución del tamaño de los 84 <i>tar balls</i> históricos de la documentación, frente a la fecha	155
44	Tamaño de los instaladores binarios durante los diez últimos años	156
45	Días de diferencia entre versiones sucesivas de LilyPond durante los diez últimos años	157
46	Número total de intervenciones frente a la hora del día, para toda la historia del proyecto.	158
47	Número total de intervenciones por día de la semana.	158
48	Número total de intervenciones por mes del año.	159
49	Líneas de código modificadas durante la vida del proyecto LilyPond	160
50	Carga de trabajo sobre la documentación del proyecto, frente a la fecha	162
51	Historia de las intervenciones sobre la partitura de la ópera <i>Affaire Étrangère</i> de Valentin Villenave	163
52	Curva de contribuciones por autor frente a su ranking	165
53	Evolución de la curva de Zipf en seis períodos de tres años, para los 18 años de historia de contribuciones al proyecto LilyPond.	166
54	Evolución del número de contribuciones registradas en los 18 años del proyecto LilyPond por parte de los autores sucesivamente más prolíficos.	167
55	Descomposición de la serie temporal original en la curva de tendencia, variaciones estacionales y componente aleatoria.	169
56	Diagnósticos sobre el modelo: residuos estandarizados, coeficientes de autocorrelación de los residuos y valores p del estadístico de Ljung-Box.	171
57	Pronóstico que prolonga la tendencia de la serie temporal según ARIMA	173
58	Esquema de los elementos de la primera animación	174
59	Esquema de los elementos de la segunda animación	177
60	Juegos “A” y “B” para los diez casos, ordenados	196
61	Lista de combinaciones utilizada para aleatorizar la posición de los modelos	196
62	Juegos de elementos directo y complementario con el orden i/d aleatorizado	197
63	Hoja de correspondencias de códigos anónimos y nombres reales	198
64	Representación esquemática de los diez elementos “A”	203
65	Representación esquemática de los dos elementos “A” y “B” de un caso	204
66	Esquema simplificado de presentación de un caso	204

67	Esquema de la clasificación de los diez casos del material	206
68	Ajuste de mínimos	224
69	Ajuste de máximos	225
70	Anotación del espacio vertical de LilyPond	228
71	Detalle del modelo 3: diferencia en las ligaduras	232
72	Detalle del modelo 7: formato de los tresillos en el original y en LilyPond	238
73	Detalle del caso número 8, cuarteto de cuerda (elemento A)	240
74	Esquema definitivo del montaje de los casos.	243
75	Situación de los adhesivos	243
76	Valores del coeficiente de validación V de Aiken para la pertinencia, redacción, diseño general y modelos de partitura	248
77	Valores del coeficiente de validación V de Aiken por cada juez	249
78	Valores medios del coeficiente de validación V de Aiken por juez	249
79	Esquema de la situación de los elementos del experimento	255
80	Histograma de los valores de exposición EV para la iluminación de la escena	261
81	Recuento de las actividades declaradas por los participantes	266
82	Uso de partituras: histograma de cada respuesta	268
83	Gráfico de asociación entre el uso de partituras originales y fotocopias	272
84	Uso de partituras: representación de las 49 tablas de contingencia	274
85	Efecto de la edad sobre las horas dedicadas al manejo de partituras	277
86	Modelo polinómico para estimar la edad en que se produce el máximo	278
87	Recuento general del número de veces que los participantes prefirieron las par- tituras de LilyPond	281
88	Frecuencias de los resultados	283
89	Curvas de densidad para las puntuaciones	283
90	Efecto de la dominancia lateral sobre la elección del modelo de la izquierda o de la derecha	286
91	Efecto de la corrección visual: proporciones en función de la puntuación del baremo	287
92	Efecto de la corrección visual sobre la indiferencia: tablas de contingencia para los resultados A y B	288
93	Efecto de la corrección visual: proporciones en función de la puntuación del baremo	289
94	Modelos lineales para las puntuaciones medias por edad	290
95	Resultado global de la métrica que contabiliza la preferencia por LilyPond en función de los estudios y de la dominancia lateral	291
96	Resultado del experimento por casos	292
97	Aceptación de cada modelo en función del sexo de los participantes	293
98	Aceptación de cada modelo en función de los estudios completados	294
99	Árbol de regresión para el resultado desfavorable obtenido en la partitura de guitarra	296
100	Árbol de regresión para el rechazo del modelo 7, después de la poda	298
101	Árbol de regresión para la indiferencia	298
102	Árbol de regresión para el índice de indiferencia, después de la poda	299
103	Tamaño aparente de las notas para Maestro (Finale) y Feta (LilyPond)	301
104	Lista de los 153 desarrolladores del proyecto LilyPond desde su creación	317
105	El epitafio de Sícilo	320
106	Código de LilyPond de la transcripción del epitafio de Sícilo	321
107	Caso nº 1. Tipo: particella, para violoncello.	324

108	Caso nº 2. Tipo: pieza para piano.	325
109	Caso nº 3. Tipo: <i>Lied</i> con acompañamiento.	326
110	Caso nº 4. Tipo: pieza coral SATB.	327
111	Caso nº 5. Tipo: percusión.	328
112	Caso nº 6. Tipo: <i>Lead Sheet</i> (hoja-guía de acordes).	329
113	Caso nº 7. Tipo: pieza para guitarra.	330
114	Caso nº 8. Tipo: cuarteto de cuerda.	331
115	Caso nº 9. Tipo: música de cámara.	332
116	Caso nº 10. Tipo: partitura para orquesta sinfónica	333
117	Ejemplo de salida del programa.	346

Índice de tablas

1	Versiones del programa de edición musical Personal Composer (Fuente: Personal Composer, Inc.; Datos de 2008).	35
2	Comparación de los escenarios de traducción del software privativo y el software libre	40
3	Diferencias entre el software libre y el propietario	40
4	Correspondencias de interfaces gráficos y de texto en una selección de programas informáticos	75
5	Casos de LilyPond como servicio online	80
6	Títulos de los doce villancicos de <i>Un canto navideño a nuestra tierra</i>	101
7	Autores activos por mes	146
8	Datos para la estimación de la carga total de trabajo del proyecto LilyPond. . .	148
9	El código fuente de LilyPond, clasificado por lenguajes	151
10	Tamaño relativo de las secciones del código fuente del proyecto LilyPond . . .	153
11	Intervalos de confianza para los coeficientes del modelo	170
12	Campos de datos del cuestionario	203
13	Elementos gráficos característicos del modelo 1	211
14	Elementos gráficos característicos del modelo 2	212
15	Elementos gráficos característicos del modelo 3	213
16	Elementos gráficos característicos del modelo 4	215
17	Elementos gráficos característicos del modelo 5	216
18	Elementos gráficos característicos del modelo 6	217
19	Elementos gráficos característicos del modelo 7	218
20	Elementos gráficos característicos del modelo 8	219
21	Elementos gráficos característicos del modelo 9	221
22	Elementos gráficos característicos del modelo 10	222
23	Cuadro-resumen de los diez casos de prueba para el experimento	223
24	Variables del cuestionario de validación	246
25	Composición del panel definitivo de expertos para la validación	246
26	Análisis de varianza	251
27	Algunos locales utilizados, con sus iluminaciones	262
28	Edades de los participantes	264
29	Sexo de los participantes	264
30	Lado preferente de los participantes	265
31	Nivel de estudios musicales completados por los participantes	266
32	Participantes que utilizan algún tipo de partitura sobre una base diaria o semanal	267
33	Uso de partituras. Recuento por tipos y valores más repetidos	267
34	Frecuencias absolutas más elevadas de las respuestas al uso de partituras . . .	268
35	Tipos de partitura más frecuentemente utilizados sobre una base diaria o semanal	269
36	Tabla de contingencia para los usos de partitura más frecuentes	270

37	Estadísticos de asociación entre el uso de los tipos de partitura Original y Fotocopiada	270
38	Prueba de dependencia χ^2 de Pearson para la pregunta «uso de partituras» . . .	271
39	Reglas de asociación halladas para “uso diario de partituras originales” en el término izquierdo	273
40	Resumen de frecuencia de uso de partituras según la actividad declarada	275
41	Reglas de asociación halladas que implican el uso diario de partituras fotocopiadas	275
42	Resumen de las horas dedicadas por los participantes al manejo de partituras .	276
43	Uso de software de edición de partituras: recuento global	278
44	Formación específica en software de edición de partituras	280
45	Resultados del experimento: recuento medio por participante	282
46	Contraste para los resultados “Original” y “LilyPond”	285
47	Contraste para el baremo que incluye el factor de indiferencia, con su media teórica	286
48	Estadísticos de asociación entre la variable Corrección visual del participante y la predilección por LilyPond	288
49	Estadísticos de asociación entre la variable Sexo del participante y la predilección por LilyPond	289
50	Coefficientes de los modelos lineales	290
51	Recuento por modelos	291
52	Tabla de errores para el árbol de regresión del modelo 7	297
53	Tabla de errores para el árbol de regresión de la indiferencia	299

Resumen

Partiendo de las observaciones de que el consumo de partituras es central en el ámbito de la educación musical, y que el software libre posee características que lo hacen muy interesante para su uso en la práctica docente, en un primer estudio se describe desde diferentes puntos de vista un proyecto de software libre dedicado a la tipografía musical. Se analiza la evolución del proyecto durante un período de 18 años utilizando como documento la historia completa de su código fuente. Las técnicas de visualización y de modelización empleadas indican que el sistema ofrece robustez frente a los cambios en la composición de su equipo de desarrollo.

En un segundo estudio se diseña una prueba comparativa *ad hoc* y se someten a ella las partituras elaboradas con este programa, a través de los datos recogidos de 106 músicos participantes, analizando también sus patrones de consumo de partituras musicales. El cuestionario y el material se validan mediante un panel de jueces. Se realiza una prueba piloto y se comprueba la estabilidad en el tiempo mediante test-retest. El resultado indica que los músicos prefieren de forma estadísticamente significativa las partituras hechas con este programa de descarga libre y gratuita. Se concluye que una solución de este tipo es válida para su uso en educación y en el mundo profesional.

Los dos estudios se hacen preceder de un análisis bibliográfico completo y una exposición teórica e histórica acerca del software libre y la tipografía musical. La prospectiva reclama la necesidad de estudios más profundos al respecto, ya que se está comenzando a atisbar el potencial de la herramienta.

Abstract

Based on the observations that use of sheet music is paramount in the field of music education, and that free software has characteristics that make it very interesting for use in all kinds of learning activities, our first study describes from multiple points of view a free software project dedicated to music typography. Over a span of 18 years, data about development of the project are analyzed from the whole registered history of its source code. Several visualization and modeling techniques suggest that such a system offers a fairly high level of robustness against deep changes in the composition of the developers' team.

In our second study, a set of scores made with this program were subject to a custom-made, comparative test through data collected from 106 participating musicians, and at the same time their consumption patterns of musical scores are analyzed. The survey questionnaire and the score models used are validated by a team of experts. A test essay is made and timewise stability is checked by means of the test/retest method. Results show that musicians seem to prefer, in a statistically significant way, music scores made with a program which happens to be freely available for download. We conclude that such a solution makes a valid option for education and professional uses.

Both studies are leaded by a complete bibliographic compendium and a theoretical and historical framework about free software and music typesetting. Further and deeper studies are claimed to be done, as only the surface of the tool has started to show its full potential.

Agradecimientos

Deseo reconocer a la Facultad de Educación de la UEX y al Departamento de Didáctica de la expresión musical, plástica y corporal su iniciativa de abrir a todos los titulados superiores de Conservatorio la puerta a los estudios del ciclo de Doctorado.

Agradezco al director de la Tesis, Dr. D. Sebastián Feu, por su paciente e incansable labor de orientación y por todo lo que he aprendido de él.

Gracias también al Dr. D. Zacarías Calzado por ese apoyo entusiasta que me ha brindado y que sin duda ha contribuido en mí a una dedicación que difícilmente se habría producido de otro modo.

A la Comunidad de usuarios y desarrolladores de Software Libre que han creado tantas y tantas soluciones a todos los problemas imaginables, y que inspiran unos valores tan necesarios para nuestro mundo.

Deseo agradecer especialmente a los músicos Laura Conrad, Jack Cooper, Valentin Villenave, Daniel López, Nancho Álvarez, Marcus Brinkman y Hu Haipeng que hayan puesto desinteresadamente a mi disposición muestras de su trabajo para ser incluidas en esta memoria.

Gracias a Han-Wen, a Jan y al resto del equipo por su visión de futuro y por regalarnos a todos su esfuerzo durante tantos años.

A todos los participantes en la encuesta y el experimento: músicos, alumnos, compañeros, tan dispuestos a ayudar desinteresadamente, porque vuestra vista de lince y vuestro agudo criterio han sido fundamentales para la investigación.

A mi familia y en especial a las niñas, que espero y deseo que no odien la palabra “tesis” tantas veces pronunciada en los últimos años. Gracias por vuestra tolerancia y comprensión. Sois ya expertas en partituras: esos dos puntitos situados abajo a la izquierda en la figura 85, sois vosotras.

Chari, a ti el agradecimiento más personal, por haber sido tan buena compañera de estudios, por tu optimismo, por tu energía, por tus sabios consejos y por la paciencia que has demostrado.

*Todo está escrito en la
partitura, excepto lo esencial.*

Gustav Mahler

Presentación. Interés de los dos estudios

El presente trabajo, dividido en dos estudios, se desarrolla dentro de la zona de intersección entre la tipografía musical, el *software* libre y la Educación.

En su origen están una serie de observaciones:

- La educación musical emplea notación musical para la fijación y transmisión del contenido.
- Es importante que la notación musical sea correcta, clara y estéticamente satisfactoria. Algunos de estos criterios pueden ser subjetivos, una cuestión de preferencia personal.
- La partitura musical tiene dos tipos de agentes: los productores y los consumidores.
- A su vez, se distinguen tres grupos principales de personas productoras y consumidoras de partituras musicales: los músicos profesionales, los estudiantes y los docentes.
- Hay dos tipos principales de material de notación musical: el efímero y el permanente. Son ejemplos de material efímero la música escrita en pizarra, los arreglos rápidos provisionales, los apuntes de un compositor o las partituras pensadas para ser ejecutadas una sola vez. Las ediciones profesionales y en general todo el material comercial en papel se considera que tiene un carácter de relativa permanencia.
- La única alternativa a la escritura manual de música es el empleo de un programa de ordenador, un *software* de edición.
- Los programas de edición de partituras más empleados son productos comerciales que pueden tener un elevado precio en el mercado.¹
- Los productos comerciales se atienen a las conveniencias del mercado y pueden desaparecer. Dicho de otro modo, las actualizaciones pueden espaciarse arbitrariamente en el tiempo o cesar definitivamente, y los usuarios no deciden las funcionalidades que el producto incorpora.
- El *software* libre ofrece varias alternativas a los productos comerciales más populares.
- El *software* libre presenta ciertas ventajas, unas generales y otras que tienen un valor especial para la educación.
- Las partituras musicales elaboradas con *software* libre cumplen los criterios de corrección, claridad y armonía estética que los músicos aprecian y valoran, pero no se ha constatado empíricamente.

¹Finale 2014: 499,00€. Fuente: tienda online ST Midi Music. Sibelius 7.5: 665,21€. Fuente: AVID store. Versiones completas. Precios de mayo de 2015.

Pueden relacionarse algunos escenarios comunes en los que interviene la notación musical:

- Un profesor de solfeo distribuye entre sus alumnos un ejercicio para realizar en clase.
- Un musicólogo prepara la edición comentada de un manuscrito.
- Un profesor de análisis musical elabora unos apuntes en los que inserta ejemplos de la obra objeto del estudio.
- Un compositor va a estrenar una sinfonía y los músicos necesitan los papeles de atril para preparar sus partes y el ensayo general.
- Un investigador recoge una canción popular y la transcribe.
- Un instrumentista perfecciona su técnica ejecutando estudios de dificultad progresiva.
- Una orquesta de cámara interpreta una ópera barroca.
- Un pianista acompañante presta servicios en una academia de ballet.
- Un coro va a preparar un programa de música del renacimiento.
- Un grupo escolar ensaya un musical para la actuación de fin de curso.
- Un músico de *jazz* compone una pieza y la arregla para *big band*.

Son pocas las actividades musicales en las que no esté presente la música escrita: incluso cuando la práctica original es ajena a la escritura, como las músicas de tradición oral o el *jazz*, otros músicos tratan de fijarla y transmitirla en notación convencional para su estudio y conservación. Es obvia, pues, la importancia de la notación musical moderna como soporte de toda la tradición musical de Occidente, y como vehículo de enseñanza e investigación académica. La tipografía musical (a veces denominada más genéricamente “edición de partituras”) es una técnica utilizada para producir esta notación musical. Cómo producir esta notación es, pues, un problema con el que todo músico se enfrenta tarde o temprano.

En general no puede decirse, hablando en propiedad, que un programa de edición de partituras sea un *software* educativo, pero se observa que los docentes y los estudiantes manejan partituras musicales de todo tipo, diariamente, una gran cantidad de horas a la semana: la encuesta del segundo estudio aportará objetividad a esta afirmación. Así pues, la tipografía musical y el manejo (producción y consumo) de partituras son una disciplina transversal en el mundo de la música, tanto en el aspecto profesional como en el de la docencia, como lo son la lectura y la escritura del lenguaje ordinario en otros ámbitos.

Después de 500 años desde que la imprenta se extendió por Europa, es difícil hallar aún libros escritos a mano; no ocurre lo mismo con las partituras. Pueden encontrarse obras de la literatura musical cuya notación es manuscrita, por ejemplo: libros serios sobre historia de la música española (Salazar, 1989; Vega, 2012), un cancionero de arreglos de música extremeña (Pérez-Ribes, 1989), un libro de texto para conservatorios (López García, 1977) o artículos académicos sobre informática musical como el de (Pestana y Pestana, 2012), además de un notable caso de artículo técnico reciente sobre un instrumento musical que ha recibido premios por su diseño (Gruenbaum, 2014). Es cierto que la escritura musical es más compleja

que la verbal, y que si bien como explica detalladamente Mele (2004), los primeros libros impresos con notas musicales aparecieron de manera casi contemporánea en Italia y Alemania en torno a 1475, no fue hasta 1501, cuatro décadas después de la Biblia de Gutenberg, cuando Ottaviano Petrucci publica el primer libro enteramente musical realizado con tipos móviles. Esto da una idea de las dificultades inherentes a la técnica de la composición tipográfica de notación musical.

Los orígenes de la impresión musical mediante ordenador son tan remotos como la propia utilización del ordenador para la producción de material impreso en general. Como se ha dicho, varias opciones muy populares están disponibles en el mercado. Sin embargo, el empleo de programas de edición de partituras puede producir una situación muy incómoda. El músico que quiere hacer partituras está muy presionado para utilizar soluciones de amplia difusión y elevado coste, colocándose en un dilema que se resuelve con gran frecuencia, según se observa, utilizando copias no autorizadas.

A mediados de la década de 1980 surge una corriente que fluye en una dirección alternativa a la que acabamos de plantear. Existe una categoría de programas que no está sujeta al dilema expuesto. El usuario tiene la posibilidad de emplear *software* libre, que describiremos oportunamente, y aunque ciertamente nunca estará seguro de si es una apuesta en la que puede confiar, quizá tampoco lo esté con las soluciones habituales, cerradas, privativas, de pago, y que pueden desaparecer.² Y aunque entre los implicados se ha sugerido muchas veces la conveniencia de ofrecer argumentos bien fundados a los usuarios potenciales (ya sean particulares o instituciones educativas), nunca hasta ahora, que sepamos, se ha planteado seriamente la posibilidad de probar de forma empírica que una solución basada en *software* libre puede cumplir lo que se espera de una herramienta de este tipo, situándose en una posición de alternativa real.

Con los dos estudios de esta Tesis se abordan sendos enfoques orientados al propósito de establecer si este *software* u otros similares tienen unas características de seriedad, estabilidad, versatilidad, adaptabilidad y desempeño o adecuación a un determinado objetivo dentro del campo de la educación y en el mundo profesional. Los dos estudios supondrán, esperamos, modestas aportaciones a la mejora de la educación musical a través del empleo de *software*³ libre para la producción de partituras.

²La mención a esta posibilidad se hace en referencia al programa Sibelius, que aún se comercializa pero solamente publica versiones sin funcionalidades nuevas desde que la empresa AVID despidió al equipo de desarrollo, lo que es una pérdida neta para el mundo de la tipografía musical.

³Aunque *software* es una voz inglesa, en adelante y para evitar el posible cansancio visual producido por una palabra tantas veces repetida en cursiva, se escribirá el término en tipo recto.

Estudio 1

El estudio descriptivo que comienza en la página 129 expone los aspectos internos fundamentales del proyecto LilyPond (Nienhuys y Nieuwenhuizen, 2003), situándolo en su contexto dentro de un ecosistema de aplicaciones que serán también descritas oportunamente. Analizando su historia, y utilizando como fuente los detallados registros informáticos que resultan del empleo de un sistema (también libre) de control de versiones, se visualiza el pasado y el presente del proyecto, persiguiendo arrojar alguna luz sobre su futuro.

El software puede llegar a ser un asunto complicado para la mayoría de las personas, quizá porque es inmaterial por definición, y hacerse una idea del papel que representa en un determinado proceso, producto o actividad, puede exigir un esfuerzo de imaginación. Los programas de software son objetos complejos, con muchos aspectos que se van desarrollando de forma paralela en el tiempo, acumulando multitud de aportaciones en pequeñas dosis. El software libre es una interesante materia de estudio por los siguientes motivos:

- Intervienen en su creación muchos desarrolladores, pagados o voluntarios en su tiempo libre.⁴
- El código está siempre disponible y sabemos cuándo y quién hizo cada modificación.⁵
- A pesar de tanta transparencia, se mantiene un velo de misterio sobre su futuro porque es difícil visualizarlo en toda su complejidad, con una perspectiva amplia.

LilyPond entró en su mayoría de edad a principios de 2015, y en esos 18 años se ha desarrollado de forma semejante al crecimiento de un ser vivo. Existe, por tanto, una *dinámica* del software libre que se analiza a través de este caso de estudio.

Desde el punto de vista personal vemos interés en el logro que sería contribuir, dentro de nuestras posibilidades, a una mejor visualización de los procesos de desarrollo del software, y, una vez examinados los métodos que permitirían hacerlo, poder aplicarlos también a otros proyectos, un logro que tendría el valor añadido de extender su utilidad a otros ámbitos dentro de la “gran comunidad” del software libre.

El primer trabajo finaliza con dos ejemplos de visualizaciones en vídeo, creadas por el autor, que constituyen sendas aplicaciones del análisis de software.

⁴En el caso de LilyPond, 153 desarrolladores en total.

⁵Un total de 26.067 modificaciones individuales (con una media de 4 al día) para LilyPond.

Estudio 2

El segundo estudio es el diseño, validación y empleo de una encuesta que contiene un experimento sobre el desempeño de la herramienta libre de producción de partituras musicales LilyPond en entornos reales de educación y del mundo profesional.

Hemos visto que el primer estudio describe el proyecto LilyPond, usando como fuente de información un extenso documento como es la historia registrada de su código fuente, pero se ha mencionado un aspecto que queda aún en el terreno de la opinión subjetiva. Se trata de la calidad de los resultados, expresada en términos como “LilyPond hace las partituras más bellas del mundo”, “recrea la mejor tradición de las partituras grabadas a mano”, etcétera. Uno de sus creadores, el holandés Han-Wen Nienhuys, tiene el lema “*La música hermosa merece ser tipografiada de manera hermosa*”, que da un aire poético a un término técnico como la tipografía.

En un buscador general de web, la búsqueda bajo los términos *estética de la notación musical* ofrece como primer resultado la página principal de LilyPond en español.

Al buscar en Google Académico por los términos “*music notation*” “*high quality*” (notación musical, alta calidad), el segundo resultado es «LilyPond, a system for automated music engraving» (Nienhuys y Nieuwenhuizen, 2003). Al buscar por los términos “*music engraving*” (edición de partituras), la misma referencia aparece como el primer resultado.

Ahora bien: ¿qué opinan los músicos? Averiguarlo es el motivo de nuestro segundo estudio, que aportará objetividad a la cuestión de si las partituras producidas con una determinada herramienta de software libre están a la altura de las expectativas de los músicos. Para ello se ha sometido una colección de partituras recreadas con este sistema al examen por parte de 106 participantes del mundo de la música para que las compararan con sus originales.

Uno de los detonantes de un estudio como el presente, que trata de objetivar la belleza relativa de una partitura, fue el siguiente comentario en un círculo *online* especializado:

```
From:          Michael Rivers
Subject: (...) Sibelius Software UK office shutdown
Date:          Tue, 14 Aug 2012 00:05:45 -0700 (PDT)
(...)
I just asked my wife to choose among the same piece I printed using
Sibelius, the Finale demo, and LP, and she immediately pointed to the
LP score and said "That one's perfect."
```

Traducción:

De: Michael Rivers
Asunto: (...) Cierra la oficina de Sibelius Software en el Reino Unido
Fecha: mar, 14 ago 2012 00:05:45 -0700 (PDT)
(...)
Le acabo de pedir a mi esposa que elija entre tres versiones de la misma pieza que yo mismo he impreso, respectivamente, mediante Sibelius, una demo de Finale, y LilyPond, e inmediatamente ha señalado la partitura de LilyPond diciendo «Ésa es perfecta».

El contexto de la conversación era que Sibelius, uno de los programas más utilizados para tipografía musical, desaparecía del mercado sin previo aviso, lo cual (aunque siempre es una mala noticia para sus usuarios y para la edición de partituras en general) podía ser una oportunidad para atraer la atención de un público nuevo hacia las soluciones de la competencia. Entonces, como se lee arriba, surgió la cuestión de si podía probarse la pretendida superioridad de LilyPond que sus partidarios preconizan. En otras palabras, salir del ámbito de la especulación e iniciar unas bases académicas firmes para la validez de esta herramienta.⁶

Observamos que las partituras, los materiales de notación musical, son de manejo tan habitual en educación musical como lo son los números en una clase de matemáticas o las palabras en una clase de lengua. Con esta encuesta y el experimento que contiene, tendremos la oportunidad de recabar de los profesionales, los profesores y los alumnos una valiosa información acerca del uso que hacen de la partitura, ya sea como productores o como consumidores.

La tarea de producir partituras, siempre en el campo de la educación, no es fácil. Se trata de un tipo de notación que requiere su estudio detenido (algo que se saldría de los límites de este trabajo; en el compendio bibliográfico hemos dado cuenta de la literatura al respecto). Las herramientas informáticas son una ayuda innegable, similar a la que aporta un editor de texto. Así como ocurre con otros programas de ordenador, un editor de este tipo no es nada más que una herramienta; pero debemos matizar dicha afirmación cuidadosamente. No es una herramienta como el pincel de un pintor o el arco de un violinista. Por supuesto, una herramienta de este tipo solo da buenos resultados en manos expertas. Con un programa informático sucede lo mismo, ya que el mayor rendimiento lo obtiene un usuario experimentado que conozca todos sus secretos. Sin embargo, en su diseño y en su funcionamiento se observa una característica interesante: para producir la notación musical, el programa busca automáticamente y de forma sistemática la mejor disposición para la música, tratando de

⁶En realidad la fascinación del autor por la producción de partituras de alta calidad y su puesta al alcance de los usuarios normales, se remota mucho más atrás en el tiempo: fue probablemente al leer el número de abril de 1982 de la revista *Scientific American*, en la que aparecía un artículo visualmente sorprendente y de efecto casi hipnótico, ilustrado prácticamente en exclusiva con partituras de Chopin (Hofstadter, 1982). Las partituras estaban hechas con el primitivo sistema de la Universidad de Indiana denominado SMUT, obra de Donald Byrd (Byrd, 1974).

relevante al usuario (experto o inexperto) de la tarea de distribuir y espaciar los distintos elementos de notación, de manera que formen un conjunto que sea a la vez técnicamente correcto y estéticamente agradable a la vista. En la práctica, el llamado *grabado automatizado* persigue que las partituras sencillas sean extremadamente fáciles de elaborar y estén listas para la impresión con un mínimo trabajo, y que la composición tipográfica de partituras complejas se beneficie también de estos automatismos para que la tarea sea más cómoda.

Con esto queremos decir que, si una partitura original se recrea con el programa, y dicha recreación resulta *estadísticamente* aceptable según la opinión de una serie de músicos voluntarios, no será debido en primer lugar a que el ejemplar esté hecho por un usuario más o menos experto, sino que el programa ayuda a que el resultado se pueda presentar tal como está sin necesidad de muchos retoques adicionales. Obviamente, se deberán documentar detalladamente todas las intervenciones manuales que sea necesario hacer sobre las partituras utilizadas como modelos, como parte del estudio y en favor de la repetibilidad del mismo. Así se ha hecho en el apartado 11.5.3 (pág. 223) del capítulo de Metodología.

Si el primer trabajo describe el proyecto desde ángulos diversos, el segundo lo utiliza como un producto completo, sin cuestionarnos cómo ha llegado a existir ni cuál ha sido su historia, y emplea los documentos producidos mediante aquél para compararlos con otras partituras existentes, extraídos de situaciones reales. De esta forma se trata de probar por primera vez que existe al menos una aplicación libre cuyos resultados pueden igualar o superar el nivel de aceptación que los músicos exigen a las partituras.

Ambos estudios se complementan para dar un primer apoyo académico al software libre de edición de partituras musicales.

Interés

Un estudio sobre dinámica del software libre tiene el interés de contribuir a aclarar en lo posible las interioridades de este mundo: cómo se desarrolla, quién lo hace, cómo lo hace, qué mecanismos intervienen.

El interés personal y profesional en los dos estudios proviene de varias fuentes:

- La actividad personal del autor en el manejo de ordenadores desde principios de la década de 1980; su ocupación laboral como profesor de música desde 1993 y de informática musical desde 1994; su implicación en proyectos de software libre desde 1996, y en el proyecto LilyPond desde 2006.
- La búsqueda de la corrección y la claridad en la notación musical.
- La atracción por las técnicas de impresión, diseño y tipografía.

- Psicológicamente, tiene interés lo gratificante del estudio de unos temas tan complejos e interdisciplinarios como la informática, la música y la libertad de uso y disfrute de programas informáticos, que sólo en nuestros días comienzan a valorarse por parte de las administraciones.

Además se aprecia un interés social, en la medida en que esta investigación trata de arrojar algo de luz sobre unos temas, el software libre y la impresión de partituras musicales, que están siendo objeto de creciente interés en los ámbitos educativo, profesional e institucional.

Para la comunidad educativa, tiene interés acercarse a una solución para el problema de la tipografía, que se describe en el apartado de fundamentación teórica. Si los docentes y los estudiantes tuvieran en sus manos, sin ninguna traba económica ni ética, una herramienta de producción de partituras de alta calidad, ello solo podría redundar en una mejora de la educación musical.

Los estudios sobre transcripción, conservación y difusión del patrimonio musical se pueden beneficiar de los avances en la edición de partituras con sistemas de carácter abierto, democrático y transparente.

El interés general de un trabajo de estas características se basa en la mejora de la calidad de la enseñanza musical por las siguientes razones:

- El tema de estudio se centra en la notación musical y su impresión o publicación electrónica.
- El software libre posee propiedades que lo hacen idóneo para la educación, entre ellas su frecuente actualización, sus profundas posibilidades de personalización y su bajo coste.
- Los valores del software libre ayudan a una educación que promueva y defienda el espíritu de cooperación y la libertad del ser humano.

Marco teórico e histórico
Software libre, tipografía musical y educación

1. Compendio bibliográfico

En esta sección aparecen comentadas tanto las referencias consultadas para el desarrollo de la investigación como las que se han utilizado para la elaboración posterior de la memoria.

Las principales bases de datos que se han consultado son (en orden alfabético):

- ACM Digital Library
- DBLP computer science bibliography
- Google Scholar
- Harmonie Park Press Music Index
- Journal Citation Reports (JCR): Science Edition
- JSTOR
- Library Music Source
- NIH
- Oxford Music Online
- Rebiun/CRUE
- ResearchGate
- ScienceDirect
- Scopus
- Springer

La mayoría están en inglés; los términos de búsqueda empleados como punto de partida incluyen los siguientes: *music typography, music engraving, computer music engraving, music engraving software, free software, music education, music reading, music notation*.

Los filtros de idioma se han establecido a inglés y español pero algunas referencias están en alemán y francés; los artículos de Filgueiras y Azevedo (Portugal) están en inglés. Otros filtros como la fecha han variado de acuerdo con los apartados correspondientes y es imposible fijar un solo criterio. Especialmente, los mejores libros sobre la notación musical tienen ya algunos años pero describen un sistema que no ha cambiado significativamente desde el s.XVIII, y prácticamente nada desde el s.XIX. No se ha filtrado por año en temas relacionados con la historia del software de tipografía musical o de software libre, ya que son términos de finales del s.XX que interesa conocer desde sus orígenes. Los libros que contienen descripciones de métodos estadísticos, en ocasiones tienen también muchos años; por ejemplo, el método Score de Wilson para los intervalos de confianza en procesos de validación se publicó en 1927.

Para la discusión de resultados se ha filtrado inicialmente por fecha posterior al año 2000 aunque alguna referencia anterior también ha sido incluida por su interés.

En el campo del software de tipografía musical existe un problema a la hora de buscar documentación relativa, porque los programas adoptan nombres que son excesivamente comunes en la terminología. Vamos a examinar estos cinco ejemplos: *SCORE*, *Amadeus*, *Finale*, *Sibelius*, y *Encore*. Dentro de la cultura de la edición musical, está claro que los cinco son programas editores. Sin embargo:

- *Score* es “partitura” en inglés, y también “puntuación”.
- *Amadeus* es el nombre de Mozart.
- *Finale* (en italiano) es el último movimiento de una sinfonía, y en inglés se usa también para el último capítulo de una serie cualquiera.
- *Sibelius* fue un famoso compositor finlandés.
- *Encore* (en francés y en inglés) es una repetición o una reposición.

Esto obliga a elaborar filtros de búsqueda bastante complejos y a eliminar manualmente muchos falsos positivos.

La figura 1 muestra la clasificación de la bibliografía, por tipos de publicación.

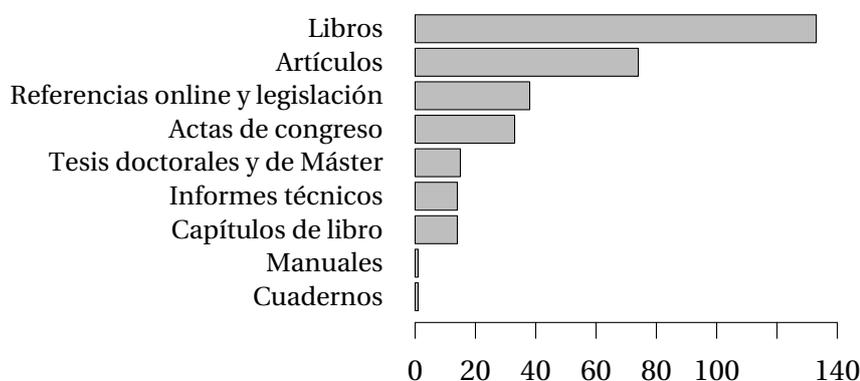


Figura 1: Bibliografía utilizada, recuento por tipos.

El manual de estilo de citas es el de la American Psychological Association y col. (APA, 2011), 6ª edición; en esta guía, las especificaciones sobre citas y referencias ocupan más de 60 páginas. Para la preparación del documento se creó una base de datos bibliográfica con Biblatex.¹ Al no estar disponible en Biblatex la funcionalidad del estilo APA para el idioma español, se hizo necesario un módulo de Biblatex nuevo² con las reglas correspondientes. Los datos se

¹Biblatex versión 2.9a (Lehman y col., 2006), de Philipp Lehman (con Philip Kime, Audrey Boruvka y Joseph Wright) es software libre licenciado bajo la GPL; lo mismo puede decirse del resto de los programas utilizados en esta sección.

²El módulo original del estilo APA para Biblatex es de Philip Kime; las reglas para el idioma español son del autor. El módulo anterior de la Red de recursos de T_EX (CTAN) estaba basado en el idioma alemán.

pasaron a JabRef³ para después exportarlos a formato de texto CSV; el análisis y los gráficos del presente compendio bibliográfico están hechos con el programa R⁴ a partir de este formato.

En la figura 2 aparece el tramo de fechas posteriores a 1980 para mostrar en detalle el recuento de las referencias bibliográficas de este período.

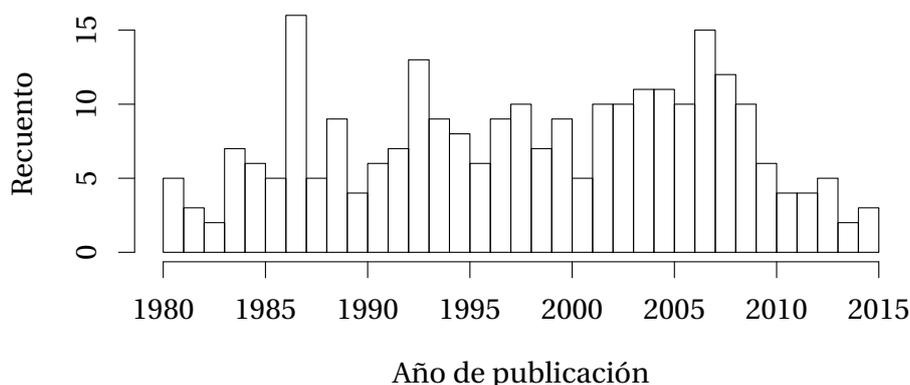


Figura 2: Año de publicación de las referencias bibliográficas posteriores a 1980.

Las publicaciones anteriores al año 2000 son fundamentales para entender la historia del software de tipografía musical. El año con más publicaciones es 1987, no solo porque proliferan los informes técnicos de la Universidad de Ohio, obra de Gourlay y otros pioneros sobre algoritmos de tipografía musical, sino también porque los libros más importantes sobre notación musical en general son de este año. Dentro del s.XXI, el año 2007 es el que tiene referencias más numerosas, sin una temática única que pueda hacerse responsable. En ese año se celebró la Conferencia Internacional de Software Libre en Badajoz.

1.1. Generalidades sobre investigación y metodología

Algunas referencias en este campo son Montero y León (2007), que es útil para la clasificación del método, Bell (2002), Booth, Colomb y Williams (2001), Currás (1985), Eco (1993), de Landsheere (1986), López, Ros, Orera, Fernández y Prat (2005) y Sierra Bravo (2007), entre otros.

Las definiciones están tomadas del diccionario de la Real Academia Española (RAE, 2003) y del *Collins concise dictionary and concise thesaurus* (Sinclair y Gilmour, 2002).

Aiken (1980, 1985), Candela (1992), Miller y Penfield (2006), Penfield y Giacobbi (2004) y Wilson (1927) se han empleado como documentación para temas de validación.

Se emplearon los trabajos de Eastman Kodak Co. (1999) y de Ray (2000) como referencia para

³JabRef versión 2.5 (JabRef Development Team, 2015).

⁴R versión 3.1.1 (R Core Development Team [R-CDT], 2000).

la recopilación de datos de iluminación de la escena para el experimento comparativo del estudio 2.

1.2. Estadística y gráficos estadísticos

Las referencias generales utilizadas han sido Agrawal, Imielinski y Swami (1993), Agrawal y Srikanth (1994), Agresti (2007), Spiegel (1991), Spiegel, Schiller y Srinivasan (2007), Presnell (2000) (que reproduce los ejemplos de Agresti (2007)) y Meyer, Zeileis y Hornik (2005).

Otras referencias son Tabachnick y Fidell (2007) para el análisis multivariado; la minería de datos y el software estadístico se estudian en títulos generales como Guisande (2006), Murphy (1999), Venables y Ripley (1997) y Hernández, Ramírez y Ferri (2004). Borgelt y Kruse (2002) es un título sobre reglas de asociación, mientras que Calle y Sánchez-Espigares (2007), Rojo (2006) y Therneau, Atkinson y col. (2014) estudian los árboles de clasificación. Los modelos log-lineales están tratados en Christensen (1997b) y Simkiss, Ebrahim y Waterston (2012). También se utilizó la referencia estándar del programa R (R-CDT, 2000). Ciertas referencias que justifican específicamente los estadísticos utilizados se citan directamente en el apartado correspondiente.

El código de los guiones de R desarrollados para el análisis de los datos se ha preparado con el entorno de desarrollo RStudio, descrito en Racine (2012). Se ha empleado como referencia en este ámbito el libro de Gandrud (2013) sobre investigación reproducible utilizando R y RStudio.

La combinatoria para estadística se trata en Erdős (1995); la estadística para pronósticos se estudia en Hyndman y Athanasopoulos (2013), y el análisis de varianza en Tejedor (1999).

Los gráficos de mosaico empleados en el segundo estudio están descritos en muchos artículos. Los más citados son de Friendly (1992, 1994, 1999), de Hofmann, Siebes y Wilhelm (2000) y de Huh (2004). Para los modelos de regresión local se ha consultado el libro de Linton (2008).

Se han empleado los manuales de Reas y Fry (2008, 2007) como referencias para la visualización de datos mediante el programa Processing, empleado en el primer estudio, y la documentación de Python-PyX (Lehmann y Wobst, 2015) que se ha usado para los guiones de Python que recorren la historia.

Meyer y col. (2005) es la referencia para los estadísticos de asociación.

Casi todas las ilustraciones se han confeccionado gracias al paquete TikZ de gráficos nativos para \LaTeX (Tantau, 2015). Esta inmensa referencia de 1165 páginas ofrece soluciones para la mayoría de los problemas relativos a la preparación de gráficos esquemáticos.

1.3. Control de versiones

Para ilustrar el mecanismo del control de versiones, que es esencial para documentar el registro histórico de un proyecto de software, se han buscado referencias históricas. Los primeros antecedentes conocidos en este campo son Rochkind (1975) y Tichy (1982, 1985). La referencia de Git como software es Torvalds y Hamano (2005) y su documentación es Greaves, Hamano, Torvalds y col. (2010). Los libros más citados que tratan específicamente del control de versiones con la herramienta Git son Swicegood (2008) y Loeliger (2009).

La visualización del control de versiones tiene su propio Simposio organizado por la ACM. En lo que respecta a nuestro estudio, las referencias principales han sido Ogawa y Caudwell (Caudwell, 2010; Ogawa y Ma, 2009, 2010).

1.4. Software libre y educación

Se cita en la fundamentación teórica un libro histórico sobre tecnología básica del software en general (Wirth, 1975) y uno de los escritos más citados específicamente sobre software libre (Raymond, 1999).

Son imprescindibles, en español, la *Introducción al Software Libre* de González-Barahona, Seoane y Robles (2003) y *La Pastilla Roja* de García (2003).

En relación con el tema del estudio, el software libre y sus metodologías de desarrollo se tratan en Christensen (1997a), Flores (2008), Gago (2010), Hardings (2004), Mas i Hernández (2005), Nienhuys (2006), de la Orden (2013), Percival (2010), Vercelli (2006), documentos (como Microsoft Corporation, 2008), artículos periodísticos (como el de Eunjung-Cha, 2002) y, por supuesto, los escritos de Stallman (1998, 2004, 2007, 2008).

En la Comunidad Autónoma de Extremadura, que fue pionera en el empleo del software libre en educación, algunos autores han tratado el tema como Abella y col. (2003), Gómez Deck (2005) y Márquez (2007).

Las leyes de la evolución de los sistemas de software han sido examinadas recientemente por González-Barahona, Robles, Herraiz y Ortega (2014).

Las tesis de Máster de Torres (2013) sobre los orígenes de GNU, con documentos y referencias del máximo interés histórico, y de Arjona (2012) sobre traducción de software libre, ofrecen panorámicas a distintos niveles de detalle sobre aspectos concretos del software libre, y son además interesantes puntos de partida para facilitar el acceso al desarrollo cooperativo de usuarios no técnicos. Otra tesis que conecta directamente con el objeto de nuestra investi-

gación es Gorhan, Hettinger, Schulz y Wolter (2012), en la que se desarrollan modelos para evaluar la madurez del software libre.

En relación con la educación, el software libre es tema de estudio en Adell y Bernabé (2007), Caba (2009), Cabero y Llorente (2007), Percival, Wang y Tzanetakis (2007), Robine, Percival y Lagrange (2007) y Villenave (2009a). La integración, la igualdad y la diversidad en grandes proyectos como Debian (posiblemente la mayor y más influyente distribución de GNU/Linux) se discuten en Robles, Arjona, Serebrenik, Vasilescu y González-Barahona (2014).

Se han citado referencias sobre las licencias del software de Pérez y Sánchez-Crespo (2006, 2009). Un manual de la era *pre-Windows* (Ramos, 1991) cubre desde un punto de vista novedoso algunos problemas cotidianos y universales del usuario de software. Dos intervenciones propias abordan el uso de software libre en educación musical (Vila, 2001, 2013).

1.5. Tipografía

Son muy numerosos los títulos generales sobre tipografía y software de tipografía; algunos que ofrecen amplia información sobre este arte, son los tratados de tipografía básica Baines y Haslam (2005), Biggs (1973), Chappell (1973), Frutiger (2002), el clásico *An Essay on Typography* (Gill, 1931), Kane (2005), Luidl (2004), además de Swann (1990) para diseño gráfico, tesis como Fuentes y Huidobro (2004) y el práctico *Primeros auxilios en tipografía* de Willberg y Forssman (2002) acerca de problemas comunes en esta materia.

Más específicos sobre software de tipografía, son los textos de Goossens, Mittelbach, Rahtz, Roegel y Voß (1997, 1994), Knuth (1979, 1992), Kopka y Daly (1995), Lamport (1994) y en español Cascales, Lucas, Mira, Pallarés y Sánchez-Pedreño (2000) y Sanguino (1997) sobre \LaTeX .

1.6. Notación y tipografía musical

temas de la investigación es la belleza y el atractivo de la notación musical. Varios autores sugieren que la escritura musical es más efectiva cuando su lectura es más placentera. Es curiosa la dificultad de encontrar referencias relacionadas con la belleza de la notación musical, ya que la búsqueda de diversas combinaciones de los términos *belleza* y *partituras musicales* suele remitir a artículos sobre la belleza de la música, que se aparta de nuestro tema, o a lo sumo, sobre obras musicales cuyo título tiene alguna relación con la belleza, como el musical “La Bella y la Bestia” de Alan Menken o “La Bella durmiente” de Tchaikovski. Si se busca por los términos “*music notation*”; *aesthetics*, los resultados están más relacionados con las enseñanzas musicales como educación estética, y con la estética en general, que con la estética de

la notación en sí. Montel (1997) (en francés) es de los pocos libros que se ocupan de la estética de la notación. Sloboda (1984) trata también aspectos específicos de la lectura musical.

Como el primer libro de música impreso con tipos móviles suele indicarse el *Odhecaton* de Ottaviano Petrucci, con un facsímil crítico en Petrucci (2001). Sin embargo, hay matizaciones importantes en Kostylo (2008). Es muy extensa la bibliografía existente que estudia la notación musical. Veamos algunos de los títulos más citados:

- Gould, con *Behind Bars: The Definitive Guide to Music Notation* (Gould, 2011), ha escrito posiblemente la obra más importante de los últimos tiempos sobre notación musical.
- Schirmer (2001) es una conocida guía de estilo, muy completa.
- Rastall (1983) es una interesante referencia de la evolución y el origen de de la notación común a partir de la notación neumática.
- Jofré (2003) abarca desde los orígenes hasta la actualidad con profusión de ejemplos gráficos, aunque no profundiza en los problemas de la tipografía.
- El *Essential Dictionary of Music Notation* (traducción del *Diccionario esencial de la notación musical*) de Gerou y Lusk (1996, 2004) es una obra de referencia ampliamente citada: se trata de una lista concisa y ordenada alfabéticamente de todos los problemas de composición tipográfica y notación musicales, que cubre la gran mayoría de los casos normales.
- Otra obra estándar sobre notación musical es *Music Notation: a Manual of Modern Practice* de Read (1969).
- Ross (1987) trata sobre el grabado musical, es decir, la composición tipográfica profesional. Contiene instrucciones sobre el estampado, uso de las plumillas y convenciones de la notación; también son interesantes las secciones sobre los aspectos técnicos y la historia de la reproducción.
- Gamble (1923) y Hader (1948) se centran en los detalles técnicos del grabado en metal.
- Johnson (1946) y Heussenstamm (1987) se centran en la música manuscrita.
- Heussenstamm cita a su vez el libro *Music Notation in the Twentieth Century* (Stone, 1980) y lo califica como «el libro sobre notación musical más importante de los últimos años» (con permiso de Gould, 2011, que es posterior).

The Gale Group (1998) trata generalidades sobre notación musical en tono didáctico, y Devore, Kulma, Naxer, Clark y Kuhns (2010) contiene una fuente de ejemplos musicales extraídos de un contexto educativo, útiles para el experimento del estudio número 2. Salazar (1989) es una fuente para la historia de la notación musical, con música manuscrita citada como ejemplo. López García (1977) es otro libro muy difundido entre los estudiantes de conservatorio que contiene interesantes ejemplos de música manuscrita.

La figura de Byrd es extraordinaria; algunos títulos son importantes por su oportunidad histórica en los comienzos mismos de la informática de la confección de partituras; algún escrito

suyo tiene además una significación especial para el autor. Byrd (1994) describe fragmentos de notación problemáticos cuestionando los programas de notación que tratan de exhibir un comportamiento inteligente. Sin embargo, también investigó sobre música, informática e inteligencia; en realidad toda la producción de Byrd se muestra de interés, ya sea en solitario o en colaboración: Byrd (1974, 1985), Byrd y Crawford (2002), Byrd e Isaacson (2003a, 2003b).

Además, se han consultado ciertas referencias con una importancia algo menor para nuestro estudio. Reinach y Agostini (1984), Solomon (1984), de Souza (2012), Wolf (1919) hablan de historia de la notación y la música en Grecia. Se han utilizado para documentar el ejemplo del epitafio de Sículo.

Button (1920), con *System in Musical Notation*, hizo uno de los muchos intentos de sistematizar completamente la notación musical. Boehm (1961), Chlapik (1987), Donato (1963), Donemus (1982), Mc Grain (1991) y Tyboni (1994) son títulos generales sobre cómo imprimir música, y Barksdale (1957) contiene una amplia historia de la impresión de notación musical. Rosenthal (1967), Wanske (1988), Weaner y Boelke (1993) y Vinci (1989) se ocupan de la notación y el grabado tradicionales. El artículo de Mele (2004) contribuye a la divulgación sobre historia de la impresión musical.

Apel (1961) se ocupa de la notación de la música polifónica. La notación de la música barroca está estudiada en la tesis de Máster de Le Gall (1997) y en Bouzaiene, Le Gall y Saint-James (1998).

La notación de acordes necesita de ciertas referencias obligadas como el conocido libro de Ignatzek (1995), que contiene el sistema de nomenclatura de acordes utilizado en LilyPond; Banter (1987) es otra guía importante acerca de la notación de acordes. Hofstadter (1982) recopila artículos de la revista *Scientific American* y alberga un capítulo que se ocupa entre otras cosas de las paradojas de la notación. Por su parte, el trabajo de Jaschinski (2000) contiene referencias sobre notación alternativa y de la música antigua.

Lamb (1996) era un recurso universitario en línea (no disponible actualmente) sobre notación, aunque centrado en el uso de Finale.

MacLachlan (1998) y Wright (2014) hablan de la notación de música para guitarra. El primero, además, hace algunas consideraciones generales sobre el aprendizaje de la notación, y el segundo (no disponible actualmente) presentaba un tipo de notación alfanumérica sustitutiva de la tablatura o notación de cifra sobre seis líneas.

Read (1978) estudia la notación rítmica contemporánea. Roemer (1984) y Rosecrans (1979) se centran en el campo comercial y Belkin (1994) realiza un análisis comparativo de varios sistemas de notación experimentales.

1.7. Otras obras sobre software de tipografía y codificación musicales

Smith (1973) presenta el histórico programa SCORE en la que parece ser la referencia más antigua sobre tipografía musical por ordenador. Assayaag y Timis (1986) plantea ciertas cuestiones generales sobre el empleo de la informática para la notación.

Blostein y Haken (1991, 1994), Dunker (2003), Haken y Blostein (1993) describen el editor LIME y otros antecesores de LilyPond. El texto de Mariano, Nienhuys y Nieuwenhuizen (1999) habla de Mudela, que fue la primera denominación de este lenguaje hasta que se hizo llamar simplemente con el mismo nombre que el programa. La fuente primaria para el código del programa es del año 1995, y por parte de los mismos autores se presenta en la academia con el artículo de 2003. Sandberg (2006) realiza una tesis que informa sobre su funcionamiento interno.

La fuente más importante de artículos sobre informática que hemos utilizado para el apartado histórico de antecedentes, es el archivo de Communications of the Association of Computer Machinery (ACM). Incluye los textos completos de todos los artículos de los últimos 50 años de esta publicación.

Rubenstein (1987) estudia en «A database design for musical information» las bases de datos para información musical, que fueron un campo de intensa actividad de investigación a finales de los años 90.

En el campo de los lenguajes informáticos de codificación musical son importantes el gran catálogo de códigos musicales de Selfridge-Field (1997) y Almeida, Carvalho y Oliveira (2011), Dannenberg (1993), Droettboom (2000, 2006), Hewlett y Selfridge-Field (1985, 2001), Loy y Abbott (1985), Sloan (1993), Wiggins, Miranda, Smaill y Harris (1993) como obras generales; Foxley (1987), Prather (1996), Raskin (1997) sobre métodos arcaicos para imprimir música; Balaban (1988) respecto al software musical para ayuda de la composición; Hoos, Hamel, Renz y Kilian (1998, 2001), Renz (2002) sobre el sistema GUIDO de notación musical, mientras que Huron (1993), Knopke (2008) y Langston (1990) describen herramientas para composición y manejo simbólico de la música.

Erickson (1975) y Lincoln (1977) describen el lenguaje DARMS para indexar música del renacimiento. La tesis de Byrd (1985) describe el sistema SMUT, y la tesis de máster (compartida) de Steinbach y Schofer (1987) describe el lenguaje MuTeX; Dannenberg (1989), el lenguaje de partituras Canon; Azevedo y Almeida (2013) y Walshaw (1993) la codificación con ABC, y Laurie (2005) el sistema M-Tx. Hegazy (1987), Parrish, Hegazy, Gourlay, Roush y Sola (1987) y Roush (1987) se ocupan del sistema MusiCopy. Ornstein y Turner (1983, 1984) se centran en Mockingbird, el llamado “amanuense” del compositor; los artículos de Cindy Grande (Grande, 1995, 1997) describen formatos binarios para la notación.

Bellini, Fioravanti y Nesi (1999) y Bellini, Nesi y Spinu (2002) justifican la necesidad de medidas para gestionar la música de forma eficiente y proponen sistemas colaborativos de partitura electrónica.

Byrd (2005) y Field-Richards (1993) están dedicados al lenguaje Cadenza de descripción musical; los trabajos de Filgueiras en el Laboratorio de informática e inteligencia artificial (LIACC) de la Universidad de Oporto (Filgueiras, 1996, 1998; Filgueiras y Leal, 1993) desarrollaron algoritmos de composición musical y el sistema ScEX; Cataldi (2003), Simons (2004) están dedicados al preprocesador PMX. Roads (1979) y Sinclair, Droettboom y Fujinaga (2006) son tratados sobre búsqueda de gramáticas y representaciones o codificaciones musicales; Goldfarb (1992) discute formatos de representación musical para la web, y Prather (1996) investiga el análisis armónico a partir de representaciones musicales.

Los libros de Grøver (1989a, 1989b, 1989c), Parrish y Gourlay (1987), Roelofs (1991), Smithers (2000), Taupin (1993), Taupin, Mitchell y Egler (1995) abordan el diseño de fuentes tipográficas, símbolos musicales y establecen reglas para la disposición polifónica. También Martin (1998) explora el diseño de tipos musicales; el cuaderno de la Music Publisher's Association (MPA, 1996) establece además una serie de reglas oficiales para la publicación de música.

La separación de los símbolos y su distribución sobre el pentagrama motivan los textos de Gomberg (1975, 1977) y de Gourlay (1987), que investigan los algoritmos de justificación de líneas de música.

La confección de los modelos de partitura recreados tienen como referencia, entre otros, los informes técnicos de Sola (1987) y de Sola y Roush (1987) que tratan el cálculo de ligaduras y barras de corchea.

Tratan específicamente el software interactivo de tipografía musical Giesecking (2001) y Müller (1990).

1.8. Discusión de resultados y prospectiva

Los artículos que se han consultado específicamente relacionados con los resultados obtenidos sobre la ley de Zipf han sido Manaris, Vaughan, Wagner, Romero y Davis (2003) y Richter (1958); Christman y Pinger (1997) y Levy (2002) sobre asuntos de dominancia lateral; Debès, Schneider y Malchaire (2004), Pangert y Looock (2004) sobre iluminación de la escena; Knauff y Nejasmic (2014) sobre sistemas de preparación de documentos; Siebra, Anjos y Rolim (2014) sobre desarrollo del software; Page (1988), Uitdenbogerd y Yap (2003) sobre legibilidad de partituras y recuperación de la información musical; Purchase, Alder y Carrington (2002) sobre estética de la notación no musical, y Chen y Zhang (2007), Davis y Bostrom (1992), Gram y Cockton (1996) y Rauterberg (1992) sobre usabilidad de programas informáticos.

La generación de ejemplos para no depender de material protegido por copyright también puede usar un sistema similar al descrito en el artículo de Rader (1996) y otros como Anders (2007), Anders, Anagnostopoulou y Alcorn (2005), Lagrange, Percival y Tzanetakis (2007) y Pestana y Pestana (2012) que emplean Strasheela, cadenas de Markov y otros sistemas de composición mediante reglas informáticas para la generación de música y producción automática de ejemplos musicales.

1.9. Cuestiones de propiedad intelectual

No todas estas referencias están citadas explícitamente en el texto, pero los asuntos que se tratan requieren tener presentes al menos los textos de la Organización Mundial de la Propiedad Intelectual (OMPI, 1967, 1989, 1994) que contienen los convenios firmados en Estocolmo y Berna y otros documentos. Una intervención propia trata el problema de la liberación de las partituras en el mundo digital (Vila, 2012). La referencia López-Sintas, Cebollada, Filimon y Gharhaman (2014) ofrece resultados sobre consumo de material musical. ColorIuris (2008) ofrece licencias alternativas; Liebowitz y Watt (2006) repasa la historia del copyright, y Vives (2006) estudia las generalidades, los límites y las excepciones de las leyes de propiedad intelectual.

2. Software libre y control de versiones

En este estudio nos ocupamos de un proyecto de software libre, siendo éste una realidad cuyos orígenes, dentro de la joven tecnología de la informática, data de los años 60; es fundamental delimitar previamente el concepto de software libre para poder discutir posteriormente acerca de sus implicaciones.

2.1. Conceptos

Las ideas que se desarrollan aquí brevemente están enmarcadas dentro de la informática fundamental y su terminología.

2.1.1. Programas y código fuente

Un programa de ordenador consiste en una larga serie de instrucciones que el procesador ejecuta. El programador (el artífice del programa) emplea como punto de partida unos **algoritmos** que son la expresión de los procedimientos que sigue el programa, y de unas **estructuras de datos** que son sistemas de organización de la información.

Un texto fundamental de informática lo dice en el mismo título:

Algoritmos + estructuras de datos = programas (Wirth, 1975)

El programador emplea un cierto **lenguaje de programación** para expresar en él los algoritmos y las estructuras de datos que constituyen el programa. Llamamos **código fuente** a este texto que el programador elabora en el lenguaje de su elección.

Para cada lenguaje de programación existe su correspondiente **intérprete** o **compilador** que traduce el código fuente del programa en otro que el procesador de la máquina puede ejecutar directamente: el **código máquina**.

Los intérpretes se diferencian de los compiladores en que el compilador traduce todo a código máquina, antes de que la máquina lo ejecute; el intérprete, en cambio, traduce las instrucciones una a una *sobre la marcha* mientras hace que la máquina vaya ejecutando las partes traducidas.

La solución más común que algunos lenguajes emplean es la del compilador JIT o *justo a tiempo*: el texto del programa se compila de forma transparente para el programador, solamente la primera vez a partir del momento en que código ha sido modificado. La segunda vez y siguientes, se ejecuta el código compilado. Adoptan este enfoque lenguajes muy populares como Java y Python (Hugunin, 1997).

El código máquina es específico de cada tipo de procesador, aunque hay familias de procesadores que pueden compartir el mismo código máquina, lo que ofrece una cierta compatibilidad *hacia abajo*, según la cual los procesadores más modernos son capaces de ejecutar el código máquina de procesadores anteriores de la misma familia. Los archivos de ordenador que constituyen programas listos para ejecutar se llaman archivos **binarios ejecutables**. El mismo código fuente puede producir los binarios ejecutables para distintos procesadores: sólo se necesita el traductor adecuado (intérprete o compilador).

De la informática fundamental en un sentido general, pasamos a referirnos ahora a la informática más cotidiana de los programas de oficina, educativos, etc. A excepción de los sistemas empotrados, comunes en el mundo industrial y en algunos dispositivos móviles, son raros los casos en que el programa contiene absolutamente todas las instrucciones que el procesador debe ejecutar. En lugar de esto, un **sistema operativo** o gran programa básico de una máquina, ofrece sus **llamadas del sistema** a los programas que desean funcionar dentro de la máquina en cuestión, bajo este sistema operativo. El sistema operativo coordina a todos los programas que corren en la máquina, entre sí y con los controladores de dispositivo, de forma que no se interfieran y obedezcan las indicaciones del **usuario**.

Dadas las definiciones breves que hemos presentado, se entiende que para producir un programa en forma de binario ejecutable, necesitamos su código fuente. Se entiende también que del examen de este código fuente (suponiendo que estamos versados en el lenguaje de programación en que aquél está escrito) podemos extraer los algoritmos que utiliza y la forma de sus estructuras de datos. Todo lo que el programa hace, debe estar descrito con absoluta precisión en un lugar u otro de su código fuente. Los programadores que quieren facilitar esta labor de revisión a otros, acompañan junto al código una **documentación** más o menos detallada. Idealmente la documentación y el código deberían bastar para que cualquiera pudiese comprender el programa a la perfección, mantenerlo y mejorarlo.

Si comparamos un programa con un plato de comida, podemos hacer las siguientes analogías: el plato está descrito en una **receta** que es el código fuente y su documentación. Los **ingredientes** y sus cantidades serían datos. Una **ración** de este plato, sería el binario ejecutable. Los **utensilios** son el sistema operativo y el **cocinero** es el compilador. Esta metáfora de la receta de cocina está empleada por Stallman (1998) para mostrar cómo el conocimiento se comparte a través del código fuente. Las instrucciones para freír un huevo pueden (y creemos que deben) estar bajo la libre disposición de todos; patentar una receta de cocina

puede resultar un absurdo, aunque algún famoso cocinero ya lo haya hecho. Cerrar la receta bajo siete llaves, puede provocar recelos en el consumidor. La realidad del software es que a veces los procedimientos están patentados y el código no suele estar disponible. El software libre surge precisamente para oponerse a esta práctica por entender que perjudica al usuario, frena la innovación e incide negativamente en la seguridad.

2.1.2. Software

Antes de abordar el concepto de software libre, que constituye un marco dentro del que se encuadra del objeto de nuestro estudio, en primer lugar debemos examinar la denominación de software. Hemos hablado de «programas»; el término «software» viene a ser un sinónimo.

Entendemos por *software* (RAE, 2003):

software (Voz inglesa) **1.** m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

El software, palabra que contiene la raíz *soft* («blando») se opone y complementa al *hardware*¹ que a su vez contiene la raíz *hard* («duro»), como los dos constitutivos fundamentales de un ordenador. La parte «dura» o tangible, los circuitos, cables que los unen y *chips* que los habitan, no puede funcionar sin la parte «blanda» o intangible, los programas y también los datos como información textual, imágenes y sonidos digitalizados, todo ello codificado en binario.²

Es común referirse a los dos términos expuestos, huyendo del empleo del inglés, como «equipo»³ y «programas» para *hardware* y *software*, respectivamente. Con frecuencia los programas incluyen numerosos datos preescritos: imágenes, documentación... Éstos también se consideran habitualmente como software.

La expresión inglesa *software program* es redundante al traducirla al español.

2.1.3. Libertad

El software libre combina conceptos relacionados con los programas de ordenador y con la libertad. Por *libertad* entendemos (RAE, 2003):

libertad. (Del lat. *libertas*, *-ātis*). **1.** f. Facultad natural que tiene el hombre de obrar de una manera o de otra, y de no obrar, por lo que es responsable de sus actos.||**2.** f. Estado o condición de

¹**hardware.**(Voz ingl.).**1.** m. Inform. Conjunto de los componentes que integran la parte material de una computadora (RAE, 2003).

²El sistema binario es un sistema de numeración *en base dos*, que sólo utiliza dos símbolos: el 1 y el 0.

³También «máquina» o simplemente «ordenador». En el español de América: «computadora».

quien no es esclavo.||3. f. Estado de quien no está preso.||4. f. Falta de sujeción y subordinación. *A los jóvenes los pierde la libertad.*||5. f. Facultad que se disfruta en las naciones bien gobernadas de hacer y decir cuanto no se oponga a las leyes ni a las buenas costumbres.||6. f. Prerrogativa, privilegio, licencia. U. m. en pl.||(...)12. f. Facilidad, soltura, disposición natural para hacer algo con destreza. *Algunos pintores tienen libertad de pincel. Ciertos grabadores tienen libertad de buril.*||

Con una definición del sustantivo libertad, se examinarán a continuación las definiciones del adjetivo «libre» según la RAE (2003), porque en ellas no se incluye la palabra «libertad», lo que es coherente con el principio de no incluir en la definición el término que se define. Con todo, estas definiciones aportan algunos elementos nuevos:

libre. (Del lat. *liber*, -*ĕra*). **1.** adj. Que tiene facultad para obrar o no obrar.||**2.** adj. Que no es esclavo.||**3.** adj. Que no está preso.||(...)9. adj. Exento, privilegiado, dispensado. *Estoy libre del voto.*||(...)11. adj. independiente. *El que no está sujeto a padres ni amos o superiores domésticos, es libre.*||12. adj. Desembarazado o exento de un daño o peligro. *Renta libre de penas, de cuidados.*||13. adj. Que tiene esfuerzo y ánimo para hablar lo que conviene a su estado u oficio.||(...)15. adj. inocente (||sin culpa).||(...)

Podremos apreciar el valor de la combinación de estos dos conceptos, el de software y el de libertad, cuando antes hayamos conocido el caso que se le opone, el de software propietario.

2.1.4. El software propietario

Como un producto a la venta y en competencia con otros productores, el software es con frecuencia un bien ofrecido por una compañía a un precio y bajo unas condiciones que están en la forma de una *licencia del usuario final*.⁴

Examinemos algunas de las cláusulas de la licencia de ciertos programas, por ejemplo las que siguen, extraídas de la licencia de Windows XP (Microsoft Corporation, 2008):

- **Instalación y uso.** *Puede instalar, utilizar, tener acceso, presentar y ejecutar una copia del Producto en un único equipo (...). El Producto no puede ser utilizado por más de un (1) procesador a la vez en un único Equipo. Puede permitir a un máximo de cinco (5) equipos (...) conectarse al Equipo estación de trabajo para utilizar los servicios del Producto sólo para servicios de Archivo e impresión (...)*
- **Copia de seguridad.** *Puede hacer una sola copia de seguridad del software. Puede usar la copia de seguridad exclusivamente para sus necesidades de almacenamiento y para reinstalar el software en el ordenador (...). En caso contrario, no podrá hacer copias del software (...) No podrá alquilar, prestar, ceder o transferir de cualquier manera el CD o la copia de seguridad a otro usuario.*

⁴La licencia del usuario final se conoce a veces como EULA por sus siglas en inglés, de *End User License Agreement* y en español como CLUF, de *Contrato de Licencia para el Usuario Final*.

- **Limitaciones en materia de ingeniería inversa, descompilación y desensamblaje.** *Usted no podrá utilizar técnicas de ingeniería inversa, descompilar ni desensamblar el Producto, (...)*
- **Separación de componentes.** *El software se licencia como un producto unitario. Sus partes constituyentes no se pueden separar para usarlas en más de un ordenador. (...)*

Ignoramos si existe algún estudio acerca del porcentaje real de usuarios que cumplen todas las condiciones de los acuerdos de licencia. Lo que es seguro es que *no siempre* se cumplen.⁵

2.2. El software libre

Parte del material de esta sección se basa en la selección de ensayos de Stallman (1998) y otras fuentes relacionadas como Williams (2011).

El software libre es el que incluye el permiso expreso, para cualquier persona, de usarlo, copiarlo y distribuirlo, bien sea con modificaciones o sin ellas. Para ello, el código fuente debe estar disponible. Esto es lo que conoce como **las cuatro libertades** del software libre. La palabra «Libre» aplicada al software, se refiere al usuario.

2.2.1. Libertad frente a gratuidad

El calificativo «libre» del software libre se refiere a libertad, no a gratuidad. Sin embargo, las compañías de software privativo emplean en ocasiones el término para referirse al precio. En inglés «software libre», se escribe *free software* y el término *free* puede significar «gratis» o «libre».

Muchos idiomas tienen dos palabras diferentes para *free* como libertad y *free* como «precio cero». El español, por ejemplo, dispone de los términos «libre» y «gratuito» (en francés sucede exactamente lo mismo con los términos *libre* y *gratuit*). En inglés existe la palabra *gratis*, que se refiere sin ambigüedad al precio, pero este idioma no dispone de un adjetivo común que se refiera sin ambigüedad a la libertad; por ese motivo hay que recurrir al enrevesado adjetivo artificial *Free as in freedom*, es decir, libre con el significado de libertad.

Dado que existen muchas categorías de software, para ilustrar de qué forma se delimitan las definiciones de sus tipos entre sí, véase la figura 3, según Chao-Kuei Hung tal y como aparece en Stallman (1998).

⁵Un apartado del libro “Tu segundo manual del Atari ST”, de Andreas Ramos, lleva por título: *El acuerdo de licencia del software: léalo y llóre*. (Ramos, 1991)

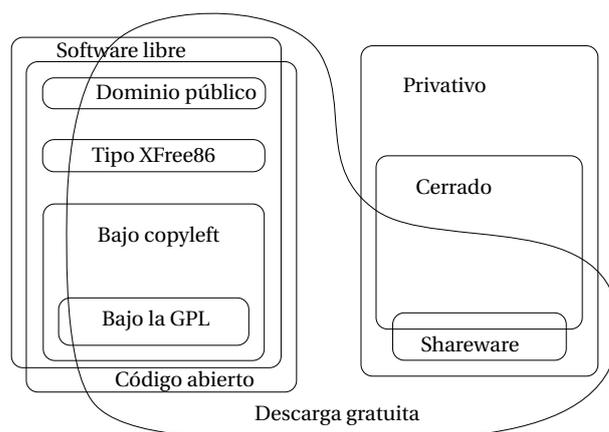


Figura 3: Las categorías del software respecto del concepto de libertad, según Hung.

La figura pone de manifiesto una serie de distinciones que es necesario describir. Se debe interpretar este gráfico como la serie de subconjuntos existentes dentro de todas las licencias que los programas de ordenador pueden ostentar, cuando se atiende al tratamiento de la libertad del usuario que cada tipo de licencia contempla. Para ver algunos ejemplos concretos de programas de tipografía musical adscritos a estas categorías, debe consultarse la figura 13 en la página 71.

En primer lugar, como muestra el gráfico, la descarga gratuita no tiene nada que ver con el carácter de libre o privativo, ya que por un lado el software libre se puede descargar por un precio, y por otro existe software privativo que se ofrece gratuitamente para su descarga.

Por ejemplo, el *shareware* siempre es de descarga gratuita, pero no es software libre. Obsérvese en el gráfico la situación de los programas de *shareware* a la derecha, dentro del apartado de software privativo.

De forma complementaria, uno de los primeros ejemplos de software libre que se estuvo vendiendo durante un cierto tiempo es el editor GNU Emacs.

La ilustración muestra que existe una dicotomía entre el software privativo, a la derecha, y un par de categorías *cuasi*-coincidentes como son el software libre y el software de código abierto, a la izquierda. Sobre esta sutil distinción se pronuncian González-Barahona y col. (2003) en su *Introducción al Software Libre*:

Equivalente a software libre es el término Open Source Software («programas de fuente abierto»), promovido por Eric Raymond y la Open Source Initiative. Filosóficamente, el término es muy distinto, ya que hace énfasis en la disponibilidad de código fuente, no en la libertad. (...) Este nombre es más políticamente aséptico y recalca un aspecto técnico que puede dar lugar a ventajas técnicas, como mejores modelos de desarrollo y negocio, mayor seguridad, etc. Fuertemente criticado por Richard Stallman y la Free Software Foundation (FSF),⁶ ha encontrado

⁶Fundación para el Software Libre.

mucho más eco en la literatura comercial y en las estrategias de las empresas que de una manera u otra apoyan el modelo.⁷

Una de las consignas del movimiento del software libre (promovido en sus orígenes por la FSF) es la de no llamar «cerrado» al software privativo, pues podría hacer pensar que si la fuente de un programa no está cerrada, entonces es *Open* («abierta»), y el movimiento Open Source no es el que la FSF promueve. En cualquier caso es significativo que las categorías comprendidas en el «núcleo duro» de la parte libre del gráfico (copyleft, GPL, etc.) son, al mismo tiempo, software libre y de código abierto.

2.2.2. La cláusula copyleft. La licencia GPL

Se dice que un determinado trabajo u obra está «protegido por copyright»; de forma similar, existe el concepto de *copyleft*, que protege a las obras que voluntariamente han sido depositadas bajo él por su autor. Afecta a la forma de distribución y puede usarse siempre y cuando no entre en conflicto con las libertades centrales.

El copyleft es una regla, o cláusula, que implica que al redistribuir la obra o el programa, no se puede añadir ninguna restricción que deniegue a otras personas las cuatro libertades centrales que veremos luego en la definición formal de software libre. El término copyleft, que parece una parodia del copyright, y efectivamente tiene algo de paródico, constituye un juego de palabras que aprovecha la afortunada relación entre los dos significados de la palabra *right*: «derecha» y «derecho». Cuando se sustituye por *left* («izquierda» y también «abandono») tenemos que el copyleft viene a ser un «izquierdo de copia» o «abandono de copia» opuesto a los «derechos de copia» del copyright. Todo software protegido bajo copyleft es libre, si bien existen otras categorías que respetan las cuatro libertades y también son software libre sin caer bajo este concepto.

No confunda el copyleft con el respeto a las libertades [del software libre]. Si las cuatro libertades se respetan, el software es libre, aunque no esté bajo licencia copyleft. Si está bajo licencia copyleft, entonces permanecerá Libre, porque el copyleft no sólo respeta las libertades, sino que también las defiende (Oliva, 2007).

También existe el software libre sin copyleft, aunque obviamente la posibilidad de que un desarrollo pueda revocar su licencia sin motivo, lo hace más vulnerable. Por ello, el uso de copyleft para una licencia es recomendable aunque se mantiene la libertad de usar software libre aunque no lleve esta cláusula. La regla copyleft protege las libertades centrales, aunque pueda parecer que entra en conflicto con ellas.

Vemos en el gráfico mencionado que existe una categoría de programas que se encuentran

⁷ *Op. cit.*, pág. 17.

protegidos bajo las siglas GPL. Todos ellos disfrutaban de una protección de tipo copyleft. Aunque existen otras licencias dentro del copyleft, la así llamada Licencia General Pública fue redactada por Richard Stallman y el equipo de abogados de la Fundación para el Software Libre con el objeto de dotar de validez jurídica al software libre y aplicarla a su gran proyecto de un sistema operativo completamente libre: el proyecto GNU.

Torres (2013), en su tesis de Máster, se extiende sobre los orígenes de GNU e incluye documentos y referencias del máximo interés histórico.

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia (véase OMPI, 1994, p. 145). El valor legal de esta licencia ya ha sido probado en los tribunales: el 13 de agosto de 2008, una corte de apelaciones de Nueva York dictaminó que el incumplimiento de las condiciones de una licencia libre es una violación de la propiedad intelectual;⁸ El escrito de Pérez y Sánchez-Crespo (2006) es una referencia esclarecedora en relación a la situación legal de estas licencias en España. La versión 3 de la licencia GPL se publicó el 29 de junio de 2007; el mismo equipo de abogados ha elaborado un estudio (Pérez y Sánchez-Crespo, 2009) sobre la nueva versión de la licencia.

Tenemos constancia de un caso único de aplicación de la licencia GPL (que por lo general se aplica solamente al software), a una ópera completa, la denominada «Ópera libre» de Valentin Villenave. Se dan más detalles sobre esta curiosa situación en el apartado 5.8 (pág. 106). El autor pudo haber elegido licenciar su obra bajo la cláusula copyleft, por ejemplo mediante alguna de las licencias de Creative Commons (cuya descripción escapa al ámbito de este trabajo). Sin embargo, la aplicación de la licencia GPL a una ópera es posible sólo gracias a que ésta se encuentra «codificada», es decir, descrita hasta el último detalle en un lenguaje formal de tipografía musical como LilyPond. Con el tiempo, esta relación entre licencias libres y obras artísticas (o distintas del software) dejará de sorprendernos.

Una definición completa y canónica del software libre puede verse en los ensayos de Richard Stallman (Stallman, 1998) que son una referencia obligada, y de los cuales se han adaptado las consideraciones de esta sección. Debemos recordar que no hay un concepto monolítico y universal alrededor de este tema y existe al menos una gran alternativa a la definición «oficial» de Stallman, ya que como él mismo cita, el movimiento *Open Source* («fuente abierta») se opone a los postulados de la FSF en algunos aspectos clave.

El software libre es una cuestión de libertad, no de precio. Para entender el uso que se da aquí a la palabra, debemos pensar en la palabra libertad con el sentido que tiene dentro de la frase «libertad de expresión» (*free speech*). Como se ha dicho anteriormente, en inglés, la misma palabra (*free*) significa tanto «libre» como «gratis», lo que ha dado lugar a cierta confusión. La

⁸Reseñado por David Maeztu, de la Fundación copyleft, en su blog <http://derechoynormas.blogspot.com/2008/08/importante-sentencia-sobre.html>

necesidad de dejar esto más claro de lo que la palabra inglesa permite, ha creado curiosos acrónimos para referirse a todo software remotamente libre más allá de “OSS”, *Open Source Software*, como “FOSS”, de *Free/Open Source Software* o “FLOSS”, de *Free-Libre-Open Source Software*.

Formalmente, el software libre viene definido por las **cuatro libertades** de que disfruta el usuario. Son las siguientes:

- **Libertad cero.** La libertad de usar el programa con cualquier propósito.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier tipo de tarea, y sin la obligación de tener que comunicárselo al desarrollador ni a alguna otra entidad específica. La redacción de la libertad cero deja fuera, en principio, iniciativas del tipo de la llamada «cláusula mariposa»,⁹ que impide ciertos usos del software como por ejemplo todo aquel que menoscabe los derechos humanos. Otro caso es el de la licencia JSON, que especifica que el software *debe usarse solamente para el bien y no para el mal*. Los críticos aducen que estos usos ya están prohibidos por las leyes internacionales sin necesidad de otra cláusula en la licencia del software utilizado. En la práctica, las licencias que contienen estas cláusulas son, sencillamente, incompatibles con la GPL.¹⁰

- **Libertad uno.** La libertad de estudiar cómo funciona el programa, y adaptarlo a las necesidades de cada usuario. Para esto es condición previa el acceso al código fuente.

La consecuencia directa es la transparencia y la seguridad. Ejemplos obvios de software inherentemente inseguro por opacidad es el de las máquinas de votación y los sistemas de cifrado para el comercio electrónico, aunque todo el software en circulación es más inseguro para el usuario cuanto más se base en la opacidad para protegerse.

- **Libertad dos.** La libertad de distribuir copias.

Se entiende que el software libre es difícil de vender si en su licencia va incluido el permiso para distribuirlo libremente. Los negocios basados en software libre no se basan en tratar de vender algo que uno puede conseguir en alguna otra parte, sino en ofrecer servicios o desarrollos a medida.

- **Libertad tres.** La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. Para que tenga algún sentido esta libertad, es una condición previa el acceso al código fuente.

⁹El sitio oficial de esta propuesta era clausulamariposa.es, visitado el 2 de junio de 2013. No disponible actualmente.

¹⁰Que sepamos, el software privativo no impone estas restricciones en sus acuerdos de licencia.

En la línea del comentario anterior puede decirse que incluso el valor de un desarrollo a medida no está menoscabado por el hecho de que el producto esté disponible para la comunidad; más bien esta disponibilidad puede tener beneficios no previstos como que el programa se traduce por parte de usuarios de otros idiomas, lo que puede acrecentar su difusión.

En resumen, las cuatro libertades implican que el usuario no tiene que pedir permiso para ejecutar, copiar, distribuir, estudiar, cambiar o mejorar el software. En cambio, una empresa no puede legalmente emplear software libre distribuido como parte de un producto sin publicar también el código.

Las modificaciones privadas sobre el código del software no obligan al autor a anunciarlas o publicarlas; pero si lo hace, debe acompañar siempre el código de las modificaciones.

Las libertades son, además, irrevocables (en condiciones normales dentro de la legalidad): no sería libre según esta definición, una licencia que permitiera al desarrollador revocarla sin haberle dado motivos para ello.

El derecho a vender copias de software libre tampoco entra en conflicto con las definiciones anteriores. El ejemplo que suele ofrecerse más a menudo es el de aquel que posiblemente fuera el primer desarrollo de software libre: el editor Emacs, que R. Stallman estuvo vendiendo durante algún tiempo. De esta manera «software libre» no significa «no comercial». Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante porque muchas empresas que tienen como modelo de negocio la venta de soporte, están involucradas de forma muy beneficiosa en el desarrollo del producto para el que ofrecen este soporte, incluso en un mundo digital donde la copia es fácil. Es el caso de Red Hat, que es uno de los mayores contribuyentes a la evolución del núcleo Linux y vende la distribución Fedora con soporte, aunque es de descarga gratuita.

Cuando se habla de software libre, según aparece en los escritos que analizan sus implicaciones, se considera que es mejor evitar términos como «regalar» o «gratis», porque esos términos implican que lo importante es el precio, y no la libertad. Algunos términos comunes como «piratería» tienen una connotación que produce rechazo. Los medios se refieren con frecuencia a la copia prohibida como «piratería». De esa forma dan por hecho que la copia ilegal es equivalente éticamente a atacar barcos en alta mar, secuestrar y asesinar a sus pasajeros. Dado que nadie o casi nadie apoya seriamente la idea de que la copia ilegal sea algo parecido al secuestro o al asesinato, es preferible no utilizar la palabra «piratería» para describirla. En vez de eso, están disponibles términos más neutros como «copia prohibida» o «no autorizada». Stallman comenta en tono algo jocoso (Stallman, 2007) que incluso es preferible usar un término positivo como *“share some information with your neighbor”* («compartir información con el prójimo»).

2.3. Por qué software libre: una cuestión de libertad

Como trataremos de fundamentar en este apartado, existen poderosas razones para utilizar software libre. Entre estas razones se encuentran algunas que justifican también la utilización *exclusiva* de software libre.

En el mundo actual se observa con frecuencia un dilema¹¹ cuando una persona es el propietario o usuario registrado de un producto de software privativo, y un amigo, vecino o compañero de trabajo le pide una copia de dicho software.

El dilema es muy fácil de ver: o aquél le niega dicha copia (lo que es negar la ayuda a un vecino) o se la cede (con lo que viola la licencia que aceptó en su momento como único usuario legal del programa).

Hay dos maneras de resolver este dilema: una es no tener amigos (y se diría que esto es lo que ellos quieren). La otra es no aceptar nunca más, jamás, una licencia de software que no respete tu libertad. (Oliva, 2007)¹²

Para la otra persona, este dilema tiene solamente dos resultados cuando se trata de software protegido por licencias privativas y de pago. La persona que quiere el programa sólo puede: *a)* comprarlo, o bien *b)* conseguirlo de forma ilegal.

La cuestión relevante para el ámbito de la educación, es si se puede seriamente considerar la segunda alternativa.

- ¿Puede un profesor ceder copias no autorizadas a sus alumnos para la realización de sus trabajos?
- ¿Puede obligarles a que consigan dicha copia por canales *no oficiales*?
- ¿Puede, en caso contrario, obligarles a comprarlo?

El software libre ofrece una vía alternativa: dado que se puede copiar o descargar gratuitamente en muchos casos, de manera absolutamente legal y sin violar la licencia, el usuario no queda expuesto a los inconvenientes derivados del dilema expuesto. Por ello se dice que el software libre es una opción con valor *ético*.

¹¹ Este dilema ha sido expuesto en Stallman (2007).

¹² Esta cita proviene de los subtítulos en español de un *remix* que el brasileño Alexandre Oliva, fundador de la FSF de latinoamérica, realizó sobre la película *The Matrix*, de los hermanos Wachowski, postulando que todo su argumento gira en torno a la defensa de este modelo.

2.4. Aspectos éticos

En su origen, el concepto de software libre fue defendido por Richard Stallman como una necesidad ética. En su conferencia (Stallman, 2007), pronunciada en varias ocasiones con pocas diferencias, relata cómo las restricciones no justificadas de las compañías de software sobre sus productos, colocan al usuario en una encrucijada de la que es difícil salir. Hay un vínculo estrecho entre ética y libertad, y una discusión en profundidad se saldría del tema de este escrito. La referencia más citada son los trabajos del propio Stallman (1998).

2.5. El aspecto económico

Los titulares de la propiedad intelectual del software declaran como *pérdidas* el valor de los productos que están en manos de usuarios que no han abonado la correspondiente licencia. Es cierto que la idea de propiedad intelectual trata de proteger a los autores y está respaldada por acuerdos internacionales. Sin embargo, acompañando al valor moral de una obra se encuentra el valor económico, artificial, de las copias. Si un usuario está dispuesto a pagar o no el coste de una licencia, posiblemente es algo que debiera tenerse en cuenta para computar las referidas «pérdidas». Creemos que es evidente que si un usuario nunca hubiese pagado la licencia en primer lugar, no hay «pérdida» ni «lucro cesante» porque para ello debe existir primero una posibilidad real de venta, extremo cuya generalización muchos analistas rechazan y cuyo contraejemplo es el de este usuario que nunca hubiera adquirido el producto, de no haber tenido la otra opción.

Dentro de los esquemas de consumo de software privativo se produce un fenómeno que puede tener como consecuencia convertir al usuario en un esclavo de un proveedor o fabricante. Si el usuario quiere actualizar *legalmente* su versión del programa, no tiene otra alternativa que comprarla con la frecuencia que aquél decida. Habitualmente las actualizaciones se realizan bajo la esperanza de que tal o cual fallo de las versiones anteriores haya sido corregida, pero la estrategia del proveedor de «mantener contento al usuario» no siempre incluye la corrección de aquellos errores que más le interesan a éste ver corregidos. Ante esto surge una actitud que sospechamos es bastante frecuente, y se podría expresar como algo así: «si no voy a ser un usuario satisfecho, al menos no gastaré dinero».

Es decir, que las actualizaciones onerosas para el usuario, que no tengan una justificación clara para éste, posiblemente promuevan una cierta tendencia a la obtención *por otras vías* de esas actualizaciones, sin el sentimiento de culpa que en otros ámbitos podría esperarse.¹³

¹³No estamos promoviendo una actitud de rebeldía frente a los derechos legales que amparan a los titulares de la propiedad intelectual, por más que se haya abusado de la normativa sobre copyright. Más bien pensamos que éstos no tienen motivos para extrañarse de que esta rebeldía exista en razón, precisamente, de estos abusos.

La ironía del tema expuesto es que el Software libre abre vías a esta satisfacción que no pueden encontrarse en el software propietario, y son la posibilidad de adaptar el producto a nuestro gusto, o la de devolver algo al proyecto y a la comunidad, como veremos posteriormente. Son ejemplos de estas dos posibilidades que el software libre contempla, la traducción del programa y su documentación, y la redistribución de estas versiones traducidas. Estas dos posibilidades están al alcance de muchas personas con conocimientos de informática elementales y se consideran la puerta de entrada a colaboraciones más profundas o con un umbral de acceso más elevado.

Las restricciones artificiales a los productos de software privativo, como las que sufren las versiones “light” del mismo programa, tienen un efecto de *rebote* de consecuencias quizá no del todo previstas por el fabricante: cuando el usuario se da cuenta de que esta restricción no obedece, como si fuera un coche muy potente, a un mayor coste de fabricación, sino que es un bloqueo puesto ahí para eliminar ciertas funcionalidades, que de todas formas ya estaban programadas para la versión completa, se produce, suponemos, un irresistible deseo de poseer esta versión sin los mencionados bloqueos, y por ello de conseguirla por otros cauces. El ejemplo de la tabla 1 relaciona las posibilidades de varias versiones del programa Personal Composer junto a su precio.

Tabla 1: Versiones del programa de edición musical Personal Composer (Fuente: Personal Composer, Inc.; Datos de 2008).

Versión	Posibilidades	Precio (dólares)
PC-Pro	Hasta 128 pentagramas por sistema	210
PC-16	Hasta 16 pentagramas por sistema	150
PC-8	Hasta 8 pentagramas por sistema	75

Parece obvio que estas limitaciones son artificiales. Sería tema de un amplio estudio la demostración formal de todos estos supuestos, y la psicología del comprador de software frente al dilema económico de estar comprando algo cuyo precio no tiene ninguna correspondencia con su coste. El software libre, como una innovación disruptiva, tiende a romper la ley de la oferta y la demanda tal y como funcionaba para el software propietario.

En contra de estos criterios económicos que apoyan la utilización de software libre, y en particular respecto al entorno educativo, se podría argumentar que la obligación de los alumnos de comprar determinados libros de una cierta editorial es habitual, pongamos por caso, en los estudios de Medicina. Todos sabemos que los libros tienen a veces un coste elevado. Pero también es cierto que en la sociedad del conocimiento, éste crece cuando se comparte, sin disminución de su valor, muy al contrario de lo que ocurre al compartir un objeto. Cuando se hacen copias de software, éstas tienen un coste simbólico, aunque la información que contienen no disminuye un ápice respecto al original copiado. En educación, es un valor la puesta en común del conocimiento, y uno de los objetivos últimos de la Universidad.

Aunque existe software libre comercial, cuando tenemos la posibilidad de obtener un producto con funcionalidades *semejantes* a otro por el que habríamos de pagar, con la diferencia de que tiene un coste de licencias igual a cero, sólo una poderosa campaña de Marketing podrá (de hecho lo hace en todo el mundo) convencer a amplios sectores de la población de que tiene que pagar por el otro. Al menos era así hasta que en los últimos años el software libre se está adoptando en muchos ámbitos sin más publicidad que sus propios méritos; incluso (algo que antes era inconcebible) grandes compañías como Dell han incluido en su catálogo de productos, ordenadores con sistemas operativos libres preinstalados.

El coste nulo en ocasiones, muy barato casi siempre, del software libre, a pesar de ser un argumento poderoso, no es el mayor que puede defenderlo. Aún se debe tener en cuenta la seguridad mediante transparencia, veámoslo en el siguiente apartado.

2.6. Seguridad, fiabilidad

Cuando se habla de *seguridad*, el software privativo ha llegado a hacerse famoso por su «seguridad mediante opacidad» que consiste simplemente en esconder los fallos, pretendiendo quizá que no existen. Evidentemente, los virus y otros programas peligrosos no reconocen estas barreras como *hechas para ellos* y constituyen uno de los peores y más tristemente conocidos problemas de la informática actual, al menos a nivel doméstico. ¿Cuál es el origen de esta vulnerabilidad? En parte, por oposición, la apertura del código que ostenta el software libre, es fuente de seguridad. Expliquémoslo someramente.

Todos los sistemas están expuestos a inseguridades y al ataque de los programas maliciosos como los virus informáticos. Pero cuando el código de una aplicación como el sistema operativo de un ordenador está en situación de «abierto», lo que es una propiedad del software libre, esto implica que está a la vista de todo el mundo y que cualquier persona puede, con los conocimientos adecuados (cuya fuente está asimismo disponible), recrear o «compilar» la aplicación a partir de este código fuente, del mismo modo que se hizo con la aplicación previamente compilada que se distribuye para el público.

Y en este código abierto, no solamente los fallos de seguridad se pueden detectar antes de que se manifiesten, sino que cualquier usuario tiene, en principio, la posibilidad de corregir esta vulnerabilidad, y distribuir la versión corregida, antes incluso de que se haya producido algún problema por su causa. Una vez detectados, los problemas de seguridad de proyectos grandes, con muchos usuarios y/o desarrolladores, se corrigen en cuestión de días u horas. Ya sea a corto o a largo plazo, esto redundará en una seguridad *intrínseca* del software libre respecto a otros paradigmas.

Respecto a la fiabilidad, estabilidad o la así llamada *robustez*, en primer lugar debe recordarse

que no existe ningún software libre de fallos. La ventaja relativa del software libre en este campo no radica en la ausencia de problemas, sino en que los problemas, igual que hemos visto al hablar de los fallos de seguridad, se ven y se pueden identificar y corregir en un ciclo mucho más corto merced al estado de apertura del código.

La Ley de Linus, bautizada así según Linus Torvalds por Eric S. Raymond (Raymond, 1999), establece que «dado un número suficientemente elevado de ojos, todos los errores se convierten en obvios». Todo problema deberá ser transparente para alguien. Las personas que entienden un problema y las que lo resuelven no son necesariamente las mismas, ni siquiera en la mayoría de los casos. Linus Torvalds dice que «alguien encuentra el problema y otro lo resuelve». Pero lo más interesante es que ambas cosas suelen suceder muy rápidamente. Raymond (1999) propone una prueba para la ley de Linus: si la ley fuera falsa, entonces cualquier sistema suficientemente complejo como el núcleo Linux, que está siendo manipulado por tantos, debería haber colapsado bajo el peso de interacciones imprevistas y errores “muy profundos” inadvertidos. Pero si es cierta, bastaría para explicar la relativa ausencia de errores en el código de Linux, y para demostrar la superioridad, en cuanto a la calidad del producto final, del modelo de desarrollo de software libre sobre el modelo tradicional usado para desarrollar software propietario.¹⁴

2.7. Disponibilidad dentro de la legalidad

Tratándose de software libre, un factor de confianza apoya al usuario: éste dispone del software en todo momento, por los canales oficiales o por una redistribución. Aunque el equipo en pleno abandonase el proyecto, el código fuente seguiría estando disponible y siempre podría retomarse posteriormente. Por su parte, el usuario tiene permiso para realizar cuantas copias estime oportuno. El usuario puede (y de hecho se le anima a que lo haga) redistribuir el programa y promover su descarga, de forma legal y con importantes consecuencias desde el punto de vista práctico.

2.8. La paradoja de la masa crítica. La innovación disruptiva

Se puede observar, aunque ignoramos si ha sido demostrado formalmente, que el conocimiento global que una comunidad tiene sobre una herramienta (pongamos por caso el manejo de un procesador de textos como Word o de un editor de partituras de música como

¹⁴«A lo largo de los diez años, más o menos, que estuve pagando el “impuesto Final”, remití docenas, quizá más de cien, informes de fallos y solicitudes de funcionalidades nuevas, y por la época en que abandoné su uso (si no recuerdo mal, en 1999 lo actualicé por última vez) ni una sola corrección o funcionalidad fue *jamás* implementada. Ni una sola.» (Kieren McMillan, lists.gnu.org/archive/html/lilypond-user, mayo 2015)

Finale por parte de los estudiantes) se mantiene o se sostiene a sí mismo en un equilibrio parecido a la extensión de una llama sobre un material inflamable. Hay una reacción en cadena en virtud de la cual llega un momento en que es fácil, casi seguro, que siempre haya alguien cerca que pueda resolver el problema que se plantee. Se ha alcanzado la «masa crítica»: una implantación en la sociedad que no precisa de más promoción que el simple hecho de que estas herramientas *se usen*.

Por otro lado, los productos de los que estamos hablando tienen un valor comercial y no son precisamente del dominio público a pesar de su amplia difusión.

La paradoja es que a las empresas manufactureras, sin ninguna duda, *les interesa* que exista esta masa crítica, pues sus productos se convierten en estándares de facto que tienden a anular a la competencia, erigiéndose en la opción indiscutible.

En más de una ocasión se ha sugerido la posibilidad de una *perversión* por parte de las empresas de software que «toleran» hasta cierto punto la distribución no autorizada de sus productos, pues de esta manera mantienen a largo plazo la mencionada masa crítica y su *estatus* como estándares de facto. Los estándares sin masa crítica no son estándares. Pensemos por un momento en qué ocurriría si, por ejemplo, *solamente* pudieran utilizar un editor de partituras popular como Finale, los usuarios que hubieran adquirido el producto. En nuestra opinión, basada en la lógica empresarial de la oferta y la demanda, el producto tendría que haber recortado su precio drásticamente para facilitar un aumento de su parque de usuarios y recuperar su masa crítica, si es que alguna vez la hubiera tenido.

En 1976, durante el período inicial de la informática casera con el lenguaje BASIC, Bill Gates se quejaba públicamente de que los *hackers* instalaban sus programas a partir de copias no autorizadas, mientras que existen pruebas (Torres, 2013) de que él mismo facilitaba copias a modo de *marketing*, una práctica de la que se beneficiaba porque los destinatarios remitían informes de fallos y sugerencias de mejora que le resultaban sumamente útiles, y el número de usuarios se acercaba a la masa crítica.

El software libre es una innovación *disruptiva* (Flores, 2008; Hardings, 2004). Su estrategia implica innovar de una forma nunca vista y progresar *por debajo de un negocio existente* hasta que termina sustituyéndolo. Tenemos ejemplos en las cámaras fotográficas digitales, los teléfonos móviles, la propia Internet e incluso la imprenta.

Es difícil saber *a priori* si una nueva tecnología se transformará en una amenaza o sólo será un producto más. Clayton Christensen, en su libro “El dilema del innovador” (Christensen, 1997a), explica que para que una innovación sea disruptiva, “debe trepar sigilosamente desde abajo, cerca de la base de un negocio existente y amenazar, poco a poco, con desplazarlo”. En un comienzo lo disruptivo ofrece un desempeño inferior, con menores funcionalidades, situándose a un bajo precio en el mercado. Sin embargo, esta oferta podría constituir una

atractiva alternativa para un determinado tipo de cliente, que no busca necesariamente un producto con todas las prestaciones: basta con que cumpla con las expectativas. Con la disrupción en escena, la nueva tecnología empieza gradualmente a desplazar al antiguo producto, mejorando sus características, en un proceso que finalmente genera un cambio en las costumbres de los consumidores que terminan inclinándose en masa hacia la nueva propuesta (Flores, 2008).

Según Juantomás García, expresidente de la Asociación Española de Usuarios de Software Libre (HispaLinux), hacer y compartir software es una de las cosas más progresistas y revolucionarias que se pueden hacer (García, 2008).

2.9. Desarrollo cooperativo

En el contexto del análisis de la propiedad intelectual, Sádaba (2008) hace notar la superioridad del modelo del software libre, que explota el lado cooperativo e interactivo del trabajo intelectual frente al patrón rígido del software propietario. Por su naturaleza abierta, el software libre es un caldo de cultivo para la cooperación en todos los sentidos. Se estudia en bastante profundidad en la tesina de J.L. Salcedo (Salcedo, Anduiza y col., 2010) el aspecto del desarrollo colectivo.

Es cierto que el software se ve generalmente como un objeto cerrado que uno simplemente instala y utiliza. Pero sin duda la barrera de entrada al desarrollo del software tiene muchos lugares fáciles de traspasar incluso por quien tiene menos experiencia en programación. Son ejemplos de ello la redacción de documentación, la traducción, las pruebas de usabilidad o el diseño de interfaces (Arjona, 2012; Arjona y Robles, 2012; Robles y col., 2014).

El desarrollo cooperativo de software libre es cualitativamente diferente de otros tipos de cooperación en el sentido de que está deslocalizado geográficamente, y su auge se explica por la existencia de fuentes de motivación desconocidas con anterioridad, nuevos incentivos que rehúsan ceñirse a categorías económicas (Moody, 2002; Wayner, 2000).

En referencia a las traducciones, la misma autora mencionada anteriormente (Arjona, 2012) compara los dos escenarios en que se mueve la localización del software, y que aparecen resumidos en la tabla 2. En el contexto de este ejemplo, el concepto de desarrollador se refiere solamente al autor exclusivo del software privativo.

2. Software libre y control de versiones

Tabla 2: Comparación de los escenarios de traducción del software privativo y el software libre, según Arjona (2012).

Software privativo

- Licencia: el desarrollador tiene el control
- El desarrollador provee las traducciones
- Los usuarios dependen del desarrollador para la existencia de nuevas traducciones
- El desarrollador encarga los servicios de traducción

Software libre

- Licencia: los usuarios tienen el control
 - Tanto el desarrollador como los usuarios proveen las traducciones
 - Tanto el desarrollador como los usuarios pueden encargar los servicios de traducción
 - Todos se benefician de las nuevas traducciones
-

El trabajo en proyectos de software libre no está necesariamente descentralizado, aunque es lo más frecuente. Con su oposición a la existencia de monopolios y a la aplicación de patentes sobre cualquier producto de software, los equipos abiertos de software libre se han convertido en el paradigma de la producción intelectual colectiva (Benkler, 2004; Sádaba, 2008). Sin embargo, no debe olvidarse que las licencias libres, a pesar del apelativo de *rebeldes* con que la prensa suele etiquetarlas, están basadas en la legalidad vigente y dependen de los esquemas de copyright para proteger y hacer valer su existencia.

La tabla 3 resume las diferencias entre el software libre y el propietario, según el *Libro Blanco del Software Libre en España* de Abella y col. (2003, p. 31).

Tabla 3: Diferencias entre el software libre y el propietario.

Aspecto a considerar	Software propietario	Software libre (GPL)
Acceso al código fuente	Prohibido por licencia	Sí, garantizado
Corrección de errores en el programa por el cliente/usuario	No	Sí
Duplicación del software	Prohibida	Posible y recomendada
Libertad de competencia para el mantenimiento	No, depende del fabricante	Sí, imposible limitarla
Posibilidad de examinar el código del producto	Prohibido, salvo permiso del fabricante.	Sí
Venta de segunda mano	Prohibida	No aplicable
Respeto a estándares globales	En función del fabricante	En la mayoría de los casos
Adaptaciones al cliente (p.ej.: idioma)	En función del fabricante	Disponible
Virus, gusanos	Frecuentes	Muy infrecuentes

2.10. Control de versiones. Git

Una revisión del sistema Git en este apartado de fundamentación teórica es de una gran importancia para el primer estudio, porque es la herramienta que permite extraer los datos para su análisis, a partir de la fuente primaria.

La idea del control de versiones surge con los inicios de la ciencia de la computación, del software y de los problemas que plantea. El concepto de **versión** es una idea que pertenece a la informática fundamental y difiere de la acepción usual del término, es decir, ese número que a modo de apellido acompaña al nombre de un programa de ordenador como producto.¹⁵ Aquí la versión obedece más bien al contenido de un texto, y se aplica a la gestión del código fuente.

Al tratar con el control de versiones, el código se encuentra en principio en un determinado estado o versión. Tan pronto un programador modifica el código, éste cambia a un nuevo estado. Cuando se registra esta modificación se produce una **intervención** y una nueva versión. Así, la sucesión de intervenciones conduce al código a través de los correspondientes estados o versiones.

Con un sistema de control de versiones se persiguen los siguientes objetivos:

- Registrar el estado actual del código, poderlo etiquetar y examinar, de forma que podamos recuperar la configuración registrada en cualquier momento.
- Permitir la conservación de las modificaciones de forma secuencial en una *historia* que se pueda recorrer en los dos sentidos, desde sus inicios hasta la configuración actual.
- Facilitar las aportaciones concurrentes de distintos desarrolladores, y gestionar los conflictos que se puedan producir entre ellas.
- Posibilitar la ramificación o bifurcación del código, para pruebas o para sub-proyectos separados que, en su caso, se puedan reintegrar al proyecto principal.
- Servir como copia de seguridad y como fuente de documentación del código.

Existe una amplia variedad de sistemas de control de versiones. El sistema más relevante históricamente ha sido CVS,¹⁶ diseñado e implementado en 1986 por Dick Grune y vuelto a implementar cuatro años más tarde, partiendo de cero, por Brian Berliner y otros (Berliner y col., 1990), aunque los orígenes se remontan a la tesis de 1970 del profesor Leon Pressor de

¹⁵Los números de versión según esta acepción popular se han extendido a otras ideas que no son productos de software; por ejemplo: la *Web 2.0*.

¹⁶CVS: Concurrent Version System (Sistema de versiones concurrentes).

la Universidad de California en Santa Bárbara, sobre control de cambios y configuraciones. Para sistemas Unix, el primer sistema de control de versiones fue SCCS,¹⁷ desarrollado por M. J. Rochkind en esas mismas fechas (Rochkind, 1975). El almacenamiento central que proporcionaba SCCS recibía el nombre de **repositorio**, un concepto fundamental que sigue siendo utilizado hasta hoy día.

A principios de los años 80, Walter Tichy presentó el RCS,¹⁸ (Tichy, 1982, 1985), que introdujo los conceptos de *Delta* directa e inversa para el almacenamiento eficiente de distintas revisiones de un archivo. El RCS fue superado en varios aspectos por CVS, que se volvió tremendamente popular y fue el estándar de hecho durante muchos años para el desarrollo de software de fuentes abiertas. Alrededor de 2001 se comenzó a usar el sistema Subversion (SVN) que era mucho más rápido que CVS y tenía el concepto de intervenciones atómicas (no divisibles) y una mejor infraestructura para el manejo de **ramas** o líneas paralelas en la historia de las modificaciones. Finalmente, entre otros sistemas de importancia deben citarse BitKeeper y Mercurial porque eliminaron la idea de un repositorio central maestro y abrieron la puerta a la gestión descentralizada del código.

BitKeeper se estuvo utilizando para registrar las versiones del código del núcleo Linux hasta que fue imposible utilizarlo de manera gratuita por decisión de la compañía, lo que detonó la decisión de Linus Torvalds de desarrollar **Git** en el año 2005 y publicarlo como Software Libre.

Con un sistema de control de versiones como Git, las diferentes versiones de un archivo se guardan en la **historia**, no en archivos separados. La historia registra las versiones anteriores de un archivo, con el mismo nombre u otro distinto, y en la misma carpeta o en otra diferente.

El sistema Git es significativo para el objeto del presente estudio porque es el sistema con que está gestionado el proyecto LilyPond. Así pues, tanto *a*) los trabajos de cooperación directa en él, ya sea como desarrollador o como artífice de la documentación o su traducción, como *b*) los trabajos de análisis de su historia y evolución, han de realizarse utilizando esta herramienta.

Git es un sistema distribuido. Su énfasis en la descentralización no impide que lo pueda utilizar un particular de forma local, para la gestión de sus propios proyectos.¹⁹ Como se refirió anteriormente, el sistema Git fue diseñado y desarrollado inicialmente por el programador

¹⁷SCCS: Source Code Control System (Sistema de control de código fuente).

¹⁸RCS: Revision Control System (Sistema de control de revisiones).

¹⁹El código fuente \LaTeX de la propia tesis está gestionado mediante Git, y reside en un repositorio distribuido entre tres máquinas: un ordenador de sobremesa y dos portátiles, cada uno de los cuales contiene una copia de la historia del documento, historia que se sincroniza periódicamente en las dos direcciones. Ello posibilita la independencia del lugar de trabajo y que se puedan resolver con éxito los posibles conflictos en caso de que el documento se edite al mismo tiempo en los dos lugares.

finlandés Linus Torvalds para la gestión del proyecto de núcleo Linux,²⁰ que él mismo creó en el año 1991, y entre las directrices de su diseño se encontraba curiosamente la de «tomar el mencionado sistema CVS como un modelo de lo que *no* hacer, y en caso de duda, tomar la decisión diametralmente opuesta»,²¹ lo que da una idea de lo distintos que pueden ser los enfoques en estas herramientas de trabajo.

Cada una de las copias de una carpeta de trabajo gestionada por Git, es un repositorio completamente funcional con toda la historia y con plenas posibilidades de seguimiento de las revisiones, de una forma no dependiente de un acceso activo a la red ni a un servidor central. Por esto decimos que Git es un sistema *descentralizado*.

Git es libre y de código abierto. Como un proyecto cualquiera de software, susceptible de un control de versiones, el propio Git está a su vez gestionado por Git.²² De esta forma, los desarrolladores han cerrado un círculo sobre el propio programa, haciéndolo responsable de su propio control. No es una situación tan paradójica como puede parecer, ya que estamos hablando de una herramienta muy flexible para todo tipo de utilizaciones en el campo del desarrollo de software, que los autores realizaron para facilitar su propio trabajo.

Ya hemos mencionado que Git se usa para su propio desarrollo así como para el del núcleo Linux, y también para el mantenimiento del proyecto LilyPond, su documentación (junto a las traducciones de ésta) y su página web, en distintas ramas de un único repositorio. También hemos dicho que Git es el sistema de control de versiones que estamos utilizando para la elaboración de este trabajo. Así pues, Git se puede usar para el control de versiones de cualquier tipo de código, incluyendo partituras musicales representadas en algún código formal como el lenguaje de tipografía musical LilyPond, siendo algunos de los primeros casos dignos de mención la ópera *Affaire Étrangère* de Valentin Villenave²³ y las óperas barrocas de Haendel y compositores franceses tipografiadas por Nicolas Sceaux. Por estas razones creemos que es necesario mencionarlo aquí y centrar nuestra atención sobre sus conceptos.

2.10.1. Nociones de Git

Es necesario dar, cuando menos, una ligera pincelada terminológica sobre las nociones de Git que son de relevancia para nuestro estudio. Sería superfluo impartir aquí un curso de Git a causa de la ya abundante información que está a disposición de la comunidad, y ello a pesar de que este programa no se caracteriza por disponer de una documentación oficial que sea

²⁰El sistema operativo libre comúnmente conocido como “Linux” es en realidad un complejo formado por un sistema operativo denominado GNU, provisto de una pieza central o núcleo que es el Linux, lo que le otorga al complejo su verdadero nombre: GNU/Linux.

²¹Esta cita de Linus Torvalds se encuentra en el código fuente del programa Git (Torvalds y Hamano, 2005).

²²El auto-alojamiento de Git, o su utilización para gestionarse a sí mismo como proyecto, conocido como *bootstrap*, se produjo el 7 de abril de 2005, cuatro días después del inicio de su desarrollo.

²³Estrenada en enero de 2009. Véase el apartado 5.8 (pág. 106).

muy completa o abundante en ejemplos. Git es un programa en modo texto y, según su autor, no pensado para otra cosa que ser rápido y eficiente en su funcionamiento.

Estas nociones pueden encontrarse muy bien explicadas en toda clase de recursos en línea aportados por usuarios del programa. El objetivo principal de un sistema de control de versiones como el que nos ocupa es posibilitar y facilitar el desarrollo colaborativo por parte de distintas personas, sobre el mismo código.

Git mantiene una base de datos llamada **árbol** que registra el contenido de una carpeta bajo su control. Cualquier cambio en el contenido de esta carpeta se detecta y produce lo que llamamos una **diferencia**.²⁴ Así pues, un desarrollador tiene en su sistema una copia del árbol de código y, para hacer aportaciones al mismo, modifica uno o varios archivos con un editor de texto. En este momento, su copia del árbol está modificada y presenta un estado distinto al anterior, extremo que el programa Git es capaz de presentar como una serie de diferencias del estado actual del árbol, respecto del estado anterior.

Cuando se incorporan estas diferencias a la historia de las versiones, se produce una contribución o **intervención**.²⁵ Al producir una intervención, se registran en la historia del árbol, además del conjunto de todas las diferencias, el nombre y dirección del desarrollador que **comete** la intervención, título de la misma, la fecha y la hora, así como un comentario opcional, todo ello firmado electrónicamente. Una intervención está enlazada con la versión anterior, conocida como **ancestro**.

Puede transformarse un conjunto de diferencias o una intervención ya registrada, en un archivo. Si se vuelcan las diferencias en un archivo, se obtiene lo que se conoce como **parche**.²⁶ El nombre obedece a la metáfora del “parcheado” de un sistema para arreglar algún fallo. Cuando un parche se aplica en un sistema distinto a aquél en que el parche se creó, se producen en este árbol las mismas diferencias que dieron lugar al parche en el momento de su creación. Si el parche se ha obtenido a partir de las diferencias entre dos intervenciones, al aplicarlo a un árbol se producen en éste copias exactas de esas mismas intervenciones.

Los parches se pueden enviar por correo electrónico o llevarse en un medio de almacenamiento externo y se pueden intercambiar entre distintos desarrolladores. Un colaborador externo al proyecto puede enviar parches sobre un código, que los desarrolladores principales son libres de aplicar sobre su árbol. Cuando los distintos desarrolladores sincronizan sus respectivos árboles, el parche se propaga paulatinamente entre todos ellos.

²⁴El término inglés usual para referirse a una diferencia es la abreviatura *diff*.

²⁵“Intervención” es el término que hemos decidido adoptar para la voz inglesa *commit*, que no parece gozar de un consenso en cuanto a su traducción del inglés.

²⁶El término inglés para “parche” es *patch*.

2.10.2. Ramas y mezclas

En cualquier momento, un autor puede producir una rama nueva en la historia del proyecto; el fenómeno de la ramificación en la historia de versiones y la reunificación de estas ramas, produce un tipo especial de intervención conocido como “merge”, **mezcla** o combinación. Se produce una ramificación seguida de una reunificación cuando un desarrollador modifica su propio árbol de código y posteriormente este mismo desarrollador efectúa la mezcla de su rama con la historia principal del proyecto. La figura 4 muestra un ejemplo de ramificación y combinación del historial de modificaciones.

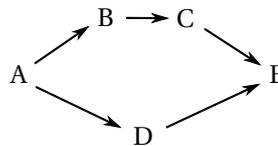


Figura 4: Esquema de una ramificación y mezcla en el historial de versiones.

En el esquema, el estado A evoluciona hacia los estados B y C. Mientras tanto, un desarrollador actúa sobre su propio ejemplar, produciendo el estado D. En el estado E las historias se combinan, ambas se convierten en una sola y todas las intervenciones de la rama que se mezcla se incorporan a la rama mezclada. Al producirse el hecho de la mezcla, hay una intervención que registra todos los cambios efectuados tanto en la rama remota del desarrollador como aquellas que se han producido, durante el mismo período, en la rama principal. Así pues, la mezcla en sí repite datos de intervenciones de las dos ramas. Por ello, al hacer el análisis de los datos históricos, deben eliminarse del análisis todas aquellas intervenciones caracterizadas como mezclas, que son las que cumplen la propiedad de tener más de un ancestro. Según el mismo ejemplo, sabemos que la intervención E es una mezcla porque tiene dos ancestros que son C y D: no hay datos nuevos que contabilizar. Las mezclas desvirtúan el análisis cuantitativo porque tienden a ser muy grandes si las dos ramas han evolucionado separadas durante un tiempo prolongado.

La utilidad de las ramas es bifurcar la historia para efectuar modificaciones temporales sin que afecten al árbol principal.

Cuando se produce una mezcla existe la posibilidad de un **conflicto**. En el caso más habitual de ausencia de conflicto, si dos conjuntos de diferencias afectan a archivos distintos o si, afectando a los mismos archivos, se producen en distintas partes de ellos, las dos diferencias pueden aplicarse de forma concurrente sin que se produzca ningún conflicto. Sin embargo, tendremos un conflicto siempre y cuando dos ramas o dos conjuntos de diferencias afecten a las mismas líneas de los mismos archivos. En este caso, la persona que realiza la fusión tiene que decidir qué líneas sobreviven al proceso, dando como resultado una intervención limpia que mantiene, no obstante, el registro de los conflictos producidos.

Dos completas guías sobre el uso y posibilidades de Git son Loeliger (2009) y Swicegood (2008). Para tratar de enmarcar con claridad los conceptos que se han tocado en estos apartados de fundamentación, la figura 5 presenta la relación entre los conceptos de LilyPond, Git y otros asociados dentro del mismo contexto.²⁷

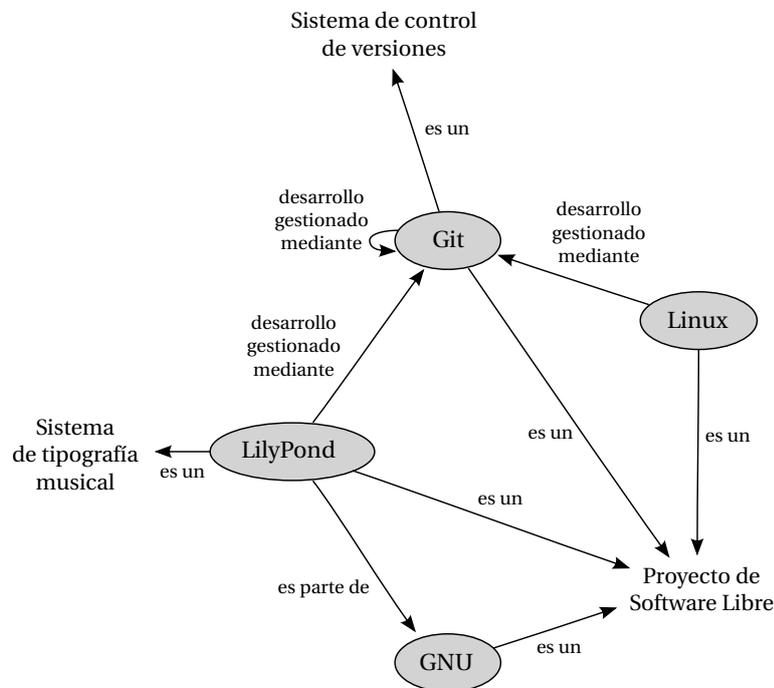


Figura 5: Mapa conceptual que relaciona a Git, LilyPond, GNU y Linux.

2.11. El Software libre en Extremadura

2.11.1. El proyecto gnuLinEx

El gobierno extremeño decidió a finales de 2000 que implantaría un ordenador por cada dos alumnos en cada centro educativo de enseñanza secundaria que se construyera a partir de aquel momento (Gómez Deck, 2005). El alto costo de las licencias de software necesarias para dotar adecuadamente los nuevos centros, aumentado por la necesidad continua de renovación de versiones de las aplicaciones de software propietario, la imposibilidad de acceder a sus códigos para hacer modificaciones y/o adaptaciones, etc. hizo ver a los responsables públicos de aquel momento que la única forma de llevar a cabo este proyecto era mediante el uso de software libre. En abril de 2002 se presentó la primera versión de gnuLinEx que contenía además del sistema operativo una amplia colección de aplicaciones útiles de ofimática, uso de Internet, tratamiento gráfico, etc. y que fue instalada en los nuevos centros.

²⁷El gráfico está elaborado con GraphViz (AT&T, 2006).

El proyecto LinEx consistió en desarrollar una distribución personalizada con dos objetivos claros: penetración de ordenadores por alumno en los centros educativos y el desarrollo tecnológico del tejido socioeconómico de la región extremeña.

Dado que los primeros meses del curso 2002/2003 en los centros nuevos no fueron traumáticos ni para profesores ni para alumnos, tal y como muchos auguraban, en las navidades de 2003 se tomó la decisión política de implantar la ratio de dos alumnos por ordenador en todos los institutos públicos de enseñanza secundaria de la región. Es decir, desde los nuevos hasta aquellos con tradición de muchas décadas de enseñanza.

Con ciertas matizaciones, más allá de su carácter de proyecto piloto, constituyó una apuesta clara y decidida por integrar la informática como una herramienta más en el aula para ayuda de profesores y alumnos, usando en lo posible todo su potencial educativo y familiarizando a los alumnos con el trabajo con el ordenador que, con toda probabilidad, se van a encontrar en todas partes en su vida futura (Gómez Deck, 2005, pp. 114–115).

García (2003) aprecia que Extremadura es, sin duda alguna, la comunidad autónoma española que más ha apostado por el uso del Software Libre en la Educación y una de las más avanzadas en el mundo en cuanto a su desarrollo. Gracias al proyecto LinEx, Extremadura ha aparecido en portadas del Washington Post (Eunjung-Cha, 2002) y es frecuentemente nombrada como paradigma tecnológico del Software Libre.

2.11.2. El CIRL

El Centro Internacional de Referencia Linux (CIRL)²⁸ (hoy desaparecido) nació en 2006 de un convenio entre la Junta de Extremadura, la empresa IBM y la Universidad de Extremadura con el objetivo principal de promocionar Linux en nuestra región, proporcionando sus servicios sin ánimo de lucro. El CIRL se encontraba ubicado en la Escuela Politécnica de Cáceres.

Entre sus objetivos estaba el de ayudar a potenciar el uso del sistema operativo Linux en empresas e instituciones de nuestra región. Además trataba de servir de apoyo al esfuerzo realizado por la Junta de Extremadura durante varios años con gnuLinEx, con una gran repercusión a nivel internacional. Los objetivos principales del CIRL eran, por un lado, impulsar el uso de Linux como una alternativa mejor frente a sistemas operativos propietarios y, por otro lado, servir de soporte a particulares, PYMES e instituciones extremeñas que deseen migrar sus sistemas a Linux. El Centro también subvencionaba trabajos de investigación de la UEX en entorno Linux, sirviendo de banco de pruebas para el uso de esta tecnología.

Para lograr estos objetivos, el Centro preparó manuales de ayuda a varios niveles, impartió

²⁸El sitio oficial del CIRL era cirl.unex.es. Visitado el 19 de enero de 2013. No disponible actualmente.

cursos de ayuda tanto a usuarios particulares como a administradores de sistemas de empresas e instituciones, realizó asistencia personalizada a las PYMES que desearan migrar sus sistemas a Linux, organizaba jornadas periódicas de debate y difusión de Linux, subvencionaba seminarios y proyectos relacionados con Linux y el Software Libre y participaba en cualquier actividad que ayudase a la difusión de este sistema operativo. Estas actuaciones se realizarían de forma coordinada con las acciones que se hubieran llevado a cabo desde la Consejería de Infraestructuras y Desarrollo Tecnológico de la Junta de Extremadura.

2.11.3. CENATIC

CENATIC²⁹ es el Centro Nacional de Referencia de Aplicación de las Tecnologías de la Información y la Comunicación (TIC) basadas en fuentes abiertas, y tiene su sede en la ciudad de Almedralejo.

CENATIC es una Fundación Pública Estatal, promovida por el Ministerio de Industria, Turismo y Comercio (a través de la Secretaría de Telecomunicaciones y para la Sociedad de la Información y la entidad pública Red.es) y la Junta de Extremadura, que además cuenta en su Patronato con las comunidades autónomas de Andalucía, Aragón, Asturias, Cantabria, Cataluña e Illes Balears. También forma parte del Patronato de CENATIC las empresas Atos Origin, Sun Microsystems, Bull, Telefónica y Gpex.

CENATIC es el único proyecto estratégico del Gobierno de España para impulsar el conocimiento y uso del software de fuentes abiertas, en todos los ámbitos de la sociedad.

La vocación de la Fundación es posicionarse como centro de excelencia nacional, con proyección internacional tanto en el ámbito europeo como iberoamericano.

2.11.4. El proyecto UEX-Linux

Después del paso pionero que dio la Junta de Extremadura en 2002 al desarrollar un sistema operativo libre con un fuerte sabor autonómico, la distribución de Linux gnuLinEx de la que hemos hablado en el apartado 2.11.1, muchas otras iniciativas siguieron sus pasos y a día de hoy se pueden contar doce distribuciones hechas en España, con una importancia variable; las más significativas son Guadalinux (Andalucía), MAX (Madrid), LliureX (Valencia), Molunix (Castilla-La Mancha), Melinux (Melilla) y mEDUXa (Canarias), según Márquez (2007). Aparte de las distribuciones autonómicas, de entre las comunidades con distribuciones no oficiales, Cantabria cuenta con dos distribuciones diferentes y Galicia con tres.

²⁹El sitio oficial del CENATIC es cenatic.es. Visitado el 21 de agosto de 2015.

Todas las distribuciones se pueden usar en todos los idiomas que soporten los programas incluidos, a pesar de que la estética y la selección de aplicaciones pueden estar configuradas, de forma predeterminada, a lo que sus desarrolladores han diseñado. La flexibilidad propia del software libre facilita que, en la práctica, todas las distribuciones puedan utilizarse para cualquier propósito.

La mayoría de los objetivos que las distribuciones autonómicas satisfacen, son compartidos por las universidades. Canarias es la Comunidad Autónoma con más distribuciones de GNU/Linux universitarias activas, concretamente existe una por cada provincia: Bardinix de la ULL en Santa Cruz de Tenerife y SILU de la ULPG en Las Palmas. Además de las dos mencionadas, en 2008 había 6 distribuciones universitarias activas en España: EHUX (Univ. País Vasco), GLUZ (Univ. Zaragoza, Aragón), LUC3M (Univ. Carlos III, Madrid), PTD (Univ. Oberta de Catalunya), Trisquel (Univ. Vigo) y UEX Linux (Univ. Extremadura). De la Orden (2013) ha publicado un mapa de las distribuciones universitarias para situarlas físicamente, además de un listado en el que incluye también las que ya no están activas (en total sumaríamos 18 distribuciones). En el mismo recurso puede verse un listado completo en el que relaciona hasta 46 distribuciones españolas, en enero de 2009. Caba (2009) ofrece un contenido similar, en el que encuentra 50 distribuciones entre universitarias, autonómicas y municipales, la mayoría de ellas basadas en Ubuntu y Debian.

De las distribuciones universitarias, se hallaban activas y en desarrollo un total de siete, según de la Orden (2013). Precisamente una de éstas es la de la Universidad de Extremadura, de nombre UEX-Linux.

El Centro Internacional de Referencia Linux (CIRL, mencionado en el apartado 2.11.2) coordina el desarrollo de esta distribución. Para la primera versión *alpha* de 2008 se tomó como base una distribución existente de mucha difusión, Ubuntu Linux 8.04 LTS, tratándose de un prototipo que tenía como objetivo recopilar aplicaciones software usadas en la parte práctica de asignaturas impartidas en la propia Universidad de Extremadura. Inicialmente se incluyeron algunas herramientas y aplicaciones principalmente de titulaciones de Informática. El DVD de UEX Linux se entregaría a los alumnos que cursaran estudios en las titulaciones de Informática en la Escuela Politécnica de Cáceres y en el Centro Universitario de Mérida.

Posteriormente, entre los objetivos de UEX Linux para el curso 2009/2010 está el de recopilar todas las aplicaciones docentes y de prácticas usadas por los profesores de la UEX. Se basa en la distribución Ubuntu Linux 8.10. Es muy interesante que al no encontrarse todo el software requerido en los repositorios de Ubuntu, algunas de las aplicaciones y bibliotecas que forman parte de UEX Linux se obtuvieron a partir de sus fuentes originales. Según el CIRL, este es uno de los motivos por el que no se descarta la posibilidad de añadir un nuevo repositorio de software propio en el futuro. El portal del CIRL ofrece la resolución de dudas y preguntas sobre UEX Linux y sobre cualquier otra consulta relacionada con Linux. Puesto que UEX Linux está

basada en Ubuntu, el usuario también puede acudir al soporte de la Comunidad Ubuntu en caso necesario.

Desde la liberación de UEx Linux 2009 en diciembre de 2008 se habían registrado hasta junio de 2009 cerca de 5000 descargas, sin contar las casi 3500 descargas de la versión de prueba lanzada en verano del año anterior. No disponemos de datos posteriores.

2.12. Licencias libres en Extremadura

Por su interés que relaciona a la comunidad de Extremadura con el tema del presente trabajo, reseñamos esta actuación de la Asamblea Regional orientada a la legislación nacional, en la que se tratan directamente las licencias Creative Commons, la licencia GPL y la Ley 23/2006, de 7 de julio, por la que se modifica el texto refundido de la Ley de Propiedad Intelectual, aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril. [LPI].

El Boletín Oficial de las Cortes Generales (VIII Legislatura) publicó en el número 281-1 de 25 de mayo de 2007, una proposición de ley que fue presentada por la Asamblea de Extremadura con el siguiente título:

125/000002 Proposición de Ley de modificación del Texto Refundido de la Ley de Propiedad Intelectual, de forma que se reconozca al Autor el derecho de explotación de su obra bajo las denominadas Licencias Libres. [Prop-TR-LPI]

La proposición vuelve a presentarse posteriormente, apareciendo en el Boletín de la X legislatura en el número 18-1 de 28 de diciembre de 2011 (Prop-TR-LPI, 2011).

En la Exposición de motivos podemos leer:

La Ley 23/2006, de 7 de julio, de modificación del texto refundido de la Ley de Propiedad Intelectual, dio respuesta a la necesidad existente de incorporar al Derecho español la Directiva 2001/29/CE del Parlamento Europeo y del Consejo de 22 de mayo de 2001, procediéndose a una armonización de los derechos patrimoniales de reproducción, distribución y comunicación pública.

La Ley parte del reconocimiento expreso del autor de la obra a su plena disposición y explotarla y disponer de ella a su voluntad. Ello implica un derecho de opción por el autor, en el sentido de decidir qué derechos quiere mantener sobre su obra y cuáles podrían ser objeto de transferencia o cesión o ser objeto de licencias contractuales.

Con la aplicación al sector de los contenidos y el conocimiento del mismo modelo de licencias que se ha venido utilizando en el software, como es el caso de la Licencia Pública General (GPL) o las Creative Commons, se abre un camino para el desarrollo de un modelo de creación de contenidos sostenible, que prime y fomente el desarrollo de nuevas creaciones y la aparición de creadores de nuevos tipos de contenidos.

Los poderes públicos, como responsables máximos de la mejora de calidad de vida de los ciudadanos, deben participar en la evolución de las Tecnologías de la Información y las Comunicaciones (TIC), así como velar, en todo momento, para aplicarlas en beneficio colectivo, y en este marco las Comunidades Autónomas deben tener un papel relevante. En aras de este objetivo, el Parlamento extremeño está convencido de la necesidad de arbitrar mecanismos que posibiliten la utilización de licencias que sean verdaderos instrumentos que flexibilicen el desarrollo de la propiedad intelectual. Una flexibilización que, partiendo del escrupuloso respeto a los derechos de propiedad intelectual de los derechohabientes, debe otorgar un mayor control al autor en la difusión de sus obras a través de las nuevas tecnologías y hacer posibles las múltiples posibilidades de acceso y difusión de obras que ofrecen las redes de comunicaciones electrónicas.

Para la consecución de estos objetivos se considera necesario dar un paso más en ese campo y hacer una apuesta contundente por este tipo de licencias que fomentan compartir conocimiento y dar satisfacción a las demandas de la sociedad civil en este sentido.

Ello pasa por una reforma del Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, en el sentido de que el texto legal haga un reconocimiento expreso de los instrumentos concretos, licencias, respecto de los usos específicos de una obra en el entorno digital, generando seguridad jurídica no sólo en favor del autor, al permitirle conocer qué derechos está cediendo y graduar los mismos, sino a los usuarios de las obras, dado que la licencia expresará claramente los derechos que pueda ejercer sobre la obra.

El texto de las modificaciones reza así:

Por ello, la Asamblea de Extremadura aprueba la siguiente Proposición de Ley para que se modifique el Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.

Artículo uno.

Se añade un nuevo apartado al artículo 17, que queda redactado como sigue:

«Artículo 17. Derecho exclusivo de explotación y sus modalidades.

1. Corresponde al autor de la obra el ejercicio exclusivo de los derechos de explotación de su obra en cualquier forma y, en especial, los derechos de reproducción, distribución, comunicación pública y transformación, que no podrán ser realizadas sin su autorización, salvo en los casos previstos en la presente Ley.

2. A estos efectos y en el entorno digital, se reconoce al autor de la obra la posibilidad de poner a disposición del público sus creaciones mediante las denominadas licencias libres, entendiendo por tales las que posibilitan la cesión de sus obras para los usos que consideren adecuados, que se extenderán desde usos eminentemente restrictivos de la obra, hasta decidir que pase automáticamente a dominio público».

Artículo dos.

Se añade una nueva disposición adicional:

«Disposición adicional quinta. Fomento de la creación Cultural digital libre.

Las Administraciones Públicas establecerán políticas que favorezcan la creación cultural digital libre en sus diferentes manifestaciones, y velarán especialmente por el desarrollo de obras bajo las denominadas licencias libres. Con estas políticas se perseguirá promover un desarrollo cultural armónico del conjunto de los ciudadanos aprovechando las posibilidades que ofrecen las tecnologías de la información y la comunicación.

Dichas políticas abarcarán tanto el software libre, herramienta fundamental para la creación y difusión de obras libres, como a las diferentes creaciones culturales, científicas y artísticas que están permitiendo las tecnologías de la información y la comunicación.

Procurando en todo caso una permanente ampliación del dominio público del conocimiento.»

Aunque por los cambios en la política nacional esta proposición no llegó a afectar a la ley de Propiedad Intelectual, la misma es significativa como muestra de que, en ocasiones, los gobiernos autonómicos muestran acercamientos a los postulados expresados en los apartados anteriores sobre las virtudes de las licencias de software libre como la GPL.

3. Tipografía musical y el proyecto LilyPond

Dentro de la tipografía puede distinguirse, por un lado, entre procedimientos manuales o mecánicos y digitales, y por otro la tipografía general de la musical. Para situar el caso que nos ocupa de tipografía musical por ordenador dentro de este sistema de categorías, véase la figura 6. El gráfico pretende situar a los objetos relacionados con nuestro tema de estudio, dentro del marco más amplio de la composición tipográfica, y contiene ciertos ejemplos de software libre que constituyen casos de aplicación en cada una de estas categorías.

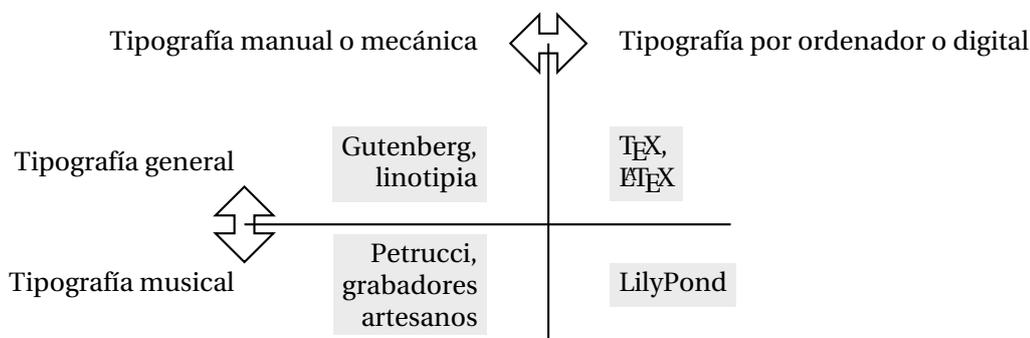


Figura 6: Categorías de la tipografía, atendiendo a su carácter general o musical y a su procedimiento mecánico o digital.

Cada uno de estos apartados será revisado a continuación.

3.1. Tipografía

Parece natural comenzar con una definición del término *tipografía*, aunque hay diversas definiciones que corresponden al doble carácter general y particular de la palabra.

La siguiente definición no es incorrecta aunque sí muy limitada. Tal y como recogen Baines y Haslam (2005), podría haber sido escrita hace cien años o más:

tipografía. 1. f. Arte, técnica o proceso de componer tipos e imprimir con ellos. ||2. Planificación, selección y establecimiento del tipo para una obra impresa.

Collins concise dictionary and concise thesaurus (Sinclair y Gilmour, 2002).

En obras que tratan de ordenadores e informática vemos una acepción más general:

tipografía. f. Del latín medieval *typographia*, y a su vez del griego *tupos*, impresión, y del latín *-graphia*, *-grafía*. Disposición de texto sobre una página impresa, cartel u otro objeto. Se refiere al estilo, tamaño y presentación de los caracteres textuales que forman la fuente.

Computer Desktop Encyclopedia (Freedman y Morrison, 2009): *typography*.

Se sale completamente de nuestro estudio sobre tipografía musical con software libre, una revisión de la historia de la tipografía. Sin embargo es obligado mencionar que la obra más destacada de los inicios de la imprenta con tipos móviles fue la Biblia de 42 líneas de Gutenberg. Según Fuentes y Huidobro (2004):

Con la disponibilidad del papel, la impresión con bloques de madera tallada y la creciente demanda de libros, naciones como Italia, Alemania, Francia y los Países Bajos, buscaban la producción de textos por medio de la mecanización utilizando tipos móviles. Alrededor de 1440, Johann Gensfleisch zum Gutenberg, desarrolló en Alemania una técnica para elaborar moldes de tipos que se utilizarían para la impresión de letras individuales (los chinos fueron los primeros en utilizar tipos móviles, hechos de madera, durante la primera mitad del siglo XI); escogió el tipo *Textura* como modelo, imitando el trabajo de los escribas de la época, para que el nuevo estilo fuera aceptado. La obra más representativa de Gutenberg es su Biblia de 42 líneas que desarrolló con esta técnica, y que estuvo vigente durante 500 años.

3.2. Notación musical

En una línea que va de las representaciones abstractas a las concretas, la notación musical es una representación gráfica que está en el extremo de lo concreto, junto a la codificación. La diferencia entre codificación y notación es que aquella está destinada al manejo por parte de ordenadores, mientras que esta última es para uso humano.

La notación es una representación visual concreta de la música, que comprende gráficos y/o tipografía.

En la semántica general hay dos aspectos: el de la denotación (explícita y bien definida) y el de la connotación (implícita y mal definida). Por ejemplo, tratándose de texto, la prosa es principalmente denotación, mientras que en el arte de la poesía la connotación es mucho más importante. Si la música siempre es arte, entonces es pura connotación. La representación del significado musical plantea numerosos problemas.

La representación es una cuestión de qué información presentar. La notación se ocupa de cómo se presenta dicha información. La diferencia es muy importante; Byrd (1985) presenta el caso de una partitura de piano, extraída de *Le Tombeau de Couperin* de M. Ravel, en la que aparece una singular ligadura con siete puntos de inflexión. La notación es extremadamente compleja, y sin embargo esta complejidad es exclusivamente gráfica. Sus implicaciones para la representación son mínimas y el mensaje es relativamente simple: ligar todas las notas al mismo tiempo que su altura sube y baja.

Al seleccionar una representación para un ente abstracto, se introduce un sesgo. Es más, dada una representación, la elección de una notación (es decir, de una manera de presentar la información gráficamente) introduce inevitablemente un sesgo aún mayor, y a menudo unos límites más estrechos, y la música no es una excepción (Wiggins y col., 1993). Para el inmenso cuerpo de la música compuesta a partir de 1650, no existe otra elección que la notación occidental común. Para ciertos casos concretos existen alternativas como la tablatura, o notación por cifras y/o letras sobre pautas de líneas paralelas. La notación musical occidental común ha tenido un gran éxito y difusión, pero es extraordinariamente compleja y está llena de sutilezas.

La bibliografía sobre notación musical está descrita en el apartado introductorio del marco teórico.

La notación común evolucionó para satisfacer las necesidades de representación de la música clásica, pero tiene un importante campo de utilización en todo el abanico musical: música popular, jazz o para la transcripción del patrimonio musical tradicional. Se trata de un sistema mixto, que combina símbolos gráficos con expresiones textuales y números.

La escritura musical tiene una componente temporal y quizá una de las metas más comunes de la lectura musical sea la producción de una interpretación musical coherente. El lector de música convierte la información visual que recibe en un conjunto de prescripciones para la interpretación: determina qué notas tocar y en qué secuencia y combinación deben producirse (Sloboda, 1984), además de otros muchos contenidos de la notación. Básicamente, la notación musical utiliza símbolos llamados “notas” para representar los sonidos. Los cuatro parámetros básicos de una nota musical son los siguientes, en orden decreciente de importancia (en la mayor parte de la música occidental):

- La altura,
- la duración,
- la fuerza o intensidad, y
- el timbre.

Para la escritura de estos símbolos se usan las convenciones siguientes:

- La altura se representa en el eje vertical, y la clave (, , ) establece el origen de coordenadas;
- la duración se indica por la forma de la figura (, ) o el silencio (, );
- el instante de inicio depende de la suma de las duraciones anteriores, y le corresponde el eje horizontal; la indicación de compás (, ) define una rejilla regular.
- la intensidad del sonido se indica mediante indicaciones textuales como *mf* o *pp* y gráficas como \langle y \rangle ;
- el timbre se indica mediante palabras como “violín”, “flauta” o “pizzicato”.

Existen notaciones alternativas para la música, aunque todas tienen muy difícil abrirse paso en una práctica tan extendida y con tan larga tradición.

La belleza de la notación musical estándar sobre otras alternativas es que, una vez que uno ha superado el nivel básico, puede mirar la partitura e imaginar el sonido, mucho tiempo antes de que el oído alcance un nivel de entrenamiento como para hacerlo con un cien por cien de exactitud. Puede verse a la melodía subir y bajar y es fácil determinar si el ritmo es sencillo o enrevesado, a tiempo o sincopado, etcétera (MacLachlan, 1998).

3.2.1. Meta-principios de la notación musical

Según Byrd (2005), el conjunto de reglas de la notación musical tiene, a su vez, un par de principios comunes. El primero sería maximizar la legibilidad o inteligibilidad de la notación: evitar la acumulación innecesaria de símbolos, usar las abreviaturas convencionales justamente necesarias, y pensar en la audiencia (que para la notación musical son principalmente los intérpretes). Esto es un principio general de cualquier comunicación.

El segundo sería minimizar el espacio utilizado: reducir los pasos de página (lo que es bueno para el intérprete) o reducir costes (lo que, por su parte, es bueno para el editor). Esta comprensión de mucha música en poco espacio es, posiblemente, el principal factor de complejidad de la notación musical común.

En la música, además hay un factor de tiempo real (la lectura a primera vista) que está en la frontera de las habilidades y la preparación del músico.

Es interesante observar que las reglas han evolucionado en general sin ninguna autoridad establecida, y que muchas de ellas son algo imprecisas. Las exigencias de los compositores en cuanto a notación se ha satisfecho por ellos mismos cuando la necesidad lo ha requerido. Al mismo tiempo, los medios técnicos han impuesto límites, como el repertorio de símbolos de la imprenta de Petrucci o de las máquinas de escribir musicales. Además, las reglas interactúan entre sí a causa de la meta-reglas del ahorro de espacio y la concisión en la escritura: tarde o temprano una de las dos tiene que ceder.

En un escrito anterior, Byrd (1994) presenta una selección de casos de la práctica común que muestran esta interacción entre las reglas de la notación y desafían los límites de la corrección y la expresividad.

- Un fragmento del nocturno Op. 15 núm. 2 de F. Chopin en el que la última nota del compás 33 tiene dos duraciones distintas al mismo tiempo: es a la vez semicorchea de tresillo y semicorchea normal. La lógica de la notación obliga (en teoría) a tocar la nota dos veces.
- Otro fragmento del mismo preludio (compases 8 y 9) presenta notas simultáneas, donde

la segunda de ellas viene precedida de un adorno que es posterior a la primera de las notas. La lógica de la notación, en este caso, obliga a leer de derecha a izquierda y tocar el adorno *después de las notas*.

- Un caso, también para piano, del Capricho Op. 76 núm. 1 de J. Brahms. Una blanca con puntillo (compases 26 y 30) aparece con una duración de 11 semicorcheas, en lugar de 12. Obviamente, la notación exacta de esta duración requeriría varias figuras ligadas, pero Brahms no lo hace por motivos de simplicidad visual.

Es probable que los pianistas toquen los ejemplos de Chopin y de Brahms sin notar nada raro, y también es probable que el oyente no perciba ningún fenómeno extraño al escucharlo. El motivo es que la música en sí es un ente abstracto que la notación solo refleja de manera aproximada. La notación musical moderna no pretende reflejar de forma escrita todo el contenido musical. Sin embargo, gran parte de los aspectos del lenguaje musical sí corresponden con elementos de notación, a partir de los cuales un intérprete que *lee* esta notación *interpreta* la misma con su instrumento o su voz para producir la música. Todas las notaciones que durante la historia se han desarrollado, han aportado refinamientos en la exactitud de los signos para satisfacer el objetivo del compositor de transmitir su creación. Pero dado que no es posible registrar absolutamente todo lo que la mente del compositor puede imaginar, debido a las limitaciones del lenguaje escrito, se alcanza un equilibrio entre aquello que desearía transmitir, idealmente, y lo que es más conveniente o útil escribir en la práctica cotidiana del proceso composición-interpretación.

Esto se aplica a otras artes, por ejemplo la notación de la danza es posible cuando el repertorio de posiciones y movimientos es un conjunto bien delimitado como en el ballet clásico, pero para la danza contemporánea, donde la variedad es mucho mayor, se presentan grandes problemas de notación y se recurre al vídeo como transmisor. En ambos casos el intérprete tiene un cierto margen de movimiento, mayor cuanto más escueta o “interpretable” es la notación. En la notación musical ocurre lo mismo, como veremos inmediatamente.

3.3. Orígenes de la notación musical

La evolución de la notación musical viene trazada por Read (1969) y muchos otros. El pueblo mesopotámico ya disponía, según Jofré (2003), de una teoría musical y un sistema de notación. En la Grecia antigua se utilizaba un sistema alfabético para expresar las alturas de entonación. La primera partitura musical completa que se conoce es el epitafio de Sícilo, que aparece junto a una transcripción del mismo en la figura 105, apéndice B.

Posteriormente, la música religiosa bizantina antigua y medieval empleaba un tipo de notación llamado *ecfonético*, consistente en trazos sobre el texto a modo de acentos: puntos y curvas que suben o bajan para expresar convencionalmente modulaciones de altura tonal.

3. Tipografía musical y el proyecto LilyPond

Véase un ejemplo de notación bizantina medieval en la figura 7; se trata de una lista de los símbolos ecfonéticos con su transcripción a neumas melódicos paleobizantinos. Textos hebreos del s.VI muestran un sistema de nueve acentos para indicar la melodía por encima del texto, esquema que fue adoptado por los repertorios monódicos de las iglesias bizantina, siria y armenia.

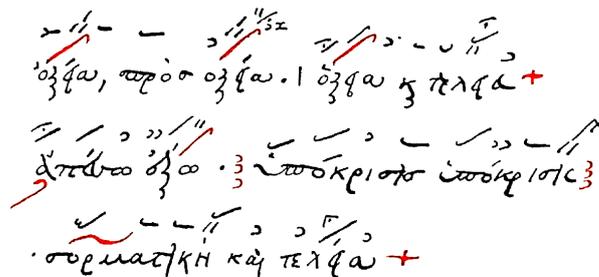


Figura 7: Notación musical ecfonética bizantina. Fuente: MS Sinai 8 (Prophetologion, s.X/XI.), f. 303r. Tomado de Lingas (2014).

La notación ecfonética derivó en la notación neumática¹ que sin duda es un ancestro de nuestra notación actual. Para la notación del canto gregoriano se empleó un sistema bien estructurado de símbolos llamados «neumas» que en un principio se situaban sobre el texto sin altura definida (notación *in campo aperto* o adiaستمática, véase la figura 8). Más tarde se disponen sobre una línea de referencia, posteriormente dos líneas de distintos colores para marcar las alturas del Do y del Fa, y finalmente mediante numerosas líneas paralelas (notación *diastemática*), que para el canto gregoriano oficial, la edición Vaticana, se estabilizó en un número de cuatro. Salazar (1989) profundiza en el caso español distinguiendo entre todas sus variantes.



Figura 8: Fragmento de un «Alleluia» que presenta un ejemplo de notación neumática *in campo aperto* (sin líneas). Fuente: Catedral de Worcester (R.U.), MS F173, f. 6v. Tomado de Barton, Terzopoulos y Craig-McFeely (2007).

En el s. XIII aparece la notación mensural, en la que signos equivalentes a nuestras *figuras* expresan distintas duraciones según su forma. La notación mensural blanca, cuya aparición coincide con las primeras ediciones de música impresa, aporta además una distinción entre figuras «huecas» y «rellenas» que determina su valor respectivo.

¹El adjetivo «neumática» proviene del griego *pneuma*: soplo, respiración.

3.4. Tipografía musical

Gomberg (1977) afirma que Peter Schoeffer utilizó tipos móviles para la impresión de música en 1490 y que con anterioridad toda la producción de música se había realizado a mano. Esta fecha es muy tardía y se ha revisado posteriormente. En cualquier caso, Read (1969) revela un importante efecto de la aparición del proceso de impresión de música sobre la notación musical: en cuanto los punzones de símbolos musicales quedaron estandarizados después de la llegada de la imprenta, también la notación musical se estabilizó cuando la impresión de música sustituyó a la copia manual.

Por su gran importancia es necesario mencionar la figura de Ottaviano Petrucci (1466-1539), que consigue una patente en 1498 sobre impresión de música (Kostylo, 2008) y publica en 1501 el primer libro completamente musical elaborado con tipos móviles, «*Harmonice Musices Odhecaton*»,² una colección de 96 piezas polifónicas.

Según Kostylo (2008), la patente de Petrucci es el primer privilegio concedido para la impresión de música, y también uno de los primeros relativos a invenciones y mejoras del mecanismo de la imprenta en general.

De Petrucci (2001), un facsímil de la tercera edición (1504) del «*Odhecaton*», podemos ver una muestra en la figura 9 que es un bello ejemplo de notación mensural blanca. Contiene las cuatro voces de la *chanson* «*Por quoy non*», de Pierre de la Rue, pieza número 18.



Figura 9: Doble página del «*Harmonice Musices Odhecaton*», impreso por primera vez en 1501 por Ottaviano Petrucci.

En «*Los orígenes de la imprenta musical*», Mele (2004) explica en términos semejantes a los de Read (1969), pero sin nombrar a Schoeffer, que las ediciones de Petrucci dieron forma a la notación estándar de las décadas que siguieron porque los compositores tuvieron que atenerse a las convenciones y al conjunto de símbolos que Petrucci había establecido para

²Erróneamente nombrado como «*Odhedeon*» por Jofré (2003).

sus ediciones, es decir, forzó la adaptación de los compositores a un estilo de escritura, dando unidad y sistema a la notación musical.

A pesar de la pretensión del propio Petrucci de haber sido el inventor del arte de imprimir música, en realidad no fue el primero en términos absolutos. La obra más antigua conocida que contenía música era el *Psalterium*, que apareció en Mainz en 1457 a partir de la prensa de Johann Fust y Peter Schoeffer. Tampoco fue Petrucci el primero en usar la técnica de la doble impresión. La opinión generalizada actualmente da mucho más crédito a los impresores del s.XV por preparar el terreno (Kostylo, 2008).

Petrucci retiene el mérito de haber sido el primero que imprimió música polifónica a gran escala y el primero en haber establecido un mercado para la música impresa como un sector diferenciado, dentro del negocio de la publicación.

El grabado en placas de cobre se estaba usando desde 1452 para usos puramente decorativos, pero no fue adoptado por la música. Dice Ross: “Parece incomprensible que este proceso no se aplicara a la música hasta la última parte del s.XVI, dado que estaba bien establecido y no requería experimentación” (Ross, 1987). El grabado en metal estuvo compitiendo con la tipografía aproximadamente hasta 1700, cuando las placas de peltre³ hicieron del grabado un proceso mucho más barato. Hacia la misma época, punzones de acero con “caracteres” musicales aceleraron el trabajo. Hacia 1730 muy escasa proporción del grabado en metal se hacía con buriles. La música se imprimía directamente a partir de las planchas, ya fueran grabadas con buril o hendidas con punzones, hasta 1860 aproximadamente, en que se inventaron métodos para reproducir las planchas. En esencia, aunque el grabado musical ha permanecido sin cambios durante más de doscientos años, hasta los años 90 del s.XX se estuvo considerando el mejor método para la reproducción de música, a causa de su calidad uniforme. Paralelamente se había estado empleando el proceso de litografía, que data de 1796 y se usó casi inmediatamente para la música.

La notación musical occidental actual es un sistema plenamente consolidado desde el s. XVII, según Jofré (2003). Con el fragmento de música etiquetado que aparece en la figura 10 ejemplificamos la mayoría de los elementos de la notación musical occidental común.

Llegados a este punto, vamos a delimitar el significado que se da en tipografía a la palabra *componer* (RAE, 2003):

componer. (Del lat. *componĕre*). **1.** tr. Formar de varias cosas una, juntándolas y colocándolas con cierto modo y orden.||**2.** tr. Constituir, formar, dar ser a un cuerpo o agregado de varias cosas o personas. Apl. a las partes de que consta un todo, respecto del mismo, u. t. c. prnl.||(...) **13.** tr. Impr. Formar las palabras, líneas y planas, juntando las letras o caracteres. (...)

³El peltre es una aleación de estaño con cobre, antimonio y plomo, que hace un sustituto del estaño con mayor dureza.

durante muchos años, se pudo encontrar de forma predeterminada en varios cientos de miles de máquinas, muchas más que ningún otro formateador de texto para Unix. Se originó en los laboratorios de la compañía AT&T Bell en 1971 y su autor fue Joseph Ossanna. Su padre legítimo fue un programa llamado `runoff`, escrito en el Instituto de Tecnología de Massachussets en la década de 1960.

Fue Donald E. Knuth quien dio un notable impulso a la tipografía mediante ordenador al desarrollar en la Universidad de Stanford su sistema Metafont a partir de 1977 y el sistema $\text{T}_{\text{E}}\text{X}$ durante el período 1978-1985. Posteriormente continuó experimentando con la notación matemática por ordenador durante la preparación del libro *Concrete mathematics* (Graham, Knuth y Patashnik, 1988).

La primera versión de Metafont es de 1979. La idea de Metafont es definir las formas de los tipos programáticamente, es decir, mediante expresiones matemáticas que producen lugares geométricos mediante resolución de ecuaciones. Por ejemplo: en lugar de trazar el contorno de una forma, se define una plumilla virtual y se traza el camino que sigue la plumilla como una función o un encadenamiento de funciones: el límite externo de dicho trazo es la forma del glifo.

El nombre de $\text{T}_{\text{E}}\text{X}$ proviene del griego $\tau\epsilon\chi$, Tau-Epsilon-Xi, que es la raíz de palabras como «técnica» y «tecnología». El primer artículo de Knuth sobre tipografía es «Mathematical Typography» (Knuth, 1979). Knuth es matemático, no tipógrafo, y no está suficientemente bien reconocido en los textos de tipografía posteriores a él. Por su parte, un diseñador profesional tiene una tarea colosal por delante si quiere usar Metafont para diseñar sus tipos. Posiblemente ambos factores se combinan para hacer de Metafont una solución minoritaria pero de gran calidad potencial e infinitas posibilidades sobre el papel.

En lo que concierne al ámbito de la presente investigación, tiene importancia el hecho de que las fuentes tipográficas libres Feta, Parmesan y Emmentaler están definidas en el lenguaje Metafont, que es libre a su vez.

El programa de tipografía $\text{T}_{\text{E}}\text{X}$ se amplió envolviéndolo en un sistema de conveniencia, llamado $\mathbb{T}_{\text{E}}\text{X}$,⁴ desarrollado por Leslie Lamport en los laboratorios de la compañía Digital Equipment. Este nuevo conjunto de herramientas se creó bajo la denominación de «procesador de documentos» más que como un programa para la composición tipográfica.

La Sociedad Estadounidense de Matemáticas (AMS) estuvo desarrollando y utilizando $\text{T}_{\text{E}}\text{X}$ desde 1983 hasta 1985 por su especial adecuación a la tipografía matemática, siendo $\mathcal{A}\mathcal{M}\mathcal{S}$ -

⁴Muchos de estos programas surgen en un entorno universitario pero se inician como proyectos personales y tienen ciertos aspectos ligeramente humorísticos. Al escribir sobre ellos hemos optado por respetar la tipografía original de algunos de sus nombres, que al mismo tiempo son logotipos. Entre ellos se encuentran los mencionados $\text{T}_{\text{E}}\text{X}$, $\mathbb{T}_{\text{E}}\text{X}$ y derivados suyos como $\text{BibT}_{\text{E}}\text{X}$, pero también $\text{S}_{\text{C}}\text{E}\text{X}$, $\text{M}^{\text{P}}\text{P}$, $\text{Ti}\text{k}\text{Z}$ y otros.

$\text{T}_{\text{E}}\text{X}$, escrito originalmente por Michael Spivak, un requisito para publicar en los diarios de la sociedad. Cuando el desarrollo de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ alcanzó una madurez suficiente, también la AMS pasó a utilizarlo y fomentarlo mediante la distribución de sus propios paquetes como plantillas, denominadas $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, que los autores deben usar en sus publicaciones.

3.6. Tipografía musical por ordenador. Orígenes de LilyPond

El lugar exacto que ocupa GNU LilyPond como pieza de software libre de tipografía musical, se presenta dentro del mapa de la figura 11 junto a MuseScore.

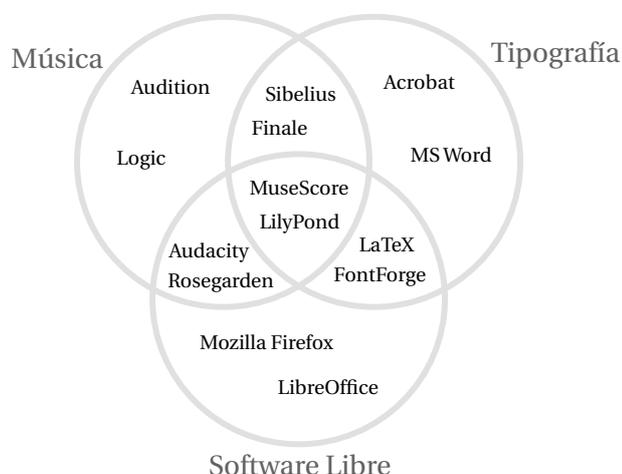


Figura 11: Situación de GNU LilyPond dentro de las categorías de software relacionadas con el tema de la investigación.

Con una aplicación específica como es la producción de partituras, LilyPond no es un programa educativo; pero la producción y el consumo de partituras musicales es central dentro del ámbito de la educación musical. Por otro lado, las soluciones de software no libre para esta aplicación tienen un problema para su difusión amplia en entornos educativos: dada la existencia de una solución alternativa, los estudiantes y los profesores no deberían tener que pasar por la situación incómoda de *piratear* obligatoriamente un programa cuyo precio no está a su alcance, según se expuso anteriormente. Esto es lo que se denomina *el problema de la tipografía*. La opción libre, sin embargo, no ofrece simplemente una alternativa “gratis”, ya que la copia no autorizada también es “gratis” en la práctica. El software libre ofrece una licencia que cualquiera puede fácilmente cumplir; la única objeción sería que no cumpliera con unos estándares de calidad, y precisamente demostrar dicho punto es uno de los objetivos del presente trabajo.

LilyPond es un sistema de **grabado musical automatizado**. Veamos el significado que tiene el término *grabar* en este contexto (RAE, 2003):

grabar. (Del fr. *graver*). **1.** tr. Señalar con incisión o abrir y labrar en hueco o en relieve sobre

una superficie un letrero, una figura o una representación de cualquier objeto.||(...)

El grabado es el método de composición tipográfica de música que se ha estado utilizando históricamente, casi hasta la llegada de los sistemas eléctricos, electroquímicos, fotográficos e informáticos. Consistía en labrar mediante un buril y en forma inversa, como en un espejo, las incisiones correspondientes a la pauta y a la forma de las notas y símbolos musicales, sobre una plancha de cobre que, después de entintada, imprimía la notación sobre un papel.

Existe un mito popular de la tipografía musical por ordenador con un ejemplo en la siguiente cita tomada del manual divulgativo «¿Cómo se hacen las cosas? Partituras musicales.» (The Gale Group, 1998) en la que el autor ha ejercitado su fantasía, pero que resulta completamente falsa:

Con frecuencia, el editor o el servicio de notación musical tiene un teclado como el de un piano, conectado al ordenador, y la notación se introduce tocando la pieza.

La notación musical usual difiere tanto de los detalles espacio-temporales de la ejecución de la música en un instrumento, que esta traducción automática de música a partitura no tiene mucho sentido porque la correspondencia entre notación y ejecución admite unos márgenes de interpretación muy amplios, incluso si eliminamos cualquier resto de interpretación artística o expresiva. Las ejecuciones en un teclado más mecánicas que podamos imaginar, no pueden remontar un proceso, el de la lectura musical, que es de un solo sentido, es decir, no tiene vuelta atrás, no es factible la reconstrucción de la partitura a partir de su interpretación. Sin embargo, nunca desde que el sistema MIDI de interconexión entre teclados y ordenadores se desarrolló en los años 80, ha dejado de resultar atractiva la idea de tocar una pieza y obtener inmediatamente la partitura impresa de lo que hemos tocado.

La búsqueda de un sistema para imprimir música ha atraído la atención del mundo universitario en los campos de los lenguajes formales, los gráficos por ordenador, la inteligencia artificial y la teoría de la información. En los siguientes párrafos examinamos algunos de los logros que históricamente son más destacados. En el espíritu abierto del ambiente universitario, el software libre ha sido el objeto en que muchas de estas investigaciones se han concretado, mucho antes de que el gran público con interés en la impresión de música tuviera a su disposición productos de software comercial. La mayoría de las aproximaciones al problema han pasado por el desarrollo de una técnica para codificar la música mediante algún lenguaje cuyas reglas sintácticas se puedan verificar de forma mecánica. El procedimiento, con diversos grados de éxito, suele consistir en el proceso del código escrito en este lenguaje para producir, en un número variable de fases, un resultado que se pueda imprimir.

Desde la aparición del sistema operativo UNIX en 1969 hasta el lanzamiento de LilyPond 1.0 en 1998 y MuseScore en 2002, se muestra en la figura 12 un mapa cronológico de los principales desarrollos acaecidos en el tema de la investigación.

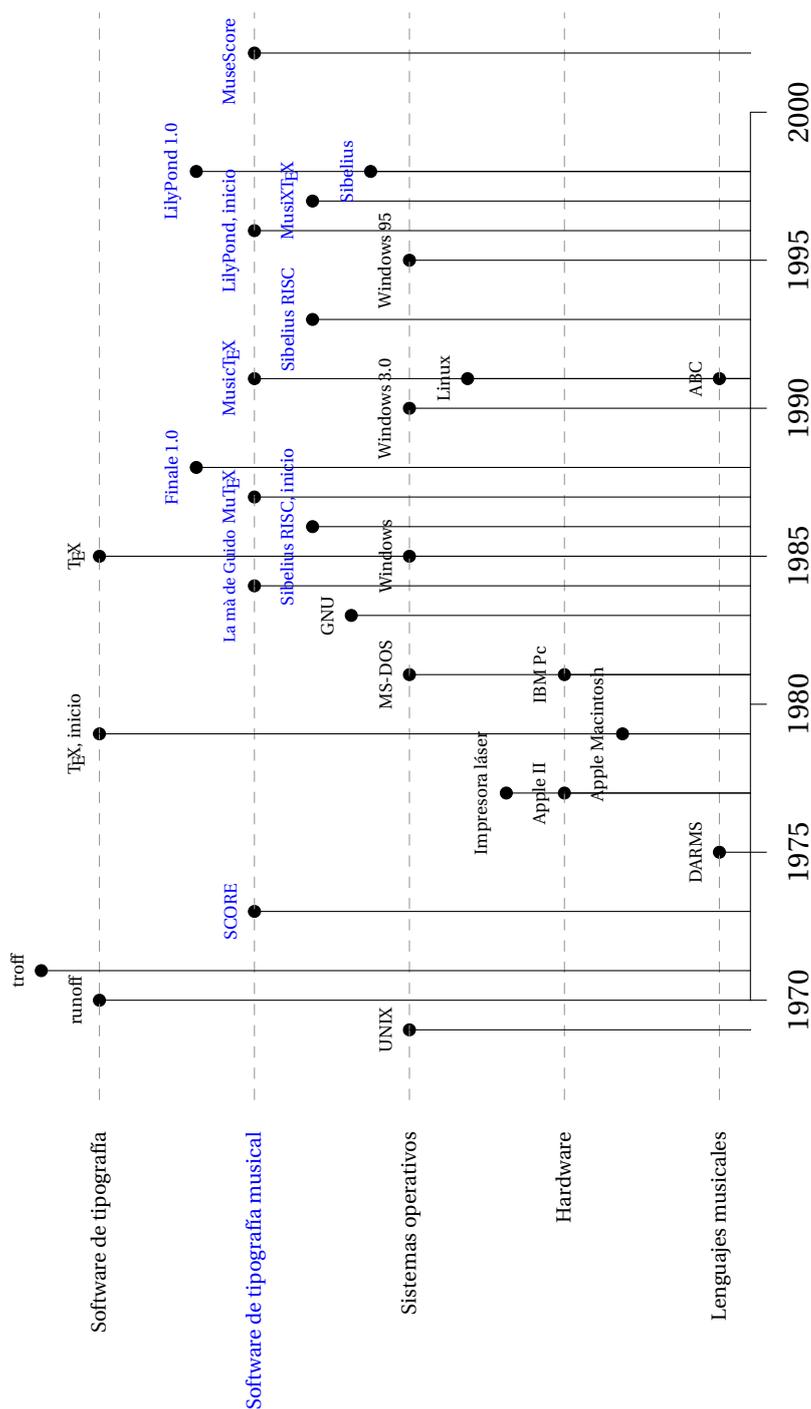


Figura 12: Mapa cronológico de los principales acontecimientos en el desarrollo de la informática y la tipografía musical por ordenador.

En 1975 aparece el proyecto DARMs⁵ que fue el método elegido para codificar la música en

⁵DARMs son las siglas de *Digital Alternate Representation of Musical Scores*: Representación digital alternativa de partituras musicales.

una gran base de datos de *incipits* o comienzos de piezas de música polifónica del renacimiento italiano, dirigida a investigadores interesados en este período. Este lenguaje de descripción musical está descrito en Erickson (1975) y la base de datos en Lincoln (1977). El mismo año surge una tesis doctoral en la Universidad de Washington sobre impresión musical (Gomberg, 1975). Este trabajo es una discusión sobre los problemas de representar las convenciones de la notación musical en forma de algoritmos de ordenador.

El artículo «Las gramáticas como representaciones de la música» (Roads, 1979) es contemporáneo del trabajo de Donald Knuth sobre tipografía. Podemos ver la manera en que las teorías sobre la representación de la música como un lenguaje y su traducción a un lenguaje de ordenador, coinciden en el tiempo con tímidas implementaciones de estas teorías en sistemas reales capaces de producir música impresa.

El sistema completo y funcional de notación musical por ordenador más antiguo que conocemos, y también el primero en distribuirse de forma comercial, es SCORE de Leland Smith, del año 1979 (observemos que el MS-DOS es de 1981), escrito en el lenguaje FORTRAN para los ordenadores de la familia PDP-10. Smith ya había producido la primera partitura por ordenador en 1971, no con una impresora, sino mediante una mesa trazadora de plumillas; sus hallazgos están descritos en Smith (1973). En este momento inicial, las partituras no presentaban curvas, sino que los objetos se creaban a base de cortos segmentos de recta. Aunque parezca increíble, las principales editoriales de música actuales aún utilizan este sistema, según el propio Smith. No se trata de software libre, de hecho el programa cuesta 1.000 dólares en su versión para MS-DOS. Este proyecto quedó estancado hasta que su autor lanzó en 2008 una versión «beta» para Windows.⁶

Subjetivamente, a juzgar por el resultado visual de las ediciones de aquellos impresores a los que Smith afirmaba proveer su software (Barenreiter, Breitkopf & Hartel, Peters, etc) el programa SCORE permite afinar los detalles de la salida hasta un grado muy profundo y conseguir así una buena calidad. No se puede ignorar que estas editoriales han dispuesto tradicionalmente de grabadores profesionales del más alto nivel de destreza, por lo que suponemos que el programa se encuentra en manos de expertos. Al mismo tiempo, el programa tiene reputación de ser extremadamente difícil de manejar, lo que es otro dato subjetivo y difícil de constatar como consecuencia de lo dicho anteriormente acerca de su precio. Sí sabemos que el enfoque de SCORE está centrado exclusivamente al aspecto gráfico del grabado, es decir, no maneja el lenguaje musical como tal, y no permite ningún tipo de operaciones sobre la música en función de su contenido como, por ejemplo, la transposición⁷ o el despliegue de repeticiones.⁸

⁶Smith falleció en diciembre de 2013, a los 88 años. Dejó la versión de SCORE para Windows en estado de «beta» o prueba provisional y como tal se comercializó.

⁷La transposición es un procedimiento de la escritura y la lectura musicales, que consiste en subir o bajar la altura de todas las notas en una medida fija, lo que puede afectar a un gran número de símbolos.

⁸El despliegue de repeticiones es la aplicación de una regla común de la escritura musical, la abreviatura de fragmentos repetidos.

Aunque ambos procedimientos son sencillos para una persona con conocimientos musicales, este tipo de manejo simbólico del contenido musical no es fácil de implementar en un programa de ordenador.

Al sistema SCORE de L. Smith le sirvió de competencia un notable caso español: *La mà de Guido Music Setting System*, de Llorenç Balsach (Puig, 2003), que en 1984 ganó el premio Simolog de informática y que fue presentado en la feria musical de Francfort en 1986. Como todos los productos de una época en la que no existían Windows ni impresoras láser o de inyección de tinta, estaba basado en una consola de texto, y la salida gráfica se producía a través de una mesa trazadora de plumillas. Tanto en SCORE como en *La mà de Guido* los trazos curvos se componían de segmentos rectilíneos y no existía la variedad de fuentes tipográficas de que puede disponerse actualmente.

Tenemos otros antecedentes en el desarrollo de sistemas de notación musical por ordenador en lenguajes como el esbozado por Gourlay (1986), trece años posterior al primer artículo de Smith, y poco después el sistema Music, de Eric Foxley (Foxley, 1987). Todos estos sistemas son no visuales: definen una sintaxis capaz de recrear la estructura del texto musical y reinterpretan este código para producir notación impresa. Los sistemas visuales se encontraban con una plataforma aún muy joven sobre la que funcionar: los primeros ordenadores con ratón fueron los Apple Lisa y Macintosh de 1983 y 1984, respectivamente. La primera versión de Windows aparece en 1985. Cualquier software anterior había de manejarse exclusivamente a través del teclado, como el Personal Composer de 1983, que aún sigue vigente. Un precursor de los sistemas visuales, muy avanzado para su época, fue el sistema Mockingbird (Ornstein y Turner, 1983, 1984), que se desarrolló en el centro de investigación de Xerox en Palo Alto, y que estaba orientado a facilitar la tarea de los compositores mediante herramientas gráficas interactivas.

Es necesario aclarar que el manejo visual de un programa de notación musical no es un objetivo en sí mismo; codificado de la forma adecuada, es posible dar formato a una variedad de objetos (música, texto o dibujos) mediante enfoques basados *exclusivamente* en texto, que ya han sido discutidos aquí. Como se ha dicho, los documentos de texto tienen la propiedad de que pueden ser leídos por los humanos además de por el programa que los procesa, a diferencia de los formatos binarios, que las personas no pueden leer directamente. Si además la sintaxis de estos documentos de texto fuese pública y su uso no estuviese restringido, estaríamos ante un formato abierto, que es una de las bases del software libre. Ofrecemos más adelante, en la sección 3.7 (Interfaces de texto) una discusión más profunda del concepto de interfaces de texto para la tipografía general y musical.

Pueden encontrarse descripciones más detalladas de algunos de los siguientes antecedentes en Goossens y col. (1997).

Los sistemas Mu $\text{T}_{\text{E}}\text{X}$, de Andrea Steinbach y Angelika Schofer (Steinbach y Schofer, 1987),

MusicT_EX de Daniel Taupin (Taupin, 1993), inspirado en el anterior, y MusiXT_EX (Taupin y col., 1995) de Daniel Taupin, Ross Mitchell y Andreas Egler, aparecen ya como sistemas muy próximos a LilyPond: sistemas basados en T_EX para realizar notación musical. Estamos hablando de desarrollos libres y formatos abiertos en todos los casos, amparados por un entorno universitario.

MuT_EX solo podía producir monodias en un pentagrama, pero superaba a MusicT_EX en el manejo de las ligaduras y en el espaciado de las notas. Sin embargo, las dos autoras dejaron de desarrollarlo, mientras que Daniel Taupin y su equipo continuaron mejorando su sistema.

El sistema MusiXT_EX y sus predecesores inmediatos no estaban indicados para traducir notación musical estándar en código de T_EX, ni trataban de incluir ninguna consideración estética (Goossens y col., 1997). Componían la información gráfica siguiendo ciegamente las instrucciones del compositor humano. En un principio el sistema LilyPond también utilizaba a T_EX como motor de tipografía. Hasta el año 2008 no se eliminaron del código fuente los últimos vestigios de esta arquitectura, y ahora la salida de notación musical de LilyPond pasa por el lenguaje de descripción de páginas PostScript que después se transforma en PDF. Los sistemas mencionados disponían de un preprocesador llamado PMX (Cataldi, 2003; Simons, 2004), obra de Don Simons, pensado para simplificar la introducción de música instrumental polifónica, y a su vez se podía preprocesar con M-Tx (Laurie, 2005), de Dirk Laurie, cuya funcionalidad era la introducción de las letras de las canciones. En su descripción de M-Tx, Laurie explica que la implementación descansa sobre PMX, MusiXT_EX y MusiXLyr, de Rainer Dunker, descrito en Dunker (2003). Una sintaxis muy simplificada podía procesarse mediante el programa `abc2mtex` (Walshaw, 1993), que estaba pensado para realizar la notación de canciones almacenadas en código ASCII, la llamada «notación ABC». El desarrollo de MusiXT_EX se mantuvo algún tiempo después de la muerte de Daniel Taupin,⁹ y la última versión es de junio de 2006.

El mencionado lenguaje de la notación ABC fue inventado por Chris Walshaw en 1991; su diseño estaba orientado a canciones tradicionales de origen europeo occidental, como las de Irlanda, Inglaterra y Escocia, que se pudieran escribir sobre un solo pentagrama en notación clásica estándar. Nada impedía, en principio, su utilización para otros tipos de música, pero estos presupuestos dan una idea de sus limitaciones. La objeción que desde LilyPond se viene sobre el estándar ABC es que no es muy «estándar», es decir: no está formalizado, no tiene una sintaxis estricta. ABC no es un programa, sino una convención incompleta para un lenguaje; la visualización de la música escrita en este lenguaje depende de convertidores como el referido `abc2mtex` o bien `abc2ps`, de Michael Methfessel, o el reciente ABC::DT (Azevedo y Almeida, 2013), o bien de algunas de las muchas soluciones para Windows.

⁹Taupin falleció en 2003 en un accidente de escalada. El programador David Kastrup, principal desarrollador de LilyPond a fecha de 2015, es un gran aficionado a este deporte.

Posteriormente, los artículos de Miguel Filgueiras, del departamento de informática e inteligencia artificial de la Universidad de Oporto (Filgueiras, 1996, 1998; Filgueiras y Leal, 1993), aportan numerosas adaptaciones de las convenciones de la notación musical a las exigencias formales de los programas informáticos. En ellos, Filgueiras presenta aspectos del desarrollo del sistema ScTeX , aparentemente estancado desde 1999.

Byrd (1994) presenta en su artículo “Software de notación e inteligencia” cuatro fragmentos de notación problemáticos que utiliza como base para entonar un canto de protesta en contra de los programas de notación que tratan de exhibir un comportamiento inteligente. Ello nos da una idea del estado, aún poco desarrollado, que debían tener estos programas.

La línea de sucesión directa que conduce a LilyPond se inicia en los antecesores de M^{PP} y su sucesor M^{PP95} . Se trataba de preprocesadores que interpretaban código de notación musical insertado dentro de documentos de MusixTeX y producían otro código en TeX estándar, que luego se sometía al proceso normal de este lenguaje para producir el resultado gráfico.¹⁰

Los dos autores originales abandonaron el enfoque de tomar un código que se debía preprocesar; el resultado debía someterse posteriormente a un motor de tipografía para su representación gráfica. Al considerar que el proyecto M^{PP} no daría posiblemente los frutos apetecidos, se embarcaron en la programación de un nuevo sistema que a la larga realizaría el ciclo completo en un solo paso. Aprovechando parte del código de M^{PP} , el embrión del sistema LilyPond surge en 1996, y su primera versión provisional ve la luz en 1997. La versión 1.0 está fechada en el 31 de julio de 1998.

Por las mismas fechas se presentaba en la academia un sistema con ciertos puntos en común, conocido como GUIDO (Hoos y col., 1998) que muchos años después ha seguido vigente (Hoos y col., 2001; Renz, 2002).¹¹

El lenguaje de representación del contenido musical utilizado en LilyPond recibió en un principio el nombre de “Mudela”, procedente de las palabras *Music Definition Language* (Lenguaje de definición de música). Según Droettboom (2000), que compara a Mudela con GUIDO, Mudela está ligado más estrechamente a las consideraciones prácticas de la composición tipográfica musical.

Mudela se originó como el lenguaje de entrada al “compilador musical” LilyPond basado en TeX , y ahora es también el lenguaje oficial del “Proyecto Música” de GNU. Su forma original proviene claramente de la representación *Tilia* (Haken y Blostein, 1993) usada por el editor musical Lime (Blostein y Haken, 1991, 1994). Mudela no tiene una especificación formal, solo una *guía del usuario* preliminar (Nienhuys, Nieuwenhuizen y Percival, 2015a).

¹⁰Los antecesores de M^{PP} fueron obra de Jan Nieuwenhuizen, en la Universidad Técnica de Eindhoven; junto a Han-Wen Nienhuys trabajó sobre el propio M^{PP} ; Nienhuys trabajó posteriormente en solitario sobre M^{PP95} .

¹¹No debe confundirse el formato de notación musical GUIDO de Holger Hoos con *La mà de Guido* de Llorenç Balsach, que fue un desarrollo español iniciado en 1984 convertido en una editorial y sello discográfico con el mismo nombre.

A partir de 2008, las obligadas matizaciones al párrafo anterior son que LilyPond ya no está basado en T_EX, que efectivamente posee una especificación formal y que las guías del usuario no son ya exactamente preliminares.

El desarrollo de las primeras versiones de LilyPond como sistema autónomo se produjo de forma simultánea con el diseño de las tipografías *Feta*. Feta es un conjunto de fuentes codificadas en el sistema Metafont que define los glifos¹² de todos los símbolos musicales y componentes de las figuras, como las cabezas de nota o los corchetes.

El uso de un lenguaje formal como Metafont para el diseño de las tipografías implica que los cambios en el código de un glifo permiten recrear la colección completa de fuentes, para todos los tamaños. De la misma forma que los tipos Computer Modern de D. Knuth para T_EX, los tipos Feta adaptan dinámicamente el grosor del trazo de un carácter al tamaño general de la tipografía, lo que produce un efecto parecido al grabado en metal de símbolos pequeños con un tamaño similar de la punta del buril grabador.

Esta falta de proporcionalidad, intencionada, está enfocada a la producción de una notación con más peso visual en los tamaños menores, de forma transparente para el usuario.

3.6.1. LilyPond como software libre

Si retomamos el gráfico de la figura 3 (pág. 28) podemos situar LilyPond en él, enfrentado a las soluciones más populares para edición de partituras. Véase el gráfico 13 para apreciar lo siguiente:

- LilyPond es software libre y de código abierto, y está licenciado bajo la GPL,¹³ que es un tipo de protección *copyleft*.¹⁴ Su descarga es gratuita.
- Los programas Finale (Powell, 2007), Sibelius (Bielawa, 1993) y Encore (Rothstein, 1993) son software privativo y cerrado, que no cabe en la categoría “Shareware”. Su descarga no es gratuita.

En los orígenes de LilyPond se encuentra la idea de que la amplia difusión de los así llamados «editores de partituras» acabó, según Nienhuys y col. (2015a) con el arte de la tipografía musical y condujo a partituras baratas pero de aspecto mecánico, con poco atractivo visual. LilyPond no es un editor de partituras, sino un sistema para la composición tipográfica de la música en el espíritu de las ediciones *grabadas* a mano de la antigüedad.

¹²Recordemos que un glifo es la forma de un carácter.

¹³En noviembre de 2009 se elevó la licencia de LilyPond a la versión v3+ de la GPL.

¹⁴Véase el glosario, apéndice G.

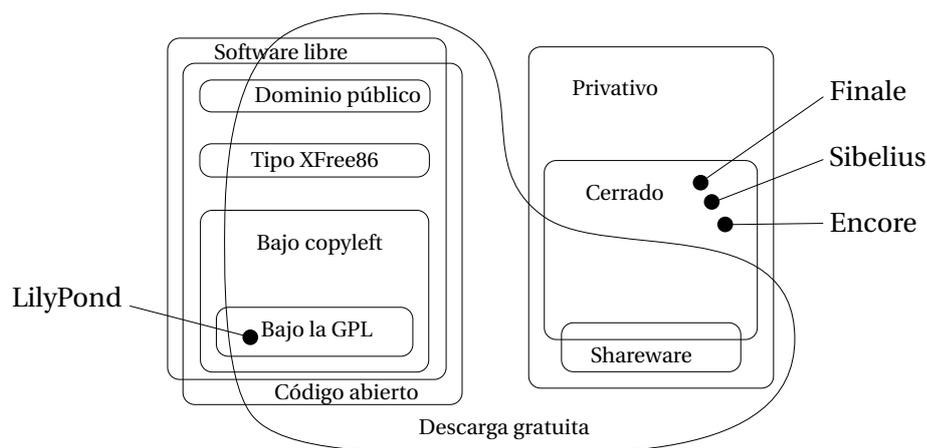


Figura 13: Situación de LilyPond frente a los programas de tipografía musical más populares dentro del esquema de las categorías del software, respecto del concepto de libertad.

Como se dijo en la nota histórica, el proyecto LilyPond (Nienhuys y Nieuwenhuizen, 2003) fue iniciado en 1996 por los músicos y programadores holandeses Han-Wen Nienhuys (intérprete de trompa) y Jan Nieuwenhuizen (intérprete de viola) como un proyecto personal surgido del ambiente musical de la Joven Orquesta Sinfónica de Eindhoven. En el *Manual de aprendizaje* (Nienhuys y col., 2015a) cuentan:

Igual que hacer música, trabajar en LilyPond puede ser 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!¹⁵

LilyPond aparece en las actas del XIV congreso de informática musical de Florencia, en mayo de 2003. Posteriormente otros desarrolladores muy activos como David Kastrup (Alemania), Joe Neeman (EE.UU.), Graham Percival (Canadá), John Mandereau y Valentin Villenave (Francia), Mats Bengtsson (Suecia), Trevor Daniels (Reino Unido) y otros hasta un total de 153 personas, entre asiduos y esporádicos, han tomado el relevo en las tareas de desarrollo, docu-

¹⁵Traducción del autor.

mentación, seguimiento de fallos y soporte a los usuarios, todo ello de forma desinteresada. La lista completa de los 153 desarrolladores históricos¹⁶ puede verse en la página 317.

3.7. Interfaces de texto

Debe distinguirse claramente entre los procesos de edición y formateado, cuando se habla de composición tipográfica. Después aplicaremos esta teoría al campo de la notación musical.

Se aprecian dos tareas principales en la preparación de un documento: la edición del texto y su composición. Editar es cambiar el texto. Normalmente, cuando tecleamos algo en el ordenador, estamos casi siempre editando, no formateando. Pero observemos los editores de texto más populares: también dan formato al texto, por lo que se ocupan de las dos tareas. Hemos visto a veces ejemplificado este paradigma con la metáfora de la autocaravana; una autocaravana es un coche y al mismo tiempo una vivienda. No es demasiado buena como ninguna de las dos cosas, ya que un buen coche seguramente tendría una mejor respuesta, y una buena casa suele ser mejor vivienda. La ventaja, si tiene alguna, está solamente en la combinación de ambas funcionalidades en un solo paquete.

Uno de los métodos de funcionamiento más extendidos entre los programas informáticos que producen una salida impresa (pónganse por caso los editores de texto, de partituras...) es el que se conoce por las siglas WYSIWYG,¹⁷ que viene a significar: «lo que se ve es lo que se obtiene». Este método tiene sus ventajas para los usuarios ocasionales y gran parte de los usuarios *serios*. También tiene grandes inconvenientes, principalmente para aquellos usuarios que buscan un control detallista y al mismo tiempo poder ajustar fácilmente el aspecto general de documentos extensos.

Los editores de este tipo realizan lo que se conoce como edición *rica*: se puede seleccionar el texto y darle una forma itálica o negrita, o ambas. A este proceso se le llama formateado, porque cambiamos el formato del texto. En el mundo de los procesadores de texto, “edición” es un término que se refiere a una combinación de ambas tareas, siendo el “formateado” una parte de la edición. Pero dentro de una terminología informática más estricta, el formateado no es parte de la edición, es una actividad separada. Los distintos usos de estas palabras en diferentes contextos pueden resultarnos confusos si nos estamos planteando por primera vez sus significados reales.

La “edición rica”, escribir el texto y formatearlo al mismo tiempo, es mejor para documentos cortos y más rápida de aprender para usuarios esporádicos o que no tienen tiempo de aprender otro sistema. Los editores visuales tipo Word son más eficientes para cierto tipo de

¹⁶En nuestro estudio de 2009, la lista era de 71 desarrolladores.

¹⁷WYSIWYG son las siglas del inglés *What You See Is What You Get*.

documento breve y sin muchas fórmulas matemáticas (Knauff y Nejasmic, 2014). Aparecen, problemas, sin embargo, derivados del *marcado oculto*. Los cambios de estilo (negrita, itálica, etc.) están insertados en el texto pero son invisibles, y uno nunca está seguro de tener el cursor en el lado correcto de una marca oculta de formato. Como consecuencia, es frecuente obtener un formato distinto al deseado. Se puede caracterizar la inmediatez de la edición rica como “interactiva” y “volátil”.

Habitualmente, según Gram y Cockton (1996), se ensalza la interactividad del software como un requisito para el ejercicio de la creatividad y el carácter intuitivo como la mejor vía para aprender el manejo de un programa, y es posible que así sea. Sin embargo, se observa que la volatilidad está en razón directa de la interactividad. Recordemos cuándo fue la última vez que dijimos frente a un documento “*No sé lo que he hecho, pero algo ha cambiado y no puedo dejarlo como estaba*”. Esto es un síntoma de la falta de robustez de cualquier sistema interactivo que, insistimos, sin duda es ideal para documentos sencillos.

La alternativa recibe las siglas WYGIWYM:¹⁸ «lo que se obtiene es lo que se pretende». Bajo este paradigma, el usuario confecciona un documento de *texto normal* o sin formato, que se conoce como *código fuente* o *código de entrada*, en un editor de texto plano.¹⁹ Un editor de texto plano es un programa que nos permite editar el propio texto, no el formato. Normalmente no hay disposición del texto que considerar, no hay distintos tipos de letra, estilos, ni nada parecido. Sólo están el texto, las palabras, los caracteres, las frases. Una de las utilidades de estos editores especializados en texto sencillo es que se encuentran mejor preparados para documentos muy extensos.

Al utilizar el editor de texto plano para editar el texto fuente de una composición tipográfica, se hace utilizando la sintaxis de un cierto lenguaje formal, estricto. El documento fuente contiene el texto combinado con ciertas palabras clave, siguiendo esta sintaxis estricta, que son mensajes para que en una fase posterior, otro programa distinto, el formateador, realice ciertas tareas relacionadas con los tipos, los estilos y la disposición. Este otro programa realiza el formateado: *procesa* el texto fuente y entrega un resultado, la *salida*. Vemos en la figura 14 un esquema de este proceso.

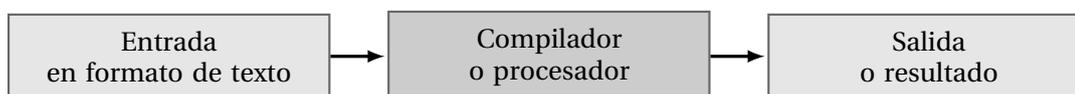


Figura 14: Función del compilador respecto del código fuente.

Una característica del procedimiento descrito es que el código fuente debe seguir la misma lógica interna que el documento que se pretende obtener.

¹⁸WYGIWYM son las siglas del inglés *What You Get Is What You Mean*.

¹⁹Del inglés *plain* “sencillo” o “normal”. Tenemos un ejemplo de editor de texto plano en el accesorio Bloc de notas.

LilyPond sigue este esquema, que es habitual bajo sistemas operativos tipo UNIX. Este mismo documento, tanto en lo que atañe al texto como a los gráficos de las ilustraciones propias, se ha elaborado íntegramente como textos escritos en varios lenguajes²⁰ que luego se han procesado para obtener el resultado que el lector tiene en sus manos.

La adaptación del esquema de bloques de la figura 14 al caso de LilyPond se muestra en la figura 15. Una vez preparado el código fuente, el programa LilyPond produce una composición tipográfica de la partitura, dando como resultado un documento en formato PDF, listo para imprimir.



Figura 15: Esquema de bloques que relaciona LilyPond con su entrada y su salida.

El espaciado y la distribución de las notas son completamente automáticos, aunque es posible modificarlos para que se adapten a nuestras necesidades. Para los detalles concretos acerca del uso del programa en la práctica, referimos al apartado 3.8.

La orientación de LilyPond, \LaTeX y otros formateadores de tipografía especializados, es obtener una salida de buena calidad estableciendo un compromiso con la facilidad, por parte del usuario, para redactar el texto (de esto se ocupa el editor del texto, que como hemos visto tiene sus propias ventajas cuando es un programa separado, especializado en la edición). Volviendo a la metáfora de la autocaravana, el uso de LilyPond significa cambiar esta combinación de casa y coche, por un coche y una casa diferenciados. Por esta razón, el primer contacto con LilyPond o \LaTeX cuando el usuario está acostumbrado a utilizar solamente editores de texto *visuales* como Microsoft Word (para texto) o Finale (para música), puede en ocasiones resultar frustrante.

Esta forma de trabajo está orientada a materializar una separación entre el contenido y el formato. Normalmente no nos damos cuenta del enorme esfuerzo que hacen los programas editores “convencionales” para tratar el problema de manejar al mismo tiempo el texto y su formato. En cambio, al tratar con texto sencillo y sin marcado oculto, el flujo de trabajo es más inmediato y la sensación de manejo siempre es la de una respuesta instantánea. Por contra, la obtención del resultado final se produce en diferido, a demanda del usuario, y no de forma continua como en un editor visual.

²⁰Los lenguajes utilizados para elaborar este documento han sido: Graphviz y TikZ para los gráficos en forma de grafo, Python y R para los gráficos en ejes coordenados y los cálculos estadísticos, LilyPond para los ejemplos musicales, Biblatex para la confección de la bibliografía, \LaTeX para la composición tipográfica y maquetación, y GNU Make para coordinar las dependencias.

3.7.1. Ejemplos de interfaces de texto

Los interfaces gráficos están basados en la manipulación directa del contenido (Chen y Zhang, 2007) y se han caracterizado desde los primeros tiempos de la informática interactiva como superiores por estudios como el de Davis y Bostrom (1992) o el de Rauterberg (1992). Sin embargo, una interfaz gráfica no significa automáticamente manipulación directa o facilidad de uso. A la *usabilidad* de una interface, tanto si es gráfica como basada en texto, pueden afectar muchos factores como el tipo de usuario, su experiencia y la naturaleza de la tarea. En la tabla 4 se muestran algunos ejemplos por tipo de tarea, de programas orientados a texto enfrentados a su contrapartida con interface gráfico.

Tabla 4: Correspondencias de interfaces gráficos y de texto en una selección de programas informáticos.

Tipo de tarea	Interactivo	Orientado a texto
Edición de texto	Word/LibreOffice	Tex/Latex
Edición de música	Finale/Sibelius/MuseScore	LilyPond
Modelado 3D	Blender	POVray
Diseño de fuentes tipográficas	Fontographer/FontForge	Metafont
Gráficos y cálculos estadísticos	SPSS	R, TikZ

Los interfaces de texto para aplicaciones de uso general como la edición de documentos, tipografía musical y otras, tienen para los ciegos y deficientes visuales un alto valor porque se pueden adaptar mediante lectores de pantalla y otros mecanismos de accesibilidad. Se mencionan algunos casos en el apartado de Aplicaciones.

3.7.2. Entornos de desarrollo

Todo lo que es posible hacer con LilyPond, se puede codificar en forma de texto con un editor sencillo como el accesorio de Windows «Bloc de Notas». Dado el énfasis en la flexibilidad y la potencia de la idea de expresarlo todo mediante un lenguaje formal, la pregunta que se hace el usuario es: ya que podemos hacer cualquier cosa que queramos, ¿qué queremos hacer, exactamente, y cómo se expresa? La respuesta está en la documentación del lenguaje (capítulo 4). Ahora bien, el concepto de interfaz de texto, visto superficialmente, sugiere una experiencia ardua, poco intuitiva y frustrante. Además, el usuario nuevo no tiene forma de experimentar por sí solo y aprender por ensayo y error. Se encuentra con un documento en blanco, en teoría lleno de posibilidades, pero sin ninguna pista sobre cuál debe ser el siguiente paso a tomar.

En todo caso, se ha visto que los interfaces de texto tienen un campo de aplicación en las principales tareas de producción de contenidos. La variedad de aplicaciones demanda, des-

de una variedad de frentes, una necesidad de facilitar la tarea al usuario (principiante o avanzado) y poner a su disposición una serie de herramientas de conveniencia, integradas en un solo programa, que configuran lo que se conoce como IDE²¹ o *entorno de desarrollo*. Pueden entenderse los entornos de desarrollo como programas amigables para una tarea, como la edición de código, que en principio se percibe como poco amigable. Los IDE ofrecen soluciones cómodas para el usuario que persiguen un ciclo de desarrollo más rápido y directo para los resultados finales pretendidos, que en el caso de LilyPond son partituras.

La idea de usar un entorno de desarrollo está enfocada a ocultar el compilador de la vista, de manera semejante a cómo el motor de un vehículo está oculto bajo el capó. Al conducir, el piloto maneja las funciones del vehículo a través del volante, los pedales y la palanca de cambio, interactuando con el vehículo a través de estos controles, no directamente. Cuando el compilador está en una capa oculta, sobre la que se superpone otra capa conocida como interfaz de usuario, se convierte en un motor que está a las órdenes de los controles de dicha interfaz. Ambas zonas, la del interfaz y la del motor, están interconectadas según se aprecia en la figura 16 que amplía a la figura 15 superponiendo la capa del IDE. El ejemplo mostrado corresponde al editor Frescobaldi.

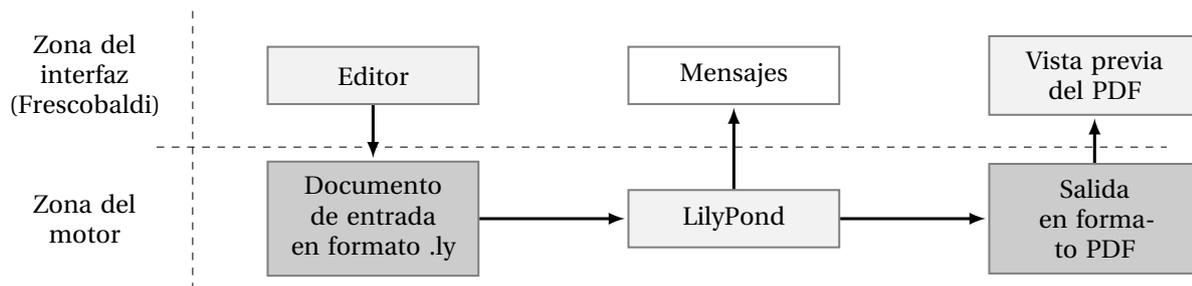


Figura 16: Superposición de la capa del interfaz de usuario a la capa del motor, en el caso de LilyPond, y relación entre las funciones principales de ambos.

La característica de “amigable” se basa en la siguiente lista de funciones que un IDE puede tener y que el motor o el compilador no proporcionan por sí solos:

- Coloreado de la sintaxis. Cada elemento de código toma un color dependiendo de su función. Esto hace menos trabajoso para la vista localizar la información relevante en cada momento.
- Funciones genéricas del editor. A las funciones de copiar y pegar se le añaden habitualmente el plegado y desplegado de bloques, ofreciendo una vista más compacta del código.
- Autocompletado de palabras clave (“texto predictivo”). El editor se anticipa a las construcciones más frecuentes que el lenguaje en particular suele demandar.
- Vista previa del resultado. Si el resultado es gráfico, puede mostrarse en una ventana

²¹IDE son las siglas en inglés de *Integrated developing environment*.

que admita niveles de ampliación y funciones de navegación por el documento.

- Interacción mutua entre el código y la vista previa. Si una pieza de código corresponde a un elemento gráfico, es posible resaltar éste para llevar el cursor al lugar adecuado, y a la inversa, si se mueve el cursor es posible destacar gráficamente el elemento correspondiente en la salida gráfica.
- Presentación de los mensajes de salida. El motor emite un registro de los acontecimientos, que en caso de error ayuda a localizar su causa, y en caso de éxito debe presentar un mensaje tranquilizador.
- Acceso a la documentación. El entorno puede incluir funciones de búsqueda que abarcan toda la documentación del lenguaje.
- Ayuda sensible al contexto. Por ejemplo: utilización de las palabras clave seleccionadas del documento como términos de búsqueda dentro de la documentación.

Muchas de las funciones de un IDE están ligadas a la naturaleza de un determinado lenguaje. Si bien es posible que un entorno tenga la necesaria flexibilidad para poder adaptarse a diversos lenguajes, la especialización añade potencia al conjunto porque las dos capas, la del interfaz y la del motor, están unidas más fuertemente en beneficio del usuario. En el caso de LilyPond, la orientación a la música determina que los entornos faciliten, por ejemplo, el manejo de la altura de las notas o la creación de plantillas para diversos tipos de partitura. Obviamente, cuanto más especializada es esta relación entre el compilador y la interfaz de usuario, más vulnerable es a los cambios en el motor, como cuando la sintaxis de los documentos se modifica o cuando aparecen funcionalidades nuevas para las que el entorno no está preparado.

3.7.3. LilyPondTool

La herramienta LilyPondTool (Fodor, 2010), de Bertalan Fodor, es un añadido para el editor multiplataforma jEdit (Anson, Ezust, Casanova, Hoyt y Dillon, 2013), que funciona bajo Java en Windows, MacOS y Linux. Se encuentra bajo la licencia GPL.²²

LilyPondTool era, hasta la aparición de Frescobaldi, el único entorno importante para LilyPond, con una serie de ventajas sobre un editor simple, entre las que se encuentran las más frecuentes que ya se han mencionado: el resaltado de la sintaxis mediante coloreado del código, asistentes para la preparación de plantillas, colocación automática del cursor en el código al pulsar sobre el documento PDF resultante, resaltado inverso del PDF al pulsar sobre el código o el auto-completado de las palabras clave.

Con la renuncia de B. Fodor a continuar su mantenimiento para dedicarse al desarrollo de MuseScore, el proyecto jEdit ha quedado huérfano y su desarrollo está completamente estancado a fecha de 2015.

²²Una breve referencia a esta licencia puede verse en la página 30.

3.7.4. Frescobaldi

El organista y programador Wilbert Berendsen había elaborado en 2008 un complemento de LilyPond para el editor Kate, llamado LilyKDE (Berendsen, 2008b) que funcionaba dentro del entorno de escritorio KDE. Desde entonces se ha traducido al holandés, francés, turco, español, ruso e italiano. Más tarde, en diciembre de 2008, este proyecto fue reemplazado por dos alternativas del mismo autor: LilyPondKDE4 (Berendsen, 2008c), que quedó en un estado de desarrollo muy temprano, y Frescobaldi (Berendsen, 2008a), para Windows, MacOs y Linux, ya no como un complemento de un editor existente, sino con entidad propia como programa independiente. Los tres proyectos son software libre; su licencia es la GPL versión 2 de GNU. Podemos observar en la figura 17 una captura de pantalla de la versión española²³ del editor Frescobaldi.

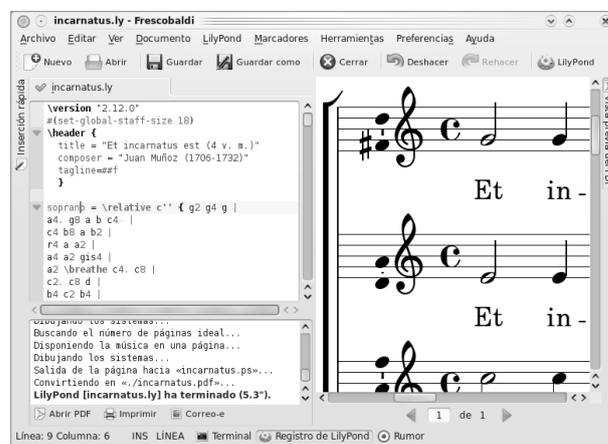


Figura 17: El entorno Frescobaldi, de Wilbert Berendsen, en una captura de pantalla en proceso de edición de un documento de LilyPond, a la izquierda. La vista previa del PDF se muestra en el panel de la derecha.

La orientación del programa Frescobaldi es la de ser ligero y fácil de utilizar. Entre sus posibilidades debemos mencionar: Introducción de partituras de LilyPond y vista previa con un solo clic del ratón; pulsación sobre notas del PDF y colocación automática del cursor sobre la fuente; asistente de partitura para preparar un trabajo en pocos pasos; herramientas de edición para manipular el ritmo, enguionar la letra, añadir articulaciones y otros símbolos a la música ya introducida, actualización de los documentos a las versiones recientes de LilyPond y traducir nombres de notas; autocompletado de instrucciones de LilyPond, sensible al contexto; gestor de expansiones para introducir fragmentos grandes mediante mnemónicos; y un complemento para introducir notas paso a paso con un teclado MIDI. Está traducido a 13 idiomas hasta la fecha, verano de 2015. Recibe su nombre como homenaje por parte de W. Berendsen hacia el compositor y organista italiano del período barroco Girolamo Frescobaldi (1583-1643).

²³En traducción del autor.

3.7.5. Otros entornos de escritorio

Desde que entra en escena el entorno Frescobaldi, las soluciones alternativas tienen menos interés porque son solamente adaptaciones de otros editores en lugar de orientarse específicamente al trabajo con LilyPond. Las citamos en con el propósito de presentar una panorámica completa.

El editor conocido como Notepad++ para Windows (Ho, 2007), de Donald Ho, libre y con licencia GPL, admite la programación de las tareas más comunes en una sola pulsación de tecla, permite el plegado y desplegado de bloques de código, resaltado parcial de la sintaxis mediante colores, y algunas otras ventajas. Ya se encontraba completamente sobrepasado en funcionalidad por Frescobaldi desde la aparición de éste.

Dentro de una de las distribuciones de software libre más populares en el mundo GNU/Linux, la distribución Ubuntu²⁴ ofrece el entorno GNOME (un escritorio con un conjunto de aplicaciones que funcionan coordinadamente), y con él, el mencionado editor de texto Gedit, que disfruta como los anteriores de una licencia GPL. En este editor es posible programar una tecla para que realice el procesado de la fuente por parte de LilyPond y la visualización inmediata del PDF.

3.7.6. Entornos online (SAAS)

Si se observa la figura 16 (pág. 76), es posible imaginar que la frontera que separa a la zona del interfaz de la zona del motor, es una conexión a Internet. En este caso, el motor es un *servicio web* o SAAS (software como servicio),²⁵ se ejecuta en un servidor, y el usuario no lo tiene instalado en su sistema, sino que lo maneja a través del navegador web.

El editor es un panel ordinario de texto. Los documentos individuales están asociados a la sesión web y a un enlace único que se puede compartir. La persistencia de los documentos se consigue mediante almacenamiento “en la nube” con servicios como Dropbox. El motor puede funcionar en una máquina arbitrariamente potente, presumiblemente mucho más potente que el sistema propio. La salida en PDF se puede visualizar y descargar.

Dependiendo de algunos factores condicionantes, la experiencia puede ser similar a la de un IDE instalado en el escritorio. En el lado positivo está la disponibilidad del programa sin instalarlo, lo que puede ser útil para una demostración rápida. Hace posible algo bastante demandado, que es el manejo desde un dispositivo móvil como una *tablet* siempre que tenga un teclado mecánico y conexión de datos. En el lado negativo, hasta que las funciones de con-

²⁴<http://www.ubuntu.com>. Véase el glosario del apéndice G.

²⁵SAAS son las siglas en inglés de *Software as a service*.

veniencia aportadas por el editor no igualen a las que ofrece una aplicación de escritorio, la situación es semejante a usar el bloc de notas, una experiencia espartana y con poco atractivo para usuarios nuevos.

Se encuentran en funcionamiento varias instancias de LilyPond funcionando como SAAS, siempre por parte de voluntarios sin ánimo de lucro, con o sin registro. La tabla 5 muestra una selección.

Tabla 5: Casos de LilyPond como servicio online.

Nombre	URL
LilyBin	lilybin.com
Tunefl	tunefl.com
WebLily	weblily.net
OMET	omet.ca

Poner la funcionalidad del programa accesible a cualquier usuario de la web es una idea atractiva, pero conlleva un peligro técnico para el proveedor. El lenguaje permite, dentro de su sintaxis, ciertas construcciones que son capaces de bloquear al motor con bucles infinitos o con instrucciones que acceden directamente al sistema. La barrera para estas partituras “malintencionadas” es un modo especial de funcionamiento denominado *jaula*, que restringe drásticamente el abanico de funciones potencialmente dañinas, como las macroinstrucciones en Scheme.²⁶ Se entiende que un usuario no va a producir daños a propósito en su propio sistema, por lo que el modo jaula no tiene sentido en una aplicación de escritorio.

3.8. LilyPond en la práctica

Después de cualquier disertación teórica sobre el software libre, cabe cuestionarse si existe un aspecto realmente práctico de esta tecnología y si efectivamente los docentes o estudiantes podrían usarlo para la producción de sus pequeñas partituras de cada día, o si los profesionales podrían vencer la reticencia inicial y llegar a emplear software libre para proyectos serios del mundo comercial. De la calidad de los resultados nos ocupamos en el segundo estudio de este trabajo.

Varios aspectos han de tenerse en cuenta desde el punto de vista puramente práctico.

- Disponibilidad, facilidad o dificultad de instalación.
- Actualizaciones.
- Estabilidad a largo plazo.
- Dificultad del aprendizaje (sencillo y avanzado).

²⁶El lenguaje Scheme (Abelson y col., 1998) se utiliza como mecanismo de expansión para LilyPond.

- Manejo.²⁷
- Coste total de posesión.
- Su orientación a documentos de texto.

Las dificultades que surgen inevitablemente durante el ciclo de utilización de un programa no comercial, han de ser aliviadas necesariamente por medio del apoyo que la comunidad presta a sus miembros,²⁸ ya que los productos de este tipo no incluyen ninguna contrapartida de soporte.

Los programas de software libre suelen tener una alta disponibilidad en forma de descargas gratuitas. Éstas tienen un coste de alojamiento que es relativamente pequeño; dejaremos los detalles para el cuerpo de nuestro primer estudio.

LilyPond es, como ya se ha dicho, un programa perteneciente al tipo de los denominados “compiladores” y es, por tanto, una aplicación de consola, que se utiliza en un terminal de texto. El nombre de la instrucción es `lilypond` y va seguido del nombre del archivo que se desea compilar, pulsando finalmente la tecla <Enter>. Por ejemplo:

```
$ lilypond partitura.ly
```

donde «\$» es el símbolo del sistema de la consola, y `partitura.ly` es el nombre de archivo del documento. Como resultado, se obtiene la partitura en formato PDF.

Así como hace algunos años, antes de la utilización generalizada de los entornos de desarrollo, habría sido necesario dedicar una considerable cantidad de espacio a describir el uso de LilyPond desde la consola, actualmente el enfoque más apropiado es la idea de que la instrucción `lilypond` hace las veces de *motor* de tipografía. Este motor dota al entorno Frescobaldi de la capacidad de producir la notación musical. Por tanto, LilyPond permanece *invisible* ya que el usuario no trata directamente con él, sino con la función de Frescobaldi (o del entorno editor elegido) de invocar al motor.

3.8.1. Descarga e instalación

En comparación con nuestra primera incursión en el estudio de LilyPond en 2009, algunos aspectos de la instalación, así como el panorama de los entornos de desarrollo, se han simpli-

²⁷En el caso de los lenguajes de programación como LilyPond, el concepto de “manejo” es un poco distinto del de los programas más usuales que requieren conocer combinaciones de teclas, distribución de funciones por los menús y ciertas operaciones del ratón. Aquí el “manejo” consiste en saber qué construcción sintáctica resuelve un determinado problema y dónde encontrar dicha construcción en la documentación, en caso necesario. Sería más exacto hablar de *manejo de los manuales* que de *manejo del programa*, ya que la orden de compilar el código fuente es siempre la misma y se dispara con la pulsación de una sola tecla.

²⁸La “comunidad” está formada por todos los usuarios y desarrolladores que alguna vez tuvieron cualquier actividad relacionada directa o indirectamente con el proyecto, aunque solo fuese descargar e instalar el programa.

ficado notablemente. La instalación del conjunto de programas recomendado para trabajar con LilyPond es sencilla y se reduce a la descarga e instalación de LilyPond y Frescobaldi, ya sea para Windows, MacOS o GNU/Linux, aceptando todos los acuerdos de licencia. Tratándose de software libre, debemos decir que la diferencia con la mayoría de los textos que uno tiene que leer antes de pulsar el botón “Siguiente” durante el proceso de instalación, es que en este caso el usuario sí puede decir que acepta todas las condiciones, y después cumplirlas. LilyPond y Frescobaldi están disponibles para su descarga gratuita en las correspondientes páginas oficiales del proyecto.²⁹ Después de ejecutar los respectivos instaladores, las dos aplicaciones están listas para su uso.

Las versiones para Windows se publican por parte del equipo de desarrollo *al mismo tiempo* que las destinadas a otros sistemas operativos: GNU/Linux, MacOS X y FreeBSD, y poseen idénticas funcionalidades.

3.8.2. Creación de una partitura sencilla

Al abrir la aplicación Frescobaldi, es posible de forma inmediata producir un primer ejemplo. Se propone el texto de la figura 18; las llaves se consiguen con AltGr pulsando una tecla que en los teclados de PC suele estar junto a la ‘Ñ’. Los apóstrofos se consiguen mediante la tecla que está justo a la derecha del número 0. Este texto es el *código de entrada* como se ha visto en el apartado dedicado a los interfaces de texto.

```
{ c' d' e' f' g'2 e' }
```

Figura 18: Código de entrada propuesto para un ejemplo

3.8.3. Procesado del documento

Según los esquemas de bloques presentados anteriormente y las descripciones de LilyPond como motor de tipografía, ya sabemos que el programa no se utiliza para editar la partitura, sino para producir una salida en formato PDF a partir del documento de texto que hemos preparado. Esto se denomina *procesar el código de entrada*. Para procesar el código de entrada, la combinación de teclas predeterminada de Frescobaldi es Ctrl+M. LilyPond hará su trabajo: interpretar el código de entrada y producir una salida.

El procesado de documentos muy sencillos tarda un segundo o menos. El resultado es un archivo PDF que aparece en el panel de vista previa. En este caso, su aspecto es que aparece en la figura 19.

²⁹Las páginas oficiales de LilyPond y de Frescobaldi son lilypond.org y frescobaldi.org, respectivamente.



Figura 19: Resultado correspondiente al código de ejemplo de la figura 18

Como se observa en este ejemplo, en general aparecen automáticamente la clave de Sol y la indicación de compás además de las notas y las líneas divisorias.

3.9. Aspectos del resultado visual

3.9.1. La metáfora de la máquina de pan

La “máquina de pan” es un ejemplo ideado por Valentin Villenave para representar la función en la sociedad de programas como LilyPond que tratan de acercar productos de alta calidad, manufacturados casi artesanalmente, a la realidad de las actividades cotidianas. Cuando uno hace pan en su casa, con una pequeña máquina automática, puede tener en su cocina todas las mañanas auténtico pan caliente recién hecho. Al trasladar el ejemplo a la tipografía musical, con programas muy especializados como el que nos ocupa podemos emular la actividad profesional y artística de los grabadores de música tradicionales, o al menos éste es su propósito.

Presentaremos un ejemplo de los aspectos que LilyPond pretende emular respecto de las partituras grabadas profesionalmente.

3.9.2. Líneas adicionales

Un problema clásico de la tipografía musical es el espaciado horizontal, que con frecuencia hay que comprimir o expandir en función del contenido musical de las líneas y las páginas, y del número de éstas que dan una distribución de páginas aceptable. Si se tiene en cuenta que la notación musical tiene dos dimensiones, puesto que debe representar al mismo tiempo una secuencia lineal horizontal (temporal) y una simultaneidad vertical entre voces y partes armónicas, se observará que el problema de la distribución de las páginas está sujeto a fuertes restricciones. Son muy numerosas las variables cuyos valores se deben conservar a través de los compases, las líneas y las páginas, y al tiempo que estas variables se tratan de conservar para que la partitura tenga un aspecto uniforme, las restricciones mencionadas prohíben situaciones que tienen su paralelismo en la división de palabras al final de una línea de texto normal. Esto se puede conseguir considerando los elementos gráficos como unidos por una

Si accedemos a algunas ediciones comerciales y comparamos este aspecto de manera relativamente informal, podemos comprobar que el recurso está reflejado en el resultado de otros programas, aunque no todos. Entre nuestra selección, que está sesgada de manera que incluya una edición considerada subjetivamente “barata”, vemos en cambio otras consideradas como de buena calidad en las que se aplica este método. La comparativa está en la figura 21. Las fuentes son:

1. Edition Peters: Brahms, concierto para piano y orquesta núm. 2, Op.83. Primer movimiento, compás 13, parte del piano, mano derecha (Brahms, 1927).
2. Schott/Universal, Wiener Urtext Edition: Chopin, 24 preludios Op.28, preludeo núm. 19, compás 49, mano derecha (Schott, 1973).
3. Könemann Music Budapest: Grieg, piezas líricas para piano, número 43, Op.62 núm. 2, compás 59 de la mano derecha (Könemann, 1994).

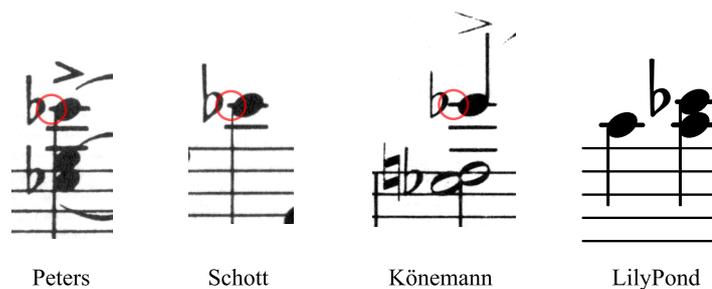


Figura 21: Comparativa de los ajustes de las líneas adicionales en presencia de una alteración, en tres publicaciones comerciales. Dos de ellas se consideran de buena calidad: Brahms (1927) y Schott (1973). La tercera es una edición barata: Könemann (1994). Presentamos de nuevo a LilyPond en cuarto lugar.

Las muestras proceden de ediciones en papel escaneadas a 600 ppp; los tratamientos dados a las imágenes han sido la ecualización de los niveles de gris y la escala en función de los espacios del pentagrama.

En el caso de Könemann (1994) no se hace este ajuste, e incluso se produce una colisión. Es posible que exista una asociación entre la presencia de estos detalles tipográficos en una edición, y la consideración subjetiva de su calidad visual como la que se investiga en el estudio 2.

3.10. MuseScore

El proyecto MuseScore debe tener aquí un lugar porque es, junto a LilyPond, uno de las principales herramientas para la producción de partituras musicales con software libre. Está licenciado bajo la GPL.

MuseScore,³¹ de Werner Schweer (Schweer y col., 2012), se creó como una ramificación del código del programa secuenciador MusE para Linux. En principio, MusE incluía algunas capacidades de notación musical, pero Werner decidió eliminar todas las funcionalidades para imprimir partituras y produjo este programa independiente dedicado solamente a dicha tarea, en el año 2002. A partir de 2004 se unieron al autor principal los desarrolladores Thomas Bonte y Nicolas Froment.

MuseScore es un editor visual de descarga libre y gratuita, con una fuerte inclinación a la facilidad de uso, existe para Windows, Linux³² y Mac, y se encuentra traducido a 48 idiomas. El proyecto se financia principalmente a través de las suscripciones a su plataforma social de publicación de partituras.

Posiblemente la funcionalidad más interesante de este programa sea la facilidad con que se crean “videopartituras” o secuencias animadas en que la música y la partitura están sincronizadas, lo que resulta muy atractivo para su aplicación en la enseñanza por la fuerte asociación que se produce entre la vista y el oído. El empleo en educación aprovechando esta y otras funcionalidades está tratado en los artículos de Shinn (2013), Galera (2013) y Núñez (2009), entre otros. Mar Galera, del Departamento de Didáctica de la Expresión Musical y Plástica de la Universidad de Sevilla, es con 26 publicaciones relacionadas con el uso de la tecnología en la enseñanza musical, una de las investigadoras más activas en esta especialidad y con esta herramienta.

El código fuente de MuseScore está registrado en un sistema de control de versiones Git, pero el número de desarrolladores diferentes en su corta historia es insuficiente para poder elaborar cualquier tipo de pronóstico; sin embargo, todo indica que su desarrollo se mantiene de forma activa y las versiones son estables.

Por otro lado, este proyecto está relacionado con dos grandes acontecimientos en el mundo de la cultura abierta: la preparación de las partituras completas de las Variaciones Goldberg y del primer libro de “El clave bien temperado”, ambos de J.S. Bach, coincidiendo con las respectivas interpretaciones y grabaciones discográficas por parte de la pianista Kimiko Ishizaka. La primera iniciativa, denominada “Open Goldberg Variations”, se hizo pública el 28 de mayo de 2012 y contó con la revisión colectiva de un grupo de usuarios de MuseScore. El segundo proyecto, bajo el nombre “Open Well-Tempered Clavier”, contó para su financiación con un total de 44.083 dólares americanos aportados por 904 patrocinadores diferentes, en otra iniciativa de provisión colectiva de fondos lanzada el 20 de septiembre de 2013.

Aparte de ser dos colecciones de gran formato y obras de arte fundamentales de la historia de la música clásica, ambos casos de producciones depositadas explícitamente bajo licencias

³¹El sitio oficial del MuseScore es musescore.org.

³²La razón por la que este programa se desarrolla para Linux es que se inició como una ramificación del mencionado secuenciador MusE, que solamente contempla este sistema operativo.

libres han dado como resultado dos tipos diferentes de material de alto valor para la educación y la investigación. Por un lado, unas partituras de buena calidad visual, muy revisadas, exentas de derechos, en un formato abierto y aptas para su modificación y adaptación por parte del público, con software libre. Por otro lado, unas buenas interpretaciones de la mano de una prometedora artista y unos registros sonoros técnicamente impecables, igualmente exentos de restricciones legales para su libre utilización.

4. Introducción al lenguaje LilyPond

LilyPond es un lenguaje para codificar partituras fácil de aprender¹ en general, aunque presenta un tipo de flexibilidad que para ciertas tareas puede ofrecer una pendiente algo más pronunciada en la curva de aprendizaje; sin embargo, la recompensa que supone la obtención de resultados de buena calidad activa un mecanismo de autoreforzamiento que fomenta la dedicación de más tiempo a la tarea de su estudio. La experiencia con usuarios que se acercan por primera vez a este sistema pone de relieve la importancia de que tomen conciencia cuanto antes de que toda la interacción del usuario con el programa se hace a través de este lenguaje. La codificación de la música, la letra, las expresiones gráficas y textuales y el resto de las decisiones de tipografía, se hacen única y exclusivamente a través de modificaciones en el texto fuente del documento.

Este paradigma de la tipografía musical, basado en documentos de texto normal, tiene una gran flexibilidad y también ciertos aspectos incómodos. Desde los comienzos del proceso de diseño del lenguaje y durante su evolución posterior, el objetivo principal, que había que armonizar con la posibilidad de escribirlo y leerlo cómodamente por seres humanos, era que fuese capaz de expresar la misma complejidad que presenta la música que se ha de tipografiar, lo que (según el contenido de notación musical de que se trate) implica que las construcciones pueden llegar a ser bastante intrincadas. Una premisa utilizada por todos los desarrolladores ha sido siempre la de que los contenidos musicales sencillos y habituales habían de requerir construcciones del lenguaje que fuesen sencillas también. En los apartados siguientes se ofrecen ejemplos de ello y también muestras del caso opuesto, es decir, código más enrevesado que responde a situaciones más exigentes.²

¹Las fuentes principales de información respecto a la sintaxis para la notación musical con LilyPond son la documentación completa del programa (Nienhuys y col., 2015a, 2015b, 2015c), y el grupo de usuarios «lilypond-es» o lista de correo en español para usuarios, abierto al efecto el 24 de agosto de 2007 y después transferido a los grupos oficiales de GNU como lists.gnu.org/mailman/listinfo/lilypond-es, que contaba en noviembre de 2009 con 61 miembros y en septiembre de 2014 con 127.

²En todo caso, el aprendizaje a través de ejemplos de código, exclusivo del enfoque orientado a texto, fomenta el progreso autónomo del aprendiz. Los usuarios valoran que los ejemplos puedan copiarse como tal texto e insertarse en los propios documentos para conseguir lo mismo en distintas máquinas. A su vez, el interfaz orientado a texto allana de forma natural el camino del *e-learning* porque la interacción con el programa se compone de acciones fundamentalmente simples: editar código.

4.1. Elementos de notación más frecuentes

En este condensado resumen, todos los ejemplos se han obtenido directamente del código fuente que aparece sobre ellos. Debe advertirse que en la mayoría de los ejemplos se utiliza el modo relativo implícitamente, tal y como se explica en el apartado siguiente. Además, una partitura completa requiere llaves { } encerrando el código, algo que hemos evitado por simplicidad a partir del tercer ejemplo: no deben olvidarse al probar estas muestras por uno mismo.

4.1.1. Altura y duración de las notas

La altura de las notas se expresa de forma absoluta con las letras minúsculas `c d e f g a b` para referirse a Do, Re, Mi, Fa, Sol, La y Si, seguidas de apóstrofos o comas. Cada apóstrofo sube una octava, y cada coma baja una octava. El Do central es `c'`

```
{ c' d' e' f' g' a' b' c'' c' b a g f e d c g, }
```



La altura también se puede expresar de modo relativo, indicando una nota de partida y suponiendo que el intervalo máximo es una cuarta; en caso contrario se añade una coma o un apóstrofo para bajar o elevar una octava.

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



Independientemente de la armadura, cada nota lleva su altura exacta, con su alteración. El sostenido (`#`) se expresa añadiendo «`is`» y el bemol (`b`) añadiendo «`es`»; las alteraciones forzadas con una admiración «`!`» y las de cortesía con una interrogación «`?`»:

c b c cis d e d des c d! c d?



Las duraciones se expresan añadiendo a cada nota uno de los números 1 2 4 8 16 32... que significan: redonda, blanca, negra, corchea, semicorchea, fusa... Cada valor se mantiene para las notas siguientes, hasta nuevo aviso.

d2 e4 c d4 e8 f e4 c



El punto ortográfico después del número que indica la duración, produce una figura con puntillo.

c2. d4 e4. f8 g8. a16 b8. a16 g1



4.1.2. Silencios y ligaduras

Los silencios se expresan mediante la letra r minúscula; los compases en silencio con R mayúscula; las ligaduras de unión mediante una tilde curva, y las de expresión se trazan desde la nota marcada con paréntesis de apertura «(» hasta la nota marcada con paréntesis de cierre «)».

R1 c4 r e r g2 r f2~ f8 e(d c) d1



4.1.3. Clave, tonalidad, compás

La instrucción `\clef` cambia la clave, `\key` cambia la tonalidad y `\time` sirve para indicar el compás. Vemos en el ejemplo cómo el puntillo se codifica mediante un punto ortográfico.

```
\key f \minor
\time 2/4 \clef bass e2 f
\time 6/8 \clef alto g8 f es d4.
```



4.1.4. Polifonía

Los archivos fuente de LilyPond deben contener una única expresión musical, ya sea una corta melodía o un tiempo completo de sinfonía. Una expresión musical es un par de llaves que encierra notas u otras expresiones.

Las expresiones en secuencia se encierran entre llaves, las de música simultánea se encierran entre ángulos dobles `<< >>`, y las voces se separan mediante barras invertidas `\\`:

```
<< { e4 f g2 e4 f g2 g8 a g f e4 c4}
  \\
  { c,4 d e c c d e c e f g2 } >>
```



4.1.5. Música vocal

Mediante la construcción `{ Música } \addlyrics { Letra }` es sencillo elaborar canciones simples. Las sílabas se separan mediante dobles guiones y las sinalefas se expresan mediante una tilde curva.

```
{ g8 e g e g4. e8 d2 c }
\addlyrics { Es -- ta -- ba-el se -- ñor Don Ga -- to }
```



Añadiendo un guión bajo por cada sílaba adicional, se pueden tipografiar pasajes melismáticos:

```
{ g4 a g2 g4 a b a g a a2 g1 }
\addlyrics { Ky -- ri -- e, e -- _ _ _ _ _ lei -- son. }
```



Otra forma de codificar un melisma es mediante ligaduras de expresión:

```
{ g4 a g2 g4( a b a g a) a2 g1 }
\addlyrics { Ky -- ri -- e, e -- lei -- son. }
```



Las líneas de extensión al final de una palabra,³ se tipografían escribiendo dos guiones bajos.

```
{ r4 d( e d) c1( d) }
\addlyrics { A -- men. _ _ }
```



³Según Gerou y Lusk (2004), la línea de extensión debe limitarse a la última sílaba.

4.1.6. Articulaciones y matices

Es fácil añadir matices y articulaciones a una nota siguiendo los ejemplos que aparecen a continuación:

```
c16\< d e f g f e d e8\f a-. a4->
```



4.1.7. Acordes, anacrusas, tresillos, mordentes, calderón, doble barra

El ejemplo siguiente contiene un compás de anacrusa (con un valor de negra) y un acorde al final con calderón. Acaba en barra final:

```
\time 3/4 \partial 4  
f8. f16 e4 c d <c g e>2.\fermata \bar "|."
```



El ejemplo siguiente ilustra los tresillos y las notas de adorno:

```
e4 \acciaccatura g8 \times 2/3 { f8 e d } e4 r
```



4.1.8. Repeticiones

Las repeticiones no se indican gráficamente, sino que se expresan de forma lógica mediante una sintaxis que permite introducir fácilmente finales alternativos (primera y segunda vez):

```
\repeat volta 2 { e2 f4 g g f e d c c d e }
  \alternative { { e2 d } { d2 c } }
```



4.1.9. Variables y transposición

El concepto de *variable* es fundamental para la creación de grandes proyectos. La asignación de un nombre arbitrario a un fragmento de música, permite la reutilización posterior del código, invocándolo a través de este nombre:

```
motivo = { c'16 g' e'' c'' }
{ \motivo \motivo \motivo \motivo }
```



La reutilización del código abre la puerta a funcionalidades interesantes como extraer las partituras y crear la partitura del director a partir de la misma fuente, esto es, escrita una sola vez.

```
clarinete = { \key c \major c'2 d'4 e' f'2 e' }
{ <<
  \clarinete
  \transpose c d \clarinete
  >> }
```



5. Aplicaciones

LilyPond se aplica en situaciones reales, de forma profesional, como muestran los apartados que siguen, donde se examinan las aplicaciones del programa que mayor renombre han tenido y se han difundido en el entorno de su desarrollo. Utilizamos la observación de esta acumulación de usos profesionales en el mundo real como argumento alrededor de la idea de que el programa constituye un posible mecanismo válido para la producción de partituras musicales para la transcripción del patrimonio, la educación y el mundo profesional al mismo nivel que otros productos de software de tipografía musical en cuanto a calidad, facilidad de uso y robustez.

En la enumeración de las aplicaciones distinguimos dos grandes apartados. Las primeras secciones corresponden a hechos pasados que constituyen *hitos* en la utilización de este software libre. Entre ellos se encuentran cancioneros, ediciones digitales de obras existentes, colecciones publicadas en papel o en línea, o arreglos estrenados.

La fuente de documentación para todos los casos, excepto allí donde se especifique lo contrario, es información de primera mano que se puso inicialmente a disposición de los desarrolladores a través de los canales internos oficiales del proyecto.

El último apartado ofrece ejemplos de aplicaciones potenciales para las que el programa ya ha demostrado ser válido a la vista de las aplicaciones que se han enumerado anteriormente. Se sugieren campos de acción en los que el enfoque robusto de la orientación a documentos de texto y su idoneidad para el control de versiones lo hacen especialmente adecuado.

5.1. El Liedboek

El libro holandés *Liedboek* (ISK, 2013), en español “cancionero”, es una gran recopilación de música litúrgica protestante compuesta por más de mil canciones. Tiene 1620 páginas y fue elaborado completamente con LilyPond. El equipo editorial está formado por 70 personas en 8 grupos de trabajo y la edición musical corre a cargo de un conjunto de 19 personas coordinadas por Wilbert Berendsen, el mismo que desarrolla y mantiene la aplicación Frescobaldi (Berendsen, 2008a).

Cruzando la lista de los editores musicales que aparece en los agradecimientos del prefacio

con la lista de desarrolladores de LilyPond, se obtiene que al menos dos personas están en los dos equipos.

El contenido quedó terminado en 2012 y es una nueva versión aumentada de un cancionero anterior (ISK, 1973) que tenía 491 canciones. Se emplea para los textos un tipo de letra grande (Arial de 18pt), pensado para personas con dificultad para la lectura. El proyecto recibió el apoyo financiero de BV Songbook y de la CBB, una biblioteca para ciegos y deficientes visuales. Contiene piezas cantadas tradicionales y de nueva factura, entre ellas salmos, himnos, piezas de carácter familiar, canciones de otros países (se entiende: distintos de Holanda) y canciones para niños.

Hasta julio de 2013 se habían realizado tres ediciones con un total de 140.000 copias vendidas, y existe también como libro electrónico en línea por el que se puede navegar como por una base de datos, bajo suscripción.

5.2. La integral de Oskar Fried

La edición completa de las canciones para una y dos voces de Oskar Fried (Fried, 2013), con acompañamiento de piano, se realizó durante los años 2011 y 2012 por parte del musicólogo Alexander Gurdon y el pianista Urs Liska, y fue elaborada sobre LilyPond por Urs Liska y Janek Warchoń. Oskar Fried (1871-1941) gozó de fama en su época principalmente como director de orquesta, con algunas versiones de referencia históricas de sinfonías de Mahler. El libro de los *lieder* y dúos completos tiene 128 páginas y resultó un proyecto extremadamente complejo según los autores de la tipografía.

La nota que acompaña a la edición comenta lo siguiente:

Una característica especial de la edición es el empleo de “herramientas” informáticas para el trabajo musicológico que permiten nuevos procesos de trabajo integrados y ofrecen un gran potencial en términos de eficiencia y control de calidad. La edición ha sido totalmente procesada con software de código abierto sobre archivos de texto.

donde por “código abierto” debe entenderse “software libre” en el caso de LilyPond.¹

Este trabajo recibió el premio Deutscher Musikeditonspreis «Mejor Edición 2014» de la asociación de editores de música alemanes, la Deutsche Musikverleger-Verband (DMV).

¹Aunque tratemos el asunto con más detalle en otro apartado, es conveniente indicar aquí la diferencia entre «código abierto» y «software libre». Por un lado, son dos movimientos o iniciativas distintas que tienen en común la disponibilidad del código fuente como requisito para la libertad del usuario a la hora de compilar y modificar el software. Por otro lado, la expresión «código abierto» o «fuentes abiertas» se emplea con cierta frecuencia a modo de eufemismo que a algunas personas con poder de decisión les suena más aséptico y menos comprometido que la expresión «software libre».

Las palabras del jurado² fueron las siguientes:

Toda edición con mérito propio busca y desea una amplia audiencia. Oskar Fried fue aclamado mundialmente (y aún hoy lo es por parte de muchos profesionales) como director de sinfonías de Mahler y de Bruckner, y fue también un compositor de talento. Sin embargo, su faceta como compositor es prácticamente desconocida, y merece estar (de nuevo) puesta a disposición de un público más amplio. Esta edición es, por ello, digna de alabanza en sí misma. Al mismo tiempo la presentación del conjunto es extraordinaria: la encuadernación es hermosa, la impresión musical es muy buena y atractiva para su lectura, y contiene también una serie de imágenes entre las que se encuentran las portadas de las ediciones originales. Es encomiable también el informe crítico que aparece en el apéndice de este volumen, donde se incluyen comentarios sobre las fuentes y unas notas sobre la escucha.

La figura 22 muestra un pequeño fragmento de las notas críticas.

<p>Liebeständelei op. 1, Nr. 3</p> <p>[13, 49] 11. ♩-Triole r. H. — ♯ vor gis' fehlt.</p> <p>[18] 1. ♩. l. H. — Verl.g.pkt. zum D fehlt.</p> <p>[24] 3. – 4. ♩ Sgst. — Der für das Melisma erforderliche Bg. fehlt. Fried hätte ihn vermutlich bis zum b' weitergeführt (vgl. etwa T. 48). Hier wurde jedoch orthographisch korrekt ergänzt.</p> <p>[27] 1. ♩. l. H. — ♯ vor C fehlt.</p> <p>[35] 1. ♩. l. H. — Quinte [G/d] als \circ notiert.</p> <p>[40] 5. – 8. ♩ Sgst. — Bg. ergänzt entspr. T. 4</p> <p>[44] 3. ♩ r. H. — Zweite ♩-Triole fis" statt e', vgl. T. 8.</p> <p>[55] Sgst./l. H. o. St. — \curvearrowright fehlen.</p> <p>Ergänzte Erinnerungsvorzeichen:</p> <p>[1 – 3, 16, 17] vorletzte ♩-Triole h</p> <p>[2] 2. ♩ H</p> <p>[9, 45] 6. ♩-Triole g</p> <p>[13, 49] 9. ♩-Triole d'</p>	<p>— Schließlich wurde eine fehlende \curvearrowright vor der Oktave ergänzt.</p> <p>[26] 1. ♩. l. H. — Verl.g.pkt. nach [gis'/gis"] fehlen.</p> <p>[32] — Zwischen den Systemen stehen Fortsetzungsstriche wie nach einem <i>cresc.</i> Getilgt, da keinerlei Zweck erkennbar ist. ↘ 5. ♩ r. H. — ♯/♯ vor [f'''/ais"] fehlen.</p> <p>[36] 4. – 5. ♩ r. H./Sgst. — ♭ vor es" und es' fehlen.</p> <p>[48 – 52] l. H. — Artikulation wurde entsprechend T. 13 – 17 angeglichen. Einzelheiten siehe dort.</p> <p>[59] Kl. — Entsprechend T. 24 wurde die Oktave [b/b'] auf der 2. ♩. ins o. Sys. versetzt. Die Haltebgg. der Oktave fehlen in der Vorlage, an ihrer Stelle stehen lediglich zwei Stimmführungsstriche.</p> <p>[61] 1. Schlag l. H. — ♩ fehlt. ↘ 5., 6. ♩ r. H. — ♭ vor [des'''/fes'''/a""], b" fehlen.</p> <p>[71] 4. – 5. ♩ r. H./Sgst. — ♭ vor es", es' fehlen.</p>
--	---

Figura 22: Fragmento de las notas críticas, obra de Alexander Gurdon y Urs Liska, que aparecen en la integral de los *lieder* de Oskar Fried (Fried, 2013).

Junto al informe crítico, se explica en el apéndice que toda la edición del texto se realizó sobre el sistema de tipografía Lua \LaTeX , un desarrollo basado en \TeX/\LaTeX , y que el control de versiones se gestionó con Git, el sistema que se describe en el apartado 2.10 (pág. 41). Para la inserción de caracteres musicales dentro del texto en las notas críticas se empleó un sistema hecho a propósito y publicado bajo licencia libre, denominado *lilyglyphs* (*lilyglyphs*),³ del propio Urs Liska. La figura anterior es una muestra de esta aplicación.

²El sitio oficial del premio «Mejor edición» es best-edition.de. El jurado estaba compuesto por: Prof. Iris Utikal, Prof. Dr. Wolfram Steinbeck, Michael Struck-Schloen, Thomas Neuhof y Daniela Zimmer.

³El sitio de *lilyglyphs* dentro de la red de recursos de \TeX (CTAN) es ctan.org/pkg/lilyglyphs

Los autores de la tipografía musical comentan que la parte más laboriosa de la edición musical fue el proceso que denominan “embellecimiento de las ligaduras”, que es tedioso porque consiste en una larga sucesión de ciclos de ensayo y error que no termina hasta que se obtiene un conjunto de parámetros numéricos óptimo para el trazado de las curvas.

5.3. Los arreglos y orquestaciones de Aurélien Bello

En este apartado damos cuenta de un tipo de evento que se puede caracterizar como de “actualidad” y sin embargo no podemos olvidar que las partituras que se están haciendo son material editorial que permanecerá en nuestro acervo cultural durante un tiempo indefinido. Se trata de trabajos de mucha envergadura y que están en la línea principal de la cultura “seria”, en los ámbitos internacionales de música sinfónica y los festivales de música clásica centroeuropeos.

Aurélien Bello es un joven director de orquesta y compositor con un impresionante currículum.⁴ Su primer trabajo importante con LilyPond ha sido una orquestación de los *Cuadros de una Exposición* de Modest Moussorgsky, hecha en 2011, que estrenó con la orquesta Junge Philharmonie Brandenburg el 10 de octubre de 2011 y luego volvió a interpretar los días 15 y 16 de abril de 2012.

Posteriormente reorquestó la ópera *El Anillo de los Nibelungos* de Richard Wagner a una versión para niños, escrita para orquesta sinfónica estándar y un conjunto de cantantes reducido. Se interpretó por parte de la Orquesta Sinfónica de la Radio de Berlín los días 5 y 6 de abril de 2014 en el Atze Musiktheater de Berlín con Heiko Matthias Förster como director.

En 2015, para el festival anual de Pascua de la Orquesta Filarmónica de Berlín, en Baden-Baden, donde siempre se interpreta una gran ópera y un buen número de conciertos sinfónicos, se eligió *Der Rosenkavalier* (“El caballero de la rosa”) de Richard Strauss. De forma adicional y gracias al muy activo programa educativo del director titular Sir Simon Rattle, se suele proponer una versión infantil de la ópera. El director Aurélien Bello recibió entonces el encargo de elaborar el arreglo de una versión reducida del *Rosenkavalier*. Está confeccionada para una plantilla de cuatro cantantes y orquesta de cámara con 13 músicos. Todas las partituras (vocales, guión de director y particellas) están hechas con LilyPond.

Este arreglo está comprometido para su edición por parte del sello Schott.⁵

⁴Véase su sitio personal aurelienbello.com. Ha dirigido la Orquesta Konzerthaus de Berlín, la Orquesta de Komische Oper Berlin, la Orquesta de la Radio de Munich, la Orquesta Sinfónica del Tirol (Innsbruck), la Brandebourgeois de Francfort, la Filarmónica de Jena, la Filarmónica de Bremen, la Orquesta de Cámara Folkwang de Essen, la Orquesta de Cámara de Pforzheim, la Orquesta Filarmónica de Baden-Baden, el Concerto Köln, etc.

⁵Desgraciadamente, en palabras del protagonista de este evento, en Schott *no saben, no pueden o no quieren trabajar con LilyPond* y van a volver a hacer la edición por su cuenta. De todas formas, el estreno se hizo con las partituras de LilyPond. Las fechas fueron 2, 6 y 26 de abril de 2015 en la Festspielhaus de Baden-Baden y en la Gran Sala de la Filarmónica de Berlín, dirigidas por Stephan Koncz (violoncelista de la Filarmónica) y por el propio

5.4. El libro *Un canto navideño a nuestra tierra, de Daniel López y Emilio Díaz*

Este libro-CD⁶ consiste en una colección de doce villancicos navideños originales sobre distintos pueblos de la provincia de Badajoz, y contiene como recurso educativo, para cada pieza, una explicación de cómo trabajarla y llevarla a la práctica en el aula. La música es de Daniel López Galván, docente de Educación Musical en el CEIP Santa Marina de Badajoz y de Emilio Díaz Vinuesa, docente de Educación Musical en el CEIP Ntra. Sra. de Guadalupe de Burguillos del Cerro, y la letra es de Antonio Jiménez Porro. El diseño y maquetación corrieron a cargo de Daniel Hervás Clivillés, diseñador gráfico y profesor de Dibujo en el IES Tierra Blanca de la Zarza. La tipografía musical es obra del autor.

Los títulos de los doce villancicos son los que aparecen en la tabla 6.

Tabla 6: Títulos de los doce villancicos del libro-CD *Un canto navideño a nuestra tierra* de D. López y E. Díaz.

Núm.	Título
1	<i>Plaza de la Soledad</i>
2	<i>Al huerto de San José</i>
3	<i>A jugar</i>
4	<i>Una patrulla de grullas</i>
5	<i>Un paseo por Llerena</i>
6	<i>Puerta de Palmas</i>
7	<i>La luna de Fregenal</i>
8	<i>Por el Puente Real</i>
9	<i>Navidades jerezanas</i>
10	<i>Dicen que por la Siberia</i>
11	<i>Bajo el Puente Romano</i>
12	<i>Que abran las puertas de Zafra</i>

Alrededor de esta obra se celebraron unas jornadas en el CPR de Jerez de los Caballeros durante cuatro días: 25 y 27 de noviembre, y 3 y 10 de diciembre de 2013, que contaba como ponentes a los mencionados D. López, E. Díaz y D. Hervás.⁷

El libro-CD no había sido publicado aún a fecha del verano de 2015, pero es posible disponer de alguno de los escasos ejemplares que se han impreso tanto en escala de grises como en color, de forma limitada, y en PDF bajo licencia *copyleft*.⁸

arreglista, respectivamente.

⁶Fuente: autores del libro.

⁷Los objetivos de estas jornadas rezaban así: “Conocer, comprender y enriquecer el patrimonio cultural extremeño desarrollando valores y comportamientos expresivos; explorar y conocer instrumentos musicales y su técnica como medio de acompañamiento y enriquecimiento en la interpretación musical; optimizar los recursos culturales de nuestra comunidad autónoma extremeña dotando con ello a los maestros de un repertorio sencillo, accesible y práctico; disfrutar a través de la interpretación de la música navideña; valorar las publicaciones escolares como recurso didáctico y de aprendizaje e iniciarse en la autoedición sobre la música tradicional navideña; respetar y valorar la riqueza del patrimonio musical de nuestra región; utilizar las TIC para la elaboración de recursos musicales; aprender e interpretar piezas instrumentales y vocales”.

⁸Véase el glosario, apéndice G.

Véase en la figura 23 un ejemplo extraído de este libro. La licencia exacta bajo la que se protege esta obra es CC-ND-NC (Creative Commons - Sin obra Derivada - No Comercial). La tipografía musical exclusivamente, está disponible bajo CC-BY-SA (Creative Commons - Atribución - Compartir igual). Aunque la letra de los villancicos son obra de un autor registrado en la SGAE, y aparece en las partituras musicales, el autor de la tipografía musical se reservó el derecho a publicar las partituras de forma independiente.

Plaza de la Soledad

Letra: Antonio Jiménez Porro Música: Daniel López Galván

$\text{♩} = 105$

The musical score is written in G major (one sharp) and 8/8 time. It consists of five staves of music. The first staff shows the beginning of the piece with a tempo marking of 105. The second and third staves contain the main verses of the villancico. The fourth staff is the chorus, marked 'Estribillo'. The fifth staff is a final line of the chorus. Above the notes, guitar chords are indicated with letters and diagrams. The lyrics are written below the notes.

A la Plaza de la Soledad
 llegaron pastores por la Navidad con un tarro de miel y un pastel, un trozo de
 queso y un poco de pan. Y la Virgen le dio de comer el trozo de
 queso y el poco de pan. Con un tarro de miel y un pastel, un trozo de queso y un poco de pan. A la
 plaza. Vamos a la Plaza de la Soledad, que las noches buenas son para cantar. Vamos a la
 plaza. Mira San José con la pande-reta y con el almi- rez. Vamos a la -reta y con el almi- rez.

Figura 23: Muestra de uno de los doce villancicos del libro “Un canto navideño a nuestra tierra”, de Daniel López y Emilio Díaz, con tipografía musical del autor.

Respecto al trabajo de edición mediante LilyPond, técnicamente no hay que mencionar ningún reto especialmente difícil, ya que el contenido musical es relativamente sencillo y tratamos deliberadamente de mantener los trucos de código reducidos a la mínima expresión. Todas las piezas contienen símbolos de repetición y se añadieron diagramas de posiciones para los acordes de la guitarra. Para la letra de las canciones se utilizó un tipo Square-BT a propuesta del autor de la maquetación.

Los doce archivos gráficos para las correspondientes piezas fueron aportados por el autor en

los formatos vectoriales EPS y PDF, y en los formatos de matriz de puntos PNG y TIFF a una resolución de 600 puntos por pulgada. Dado que la escala de la composición tipográfica y la medida de los márgenes no era conocida en el momento de la realización de las partituras, se pudo aprovechar la característica de orientación a texto de los documentos para elaborar un guión que programáticamente realizara todas las partituras a partir del código, sacando estos ajustes de disposición de página de un solo lugar centralizado. El guión producía todos los archivos gráficos cada vez; de esta manera, a partir del momento en que las mencionadas medidas estuviesen decididas, el material gráfico estaría inmediatamente disponible y con unos parámetros de maquetación uniformes. Al mismo tiempo se hizo, cada vez, un montaje en que se mostraban todas las páginas en una sola imagen, y el conjunto completo de archivos se comprimía mediante el mismo guión para facilitar la tarea de compartirllos *online* entre los miembros del equipo.

5.5. **Grandes proyectos: Mutopia**

El proyecto Mutopia⁹ es, sin duda, el proyecto basado en LilyPond más ambicioso que existe. Se trata de una amplia colección de partituras de libre acceso aportadas por los usuarios. La mayor parte del material existente se encuentra protegido por la licencia Creative Commons. En total se encontraban disponibles en noviembre de 2014, 1.888 piezas¹⁰ de música en partitura completa, clasificadas en 13 estilos diferentes de música, para 32 tipos de conjunto instrumental, y por cada uno de sus 298 autores.¹¹

Las piezas son en su mayoría de J.S. Bach (con 410 partituras) y de W.A. Mozart (con 88). Los estilos más representados son el llamado “barroco”, con 593 obras, y el “clásico” con 555. El instrumento más frecuente es el piano (con 654 obras) seguido de la voz (con 376). Al comparar estas cifras con las de 2009, sorprende que el aumento en el número de obras de J.S. Bach no ha hecho aumentar el número de obras de estilo barroco, algo que solamente puede explicarse por una falta de actualización de la base de datos. Sin embargo, el número de obras clasificadas como “clásicas” ha aumentado en 93 piezas desde las 462 que había en 2009.

Su nombre proviene de la unión de las palabras «Música» y «Utopía», lo que ofrece una pista sobre la intención que mueve este proyecto. La procedencia de las partituras debe cumplir una serie de requisitos entre los que citamos los siguientes que aparecen bajo el título «*Compruebe el estatus de copyright de la fuente*»:

Si contribuye con música compuesta por usted mismo, esto no es de aplicación para usted. Sin embargo, si contribuye con música clásica, incluso si es un arreglo propio, entonces debe com-

⁹El sitio oficial del Proyecto Mutopia es mutopiaproject.org

¹⁰El 2009 se realizó un registro provisional de las cifras de este proyecto y otros similares, que sirve de comparación con las cifras posteriores. Según este registro, el número total de piezas era de 1.620.

¹¹Eran 270 autores en 2009. Incluye pseudo-autores como “Anónimo” y “Tradicional”.

probar que la fuente de la música que va a introducir está libre de copyright. Para nuestros propósitos tiene que cumplir los criterios siguientes. Esto nos asegura que la música se encuentra en el dominio público en tantas partes del mundo como sea posible.

1. El compositor, letrista, arreglista y editor deben haber muerto hace más de setenta años completos.
2. La obra debe haber sido publicada antes de 1923.

A pesar del magnífico esfuerzo colectivo de tipografiar todas estas partituras utilizando LilyPond como *lingua franca*, el proyecto presenta algunos problemas. En primer lugar, el tamaño de la colección crece a un ritmo relativamente lento, pues no se admiten partituras digitalizadas de originales impresos, sino únicamente reediciones originales, y solo aquellas hechas con LilyPond, como se ha dicho.

Además, el hecho de que las partituras sean una contribución de usuarios con muy distintos niveles de *pericia* en el manejo de la sintaxis del lenguaje, produce un resultado de calidad irregular: se encuentran pésimos trabajos sin ninguna elaboración junto a extraordinarias composiciones tipográficas que igualan en calidad a partituras comerciales.

Puesto que en el mundo de las ediciones comerciales también existe una abismal diferencia entre las ediciones de buena y mala calidad, es justo admitir que el problema referido no es exclusivo del proyecto Mutopia, es más, ni siquiera de los repositorios de partituras de libre acceso aportadas voluntariamente por usuarios, de las que la gran Biblioteca Coral de Dominio Público (CPDL) es un buen ejemplo.

Al tratar de resolver la dialéctica de la comparación «a tres bandas» entre la calidad aceptable que LilyPond produce con una mínima intervención, la calidad irregular de la que otros editores de partituras no están a salvo, y la calidad asimismo irregular de las partituras impresas comerciales, podemos comprender por qué se ha dicho del proyecto Mutopia que «erosiona la buena imagen que LilyPond se ha hecho en otros lugares».¹²

Esta situación está cambiando precisamente en unos momentos en que se está desplegando, en el curso del año 2015, un esfuerzo colectivo por actualizar el material alojado en Mutopia a versiones de LilyPond más recientes, limpiando el código y simplificando las instrucciones necesarias para conseguir determinados ajustes visuales. El esfuerzo está coordinado por Chris Sawyer y se ha centrado en una primera fase en los archivos más antiguos, que estaban hechos con versiones del programa lanzadas en los años 90.

Afortunadamente, el programa contempla un mecanismo de actualización semi-automática de los archivos antiguos. No es completamente automática porque la sintaxis del lenguaje cambia en ocasiones de tal manera que es imposible saber exactamente qué quiso decir el

¹²La cita aparece en article.gmane.org/gmane.comp.gnu.lilypond.general/39516/

autor original con la sintaxis anterior. Sin embargo, el tiempo necesario para completar la tarea con éxito se reduce de forma muy apreciable, sobre todo en archivos grandes.

5.6. Musipedia

El proyecto Musipedia¹³ es una colección enciclopédica de melodías y temas musicales que se puede explorar, editar y expandir.

En este proyecto colaborativo se puede buscar una melodía por uno o varios de los siguientes criterios:

- Ejecución de la melodía en un piano representado en la página.
- Silbar o tararear la melodía.
- Percutir el ritmo en el teclado del ordenador.
- Introducir el *código de Parsons* de la melodía, que representa el perfil de intervalos ascendentes y descendentes
- por su codificación en el lenguaje de LilyPond.

Los miembros registrados editan y añaden melodías nuevas. El sistema, de libre acceso, permite a cualquiera localizar rápidamente melodías parecidas a una dada o basadas en ella. Internamente, las melodías se guardan en el código de LilyPond, que también genera las muestras de partitura que se usan para presentar los fragmentos musicales. El resto de los componentes que forman la infraestructura de este servicio también pertenecen a la categoría de Software Libre, como la base de datos que almacena la información y el sistema de gestión de los contenidos.

5.7. Transcripción del patrimonio cultural: la colección de T. L. de Victoria, de Nancho Álvarez

El joven violinista y profesor Nancho Álvarez, doctor en Matemáticas, es un usuario de LilyPond que se cuenta sin duda entre los primeros de España. Para el grupo vocal de la Fundación de la Universidad de Málaga, coro al que pertenece, ha transcrito nada menos que 321 piezas polifónicas del compositor abulense del s.XVI Tomás Luis de Victoria, además de 356 obras de Francisco Guerrero, 117 de Cristóbal de Morales y 486 obras de otros compositores,¹⁴ empleando el sistema LilyPond. La colección completa comprende 1.359 partituras.

¹³El sitio oficial del Proyecto Musipedia es musipedia.org

¹⁴Estas cifras eran en el año 2009 de 328 para Victoria, 160 de Guerrero, 109 de Morales y 351 de otros. Álvarez ha efectuado algún tipo de reestructuración en las obras de Victoria; y ha aumentado en 196, 8 y 135 el resto, respectivamente. Vemos que ahora Morales supera a Victoria en número de piezas. Fuente: Nancho Álvarez.

En su gran mayoría, la fuente para la música de Victoria es la edición de Felipe Pedrell:

A principios del siglo XX, Felipe Pedrell editó las obras completas de Victoria en 8 tomos (Leipzig: Breitkopf und Härtel, 1902-1913). Sin duda fue un esfuerzo editorial muy meritorio, sin embargo esta edición contiene innumerables errores, algunos tan evidentes que hasta un inexperto puede verlos. Por otra parte, es un poco incómoda de leer porque las partituras están escritas con las claves originales. De cualquier manera, esta fue la única referencia para muchas obras de Victoria durante casi 100 años.

En 1965 la editorial Gregg Press (Ridgeway, NJ), hizo una reedición. Tanto esta reedición como la original son imposibles de encontrar hoy en día. Afortunadamente, la empresa University Music Editions ha vuelto a reeditar esta obra recientemente, en esta ocasión en formato de microfichas. La ventaja es que ocupa muy poco espacio, pero por contra, es necesario un lector especial (...) ¹⁵

La ingente aportación de Nancho Álvarez al mundo coral, así como a la difusión del software libre para la tipografía musical, se encuentra en forma de archivos PDF, PostScript, archivos MIDI didácticos de conjunto y por voces separadas, y código fuente de LilyPond. De especial importancia en el entorno de los coros no profesionales, y muy al contrario que las ediciones impresas o en el formato de algún editor de partituras *cerrado*, a partir de este trabajo es posible por parte de cualquier persona realizar las correcciones oportunas y volver a generar el material en PDF.

En este caso no hablamos de estrictas ediciones musicológicas, anotadas o comentadas, sino de partituras listas para cantar como instrumento de orden eminentemente práctico para la actividad coral. El material está revisado varias veces, por lo que contiene pocas erratas.

Aunque se encuentra en proceso de actualización, muchas de las partituras aún aparecen realizadas en el lenguaje de una versión ya obsoleta de LilyPond, y por ello su realización con versiones actuales posiblemente requiera una cierta intervención manual, ya que la sintaxis del lenguaje ha variado en cierta medida.

En el campo de la transcripción del patrimonio cultural, se apuntan en la sección 5.15 otros usos especialmente interesantes.

5.8. La *Ópera Libre* de Valentin Villenave

Esta ópera se estrenó en febrero de 2009 (Etapà, 2008, 2009) en la *Opéra Comédie* de Montpellier, uno de los tres principales teatros de Francia, con lo que el autor esperaba que produjera una fuerte repercusión en los medios que hablan de LilyPond (al menos en Francia) y de la Cultura Libre, ya que era la primera ópera que se iba a publicar bajo una licencia de software libre GPL/Creative Commons. Véase una muestra en la página 107.

¹⁵La cita procede del sitio tomasluisdevictoria.org, alojado en los servidores de la Universidad Politécnica de Valencia.

Prologue

The musical score is titled "Prologue" and is set in a Largo tempo (♩ = 48). It is divided into three systems, each with a 3/4, 4/4, and 2/4 time signature. The first system includes Flute 1, Flute 2, Clarinette en La, Clarinette en Sib, Saxophone Alto 1, Saxophone Baryton, Marimba, Percussions, and Timbales. The second system includes Le Chef and Le Roi. The third system includes Violon 1, Violon 2, Violon 3, Alto 1, Alto 2, Violoncelle 1, Violoncelle 2, Contrebasse, and Piano. Dynamics range from mp to f. Performance instructions include "bag. dures", "en dehors", and "sui ponti".

Figura 24: Primera página del prólogo de la Ópera libre de Valentin Villenave.

Los datos siguientes provienen de una entrevista personal con el autor en la época de la composición de la ópera, y de los artículos al respecto de Estapà (2008, 2009).

Valentin Villenave es un pianista y profesor de música de 24 años. Ha estudiado literatura francesa y lleva componiendo muchos años, sin que hasta ahora se hayan interpretado ni grabado obras suyas.

En 2005 se encontraba trabajando para su tesis de Máster sobre un popular autor francés de historietas, llamado Lewis Trondheim.¹⁶ Se puso en contacto con él por correo electrónico y le preguntó si estaría dispuesto a escribir una ópera con él. Aceptó inmediatamente, y ambos comenzaron a trabajar en un pequeño proyecto que estaba destinado a ciertos cantantes a los que Valentin ya conocía por su actividad como pianista acompañante.

En 2006, Lewis Trondheim obtuvo el más alto reconocimiento que se concede a los escritores de cómic en Francia, en el Festival de Angoulême. Uno de los cantantes para los que estaba escribiendo la ópera le aconsejó que presentara el proyecto a algunos grandes teatros de Francia; como Lewis vivía en la ciudad de Montpellier, se puso en contacto con René Kœring, que estaba al cargo de la ópera de Montpellier, una de las más famosas de Francia. Dos horas más tarde contestó que representaría encantado su ópera...

Desde el primer momento tenía claro que iba a editar y publicar la partitura por su propia cuenta, y que iba a lanzarla bajo una licencia “Libre”. Como músico y profesor, se encontraba en una buena situación desde la que observar la manera en que el actual sistema de *copyright* pone en riesgo a la música contemporánea; como ciudadano, no podía quedarse parado viendo cómo se violaban los derechos humanos y civiles en nombre de la llamada “propiedad intelectual”. Él sabía que, de ser alguno, el sistema de software libre para la notación musical se llamaría LilyPond (aunque nunca lo había usado). Así que empezó a aprender a usarlo para tipografiar la partitura de la ópera, comenzó a estudiar inglés para estar en contacto con la comunidad de usuarios y además se fue convirtiendo en patrocinador, colaborador y miembro de sus equipos de documentación y desarrollo.

«No solo LilyPond hace que mi partitura sea mucho más atractiva visualmente que nada de lo que recuerde haberse publicado en las últimas décadas, sino que además me permite algunos efectos muy especiales (habiendo sido una ópera adaptada a partir de un cómic, los elementos gráficos jugaban un importante papel). Por encima de todo, me produce un gran ahorro de tiempo y una increíble tranquilidad en comparación con la que habría tenido que emplear con cualquier otro programa.¹⁷ Por último, el enfoque basado en código fuente es más *robusto* que ningún otro, pues me permite desarrollar y licenciar mi partitura como un programa bajo la licencia GPL,¹⁸ permitiendo así futuras traducciones del libreto a otros idiomas, posibles transcripciones por ejemplo para conservatorios, etc.

»La primera representación de *Affaire Étrangère* (“Asuntos exteriores”) está programada para

¹⁶El sitio oficial del Lewis Trondheim, el libretista de la ópera de Valentin Villenave, es lewistrondheim.com.

¹⁷Valentin Villenave declaró más tarde que aprendió el lenguaje de LilyPond cuando ya tenía doscientas páginas de su ópera tipografiadas con otro programa, y que empezó de cero al cambiar de sistema.

¹⁸La Licencia Pública General del proyecto GNU se ha descrito anteriormente; recordemos que es una amplia declaración de permisos para el usuario de software libre. Véase el glosario del apéndice G.

febrero de 2009 en Montpellier. Junto a la representación, he propuesto también que la ópera suponga la organización de una gran cantidad de eventos didácticos (con conservatorios, en librerías, etc.) en la ciudad durante varias semanas. El hecho de tener una partitura libremente reproducible y modificable por cualquiera es una inmensa ayuda en este enfoque, y ello solamente es posible con LilyPond.»

El autor personalmente cede unas referencias que ilustran esta descripción y facilitan la ampliación de información, así como la partitura completa de la ópera.¹⁹

En estas líneas vemos que Valentin aprecia una ventaja en la orientación a documentos de texto en el sentido de que aporta *robustez* al manejo del material fuente, y aunque no menciona haber utilizado ningún sistema de control de versiones, sabemos que empleó el sistema Git (Torvalds y Hamano, 2005) y de hecho la forma en que el documento original de su ópera está publicado, es la de un repositorio público de Git. Esto implica que no solo el código, sino también la historia detallada de su evolución, están disponibles bajo la misma referencia.

5.9. El cancionero de A. Simpson

Continuando con las aplicaciones de algún renombre que utilizan LilyPond, es digna de mención la edición digital del cancionero de Allen, Ware y Garrison (1867) *Canciones de esclavos de los Estados Unidos*, editado originalmente por A. Simpson.

Esta reedición digital de 2007, que se publicó bajo una licencia Creative Commons, estuvo realizada por el matemático alemán Marcus Brinkmann y contiene 136 canciones. Una versión preliminar se encuentra disponible para su descarga gratuita.²⁰

El formato es A5 vertical (véase una muestra en la figura 25, pág. 110).

Los textos que se encuentran fuera de la partitura se han tipografiado con la fuente Century SchoolBook, que es la misma que se usa en la letra de los fragmentos musicales, por lo que el resultado es de gran coherencia visual. Este tipo de letra es el que LilyPond utiliza de forma predeterminada.

¹⁹La partitura completa de la ópera “Affaire Étrangère” de Valentin Villenave, con libreto de Lewis Trondheim está en repo.or.cz/w/opera_libre.git.

²⁰La edición digital se aloja en marcus-brinkmann.de/slave-songs.html.

1. Roll, Jordan, roll.

1. My brudder¹ sittin' on de tree of life, An' he yearde when Jordan
roll; Roll, Jordan, Roll, Jordan, Roll, Jordan, roll!

O march de an-gel march, O march de an-gel march; O my
soul arise in Heaven, Lord, For to yearde when Jordan roll.

2. Little chil'en, learn to fear de Lord,
And let your days be long;
Roll, Jordan, &c.

3. O, let no false nor spiteful word
Be found upon your tongue;
Roll, Jordan, &c.

[This spiritual probably extends from South Carolina to Florida, and is one of the best known and noblest of the songs.]
[The following variation appears above bars 4–6 in the original print. - Ed.]

roll; Roll, Jordan, Roll, Jordan,

¹Parson Fuller, Deacon Henshaw, Brudder Mosey, Massa Linkum, &c.

Figura 25: Muestra extraída de la edición digital del cancionero *Canciones de esclavos de los Estados Unidos* (Allen, Ware y Garrison, 1867) editado originalmente por A. Simpson.

5.10. El cancionero *Land of plenty*, de Jack Cooper

En el año 2001, el cantautor americano/canadiense Jack Cooper lanzó una grabación discográfica de canciones bajo el título *Land of plenty* («Tierra de plenitud»).

Este lanzamiento estuvo acompañado de la edición mediante LilyPond de las partituras completas para voz y guitarra de todas las canciones del disco, como descarga gratuita, a modo de tarjeta de presentación y promoción de su CD. Hay una muestra extraída de este cancionero en la figura 26.

36

These Bones Are Gonna Rise Again

Guitar and Voice

Jack Cooper- c.2001

Slowly and Expressively

Verse 1: The Lord thought he'd make a man

These bones are gonna rise a-gain Made him of mud and a handful of sand

These bones are gonna rise a-gain Refrain: I know it Oh, Lord I know it

I know it These bones are gon-na rise a-gain

-- To play tabs and chords shown in the fret diagrams, place capo on second fret --

Figura 26: Muestra extraída del cancionero *Land of Plenty* de Jack Cooper

Incluimos esta muestra por su interés como ejemplo de música *pop* cuya partitura incluye innumerables efectos de uso común en este estilo, como: diagramas de trastes y cejilla para guitarra; secciones repetidas; títulos de sección en las canciones; repeticiones de primera y segunda vez; varias letras sobre una melodía común; tablatura de guitarra (notación por cifra); cambios de compás; indicaciones de afinación alternativa; indicaciones de tempo complejas; calderones; indicaciones dinámicas y expresivas, e indicaciones e instrucciones complementarias.

Agradezco a Ken Brown que me presentara este programa y a la comunidad de Lilypond por proporcionar tanta cantidad de documentación y un fórum para la solución de las cuestiones técnicas. A diferencia de la mayoría de los programas editores de partituras que utilizan un interfaz

de «apuntar y pulsar», Lilypond es un lenguaje que produce un resultado de aspecto increíblemente bueno y aporta infinitas formas de personalizar la salida. ¡Lilypond²¹ es el software más sorprendente que he tenido el placer de utilizar!

Es justo decir que actualmente el programa tiene una forma (orientada a texto) de utilizar el ratón, y que se denomina precisamente “apuntar y pulsar”. Se puede explotar siempre y cuando se utilice un editor de texto que tenga un visor integrado de PDF, como Frescobaldi (Berendsen, 2008a), algo que no existía cuando Cooper hizo estos comentarios.

El formato del papel corresponde al tamaño Letter (Carta americana). El tipo de letra del texto es la tipografía *Maiandra GD*, de aspecto informal/rústico, de la fundición Agfa Monotype y bajo licencia comercial.

El resultado de este trabajo no está exento de problemas, entre los que podemos citar los siguientes (que pueden apreciarse en el ejemplo incluido) y que se pueden corregir con una edición del código más elaborada, o bien utilizando una versión del programa más reciente: los silencios de un compás completo no están centrados entre las líneas divisorias; los acordes con sufijo no están centrados sobre el diagrama; por último, las sílabas de la letra se alinean a la izquierda con la figura correspondiente (deberían ir centradas, según Gerou y Lusk (2004)). La versión actual de LilyPond corrige estos problemas.

5.11. Serpent Publications, de Laura Conrad

Laura Conrad es una estudiosa de la música del Renacimiento residente en Cambridge, Massachusetts (EUA), que posee una editorial dedicada a la música vocal e instrumental del Siglo de Oro de la polifonía bajo la marca “Serpent”.

En su fondo editorial de partituras, la mayoría editadas con LilyPond aunque las más antiguas se tipografiaron con el sistema ABC, se pueden encontrar obras dispuestas al modo tradicional, con detalles historicistas como las partes separadas (una *particella* independiente por cada voz polifónica), las notas guía al final de las líneas y la escritura sin compasear, como los originales que transcribe. Una muestra del trabajo de Laura Conrad puede verse en la figura 27.

Dos tipos de publicaciones podemos encontrar en el catálogo de Serpent: piezas sueltas, que están disponibles para su libre descarga tanto en formato PDF como en forma de código de LilyPond nativo, y libros completos. La propia editorial no imprime los libros, sino que un servicio de impresión a demanda los hace bajo pedido.

²¹ Obsérvese que Cooper utiliza la capitalización normal porque aún no se había establecido oficialmente en el proyecto la capitalización llamada “de jorobas de camello” en la que aparece la letra P mayúscula dentro de la palabra.

53. [Pavan:] Last will and testament
Cantus

Anthony Holborne

Altus

1

Figura 27: Muestra de una *pavana* de Anthony Holborne editada por Laura Conrad para el catálogo de Serpent Publications. La partitura aparece por cortesía de Laura Conrad.

Conrad transcribe piezas que en su mayoría son de compositores ingleses como John Dowland, Thomas Morley, John Mundy o Thomas Weelkes, aunque también están representados los flamencos como Josquin Desprez y los franceses como Claudin de Sermisy.

La licencia de las publicaciones es la Licencia General Pública de GNU (GPL), que como sabemos es una licencia que incluye la cláusula *copyleft*.

El hecho de acogerse a la cláusula *copyleft* significa que podemos utilizar la música con cualquier propósito, pero que si se la entregamos a otra persona debemos incluir una copia del texto de la licencia GPL, y debemos continuar reconociendo los créditos a favor de Serpent Pu-

blications, así como ofrecer un enlace al lugar en que las fuentes de las transcripciones están disponibles.

Según Laura Conrad,

(...) [LilyPond] es un proyecto de software libre para producir música impresa en la tradición de los grandes grabadores del s.XIX. Un cierto número de personas dentro de la comunidad de desarrolladores de LilyPond están interesadas en la transcripción de música del Renacimiento e incluso más antiguas con este programa, motivo por el cual el soporte para funcionalidades de música antigua es mejor que en muchos de los otros programas de transcripción que existen.

5.12. Kainhofer Edition, de Reinhold Kainhofer

La Edición Kainhofer de Viena,²² fundada en 2010, publica joyas musicales de los siglos preteritos que habían quedado relegadas al olvido, principalmente obras corales, orquestales y de música de cámara de los períodos barroco y clásico, además de arreglos para diversos conjuntos de cuerda.

En la sección Eybler-Edition se editan científicamente y se publican como *Urtext* (textos originales) numerosos graduales y ofertorios del maestro de capilla Joseph Edler von Eybler, amigo personal de Mozart, que recibió inicialmente el encargo de concluir la composición de su famoso Requiem.

También publica nuevas ediciones para trío de cuerda, principalmente arreglos escritos por los miembros del Trío Aleksic.

El Dr. Reinhold Kainhofer, fundador del sello editorial, estudió matemáticas y física teórica y trabaja como investigador en la Universidad de Tecnología de Viena. Es cantante y miembro de los coros Arnold Schönberg Choir, Wiener Singakademie y coro de St. Augustine.

Kainhofer es un destacado miembro del equipo de desarrollo de LilyPond, con una intensa actividad enfocada en la compatibilidad con el estándar MusicXML y en las secciones de expansión en lenguaje Scheme. Además se le conoce por haber diseñado una de las mejores hojas de referencia del lenguaje, que estuvo vendiendo en la tienda online de su editorial hasta que decidió ofrecerla para su libre descarga y abrir también el código fuente. Precisamente fue a partir de su trabajo en esta hoja compacta de referencia, desde donde se preparó la versión española. La figura 28 ofrece una visión reducida de dicha hoja.

²²El sitio oficial de esta editorial es edition-kainhofer.com



LilyPond - Music Typesetting

Basic usage - LilyPond Version 2.14 and above
 Cheatsheet by R. Kainhofer, Edition Kainhofer, <http://www.edition-kainhofer.com/>



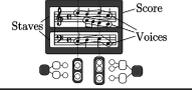
<p>Command-line usage</p> <pre>lilypond [-l LOGLEVEL] [-dSCOPTIONS] [-o OUTPUT] [-V] FILE.ly [FILE2.ly]</pre> <p>Common options:</p> <table style="width: 100%;"> <tr> <td>-pdf, -png, -ps Output file format</td> <td>-dpreview Cropped "preview" image</td> </tr> <tr> <td>-dbackends-eps Use different backend</td> <td>-dlog-file=FILE Create log file</td> </tr> <tr> <td>-l LOGLEVEL ERR/WARN/INFO/DEBUG</td> <td>-dno-point-and-click No Point & Click info</td> </tr> <tr> <td>-o OUTDIR Name of output dir/file</td> <td>-djob-count=NR Process files in parallel</td> </tr> <tr> <td>-v Verbose output</td> <td>-dpxmap-format=pngalpha Transparent PNG</td> </tr> <tr> <td>-dhelp Help on options</td> <td>-dno-delete-intermediate-files Keep .ps files</td> </tr> </table>	-pdf, -png, -ps Output file format	-dpreview Cropped "preview" image	-dbackends-eps Use different backend	-dlog-file=FILE Create log file	-l LOGLEVEL ERR/WARN/INFO/DEBUG	-dno-point-and-click No Point & Click info	-o OUTDIR Name of output dir/file	-djob-count=NR Process files in parallel	-v Verbose output	-dpxmap-format=pngalpha Transparent PNG	-dhelp Help on options	-dno-delete-intermediate-files Keep .ps files	<p>General Syntax</p> <pre>\xxxx function or variable { ... } Code grouping var = {...} Variable assignment \version "2.14.0" LilyPond version % dots Comment %{ ... %} Block comment c{...} Postfix notation (notes) #{...}, ##t, #sym Scheme list, true, symb. x...x...x... Directions</pre>																																																		
-pdf, -png, -ps Output file format	-dpreview Cropped "preview" image																																																														
-dbackends-eps Use different backend	-dlog-file=FILE Create log file																																																														
-l LOGLEVEL ERR/WARN/INFO/DEBUG	-dno-point-and-click No Point & Click info																																																														
-o OUTDIR Name of output dir/file	-djob-count=NR Process files in parallel																																																														
-v Verbose output	-dpxmap-format=pngalpha Transparent PNG																																																														
-dhelp Help on options	-dno-delete-intermediate-files Keep .ps files																																																														
<p>Basic Notation</p> <table style="width: 100%;"> <tr> <td></td> <td>Note names (Dutch)</td> </tr> <tr> <td></td> <td>Alterations: -is/-es for sharp/flat, -is/is/-eses for double, ! forces, ? shows cautionary accidental</td> </tr> <tr> <td></td> <td>Relative mode (change less than a fifth), raise/lower one octave</td> </tr> <tr> <td></td> <td>Automatic and manual beaming</td> </tr> <tr> <td></td> <td>rests: r (like a note name), s for invisible rests</td> </tr> <tr> <td></td> <td>Clefs</td> </tr> <tr> <td></td> <td>Time signature</td> </tr> <tr> <td></td> <td>Key</td> </tr> <tr> <td></td> <td>Chords, repeated chord</td> </tr> <tr> <td></td> <td>Tie</td> </tr> <tr> <td></td> <td>Slur / Phrasing slur</td> </tr> <tr> <td></td> <td>Articulations and text markup</td> </tr> <tr> <td></td> <td>Partial measure, bar line check, bar lines</td> </tr> <tr> <td></td> <td>Full-/Multimeasure rests: R (with exact duration)</td> </tr> <tr> <td></td> <td>Repeats (with alternative endings; can be left out)</td> </tr> <tr> <td></td> <td>Tempo indications, text marks</td> </tr> <tr> <td></td> <td>Grace notes</td> </tr> <tr> <td></td> <td>Triplet, tuplets</td> </tr> <tr> <td></td> <td>Parallel music</td> </tr> </table>		Note names (Dutch)		Alterations: -is/-es for sharp/flat, -is/is/-eses for double, ! forces, ? shows cautionary accidental		Relative mode (change less than a fifth), raise/lower one octave		Automatic and manual beaming		rests: r (like a note name), s for invisible rests		Clefs		Time signature		Key		Chords, repeated chord		Tie		Slur / Phrasing slur		Articulations and text markup		Partial measure, bar line check, bar lines		Full-/Multimeasure rests: R (with exact duration)		Repeats (with alternative endings; can be left out)		Tempo indications, text marks		Grace notes		Triplet, tuplets		Parallel music	<p>Creating Staves, Voices and Groups</p> <pre>\version "2.16.0" \SMusic = \relative c'' { c1\p } \SLyrics = \lyricmode { Oh! } \AMusic = \relative c' { e1 } \score { \new ChoirStaff << \new Staff { \new Voice = "Sop" { \dynamicUp \SMusic } } \new Lyrics = "SLyrics" \lyricsto "Sop" \SLyrics } \new Staff { \new Voice = "Alto" { \dynamicUp \AMusic } } } >></pre> <p>Useful Contexts</p> <table style="width: 100%;"> <tr> <td>Score</td> <td>The whole score</td> </tr> <tr> <td><< ... >></td> <td>Staves in parallel</td> </tr> <tr> <td>StaffGroup</td> <td>Staffgroup with bracket</td> </tr> <tr> <td>GrandStaff</td> <td>Staffgroup with brace</td> </tr> <tr> <td>ChoirStaff</td> <td>Staffgroup with bracket</td> </tr> <tr> <td>PianoStaff</td> <td>Staffgroup with brace</td> </tr> <tr> <td>Staff</td> <td>One single staff</td> </tr> <tr> <td>Voice</td> <td>A voice inside a staff</td> </tr> <tr> <td>CueVoice</td> <td>Smaller voice for cues</td> </tr> <tr> <td>Lyrics</td> <td>Lyrics inside staff/voice</td> </tr> <tr> <td>ChordNames</td> <td>Chords (lead sheet)</td> </tr> <tr> <td>Devnull</td> <td>No output</td> </tr> </table> <p>Also: Dynamics, FiguresBass, Tab(Staff Voice), Drum(Staff Voice), RhythmicStaff, FretBoards, ...</p> 	Score	The whole score	<< ... >>	Staves in parallel	StaffGroup	Staffgroup with bracket	GrandStaff	Staffgroup with brace	ChoirStaff	Staffgroup with bracket	PianoStaff	Staffgroup with brace	Staff	One single staff	Voice	A voice inside a staff	CueVoice	Smaller voice for cues	Lyrics	Lyrics inside staff/voice	ChordNames	Chords (lead sheet)	Devnull	No output
	Note names (Dutch)																																																														
	Alterations: -is/-es for sharp/flat, -is/is/-eses for double, ! forces, ? shows cautionary accidental																																																														
	Relative mode (change less than a fifth), raise/lower one octave																																																														
	Automatic and manual beaming																																																														
	rests: r (like a note name), s for invisible rests																																																														
	Clefs																																																														
	Time signature																																																														
	Key																																																														
	Chords, repeated chord																																																														
	Tie																																																														
	Slur / Phrasing slur																																																														
	Articulations and text markup																																																														
	Partial measure, bar line check, bar lines																																																														
	Full-/Multimeasure rests: R (with exact duration)																																																														
	Repeats (with alternative endings; can be left out)																																																														
	Tempo indications, text marks																																																														
	Grace notes																																																														
	Triplet, tuplets																																																														
	Parallel music																																																														
Score	The whole score																																																														
<< ... >>	Staves in parallel																																																														
StaffGroup	Staffgroup with bracket																																																														
GrandStaff	Staffgroup with brace																																																														
ChoirStaff	Staffgroup with bracket																																																														
PianoStaff	Staffgroup with brace																																																														
Staff	One single staff																																																														
Voice	A voice inside a staff																																																														
CueVoice	Smaller voice for cues																																																														
Lyrics	Lyrics inside staff/voice																																																														
ChordNames	Chords (lead sheet)																																																														
Devnull	No output																																																														
<p>Adding Lyrics</p> <pre><< \new Voice = "m" { c4 c8(\b) a8(\b) c4 c4- c4 } \new Lyrics \lyricsto "m" { Sim -- ple e -- xam -- ple .. } >></pre> 	<p>Writing text and markup</p> <pre>\newkup { \justify A underline justified \bold paragraph, \fontsize #2 automatically line-broken. Some symbols: \dynamic ff \keyglasses \rounded-box text, \with-color #red Text } A justified paragraph, automatically line-broken. Some symbols: ff \keyglasses [Text] Text</pre>	<p>Links</p> <p>LilyPond Homepage: http://www.lilypond.org/</p> <p>LilyPond Snippet Repository (LSR): http://lir.dsi.unimi.it/</p> <p>User Mailing list: lilypond-user@gnu.org</p> <p>Cheatsheet by R. Kainhofer, License:  Creative Commons Attribution-NonCommercial 3.0</p>																																																													

Figura 28: Hoja de referencia preparada por Reinhold Kainhofer.

5.13. LilyPond en educación y en transcripción.

Varios casos de aplicación en el mundo de la educación pueden documentarse en este apartado. Mencionaremos los casos de utilización en instituciones oficiales de enseñanza musical y en cursos de música.

5.13.1. Cursos en Badajoz

La comunidad autónoma de Extremadura es pionera en el empleo de software libre en la administración y en la educación; hemos hablado someramente del proyecto GNU/Linux en el apartado 2.11.1. Por otro lado, la ciudad de Badajoz dispone de un Conservatorio superior que es de titularidad provincial y es el único superior de Extremadura. En lo que respecta

a la edición de partituras, sin embargo, no fue hasta 2005 cuando en este conservatorio se comenzó a emplear software libre, en un proceso que impulsó la traducción completa de su documentación para facilitar a los alumnos su uso y aprendizaje.

La asociación nacional francesa de padres de alumnos de conservatorios de música (FNAPEC) se hizo eco de esta actividad (Villenave, 2009a) y en el número 54 (agosto de 2009) de su revista, difundida gratuitamente por todos los conservatorios del país vecino, dedica varias páginas a ello. Los párrafos que siguen, redactados por el autor, están tomados de esta revista.

[En España, un conservatorio ya tiene una clase completa de LilyPond.]

Aún a riesgo de aburrirlos con la historia de la filosofía de la libertad del software, debo decir que para la educación no veo ninguna alternativa al software libre. Es más, la tipografía musical es una tarea común, popular entre los alumnos, con un inmenso valor en la educación musical. Aquí es donde aparece en escena el sistema de tipografía musical GNU LilyPond.

Cada alumno del grupo al que imparto la enseñanza de LilyPond, tiene que realizar una serie progresiva de grabados musicales, la mitad de las cuales se toma de modelos fijos, mientras que la segunda mitad implica la composición tipográfica de piezas propuestas por ellos mismos y aprobadas por mí. Así, tienen que realizar una pieza solista libremente escogida para su propio instrumento; otra para un instrumento distinto, preferentemente un instrumento transpositor; una pieza de piano; una pieza de música de cámara; una pieza coral con letra y una partitura orquestal con todos los pentagramas (no la pieza entera, solo las dos o tres primeras páginas). Cada una de las treinta sesiones de un curso tiene una parte de teoría que utilizo para explicar el tema del día, la instalación y uso de los entornos de ayuda e interfaces de usuario, y asuntos tales como la filosofía del software libre. En el tiempo dedicado a la parte práctica, reviso sus trabajos de la sesión anterior por turnos, momento que aprovechamos para discutir el lenguaje y quizá proponer mejoras a su código. Finalmente, les encargo una nueva tarea esbozando muy brevemente lo que quiero que hagan e intentando centrar su atención sobre la amplísima documentación oficial, que se encuentra finalmente en español.

La longitud y complejidad de las piezas abarcan desde ejemplos de una sola línea que muestran una aplicación al mundo real de una determinada construcción del lenguaje LilyPond, hasta partituras orquestales completas. Siendo la mayor parte de todo el esfuerzo de la composición tipográfica de una partitura orquestal establecer la estructura de pentagramas y grupos de pentagramas, creo que una o dos páginas de este tipo de obras es suficiente para nuestros propósitos (aunque impartiese un segundo curso de 30 horas de nivel avanzado, no encargaría movimientos completos de obras orquestales, porque éstas no ofrecerían nada nuevo aparte del duro trabajo de teclado).

Los alumnos están, por lo general, muy orgullosos de la llamativa calidad de los resultados que se consiguen en cuestión de minutos (una vez que ya dominan los aspectos básicos del lenguaje), pero las primeras lecciones pueden ser graciosas porque en sus rostros se refleja una expresión de terror, que desaparece rápidamente cuando los primeros resultados se presentan en las pantallas de sus ordenadores.

Estando basado completamente en texto, LilyPond como sistema de grabado musical tiene la ven-

taja de mostrarnos a la vista todo lo que es necesario saber hacer para conseguir tal o cual elemento de notación, cualquiera que sea su complejidad. Este enfoque es el mismo que emplea el sistema de tipografía \LaTeX para libros científicos y cualquier texto complejo. Un documento de texto normal lo contiene todo, y el más rebuscado truco que podamos contemplar en una partitura hecha con LilyPond, siempre podemos adaptarlo a nuestras necesidades con tan solo modificar la altura de las notas, a condición de que podamos examinar el código que dio lugar a dicha partitura. En ocasiones intento imaginarme de qué otra forma podría haber corregido con exactitud y solo por correo electrónico, ejercicios que mis alumnos me hubieran enviado para su revisión cuando no hemos podido vernos cara a cara. Recuerdo haber escrito interminables explicaciones sobre la utilización de nuestro anterior software musical, basado en ratón, con frecuencia con poco éxito. A menudo debía concluir: “deja que te explique cómo se hace”, quitándoles el ratón de las manos, con respecto a algún problema de notación con aquellos programas. La falta de interfaz gráfico que «sufre» LilyPond nos permite concentrar nuestra atención sobre la estructura de la música y el significado de nuestro código. Esto podría sonar algo primitivo, pero no olvidemos que todo lo que queremos al final es tener la partitura hecha, con la mayor calidad que nos podamos permitir y con frecuencia de una forma extremadamente sencilla y casi mágica. Por decirlo de una manera simplista, la potencia y flexibilidad de este enfoque son casi ilimitados.

LilyPond no es difícil de aprender y de hecho es realmente rápido si te lo explica una persona junto a ti. Si mi curso se ofreciese como un seminario ininterrumpido, los alumnos podrán aprender el tipografiado de prácticamente cualquier partitura musical en lo que dura un curso de un fin de semana. A aquellos que a veces protestan por tener que manejar código, alegando que son músicos y no programadores, les contesto que LilyPond tiene una amplia base de usuarios compuesta en su totalidad por músicos. El solfeo, la armonía y la teoría musical no se encuentran tan lejos del código de tipografiado musical de LilyPond, después de todo.

Así pues, mi consejo es: denle una oportunidad a LilyPond, está pensado para todos. Localice en su entorno a una persona que le pueda mostrar el programa en acción, o suscríbase a uno de los grupos de Internet de apoyo entre usuarios, o apunte su navegador directamente a lilypond.org, descargue el programa y siga el tutorial. El grabado musical fácil y de alta calidad, es algo a lo que todos tenemos derecho.

(Trad. del autor).

La actividad aparece mencionada también en Lizarán (2010). De entre los aspectos mencionados, destacamos el empleo de técnicas de *e-learning* ya desde 2005 y el uso de las TIC, concretamente de listas de distribución de correo electrónico, en un centro de educación musical.

5.13.2. Congreso de la AEC, Strasbourg 2007

Del 8 al 10 de noviembre de 2007 se celebró en la ciudad de Strasbourg (Francia) el congreso anual de la Asociación Europea de Conservatorios (AEC)²³. Las actas (Association Européenne des Conservatoires, Académies de Musique et Musikhochschulen (AEC), 2007) refle-

²³El sitio oficial de la Asociación Europea de Conservatorios es aec-music.eu

jan las intervenciones de un cierto número de representantes de sus respectivos centros, una de las cuales versó sobre el empleo de software libre para la producción de partituras en educación musical. En la figura 29 puede verse la primera página de la presentación utilizada (Vila, 2007).

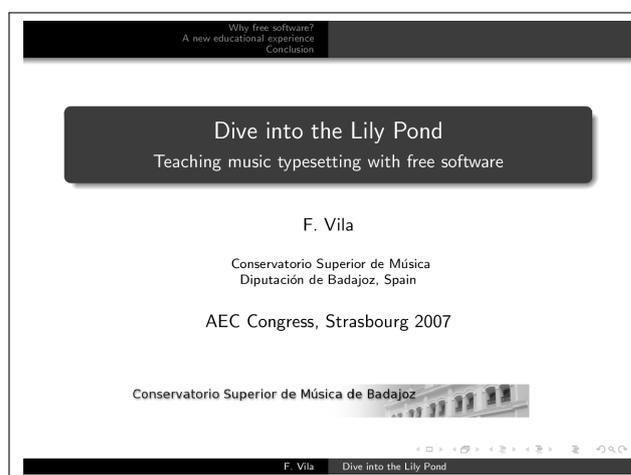


Figura 29: Primera página de la presentación sobre LilyPond que se hizo en el congreso anual de la Asociación Europea de Conservatorios en Strasbourg, noviembre de 2007.

Traducimos de la página 20 de la edición francesa de las actas:

Sumergirse en «el estanque de los nenúfares». Enseñanza de tipografía musical con software libre (Conservatorio de Badajoz).

El representante del centro [...], realiza una presentación sobre el software libre LilyPond y promueve el uso de este sistema por parte de los profesores de música como alternativa a los productos propietarios, con frecuencia caros y de utilidad limitada. LilyPond es un sistema de tipografiado de partituras musicales sencillo y libre, con todos los grados de complejidad (desde partituras a una voz hasta arreglos completos de orquesta). Como software libre, viene con el permiso para ejecutarlo con cualquier propósito y es una herramienta muy útil para la educación, pues p.ej. el profesor siempre puede hacer copias para los alumnos que a su vez son libres de instalar este sistema en sus hogares. El sistema se puede descargar desde www.lilypond.org.

La AEC es una red europea cultural y educativa, fundada en 1953, que representa los intereses de instituciones preocupadas por la preparación del músico. En la actualidad cuenta con 280 centros miembros de 55 países.

Entre los objetivos y aspiraciones de la AEC se encuentran la estimulación y apoyo a la colaboración internacional entre los centros miembros, la realización de distintos proyectos internacionales sobre temas de relevancia en la educación musical profesional, la búsqueda y promoción de estándares de control de calidad en la formación y la representación de los intereses del sector de la educación musical profesional en los niveles nacional, europeo e internacional.

5.13.3. Talleres en Valencia de Don Juan

Los alumnos de música, como en otras áreas, pueden completar su formación a través de una oferta de cursos que tienen lugar durante el descanso estival. Tenemos constancia de alumnos que han recibido formación específica sobre preparación de partituras musicales en un curso de verano impartido por 21 profesores de todas las especialidades musicales, celebrado del 7 al 16 de agosto de 2009 en Valencia de Don Juan (provincia de León),²⁴ en el que se ofreció un *Taller de iniciación a LilyPond: programa libre de edición informática de partituras* a cargo del guitarrista Bertrand Piétu. Se establecieron dos grupos de alumnos que recibieron clase durante cuatro días cada uno. Del folleto de estos cursos extraemos estas líneas escritas por el profesor Piétu:

Con Lilypond el usuario solo se preocupa de escribir la partitura (notas, ritmo, etc.) con un sencillo e intuitivo lenguaje en un archivo de texto. El programa se encarga después de leer ese archivo y de dar forma a los signos musicales basándose en las reglas de edición de las mejores editoriales profesionales.

Veremos cómo podemos instalarlo y usarlo en un ordenador. Finalmente, iremos poco a poco aprendiendo los conocimientos básicos para crear partituras con este programa y dándonos cuenta de que, con una alternativa libre a Finale y Sibelius podemos obtener, con más facilidad, partituras de mejor calidad.

Los talleres se han seguido impartiendo ininterrumpidamente hasta 2015.

5.13.4. Abjad, de Víctor Adán y Trevor Bača

En este caso presentamos una aplicación de LilyPond que es otro programa desarrollado por los compositores e informáticos Víctor Adán y Trevor Bača. El programa utiliza a su vez a LilyPond para producir el resultado impreso. *Abjad*²⁵ es una serie de funciones en el lenguaje de programación Python para el control de partituras formalizadas. Está diseñado para ayudar a los compositores a construir elaboradas piezas de notación musical de una manera intuitiva e incremental, es decir, es posible comenzar con algunas construcciones mínimas y aumentar la complejidad de forma paulatina. Abjad se utiliza para crear una representación simbólica de todas las notas, silencios, pentagramas, ritmos anidados, barras de figuras, ligaduras de expresión y otros elementos de notación dentro de una partitura. Debido a que Abjad envuelve a LilyPond como una parte del sistema, puede utilizarse para controlar los detalles tipográficos de todos los elementos de la partitura.

²⁴La información de los cursos de Valencia de Don Juan se ofrece en el sitio oficial cursovalenciadedonjuan.com

²⁵El sitio oficial del proyecto Abjad es pypi.python.org/pypi/Abjad

Los compositores que deseen emplear el enfoque algorítmico para sus creaciones tienen en Abjad una potente herramienta. Abjad es software libre licenciado bajo la GPL.

5.13.5. MEAWS, de Graham Percival

El violoncelista canadiense Graham Percival era el editor de la documentación oficial de LilyPond hasta el momento de su salida del proyecto.²⁶ Para su tesis de Máster (Percival, 2008), publicada en en la Universidad de Victoria (Columbia Británica, Canadá) en agosto de 2008 alrededor de un tema multidisciplinar que abarca la música y la informática, preparó MEAWS,²⁷ un programa de software libre licenciado bajo la GPL, que está orientado a servir de ayuda para músicos principiantes y estudiantes de música.

El programa presenta ejemplos de música en la pantalla y registra la interpretación que el usuario hace de este fragmento, comparando esta interpretación con el modelo teórico correspondiente y emitiendo una puntuación como un porcentaje en función del grado de exactitud con que se ha interpretado el fragmento. Existen dos modos de práctica: ritmos (para toda clase de músicos) y afinación del violín. Los ejercicios son de dificultad progresiva y son siempre diferentes. La generación de los fragmentos propuestos se realiza internamente mediante el lenguaje *Strasheela* de composición algorítmica; los ejemplos de música en pantalla están producidos a partir de estas secuencias por el programa LilyPond.

5.13.6. La tesis de Luca Rossetto

Luca Rossetto Casel, informó en 2013 a los miembros de la comunidad de usuarios y desarrolladores de la finalización de su tesis de doctorado hecha con LilyPond (Rossetto, 2013): una edición crítica, en cinco volúmenes, de *Enea nel Lazio*, ópera seria en tres actos compuesta por Tommaso Traetta en 1760, con libreto de Vittorio Amedeo Cigna-Santi, dando las gracias a los desarrolladores de LilyPond.

Luca es uno de los miembros italianos del equipo de traductores del proyecto.

5.13.7. LilyPond para personas con discapacidad visual

Se mencionó en el apartado de interfaces de texto que los ciegos y discapacitados visuales pueden utilizar LilyPond y otros programas de consola mediante adaptaciones basadas en lectores de pantalla y en líneas táctiles de Braille. Si un programa está orientado exclusivamente al manejo mediante el ratón, resulta difícil la adaptación porque el enfoque es completamente visual; en cambio, los documentos de texto se pueden copiar, transmitir y convertir entre plataformas de accesibilidad. La usuaria y programadora Sarah K. Alawami anunció en

²⁶La última intervención de Graham es de agosto de 2012.

²⁷El software MEAWS se aloja en el sitio personal percival-music.ca/software/meaws. Del inglés Musician Evaluation and AWdition for Strings

un foro especializado que utilizaba LilyPond en un MacBook Air y en Windows con el lector de pantalla JAWS, que permite un acceso no visual al escritorio gracias a la conversión de texto a voz. El pianista chino Hu Haipeng, ciego de nacimiento, completó sus estudios de composición usando LilyPond con una línea táctil de Braille. La figura 30 (página siguiente) muestra un fragmento de su obra sinfónica *Violent Dance for Orchestra*.

The image displays a page of a musical score titled "Violent Dance For Orchestra" by Hu Haipeng. The score is written for a full orchestra and includes parts for Piccolo, Flutes I & II, Oboes I & II, Clarinets I & II in B \flat , Bassoons I & II, Horns I & II in F, Horns III & IV in F, Trumpets I & II in B \flat , Trumpet III in B \flat , Trombones I & II, Bass trombone & Tuba, Timpani in A, D & E, Triangle, Suspended cymbal, Tamtam, Tambourine, Snare drum, Bass drum, Harp, Violin I, Violin II, Viola, Violoncello, and Contrabass. The tempo is marked "Presto ($\text{♩} = 112$)". The score shows various dynamics such as *p*, *mp*, *mf*, *f*, *ff*, and *pp*, and includes performance instructions like "Molto cresc." and "Sul ponticello".

Figura 30: Fragmento de *Violent Dance for orchestra*, del compositor chino, ciego de nacimiento, Hu Haipeng.

En español, el capítulo 3 “LilyPond, software accesible para la edición de partituras por parte de músicos ciegos” del libro *Música, inclusión y discapacidad visual. Guía de Orientaciones Didácticas* del Ministerio de Cultura de Colombia está dedicado a:

(...) algunos aspectos de su funcionamiento, utilidades, ventajas, y especialmente recomendaciones para un abordaje exitoso, pedagógicamente hablando, de esta herramienta por parte de las personas ciegas (Lozano, Bello, Cancino y Martínez, 2013)

Uno de los autores de la citada guía, Juan Pablo Bello, es el editor de las actas de la Conferencia Internacional sobre recuperación de la información musical (Bello, Chew y Turnbull, 2008) y ha dado cuenta del uso de LilyPond con el lector de pantalla JAWS, que es factible gracias a que puede emplearse cualquier programa editor para la composición de los documentos. El mayor problema al que se enfrentan los usuarios con discapacidad visual es el balance entre llaves de apertura y de cierre que se emplean para delimitar las expresiones musicales. En documentos complejos se producen varios niveles de anidamiento entre expresiones, es decir, llaves que se abren y que después deben cerrarse inexcusablemente, pues el programa intérprete que analiza el texto posee muy poca tolerancia a cualquier falta de rigor léxico o sintáctico en el código.

Otra dificultad proviene del hecho de que, en ocasiones, no hay una relación directa entre el mensaje de error, obtenido a partir de la información de registro (conocida como *log*) y el lugar del documento en que reside el fallo. Las rutinas analizadoras del programa LilyPond comenzaron a permitir un cierto grado de recuperación de los errores léxicos en el año 2000 a partir de la versión 1.3, emitiendo el mensaje “*cruce los dedos*” cuando se produce alguna inconsistencia. Los entornos amigables como Frescobaldi facilitan esta tarea de reconocimiento de errores léxicos del código fuente, aunque lamentablemente no es posible utilizarlo con el lector de pantalla JAWS.

5.14. El clarinete robótico del equipo de Mark Sheahan y Peter Chubb

En junio de 2008 (Sheahan y Chubb, 2009), un equipo mixto de la Universidad de Nueva Gales del Sur (Australia) y el NICTA²⁸ fue el vencedor del desafío *Artemis Orchestra Challenge*, que consistía en que cada equipo participante tenía que hacer tocar automáticamente, con un instrumento musical no modificado, fragmentos musicales proporcionados en forma de código de LilyPond y situados en un servidor de archivos externo. A pesar de las muchas dificultades para conseguir que un robot sea capaz de tocar el clarinete, lo interesante del caso en lo que respecta a nuestro estudio es cómo los organizadores emplearon código nativo de LilyPond para determinar la música que había que interpretar, y que cada robot, de forma autónoma, debía procesar este código con su propia copia interna de LilyPond para obtener la secuencia de notas del fragmento.

²⁸El NICTA es el Centro de Excelencia para investigación en TIC del gobierno de Australia.

El equipo ganador utilizó código adicional para aplicar los efectos de *non legato*, *staccato*, trinos y otras articulaciones al resultado de procesar la partitura y conseguir así que el robot ejecutase los fragmentos de forma más expresiva, o dicho de otra forma, menos mecánica.

También es interesante que este código se incorporase al poco tiempo dentro del propio programa LilyPond bajo la denominación de “Articulate”, en forma de extensión del programa. Se trata de un buen ejemplo de cómo el proyecto se beneficia de las contribuciones de desarrolladores externos al equipo. La extensión es capaz de ofrecer una salida MIDI que contiene los efectos añadidos de articulación correspondientes a ciertas indicaciones abreviadas de la partitura. En otras palabras, además de leer la partitura en su forma básica (melodías formadas por notas con su altura y su ritmo), traduce las indicaciones expresivas, expande las abreviaturas y las interpreta acercándose a la manera en que lo haría un músico humano.

En la figura 31 se aprecia el antes y el después de la aplicación de la extensión “Articulate”.



Figura 31: Ejemplo tomado de Sheahan y Chubb (2009) que muestra (a la izquierda) un fragmento de música de LilyPond y (a la derecha) el efecto de aplicar la extensión “Articulate” a dicho fragmento.

En los dos casos se trata de la misma música, pero el fragmento de la izquierda es mucho más legible, mientras que el de la derecha es más literal al presentar las abreviaturas totalmente expandidas. Cuando el fragmento convertido se reproduce como MIDI, suenan correctamente las notas *non legato*, el *ritardando* y las notas de adorno. El ejemplo ilustra el modo en que la notación musical contiene convenciones y abreviaturas que se pueden transformar en otros símbolos existentes dentro de la propia notación, aunque al hacerlo así se complica mucho el texto y se dificulta la lectura. Sin embargo, la versión que Articulate genera para el robot suena mejor porque la interpretación mecánica de la notación musical suele ignorar dichas abreviaturas, que tienen una importante función expresiva semejante a los acentos, la entonación y las pausas de la palabra hablada.

5.15. Otras aplicaciones

Varias propiedades de LilyPond sugieren que puede ser una herramienta valiosa para la transcripción del patrimonio cultural y de tradición oral. Se han citado en este sentido las coleccio-

nes de Victoria, Guerrero y Morales (Nancho Álvarez), cancioneros tradicionales como el de Allen y col., óperas y ballets barrocos (Nicolas Sceaux), Villancicos extremeños (Daniel López y Emilio Díaz) y otras muchas.

El carácter abierto del formato de documento, que es texto sencillo, hace posible la discusión del código como si fuera lenguaje natural en prosa. Los detalles de composición tipográfica o de formato del documento se pueden copiar y pegar, por ejemplo, en mensajes de correo electrónico directamente a partir del editor empleado para la confección de la partitura.

La misma orientación a texto sencillo como única fuente para la preparación de la partitura, es un terreno ideal para su gestión mediante un sistema de control de versiones como Git. En el lugar oportuno de los estudios del presente trabajo se detalla el funcionamiento de este sistema, pero para los efectos de lo que aquí se habla diremos que el uso de este método tiene las siguientes implicaciones:

- El registro está distribuido y no depende de un solo servidor, aunque la institución promotora o encargada puede denominar a un servidor determinado como canónico.
- Los investigadores pueden en cualquier momento clonar el repositorio central, a partir del lugar canónico o de otro repositorio que es un clon a su vez.
- La existencia de un mecanismo de ramificación facilita el mantenimiento de ediciones paralelas, con distinto formato o variaciones experimentales.
- Los documentos de texto con las notas críticas pueden incluir las partituras o fragmentos de ellas, que van siempre *vinculadas* (no insertadas).
- La existencia de una historia completa del código, transparente para todos, permite operaciones avanzadas como “la vuelta atrás en el tiempo” y copias de seguridad automáticas.
- Los investigadores pueden trabajar sobre sus repositorios clonados de forma completamente independiente al servidor canónico, y pueden incorporar sus ediciones a éste, o las de otros editores a la suya.
- La producción de la colección completa de originales a partir de una instantánea del código existente, y su publicación electrónica en PDF y/o en la Web, pueden hacerse de forma automática y periódicamente.
- El profundo nivel de configuración que es posible realizar sobre la música es un factor de flexibilidad.
- La alta calidad del documento final ofrece profesionalidad y utilidad práctica en el producto terminado.

Todo ello se deriva del empleo de documentos de texto y del control de versiones. Por otro lado están las libertades que concede al usuario la licencia del programa y de las que se ha hablado extensamente. Solamente cuando se toman en su conjunto el carácter abierto del documento y el del programa, puede hablarse de Patrimonio a disposición de la ciudadanía.

Si el programa no fuera libre, la edición o la impresión del material estarían supeditadas a la compra de un producto privativo, o a su obtención por cualquier otro medio, lo que produce un situación contradictoria.

En pocas palabras, el patrimonio musical solo está verdaderamente a nuestra disposición si tanto los documentos fuente como el programa para compilarlos, son libres en el sentido expuesto.

En la Comunidad Autónoma de Extremadura hay un caso que reclama una pronta acción por parte de los músicos de la región y de fuera de ella. Se trata de la obra compositiva del gran pianista Esteban Sánchez (1934-1997), natural de Orellana la Vieja. No es el lugar de repasar su biografía ni sus logros como pianista de renombre internacional. Lo que llama la atención para los efectos de nuestra investigación, es que estas obras no estén aún editadas.

Varios intérpretes de alto nivel han acometido la interpretación y el registro discográfico de sus composiciones, que son obra obligada para los concursantes del premio «Esteban Sánchez» de piano, a pesar de que las piezas no están editadas aún, sino que existen solamente como manuscritos.

El compositor Esteban Sánchez no estaba registrado en la Sociedad General de Autores y Editores. Su familia es la única depositaria de los derechos morales de la obra, y nunca han mostrado inconveniente en permitir el uso de este material. Pensamos que una iniciativa seria, en el sentido de fijar y divulgar estas piezas, es un hecho que hace tiempo debiera haberse producido.

Aunque ya existen como ediciones en formato de libro, en la misma línea están pidiendo una edición digital el Cancionero Popular de Extremadura de Bonifacio Gil 1961 y otros. Es evidente que lo dicho se aplica también al material musical de todo tipo y de toda época que se está recuperando pacientemente por parte de los investigadores y musicólogos de todo el mundo. La disponibilidad del programa LilyPond, su carácter libre, su gratuidad, su adecuación a la edición cooperativa y el alto nivel estético del resultado, haría que estos esfuerzos tuvieran grandes alicientes y una serie de recompensas de alto valor y larga proyección.

5.15.1. El premio LOMUS

Incluimos aquí esta reseña aunque se salga en cierta medida del epígrafe “Aplicaciones de LilyPond” por falta de un apartado más adecuado.

La AFIM (*Association Française d’Informatique Musicale*) convoca anualmente una competición internacional de software musical libre. Esta asociación se orienta al desarrollo de la música por ordenador en Francia, su relaciones con otras disciplinas artísticas y científicas

y sus vínculos internacionales. Organiza las jornadas de Informática Musical JIM y participa en la comisión del grupo *Sound & Music Computing* (Computación de sonido y música). La AFIM está abierta a desarrolladores de alguna actividad de investigación, creación, análisis o pedagogía musical relacionadas con los ordenadores y las nuevas tecnologías.

El premio Lomus de la AFIM se otorga a proyectos que se aventuran en el desarrollo de software libre musical o software gratuito que pueda contribuir al proceso de creación musical, y que demuestran no solo innovación, sino sobre todo creatividad e inventiva en la creación musical. En la edición de 2012²⁹ resultó ganador el proyecto LilyPond.

5.15.2. Desarrollos personalizados

Uno de los resultados del régimen de licencias abiertas de este tipo de proyectos es que se puede adaptar a las necesidades específicas de una aplicación, sin importar lo minoritaria que ésta sea (siempre que no determine la suerte del proyecto en su conjunto), lo cual no implica necesariamente que sea fácil de conseguir.

Una vez superado el umbral técnico, ningún interés económico o legal se opone a estos desarrollos, que revierten en la comunidad y animan a nuevos autores a efectuar aportaciones. Tres ejemplos en el caso de LilyPond son los siguientes:

- La importación y exportación del estándar MusicXML, iniciada en 2002 pero desarrollada por Reinhold Kainhofer a partir de 2007.
- Las traducciones del programa y su documentación, que los traductores suelen emplear para su propio uso, o en la docencia. Las traducciones fueron iniciadas en 2007.
- El soporte para *canto kievano* (un tipo de monodia litúrgica de la Iglesia Ortodoxa) en notación cuadrada, obra de Aleksandr Andreev y comenzada en 2011. Véase la figura 32.



Figura 32: Ejemplo de canto kievano en notación cuadrada.

²⁹El sitio oficial de esta edición del premio es concours.afim-asso.org/2012

Estudio 1
Dinámica del software libre de edición de partituras

6. Estudio 1. Objetivos e hipótesis

En el marco teórico se enunció el problema de la tipografía. LilyPond es un programa de ordenador representativo del software libre de edición de partituras musicales, que constituye una potencial solución a este problema, especialmente en educación musical aunque sin limitarse a este ámbito.

Objetivos principales

El primer objetivo del presente estudio es contribuir al desarrollo de esta solución, examinando la evolución de sus aspectos más relevantes y generando confianza en el proyecto para apoyar, con argumentos sólidos, su uso de forma extensiva en educación. Dicho de otro modo, se persigue la generación de buenas expectativas de futuro depositadas en este tipo de herramientas.

El segundo objetivo es arrojar luz sobre un ente inmaterial como es el software, usando como caso de estudio un proyecto abierto de tipografía musical.

Es, por tanto, un trabajo descriptivo.

Objetivos secundarios

Los objetivos secundarios son:

- Contribuir a la creación de nuevas técnicas de análisis y visualización del software en sus aspectos estático y dinámico, y a la mejora de las técnicas existentes.
- Comprender la evolución de su ciclo de vida.

El punto de vista estático consiste en analizar el código fuente del programa. A través de este análisis, otros objetivos secundarios son los siguientes:

- Describir la composición del equipo de desarrolladores.
- Catalogar los lenguajes de programación empleados y clasificar los elementos constituyentes por la función que desempeñan dentro del sistema.
- Aplicar métricas estándar de la ciencia informática al árbol de archivos.
- Relacionar la intensidad de los desarrolladores individuales con la frecuencia por desarrollador de su actividad. Caracterizar esta distribución.
- Identificar las fuentes de error del análisis estático.

El punto de vista dinámico describe la evolución en el tiempo, o historia, del código fuente del programa y de su documentación. Un análisis dinámico persigue los siguientes objetivos:

- Desde un punto de vista general, estudiar la viabilidad de un futuro para la tipografía musical con herramientas libres.
- Mostrar la evolución en la composición del conjunto de autores.
- Presentar frente al tiempo determinadas variables clave del código, elaborando una serie temporal con estos datos.
- Construir un modelo de la serie temporal. Aplicar diagnósticos de validez al modelo para evaluarlo.
- Utilizar el modelo para extrapolar las contribuciones pasadas y así estimar su estabilidad a medio/largo plazo.
- Visualizar dinámicamente la historia de la actividad de los desarrolladores.
- Documentar las técnicas de análisis y visualización empleadas, de manera que puedan aplicarse después a otros proyectos de software libre.
- Identificar las fuentes de error del análisis dinámico.

Hipótesis

1. Los análisis desde los puntos de vista estático y dinámico de un repositorio de código fuente de software libre pueden revelar aspectos de interés previamente ocultos acerca del régimen de intervenciones que presenta este código, siempre y cuando su historia se halle registrada en un sistema de control de versiones público.
2. El pronóstico de los diversos indicadores del programa LilyPond, extrapolado a partir del análisis dinámico de la historia de este proyecto de software libre de tipografía musical, es ascendente a medio plazo.
3. Se puede efectuar un análisis dinámico sobre las partituras musicales, a condición de que estas partituras estén codificadas en forma de documentos de texto, y su historial de cambios esté también registrado y disponible.
4. Se estima que la actividad de desarrollo en el caso de LilyPond, como un indicador del grado de compromiso de la comunidad con el mantenimiento y la mejora de dicha herramienta, tiene un futuro que presenta una evolución sostenida o en auge.
5. El caso de estudio considerado, el proyecto LilyPond, presenta robustez frente a las variaciones en la composición de su equipo de desarrollo.

7. Estudio 1. Metodología

7.1. Método

El estudio es descriptivo mediante observación, evolutivo y secuencial, según la clasificación de Montero y León (2007). Ello se explica porque la variable independiente es el tiempo y porque se estudia un conjunto de variables dependientes a lo largo del mismo.

Es de carácter natural ya que no se producen intervenciones del investigador sobre el objeto observado durante el tiempo que dura la investigación.

De acuerdo con el objetivo de analizar la naturaleza de un sistema, sus componentes, sus funciones, sus procesos, sus interacciones y sus resultados, y por las características de sus métodos y técnicas, puede decirse que el estudio tiene un enfoque sistémico.

En función de esos mismos objetivos, el método se caracteriza como descriptivo. Persigue describir la estructura y propiedades del elemento de software conocido como LilyPond desde un punto de vista estático, así como los fenómenos que se producen en su evolución, es decir, bajo un punto de vista dinámico. Se persigue la identificación de los aspectos relevantes de este caso de estudio. Para ello se utilizan técnicas cuantitativas mediante las que se analiza un conjunto de variables propias de las aportaciones al código fuente del programa, aplicando métricas estándar a dichas aportaciones, y se presentan los datos frente al tiempo. Se forma una serie temporal y se elabora un modelo de ella que permite descomponerla y pronosticar su evolución.

Otro de los objetivos es una visualización expresiva de la tendencia del proyecto y su documentación. Si bien se han llegado a hacer estadísticas sencillas sobre repositorios de software basados en otros sistemas de control de versiones, no hemos encontrado uno que efectúe este tipo de análisis en función del tipo de archivo, y estas herramientas no están en modo alguno orientadas a un uso académico de las estadísticas, sino a la actualización continua de datos para la web. Debido a ello, el tipo de información gráfica que presentan es, en primer lugar, de baja resolución, no apto para su impresión en papel. En segundo lugar, no entran en ponderar el contenido de las intervenciones, sino que son neutrales respecto de la orientación de las mismas; se limitan a efectuar un recuento. Esto significa que, por ejemplo, si se quiere evaluar el peso relativo de las intervenciones orientadas al desarrollo de la documentación frente a aquellas que afectan al programa en sí (al margen de su documentación), no es

posible hacerlo con las herramientas existentes. Por ello, es necesario desarrollar un conjunto nuevo de herramientas que puedan cumplir estas funciones. Se plantea como una de las premisas fundamentales de todo el proceso de recogida de datos, análisis estadístico, generación de los gráficos de resultados y redacción de la memoria, que se utilice exclusivamente software libre para todas las tareas referidas.

7.2. Variables

Distinguimos entre variables dependientes e independientes. Las variables dependientes de este estudio son:

- La sucesión de estados instantáneos (conocida como “historia del proyecto”) del código fuente del proyecto LilyPond, como sistema de tipografía musical perteneciente a la categoría de software libre.
- Dentro de los estados del objeto en el tiempo, analizamos cuantitativamente los siguientes elementos:
 1. La documentación emitida, con la fecha de su emisión y otros datos como el nombre y la versión a que corresponde.
 2. Los archivos de instalación emitidos, asimismo con la fecha, el nombre, la versión y el tamaño.
 3. Las intervenciones efectuadas al código fuente (programa y documentación).
 4. Caracterizaciones adicionales sobre las intervenciones. Entre ellos, la ruta (desde el punto de vista informático) de los archivos a los que dichas intervenciones afectan, para determinar si pertenecen a la documentación.

Según algunos criterios metodológicos, la ruta es un “campo abierto” que podría considerarse cualitativo. Se extraen los siguientes datos adicionales a partir del grueso de las intervenciones:

- La identidad del autor de cada una de ellas, que posteriormente se debe hacer corresponder con una persona física.
- La fecha y la hora en que se efectúa la intervención.
- A partir del dato de fecha absoluta anterior, los tres siguientes que se analizan por separado:
 - el mes del año,
 - el día de la semana,
 - la hora del día.

- La zona horaria del lugar en que se produjo.
- El número de líneas de código modificadas.

A partir de estas variables se calculan los parámetros fundamentales de la evolución del proyecto en el tiempo, que son:

- La intensidad de la actividad en cada momento, que permite identificar las fases por las que atraviesa el proyecto.
- El tamaño y la composición del equipo de desarrolladores en cada momento, a través del recuento del número de autores por mes. Para la depuración de estos datos y con el objetivo de no sobreestimar el número de autores, es necesario identificar a aquellos que presentan más de una identidad.
- El número variable de archivos.
- El número variable de líneas de código.

La variable independiente del estudio es el tiempo; esta variable no se puede manipular.

La variable independiente se extiende a partir del inicio mismo del proyecto hasta los últimos datos disponibles en el momento del cierre de la investigación; damos cuenta de ello en el apartado siguiente donde se especifica la naturaleza de la muestra.

7.3. Muestra

El proyecto LilyPond se inició en 1995 y la historia registrada del código fuente comienza en octubre de 1996. En octubre de 2014 se cumplieron 18 años de historia registrada, y en abril de 2015 el proyecto cumplió 20 años.

Este estudio cuantitativo del proyecto LilyPond considera el período de tiempo que abarca desde el comienzo de la historia registrada del proyecto en octubre de 1996 hasta junio de 2015. Por su interés histórico se efectúa el análisis de todas las contribuciones realizadas al proyecto desde sus inicios. Para extraer una cierta medida de la *salud* o las *garantías de futuro* que el proyecto tiene en un momento determinado, se efectúa un pronóstico gracias a la aplicación de un modelo que tenga en cuenta todo el historial de intervenciones.

Así pues, no se ha efectuado ningún muestreo en el sentido usual de la palabra, sino que se emplea la población completa, aunque se agregan los datos correspondientes a períodos semanales, mensuales y anuales. El resultado es que se analizan todas las intervenciones registradas en la historia dentro de este período y se obtienen todos los datos que proporciona el instrumento para cada una de las intervenciones. El único dato que no se analiza es el texto de la intervención, aunque se examina el contenido de cada una para determinar su tipo.

7.3.1. Naturaleza de los datos

La principal fuente de los datos es una fuente primaria: el historial de versiones del código fuente del proyecto objeto del estudio, y de su documentación. Por “historial de versiones” debe entenderse el registro documental de todas y cada una de las modificaciones que los desarrolladores han efectuado alguna vez al código, agrupadas en objetos llamados “intervenciones” y que se cuentan por más de 26.000, efectuadas durante el período aproximado de 18 años que se ha acotado al hablar de la muestra. Cada intervención registrada contiene el detalle pormenorizado de todas y cada una de las diferencias producidas en cada archivo sobre el que interviene. El código fuente y su historial de versiones completo está disponible en la forma de un *repositorio Git* tal y como se ha descrito en la fundamentación teórica. Este repositorio es de libre acceso.

El código fuente ofrece datos objetivos sobre el programa. Estos datos están disponibles y se puede acceder a ellos porque el programa está publicado bajo una licencia libre. Las licencias libres, y específicamente aquellas que son compatibles con la Licencia General Pública de GNU, conocida como GPL, mencionan explícitamente la obligación del licenciataria de publicar el código fuente junto al programa.

7.4. Puntos de interés

El esquema sinóptico de la figura 33 presenta una panorámica de los aspectos relevantes de un proyecto de software libre, ejemplificado en el caso de LilyPond. Los elementos que se estudian en el trabajo están destacados mediante un recuadro, y son: el núcleo de programadores de la comunidad de desarrolladores; el código fuente, tanto del programa como de la documentación; los archivos compilados del programa y de su documentación; las diversas aplicaciones del proyecto en educación, transcripción del Patrimonio y otras; los entornos de desarrollo existentes; y por último las plataformas que utilizan LilyPond como servicio web para la producción remota (*online*) de partituras. Algunos de estos puntos han sido analizados ya en los apartados de fundamentación teórica.

7.5. Fuente primaria

De todas las ramas en que se divide la historia, se extraen los datos de la rama maestra o «*master*». La rama maestra es aquella a partir de la cual se publica el programa, pero cualquier otra rama tiene aproximadamente el mismo número de intervenciones que la rama maestra, en la medida en que se producen fusiones frecuentes entre ellas, lo que tiene el efecto de unificar sus historias en una sola. Las fusiones se producen en los dos sentidos: por ejemplo,

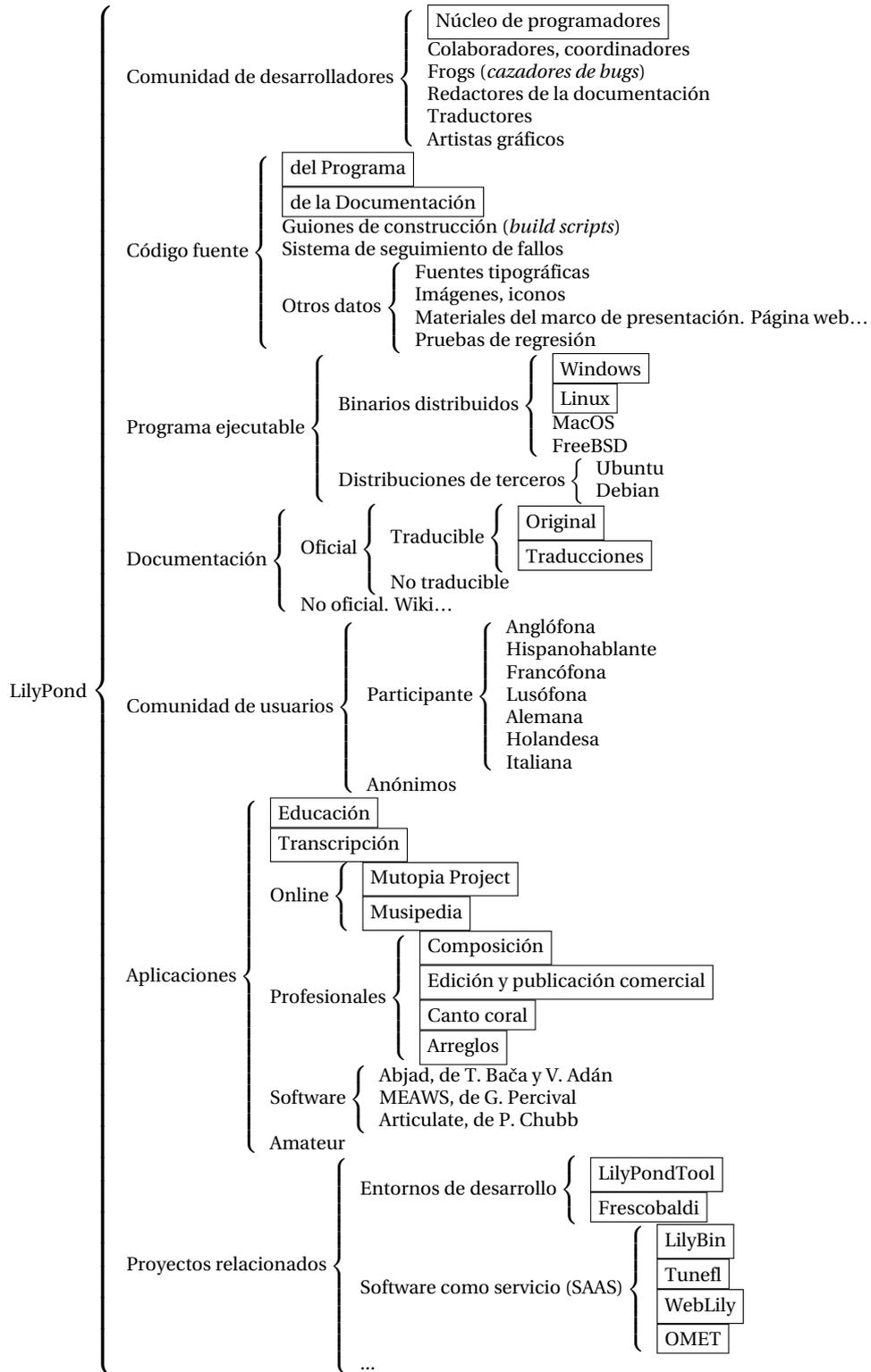


Figura 33: Elementos más relevantes del proyecto LilyPond. Están recuadrados los aspectos que centran la atención de este estudio.

considérese la rama de traducciones. Su evolución es paralela e independiente de la rama maestra, pero para incorporar a ésta las traducciones, un desarrollador efectúa la fusión de la rama de traducciones sobre la rama maestra. Mientras tanto, es posible que algunas partes de la documentación original hayan cambiado en la rama maestra. Entonces, el desarrollador al cuidado de la rama de traducciones produce la fusión de la rama maestra sobre la de traducciones. Así, después de dos fusiones o *mezclas* mutuas, las dos ramas implicadas contienen el registro de todo lo que ha ocurrido en cualquiera de ellas.

7.6. Disponibilidad de los datos

Como se ha descrito anteriormente, la fuente de todos los datos de la historia de LilyPond está gestionada mediante el sistema Git de control de versiones, que es descentralizado, aunque existe una fuente canónica de referencia: el repositorio oficial.¹ Al ser Git un sistema descentralizado, existen copias *clonadas* en multitud de sistemas por toda la red.²

El documento siempre ha sido público, aunque su forma de repositorio Git no es anterior a 2002. El primer sistema utilizado para gestionar el código fue CVS, cuya historia se convirtió a Git posteriormente. Esta conversión es una fuente potencial de error que depende de las variables que se vayan a analizar y en general son causa de pérdida de datos, pero no afectan a nuestro análisis. Nos cuenta uno de los autores principales, Jan Nieuwenhuizen (Nieuwenhuizen, 2009):

Durante los primeros seis años estuvimos usando lanzamientos en forma de archivos comprimidos y parches. El usuario “Fred” que aparece en los registros correspondientes a este período era yo, cuando poblé la historia de nuestro sistema de control de versiones utilizando estos archivos comprimidos y parches, mediante un *script* informal.³

Solíamos tener una rutina en la que Han-Wen publicaba un *tarball*,⁴ al que yo le hacía algunos cambios. Lo comprimía de nuevo y se lo enviaba de vuelta a él. Entonces él incorporaba mis cambios, hacía los suyos y volvía a emitir un paquete, que me enviaba de nuevo a mí.

(...) No todos los parches intermedios guardaron un buen registro de cambios.

A pesar de ello se incluye este período porque el recuento de autores es mínimo fuera de los dos desarrolladores principales.

¹La dirección de la fuente oficial es `git.sv.gnu.org/r/lilypond.git`.

²Se ha dado el caso (junio de 2009) de que el servidor Savannah en que se alojan los proyectos de GNU, se ha caído por un fallo en el disco duro antes de disponer de una copia de seguridad lo suficientemente reciente, y la información se ha restituido a partir de las copias clonadas de los desarrolladores que las tenían en sus sistemas como árboles de trabajo.

³El adjetivo «informal» corresponde en el original inglés a la palabra *cheesy*.

⁴Un *tarball* es semejante a un archivo comprimido Zip.

Según consta en los registros, Jeffrey Reed fue el primer colaborador del proyecto LilyPond distinto a los autores originales, estando su primera intervención fechada en septiembre de 1998. Ignoramos la fecha exacta en que el repositorio Git se hizo público como tal en el servidor Savannah de la organización FSF,⁵ que aloja todos los desarrollos del proyecto GNU.

7.7. Instrumento

7.7.1. Software empleado

Esta relación de software contiene los programas empleados para la investigación, el análisis de los datos y las visualizaciones.

- Python, de Guido van Rossum, es un lenguaje de programación. Se ha utilizado para realizar el recorrido de la historia del código y para preparar las tablas de datos para los gráficos. La documentación de Python es muy completa (van Rossum, 1991).
- AWK, de Alfred Aho, Peter Weinberger y Brian Kernighan, es un procesador de datos basados en texto y un lenguaje de programación que utiliza listas asociativas y expresiones regulares. Se ha empleado la variante de GNU conocida como GAWK para filtrar y manejar las columnas de los datos en bruto. La documentación es obra de Close, Robbins, Rubin y Stallman (1989).
- Git, de Linus Torvalds, es un sistema de control de versiones para código fuente, que se puede emplear para objetivos tan dispares como el desarrollo colaborativo de grandes proyectos de software o el control de archivos de configuración. Es el instrumento principal que se ha usado para analizar el código fuente de LilyPond, pues es la herramienta con la que se gestiona la historia del mismo. La documentación correspondiente está aportada por Greaves y col. (2010).
- GitPython, de Michael Trier y Sebastian Thiel, amplía el lenguaje Python para interactuar con repositorios Git (Trier y Thiel, 2009). Aporta funciones que se pueden utilizar en *scripts* que recorren el árbol y desglosan las diferencias de cada intervención. GitPython, que es software libre publicado bajo la licencia BSD, está basado en un trabajo anterior de Tom Preston-Werner y Chris Wanstrath, llamado `grit_library` y escrito en lenguaje Ruby. Se emplea GitPython para acceder a la historia del código mediante guiones propios en Python elaborados por el autor.
- Un intérprete del lenguaje PHP (Lerdorf, Tatroe y MacIntyre, 2006) se ha utilizado para ejecutar guiones locales que dialogan con los servidores oficiales, en los que se alojan los archivos de instalación y la documentación. PHP es un lenguaje especializado en el procesamiento de código HTML para la web, que también puede usarse para programas

⁵La *Free Software Foundation* (FSF) es la fundación para el software libre fundada por Richard Stallman en 1985.

de uso general. La licencia de PHP es libre, pero no compatible con la licencia GPL y sin copyleft. PHP fue iniciado por Rasmus Lerdorf en 1995.

- Para el recuento, modelización, evaluación del modelo y para la producción final de los gráficos se empleó el programa R (R-CDT), una aplicación de consola que es un lenguaje para cálculos estadísticos. Para el desarrollo de los guiones de R se empleó el entorno de desarrollo RStudio.
- Una versión modificada de Code_swarm, de Michael Ogawa, se utilizó para elaborar el primer vídeo, que muestra dinámicamente la historia de versiones.
- Una versión modificada del programa Gource, de Andrew Caudwell, se utilizó para elaborar el segundo vídeo, que emplea un enfoque centrado en el árbol de directorios para mostrar la historia en forma de animación.
- Las utilidades libav se han usado para la codificación de los vídeos a partir de los cuadros de imagen individuales que producen los programas Code_swarm y Gource.
- Se empleó la utilidad OpenShot para la concatenación de las secuencias de vídeo y la aplicación de títulos.
- Las herramientas estándar de UNIX en su versión GNU nos han servido para aglutinar las distintas funciones de consola a través de la línea de órdenes del terminal.

7.8. Procedimiento

Para el análisis del código fuente se utiliza la propia herramienta Git que se ha descrito arriba; programas preparados *ex profeso* por el autor en los lenguajes Python y AWK, combinados con instrucciones estándar de UNIX, recorren el código fuente, examinan las intervenciones una a una mediante expresiones regulares y preparan archivos de texto con los resultados. Otro programa elaborado por el autor en lenguaje R produce los gráficos.

Los datos que se obtienen son objetivos y no requieren validación, aunque deben examinarse las posibles fuentes de error.

7.8.1. Obtención de datos

Los datos se clonan de una de las copias descentralizadas de la fuente primaria. El clonado de una historia de Git es una operación básica y fundamental del programa. Al mismo tiempo se comprueba que la copia está actualizada; con la misma herramienta se incorporan las últimas modificaciones en caso de que se hayan producido. Se emplean instrucciones de consola de UNIX y unos cortos programas o guiones que recorren el árbol de versiones e indexan los datos de cada una.

Para examinar la distribución de la carga de trabajo en el proyecto LilyPond, se siguen los siguientes pasos:

- Extraer los datos de contribuciones de los autores.
- Depurar la lista de personas físicas a partir de las identidades registradas de autores.
- Hallar el número de autores distintos (personas físicas).
- Contar las contribuciones de cada autor.
- Producir una lista de autores numerada y ordenada por número de contribuciones, o *ranking*.
- Presentar el número de contribuciones frente al ranking de los autores, en escala logarítmica sobre los dos ejes.

Si la curva de las contribuciones frente al ranking presentada de esta forma es una línea recta, las contribuciones por autor siguen una ley potencial también conocida como *ley de Zipf* (Newman, 2005). Si la curva es cóncava hacia arriba, la carga de trabajo está más concentrada en los desarrolladores principales de lo que predice la ley de Zipf. Si la curva es cóncava hacia abajo, el conjunto del equipo está más cohesionado y la carga de trabajo está más distribuida.

Dentro del análisis dinámico, para la aplicación del procedimiento de media móvil es necesario determinar el tamaño del lapso de tiempo o “ventana” durante la cual es válido el recuento de la carga de trabajo, ya que los desarrolladores solamente están activos durante un determinado período de la historia.

7.8.2. Caracterización de las intervenciones

Aunque son posibles distintas caracterizaciones, en el estudio se efectúa solamente un tipo basado en la distinción entre las que modifican el código de la documentación, de aquellas que no lo modifican, y que llamaremos respectivamente “intervenciones de documentación” e “intervenciones de código puras”.

Una intervención contiene un conjunto de diferencias entre el estado (o “versión”) anterior y el actual. Estas diferencias afectan a uno o más archivos, que pueden estar en uno cualquiera de los directorios del árbol de código. Uno de estos directorios (una *rama* del árbol) aloja el código de la documentación, con subdirectorios para los distintos libros que la constituyen, y para los distintos idiomas a los que está traducida.

Cuando se recuperan los datos de una intervención, se desglosan las diferencias que contiene y se aplica un filtro a la ruta del archivo correspondiente a cada una de estas diferencias. El filtro es una expresión regular que discierne si en esta ruta se encuentra el nombre de la carpeta de documentación. En caso afirmativo, las coordenadas de fecha y tamaño de la interven-

ción se dirigen a un archivo; en caso negativo, a otro archivo distinto. Así pues, al finalizar el proceso se dispone de dos grandes listados de coordenadas, que se reúnen en un solo gráfico agregándolos por separado; de esta forma es posible observar cómo ha evolucionado la orientación del código fuente, atendiendo al criterio elegido.

En la figura 34 puede verse un ejemplo de intervención de código pura, efectuada por Graham Percival y fechada el 24 de septiembre de 2009. La primera línea comienza con la palabra clave `commit`, que a lo largo de todo este apartado se está traduciendo como «intervención». Le sigue un código único que diferencia a esta intervención de todas las demás. En este ejemplo, Graham modifica dos líneas de código de un archivo llamado `VERSION` de las que depende el número de versión del programa LilyPond. De la zona horaria que puede apreciarse al final de la línea de fecha, +0100, podemos determinar que Graham ha emitido esta intervención desde algún lugar de Europa central.

```
commit 675167c9e36f8e54cd8fadf2f2399ddfbe5ebdf8
Author: Graham Percival <gperciva@gperciva-desktop.(none)>
Date:   Thu Sep 24 13:54:29 2009 +0100

    Update VERSION.

diff --git a/VERSION b/VERSION
index 9e60b95..0342100 100644
--- a/VERSION
+++ b/VERSION
@@ -1,7 +1,7 @@
 PACKAGE_NAME=LilyPond
 MAJOR_VERSION=2
 MINOR_VERSION=13
-PATCH_LEVEL=4
+PATCH_LEVEL=5
 MY_PATCH_LEVEL=
 VERSION_STABLE=2.12.2
-VERSION_DEVEL=2.13.3
+VERSION_DEVEL=2.13.5
```

Figura 34: Ejemplo de una intervención de código pura, en la que un desarrollador modifica el número de versión del programa. Fuente: registro de versiones del código de LilyPond.

En el texto de la intervención aparecen el nombre y dirección del autor, la fecha de la intervención, un breve comentario explicativo y las diferencias atómicas efectuadas. Las diferencias aparecen como líneas que pueden pertenecer a uno de los tres grupos siguientes:

- Líneas de contexto, que son necesarias para hacer corresponder el texto de las diferencias con el código original y asegurar así que la diferencia se aplica a la línea correcta.
- Líneas eliminadas, que vienen indicadas por un signo menos.
- Líneas añadidas, que están precedidas de un signo más.

Una edición puntual produce una línea eliminada y otra añadida, correspondientes a la línea que se modifica en la intervención. Naturalmente, las adiciones netas de texto resultan de una diferencia positiva entre el código añadido y el eliminado, y de manera recíproca sucede

lo mismo con la eliminación neta. La creación o destrucción de archivos consiste solamente en líneas de un tipo: añadidas o eliminadas, respectivamente.

Este ejemplo es un caso mínimo en el que las modificaciones son puntuales, sobre un solo archivo; una intervención típica puede afectar a muchas líneas de un conjunto de archivos. La línea que contiene la ruta del archivo, en este caso `'/VERSION'`, es la que se analiza para la caracterización de las intervenciones en el sentido de que afecten o no a la documentación. Si la ruta contiene al directorio `Documentation/`, en que se hallan alojados los archivos del código fuente de los manuales del programa, entonces la intervención se caracteriza como una «intervención de documentación». En caso contrario, es una intervención de código pura como la de este ejemplo.

En principio, debiera ser posible analizar también la temática de las intervenciones de documentación, pues sabemos a qué libro y capítulo se dirige cada una. Los capítulos, si nos fijamos por ejemplo en el manual de referencia de la notación (Nienhuys y col., 2015b) tienen un nombre relacionado con su temática: Notación musical, Notación especializada, Problemas de espaciado... Sin embargo, existe un problema. En el transcurso de la reforma que la documentación sufrió entre mediados de 2007 y mediados de 2008, los capítulos y los propios manuales cambiaron varias veces de nombre y de directorio. Se reorganizaron bloques de texto que viajaron entre los capítulos. Se añadieron capítulos nuevos y se eliminó información obsoleta. En estas condiciones, el seguimiento de una determinada temática hubiera sido extremadamente complejo porque el repositorio Git es agnóstico respecto a la temática de las intervenciones. El criterio que se ha empleado, consistente en la determinación de si una intervención tocaba o no al directorio `Documentation/`, es válido porque en todo el período considerado el código fuente de la documentación en su conjunto no se movió de este directorio del árbol.

7.9. Fuentes de error

Se pueden identificar varios puntos en el procedimiento que son el origen potencial de errores, siendo el primero y más obvio la duplicidad de identidades que corresponden a una sola persona física.

La única forma de reducir las consecuencias de esta anomalía es elaborar una tabla de identidades reales y efectuar un recorrido semiautomático por las identificaciones registradas, sustituyendo cada vez las identidades dispares por las reales. El proceso es semiautomático porque por cada identidad real que se ha podido convertir, es posible extender la conversión al resto del registro.

La consecuencia de este error, si no se corrigiese, sería una sobreestimación de los tamaños

instantáneo y global del equipo de autores. Indirectamente, ello daría lugar a su vez a un cálculo erróneo de la carga de trabajo por autor.

Otra fuente de error, más difícil de reducir, es la sobreestimación de la carga de trabajo a partir de los datos de intervenciones puntuales por asumir que la tarea se reparte uniformemente, para cada autor, entre sus intervenciones.

Relacionado con la fuente anterior, e igualmente arduo de eliminar, es el hecho de que los cambios de diferente magnitud cuentan como uno solo. La consecuencia es que la métrica basada en el recuento simple de intervenciones no es una métrica óptima (aunque tiene la ventaja de que es sencilla de medir).

Algunos de los distintos tipos de intervención que se pueden dar, atendiendo a su función, son los siguientes:

- Soluciones a problemas de funcionalidad en el código.
- Adición de funcionalidades nuevas.
- Creación de un volumen, sección o apartado de documentación.
- Traducción de una sección o un apartado de la documentación original.
- Corrección de un error tipográfico o una falta de ortografía en la documentación o en alguna de sus traducciones.
- Modificación puntual de código de documentación *no traducible* (p.ej.: música o ilustraciones de otro tipo) y traslado inmediato a todas las traducciones.

La igualación del valor de estos tipos de intervenciones en uno solo, hace que sea imposible cuantificar el esfuerzo real que implican, en tanto no se encuentre una métrica adecuada para esta variable.

Otro error derivado es la equiparación del número de líneas con la carga de trabajo. Esto es especialmente cierto por la existencia de acciones masivas sobre muchas secciones del contenido al mismo tiempo, o reorganizaciones profundas del material.

Por último, la política de control de versiones puede no haberse mantenido inmóvil a lo largo de la historia del proyecto, o pueden haberse descuidado algunas versiones con más o menos actividad entre una y otra, lo que se conoce como “falta de granularidad” de la historia, que a su vez permite la existencia de una cierta cantidad de actividad oculta, difícil de cuantificar. La falta de granularidad se aprecia a escala microscópica (horas, días o semanas) pero tiene menos importancia a escala macroscópica, como meses o años. En general, puede ocurrir además que, dado un régimen de control de versiones, a partir de un determinado momento se utilice para su gestión un programa distinto, y si no se ha registrado absolutamente toda la actividad acaecida sobre la fuente desde el inicio del proyecto, algunos datos podrían haberse perdido en la transición.

En el caso de LilyPond, sabemos por los autores originales que:

- los primeros trabajos datan de abril de 1995;
- que no existen datos para la memoria de versiones, sobre CVS, hasta octubre de 1996;
- que este régimen duró aproximadamente seis años;
- y que posteriormente hubo una migración al sistema Git.

La relación de fuentes de error expuesta viene a indicar que, en el momento presente, estamos solamente arañando la superficie de las metodologías de investigación posibles en cuanto al análisis de la evolución del software.

7.10. Vídeos de la evolución

Para la visualización de la evolución del software, se aplican dos enfoques basados en los trabajos independientes de Michael Ogawa y Andrew Caudwell. Los trabajos de Ogawa se presentaron en 2010 en el Simposio Internacional sobre visualización del software (Ogawa y Ma, 2010); el trabajo de Caudwell aparece en 2012 en la Conferencia Internacional de la ACM sobre sistemas, lenguajes y aplicaciones de la programación orientada a objetos (Caudwell, 2010).

Los dos estudios de visualización realizados tratan de ilustrar cómo el trabajo cooperativo efectuado sobre el proyecto LilyPond ha crecido y se ha mantenido durante el período de su vida considerado. Ambas animaciones presentan a los desarrolladores simbolizados mediante iconos, y a los archivos de trabajo que les corresponden mediante partículas móviles (en el caso del método de Ogawa) o fijas sobre el árbol de directorios (en el caso de Caudwell).

Toda la documentación del proceso de su elaboración aparece en el capítulo de resultados.

8. Estudio 1. Resultados y discusión

En línea con nuestro objetivo de mostrar la historia y el futuro de un proyecto de software libre, hemos explicado ya cómo se pueden extraer de nuestra fuente primaria (a saber: los datos públicos del proyecto LilyPond de tipografía musical) una serie de datos en bruto cuyo análisis pormenorizado puede arrojar luz sobre el incierto sentido de la “salud” de un proyecto de estas características. En los apartados dedicados al número de desarrolladores, el tamaño de la documentación, tamaño de los instalables binarios y evolución de las intervenciones en el árbol del proyecto, presentamos los resultados de estos análisis junto a los correspondientes gráficos.

Posteriormente emitiremos una serie de interpretaciones sobre los gráficos obtenidos en el capítulo de resultados, y a continuación compararemos nuestro enfoque con el de otros estudios que se están realizando en el campo de la visualización del software.

8.1. Desarrolladores

El análisis del historial de intervenciones al código fuente LilyPond permite contar los autores que han realizado estas intervenciones.

Los autores del código pueden contarse fácilmente, aunque este recuento está sujeto a incertidumbre. No es trivial reducir el número de personas que han contribuido al código hasta tener representada en dicha relación una sola vez a cada persona física. En un estado inicial del estudio aparecían, con un número de intervenciones superior al resto, los dos desarrolladores del núcleo original, que no habían trabajado desde el principio con el sistema Git y que figuraban en los registros con un solo nombre de usuario, dado lo cual no era posible distinguir sus intervenciones particulares. Una revisión posterior de los propios autores ha ayudado a discernir las respectivas autorías.

La herramienta Git, a través de instrucciones de consola de UNIX, ofrece una primera aproximación del número total de desarrolladores distintos: 361 direcciones electrónicas correspondientes a 185 personas, es decir, hay una media de casi dos identidades repetidas por cada persona física. Por ejemplo, los registros de la figura 35 (reales, pero modificados por privacidad) corresponden a la misma persona.

```
Maurizio Favioli <mmf@vistamaresoft.com>
Maurizio M. Favioli <mmf@Juno.(none)>
Maurizio M. Favioli <mmf@kwannon.(none)>
Maurizio M. Favioli <mmf@luthien.homenet.telecomitalia.it>
Maurizio M. Favioli <mmf@luthien.(none)>
Maurizio M. Favioli <mmf@vistamaresoft.com>
```

Figura 35: Ejemplo de identidades diversas que corresponden al mismo autor.

Las identidades repetidas deben tratar de combinarse en una sola. Se han efectuado tres intentos independientes: uno privado, del autor, a mediados de 2008; el segundo, más exhaustivo, fue añadido oficialmente en septiembre de 2009 por Reinhold Kainhofer, de la Universidad Técnica de Viena; finalmente se vuelve a revisar a finales de 2014 para este estudio.

La depuración de las identidades de los autores es un proceso semiautomático. El examen de las posibles identidades repetidas es manual; cuando se detecta una, se le asigna el autor único correspondiente en una tabla. Se repite el proceso para cada identidad. Finalmente, a partir de esta tabla se convierten todas las identidades iguales a una dada, al autor único que les corresponde.

La aplicación de esta política de depuración objetiva ofrece una reducción a 153 colaboradores, incluyendo a los autores originales. La lista de nombres completa aparece en la figura 104 (anexo A).

Mediante un script escrito en el lenguaje de programación Python podemos recorrer el árbol en su rama maestra y acumular la cuenta de cada uno de los nombres distintos que aparecen como autores, iniciando una cuenta nueva para cada mes. Después de tabular estos datos, los presentamos de forma gráfica en la figura 36, que presenta el número de desarrolladores activos distintos por mes desde el comienzo de la historia del proyecto. La tabla 7 presenta un resumen del recuento.

Tabla 7: Autores activos por mes.

Parámetro	Mínimo	1er. cuartil	Mediana	Media	3er. cuartil	Máximo
Valor	1	3	8	9.60	14	31

El proyecto ha seguido un régimen sostenido de clara atracción de desarrolladores interesados en el mismo, si tenemos en cuenta que el equipo no tiene un número cerrado y que el abandono es frecuente. Si se contabilizan los desarrolladores distintos por mes, obtenemos un máximo histórico de 31 en diciembre de 2009, seguido de enero de 2011 con 27. La evolución del número de desarrolladores activos distintos por mes sigue fuertes variaciones estacionales.

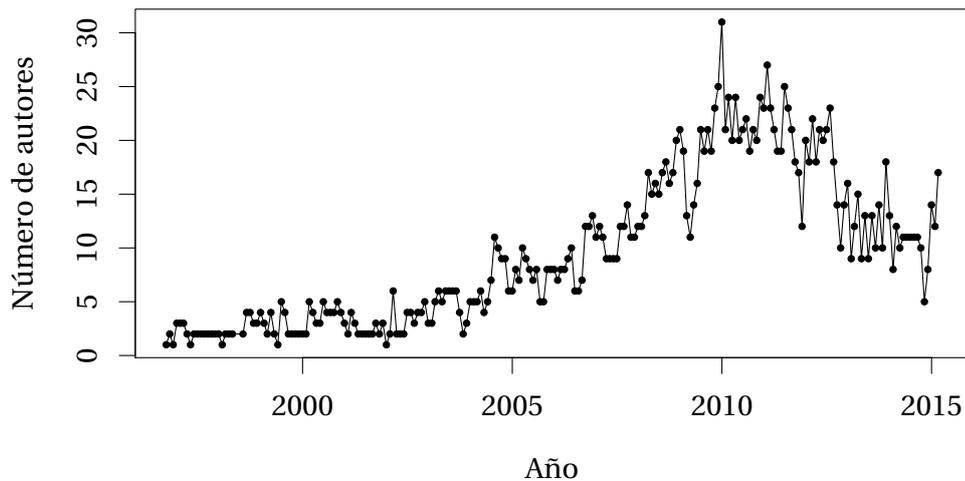


Figura 36: Número de desarrolladores activos distintos por mes desde octubre de 1996 hasta junio de 2015

El número de autores distintos por año se obtiene integrando la gráfica anterior en períodos de doce meses y depurando de nuevo los autores repetidos. El resultado es la figura 37, que presenta un máximo de 55 personas diferentes a lo largo del año 2010.

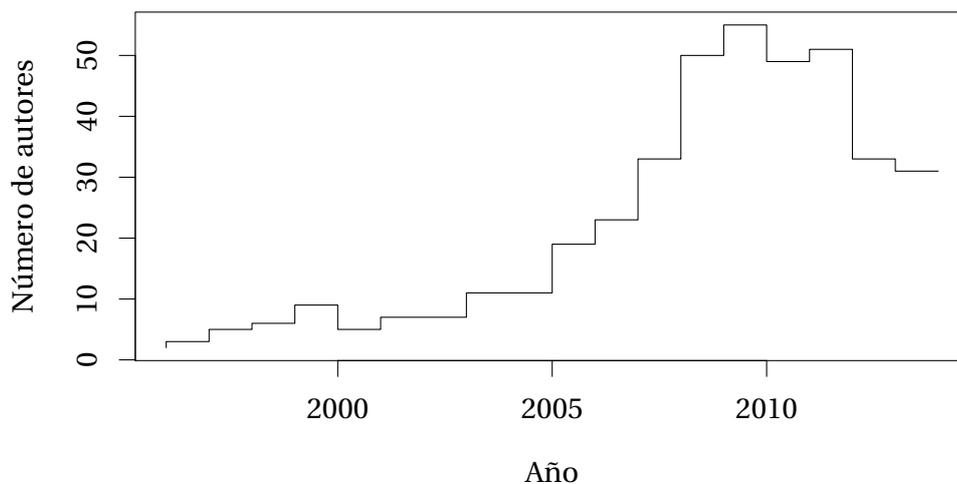


Figura 37: Número de desarrolladores activos distintos por año completo desde el inicio del proyecto en 1996 hasta 2014

El número de desarrolladores activos diferentes por año tuvo una tasa de crecimiento aparentemente exponencial hasta ese momento¹ y una estabilización en un número de 30 desde entonces. El número medio de autores por año es de 21,9.

El estudio de González-Barahona y col. (2014) también identifica entre las fuentes de error la sobreestimación del número de desarrolladores por repetición de identidades. El proceso de depuración que siguen se efectúa en períodos de un mes; no creemos necesario distinguir

¹Se calcula que la base de crecimiento exponencial es 1.4 para este período. Es imposible que este régimen se hubiera mantenido durante mucho tiempo más.

entre los desarrolladores que corresponden a una identidad determinada dentro de períodos distintos, porque diferentes autores nunca intercambian sus identidades, o al menos en el examen manual no se detecta ningún caso. Por tanto, pensamos que es más acertado asignar las identidades a los autores una sola vez para toda la historia.

8.1.1. Carga total de trabajo

Por la naturaleza y por la insuficiente granularidad de los datos disponibles, es imposible conocer el tiempo real que cada autor dedica al proyecto. Sin embargo, se dispone de una medida del número de autores en períodos de un mes. Si se agregan estos datos para todos los meses de vida del proyecto, se puede obtener la estimación de la tabla 8.

Tabla 8: Datos para la estimación de la carga total de trabajo del proyecto LilyPond.

Meses	221
Suma de los autores por mes	2.121

Por supuesto, dos mil personas-mes de trabajo a mediados de 2015 es un límite superior; se asume que el trabajo de un equipo de n autores durante un mes es, como máximo, n autores-mes, pero como apuntan González-Barahona y col. (2014), el trabajo de los autores es a tiempo parcial e incluso esporádico. Si los autores de un mes son iguales a los de otro, o son otros distintos, esto no afecta al cómputo de la carga total, excepto que como se ha dicho no se pueden agregar estos datos anualmente sin depurar de nuevo la lista de autores únicos.

Este dato equivale a 115 personas-año, cifra que resulta de dividir los 2.121 autores-mes entre los 18,4 años de actividad registrada.

El caso de estudio que consideran González-Barahona y col. (2014) es el proyecto *glibc*, con más de 20 años de historia; en este período, la carga de trabajo total obtenida es de 996 personas-mes. Tanto este dato como el de 2.121 de LilyPond son cotas superiores y en ninguno de ellos se tiene en cuenta que los autores con una sola intervención se están contabilizando como una persona-mes.

8.2. Intervenciones

El total de intervenciones por mes se muestra en la figura 38. El máximo de intervenciones en un solo mes fueron las 596 de diciembre de 2006, lo que supone una media de casi 20 al día en este período.

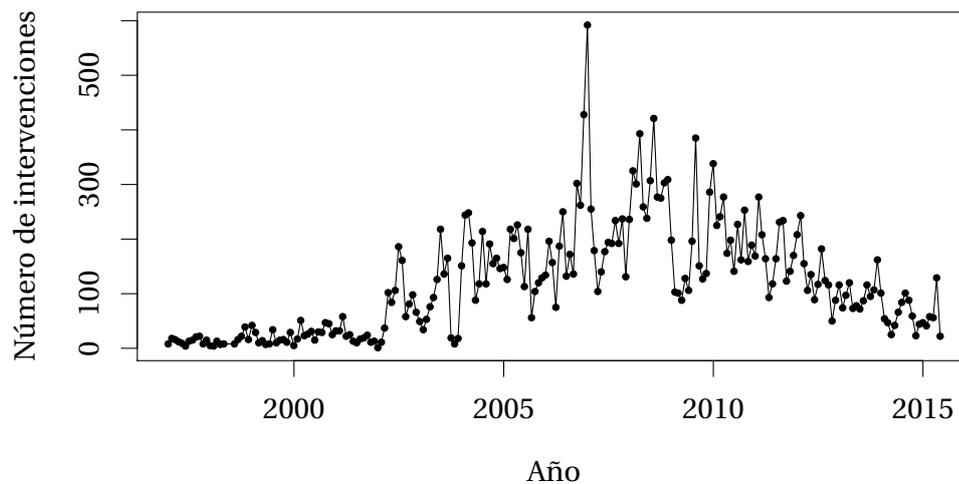


Figura 38: Número de intervenciones individuales en el código fuente de LilyPond por mes desde octubre de 1996 hasta junio de 2015.

Integrando la curva anterior en períodos de un año, se obtiene la gráfica escalonada de la figura 39.

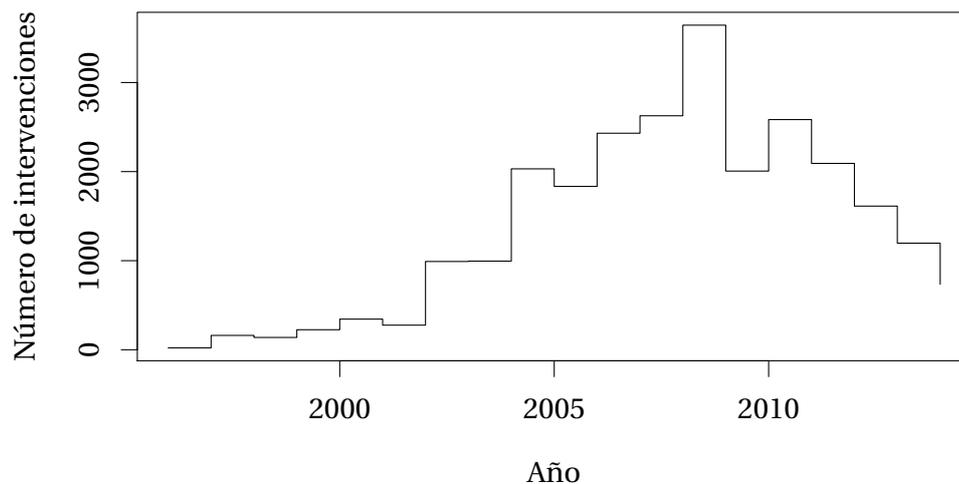


Figura 39: Número de intervenciones individuales en el código fuente de LilyPond por año completo desde el inicio del proyecto en 1996 hasta 2014

El máximo histórico en un solo año se produjo en 2008 con un total de 3644 intervenciones. A partir de ese momento se observa un descenso más lento que la subida que condujo al máximo. Los datos finalizan con el número de 734 durante 2014. No debe tenerse en cuenta el dato de las 353 intervenciones de 2015 porque es un año incompleto, ya que la medida se tomó en el mes de junio.

Dado que disponemos de los datos anuales del número de intervenciones y de un recuento anual de los desarrolladores, podemos hallar la proporción entre ambos. La figura 40 presenta la evolución del número medio de intervenciones por autor en cada año.

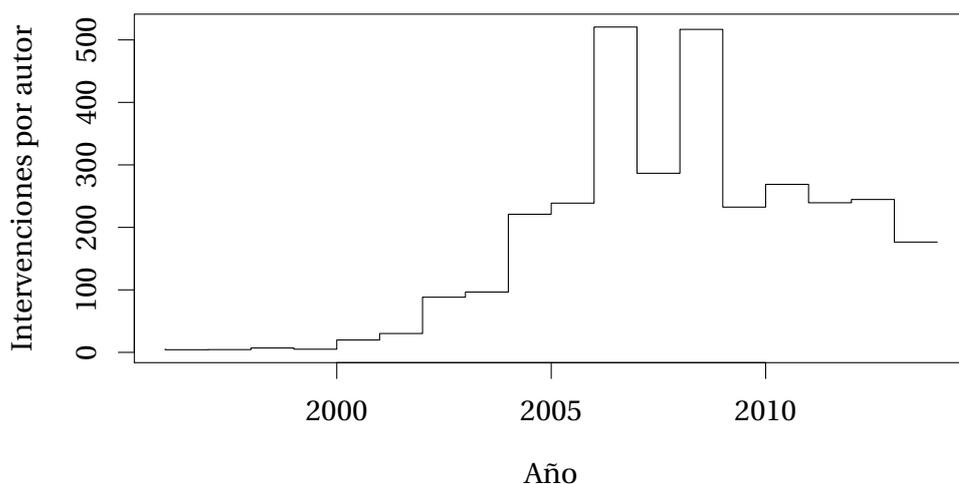


Figura 40: Evolución de la proporción entre el número de intervenciones y el número de desarrolladores por año.

Los valores máximos alcanzados en 2007 y 2009 (más de 500 intervenciones por autor) se debieron probablemente a los procesos de traducción de la documentación, ya que esta actividad se caracteriza por un elevado número de pequeñas modificaciones que se suceden a intervalos de tiempo cortos, y se llevan a cabo por un número reducido de autores. El fondo de la curva parece haberse estabilizado a partir del año 2005 en alrededor de 250 intervenciones nuevas por autor y año.

Refiriéndose a la contabilización del número de intervenciones, Percival (2010) opina que ésta no es una métrica excesivamente buena porque no todas las intervenciones son equivalentes sino que existen distintos tipos, y la misma carga de trabajo está más o menos fraccionada según el estilo de trabajo de los distintos autores, pero sí es una métrica aceptable porque es fácil de medir y de entender.

8.3. Tamaño del código

El código fuente de LilyPond consiste² en 1.123.879 líneas de código, distribuidas entre 4.983 archivos escritos en 15 lenguajes distintos.

La tabla 9 presenta un resumen de los archivos existentes por cada uno de los lenguajes (ya sean de programación o de marcado para la documentación), con expresión del número de líneas y los porcentajes respectivos sobre ambos totales. Por ejemplo, la documentación, con sus traducciones, está escrita en Texinfo³ con código de LilyPond empotrado. Se observa en la tabla que la documentación se compone de 1.894 archivos (38,02 % del total) y cuenta con

²La fecha de referencia para estos datos es junio de 2015.

³En el glosario de la página 347 figuran estos términos técnicos.

831.204 líneas de código, cerca de un 75 % del total por este concepto. El número medio de líneas en estos archivos es de 439.

Tabla 9: El código fuente de LilyPond, clasificado por lenguajes. Ordenado por número de líneas de código. Datos de junio de 2015

Lenguaje	Archivos	%	Líneas	%
Documentación: Texinfo+LilyPond	1.894	38,02	831.204	73,96
LilyPond exento	2.075	41,64	85.631	7,62
Lisp+Scheme	122	2,45	53.529	4,76
MusicXML	159	3,19	37.501	3,34
Python	87	1,75	32.395	2,88
C, C++	298	5,98	28.786	2,56
Metafont	122	2,45	23.103	2,06
TeX/LaTeX	45	0,91	13.608	1,21
sh, m4, make	162	3,24	6.551	0,58
lex,yacc	2	0,05	6.257	0,56
PostScript	4	0,07	2.557	0,23
Perl	2	0,05	2.057	0,18
Metapost	1	0,02	579	0,05
scripts de FontForge	2	0,05	76	0,01
ABC	6	0,12	43	0,00
Total	4.983	100,00	1.123.879	100,00

Los archivos más numerosos (2.075, el 41,64 %) están dedicados a las pruebas de regresión para el control de fallos, en forma de música en el propio lenguaje LilyPond, exento (es decir, no empotrado dentro de otros archivos). El hecho de que todo este material suponga solamente un 7,62 % las líneas de código, indica que son de pequeño tamaño (41 líneas/archivo, por término medio).

Los algoritmos de disposición espacial de la notación musical en el papel están escritos en los lenguajes C, C++, Lisp y Scheme. Sumados, suponen otro 8 % aproximadamente, tanto del número de archivos como de las líneas de código totales.

Una de las funciones del programa es analizar el texto de las partituras que se le ofrecen como entrada, en un paso previo a su representación gráfica. Este análisis se produce en dos fases: análisis léxico y análisis sintáctico. La gramática que necesitan estos analizadores viene codificada en los lenguajes *lex* y *yacc*. Por otro lado, el diseño de las fuentes tipográficas que contienen la forma de todos los elementos gráficos, se encuentra representado en el lenguaje Metafont. Reunidos ambos conjuntos de archivos junto a una serie de otras contribuciones menores, constituyen menos de un 10 % del total.

Tratándose de un programa que produce un resultado gráfico, como lo son las partituras musicales, destaca que el peso del código fuente de Metafont, que define la forma de todos los símbolos y figuras musicales, ocupa solamente un 2,06 % de las líneas, con 122 archivos.

Los dos gráficos de la figura 41 reflejan la evolución del número total de archivos del proyecto

y de líneas de código, respectivamente. Las dos curvas alcanzan las cantidades totales mencionadas de casi 5.000 archivos y un millón de líneas en una ascensión constante desde el inicio.

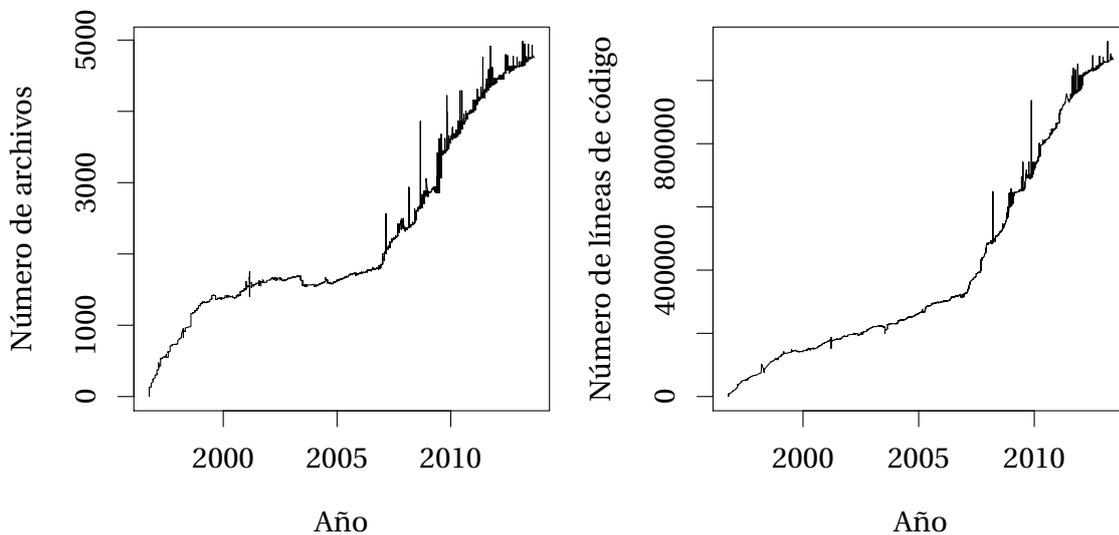


Figura 41: Evolución del número total de archivos del proyecto (a la izquierda) y del número total de líneas de código (a la derecha).

El ritmo de aumento parece acelerarse a partir de 2007. Aunque la forma de ambas curvas sugiere a primera vista un crecimiento exponencial, sabemos que no es posible por la propia naturaleza de las variables, cuyos valores aumentan de forma incremental, no multiplicativa. Dicho de otra forma, cada valor instantáneo se incrementa a partir del inmediatamente anterior mediante una suma. Si el ritmo de crecimiento cambia, es porque resulta del trabajo combinado de un número variable de autores, cada uno de los cuales aporta distintas cantidades de código al proyecto, según regímenes de intensidad que son propios de cada autor.

También a partir de 2007 comienza a aparecer un fenómeno de creación y destrucción inmediata de archivos, que se reflejan en la curva como picos de corta duración. Sin embargo, la duración no es tan corta como para que no quede registrada, de manera que conjeturamos que sobre un fondo de archivos estables en número, una intervención introduce cierta cantidad de archivos con cierto carácter provisional que poco después se deciden suprimir, volviendo al número estable.

La creación y registro de archivos que se pueden recrear automáticamente a partir del código fuente es una práctica desaconsejada (Chacon, 2009; Loeliger, 2009; Swicegood, 2008), pero puede tener dos ventajas: el proceso de construcción del programa se acelera (ya que no es necesario recrear dichos archivos) y además se aligera el volumen de dependencias (los programas adicionales necesarios para la compilación), eliminándose las que se requerían solamente para construir dichos archivos.

8.4. Tamaño de la documentación

La documentación oficial del programa se distribuye conjuntamente con el mismo e incluye una colección de libros en los formatos PDF (el formato compatible de Adobe), HTML (páginas web estáticas) e Info (un sistema de ayuda para la consola, común en entornos UNIX), y se obtienen a partir de un único código fuente escrito en el lenguaje de marcas Texinfo, que es el formato estándar de documentación de los proyectos incluidos en GNU.

El código fuente de la documentación se encuentra en el mismo árbol de directorios que el código del propio programa. Incluye tanto los originales en inglés como las traducciones, que en septiembre de 2009 comprendían los idiomas español, alemán, francés, holandés y japonés. En ese momento solamente la traducción al español se encontraba completa, aunque el grado de sincronismo con los originales es un estado frágil que no se mantiene actualizado al 100 % más que durante unos pocos días en el mejor de los casos. La explicación es que las aportaciones a los originales de los manuales por parte del equipo de redactores de la documentación se producen diariamente en un proceso continuo, como se desprende del análisis del historial de versiones. En junio de 2015 no hay ninguna traducción completa pero la lista aumenta con los idiomas adicionales checo, chino, catalán, húngaro e italiano. *La documentación es un proceso, no un producto* (di Carlo, 1998).

El código fuente de la documentación sin su traducción, tiene un tamaño de 21,4 Mb lo que representa algo más de un tercio (37,2 %), del total ocupado por los archivos del árbol de código. Si incluimos las traducciones, el tamaño de este código fuente sube a 37,5 Mb (dos tercios del total). Podemos ver estos datos en la tabla 10.

Tabla 10: Tamaño relativo de las secciones del código fuente del proyecto LilyPond a fecha de junio de 2015.

Sección de la fuente	Tamaño (Mb)	Proporción
Código del programa	20,0	34,7 %
Documentación original	21,4	37,2 %
Traducciones	16,1	28,0 %
Total documentación	37,5	65,2 %
Total código fuente	57,5	100,0 %

La aportación de las traducciones, teniendo en cuenta que son diez idiomas, es menos significativa de lo que cabría esperar por dos motivos: *a)* Están muy incompletas. *b)* Comparten con la documentación original en inglés una gran cantidad de código *no traducible*: principalmente música codificada en el lenguaje LilyPond para los ejemplos musicales de los manuales.

No toda esta documentación está orientada al usuario final, sino que existe una guía del colaborador, una guía de referencia interna para programadores, etc. En cualquier caso todos los manuales reciben aportaciones de los redactores y por el momento no haremos distinción

entre ellos a la hora de analizar la evolución de la documentación como un todo.

El conjunto de la documentación se distribuye de diversas maneras: como documentos de hipertexto HTML y PDF en el sitio oficial, y como un gran archivo único en formato tar y comprimido con bZip2, el equivalente en UNIX al popular formato Zip.

El archivo comprimido recibe el nombre informal de *tar ball* o *tarball*. De acuerdo con la licencia bajo la que se publica este contenido, el código fuente también está disponible.

El tamaño del código fuente de las documentaciones traducidas aparece en la figura 42.

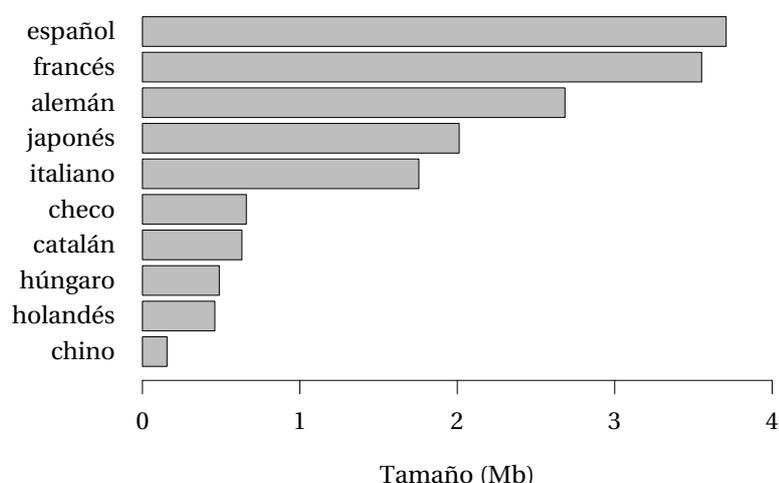


Figura 42: Tamaño del código fuente de cada una de las traducciones de la documentación de LilyPond

Las traducciones española y francesa son aquellas cuyo código fuente es mayor, aunque ninguna de ellas está completa. El tamaño de la documentación original en inglés no es relevante como comparación con el de las traducciones porque existe mucho material no traducible, como las ilustraciones musicales, que están codificadas en lenguaje LilyPond y entrelazadas con el texto normal.

Dado que las versiones obsoletas del programa no desaparecen del libre acceso por parte del público y que todas ellas van acompañadas de su correspondiente *tar ball* que contiene la documentación, podemos recabar la información correspondiente a la fecha y tamaño de cada archivo tal y como aparecen en los enlaces de descarga del sitio oficial (que no está estrictamente bajo el dominio del proyecto sino que está cedido por cortesía del instituto de tecnología de la Universidad de Virginia)⁴.

Posteriormente, este texto en bruto es analizado mediante un *script* o guión escrito en el lenguaje AWK, y se convierte en un fichero que consiste en los datos dispuestos en columnas separadas por espacios.

⁴La fuente de estos datos es el sitio oficial de descarga de la documentación de LilyPond: <http://download.linuxaudio.org/lilypond/binaries/documentation/>. Visitado en abril de 2015

El gráfico resultante, que muestra la evolución del tamaño del conjunto de la documentación, aparece en la figura 43.

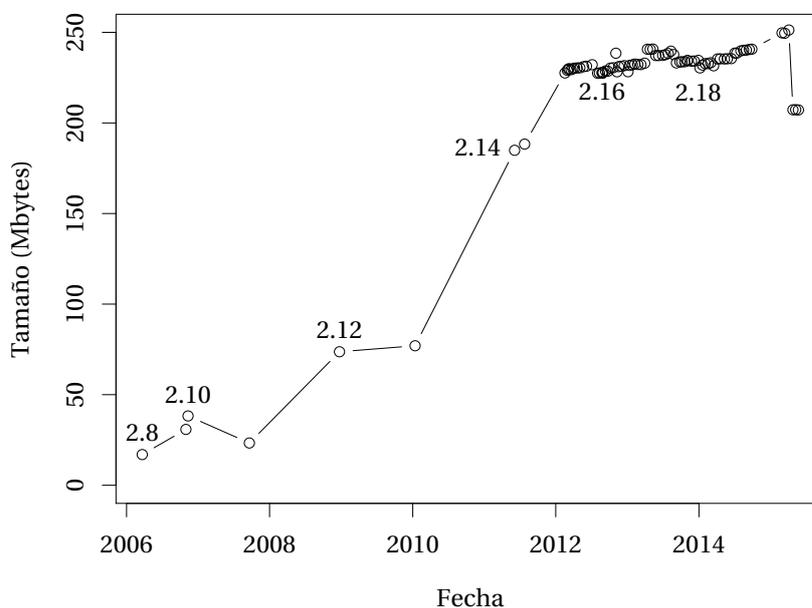


Figura 43: Evolución del tamaño de los 84 *tar balls* históricos de la documentación, en escala lineal, frente a la fecha, con indicación del número de versión.

El gran aumento del tamaño hasta 2012 se debió a la incorporación de las traducciones. El tamaño máximo alcanzado por la documentación ha sido de 250 Mb en abril de 2015, con la versión 2.19-19. Obsérvese que se trata del tamaño del *tarball* comprimido. La diferencia con los 37 Mb sin comprimir del código fuente (véase la tabla 10) es que la documentación que se distribuye está construida en varios formatos: HTML, Info y PDF; estos últimos son documentos formateados, listos para imprimir, que incluyen las fuentes tipográficas además del texto, índices y tablas de referencias cruzadas, que se expanden a partir de la fuente. Posteriormente el tamaño ha bajado a “solo” 200 Mb por la eliminación de apartados obsoletos.

8.5. Tamaño de los instalables binarios

Los instaladores binarios del programa LilyPond son los archivos que el usuario debe descargar y ejecutar para instalar el programa en su sistema. En los servidores de descarga gratuita solo se encuentran las versiones recientes y algunos binarios antiguos por razones históricas. El tamaño analizado es el de los instaladores para Windows; el tamaño de las versiones para otros sistemas operativos tiene un comportamiento similar.

Se presenta un problema al tratar de extraer una información precisa del tamaño de cada uno de estos archivos, pues aunque se encuentran listados para su libre descarga, la cifra que indi-

ca el tamaño figura redondeada a la décima de Megabyte, lo que en relación con un tamaño total de menos de 20 Mb, supone una proporción considerable del redondeo frente al valor de referencia. Se presenta a continuación la solución adoptada para aumentar la precisión de este dato.

Un enfoque de *fuerza bruta* hubiera consistido en descargar cada uno de los 86 binarios enlazados. En lugar de eso hemos analizado los enlaces examinando el código fuente de la página, de donde se obtiene una lista de nombres de archivo. Después, un script o pequeño programa local escrito en el lenguaje PHP se conecta al servidor emitiendo las solicitudes de tamaño y fecha para cada uno de estos archivos, mediante el protocolo estándar HTTP. El resultado se vuelca en una tabla CSV que es finalmente presentada en forma de gráfico por el programa R.

En la figura resultante (véase la figura 44) se hallan etiquetadas solamente las versiones estables, que tienen un número de versión par como 2.10 o 2.12. Las versiones de desarrollo tienen un número impar como 2.11.

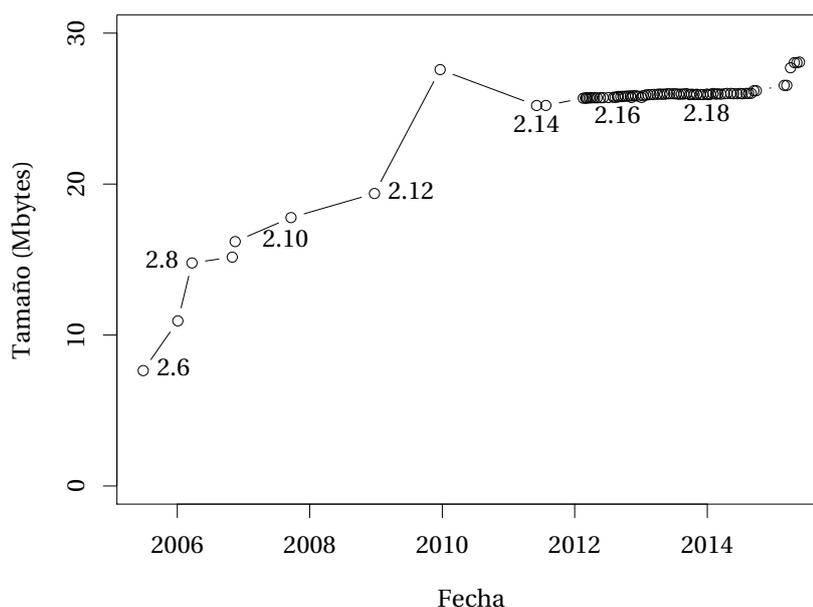


Figura 44: Tamaño de las versiones estables de los instaladores binarios lanzados para Windows durante los diez últimos años, frente a la fecha de lanzamiento.

El tamaño mínimo es de 7,28 Mb y corresponde a LilyPond 2.6, de junio de 2005. Puede observarse que el tamaño se estabiliza sobre 25 Mb en la versión 2.14 de junio de 2011, aunque sigue subiendo ligeramente a cada versión. El tramo entre 2.15 (febrero de 2012) y 2.19.15 (septiembre de 2014) contiene 69 versiones. La versión de mayo de 2015 tiene un tamaño máximo, de 26,78 Mb.

Los días de diferencia que separan cada versión de la siguiente se muestran en la figura 45, que

se ha preparado a partir de las diferencias finitas (derivada numérica) de la curva anterior. La gráfica presenta un segmento estable centrado en los 14 días para el período mencionado.

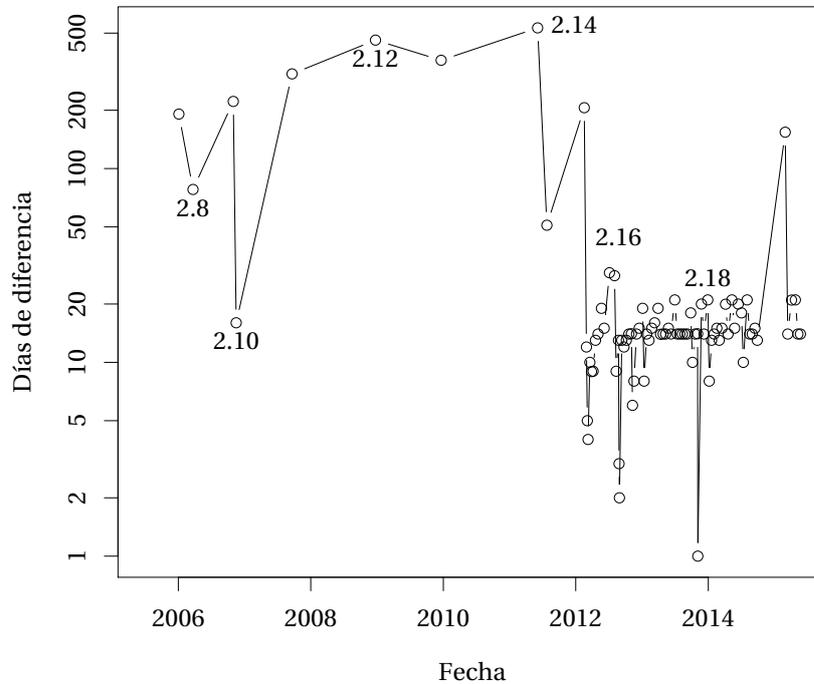


Figura 45: Días de diferencia entre versiones sucesivas de LilyPond durante los diez últimos años, frente a la fecha de lanzamiento. El eje y está en escala logarítmica.

Puede observarse, además, un cierto número de lanzamientos con una diferencia por debajo de 10 días, e incluso una (a principios de 2014) que se produjo con un solo día de diferencia.

La imagen que ofrece esta evolución es la de un programa cuyas versiones se hacen públicas con mucha frecuencia, y aunque obviamente el tamaño por sí solo no es ningún indicador de calidad, puede decirse que las pequeñas diferencias positivas indican frecuentes adiciones y funcionalidades nuevas.

8.6. Actividad media diaria, semanal y mensual

Del análisis de la marca de tiempo registrada en cada una de las intervenciones, se obtiene el mapa de la actividad diaria media de todos los desarrolladores, durante toda la vida del proyecto. Véase la figura 46.

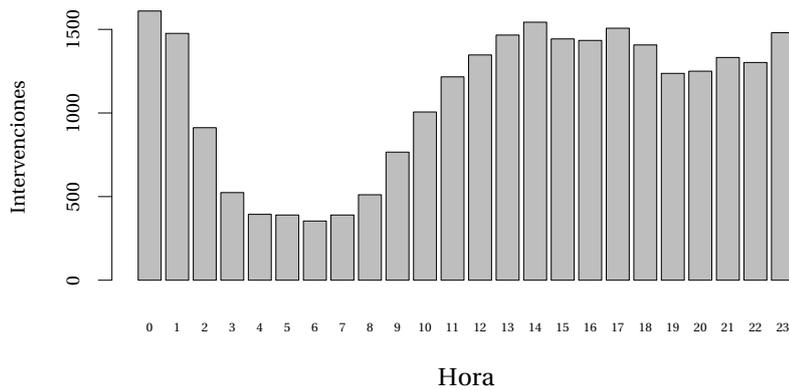


Figura 46: Número total de intervenciones frente a la hora del día, para toda la historia del proyecto.

Este dato varía ampliamente, con una desviación típica de 441 intervenciones. En este proyecto los autores participan en su tiempo libre y las horas con más intervenciones son las doce de la noche y las dos de la tarde. Las horas con menos actividad, como es de esperar, están en el rango nocturno entre las tres y las ocho de la mañana. 2.964 intervenciones se han hecho en este período, un 11 % del total. El rango de valores está en una proporción de 4,5:1.

En la figura 47 se observa la actividad por día de la semana, que es muy uniforme.

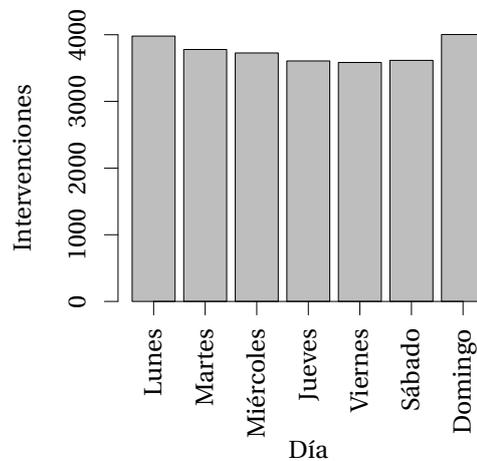


Figura 47: Número total de intervenciones por día de la semana.

La desviación típica es de solamente 175 intervenciones y el rango está en proporción 1,1:1. Los días de mayor actividad son el domingo y el lunes, pero teniendo en cuenta la actividad diaria, que presenta un máximo a las doce de la noche, se puede interpretar que una gran proporción de la actividad adscrita a los lunes pertenece en realidad a la noche del domingo.

En la figura 48 se presenta la actividad media anual por mes.

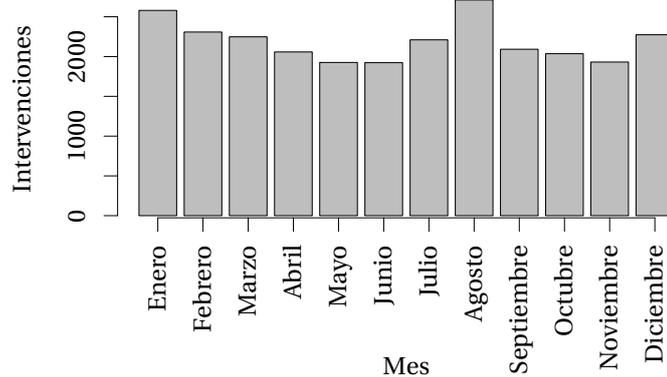


Figura 48: Número total de intervenciones por mes del año.

Los meses que tradicionalmente tienen más actividad son agosto y enero, y los de menos actividad son mayo y junio. No disponemos del dato de ocupación de los desarrolladores, pero el resultado expuesto podría explicarse si fuesen en su mayoría estudiantes o profesores. La desviación típica es de 254 intervenciones, con una proporción de rango de 1,4:1.

No se ha hecho ninguna corrección para compensar el hecho de que los meses del año tienen distintas duraciones.

8.7. Evolución del trabajo sobre el código

La unidad de trabajo que se considera en el análisis del mismo, es la intervención. Cada una de ellas hace pasar el contenido de un estado instantáneo a otro, lo cual comporta diferencias en uno o más archivos. Cuando se visualizan estas diferencias en el caso de los archivos de texto, se obtiene un listado de las líneas modificadas. Si el archivo no fuese del tipo textual, sino binario, las diferencias seguirían existiendo, aunque no podrían representarse como una relación de líneas modificadas. En todo caso, el tamaño de la intervención, con todas sus líneas añadidas o suprimidas, es una métrica aceptable sobre la magnitud de los cambios al código del programa.

Se consideró la posibilidad de mostrar todos los datos sin ninguna reducción, pero presentaban el aspecto de un bloque macizo. Además, ciertas intervenciones que repiten datos se deben suprimir por razones de equidad. Se explicó en la fundamentación teórica que la historia de intervenciones no es lineal sino que puede ramificarse, lo que produce efectivamente varias historias en paralelo. Cuando las historias se reunifican, producen una intervención especial conocida como *merge* o mezcla. Las intervenciones que son mezclas contienen todos los datos de modificaciones de las dos ramas. Estos datos repetidos pueden falsear la historia y por tanto todas las intervenciones caracterizadas como mezclas deben suprimirse del recuento; la forma de identificar una mezcla es examinar sus ancestros. Si una intervención

tiene un solo ancestro, es una intervención normal; si tiene dos ancestros, es una mezcla de ramas.

En la figura 49 se representan los datos recopilados acerca del trabajo sobre el código fuente, a lo largo de la vida del proyecto. La reducción efectuada en este gráfico consiste en que se ha agregado el total del número de líneas modificadas en períodos de una semana (puntos verdes). La agregación mensual correspondiente aparece en rojo y la anual en negro.

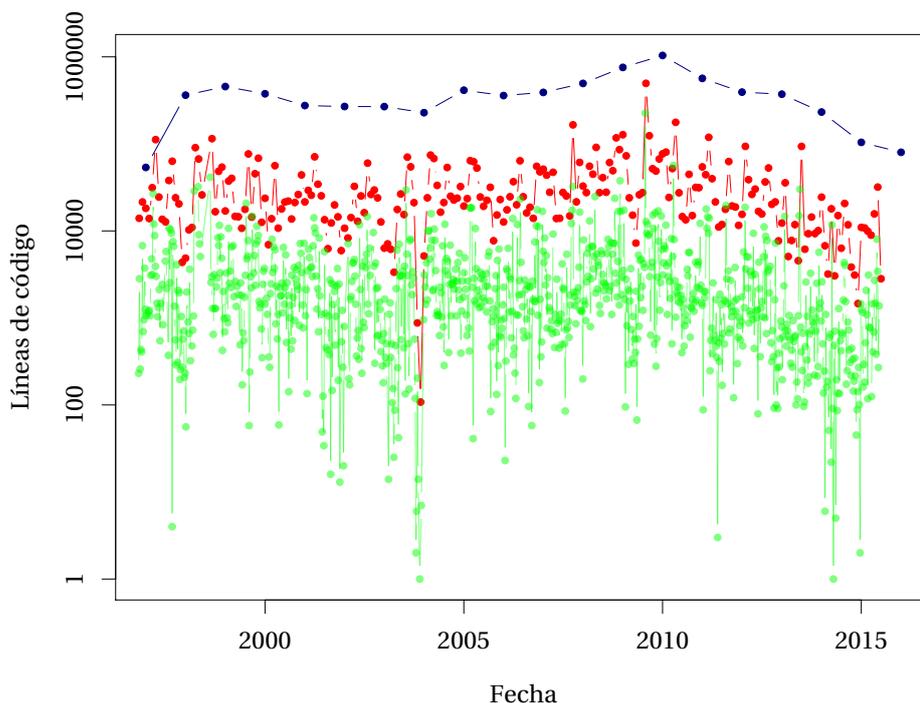


Figura 49: Líneas de código modificadas durante la vida del proyecto LilyPond. De abajo hacia arriba: agregaciones semanal, mensual y anual, en escala logarítmica, frente a la fecha de cada intervención. Los valores de 2015 son aproximados.

Se ha determinado que la escala del eje y, que corresponde al volumen de trabajo, sea logarítmica, basándonos en un hecho notable que impediría una correcta visualización si, de otro modo, se empleara una escala lineal: la densidad de intervenciones es aproximadamente uniforme en todos los órdenes de magnitud en base 10, desde un solo carácter hasta 1 Mb. Sólo los tamaños muy grandes destacarían del grueso de los elementos, que en su inmensa mayoría estarían acumulados en el extremo inferior del rango de una escala lineal. La escala logarítmica, por razones obvias, hace imposible que el valor cero sea el extremo inferior de la escala, por lo que en su lugar empleamos la unidad, que de todas formas es el mínimo absoluto para el tamaño de una modificación puntual.

Destaca el máximo que supera el millón de líneas añadidas o suprimidas durante el año 2010.

Sorprende también la aparente constancia con la que un grupo en constante renovación, formado por 153 personas, se han sucedido durante más de 18 años en la tarea de desarrollar, mejorar y mantener el proyecto en sus ratos libres. Recordemos que el listado completo aparece en el apéndice de la página 317.

Respecto a los resultados que se han presentado, es necesario comentar un aspecto de la recolección de datos que contienen información de fecha y hora. Dicha información se toma tal y como aparece en el registro del árbol de versiones. La información temporal proviene de la marca de tiempo registrada, que incluye una información de huso horario del punto del planeta en que la misma se ha realizado.⁵ El cómputo de la marca de tiempo debe incluir una compensación de zona horaria, pues en caso contrario se produciría una discrepancia de más/menos 12 horas respecto al meridiano de Greenwich, dado que los desarrolladores aportan modificaciones desde puntos tan dispares como Canadá o Singapur. Esta discrepancia es irrelevante para una representación gráfica, ya que se diluye dentro de la posición horizontal de todos los puntos de la nube. Además, la diferencia está muy comprimida en su dimensión horizontal debido a que el rango considerado es de varios años.

Por el contrario, el análisis de la hora de intervención para mostrar la actividad media a lo largo del día, debe ignorar la información de huso horario, de tal manera que el dato que se analiza es la hora local, en vez de la hora universal.

8.8. Evolución del trabajo sobre la documentación

Para visualizar el trabajo efectuado sobre la documentación del proyecto, se deben analizar individualmente cada una de las intervenciones, antes de cualquier agregación como la que se efectuó para producir la figura 49. Como se ha dicho, cada intervención afecta a uno o más archivos. Se debe determinar, para cada uno de estos archivos, si pertenece al código fuente de la documentación, o si por el contrario es código fuente del programa. A continuación se efectúa la agregación anual de estas variables por separado y el resultado es el gráfico de la figura 50. Se han presentado en color verde los datos de trabajo que incluyen exclusivamente cambios en el código, y en rojo los que incluyen solamente cambios en la documentación.

⁵Para ser exactos, la marca horaria corresponde al reloj del sistema en el que el desarrollador ha efectuado su intervención. Por lo tanto, depende de que el reloj esté correctamente ajustado tanto a la hora local como a la zona horaria correspondiente. La confianza en que estas marcas de tiempo son mayoritariamente correctas se basa en las siguientes suposiciones: a) la gran mayoría de los desarrolladores emplean sistemas operativos tipo UNIX como p.ej. GNU/Linux; b) en estos sistemas, la hora local se obtiene generalmente de la red por medio del protocolo NTP; y c) se observa que los usuarios de estos sistemas suelen tener correctamente ajustada la zona horaria del sistema.

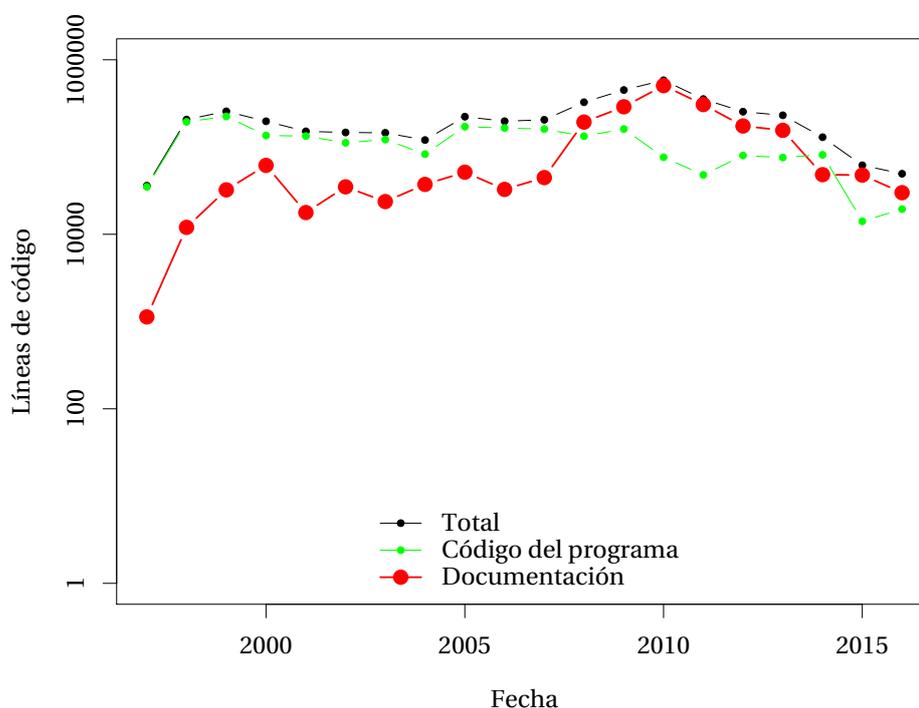


Figura 50: Carga de trabajo sobre la documentación del proyecto, frente a la fecha. Líneas de código modificadas totales, de documentación y de código del programa. Agregaciones anuales, en escala logarítmica. Los valores de 2015 son aproximados

Se observan las fases que el proyecto ha atravesado a partir de una preponderancia de las intervenciones en el código del programa, hacia una tendencia más enfocada al desarrollo de la documentación. Hacia el año 2000 se completa la documentación original en inglés y se realiza un trabajo de mantenimiento hasta 2007. En 2008 se acometen las traducciones y la carga de trabajo supera con creces a la del resto de las tareas. Una vez completadas las traducciones principales, que son la española y la francesa, sigue un régimen de mantenimiento y de aportaciones de traducciones nuevas con menor volumen y un grado de actualización moderado.

Lo más interesante de esta técnica de representación es que puede aplicarse también a la música, siempre que exista una codificación de la misma en formato de texto. En la figura siguiente se representa la historia completa del proceso de composición de una obra musical que está gestionada y publicada como si fuera un producto de software: la ópera *Affaire Étrangère* de Valentin Villenave, de la que ya hemos hablado en el apartado 5.8 (pág. 106) del capítulo 5.⁶ El universo en este caso es de 272 intervenciones.

⁶Villenave quiso publicar el código de la partitura de su ópera en el servidor Savannah de GNU, pero tuvo muchos problemas para hacerlo porque los administradores consideraban que se trataba de *música* y nunca habían hecho tal cosa. Finalmente fue publicado de forma totalmente privada e independiente.

La figura 51 muestra el tamaño, medido en caracteres y sobre escala logarítmica, de las intervenciones individuales hechas al árbol de versiones, frente a su fecha, desde el inicio de la composición de la ópera en marzo de 2008 hasta su estreno en febrero de 2009.

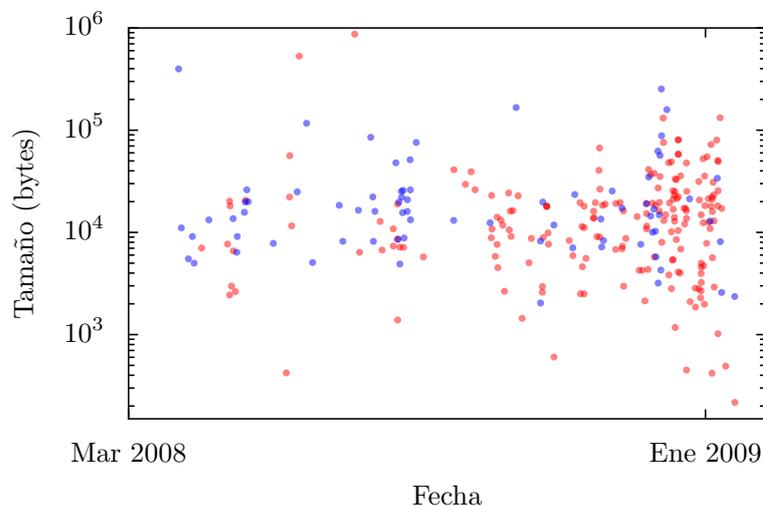


Figura 51: Historia de las intervenciones efectuadas sobre la partitura de la ópera *Affaire Étrangère* de Valentin Villenave. Fuente de los datos: código publicado por Villenave. Imagen tomada de (Vila, 2009).

Tratándose de una ópera, en la que hay material musical puro y partes cantadas, se ha diferenciado entre las intervenciones que de una u otra forma modifican la parte de los cantantes, aquí en color rojo, del resto, que modifican solamente las partes instrumentales o la estructura de la partitura misma (representadas en color azul). Con un número pequeño de intervenciones totales y un período de tiempo total reducido (comparado con el de un proyecto a largo plazo), no tiene sentido la agerregación que se ha efectuado anteriormente y la nube de puntos mostrada representa el conjunto completo de intervenciones individuales. El aumento de la densidad de puntos y la tendencia a un predominio del rojo hacia el final del gráfico dan idea de la evolución del proceso de composición en la medida en que se acercaba la fecha del estreno en febrero de 2009.

Las conclusiones provisionales que se desprenden de los análisis que se han efectuado son que la documentación del proyecto LilyPond está siendo mantenida por un equipo de redactores de forma activa y prolongada en el tiempo; el tamaño total de la documentación oficial se ha duplicado en tres años; y la tendencia que se observa parece sugerir que este régimen se mantendrá en un futuro.

Por otro lado, los instaladores binarios, que constituyen la forma en que el programa se distribuye (excepción hecha del propio código fuente),⁷ sufren un continuo y casi imperceptible aumento de tamaño con el tiempo. Las intervenciones en el código y en la documentación

⁷El hecho de que el código fuente se distribuya no es una simple cortesía frente al usuario; es una condición para que el software sea libre, tal y como recogen los criterios de la *Free Software Foundation* y la propia Licencia General Pública de GNU. Sin embargo, al menos en el caso de LilyPond es notable el hecho de que por su gran

contribuyen a ello, a pesar de que en un sentido general una intervención dada no tiene por qué aumentar directamente el tamaño neto del objeto intervenido. Si, por ejemplo, un corrector ortográfico efectuase una serie de puntualizaciones en la redacción del texto, ello no haría aumentar el tamaño del mismo, sino su corrección, siendo por otro lado la corrección una variable cuya evaluación cuantitativa nos resulta difícil de imaginar.

8.9. El *factor bus* de Percival: actividad frente a ranking y ley de Zipf

En los Encuentros Mundiales de Software Libre, Graham Percival (Percival, 2010; citando a Healy y Schussman, 2003) sugiere que si se ordenan los autores de mayor a menor y se representa el número de sus contribuciones, la distribución sigue una ley potencial.

La ley potencial, con ejemplos en la ley de Zipf, la distribución Pareto, la distribución Weibull, la función del olvido de Wickelgren, etc., se observa en fenómenos diversos en muchos campos de la ciencia. Por ejemplo, la magnitud de los terremotos frente a su frecuencia sigue esta ley, según Richter (1958). Se trata ahora de comprobar si para el proyecto LilyPond la distribución de las contribuciones de los desarrolladores al proyecto cumple también una ley potencial.

Entre otras observaciones interesantes, Percival constata que los desarrolladores del núcleo central del equipo hacen la inmensa mayoría del trabajo en un proyecto cooperativo.⁸ El tamaño del núcleo central mínimo es lo que se llama, de manera un tanto macabra, el *factor bus*. La idea del factor bus había sido apuntada por Stephany, Mens y Gîrba (2009) con un sentido inverso: un valor de riesgo que es mejor cuanto más pequeño. También aparece en los ensayos de Shlomi Fish sobre OSS⁹ Fish (2007) referida a Linus Torvalds y el proyecto Linux.

El factor bus responde a la siguiente pregunta:

¿Cuál es el número mínimo de personas tales que, si les atropella un autobús, el proyecto desaparece con ellos?

A estos efectos se puede considerar equivalente la salida del proyecto por razones familiares u otro motivo. En cualquier caso, si una sola persona hace todo el trabajo y sólo él comprende la lógica del proyecto, el factor bus es uno. Aunque haya un equipo grande de colaboradores, si toda la actividad recae sobre esta persona, es muy difícil que el desarrollo pueda continuar sin ella. La cuestión de cómo determinar el valor del factor bus no goza de consenso, pero conjeturamos que puede hacerse una estimación a partir de la forma de la curva de Zipf. De

complejidad se recomienda desde el propio proyecto la instalación del programa a partir del binario precompilado, y no su compilación *en casa*, proceso que está caracterizado por sus propios autores como “muy enrevesado”.

⁸Quizá se está usando para la definición el término definido, en este caso el *núcleo central* es precisamente el conjunto de las personas que hacen la mayoría del trabajo.

⁹OSS son las siglas en inglés de *Open Source Software* (software de fuentes abiertas).

acuerdo con el proceso expuesto en el capítulo de metodología del estudio y con el objeto de comprobar si las contribuciones de los autores siguen una distribución de ley potencial, se obtiene para la curva de contribuciones frente al ranking R del autor de cada conjunto de ellas, el resultado que se aprecia en la parte izquierda de la figura 52.

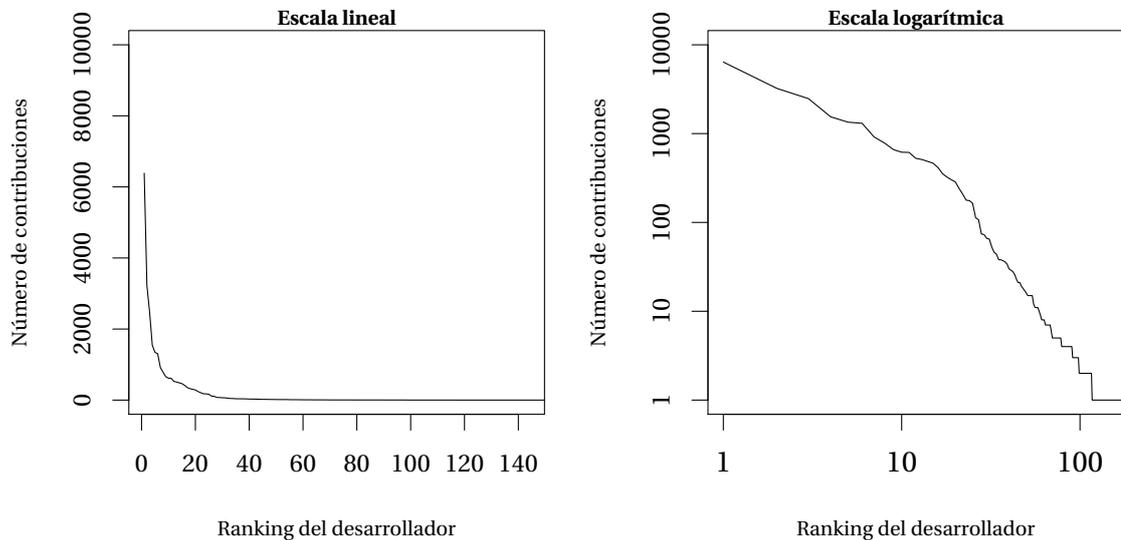


Figura 52: Curva de contribuciones por autor frente a su ranking, en escala lineal (a la izquierda) y en escala logarítmica en los dos ejes, o *curva de Zipf* (a la derecha).

La forma teórica de la curva es inversamente exponencial; para convertir el carácter exponencial en lineal se emplea una escala logarítmica en los dos ejes (parte derecha de la misma figura). Ahora, si la curva siguiese exactamente la ley de Zipf, la gráfica presentaría una línea recta.

Se observa en la segunda curva un escalonamiento extraño para $R_i \geq 95$ que obedece simplemente a que en la cola de la lista de autores ordenada por su actividad en orden descendente, muchos autores comparten el mismo número de intervenciones y por tanto el mismo ranking.

Si la pendiente de la curva es menor que -1 en la primera mitad del rango horizontal (concavidad hacia arriba), ello indica una carga de trabajo más pronunciada sobre un conjunto reducido de colaboradores, y un factor bus reducido. Si, en cambio, la pendiente es mayor de -1 en la primera mitad (concavidad hacia abajo), es un indicador de que el trabajo está más repartido de lo que la ley de Zipf predice y no recae sobre un núcleo reducido de individuos: el factor bus vendría indicado por el punto en que cambia la pendiente de la curva.

Para explorar los cambios en la forma de esta curva contabilizando períodos de tiempo más cortos, en lugar de reducir el tiempo, se acumulan los datos acaecidos dentro de una ventana de seis meses que se va desplazando una semana cada vez, y se reúnen todos los gráficos en una secuencia temporal que tiene tres dimensiones. Para los 18 años, en ventanas de 26

semanas, se produce una serie que contiene un total de 913 curvas; de ellas se seleccionan seis para ilustrar las fases de la evolución del proyecto.

Así pues, la secuencia de imágenes que aparece en la figura 53 corresponde a seis instantes separados por un tiempo de tres años.

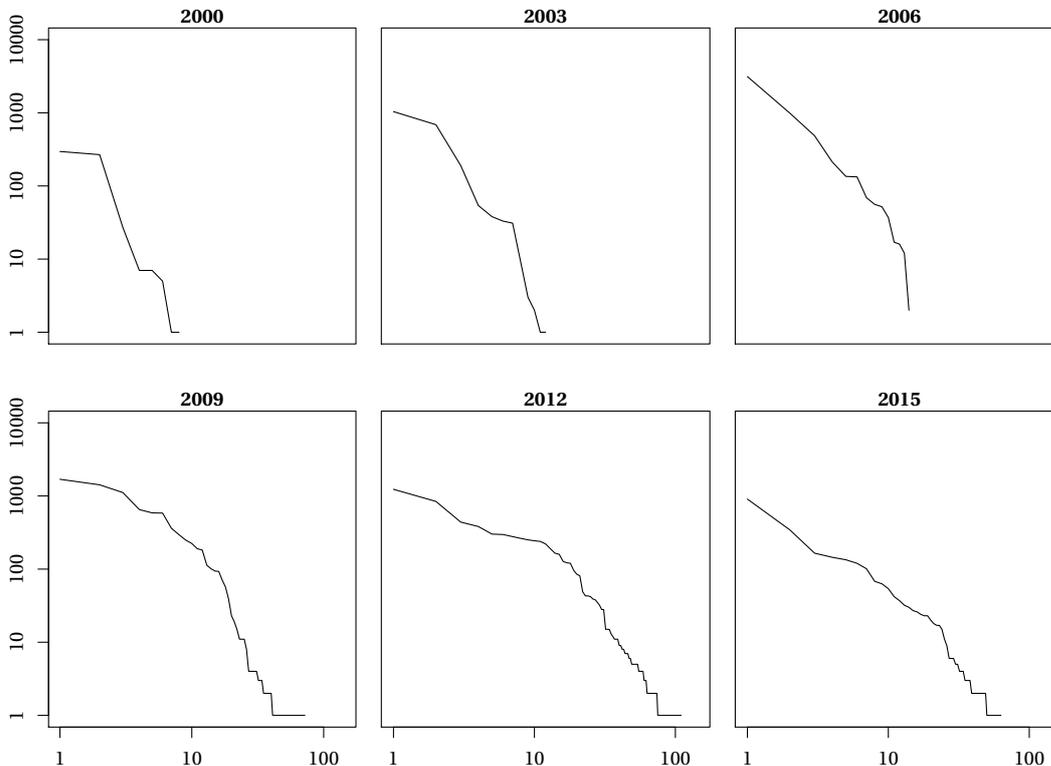


Figura 53: Evolución de la curva de Zipf al final de seis períodos de tres años, para los 18 años de historia de contribuciones al proyecto LilyPond. Número de contribuciones por cada desarrollador activo durante ese período, frente a su ranking correspondiente dentro del mismo conjunto de desarrolladores. Escala logarítmica en los dos ejes.

Se deduce del examen de esta secuencia, observando las dos primeras imágenes, que durante un período inicial de unos seis años el núcleo central de desarrolladores era muy reducido y la carga de trabajo se concentraba en los miembros de este grupo; aparece un codo después del segmento correspondiente al segundo autor del ranking. Esto sugiere un factor bus de 2.

A partir de la mitad del período considerado, la curva se vuelve claramente cóncava hacia abajo, lo que indica un reparto de la carga de trabajo más amplio que el que predice la ley de Zipf. El cambio en la pendiente se encuentra a partir del autor número 10 del ranking, lo que indicaría un factor bus de 10.

Para el proyecto LilyPond, el hecho de que el factor bus haya crecido de 2 a 10 y no ofrezca signos de descenso posterior, indicaría que la continuidad del mismo se prevé como muy probable a largo plazo. Ello permitiría depositar expectativas en el futuro de la edición de

partituras con software libre y su uso en educación, donde la estabilidad de este tipo de herramientas es fundamental.

El estudio de Avelino, Valente y Hora (2015), de julio de 2015, calculó una estimación del factor bus de 133 aplicaciones populares de GitHub, una gran colección de software libre. Los resultados mostraron que la mayoría de ellos tenía un factor bus estimado pequeño (46 % con un valor unidad y 28 % con un valor de 2), y que solo 7 sistemas presentaban un valor mayor o igual a 10 . La estimación no se hizo en función del número de contribuciones de los autores, sino del grado de autoría de los archivos. El estudio es estático y no trata de visualizar la evolución del factor bus en el tiempo, ni tiene en cuenta la edad de los proyectos.

Para estudiar la evolución en el tiempo de la altura general de la curva de Zipf, se ha reducido en una dimensión el conjunto de datos, extrayendo solamente la altura del punto situado más a la izquierda, que corresponde a la actividad del autor más prolífico en ese momento.

Se genera la serie temporal con este dato y se muestra en un gráfico frente al tiempo. El resultado aparece en la imagen 54 sobre ejes en escala logarítmica, combinado con un modelo de regresión sencillo de mínimos cuadrados. El eje horizontal es el número de semanas desde el inicio del proyecto, y el eje vertical es la media móvil bilateral, con una ventana de 26 semanas, del número de contribuciones.

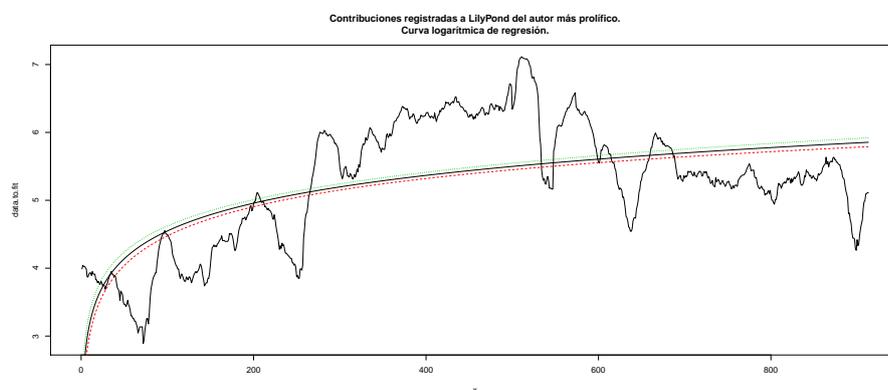


Figura 54: Evolución del número de contribuciones por semana registradas en los 18 años del proyecto LilyPond por parte de los autores sucesivamente más prolíficos, y curva de regresión logarítmica.

En lugar de preparar una serie temporal con la media móvil de los datos en bruto y elaborar después una descomposición de esta serie temporal, el reciente trabajo de González-Barahona y col. (2014) emplea la media móvil bilateral para suavizar la representación a posteriori.

8.10. Modelización de la serie temporal

Se ha calculado aquí el modelo lineal por el procedimiento de mínimos cuadrados, y se ha representado el resultado como una curva ascendente por el uso de una escala vertical logarítmica. El empleo de la escala logarítmica en el eje vertical se justifica por el hecho de que el mayor número de contribuciones son de pequeña entidad. Así pues, los valores se acumularían en la parte baja del eje si la escala fuese lineal, mientras que las contribuciones importantes y menos numerosas, se saldrían de la escala. Es necesario presentar todos los datos de una forma que acoja con igual detalle los valores que son numerosos y pequeños y los que son más raros y grandes; la escala logarítmica cumple este cometido.

El modelo de regresión lineal tiene un poder de predicción muy limitado para este caso porque los valores contienen altibajos estacionales y rachas aparentemente aleatorias, a veces muy amplias tanto en duración como en magnitud.

Para efectuar un modelo más ajustado de esta serie temporal existen dos opciones principales: ARIMA¹⁰ y STL.¹¹ Ambos métodos son capaces de descomponer una serie temporal con el objeto de efectuar un análisis de su tendencia.

El método ARIMA se desarrolló por parte de la oficina americana del censo. Se discute en Lariday y Quenneville (2012). El empleo de modelos ARIMA para la elaboración de pronósticos está descrito en Pindyck y Rubinfeld (1998). El paquete `forecasting` para R está descrito en Hyndman y Khandakar (2007). Por otro lado, el método STL fue desarrollado por Cleveland, Cleveland, McRae y Terpenning (1990) y tiene algunas ventajas, pero ofrece un resultado similar en este caso.

Vamos a proceder a elaborar un modelo ARIMA sobre la serie temporal de una variable que sigue el dato semanal ya mencionado, que es el número de intervenciones acumuladas del autor más prolífico, sobre ventanas de 6 meses (26 semanas). Esto corresponde, como se ha dicho, a la evolución en el tiempo del punto extremo izquierdo de la curva de la figura 52.

El paquete utiliza el método de máxima verosimilitud y en primer lugar busca el orden del modelo. El modelo viene denotado por tres enteros (p, d, q), donde p es el número de coeficientes de autoregresión; d es el grado de diferenciación, y q es el número de coeficientes de media móvil.

El análisis determina que el mejor orden es (3, 1, 3) para los parámetros (p, d, q) con el signifi-

¹⁰ARIMA son las siglas en inglés de *Autoregressive integrated moving average* (modelización autorregresiva integrada de media móvil)

¹¹STL son las siglas en inglés de *Seasonal and trend decomposition using LOESS* (descomposición estacional y de tendencia mediante LOESS). A su vez, LOESS corresponde de forma no exacta a *Locally-weighted polynomial regression* o «regresión polinómica localmente ponderada» y es un método de estimación de relaciones no lineales.

cado siguiente: los datos se han diferenciado una vez ($d = 1$) y después se ha seleccionado un modelo con tres coeficientes de autoregresión AR ($p = 3$) y tres coeficientes de media móvil MA ($q = 3$).

8.10.1. Descomposición de la serie temporal

El modelo permite hacer una descomposición de la serie temporal en componentes estacional, de tendencia e irregulares.

Dado que los datos son semanales, la frecuencia es 52 y el eje x está graduado en años. El resultado se ve en la figura 55.

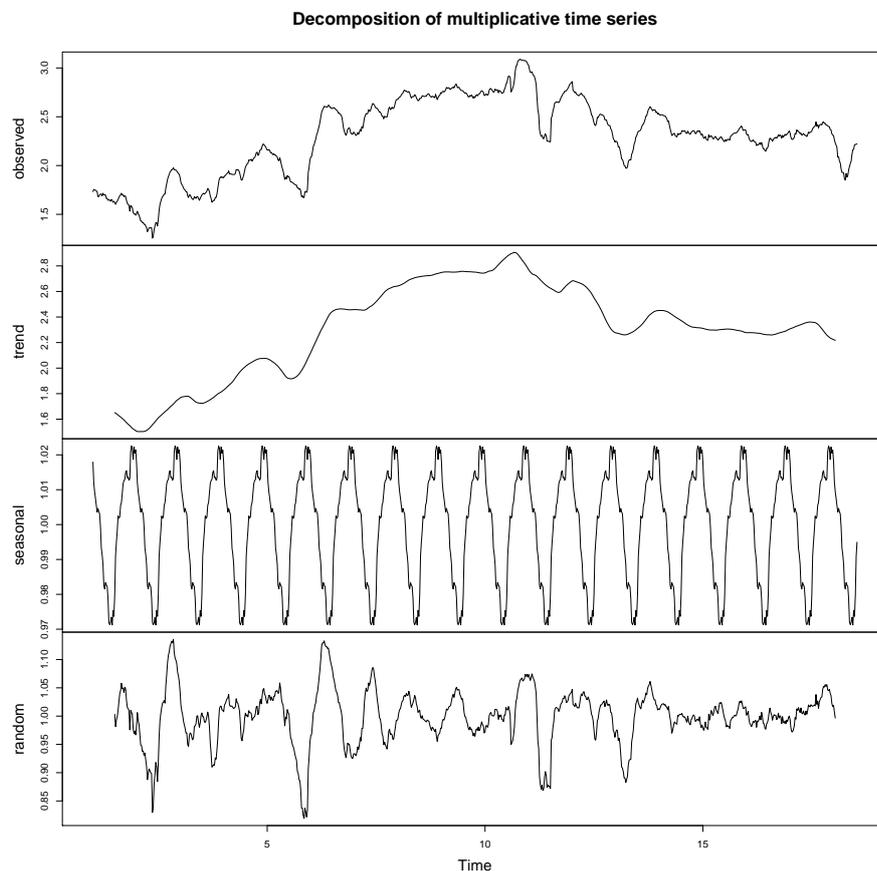


Figura 55: Descomposición de la serie temporal original (arriba) en la curva de tendencia, variaciones estacionales y componente aleatoria. El eje vertical es logarítmico (en base 10). El eje temporal está graduado en años.

En esta descomposición destaca la forma en que la curva de tendencia (la segunda por arriba) se eleva en una primera fase hasta la edad de 10 años y, después de una segunda fase de descenso, se estabiliza finalmente en una meseta con una altura mayor que la media de todo el período.

Si los diagnósticos confirman que el modelo es válido, el significado para el proyecto LilyPond sería el siguiente: que, a pesar de que los sucesivos cambios en la composición y en la intensidad de trabajo del equipo de desarrollo han producido altibajos, el proyecto en su conjunto presenta robustez y es plausible afirmar que se mantendrá activo en un futuro, como mínimo a medio plazo.

8.10.2. Diagnósticos sobre el modelo

Se valida el modelo mediante la aplicación de medidas diagnósticas. El cómputo del modelo ofrece al mismo tiempo el error estándar para cada coeficiente. A partir de estos valores se hallan los intervalos de confianza de los coeficientes para 2,5 % y para 97,5 %; a la vista de estos márgenes de error, se determina si el valor verdadero de alguno de los coeficientes podría ser cero. Podría serlo si los intervalos de confianza correspondientes contuviesen el cero. Los intervalos aparecen en la tabla 11.

Tabla 11: Intervalos de confianza para los coeficientes del modelo. AR: coeficientes de autoregresión. MA: Coeficientes de media móvil.

Coef.	2.5 %	97.5 %
AR1	1.795	2.062
AR2	-1.964	-1.620
AR3	0.528	0.770
MA1	-1.759	-1.432
MA2	1.237	1.633
MA3	-0.522	-0.208

Dado que ninguno de los intervalos contiene al cero, podemos decir que todos los coeficientes son probablemente significativos.

A continuación se efectúa una evaluación del modelo mediante el examen de tres gráficos de diagnóstico: los residuos estandarizados, la función de correlación y los valores p del estadístico de Ljung-Box.

A partir del examen de los residuos estandarizados se puede determinar si están adecuadamente dispersos por toda la serie temporal, o bien se acumulan con el mismo signo en un cierto período de tiempo, lo que se conoce como *racimos de volatilidad*. No se deben confundir los residuos estandarizados con la componente aleatoria (gráfico inferior de la figura 55).

Por su parte, la autocorrelación entre los residuos es un análisis que arroja información sobre la periodicidad de los mismos. Valores altos (más de 0,3) en algún coeficiente de la función de autocorrelación (ACF) indicarían que esta periodicidad podría haberse incorporado a los coeficientes del modelo en primer lugar.

Finalmente, se examinarán los valores p del estadístico de Ljung-Box para ver si el modelo detecta residuos que están afectados por algún factor periódico. El criterio es que los valores sean grandes (no menores de 0,05).

La figura 56 muestra gráficamente estos tres conjuntos de valores para el modelo calculado. De arriba hacia abajo están los residuos estandarizados, los coeficientes de autocorrelación de los residuos y los valores p del estadístico de Ljung-Box.

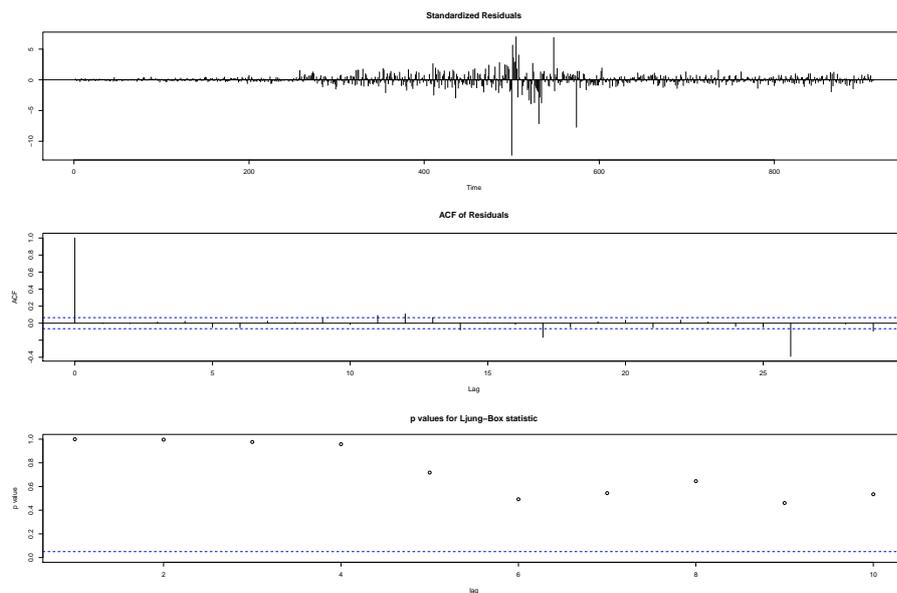


Figura 56: Diagnósticos sobre el modelo: residuos estandarizados, coeficientes de autocorrelación de los residuos y valores p del estadístico de Ljung-Box.

Aunque en el gráfico de residuos estandarizados se aprecian valores altos en forma de racimo, no se trata de un "racimo de volatilidad", es decir, no es una serie de residuos consecutivos del mismo signo. Se acumula en un período de tiempo de aproximadamente 500 semanas en el pasado (es decir, hace 9 o 10 años) y tiende a decrecer. Qué ocurrió hacia 2004 para que el error aumentase, es difícil de determinar por ahora. Es probable que las variaciones muy bruscas sean debidas a la actividad de un solo desarrollador que ha tenido un período de mucha actividad.

La función ACF no muestra una autocorrelación significativa entre los residuos, a excepción de un valor negativo elevado para un valor de retardo de 26 semanas, que interpretamos como un efecto del tamaño de la ventana que estamos utilizando.

Los valores p para el estadístico de Ljung-Box son todos grandes. Esto indica que los residuos no guardan ningún patrón, y que por tanto hemos podido extraer la información relevante y lo que queda es ruido solamente.

8.10.3. Valor predictivo del modelo

En la última fase del modelado de la serie temporal, se emplea éste para efectuar una predicción de la evolución futura. El resultado ofrecido por R con el paquete *forecast* aparece en la figura 57, donde se observa que la continuación de la serie para los próximos años es ascendente.

El resultado expuesto significa lo siguiente para el proyecto LilyPond de un programa de tipografía musical libre: que la intensidad del trabajo del equipo de desarrolladores, medida a través del régimen de sus intervenciones sobre el código fuente, presenta un perfil estable en el tiempo sin que se atisben signos de decadencia o abandono y que el estado de salud del proyecto es bueno.

8.11. Trabajos relacionados. Vídeos de la evolución

La visualización de la dinámica del software ha interesado a distintos investigadores y este objetivo ha impulsado varios trabajos relacionados. Podemos distinguir dos enfoques en cuanto al elemento que centra las visualizaciones. Uno de ellos, el método de Ogawa (Ogawa y Ma, 2009), utiliza representaciones de los desarrolladores o *avatares* como hilo conductor de la visualización; por su parte, el método de Caudwell (Caudwell, 2010) está centrado en los archivos, y son los desarrolladores quienes acuden a ellos en el ejercicio de su actividad.

8.12. Vídeos: 1. Método de Ogawa

De los dos enfoques mencionados, el primero y más antiguo fue iniciado en el grupo de Visualización del Software del departamento de informática de la Universidad de California Davis (UCDavis). De entre los diversos proyectos desarrollados allí, destacaremos el que está a cargo de Michael Ogawa, Kwan-Liu Ma y otros, denominado *code_swarm*, que está centrado en una representación visual del desarrollador individual como un polo de atracción que se hace rodear de una constelación o enjambre de archivos a su cargo.¹²

Una intervención se produce cuando un desarrollador efectúa cambios en el código o en los documentos y los transfiere al repositorio central del proyecto. Tanto los desarrolladores como los archivos se representan como elementos móviles. Cuando un desarrollador aporta un archivo, se ilumina y se desplaza hacia dicho desarrollador. Los archivos se colorean de acuerdo a su propósito, como por ejemplo si se trata de código o de documentos. Si los archivos o los desarrolladores no se han mantenido activos durante un tiempo, se desvanecen. (...) un histograma opcional en la parte baja mantiene un recuerdo de lo que ha ocurrido con anterioridad (Ogawa y Ma, 2009).

¹²El nombre *code_swarm* proviene de *code* (código) y *swarm* (enjambre).

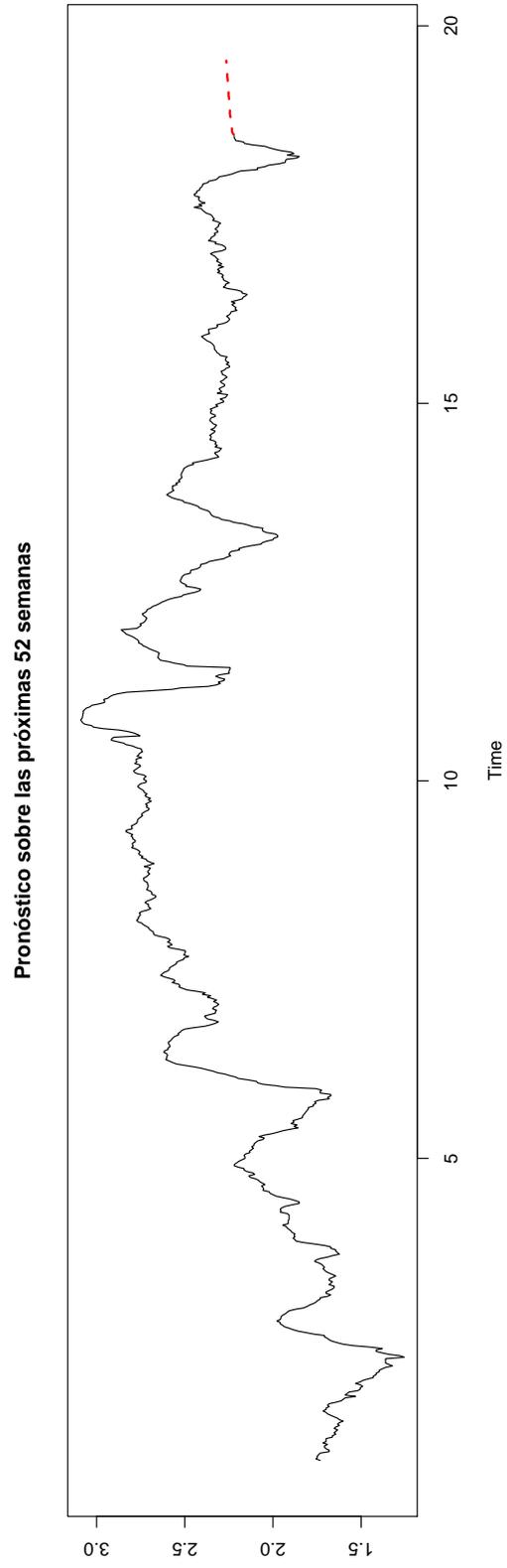


Figura 57: Pronóstico que prolonga la tendencia de la serie temporal según el modelo ARIMA.

Aunque tanto los archivos como los desarrolladores son elementos móviles, los desarrolladores son el objeto central alrededor del cual se agrupan los archivos. Un esquema de los elementos de esta primera animación aparecen en la figura 58.

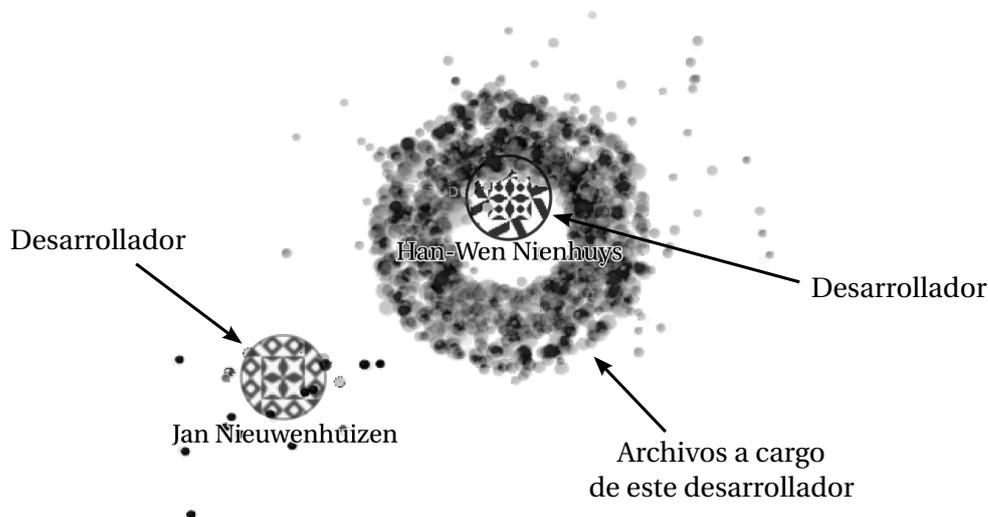


Figura 58: Esquema de los elementos de la primera animación

Code_swarm es software libre licenciado bajo la GNU GPL v3, es decir, la versión 3 de la Licencia General Pública de GNU.

La infraestructura de gráficos está construida utilizando el lenguaje de programación gráfica *Processing* de Ben Fry, descrito por Reas y Fry (2007). Puede encontrarse una completa referencia sobre visualización de datos en otra obra del mismo autor (Fry, 2008). No es necesario descargar o instalar aparte el programa Processing para su empleo por parte de este sistema, pues todas las bibliotecas de funciones necesarias están incorporadas al propio proyecto.

El lenguaje de programación utilizado es Java. Hemos usado la implementación libre de este lenguaje llamada OpenJDK para compilar el software a partir del código fuente, lo que nos ha permitido prescindir absolutamente de los archivos propietarios de Java de Oracle.

8.12.1. Documentación del proceso de generación de los vídeos por el método de Ogawa

Un “fork” o ramificación, en la jerga del desarrollo del software, es una línea de desarrollo independiente que se divide por *clonación* a partir de un proyecto original, o de otro proyecto que a su vez es un fork. En la medida en que los autores de las ramificaciones respeten la licencia del proyecto original, ésta les garantiza que pueden modificar o mejorar el código, continuar el desarrollo y publicar su versión modificada, aunque no están obligados a ello. El software empleado es un *fork* propio de code_swarm que implementa las mismas ideas con algunas adaptaciones en los modelos físicos que se utilizan para realizar las simulaciones.

Después de efectuar la ramificación, el binario ejecutable debe entonces compilarse a partir del código fuente en lenguaje Java. Es necesaria la herramienta de compilación ant de Apache, además del kit OpenJDK de desarrollo de Java, Python y una larga lista de dependencias adicionales.

El siguiente paso es el empleo de la herramienta Git para la descarga completa y actualizada del código fuente de LilyPond, junto con su historial de versiones, a partir de los repositorios oficiales.

En principio, `code_swarm` está preparado para analizar historias SVN del gestor de versiones Subversion. Por tanto, ha sido necesario convertir el historial, exportado a partir de programa Git, en un formato de datos que sea compatible. El procedimiento es volcar los registros emitidos por Git y transformarlos en un archivo XML mediante un *script* Python.

A continuación se requiere un procesado de esta historia convertida, de forma que las direcciones de correo que aparecen en la historia se sustituyan por los nombres cortos de los desarrolladores. Es posible hacer esto mediante otro *script* que realiza una edición masiva del archivo XML que contiene la historia convertida, usando expresiones regulares.

El paso que más tiempo consume es la preparación del archivo de configuración. Es un archivo de texto que hay que editar, siendo algunos de los parámetros cuyo valor debe especificarse, los siguientes:

- La anchura y altura del vídeo, en píxeles
- La imagen empleada como símbolo para los archivos. Se diseñó un símbolo sencillo en forma de esfera.
- Parámetros del modelo físico, como la masa de los objetos (archivos y desarrolladores por separado), su velocidad máxima, su tiempo de permanencia antes de desaparecer por inactividad, el tiempo de vida de las aristas que unen a los desarrolladores con los archivos y tamaño de los avatares, entre otros.
- El número de cuadros por día de actividad, para el que se eligió un valor de 1.
- Los parámetros de la tabla de recuento por tipos de archivo, que aparece sobreimpresa a la izquierda de la animación.
- Otras cuestiones de aspecto como el color de fondo, la fuente tipográfica, el tamaño de cuerpo para los nombres normales de desarrolladores inactivos y para los nombres destacados de desarrolladores activos.
- Rutas de archivo para la entrada (historial) y para la salida (cuadros de imagen).

Cada vez que se modifica un parámetro de la lista anterior se comprueba su efecto en la animación resultante con una versión reducida del historial. Después de un número de ciclos de ensayo y error, se fija una configuración y se procede a la generación de los cuadros de la animación completa. Con los valores seleccionados se obtiene, después un período de unas

ocho horas de cómputo, un conjunto de 6800 imágenes en formato PNG de 1536×1152 píxeles que se convierten a vídeo con la herramienta `avconv`.¹³

El archivo se incorpora al editor de vídeo libre OpenShot¹⁴ donde recibe los letreros sobreimpresos, elaborados con Inkscape.¹⁵

Los primeros seis años de historia de LilyPond aportan una información muy limitada sobre sus autores, como ya hemos reseñado con anterioridad. Dos desarrolladores aparecen como propietarios exclusivos de todos los archivos y como actores de todas las intervenciones. Solamente cuando el proyecto migra al sistema Git, puede apreciarse cómo los desarrolladores se acercan al proyecto y va creciendo el número de éstos, en una especie de gran explosión.

El vídeo completo creado con este enfoque tiene una duración de 4 minutos.

8.13. Vídeos: 2. Método de Caudwell

El segundo enfoque para la generación de visualizaciones de la evolución de un repositorio de código se centra en una representación del árbol de archivos como un objeto central y deja que sean los desarrolladores quienes se muevan alrededor de él, representados por su avatar correspondiente. El sistema Gource, de Andrew Caudwell, sigue este modelo, que se ha utilizado para la elaboración del segundo vídeo. En él, los archivos se agrupan según su ruta dentro del árbol, en disposiciones circulares situadas en los extremos de apéndices que corresponden a las carpetas del proyecto. De igual forma que en el enfoque de Ogawa, cada archivo viene representado por un elemento de pequeño tamaño y forma esférica, coloreado en función del propósito que cumple dentro del proyecto. El dato objetivo que utilizamos para determinar este propósito es la extensión del nombre de archivo.

En la figura 59 se muestra un esquema de los elementos de la animación, correspondiente a un instante inicial de la historia del desarrollo de LilyPond. Entre los archivos de código fuente, la distinción de colores permite, entre otras cosas, diferenciar a los del propio programa de los correspondientes a la documentación. También en esta animación se presenta una leyenda explicativa, con idéntica correspondencia de colores, que en este caso se ordena de forma dinámica en función del número total de archivos que presentan cada una de las extensiones.

¹³`avconv`, obra del equipo Libav, pertenece a la biblioteca del mismo nombre que se originó en 2011 a partir del código del codificador de vídeo `ffmpeg`, de Fabrice Bellard, que a su vez se lanzó por primera vez en diciembre de 2000. Ambos proyectos son software libre licenciado bajo GPL.

¹⁴OpenShot es un programa para Linux, obra de Jonathan Thomas, licencia GPL, originado en 2008 sobre la biblioteca MLT Framework de gestión multimedia. Financiado colectivamente por 1.463 patrocinadores que aportaron 45.028 dólares americanos.

¹⁵Inkscape, de Ted Gould, Bryce Harrington y Nathan Hurst, se originó a partir de 2003. Licencia: GPL.

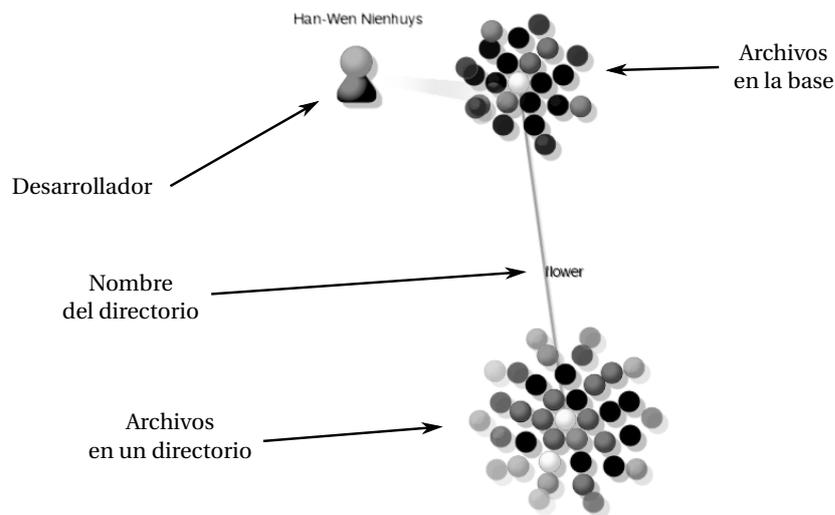


Figura 59: Esquema de los elementos de la segunda animación

Gource también es software libre y, de la misma forma que `code_swarm`, está licenciado bajo la GNU GPL v3.

8.13.1. Documentación del proceso de generación de los vídeos por el método de Caudwell

Gource, de A. Caudwell, es más elaborado en su aproximación a la generación de imágenes y trata de acelerar el proceso empleando bibliotecas gráficas como OpenGL que aprovechan la potencia de cálculo de la tarjeta gráfica del sistema. A causa de esta funcionalidad, la lista de dependencias para producir el programa ejecutable incluye a OpenGL. Aunque se puede desactivar el uso de la biblioteca, el resultado es una generación extremadamente lenta, que hace impracticable el proceso completo de una historia extensa como LilyPond. En `code_swarm` hay una posibilidad de configurar el sistema para que utilice también las rutinas aceleradoras, pero es experimental. El factor de aceleración para Gource es aproximadamente de 10 a 1, lo que ofrece un tiempo total de proceso de menos de una hora, en comparación con las ocho horas de `code_swarm`, sobre el mismo ordenador.

El lenguaje de programación es C++. Una vez satisfechas todas las dependencias necesarias, la generación del binario ejecutable puede hacerse con el compilador estándar `Gcc`, de GNU, y con la utilidad GNU `Make`.

Gource no requiere ningún tipo de conversión para adaptar el registro de Git a un formato compatible. Para la generación de imágenes de los desarrolladores es preciso recolectar sus avatares públicos (o imágenes identificativas a modo de icono, preparadas por ellos mismos) en función de la dirección de correo electrónico de cada uno de ellos. Cuando un desarrolla-

dor no tiene un avatar público, se genera aleatoriamente una imagen geométrica única.

De igual manera que en el proceso de generación de la animación anterior, una larga serie de parámetros se deben configurar dentro de un archivo de texto. Algunos parámetros son comunes; la principal diferencia es que no existe el concepto de “masa” de los objetos como parámetro de la simulación. En lugar de ello, los ajustes atienden más al trazado del árbol de directorios. Los archivos dentro de un directorio se representan como racimos de partículas. Dado que la representación del árbol es un objeto que reacciona a la aparición de nuevos archivos, es necesario configurar la “elasticidad” que afecta al comportamiento de las uniones entre estos racimos. De igual forma, el hallazgo de unos valores adecuados es un largo proceso de ensayo y error. Los iconos de desarrollador tienen un parámetro de fricción que amortigua sus movimientos. Se ajusta también la escala de tiempo para calibrar la duración total de la animación.

No es necesaria la generación de una larga serie de imágenes o cuadros para producir el vídeo a partir de ellas; en este caso puede hacerse a través de una tubería (una estructura de UNIX que permite pasar datos de un programa a otro) que desemboca en el codificador de vídeo. De esta manera, las imágenes se van descartando conforme se incorporan a la secuencia. El resultado es un menor consumo de espacio de almacenamiento.

El codificador de vídeo es, igualmente, `avconv` de la biblioteca Libav. La secuencia se incorpora al editor OpenShot dentro de la línea temporal después de la animación previa y se le superponen los letreros correspondientes.

Aunque las animaciones (considerando cualquiera de los dos métodos) son poderosas herramientas de visualización de la actividad de un proyecto de código abierto que ofrecen panorámicas muy esclarecedoras, nuestros propios resultados en imagen estática, en alta resolución y con destino al papel impreso, pueden resultar técnicamente más útiles para la toma de decisiones en el ámbito de la gestión de los proyectos de software y para la estimación de su evolución futura.

La producción de una representación plana y estática difiere de los trabajos de Michael Ogawa y Andrew Caudwell en varios puntos:

- Tratamos de combinar entre sí varias fuentes heterogéneas (documentación, instaladores binarios, intervenciones) acerca de un proyecto de software, en un estilo de visualización homogéneo. Los sistemas de visualización de Ogawa y Caudwell se limitan al código.
- La resolución del vídeo no es adecuada para la representación en papel que requieren los escritos académicos.
- Se presenta un punto por cada intervención (agregada en períodos de tiempo variables) diferenciando entre los distintos archivos afectados por las diferencias que contiene, y

con expresión del tamaño total de la intervención. El tamaño de cada uno de los archivos no se tiene en cuenta en absoluto en las visualizaciones de `code_swarm` o de Gource.

- En el vídeo se expresa la atención sobre un archivo por parte de las personas que lo modifican de forma que el punto crece cada vez que una diferencia afecta a este archivo. Esto es imposible de representar en papel.

9. Estudio 1. Conclusiones

Hipótesis 1: *Los análisis desde los puntos de vista estático y dinámico de un repositorio de código fuente de software libre pueden revelar aspectos de interés previamente ocultos acerca del régimen de intervenciones que presenta este código, siempre y cuando su historia se halle registrada en un sistema de control de versiones público.*

Se confirma la primera hipótesis, pues los enfoques de análisis estático y dinámico han hecho destacar aspectos como los hábitos de trabajo de los autores o la tendencia que sigue la actividad del equipo, que de otra forma hubieran quedado oscurecidos en medio de un conjunto voluminoso de datos. El software es un objeto complejo y abstracto que escapa a la apreciación natural a través de los sentidos, aparte del aspecto superficial de la interfaz. Los resultados del estudio ofrecen una visión de conjunto acerca de un proyecto de software libre mediante unas técnicas que pueden adaptarse a la obtención de dicha visión a otros proyectos.

La aportación de este resultado es que el investigador dispone de potentes herramientas de visualización y predicción para ofrecer a las personas con algún interés en el software (programadores, gestores, administraciones, empresas, usuarios) indicadores de su estado actual y futuro. Ello a su vez tiene un gran potencial como ayuda a la toma de decisiones y como soporte de la confianza que los interesados pueden depositar en los programas.

Hipótesis 2: *El pronóstico de los diversos indicadores del programa LilyPond, extrapolado a partir del análisis dinámico de la historia de este proyecto de software libre de tipografía musical, es ascendente a medio plazo.*

La segunda hipótesis se confirma parcialmente. Varios indicadores se han analizado y también se ha representado su evolución en el tiempo, pero las fuentes de error son numerosas, y en el estado actual de la investigación no se puede asegurar que las variables tenidas en cuenta definan el estado del proyecto de forma precisa. Las métricas empleadas son intuitivamente válidas, pero no aportan toda la validez que sería deseable. Sí puede afirmarse, a partir del examen de ciertos indicadores como las publicaciones periódicas de versiones del programa, el volumen de la documentación y el avance de sus traducciones, que la evolución no muestra signos evidentes de decadencia y que no hay ningún síntoma de abandono.

Hipótesis 3: *Se puede efectuar un análisis dinámico sobre las partituras musicales, a condición de que estas partituras estén codificadas en forma de documentos de texto, y su historial de cambios esté también registrado y disponible.*

Se confirma la tercera hipótesis. Las técnicas desarrolladas, cuya aplicación se ha documentado detalladamente, se aplican también a un objeto aparentemente muy distinto que, no obstante, guarda cierta similitud con el software: la música. El caso de estudio que se ha tomado es una obra musical extensa, estrenada en un gran teatro francés: la conocida como “ópera libre”, cuyo autor, Valentin Villenave, tuvo la precaución de registrar todos y cada uno de los estados en que la partitura iba quedando durante el proceso de composición, desde el mismo comienzo hasta el momento de su estreno.

En este caso, la aportación consiste en que mediante esta técnica salen a la luz parámetros hasta ahora desconocidos en los procesos de composición musical, orquestación y confección de arreglos. Se pone de relieve la utilidad que tiene para el compositor la técnica del seguimiento de versiones aplicado a la música. Otros casos, no analizados, pero que igualmente se aplican a la confirmación de esta hipótesis, son la gran colección Mutopia o el repertorio de óperas y ballets barrocos de Nicolas Sceaux, ambos gestionados con Git, el mismo sistema de control de versiones que la ópera de Villenave y que el propio programa LilyPond, así como los fondos editoriales de música de Serpent Publications o de la Kainhofer Edition.

Hipótesis 4: *Se estima que la actividad de desarrollo en el caso de LilyPond, como un indicador del grado de compromiso de la comunidad con el mantenimiento y la mejora de dicha herramienta, tiene un futuro que presenta una evolución sostenida o en auge.*

Se confirma la cuarta hipótesis, pues tanto el modelo lineal como el de autorregresión de media móvil, que permite extraer la tendencia de la serie temporal del trabajo sobre el proyecto, predicen un régimen sostenido de actividad del mismo, que se presenta en un estado dinámico y de continua evolución. Así como la segunda hipótesis se refería a indicadores diversos, la cuarta hace referencia una medida concreta de la actividad de desarrollo.

Hipótesis 5: *El caso de estudio considerado, el proyecto LilyPond, presenta robustez frente a las variaciones en la composición de su equipo de desarrollo.*

Se confirma la quinta hipótesis. El proyecto LilyPond se encuentra en vigor, mantenido por un amplio conjunto de programadores y traductores. Aportan razones para esta afirmación, dentro del análisis dinámico que se ha efectuado sobre su historia, una estimación del conocido como “factor bus”, así como de su evolución en el tiempo. Teniendo todo ello en cuenta, se puede afirmar que a pesar de las vastas alteraciones en el tamaño y la composición del equipo de desarrollo, formado por voluntarios, el proyecto ha mostrado una supervivencia que no ha presentado ninguno de otros programas (muy numerosos) que le han precedido en el camino hacia el objetivo final de crear una aplicación capaz de componer tipográficamente todo tipo de música.

La explicación de esta perdurabilidad podría estar fundada en: la intensa y prolongada motivación de los autores, estimulados por un sistema de recompensas ajeno a lo económico;

la flexibilidad del planteamiento del programa; su correcto diseño, que lo hace apto para su mantenimiento por parte de un conjunto amplio de desarrolladores; y por las aplicaciones educativas y profesionales de una especialidad atractiva, compleja y de alto nivel, tanto artístico como técnico, como es la tipografía musical.

9.1. Limitaciones y prospectiva

En este trabajo descriptivo, se han intentado visualizar algunos aspectos de los elementos técnicos y humanos que componen el software. Desconocemos si alguna vez se han utilizado técnicas semejantes para analizar software de edición de partituras, cuyo interés en educación es el mayor aliciente para profundizar en su estudio.

Las fuentes de error detectadas suponen la mayor limitación para la exactitud y la fiabilidad de los resultados. Es difícil encontrar un equilibrio entre la falta de granularidad a pequeñas escalas temporales y la necesidad de contabilizar grandes cantidades de datos para la extracción de modelos que permitan pronosticar satisfactoriamente la evolución del desarrollo.

Tendría un gran interés efectuar una evaluación de los resultados con el objeto de determinar si el espectador de una visualización descubre procesos del desarrollo del software que han podido quedar ocultos, o si los desarrolladores del propio proyecto representado reconocen en la visualización las estructuras y los procesos con los que están familiarizados.

Según la hipótesis del espejo (MH), la estructura organizativa de un equipo de desarrollo es un reflejo de la arquitectura del software que este equipo desarrolla (Siebra y col., 2014). La visualización de la historia de las interacciones entre los desarrolladores y el programa, tal y como se ha representado, se puede mejorar en el sentido de que no muestra la interacción de los desarrolladores entre sí. Ogawa y Ma (2010) han propuesto un algoritmo que permite una visión más analítica de los datos, visualizando juntos a los autores que formen *clusters* o racimos más o menos cohesionados. El criterio para la detección de racimos de autores sería que hubiesen trabajado sobre los mismos fragmentos del código fuente. A diferencia de las visualizaciones animadas de la evolución en formato de vídeo, el resultado sería estático y tendría un carácter más persistente. Pensamos que es posible, de igual manera, enriquecer esta imagen plana añadiendo la dimensión temporal para crear una animación.

A pesar de lo expresado acerca del conjunto de factores que se han analizado y visualizado, todo indica que el abanico de variables está lejos de agotarse.

9.2. Aplicabilidad

9.2.1. Aplicabilidad de la metodología

Las mismas técnicas de visualización que se han aplicado al software, en un caso de estudio sobre tipografía musical, se pueden aplicar a la propia música para mostrar los procesos compositivos en el tiempo, como se ha demostrado en el apartado de resultados y se ha relacionado en las conclusiones anteriores. Las únicas condiciones son que la partitura esté disponible en un formato de texto como LilyPond (aunque existen otros), y que la historia de las versiones esté registrada con un sistema de control como, por ejemplo, Git.

Si es posible realizar gráficos de alta resolución para un documento académico con gráficos destinados a su impresión en papel, también debe ser posible la producción de estadísticas gráficas para la web, actualizadas frecuentemente para el servicio de los autores del proyecto. Son ejemplos de los trabajos que se realizan en esta línea los sistemas `gitstats`, `StatCVS` y `StatSVN`, que están adaptados, respectivamente, a los sistemas de control de versiones Git, CVS y SubVersion.

Los procedimientos empleados se pueden aplicar para contribuir a guiar la gestión y el mantenimiento de los proyectos de software, sin otra limitación que la de que el código fuente esté disponible.

9.2.2. Aplicabilidad de los resultados

En los apartados introductorios nos planteamos avanzar en la visualización de un ente tan abstracto como el software. Varios eran los objetivos: por un lado, mostrar de manera visual el desarrollo que presenta un proyecto de software libre de tipografía musical. Por otro lado, basándose en los mismos datos y con la ayuda de estas visualizaciones, tratar de generar una confianza en la comunidad educativa, y de los usuarios potenciales, respecto al uso futuro de estas herramientas.

Es necesario reconocer que dicha confianza no vendrá *exclusivamente* de la evolución de las herramientas en el sentido de la madurez, la calidad o la versatilidad del software, ya sea del motor de tipografía LilyPond o de interfaces amigables como Frescobaldi. Será necesario también un esfuerzo orientado al desarrollo de metodologías educativas adecuadas, cultivo de comunidades de usuarios, materiales de referencia disponibles libremente, y otros avances. Algunos de estos procesos están ya modestamente iniciados, como la traducción de la documentación y del propio programa, o el mantenimiento del primer círculo de usuarios en nuestro idioma, pero falta mucho camino aún por recorrer en los otros sentidos.

Estudio 2

**Experimento sobre el desempeño en educación musical
de una herramienta libre para producir partituras**

10. Estudio 2. Objetivos e hipótesis

Con la ayuda de un ordenador y un programa de edición de partituras, un músico o un docente pueden confeccionar sus propios materiales. La herramienta LilyPond, que se ha descrito en los capítulos anteriores, es una opción libre que el músico tiene a su disposición para este propósito. Los desarrolladores de este programa, y también muchos usuarios, afirman que las partituras hechas con él son preferibles, por diversas razones, a otras de distinta procedencia que los músicos manejan habitualmente. Las razones que las hacen supuestamente preferibles pueden ser generales (el criterio no será expresado necesariamente de forma verbal), específicas (como una mayor legibilidad o un mejor equilibrio visual), y las puramente estéticas dentro de la corrección formal.

Dicha afirmación carece de un respaldo académico y de cualquier demostración empírica, hasta el momento, y se ha hecho sin la suficiente objetividad. Ello nos lleva al objetivo principal del estudio.

Objetivo principal

El propósito principal de este estudio es averiguar qué grado de aceptación tienen, entre los músicos, las partituras musicales hechas con LilyPond, en comparación con los originales que han servido de modelo para éstas.

Objetivos secundarios

1. Diseñar y validar un instrumento que permita objetivar este grado de aceptación.

Una posibilidad es tratar de medir la calidad subjetiva de una partitura musical y aplicar este instrumento de medida a las partituras realizadas con el sistema. En lugar de ello, el objetivo se concreta en diseñar un baremo para estimar la aceptación media de un grupo de partituras recreadas, sobre el grupo de partituras empleadas como modelo.

En este diseño interviene una componente fundamental de validación porque al recrear una partitura es lógico pensar que un músico preferirá una copia nueva editada y recién impresa a un modelo fotocopiado de un libro. Al preparar el material para el

experimento es necesario tomar las debidas precauciones para evitar este efecto: las partituras utilizadas sufren un proceso de ecualización orientado a impedir u obstaculizar la identificación a simple vista de la versión original y de la recreada. De esta manera, los participantes podrán seleccionar, en respuesta a una sola pregunta muy simple, la partitura que a su juicio es preferible sin que la calidad de impresión pueda condicionar su elección.

2. Como LilyPond es software libre, se tratará, como objetivo secundario, de dar respuesta a la pregunta de si el software libre de edición de partituras está a la altura de las necesidades de los músicos (profesionales, docentes, estudiantes y aficionados) y/o de la industria de la producción de música impresa.
3. Establecer qué asociación existe entre la preferencia por las partituras elaboradas con LilyPond, sobre otros sistemas, y los factores sociodemográficos (relacionados con la música o no) de la población.
4. Averiguar qué frecuencia de uso de las partituras hacen los músicos, y diferenciar entre los principales soportes como libros, fotocopias, música manuscrita y partituras vistas en una pantalla.
5. Investigar qué herramientas están utilizando los músicos para satisfacer la necesidad de elaborar partituras musicales. Encontrar los programas de edición de partitura más utilizados.
6. Determinar cuál es el origen de los programas que los músicos utilizan.
7. Averiguar qué formación específica han recibido los músicos sobre edición de partituras.

Hipótesis

Se formula la siguiente hipótesis principal.

Las partituras musicales elaboradas con la herramienta de tipografía musical de software libre “LilyPond” presentan un nivel de desempeño comparable al de otros sistemas privativos de edición musical en una selección razonablemente amplia de situaciones reales dentro de los terrenos educativo y profesional.

Como hipótesis de trabajo secundarias se formulan las siguientes:

1. Los músicos emplean partituras musicales diariamente en el ejercicio de su actividad habitual, y acumulan un elevado número de horas semanales dedicado al manejo de éstas.
2. Los músicos, en su actividad ordinaria, emplean partituras fotocopiadas o escaneadas con una frecuencia que iguala o supera al de partituras originales.
3. En los músicos, existe asociación entre la frecuencia de uso de partituras musicales originales y la de partituras fotocopiadas o escaneadas.
4. La actividad de producir partituras mediante algún programa informático es común entre los músicos.
5. En igualdad de otros factores visuales, si se toma una partitura musical existente y se elabora de nuevo mediante LilyPond, los músicos la encuentran preferible o más atractiva.
6. El grado de preferencia por el resultado visual de LilyPond por parte de un músico no está determinado por su nivel de estudios o por otras variables sociodemográficas.

11. Estudio 2. Metodología

11.1. Método

El objetivo principal de este estudio, que es objetivar una pretensión subjetiva como el atractivo visual de una partitura y las preferencias personales de los músicos por un determinado sistema de tipografía musical, da lugar a un tipo de metodología correlacional, con un enfoque cuantitativo, no explicativo.

Concretamente, el enfoque es no explicativo porque no se ha tratado de averiguar *por qué* las partituras hechas con LilyPond son preferibles a los resultados de otros programas, según la opinión de los músicos.

Según el objetivo principal del estudio, se pide la opinión a los músicos, lo que daría lugar a una investigación cualitativa, pero no existen preguntas abiertas, sino que la opinión se registra mediante el uso de un baremo, por lo que el procedimiento de recogida de datos es cuantitativo.

La asociación que se investiga es la que puede existir entre el hecho de que una partitura esté elaborada con el programa en cuestión o con otro diferente, y el hecho de que los músicos prefieran la primera a la segunda cuando ambas se les muestran lado a lado.

La clasificación corresponde a un experimento simple según la clasificación de Montero y León (2007), con grupo único. Todos los participantes se someten al mismo conjunto de casos preparados explícitamente.

Planteamiento

A grandes rasgos, se prepara una lista de modelos de partitura bajo un criterio de representatividad (apartado 11.4.2) que debe incluir, además de tipos de partitura distintos, tanto partituras clásicas comerciales utilizadas en la enseñanza como material pedagógico *ad hoc*. Se registran una serie de datos socio-demográficos que permitirán realizar subgrupos (apartado 11.4.1). Los casos que componen el material del experimento se aleatorizan (apartado 11.3.3) y la identificación de los participantes se anonimiza (apartado 11.3.4). La aplicación del instrumento incluye también el registro de las condiciones de iluminación (apartado 11.7).

11.2. Fases de la investigación

Las fases del procedimiento de investigación son:

- Determinación de la población y la muestra.
- Identificación de las variables.
- Diseño de un instrumento que incorpora mecanismos de aleatorización y anonimización.
- Preparación del material.
- Validación del instrumento (cuestionario y material).
- Recopilación de la información.
- Puesta en práctica.
- Análisis estadístico.

11.3. Población y muestra

Aunque en principio la investigación está limitada geográficamente a la ciudad de Badajoz, los condicionantes de la muestra por conveniencia han llevado la investigación a las poblaciones de Cáceres, Mérida y Villanueva de la Serena.

Los participantes pertenecen a la población de músicos, estudiantes, profesores, compositores y otras ocupaciones relacionadas directamente con la música, y especialmente con el uso de partituras musicales. Se seleccionaron solamente personas con una experiencia en el manejo de partituras superior a 7 años. La muestra no está limitada en modo alguno por otros condicionantes socio-demográficos como la edad o el país de origen.

No existe un censo de músicos o expertos en partituras; la muestra es necesariamente circunstancial, es decir, no aleatoria, y de conveniencia. En el ámbito geográfico de la investigación, se distinguen los siguientes conjuntos de participantes potenciales con su tamaño estimado:

- Profesores de orquesta y de banda:¹ 90
- Profesores de conservatorio elemental/medio/superior: 60
- Alumnos de conservatorio, de diversos niveles: 400
- Profesores de Escuelas municipales de música: 50

Se excluyen los alumnos de escuelas municipales de música y alumnos de conservatorio de grado elemental a causa de que es más improbable que satisfagan la premisa inicial de que

¹ Se les llama tradicionalmente “profesores” por su titulación, pero el perfil de su trabajo dentro de una orquesta los convierte en músicos intérpretes para los efectos de la investigación.

sean usuarios intensivos de partituras. En total, los candidatos que cumplen los requisitos son aproximadamente 600 personas. Desde el punto de vista cualitativo se asume que la población descrita no se diferencia mucho de una similar extraída de otro ámbito geográfico, con una importante premisa a tener en cuenta: que el objeto de la evaluación no son los encuestados, sino las partituras. Los participantes actúan como jueces que aportan su opinión personal, formando de algún modo una parte fundamental del instrumento de medida.

El ámbito geográfico tiene unas fronteras difusas por el hecho de que Badajoz es un foco de atracción musical de cierta importancia, y concretamente por las siguientes circunstancias:

- La ciudad de Badajoz es la sede del Conservatorio Superior de Música “Bonifacio Gil” que es el único centro de este nivel en la comunidad autónoma de Extremadura. El ámbito de influencia de este centro de enseñanza es amplio: acuden a estudiar en él numerosos alumnos procedentes de varias comunidades (además de Extremadura); principalmente, Andalucía y la Comunidad Valenciana, pero también Asturias, Madrid, Castilla-La Mancha, Canarias y la ciudad autónoma de Melilla.
- La Orquesta de Extremadura tiene también su sede en Badajoz, y emplea a profesores de numerosas nacionalidades que son titulados superiores en su totalidad.
- Existe en la ciudad una Banda municipal que ejerce una intensa actividad concertística; de ella forman parte músicos de muy diversa procedencia geográfica que son, en su mayoría, titulados superiores.
- Las Escuelas municipales de música de Badajoz son las mayores de la región.
- El Coro de Cámara de Extremadura reúne a músicos de toda la comunidad y también de las comunidades vecinas, realiza encuentros en la ciudad periódicamente y tiene en ella su sede.

El tamaño de la muestra, con 106 participantes, ha alcanzado algo menos del 20 % de la población estimada (no censada) dentro del ámbito considerado. El error muestral se sitúa en el 8,64 % para la población finita considerada.

La elección de este tipo de muestreo no aleatorio está justificada por una confianza en la imparcialidad de los participantes, que se basa en que ninguno de ellos tiene una relación directa con el programa objeto del estudio y solamente se pide su opinión como usuarios o consumidores del material impreso que el programa produce.

Los lugares de encuesta han sido:

- Domicilios de los participantes.
- Lugares de trabajo.
- Locales de ensayo de orquesta y coro.
- Aulas y otras dependencias de dos Conservatorios de música.
- Aulas y otras dependencias de dos Institutos de educación secundaria.

Las diversas localizaciones ofrecían diferentes condiciones de iluminación, lo que motiva la medición y registro de la luz que incide sobre el material del experimento, en aras de la repetibilidad del mismo.

Para la realización de esta actividad en determinados lugares se han recabado los permisos correspondientes.

11.3.1. Identificación de las variables

La variable explicativa es la opinión de los músicos sobre las partituras consideradas, que se mide mediante un baremo. El baremo cuenta las veces que un participante prefiere ciertos ejemplares sobre un conjunto de parejas de partituras y las veces que manifiestan su indiferencia. El material para esta prueba consiste en diez parejas de partituras representativas de los distintos tipos de partitura musical, compuestas cada una por un modelo, A, y una recreación de este modelo, B. Este número de diez y la naturaleza de dichos modelos se justifican en la sección donde se detalla el diseño del experimento (pág. 203).

La iluminación de la escena es una variable no controlada y perturbadora o contaminante.

La variable independiente es la calidad de las partituras, sobre la que los músicos expresan su opinión.

Las variables dependientes son el desempeño del programa GNU LilyPond para producir partituras, las propias partituras producidas por este programa y el grado de aceptación que aquellas tienen por parte de los participantes.

Así como en el primer estudio se ha estudiado LilyPond como un objeto abierto, del cual se ha analizado su código fuente y la historia del mismo, este estudio se considera en cambio desde el punto de vista externo, como un objeto cerrado y monolítico; no se cuestionan sus interioridades ni su método de uso o de funcionamiento, aunque sí se documenta detalladamente el proceso de creación del material utilizado para el experimento que formará parte del instrumento para obtener los datos.

11.3.2. Diseño y validación del instrumento

Se caracteriza el instrumento utilizado como una encuesta clasificada de la forma siguiente:

- Según la finalidad, el instrumento satisface dos objetivos principales distintos que determinan dos caracterizaciones diferentes. Las secciones que tienen el objetivo de identificar a la población, son descriptivas. La sección que tiene como objetivo evaluar de

forma comparativa las partituras producidas por un determinado método, es evaluativa.

- Según su contenido, recopila información sobre hechos u opiniones, dependiendo de la sección considerada.
- Según el procedimiento de encuesta, es personal. No cabe la realización autónoma por las características del material utilizado.
- Según su dimensión temporal, es transversal, pues no se analiza una tendencia en el tiempo, sino la realidad en un momento determinado, aunque puede formar parte de un diseño longitudinal posterior. Tampoco explica la causa del fenómeno.

El instrumento y el material se someten a validación y a una prueba de fiabilidad. Por un lado se valida la pertinencia, el constructo y el diseño general, tanto del cuestionario como del material del experimento, con un panel de jueces. Por otro lado, se lleva a cabo una prueba piloto y para la fiabilidad se emplea el método de test-retest sobre un subgrupo.

11.3.3. Aleatorización

El material se compone de dos juegos completos diez parejas de partituras representativas, cada uno; el objetivo de la aleatorización es eliminar el efecto que pudiese tener, dentro de una pareja, el hecho de que una u otra versión esté situada en uno de los dos lados, o que se presente al principio o al final de la prueba. Así pues, las opciones son aleatorizar la colocación izquierda/derecha de los diez modelos, o el orden de los casos, o ambos a la vez, para cada participante.

Se ha diseñado un esquema de compromiso respecto al orden y la posición en que se presentan los elementos. El compromiso consiste en que la posición izquierda/derecha de los modelos A y B para cada uno de los diez casos, es aleatoria y distinta para cada encuestado, dando dos variantes A-B y B-A para cada caso; en cambio, la presentación de las propias parejas sigue un orden fijo de 1 a 10. Se rechaza la posibilidad de emplear un segundo nivel de aleatorización según el cual el orden de presentación de los diez casos es aleatorio también, después de tener en cuenta la relación coste/beneficio de este esquema. La idea es reducir el tiempo de preparación de cada experimento para poder repetirlo varias veces seguidas sin hacer esperar demasiado a los voluntarios.

Para cada uno de los diez casos se preparan las dos versiones A-B y B-A. En el reverso del soporte de cartón se etiqueta cada elemento con su número y la variante. La figura 60 muestra, esquemáticamente, los 40 modelos necesarios, distribuidos entre los dos juegos de diez parejas cada uno.

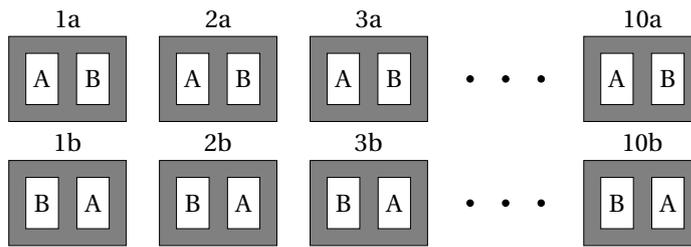


Figura 60: Juegos “A” y “B” para los diez casos, ordenados.

Las combinaciones posibles para las colocaciones izquierda/derecha² en los diez casos son $2^{10} = 1024$. Llamando “a” a los casos que tienen el ejemplar original a la izquierda y el ejemplar recreado con LilyPond a la derecha, y “b” a los que tienen el recreado a la izquierda, las 1024 combinaciones van desde “aaaaa aaaaa” hasta “bbbbbb bbbbbb”. Se selecciona un subconjunto de estas combinaciones que cumplen la restricción de que existen tantos casos a como b. Esta reducción hace que la colocación de cada ejemplar no sea completamente independiente, pero tiene la ventaja de que es más estable porque al finalizar cada encuesta habrán aparecido ejemplares A y B el mismo número de veces a la izquierda y a la derecha.

Para generar una combinación aleatoria de posiciones izquierda/derecha para los diez casos, se ha elaborado un programa en lenguaje Python³ que simula la extracción de diez bolas aaaaabbbbb sin reposición. La salida de este programa es una lista suficientemente amplia de combinaciones con 5a+5b que se van utilizando, de una en una, durante la preparación de la encuesta, y marcándolas como utilizadas para cada encuesta que se realiza. La lista tiene la forma que aparece en la figura 61.

Nº comb.	Directa	Nº comb.	Complementaria
001:	b a b a b a b b a a	002:	a b a b a b a a b b
003:	b b a b b a a a b a	004:	a a b a a b b b a b
005:	a a b b a b b a b a	006:	b b a a b a a b a b
007:	b a b b a a b a a b	008:	a b a a b b a b b a
009:	b a a a b a b b b a	010:	a b b b a b a a a b
...

Figura 61: Lista de combinaciones utilizada para aleatorizar la posición de los modelos.

En esta lista se indica, en las dos primeras columnas, el número de combinación y las diez variantes a/b directas que le corresponden; como el conjunto completo de modelos incluye

²Las posibilidades para dos colocaciones sobre n elementos, son las variaciones con repetición de 2 elementos tomados de n en n , y su número es $VR_2^n = 2^n$. En nuestro caso, $n = 10$. Se obvia, por simplicidad, el hecho de que existen dos casos de control que no requieren esta aleatorización.

³Una versión simplificada del programa en Python mencionado es el siguiente, que imprime diez combinaciones directas:

```
import random
for i in range(10):
    for a in random.sample('aaaaabbbbb',10):
        print a,
    print
```

una variante a y otra b para cada caso, al preparar una combinación directa según la tabla, se obtiene también una combinación inversa o complementaria con los modelos sobrantes, que puede utilizarse para la siguiente encuesta. En las dos columnas siguientes de la tabla aparecen el número de combinación y las diez variantes b/a complementarias de la combinación anterior. De esta manera es inmediato preparar dos juegos de elementos a partir de esa fila. Un ejemplo de ello es la figura 62.

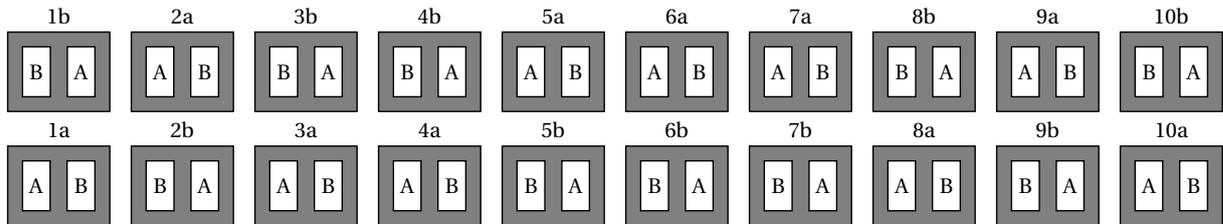


Figura 62: Dos juegos de elementos, directo y complementario, con el orden izquierda/derecha aleatorizado.

Una vez preparado el juego de elementos para un experimento, en la hoja de recogida de datos se anota el número de la combinación, lo que deja constancia de las posiciones en que se han presentado los elementos para poder repetirlos en la fase de retest. La forma de registrar la posición de los elementos A y B se ve en el modelo de encuesta. Para cada respuesta se marca la casilla A o B y la posición que ocupa, Izquierda o Derecha. Los datos recogidos de esta manera se pueden contabilizar sin ningún proceso intermedio de decodificación.

11.3.4. Anonimización

Para mantener la confidencialidad de los datos y preservar el anonimato de los participantes, se ha ideado un sistema que evita reconocer al mismo a partir de la hoja de recogida de datos. Se ha preparado una hoja con una lista de identificadores aleatorios alfanuméricos en cantidad suficiente, creada con un programa escrito en el lenguaje Python.⁴

De la misma manera en que se hizo para las combinaciones izquierda/derecha, en la generación de esta lista de identificadores se ha empleado un algoritmo que simula la extracción de bolas numeradas de una bolsa, de forma que no se repita ningún código. Para el número de participantes esperado, se calcula que combinaciones de tres caracteres hexadecimales, o sea, códigos que pueden ir desde 000 hasta FFF, son suficientes para marcar un máximo de $16^3 = 4.096$ encuestas.

Para cada nuevo entrevistado, seguimos el siguiente procedimiento:

⁴El siguiente programa simplificado en Python tiene la misma funcionalidad e imprime diez identificadores:

```
import random
for i in random.sample(range(4096), 10):
    print hex(i)
```

- Elegir el siguiente código de la lista que aún no se ha utilizado.
- Marcarlo o tacharlo como utilizado y anotarlo en la casilla “Identificador” de la hoja de recogida de datos.
- Anotar el código junto al nombre del participante en un archivo de texto para referencia futura (p.ej., para la fase de retest de la prueba de estabilidad del instrumento). Este archivo es confidencial.
- Después de cumplimentar la hoja, se archiva en orden de código para poderla localizar más tarde en caso necesario, según la lista confidencial que se ha preparado.

Véase en la figura 63 el aspecto de la hoja confidencial de nombres reales de los participantes, que relaciona a cada uno con su código anónimo.

Código anónimo	Nombre real
7AE	_____
FF7	_____
208	_____
188	_____
7BC	_____
64A	_____
...	...

Figura 63: Hoja de correspondencias de códigos anónimos y nombres reales.

11.4. Instrumento

11.4.1. Diseño del cuestionario

El instrumento utilizado es una encuesta de tipo personal. El cuestionario tiene 12 preguntas agrupadas en cuatro secciones que aparecen a continuación. De ellas, las tres primeras contienen factores que permiten la elaboración de subgrupos.

1. Datos socio-demográficos. Preguntas 1 a 6.
2. Uso de partituras musicales. Preguntas 7 y 8.
3. Uso de software de edición de partituras. Preguntas 9, 10 y 11.
4. Experimento comparativo de modelos de partitura y sistemas de tipografía musical. Pregunta 12.

La sección primera persigue el objetivo de clasificar al encuestado en función de la actividad o actividades que le relacionan con la música.

La sección segunda está orientada al objetivo de probar que la partitura es un material omnipresente en la práctica musical y que los músicos emplean un gran número de horas al manejo de partituras musicales.

La tercera sección responde al objetivo de averiguar qué sistemas emplean habitualmente los músicos para la elaboración de partituras musicales.

La cuarta sección está constituida únicamente por la pregunta que está orientada al objetivo principal del estudio, y consiste en recabar la opinión de los encuestados comparando un conjunto de partituras con recreaciones de las mismas elaboradas con LilyPond.

Se registran también el código anónimo (no identificador) del encuestado, la fecha y las condiciones de iluminación de la escena, como se ha descrito en la exposición del procedimiento.

La relación de preguntas se puede ver a continuación. Se indica en cada una el tipo de escala y de datos para ese campo, si las opciones son excluyentes, si es de respuesta múltiple y si hay opción nula.

Sección primera: datos demográficos

1. Edad

Pregunta identificativa. Valor numérico, entero. Posibilidad de No contesta (NC).

2. Sexo (M/F)

Pregunta identificativa. Cerrada y dicotómica. Abierta a NC.

3. Dominancia lateral (diestro/zurdo/ambidextro)

Pregunta identificativa. Tres opciones cerradas y excluyentes. Se refiere a la dominancia lateral declarada por el participante, pero no se comprueba.

4. ¿Utilizas graduación para la vista en tus estudios o durante tu trabajo? (Sí/No)

Pregunta cerrada y dicotómica. Por su contenido es identificativa asumiendo que la respuesta se refiere al momento actual.

5. Por favor, indica qué actividad te define mejor en el campo de la música.

Pregunta cerrada y de respuesta sugerida (se proponen unas opciones). Respuesta única, con opción abierta al final.⁵

- a) Soy músico, instrumentista o cantante.
- b) Director de orquesta, banda, coro, conjunto instrumental o vocal.
- c) Compositor o arreglista
- d) Profesor de música, en cualquier nivel.

⁵Se espera que la respuesta abierta se pueda cerrar fácilmente a posteriori.

- e) Estudiante de música, en cualquier nivel.⁶
- f) Soy aficionado a la música.
- g) Otro tipo de relación con la música. ¿Cuál?

6. ¿Qué nivel de estudios musicales has completado?

Pregunta cerrada a las opciones mutuamente excluyentes que se proponen. Respuesta única.

- a) No tengo estudios formales de música.
- b) Estudios de música no reglados (escuela municipal, academia...)
- c) Estudios de conservatorio de nivel elemental.
- d) Estudios de conservatorio de nivel profesional.
- e) Estudios superiores de música.
- f) Posgrado o doctorado en música.

Sección segunda: uso que haces habitualmente de material de notación musical (partituras)

7. En tu actividad musical o en la enseñanza ¿con qué frecuencia utilizas estos tipos de partitura? Escoge para cada tipo entre:

- 0 (nunca o menos de una vez al año),
- 1 (una o más veces al año, y menos de una vez al mes),
- 2 (una o más veces al mes, y menos de una vez a la semana),
- 3 (una o más veces por semana, pero no diariamente),
- 4 (normalmente una o más veces al día)

Cinco indicadores excluyentes. Es una pregunta de tipo cerrado; el entrevistador lee las opciones, que cubren una escala de todas las frecuencias desde “nunca” hasta “diariamente”. La escala se aplica a cada uno de los siguientes casos:

- a) Partituras comerciales originales, en papel.
- b) Partituras escaneadas o fotocopiadas.
- c) Partituras realizadas a mano, en papel (originales o copias).
- d) Partituras realizadas a mano en pizarra tradicional o electrónica.
- e) Partituras editadas mediante ordenador, hechas por otra persona.
- f) Partituras editadas mediante ordenador, hechas por mí mismo.
- g) Partituras proyectadas o en la pantalla de un ordenador, tableta o libro electrónico.
- h) Algún tipo de partitura que no pertenece a ninguno de los anteriores grupos.

Incluye: libros o partituras didácticas, piezas sueltas, estudios, ejercicios, exámenes, pequeños fragmentos, particellas, guiones o partituras de solista, orquestales y similares.

No incluye: musicogramas, pictogramas, lenguaje de signos.

⁶El nivel de estudios completados se detalla en otra pregunta.

8. Durante una semana de tiempo en tu actividad habitual ¿cuántas horas dedicas a la lectura, escritura, anotación o análisis de partituras musicales, de cualquier tipo?

Valor numérico, entero. Se pide al encuestado un número aproximado de horas haciendo énfasis en que son horas *totales* durante una semana típica, y para cualquier actividad de relación directa con la partitura.

Sección tercera: uso de software de edición de partituras y formación en el mismo

9. Si has elaborado alguna vez partituras musicales mediante ordenador, ¿qué programa o programas has utilizado, y con qué frecuencia cada uno de ellos? Escoge para cada programa entre

- 0 (nunca o menos de una vez al año),
- 1 (una o más veces al año, y menos de una vez al mes),
- 2 (una o más veces al mes, y menos de una vez a la semana),
- 3 (una o más veces por semana, pero no diariamente),
- 4 (normalmente una o más veces al día)

Cinco indicadores excluyentes, como la pregunta anterior. La escala se aplica a cada una de las opciones siguientes. Opción abierta al final.

- a) Finale.
- b) Sibelius.
- c) Encore.
- d) Musescore.
- e) LilyPond.
- f) Servicios online o aplicaciones de tableta o "smartphone".
- g) Otro. ¿Cuál?

10. Tú, tu empresa o tu centro educativo ¿habéis adquirido o instalado algún programa de edición de partituras?

- a) Sí. ¿Cuál?
- b) No.
- c) No lo sé.

Tres opciones cerradas, excluyentes y opción abierta condicionada a la respuesta "Sí".

11. ¿Has recibido alguna vez algún tipo de formación específica para la edición de partituras con medios informáticos?

- a) Sí. ¿Para qué programa o programas?
- b) No.

Pregunta dicotómica. Dos alternativas cerradas, más una opción abierta condicionada a la respuesta "Sí".

Sección cuarta: experimento comparativo

En la cuarta sección se enfrenta LilyPond a una selección de sistemas diferentes para producir partituras. Al medir el grado de aceptación de las partituras de LilyPond por parte de los músicos, pidiendo a éstos que las comparen con sus modelos de referencia y seleccionen una de ellas, no se evalúa el contenido musical, ya que es el mismo en ambos, sino que se mide implícitamente una combinación de factores:

- Su corrección técnica y formal
- La distribución espacial de los elementos de notación sobre el papel
- Su facilidad de lectura
- La belleza de la partitura

A su vez, la facilidad de lectura puede venir determinada por el tamaño de la grafía; la belleza puede obedecer a convenciones culturales; la relación entre el pigmento y el fondo blanco (a través del grosor de las líneas y el diámetro de las cabezas de nota) puede influir en la claridad de la notación en su conjunto. La distribución espacial puede ser óptima en una sección de la partitura y deficiente en otras. En suma, están en juego muchos factores que pueden conllevar un alto grado de subjetividad. Por estas razones, la estadística aporta la necesaria objetividad a la respuesta (si la hay) a la pregunta de qué partitura es preferible.

La recopilación de información genera un total de 60 columnas de datos. El campo 7, *actividad principal que define al voluntario en su relación con la música*, de tipo numérico, en principio codificaba una variable nominal en letra minúscula desde “a” hasta “g” y más tarde en respuesta múltiple, codificando en binario⁷ varias actividades. Esta solución no es práctica porque requiere una decodificación posterior. En cambio, se ha desglosado la variable en tantas columnas como opciones y se ha fijado cada una como nominal del tipo sí/no.

El campo 38, *combinación*, de tipo numérico, se registra para repetir la misma secuencia de posiciones A/B de los diez elementos en la prueba de estabilidad del instrumento mediante test-retest, como se detalla en el apartado 11.3.3 (página 195).

Las 60 columnas de datos se detallan en la tabla 12.

⁷El sistema de numeración binario permite convertir una serie de opciones sí/no en un solo número, y viceversa.

lo indican los metadatos del archivo. En otros casos solamente podemos suponer que están hechos con algún otro programa, como en el caso de las partituras comerciales impresas.

Para cada ejemplar A de una partitura original, se elabora un ejemplar B recreado con el programa LilyPond, cuyo desempeño es el objeto de este estudio (figura 65). Los dos ejemplares A y B forman una pareja o caso.



Figura 65: Representación esquemática de los dos elementos “A” y “B” de una pareja o caso.

El experimento se realiza cada vez con uno de los juegos de diez parejas, que se presentan como muestra el esquema de la figura 66.

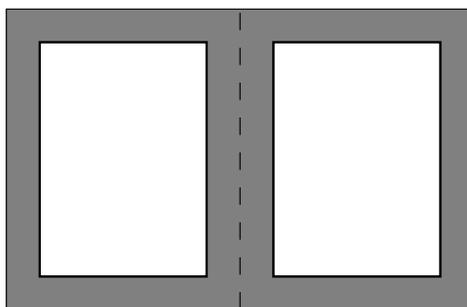


Figura 66: Esquema simplificado de presentación de un caso.

Los participantes examinan los casos uno a uno, e ignoran cuál es A y cuál es B. La procedencia de las partituras es desconocida para el participante y no es un dato explícitamente constatado en ella.⁸ Lo único que tienen en común es que todas son ejemplos reales de utilización de música en el ámbito educativo, como ensayos de orquesta o coro de alumnos, material de trabajo para una asignatura, o partituras comerciales para uso en conservatorios.

Se pide al encuestado que elija una de las dos partituras enfrentadas lado a lado, con la siguiente pregunta:

“Si tuviese usted que trabajar con una de estas dos partituras, ya sea para leerlas, estudiarlas o analizarlas, ¿cuál de ellas preferiría? Elija la A, la B o diga «cualquiera de las dos»”.

Se trata ahora de seleccionar una serie de modelos que cumplan el propósito de cubrir diferentes estilos de partitura, de distintos niveles de complejidad, y de procedencias distintas, y tan-

⁸Algunos músicos muy experimentados pueden reconocer la editorial de una partitura a simple vista. Esto no sería un problema para el experimento si no fuese porque, al mismo tiempo, algunos músicos declaran preferir alguna editorial sobre las demás. Si la editorial de los modelos de la prueba fuesen claramente reconocibles, ello falsearía los resultados, ya que existiría una cierta probabilidad de que el músico eligiese un modelo al reconocer en él su editorial preferida. Así pues, para validar el experimento debemos preguntar a los jueces si la editorial es reconocible. Esto no elimina la eventualidad expuesta, pero al menos debe servir para que conste que se ha hecho lo posible por evitarla.

to de uso didáctico como partituras comerciales en papel y sean representativos de algunas de las partituras más utilizadas en la práctica cotidiana del músico, el docente, el estudiante o el investigador. Para ello se han formado dos grupos de casos.

El **grupo 1** lo forman cinco tipos de partitura que se han usado como material didáctico, de uso en un centro educativo o para ensayos de orquesta o de coro de alumnos, y que han sido elaborados por docentes que no son profesionales de la composición tipográfica de partituras musicales, o bien que son partituras de dominio público, pertenecientes a la llamada Cultura Libre:⁹ partituras no comerciales y de libre distribución. Son partituras realizadas con algún programa de edición musical (nunca a mano) y no son ediciones comerciales.

El **grupo 2** lo forman otros cinco tipos de partitura que en este caso son comerciales, ya sean impresas en papel, escaneadas o fotocopiadas, o publicaciones electrónicas en formato PDF. Pueden ser actuales (sujetas a *copyright*) o en el dominio público.¹⁰

Mediante los dos grupos de partituras se trata de cumplir la premisa de que las partituras que conforman el experimento representan el amplio abanico de los materiales existentes, desde lo más efímero hasta lo más formal.

En cada uno de los dos grupos hay un elemento **de control**: son los elementos **Control 1** y **Control 2**. Dentro de cada uno de estos dos casos, los dos elementos de la pareja son idénticos y tratan de elevar la fiabilidad del experimento invitando tácitamente¹¹ al encuestado a manifestar su indiferencia por un elemento determinado. El propósito de los elementos de control es examinar el impacto de la dominancia lateral declarada por el voluntario: diestro, zurdo o ambidextro, que podría hacer que los participantes tuviesen una cierta tendencia a preferir los elementos de su derecha o de su izquierda. Los elementos de control son dos para que cada grupo tenga un elemento. El análisis del impacto de este factor de dominancia lateral solo es posible si hay un número significativo de personas que se declaran como zurdas o ambidextras.

Para la elaboración del experimento nos acogemos al equivalente, dentro de la legislación española en materia de propiedad intelectual, de las excepciones de uso legítimo para fines de investigación, enseñanza, crítica o parodia de fragmentos o partes no significativas de obras protegidas por derechos de autor, que existen en el Derecho anglosajón. Esto es válido también para las partituras del primer grupo, cuyo autor es, igualmente, propietario de los derechos de su obra. Sin embargo ninguna de ellas ha sido publicada, sino que se han distribuido gratuitamente, lo que entendemos que abre aún más la excepción a los efectos del estudio.

⁹En ningún caso debe confundirse la Cultura Libre con la distribución no autorizada de material protegido por el derecho de autor y las leyes de Propiedad Intelectual.

¹⁰Las partituras musicales están sujetas a *copyright* desde el momento de su creación, y no pasan a pertenecer al dominio público hasta setenta años después de la muerte del autor según la convención de Berna (OMPI, 1989) y las sucesivas reformas de los plazos del *copyright*.

¹¹Tácitamente, porque el investigador no revela al participante que se encuentra ante un caso de control.

Véase Vives (2006) donde se detalla la relación entre la doctrina del *fair use* o la “excepción de uso legítimo” del Derecho anglosajón y el régimen europeo típico de protección de la propiedad intelectual.

Los diez casos están organizados como se ve en el resumen simplificado de la figura 67.

Grupo		Tipo	A	B
1	(material docente)	1	A	B
"	"	2	A	B
"	"	3	A	B
"	"	4	A	B
Control 1	"	5	A	B
2	(partituras comerciales)	6	A	B
"	"	7	A	B
"	"	8	A	B
"	"	9	A	B
Control 2	"	10	A	B

Figura 67: Esquema de la clasificación de los diez casos del material.

El proceso de selección de la partitura modelo para cada uno de los tipos está documentado en la sección 11.5, pág. 209 y ss.; la composición definitiva de los diez casos puede verse en la tabla 23 (pág. 223). Se han establecido unos límites para el estilo y la época de la notación musical empleada en el material del experimento: es notación musical convencional, de la práctica común occidental que quedó consolidada durante el s.XVIII y que se empleó con pocas variaciones hasta la primera mitad del s.XX. Están excluidas, por tanto, la notación neumática gregoriana, la mensural renacentista, y la notación gráfica contemporánea, así como la tablatura y otras notaciones menos usadas.

La muestra de originales cubre un amplio abanico de tipos de partitura, algunos de los cuales posiblemente resulten familiares a algunos participantes, mientras que otros serán algo más inusuales para él o ella. De esta forma se somete, tanto al programa como a los participantes, a una variedad de situaciones moderadamente amplia y representativa de los estilos de notación más comúnmente usados. En un primer esbozo se han considerado los siguientes tipos:

- Particella o partitura para solista¹² (p.ej: para violoncello). Un pentagrama, en clave de Fa.
- Partitura para piano. Dos pentagramas.
- Hoja guía de acordes (*lead sheet*). Un pentagrama, nombres de acordes y diagramas de posiciones para guitarra.
- Partitura de percusión. Pautas de una a cinco líneas; varias posibilidades.

¹²La diferencia entre una particella y una partitura para solista es aquella es una parte dentro de un conjunto y esta última es una parte principal. Lo que tienen en común es que en ambos casos se trata de un solo pentagrama.

- Ejercicio de armonía: un bajo cifrado.
- Ejercicio de lenguaje musical.
- Partitura rítmica, sobre pentagrama único o pautas de una línea.
- Partitura de cuarteto de cuerda. Cuatro pentagramas.
- Canción a una voz, con texto.
- Partitura para instrumento de viento transpositor.
- Partitura de música de cámara con piano. Sistema de piano y dos o tres pautas más.
- Partitura coral, a cuatro voces SATB¹³ sobre sendos pentagramas.
- Partitura para orquesta sinfónica, 13 o más pentagramas.

El problema a la hora de seleccionar un conjunto de partituras para el experimento, es el de buscar un equilibrio entre el tamaño del conjunto, que no debe ser grande para maximizar el número de encuestas, ni demasiado pequeño para que abarque un conjunto representativo de tipos de partitura. Al tratar de hacer representar una variedad de estilos de partitura suficientemente amplia, nos encontramos con que desearíamos presentar a los encuestados quince o veinte partituras distintas, lo que habría aumentado el tiempo total dedicado a la prueba.

En una primera revisión, la lista anterior se reducirá a un ejercicio de armonía; un ejercicio de lenguaje musical; una canción con acordes en cifrado americano (posiblemente del *Real Book*¹⁴ o un *estándar* de jazz); una partitura coral; una partitura fácil para piano; una partitura de orquesta para cello; una partitura orquestal; una partitura de música de cámara; y una partitura para instrumentos de viento transpositores.

Esta lista de nueve casos necesita ahora distribuirse en dos grupos de igual número, entre partituras “rápidas” de uso didáctico, y partituras “serias” en papel. Como consecuencia, se determina que el número de casos debe ser par.

En el siguiente ciclo de revisión se determina que faltan en la lista anterior tres tipos de partitura bastante característicos, que son las canciones (con texto), las obras de percusión y las piezas de solista en un solo pentagrama, como la guitarra.

Por otra parte, son menos interesantes los ejercicios de armonía y lenguaje por ser demasiado efímeros, y el modelo destinado a instrumentos de viento porque no aporta suficientes elementos que lo haga distinguirse del resto por su aspecto visual.

Después del último ciclo de revisión se determina que el número de casos que cumple los compromisos que se han establecido, es de diez elementos:

¹³L abreviatura SATB corresponde a Soprano, Alto, Tenor y Bajo, según la terminología usual de la música coral. Son las cuatro voces clásicas de la armonía a cuatro partes y de la polifonía vocal.

¹⁴*The Real Book* es una conocida recopilación de hojas guía de acordes de canciones conocidas, con sus melodías.

1. Una *particella*
2. Una pieza para piano
3. Un *lied* o canción
4. Una obra coral
5. Una parte de percusión
6. Una hoja guía con cifrado americano
7. Una obra para guitarra
8. Un cuarteto de cuerda
9. Una obra de música de cámara (solamente el guión en Do para el pianista)
10. Una página de orquesta sinfónica

Y de ellos, los cinco primeros forman el grupo 1 y los cinco últimos son el grupo 2. Por su parte, los elementos de control son los números 5 (perteneciente al grupo 1) y 10 (del grupo 2). Durante el proceso de diseño se consideró un estudio alternativo más ambicioso, que quedase liberado de algunas de las restricciones del experimento tal y como se ha diseñado, pero se expone aquí el compromiso final alcanzado.

Es necesario evitar que el participante seleccione la partitura recreada por motivos completamente ajenos a la hipótesis que mueve el estudio. Entre las situaciones que se han evitado y que habrían falseado los resultados del estudio, están las siguientes:

- Que el papel del material original sea de color crema o amarillento, mientras que el modelo recreado estará impreso en papel blanco.
- Que, estando ambas en papel blanco, la partitura recreada esté impresa sobre un papel más blanco que la original, haciéndola parecer más nueva.
- Que la partitura original presente un cierto grado de borrosidad u otra circunstancia que la hiciese difícil de leer.
- Que, por las mismas razones, el original presente manchas o indicaciones escritas, a diferencia de la versión recreada, que sería a buen seguro un ejemplar recién impreso, limpio, nítido y sin signos de desgaste por el uso.
- Que el tamaño de la notación musical fuese diferente.
- Que, a igual tamaño, las indicaciones de notación musical tuviesen un distinto contenido.
- Que a pesar de presentar el mismo contenido y ser éste del mismo tamaño, estuviese presentado o dispuesto de distinta forma, hasta ciertos límites. Por ejemplo, que los márgenes difieran o que la música esté más *apretada* o *densa* tanto en el sentido horizontal como en el vertical.

Para evitar estas situaciones, se toman las siguientes precauciones.¹⁵

¹⁵Se detallan para cada modelo particular dentro del apartado correspondiente, al explicar el proceso de preparación del material.

- Los dos ejemplares, original y recreado, se presentan en hojas de papel blanco idénticas.
- Si el original es un documento original de algún programa de edición musical, o existe en forma de documento PDF, se obtiene una copia hecha con la misma impresora y sobre el mismo papel que la versión recreada.
- Si el original es una partitura comercial en papel, se digitaliza con un escáner a una elevada resolución. La versión recreada, por su parte, se imprime y posteriormente se digitaliza con el mismo escáner y a la misma resolución. Después de imprimir las dos versiones escaneadas, tendremos que las inevitables imperfecciones que introducen el escáner y la impresora, están “a ambos lados de la ecuación” y no introducen, por sí solos, ninguna asimetría en el aspecto de los dos ejemplares.
- Se busca un tamaño de pauta musical, o pentagrama, que coincida exactamente con el del modelo.¹⁶
- Se miden cuidadosamente los márgenes del original y se produce un documento recreado que presenta los mismos márgenes que aquél.
- Si es necesario, se fuerza manualmente la distribución de los compases sobre los sistemas,¹⁷ y el número de sistemas por página.
- En la fase de montaje del material, para la colocación de los ejemplares se emplea el mencionado proceso de aleatorización.

Algunas de estas medidas suponen la corrección manual o sobreescritura de la distribución automática de LilyPond. LilyPond es un sistema de edición de partituras automático, y una de las propiedades que sus desarrolladores han tratado de implementar es la de ser capaz de producir resultados que sean *lo mejores posible* con poca o ninguna intervención manual. La fuente principal de problemas de recreación de las partituras modelo para confeccionar el material, es que la disposición de los elementos musicales que resulta de este proceso automático, puede ser muy diferente a la del modelo. Así pues, para elaborar el material debe hallarse un equilibrio entre dos extremos: conservar la disposición predeterminada, o ajustar manualmente los elementos hasta *clonar* el aspecto del modelo.

11.5. Preparación del material

Finalizado el proceso de composición de la lista de tipos para garantizar que las partituras cubren tipos y orígenes diversos, se procede a seleccionar los propios modelos. No existe ningún censo de partituras musicales que incluya todos aquellos materiales que los estudiantes

¹⁶De la misma forma en que ocurre con la tipografía convencional de texto, en la notación musical se produce una paradoja cuando se comparan los tamaños de la grafía en dos partituras diferentes, y es que el tamaño visual aparente no coincide necesariamente con las medidas físicas. Véase la figura 103 en el apartado 12.2 (pág. 263) dentro del capítulo de resultados.

¹⁷En notación musical, un sistema es una línea de música que se lee de izquierda a derecha, con uno o más pautas simultáneas.

o los profesores de música han podido utilizar alguna vez en sus clases. Desde el punto de vista formal, la selección de partituras constituye una muestra de conveniencia.

Al seleccionar un modelo registraremos los siguientes parámetros:

- Tipo de partitura según el listado que se ha preestablecido.
- Identificación: título, autor de la composición, editor.
- Licencia de uso, estatus de *copyright*.
- El medio en que se recoge: papel o documento electrónico.
- Grupo al que pertenece: si es de uso *amateur* o profesional.
- El medio que se ha empleado para su realización, si es conocido (y no siempre lo es). Nombre del programa que lo ha generado, que puede en ocasiones verse en los metadatos.
- Autoría del tipografiado musical.
- Elementos distintivos desde el punto de vista de la notación musical, que determinarán las técnicas de recreación pertinentes.
- Otros detalles visuales del modelo, que deberán asimismo ser recreados.

11.5.1. Selección de los modelos: grupo 1. Material didáctico, no profesional o no comercial

Tipo 1 - Particella

Una «particella» es semejante a una partitura para solista y contiene la notación de una parte individual. Para el caso número uno se previó como candidata a este tipo una partitura para violoncello, en clave de Fa, porque esta clave es menos reconocible que la de Sol. Se ha dicho de LilyPond que su clave de Sol es característica¹⁸ y, si bien no podemos eliminarla de los ejemplos porque es, con mucha diferencia, la clave más usada en toda la notación musical, al menos podemos evitarla en el ejemplo de partitura más simple del conjunto de los diez casos.

El violoncello es un instrumento de sonido grave que se escribe comúnmente en clave de Fa y ocasionalmente también en las claves de Do en 4ª y de Sol para las notas agudas o muy agudas, respectivamente.

Un buen ejemplo de *particella* de nivel elemental para violoncello, es el que se distribuyó a los componentes de una orquesta escolar en un instituto de Secundaria de la ciudad de Badajoz. Podemos considerar este modelo como material didáctico y se inserta por tanto dentro del grupo 1.

¹⁸Con el desarrollo de los mecanismos para cambiar el tipo de fuente tipográfica, LilyPond ya no está tan caracterizado por la forma de su clave de Sol como solía estarlo hasta octubre de 2009 en que J. Nieuwenhuizen implementó esta funcionalidad.

Como se ha dicho, contiene un solo pentagrama; el trabajo está realizado con un editor desconocido, por un docente no profesional de la edición. Se trata de una fotocopia de calidad baja, algo borrosa, procedente de un original probablemente hecho con impresora de inyección de tinta. La pieza es “Caresse sur l’ocean”, de la película “Los chicos del Coro”. Se utiliza aquí sin permiso un pequeño fragmento de una página para uso académico.

Las características de este modelo parecen anunciar ciertos problemas que se discutirán en la página 227 donde se documenta el proceso de recreación. Los elementos más característicos del modelo 1 se resumen en la tabla 13.

Tabla 13: Elementos gráficos característicos del modelo 1.

Nombre	Ejemplo
Clave de Fa	
Indicación numérica del compás	$\frac{4}{4}$

Tipo 2 - Partitura para piano

Las partituras normales para piano tienen dos pentagramas en las claves de Sol (mano derecha, pentagrama superior) y de Fa (mano izquierda, pentagrama inferior), unidos por una llave. Se ha seleccionado una página intermedia de la sonata número 3 en Do menor, Op.35, obra de nivel de dificultad medio de Jan Ladislav Dussek (Dussek, 2012) que circula entre los estudiantes de piano de conservatorio. Aunque se presenta en papel, para obtener una mejor calidad recurrimos al documento original en forma electrónica, prescindiendo de la fotocopia. Ello tiene la ventaja de que nos hará ahorrar un paso en el proceso de preparación del caso para el experimento.

La justificación de esta elección y su inclusión en el grupo 1 tiene su base en los siguientes puntos: el modelo seleccionado procede de una colección de música de dominio público; Dussek es un autor habitual en los métodos de piano para los primeros cursos de conservatorio; es una partitura de libre descarga; está elaborado por un editor aficionado, con un programa editor poco común. No puede decirse que sea una hoja de música situada en el mercado.

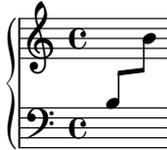
A juzgar por los metadatos del documento (que en esta ocasión no se encuentran dentro del mismo archivo), la composición tipográfica está publicada bajo una licencia *Creative Commons Attribution Share Alike 3.0* y es obra de Jean-Pierre Coulon, fechada en 2012, utilizando el programa MusiXTeX.

El programa MusiXTeX es una mejora de MusicTeX y no es un antecesor de LilyPond, ya que se creó en 1997 y aún están en uso versiones recientes, mientras que el nacimiento de LilyPond se hizo partiendo de cero en una fecha anterior, 1996. El desarrollo de ambos proyectos ha

discurrido de forma paralela. Sin embargo, es cierto que comparten algunos aspectos como la interfaz basada en texto, y que MusiXTeX utiliza como substrato el sistema de composición tipográfica TeX, como lo hacía LilyPond en sus orígenes.

La partitura presenta los siguientes elementos que merecen algo de atención respecto a su posterior recreación: emplea voces que cambian de pentagrama; contiene indicaciones de matiz centradas en el sistema de piano; algunos pasajes requieren para su notación más de una voz en una de las manos; y por último, existen barras de corchea en forma de codo. Véase la tabla 14.

Tabla 14: Elementos gráficos característicos del modelo 2.

Nombre	Ejemplo
Tresillos	
Cambio de pentagrama y barras en codo	
Matices	<i>pp.</i> <

Tipo 3 - Canción a una voz, con texto y acompañamiento de piano

Para el modelo del tipo 3, cuya existencia está justificada por la presencia de texto, se ha recurrido al material de clase de unos profesores de música de una universidad americana. Los mismos profesores imparten lecciones de edición de partituras con el programa Sibelius, por lo que sin duda este material está elaborado con dicho programa.

Concretamente, el ejemplo está extraído de los apuntes de teoría (Devore y col., 2010) que bajo el título *How Music Works: A First-Year Theory Textbook* publican los profesores de teoría de la música de la Universidad estatal de Kent (Ohio, E.E.U.U.) bajo licencia *Creative Commons Attribution ShareAlike 3.0*, para los dos primeros semestres de la asignatura.

El material se presenta en formato PDF del que pueden extraerse limpiamente las imágenes insertadas, resultando en unas dimensiones más que aceptables de 2217 × 2272 píxeles (ancho × alto) para la partitura sin los márgenes. En este libro de apuntes figuran muchos casos distintos de partitura, que podrían servir como modelo de igual modo, aunque sólo podemos escoger un ejemplar. Dado que se trata de material de clase, debe pertenecer al grupo 1.

El ejemplo elegido es un *lied* de Franz Schubert titulado *Im Haine*, que corresponde al Opus 56 número 3, D.738.¹⁹

Se trata de un fragmento de tres líneas de texto musical con un pentagrama para la voz, una línea de texto en alemán y dos pentagramas más unidos por una llave formando un sistema de piano. Para la anchura en puntos de imagen que posee el archivo, se calcula una resolución de 300 ppp.²⁰ Los elementos de notación curvos tales como las ligaduras de expresión o las cabezas de las figuras, muestran un proceso de antialias, por lo que la conversión al formato de imagen de puntos es de elevada calidad.

La parte del solista es relativamente sencilla y sólo presenta ligaduras de melisma y algún adorno melódico. La línea de texto está compuesta en un tipo similar o igual a Times New Roman²¹ y tiene líneas de extensión silábica para las palabras que terminan en vocal. Es una sola letra que constituye una simplificación a partir de las tres letras de las estrofas originales. La parte de piano tiene dos voces en cada mano en los tres compases de la introducción; después, una textura de acompañamiento casi totalmente uniforme que alterna las dos manos en forma de una corchea contra dos.

En dos ocasiones hay una expresión adosada a sendas notas, en un contexto ligado, impresa en la parte interior de la ligadura. Se hace notar este hecho porque LilyPond pone de forma predeterminada la expresión, que más concretamente es un acento, por fuera de la ligadura, por lo que es necesario efectuar intervenciones manuales a fin de asemejar la recreación al modelo. Los elementos más característicos de este modelo pueden verse en la tabla 15.

Tabla 15: Elementos gráficos característicos del modelo 3.

Nombre	Ejemplo
Texto	
Adornos	

¹⁹Léase: *Deutsch 738*. Deutsch es el catálogo de las obras de Schubert. La obra aparece, por ejemplo, en (Schubert, 1933) que es una reimpresión de (Schubert, 1894), pero existen muchas ediciones y no es posible saber de cuál procede.

²⁰Puntos por pulgada. 300 ppp equivalen a 118 puntos por centímetro y 600 ppp son 236 ppcm, pero estas cifras no significan nada para un tipógrafo al no estar dentro del vocabulario usual de las tecnologías de la impresión, por lo que a lo largo de todo el estudio nos adscribimos a la unidad ppp, en que es costumbre expresar las densidades de imagen por herencia (probablemente) de los primeros fabricantes americanos de impresoras como Epson y HP.

²¹Es imposible saberlo con exactitud porque la partitura está insertada en el PDF como imagen, no como tipografía incrustada. Por tanto, no aparece la fuente tipográfica listada en los metadatos.

Tipo 4 - Partitura coral a cuatro voces

Este original es una obra del compositor sevillano del siglo de oro de la polifonía clásica Francisco Guerrero, con el título *Ibant Magi*.²² Es una pieza del tipo “SATB”, es decir, a cuatro voces Soprano, Alto, Tenor y Bajo. La obra está en el dominio público. Pertenece al grupo 1 porque estamos hablando de material que los coros *amateur* intercambian entre sí sin ánimo de lucro, generalmente elaborado por el director de la agrupación. Su procedencia es el repertorio del Grupo Vocal “Sólo voces” de Lugo y está editada por Fernando G. Jácome.

El mundo del canto coral probablemente sería muy diferente sin la tradicional actividad de transcripción del repertorio coral de dominio público, por parte de infinidad de aficionados y mediante los más diversos métodos.

El ejemplo que nos ocupa está elaborado con Finale 2001d a juzgar por los metadatos del archivo, que han sido puestos ahí por el propio programa.

En el modelo hay un detalle casi insignificante que nos sorprende. Aunque el PDF es un formato vectorial, es decir, que presenta en teoría una resolución igual a la del dispositivo de salida, las curvas tales como ligaduras de expresión y de unión se encuentran trazadas a muy baja resolución, hasta el extremo de que se aprecian claramente los puntos, tanto en pantalla como en la copia impresa.

El factor de “anchura x” respecto a la altura de los caracteres es un tema muy discutido cuando se trata de encontrar cuál es el tipo de letra idóneo para el texto de las canciones y piezas de polifonía. El problema radica en la relación de tamaños entre la notación musical y el texto, y en la densidad lineal (en el sentido horizontal) de la parte musical. Si ésta es muy apretada y el texto correspondiente es más ancho, hay que forzar la música separándola horizontalmente para dejar espacio al texto. Por otro lado, si se reduce el tamaño del texto, se vuelve más incómodo de leer. El tipo predeterminado de LilyPond se llama “Century Schoolbook” y es más bien cuadrado de proporciones, es decir, la altura de la “x” minúscula es semejante a su anchura. Ha habido iniciativas proponiendo un tipo de caja más estrecha de forma predeterminada, sin que hasta la fecha las razones para el cambio hayan convencido a la mayoría de los desarrolladores del programa.

En la en la tabla 16 aparecen los principales elementos del modelo.

²²El título de la pieza es en realidad *Hostis Herodes impie* (Guerrero, 2003), himno de Epifanía, del que *Ibant Magi* es la segunda estrofa de texto.

Tabla 16: Elementos gráficos característicos del modelo 4.

Nombre	Ejemplo
Sistema coral SATB	
Texto	

Tipo 5 - Partitura de percusión

El caso número 5 es el elemento de control del grupo 1.

Las partituras de percusión son bastante llamativas a la vista, pues difieren del estilo más acostumbrado de notación a causa de que las notas tienen cabezas de distintas formas como aspas o triángulos y existen símbolos que no se encuentran en otros tipos de partitura. A veces se emplean pautas de un número poco común de líneas, por ejemplo una sola línea, en lugar del pentagrama normal.

La partitura que nos ocupa circula entre los alumnos de percusión en forma de fotocopia. Por una indicación en el margen se puede llegar al origen de la misma, que es la parte de batería de la canción *How You Remind Me* del grupo Nickelback, aparecida en la revista francesa *Batterie Magazine* nº 58 (junio 2009) (Adair, 2009), en la que el percusionista Daniel Adair ofrece una transcripción aproximada de la parte de percusión que empleó para la mencionada canción. Es de un nivel básico y según todos los indicios está realizada con un editor sencillo como Encore. Al ser el caso de control del grupo 1, esta partitura no se va a comparar con ninguna otra edición, aparte de consigo misma. La muestra no tiene indicación de copyright pero se está empleando un fragmento de una revista con fines académicos.

Los elementos más característicos del modelo se ven en la tabla 17. La calidad de la muestra es pobre a causa del fotocopiado repetido, y se pueden apreciar los puntos escalonados en la forma de las cabezas de nota, lo que indica que para la impresión original se utilizaron unos ajustes de baja resolución o una impresora antigua (o ambas cosas). El fondo es de color gris y el conjunto presenta un bajo contraste en general.

Tabla 17: Elementos gráficos característicos del modelo 5.

Nombre	Ejemplo
Cabezas de nota especiales	
Trémolos	
Acentos y expresiones	

Al suponer un estilo visual de partitura marcadamente distinto del resto de los casos, y estar situado en el número 5, o sea, justo en la mitad del conjunto de elementos, puede ayudar a mantener la concentración y la apreciación visual del encuestado para el resto de los casos que quedan, del 6 al 10.

11.5.2. Selección de los modelos: grupo 2. Partituras comerciales o profesionales

Tipo 6 - Hoja guía de acordes (*lead sheet*)

Las *lead sheet* son un estilo de notación armónica esquemática, que indican la sucesión de acordes de una canción, con o sin la melodía en notación estándar. Los acordes se suelen expresar por su nombre en el sistema anglosajón, lo que se conoce como “cifrado americano”. También puede haber diagramas de posiciones de trastes para la lectura de acordes para guitarra.

Este modelo es un original comercial elaborado por una editorial musical, distribuido como copia electrónica en formato PDF y adquirido por el autor. Se trata de la canción titulada *The Stand* (Houston, 2005), con letra y música de Joel Houston y que pertenece sin duda al grupo 2 de las ediciones profesionales. Elegimos esta pieza porque tiene llamadas de ensayo, ligaduras de síncopa y un texto que presenta numerosas líneas de prolongación de sílaba. Se emplea un fragmento de una página con fines académicos.

Al describir esta partitura debemos decir que está elaborada con Finale 2006 según los metadatos del archivo PDF. Es una edición profesional de la editorial Hillsnog Publishing. Cada línea de música contiene un solo pentagrama, una línea de nombres de acordes por encima y diagramas de posiciones para guitarra. Presenta una melodía en notación estándar de voz, con barras de pulso para facilitar la cuenta de la duración de los silencios prolongados; el tipo de letra es Arial, siempre según los metadatos del PDF. Hay marcas de ensayo encerradas en un cuadrado. Véase la tabla 18.

Tabla 18: Elementos gráficos característicos del modelo 6.

Nombre	Ejemplo
Diagramas de trastes	
Marcas rítmicas	
Estrofas	Verse 1: You stood Verse 2: I'll walk
Llamadas de ensayo	

Tipo 7 - Partitura para guitarra

La elección de una partitura para guitarra como representante de uno de los grupos obedece a que son partituras relativamente complejas sobre un pentagrama único, con indicaciones características que no se encuentran en otros estilos de música escrita. La complejidad de muchas de las partituras de guitarra se debe a que todas las voces se acomodan dentro de un solo pentagrama; estas voces deben mantener su propia medida y compartir el espacio disponible mientras mantienen como propios otros elementos. Nos hallamos ante un conflicto entre la verticalidad y la horizontalidad de la notación musical.

De un original editado por la Diputación de Badajoz de la obra *Estudios y obras para guitarra*, de Enrique Molina Senra y Juan Pérez Ribes (Molina y Pérez-Ribes, 1995), se ha seleccionado la segunda página de una de las piezas, titulada *Habanera*, de Pérez Ribes. Aunque el libro no lo indica, es una edición de 1995 probablemente realizada con Finale. Pertenece al grupo 1 como edición profesional por ser un libro editado en papel y puesto a la venta.

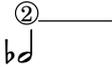
Presenta, entre otros, los siguientes aspectos y elementos de notación: corchetes de cejilla en números romanos; indicaciones de número de cuerda; dos voces en un pentagrama, con acordes de hasta cuatro notas en la voz inferior; digitaciones²³ de mano derecha en cifras arábigas; líneas de continuidad entre cifras de digitación; marcas de respiración²⁴ sobre la barra divisoria. La notación es de un tipo muy grande y la separación de pentagramas no responde a los requisitos de espacio de la música, sino que están distribuidos uniformemente.

²³Las digitaciones son cifras o letras indicadoras de los dedos, como ayuda para el intérprete.

²⁴Las marcas de respiración articulan el discurso musical de forma semejante a como lo hacen las comas del lenguaje natural.

Véase la tabla 19. Con todos estos elementos a reproducir, se presenta un trabajo de recreación bastante arduo, como se detalla más adelante en las notas del proceso de realización.

Tabla 19: Elementos gráficos característicos del modelo 7.

Nombre	Ejemplo
Acorde arpegiado en el contexto de una voz	
Acorde arpegiado en el contexto del pentagrama	
Llamada de coda	
Cejilla	
Número de cuerda	
Respiración sobre barra de compás	

Tipo 8 - Partitura para cuarteto de cuerda

Las partituras para cuarteto de cuerda pueden llegar a ser muy complejas y poseer una asombrosa riqueza visual. Ya a partir de Beethoven, la notación para instrumentos de cuerda es muy profusa en indicaciones expresivas.

En este caso partimos de una edición en facsímil producida en 1970 por la editorial neoyorquina Dover²⁵ (Beethoven, 1970) a partir de las obras completas de Beethoven de Breitkopf & Härtel (Beethoven, 1862). Se trata de un estilo de composición muy elaborado que requiere un tipo de notación visualmente denso, con mucha información. El original es un grabado en plancha de metal de 1862, que se encuadra en la *edad de oro* de la edición manual de partituras.²⁶ El número de plancha que aparece en el pie de página es B.43. La reimpresión es

²⁵Entre los estudiantes se oye decir a veces que Dover no es una editorial porque se limita a imprimir facsímiles de partituras antiguas. Aunque es cierto que no tienen un servicio de edición musical en el sentido de la tipografía de la notación, no puede negarse que editan libros en el otro sentido de la palabra. Por otro lado, las ediciones en facsímil de Dover suelen aportar contenido con valor a los originales que reimprimen.

²⁶Si excepcionalmente se permite a este investigador una apreciación subjetiva después de haber manejado literalmente miles de partituras de todos los estilos y editoriales, tenemos que decir que este ejemplar se enmarca en una de las cimas de la notación musical de todos los tiempos, por el talento y la destreza, tanto técnicas como artísticas, con que se ha realizado.

excelente. Del cuarteto número 7, Opus 59 núm. 1 en Fa mayor, seleccionamos una página del tercer movimiento, *Adagio molto e mesto* (en Fa menor), que ofrece mucha variedad de figuraciones, desde fragmentos homofónicos en negras y corcheas hasta pasajes en fusas. Sorprende la habilidad de los grabadores por aprovechar el espacio disponible sobre la plancha, haciendo caber tanta música sin que deje de ser perfectamente legible y no se lleguen a confundir unas líneas (o sistemas) con otras.

El modelo pertenece lógicamente al grupo 2 de producciones comerciales.

Se trata de una hoja con cinco sistemas de cuatro pentagramas en las claves de Sol, Do en tercera y Fa en cuarta. Cada línea oscila entre los cuatro y los ocho compases. Contiene innumerables elementos de notación musical, que aparte de las propias figuras consisten principalmente en símbolos de articulación (como los picados y las ligaduras de expresión o más propiamente *arcos*), matices de dinámica (entre ellos algunos reguladores) e indicaciones textuales. Algunos de estos elementos se ven en la tabla 20.

Tabla 20: Elementos gráficos característicos del modelo 8.

Nombre	Ejemplo
Trino con resolución	
Cambio de clave	
Fusas; ligaduras amplias	
Matices encadenados	<i>cresc.</i> - - <i>p</i> <> <> <> <i>f</i> <i>sf</i> >

Como se explicará más adelante en las notas que documentan la realización de los modelos, este caso acabará siendo la partitura más densa que hemos tenido la oportunidad de elaborar jamás usando el programa LilyPond, aunque considerando el repertorio de la notación musical en su conjunto, existen, ciertamente, partituras más complejas aún.²⁷

Tipo 9 - Partitura de música de cámara con piano

Para el tipo 9 acudimos a una buena edición, muy renombrada: G. Henle, de Munich. La música es el *Premier Trio en Sol* para violín, violoncello y piano, de Claude Debussy (1862-1918), compuesto en 1880 (Debussy, 1986). Se trata de la primera edición, de 1986 (número

²⁷Entre las partituras de mayor densidad visual se encuentran los estudios instrumentales a partir del Romanticismo y obras orquestales post-románticas.

del catálogo de Henle HN379, 66 págs.), de Ellwood Derr, con texto introductorio, notas de interpretación, arcos y digitaciones, y está impreso en 9 × 12 pulgadas (229 × 305 mm).

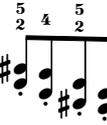
Las partituras de música de cámara con piano se venden en tantos cuadernos como partes instrumentales compongan el conjunto. Las partes de los instrumentistas *de atril* son partituras estándar como las de una parte orquestal, pero si uno de los instrumentos es un piano, su partitura contiene también las partes de los otros instrumentos, a modo de guión. Así pues, el pianista ve lo que está ocurriendo en los otros instrumentos. Los pentagramas de las partes instrumentales están situados por encima del sistema de pautas del piano y tienen un tamaño menor, aproximadamente la mitad que aquél. Sin embargo, las reglas de la escritura musical obligan a que se mantenga en todo momento la sincronía en el sentido vertical, lo que provoca que las pautas de tamaño reducido son proporcionalmente mucho más anchas que las de tamaño normal del piano. De esa forma, cada compás sigue manteniendo la misma anchura absoluta en todas las partes, y la dimensión vertical representa correctamente la simultaneidad de los sonidos.

El prólogo de la edición está firmado en 1986. Aunque no podemos asegurarlo, lo más probable es que por la fecha de la edición y por la calidad tipográfica del original, esté elaborado con el programa SCORE de Leland Smith.

Aunque este trabajo de edición se presenta como “Urtext”, es decir, como versión *auténtica*, lo cierto es que incluye numerosas indicaciones editoriales, si bien están cuidadosamente señaladas mediante paréntesis. Seleccionamos la segunda página del segundo movimiento, *Scherzo*, en Si menor. Contiene lo que podríamos caracterizar como pequeños retos para cualquier programa de edición: además de las mencionadas indicaciones entre paréntesis, tres pentagramas de distinto tamaño, pues la parte de piano incluye los pentagramas reducidos del violín y el violoncello.

Por lo demás, son cuatro sistemas con un cambio de armadura al final, en el salto de página. La notación de la parte de piano es relativamente grande. Está complementada con indicaciones de digitación como ayuda para la interpretación por parte del pianista. No es una partitura de una gran complejidad pero es atractiva a la vista y llena el papel con una gran variedad de elementos. Véase la tabla 21.

Tabla 21: Elementos gráficos característicos del modelo 9.

Nombre	Ejemplo
Sistema para trío de cámara	
Trino prolongado con resolución	
Digitaciones	

Tipo 10 - Partitura orquestal

El caso número diez es el elemento de control del grupo 2 y consiste en una página de orquesta. Una orquesta de tamaño medio gestiona una inmensa cantidad de información. La partitura general de una obra sinfónica suele ocupar más de 100 páginas, mientras que una ópera puede prolongarse a lo largo de 600 o más. A partir de la partitura general, el director extrae entre 15 y 30 particellas instrumentales diferentes y las reparte entre 40 o más atriles dirigidos a unos 70 músicos para que la lean e interpreten durante los ensayos y las actuaciones. Este volumen se incrementa de manera significativa si la obra requiere un coro (Bellini y col., 1999).

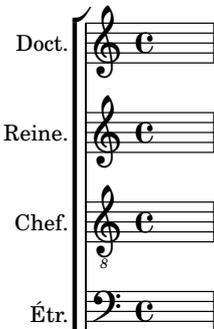
Para la partitura orquestal o de gran formato, con un mínimo de 13 pentagramas,²⁸ se ha seleccionado una página de la ópera “Affaire Étrangère” de Valentin Villenave (Villenave, 2010), elaborada con LilyPond por el propio compositor. No se recrea este modelo porque es el elemento de control del grupo 2, un caso en el que los dos modelos A y B son idénticos.

Esta obra fue estrenada en el año 2009 en el teatro de la Ópera Comédie de Montpellier. Y aunque no es una partitura comercial, pues el compositor la pone a disposición para su libre descarga y su licencia de uso la define como “ópera libre” en unas condiciones semejantes a las cláusulas Creative Commons, que permiten la libre distribución sin ningún cargo por derechos de autor, es un trabajo profesional que forma parte del material de un espectáculo formal. No es una partitura efímera con intención pedagógica.

²⁸Se cuentan en este número de 13, las pautas que no son estrictamente pentagramas por tener un número de líneas distinto de cinco, como las de percusión.

La partitura explora el límite de la cantidad de música que una hoja de tamaño A4 puede contener. Con ejemplos en la tabla 22, las pautas, de arriba a abajo, corresponden a: dos flautas; dos clarinetes (en La y en Si \flat);²⁹ dos saxofones (Alto y Barítono, ambos en Mi \flat); timbales; cuatro pautas de voz; tres secciones de violines; dos de violas; dos de violoncellos; contrabajos y piano (con dos pentagramas). En total son 21 pentagramas.

Tabla 22: Elementos gráficos característicos del modelo 10.

Nombre	Ejemplo
Compases orquestales	
Sistema de coro de ópera	
Sistemas instrumentales múltiples	

Cuadro-resumen de los casos

Una visión general de los diez casos aparece en la tabla 23. Para cada uno de ellos se indica el tipo correspondiente, su título y la forma en que se presenta, método de producción, grupo al que pertenece y finalmente una nota aclaratoria acerca del origen del elemento B.

²⁹ \flat = “bemol”. Estos dos ejemplos de instrumentos que especifican una nota musical como adjetivo, son instrumentos *transpositores*. Los detalles escapan al contexto de este estudio, pero a los efectos de la notación musical es necesario apuntar que la partitura destinada a un instrumento transpositor está *transportada* (o desplazada) en sentido opuesto al desplazamiento que produce el propio instrumento y que está indicado por dicha nota.

Tabla 23: Cuadro-resumen de los diez casos de prueba para el experimento.

Tipo	Título	Origen elemento A	Elaborado con	Grupo	Elemento B
1 Particella	Parte de violoncello de <i>Caresse sur l'océan</i> , de Bruno Coulais	Escaneada	Editor desconocido	1	Recreación del autor
2 Piano	<i>Sonata en Do menor</i> , de J.L. Dussek	PDF	MusiXTeX	1	Recreación del autor
3 Canción	<i>Im Haine</i> , de F. Schubert	PDF	Sibelius	1	Recreación del autor
4 Coral SATB	<i>Ibant Magi</i> , de F. Guerrero	PDF	Finale 2001d	1	Recreación del autor
5 Percusión	<i>How You Remind Me</i> (parte de batería), de D. Adair	Escaneada	Encore	Control 1	Igual al elemento A
6 Hoja acordes	<i>The Stand</i> , de Joel Houston	PDF. Comercial	Finale 2006	2	Recreación del autor
7 Guitarra	<i>Habanera</i> , de J. Pérez Ribes	Escaneada	Finale	2	Recreación del autor
8 Cuarteto	<i>Cuarteto nº 7</i> , de L. van Beethoven	Escaneada	Grabado en metal	2	Recreación del autor
9 Música de cámara	<i>Trío nº 1 en Sol</i> , de C. Debussy	Escaneada	SCORE	2	Recreación del autor
10 Orquestal	<i>Affaire Étrangère</i> , ópera de V. Villenave	PDF	LilyPond	Control 2	Igual al elemento A

11.5.3. Proceso de recreación: técnicas utilizadas

Después de seleccionar los diez modelos de partitura para otros tantos casos del experimento, en las fases posteriores se recrean con LilyPond, que es la herramienta objeto del estudio. Se presentan posteriormente a los jueces en la fase de validación y se ofrecen a los voluntarios dentro de la pregunta 12 del cuestionario.

En los apartados siguientes se hacen constar las actividades que ha sido necesario realizar para finalizar esta tarea.

Se reducen a un mínimo los títulos y las indicaciones textuales más prominentes de las dos imágenes, a excepción de un caso, la pieza coral. El objetivo principal es evaluar la notación musical y suprimir en lo posible la influencia del tipo de letra de estos elementos.

Las dos partituras, incluido el modelo recreado, que se trata como un original, se digitalizan con igual resolución y se imprimen después con impresora láser sobre idénticas hojas de papel a una resolución de 1200 ppp. Con este proceso de “eigualización” o “normalización” de la calidad se trata de reducir el impacto que supone el hecho de que una de las partituras está generada por ordenador e impresa en el equipo del investigador, mientras que la otra, en cada caso, tiene una procedencia distinta.

Se muestra un caso en el que se aprecia, ampliado, el aspecto del modelo y de la recreación antes y después del proceso de normalización, según un enfoque que podríamos denominar “ajuste visual de mínimos”. Véase la figura 68.

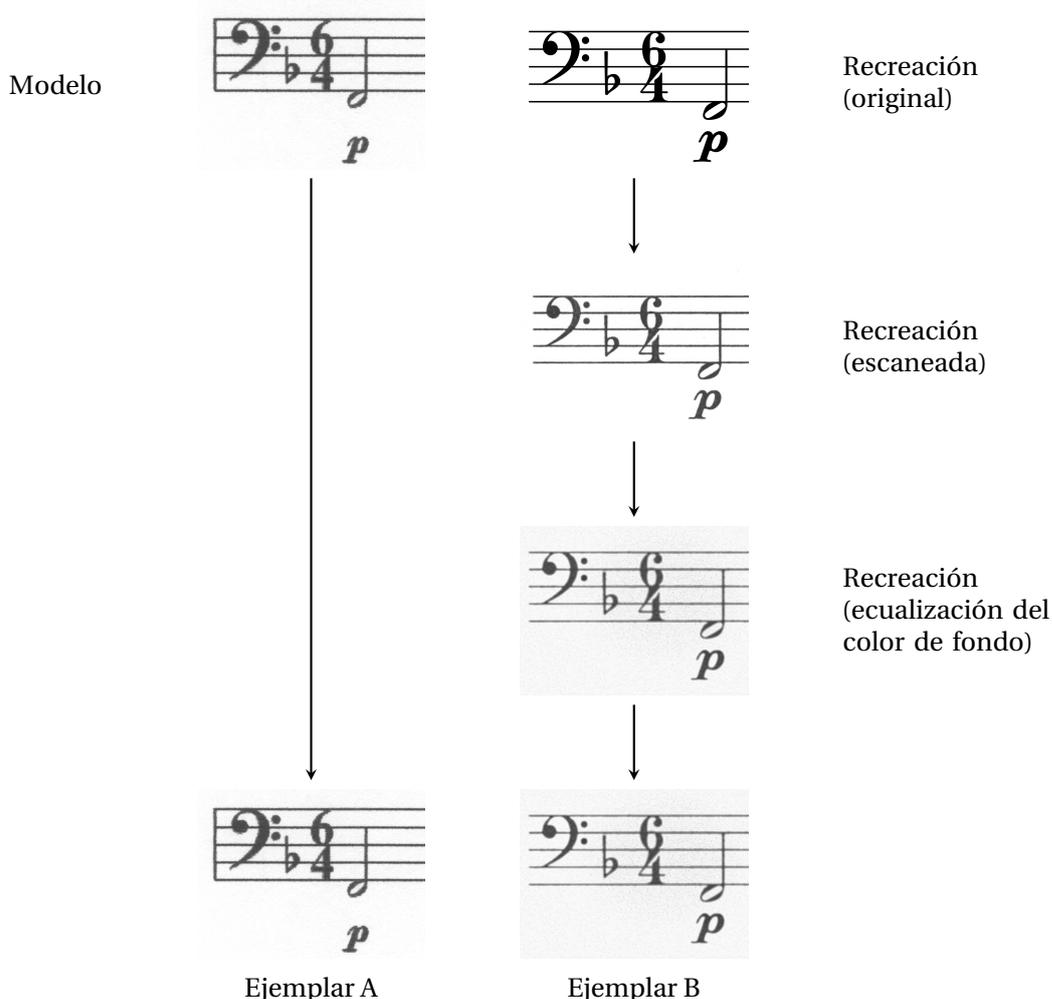


Figura 68: Ajuste de mínimos.

De todas las versiones mostradas, se muestran al participante las dos que aparecen en la parte inferior, es decir, los ejemplares A y B, según el montaje de la figura 66. El problema del enfoque de ajustar los mínimos es la dificultad de igualar el mismo tipo de borrosidad y de ruido de fondo entre las dos imágenes.

El segundo enfoque que llamaríamos “ajuste de máximos” optaría por el proceso complementario: por un lado, mejorar lo más posible el original digitalizado de forma que se asemeje a una partitura editada *ex novo*. Por otro lado, digitalizar la versión recreada y aplicar el mismo proceso. Al final se obtienen dos modelos que, procediendo de distintas fuentes, convergen en el aspecto final.

Véase la figura 69 en que aparecen de nuevo el modelo y su recreación, el proceso que sufren para su igualación visual y los dos resultados finales A y B que componen la pareja de elementos que se presenta durante el experimento.

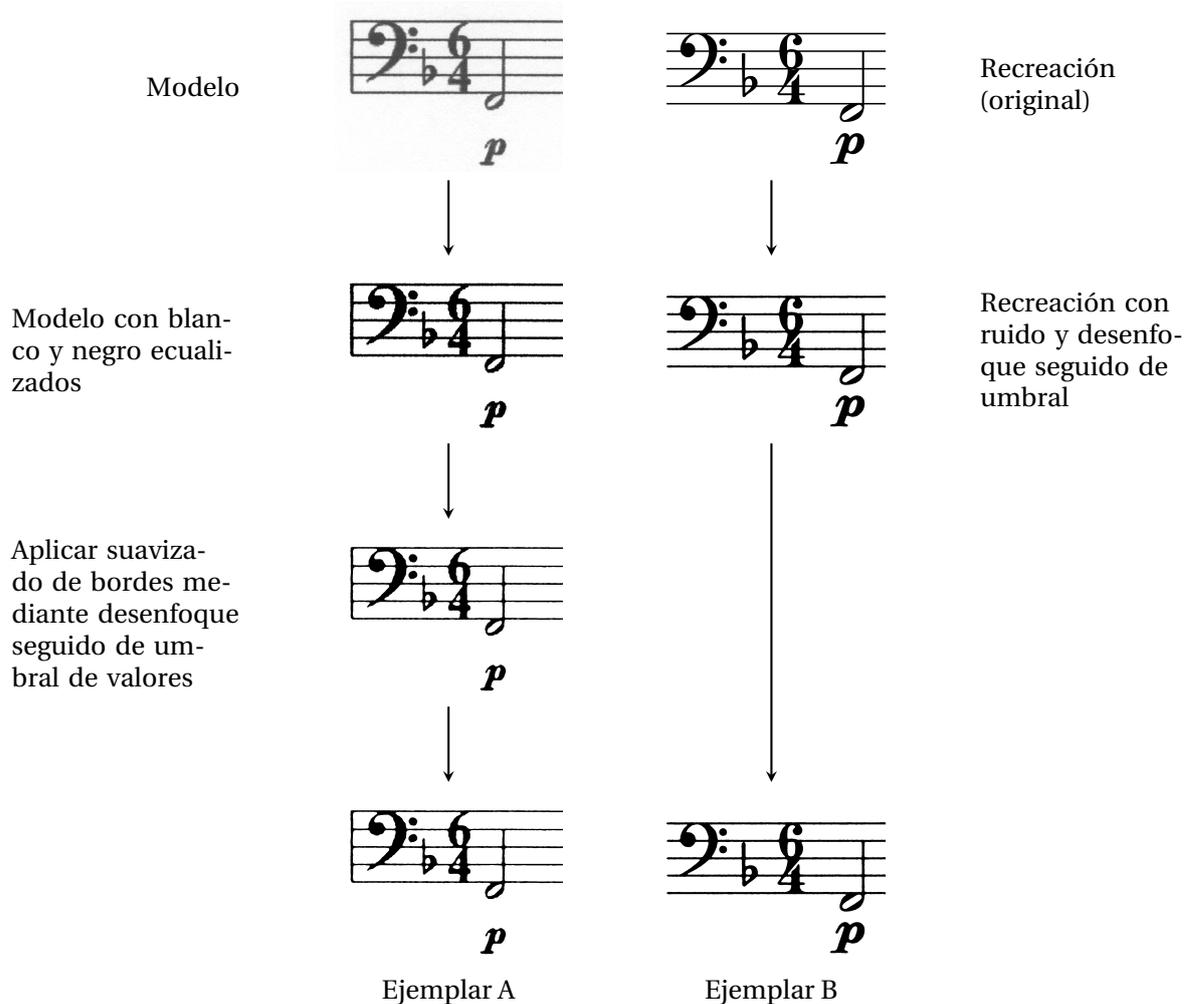


Figura 69: Ajuste de máximos.

Este enfoque alternativo tiene la ventaja de que en algunos casos es más fácil mejorar una partitura si el original es de buena calidad, que tratar de simular la borrosidad y las manchas de un original, añadiéndoselas a una partitura editada que está perfectamente limpia. Por otro lado, en ocasiones se alcanza un límite más allá del cual una partitura de baja calidad no se puede mejorar más. Es posible objetar también que las partituras comúnmente utilizadas tienen, de hecho, manchas y que la fotocopia nunca es perfecta. Por ese motivo, al final del proceso descrito siempre se digitalizan los dos modelos y se vuelven a imprimir, sufriendo así una ligera merma de calidad que es igual para ambos.

La esencia del experimento está en que A y B tienen procedencias distintas pero presentan una calidad de impresión, tamaño, color de tinta y de papel similares (parámetros que deben someterse a un proceso de validación por parte de una serie de jueces expertos,³⁰) y que la versión recreada con LilyPond, que es un sistema de software libre, podrá o no ser selecciona-

³⁰Para la validación, véase el apartado 11.6 (pág. 244).

da como preferible por los voluntarios encuestados, según sus propios criterios de legibilidad, apariencia o los que consideren oportunos.

11.5.4. Recreación y validación: el problema de la calidad

En este punto se hace necesario incidir en la estrecha relación que existe entre los procedimientos de recreación y de validación. El proceso de validación es importante porque establece el grado en que el estudio puede considerarse ciego. Ciertos detalles pueden contaminar la objetividad de la prueba.

Por ejemplo, el escáner que se utiliza para la digitalización de los modelos podría no estar perfectamente limpio o la impresora podría tener un bajo nivel de tóner. Sin embargo, el proceso de preparación del material trata de igualar estos factores entre los dos modelos de cada caso, con el objetivo de que estos efectos sean iguales para cada ejemplar de la pareja, y por tanto no deberá influir a favor ni en contra de uno de los modelos. A pesar de ello, el cristal del escáner debería limpiarse cuidadosamente antes de elaborar los modelos.

Después de escanear los ejemplares originales que están en papel, y durante el proceso de ecualización de la calidad visual de cada pareja de modelos, se trata de impedir que las inevitables manchas del modelo original desvíen la atención del encuestado hacia ellas. Por tanto, hay que eliminarlas. En los procedimientos habituales de retoque fotográfico, las manchas se eliminan copiando encima de ellas un fragmento de imagen del mismo color de fondo, procedente de un lugar distinto de aquélla y sin tocar la notación musical. La eliminación de manchas no debiera interpretarse como un proceso de aumento de la calidad, ya que las manchas no pertenecen a la notación musical, que es el vehículo del estudio. Sin embargo, no puede perderse de vista que las partituras fotocopiadas se utilizan diariamente³¹ y siempre, o casi siempre, tienen manchas.

Cuando el original tiene un fondo que no es completamente blanco, es posible llevarlo a blanco mediante ecualización de los niveles de gris, tratando de que no afecte al negro (o gris) de la notación escrita, o bien copiar el tono de gris en el ejemplar recreado, para que los dos pasen la validación del parámetro “igual color del papel” por parte de los expertos.

Para el mismo objetivo de reproducir el fondo, se ha empleado en algunos modelos una técnica denominada resíntesis digital que toma una muestra del ruido de fondo del material escaneado y lo traslada a la recreación. Se obtiene un fragmento rectangular de una zona “blanca” (no impresa) del modelo y se resintetiza haciendo que rellene toda la parte no impresa de la versión recreada. Después de este proceso se obtienen dos ejemplares que muestran la

³¹La encuesta contiene una pregunta para determinar con qué frecuencia los músicos utilizan material fotocopiado.

misma textura en el fondo de la imagen. La dificultad reside en enmascarar correctamente el contenido musical en la imagen de destino, para excluirlo y evitar que se rellene con el fondo resintetizado. Esta máscara puede realizarse aplicando una función de umbral, que produce unos límites definidos si la imagen está bien contrastada.

Medidas del original

En todos los casos nos marcamos como uno de los objetivos la realización de una partitura semejante en dimensiones al modelo. Para conseguirlo debemos medir cuidadosamente los originales, aunque no con una regla o sobre el papel, sino sobre la imagen escaneada.

Si el original está en un formato electrónico, para obtener las dimensiones de ciertos parámetros de las partituras originales y trasladar estas medidas a los modelos recreados, se ha transformado la página completa de los originales que estaban en formato PDF a un formato de imagen de puntos de tipo PNG con una densidad de 600 puntos por pulgada. Contando los puntos de las dos medidas más significativas, a saber: los márgenes y el ancho del pentagrama, y dividiendo por 600, podemos trasladar el resultado a nuestro programa como una dimensión expresada directamente en pulgadas con buena aproximación, evitando un tedioso proceso de ensayo y error.

Tipo 1 - Particella - violoncello

El original del tipo 1 nos llega en forma de fotocopia. Se escanea la hoja a 600 puntos por pulgada en formato PNG³² y se recorta la imagen hasta los límites del papel. Se anotan las medidas resultantes de la imagen y se establece este tamaño de papel en la plantilla de LilyPond donde se va a recrear la partitura. Después se toma la medida, en puntos de imagen, de los márgenes que hay desde los bordes exteriores de los pentagramas hasta el extremo de la imagen recortada. Estas medidas en puntos se trasladan a los ajustes del papel en el documento de LilyPond.

Para terminar de preparar la plantilla, se han eliminado los contenedores de los títulos y el espacio vertical que ocupan, tanto en la parte superior como en el pie de página.

Se introduce la música en LilyPond con la ayuda del editor Frescobaldi.³³ El contenido musical es sencillo, aunque contiene un gran número de cambios de compás. Las indicaciones

³²El formato PNG es preferible para todo el proceso de escaneo y retoque porque no presenta pérdidas en la compresión ni en la descompresión. El formato JPG, en cambio, es adecuado para fotografías porque produce archivos más pequeños a costa de pérdidas de calidad progresivas.

³³Frescobaldi se ha convertido en el entorno de desarrollo más popular para LilyPond. Véanse los apartados 3.8.2 (pág. 82) y 3.7.4 (pág. 78).

de compás cuaternario en el original son numéricas y así se respeta al editar la recreación. Con ello queda reproducido exactamente el contenido musical, aunque la disposición de los compases en las líneas no es exactamente igual. El fragmento consta de 36 compases; pero mientras el original contiene seis líneas de música con los compases distribuidos de la forma 5-5-8-4-9-5, LilyPond hace de forma predeterminada un reparto según 5-5-6-6-6-8. Se respeta en este caso la versión automática de LilyPond.

No haremos ningún intento de modificar la fuente de la tipografía musical en este caso ni en los que siguen, a pesar de que el mecanismo para hacerlo se encuentra a disposición del usuario: consideramos que el cambio en el tipo de fuente tipográfica para la notación musical no es una tarea incluida dentro de las modificaciones que razonablemente podemos hacer para igualar los dos modelos de cada uno de los casos.

Con la plantilla que preparamos anteriormente a partir de las medidas del modelo original, LilyPond produce una imagen PNG de la partitura recreada, a 600 puntos por pulgada. En este momento aparece un problema, y es que los márgenes de este ejemplar recreado no coinciden con los que se midieron en el original. Aparentemente LilyPond está añadiendo un espacio vertical adicional a las medidas que se han tomado, tanto por arriba como por abajo. Así pues, tenemos que recurrir a la función de anotación del espacio vertical que LilyPond ofrece, y que presenta el aspecto que puede verse en la figura 70.

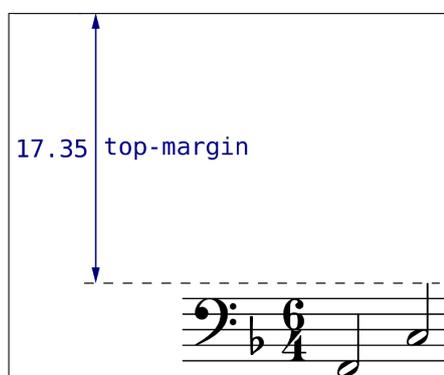


Figura 70: Anotación del espacio vertical de LilyPond. Se observa que hay que tener en cuenta el extremo superior de los elementos de notación (aquí una plica) para la medida del margen. Las unidades de medida indicadas son espacios de pentagrama.

Al analizar esta información se observa que los márgenes eran erróneos porque se había medido la distancia a partir del límite de los pentagramas del original, mientras que LilyPond aplica los márgenes a partir del punto extremo de los elementos de notación más sobresalientes, por encima y por debajo. En la imagen se observa que el margen aumenta a causa del extremo de la plica de la segunda nota. Por tanto, hay que volver a medir los márgenes a partir de los elementos de notación extremos del original y ajustar de este modo la plantilla de LilyPond.

La documentación de todo este proceso es importante para la repetibilidad y para tomar las medidas oportunas cada vez que aparezca el mismo problema, en los casos sucesivos y para otros estudios en la misma línea.

Ahora, para simular el proceso de impresión y fotocopiado del modelo recreado, lo imprimimos en papel a una escala de 100 % y a una resolución de 600 puntos por pulgada, lo que nos da una impresión con la misma escala gráfica que el modelo escaneado original.

A continuación se escanea a la misma resolución esta partitura impresa. Al escanear, se dejan fijos los ajustes de niveles blanco y negro en sus valores estándar como haría una fotocopidora: se obtiene así una imagen con aspecto fotocopiado, si bien aún presenta un elevado contraste en comparación con el modelo. Así pues, se traslada este archivo al programa de tratamiento de imagen, donde se pueden ajustar los niveles de negro y de blanco hasta que coinciden con los valores medidos de la imagen del modelo.

Después de este proceso, disponemos de una imagen que puede pasar por una fotocopia de la misma forma que el modelo, a pesar de que se ha creado de nuevo en un programa de edición e impreso con láser. De hecho, una fotocopidora moderna no es más que un escáner combinado con una impresora láser. La única diferencia es que la fotocopidora asume unos valores estándar que previsiblemente sean válidos para todo tipo de material, ya sean texto, gráficos o fotografías.

En la prueba piloto con un conjunto reducido de voluntarios, se constata que el modelo y su recreación siguen siendo “demasiado distintos” pese al proceso que se ha descrito. Al parecer, el modelo original es demasiado borroso.

Existen dos opciones: la primera sería tratar de justificar que el objetivo de nuestro estudio es probar que para cada partitura de uso habitual en enseñanza musical, hecha con cualquier método, LilyPond puede hacerla igual o mejor, y que se ha establecido la imposición de extraer ejemplos reales, a sabiendas de que algunos podrían ser de muy baja calidad. En consecuencia, esto supondría someter a comparación una fotocopia borrosa frente a una edición de nueva factura, opción que a buen seguro no superaría la validación del modelo.

La otra opción es emborronar artificialmente el modelo recreado para reducir aún más las diferencias entre los dos. Lo que nos enseña este problema es que el aspecto de un ejemplar no se debe solamente al hecho de haber sido fotocopiado, sino que el propio original puede variar mucho en calidad.

La solución adoptada está en un punto intermedio: se asimilan los dos modelos al máximo y se asume que, para el objetivo principal del estudio, algunos ejemplares puede que no se vean demasiado bien, sin que sea posible mejorarlos más allá de un cierto límite. Se añade ruido visual en escala de grises y se aplica un desenfoque gaussiano a la imagen del modelo recrea-

do, comparándolo frente a frente con el original hasta que el nivel aparente de borrosidad es aproximadamente el mismo.

El modelo y su recreación quedan así listos para la validación. El resultado puede verse (reducido al 50 %) en la figura 107 del apéndice, página 324. Todo el tratamiento de imagen se ha realizado con el programa de manipulación de imágenes de GNU.³⁴

Tipo 2 - Pieza para piano

La pieza de piano elegida como representante del tipo 2 está en el formato de archivo PDF en tamaño A4, vertical (210 × 297 mm). No es posible utilizarlo directamente como tal, ya que es necesario eliminar el pie de página. Se convierte al formato PNG mediante la utilidad GhostScript (Deutsch, 2002), con una densidad de puntos por pulgada de 600 dando una imagen de 4961 × 7016 píxeles. En la aplicación de tratamiento de imágenes suprimimos la parte inferior, manteniendo las dimensiones totales. Entonces medimos los márgenes en puntos de imagen como hicimos en el modelo del tipo 1, desde el extremo superior de la imagen hasta la parte más alta del elemento de notación situado más arriba, y respectivamente con el extremo inferior.

Obviando la parte común con lo expresado en el apartado anterior, ya que el proceso es el mismo, se registran las pequeñas diferencias que se producen entre el modelo y la edición recreada. Una de estas diferencias es que la edición del modelo presenta ligaduras uniendo los grupos de corcheas que forman un tresillo, a pesar de que son corcheas unidas por una barra. Los tratadistas (por ejemplo Gould, 2011, p. 194; Gerou y Lusk, 1996) reservan el uso de la ligadura en los grupos especiales para el caso en que las figuras no están unidas por una barra. Así pues se respeta aquí la salida predeterminada de LilyPond, que no presenta esta ligadura, y que aparece cinco veces en el sistema superior.

Otro aspecto de la notación del original que debe mencionarse es que las indicaciones textuales y de matiz dinámico están centradas en el espacio que hay entre los dos pentagramas. Así lo hacemos en LilyPond.

Se hace necesario retocar la dirección de gran número de plicas, que sin duda el editor ha modificado manualmente en su propio proceso de edición. Esta certeza nos hace modificarlo también en la recreación. Los casos en que hay diferencia entre las dos ediciones, son casi siempre plicas hacia abajo que están precedidas y seguidas de plicas hacia arriba. Estéticamente, la intervención que esto demanda es dejar las tres en la misma dirección.

Por último, la distribución en líneas de la música contenida en la página es de cinco, con

³⁴El programa de manipulación de imágenes de GNU es más conocido como el GIMP y la referencia online principal es <http://www.gimp.org>.

5-6-5-6-5 compases, mientras que el proceso automático de distribución de LilyPond da como resultado seis líneas con 5-5-5-4-4-4, sin que las seis líneas den un aspecto apreciablemente más abigarrado que las cinco, al contrario, cada línea goza de mayor holgura en la distribución horizontal. Se respeta en este caso este resultado del programa.

Con este resultado, la edición queda lista para su validación. El proceso seguido es notablemente más sencillo que en el caso del tipo 1 porque la partitura venía en formato PDF. El resultado final es el que se muestra en la figura 108 de la página 325 (apéndice C).

Tipo 3 - Canción a una voz con piano

En el caso de la canción con acompañamiento no ha habido dificultades especiales para su recreación. En el aspecto musical, se ha tenido que reproducir la repetición existente, pero dado que el modelo consta solo de la primera página, el bloque de repetición no se cierra. Por tanto, tenemos que producir un documento en el que la repetición no aparezca explícitamente en la primera página, lo que da como resultado un documento de dos páginas, de las cuales utilizaremos solamente la primera. No podemos sencillamente cerrar la repetición al final de la primera página y falsear el aspecto de la barra de cierre de repetición: LilyPond no lo permite o desconocemos la forma de hacerlo.

La plantilla se ha preparado con los márgenes adecuados como en los casos anteriores, con una salvedad. La imagen extraída del documento tiene una resolución de 300 puntos por pulgada, en lugar de 600 que es la resolución que nos se ha fijado como estándar para todos los modelos y sus réplicas. Habrá que tenerlo en cuenta tanto para el cálculo de los márgenes como para el tamaño de la hoja y del material de notación. Existe una dificultad con el cálculo de los márgenes, concretamente el margen superior, a causa de la presencia de una indicación textual. Si el margen superior se mide desde el elemento más alto en la página, y en los dos casos el elemento más alto es esta indicación textual, el espacio entre el primer pentagrama y el borde superior del papel solo coincidirán si este elemento está a la misma distancia del pentagrama en los dos casos. Pero no es así: LilyPond deja el elemento textual apreciablemente más cerca del pentagrama, lo que produce un primer pentagrama más elevado, cuando se fija el mismo margen que en el original. No nos queda más remedio que aplicar un método de aproximaciones sucesivas para conseguir que el elemento dominante en la apreciación del margen superior, esto es, la situación del primer pentagrama, esté en el mismo lugar en ambos casos. El resto de los márgenes se ha calculado midiendo los puntos de imagen del original y dividiendo por 300 (la resolución de la imagen) para dar una medida en pulgadas.

El empleo de pulgadas como unidad de medida obedece a que tradicionalmente la técnica de la impresión, y después también la tecnología informática de las artes gráficas, están basadas en la pulgada como unidad base.

Aunque el modelo original se encuentra insertado dentro de un documento PDF, la partitura en sí es una imagen insertada de puntos, es decir, no vectorial. Por tanto, no hay metadatos de la propia partitura que nos faciliten los nombres de las fuentes para los tipos de letra, tanto de la tipografía musical como del texto. Tampoco tenemos constancia, a partir de los metadatos, del editor utilizado para elaborar esta partitura, pero podemos afirmar con casi total seguridad que se trata de Sibelius, por los motivos dados en el apartado correspondiente de la fase 1. La imagen es de alta calidad, no está escaneada e insertada como fotografía, sino que es directamente la salida del programa de edición de partituras, sin ninguna pérdida visual.

El texto presenta un tipo similar a Times, que puede especificarse en LilyPond para poner letra a la canción.

Ya dijimos que se trata de una ilustración dentro de un libro de apuntes para un curso universitario de teoría musical. Como tal, es el trabajo de un profesor de música y no de un copista profesional, pero la edición es bastante correcta. Una vez que todo el contenido musical está reproducido de forma exhaustiva, a la hora de recrear el modelo surgen pequeñas diferencias entre éste y la versión recreada. No todos los elementos cuyas disposiciones son algo diferentes exigen una imitación plena: creemos que sólo aquellos que llamarían mucho la atención si no fuesen imitados adecuadamente, deben corregirse. A este conjunto de diferencias pertenecería la posición de la acentuación por dentro de la ligadura. Sin embargo, una vez hecha esta intervención, al forzar la posición de la ligadura por fuera (es decir, más alejada del pentagrama), se distancia de las notas musicales y en comparación resulta algo plana para esta distancia. A pesar de ello mantenemos esta diferencia. Véase la figura 71, que es un detalle de la figura 109 del apéndice (página 326).

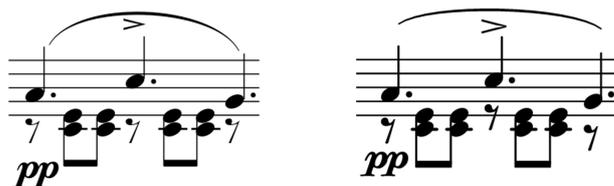


Figura 71: Detalle del modelo 3, en que se aprecia la diferencia en la forma de las ligaduras entre el original (a la izquierda) y las de LilyPond (a la derecha) después de forzar la indicación expresiva por dentro de aquélla.

Se persigue en todo momento, en este ejemplo y en el resto de los que componen la serie de casos, un equilibrio entre la preservación de las *personalidades* de ambas partituras, y la eliminación sistemática de todas las diferencias que por ser muy aparentes pudieran falsear el resultado de la prueba.

También es muy evidente la diferencia entre el tipo de letra predeterminado de LilyPond para el texto, y el del original, mucho más pequeño y estrecho. Como el tipo del original es muy común, se puede incorporar fácilmente al documento; esta intervención manual cabe dentro

de lo que se puede caracterizar como una asimilación sencilla y razonable. El resultado es muy similar como puede verse en la figura indicada.

Por otro lado, existen diferencias más sutiles, que no se van a corregir por uno de dos motivos. O bien porque surgen de la aplicación de una regla de la escritura musical que el copista original ha obviado consciente o inconscientemente, o bien porque pertenecen al proceso de tipografiado automático de LilyPond, que es precisamente lo que se pretende evaluar y no tiene ningún sentido contravenir las reglas de composición que el programa aplica a la partitura. En caso contrario, además de los márgenes y el contenido musical, hasta el último detalle del modelo habría sido *clonado* con gran esfuerzo, y no tendríamos nada que oponer a los modelos seleccionados. A causa de la función de disposición automática del contenido musical que LilyPond realiza, la distribución puede llegar a ser muy diferente entre el modelo y la recreación.

Por ello, uno de los objetivos de la validación de la prueba es asegurar que, al parecer de los jueces, ninguno de los ejemplares de cada pareja es obviamente una edición *ex novo*, a no ser que el modelo lo sea también, como es el caso.

Entre las pequeñas diferencias que mantenemos en la versión recreada está la curvatura de las ligaduras, la posición de los guiones de separación de las sílabas del texto, la distancia entre las barras de compás y las alteraciones accidentales.

Se prepara el original quitando la pequeña nota de la parte superior con el título y el autor, que no está en el formato acostumbrado de una partitura. Los modelos en su estado definitivo están en la figura 109 (pág. 326).

Tipo 4 - Partitura coral, a cuatro voces SATB sobre sendos pentagramas

En la partitura coral se han efectuado las siguientes intervenciones manuales: aparte de los márgenes, se han hecho algunas modificaciones a la partitura tal y como sale de la tipografía de LilyPond para que se asemeje más al modelo original. Se introduce un relleno de separación entre los nombres de las voces y el corchete del sistema. Se modifica el tamaño y la separación del texto musical porque en el original aparece mucho más pequeño y alejado de su pentagrama correspondiente, de lo que LilyPond hace de forma predeterminada.

Además se ha intervenido en el título y subtítulo, eliminando el estilo predeterminado en negrita y aumentando significativamente el tamaño del título principal.

La partitura original no termina dentro de los límites de la página seleccionada. En ocasiones veremos que una partitura que se continúa en otra página requiere la aplicación de algún artificio para que todo aparezca como si la última línea continuase de forma natural en la página

siguiente. Los elementos que dan algún tipo de problema en este aspecto son las ligaduras de expresión, los guiones de separación de las sílabas del texto, los cambios de tonalidad en el último compás y todos los elementos de extensión, como por ejemplo los reguladores o las indicaciones textuales extensas. En nuestro caso solamente tenemos unos guiones de división de palabras, pero es necesario hacerlos aparecer mediante la escritura de música adicional y texto correspondiente a esta música, que haremos recaer en la página siguiente mediante un salto explícito. Posteriormente se descarta esta página auxiliar que solamente contiene los elementos justamente necesarios para producir el efecto de continuación.

En este modelo hay un logotipo que se ha suprimido para el experimento, con el fin de evitar los factores de distracción como se ha hecho en el resto de los casos cuando ha sido necesario. Asimismo, debido a que se trata de la primera página, contiene un encabezamiento con el título y el autor en un tipo de letra "Bangle Normal". Se ha mantenido en el objeto recreado, si bien con el tipo de letra predeterminado de LilyPond, salvo por el detalle de que es un tipo normal, mientras que los títulos de LilyPond aparecen en negrita de forma predeterminada. El texto cantado está en un tipo "FootlightMT" que es semejante al anterior, pero más estrecho.

Después de estas modificaciones, el modelo recreado presenta un aspecto muy similar al original y se da por bueno a la espera de la apreciación de los jueces validadores. El resultado final puede consultarse en la ilustración 110 del apéndice, en la página 327.

Tipo 5 - Percusión

La partitura de percusión no se recrea porque es el elemento de control del grupo 1, pero hay algo que decir sobre la presentación del caso y por qué hubo que retroceder a una fase anterior en la elaboración de este modelo.

Del examen de la imagen impresa se calcula que la resolución aproximada del original es algo menos de 72 puntos por pulgada, que es la resolución estándar de un monitor de ordenador. Esto nos lleva a afirmar que el original impreso de esta partitura era una serie de capturas de pantalla volcadas hacia la impresora. Además, se ha fotocopiado repetidas veces a partir de copias previas. Si de algo ha de servir la utilización de un modelo de tan pobre calidad en el presente trabajo, que sea la constatación de que los alumnos manejan con frecuencia un material (bien sea aportado por sus profesores o no) que no siempre está a la altura de aquello que los agudos ojos de un músico considerarían como "partituras de buena calidad".

En primer lugar se limpia suprimiendo el color gris del fondo. Se aumenta el contraste y la nitidez. Se recortan los bordes del fotocopiado y el logotipo de la revista. Al final de este proceso de mejora, se imprime dos veces y se utilizan las dos copias como elementos del caso 5.

El modelo preparado para su examen por parte de los jueces validadores se puede ver en la figura 111 del apéndice, página 328. Debemos recordar que los elementos A y B son iguales en contenido, como corresponde al elemento de control, aunque ello no implica que sean dos copias idénticas de la misma partitura. Al fotocopiarla dos veces, normalmente resulta un aumento del ruido visual y además las manchas de la fotocopidora no son exactamente las mismas.

Tipo 6 - Canción con acordes y diagramas de posiciones para guitarra

Para la recreación de la primera pieza profesional de la serie fueron necesarias las siguientes intervenciones manuales:

- Igualación de los márgenes.
- Reducción del tamaño de la letra y configuración de un tipo Arial para el texto.
- Ligero aumento del relleno de separación entre la línea de texto y el pentagrama que le corresponde.
- Reducción significativa del tamaño de los diagramas de posiciones de acorde.
- Reducción del tamaño de los nombres de acorde que aparecen sobre los diagramas de posiciones.
- Redefinición de las letras de ensayo para que presenten un cuadrado circunscrito al símbolo de la llamada.
- Creación de una función³⁵ que haga aparecer barras inclinadas rítmicas para los cuatro primeros compases, indicando el número de pulsos de espera durante el silencio. LilyPond no presenta esta funcionalidad de forma predeterminada, pero entra dentro de lo razonable extenderlo mediante la utilización de los mecanismos estándar previstos para ello.
- Clonar los saltos de línea desde el original. De no haberlo hecho así, en una sola página habría cabido mucha más música que en el modelo, ya que éste presenta una disposición poco compacta, con bastante espacio horizontal sobrante.

Las diferencias que aún persisten y que resisten una intervención más profunda, son:

- La posición de las marcas de ensayo. LilyPond las sitúa incondicionalmente en una posición muy alta para evitar la colisión con otros elementos de notación, pero hubiera sido necesario desplazarlas a la izquierda para que ocuparan una posición más cercana a la pauta. No se hace esta intervención.

³⁵Función creada en el lenguaje de extensión Scheme por el autor. Para la realización de este punto hemos contado con las sugerencias de los usuarios de la lista oficial de distribución de correo del programa LilyPond, a los que agradecemos su aportación: Janek Warchoł, Robert Schmaus y Richard Shann. Scheme (Abelson y col., 1998) es un lenguaje basado en Lisp que se utiliza para ampliar las funciones del programa de una forma que esté accesible para el usuario final.

- Digitación de los diagramas de trastes. Los diagramas de acorde en el original presentan cejillas en el quinto traste. Mantenemos los diagramas de LilyPond ya que el acorde es el mismo, aunque la digitación no corresponda exactamente.
- El aspecto mismo de los diagramas de trastes. Los diagramas del original son muy claros, presentando unas líneas relativamente gruesas. La intervención correspondiente a este tipo de modificación en el documento recreado es poco usual y se opta por mantener el aspecto predeterminado.
- La separación de la indicación metronómica. Puede decirse lo mismo que en los casos anteriores. Esta intervención hubiera requerido tocar un código que no es habitual modificar manualmente.
- Tamaño de la indicación de estrofa, “Verse 1/2/3”. LilyPond fija, de manera predeterminada, el mismo tamaño para esta indicación que para el texto que le sigue. Se mantiene de esta forma aunque suponga una cierta diferencia visual con el original, que utiliza un cuerpo menor para la indicación de estrofa.

La recreación presenta muchas pequeñas diferencias, pero el aspecto general es muy similar al original. El resultado final se puede ver en la figura 112 del apéndice, página 329 y queda preparado para la validación por parte de los jueces.

Tipo 7 - Guitarra

Al presentar por primera vez la pieza de guitarra y justificar su elección, se anticipó que sería una partitura difícil de recrear.

Se trata de una música polifónica en un solo pentagrama. Hay que poner dos voces en el espacio que normalmente está reservado para una sola voz, y hay que mantener continuamente una distinción entre ellas, lo que se hace actuando sobre la orientación de las plicas. Las voces se introducen en LilyPond en bloques de texto independientes.

La inclinación de las barras de corchea es pronunciada en el original; LilyPond las traza con un ángulo más horizontal. No se hará ninguna intervención manual sobre este aspecto visual.

Se ha requerido un pequeño artificio para hacer que la línea de arpegio que aparece en el tercer compás de la página seleccionada abarque las dos voces de polifonía presentes en la partitura. La técnica estándar para conseguirlo es llevar el *plug-in* o complemento grabador del contexto de las voces al del pentagrama.³⁶

³⁶Los detalles de la arquitectura modular escapan al contexto de este capítulo. A grandes rasgos puede decirse que la estructura interna del programa consiste en una serie de *contextos* y *complementos*. Los contextos son niveles de la notación musical como un sistema de pautas, una pauta individual o una voz, y los complementos son funciones encapsuladas que se ocupan de trazar los elementos gráficos de estos contextos. Un contexto contiene normalmente una serie de complementos, pero éstos se pueden intercambiar entre aquéllos para conseguir efectos especiales como el que se ha visto en este modelo.

Se ha tenido que desplazar una indicación de matiz dinámico hacia la izquierda sobre una nota relativamente grave; de no hacerlo así, LilyPond la alinea justamente en la vertical, lo que da como resultado una extensión total vertical excesiva. La disposición usual para las indicaciones de matiz es en la vertical (Gould, 2011, p. 103) o un poco a la izquierda (Gerou y Lusk, 2004, p. 86), pero está acertado que se encuentre totalmente a la izquierda de la nota, pues es una nota grave y en voz secundaria, con la plica hacia abajo. Al desplazar el matiz hacia la izquierda, los algoritmos de ajuste por *línea de horizonte* se encargan de hacer encajar la indicación en una posición mucho más alta, donde dispone de espacio suficiente.

Fue necesario programar una marca nueva de respiración en forma de *V*, que en el original aparece en dos ocasiones justamente sobre la línea divisoria.

Las digitaciones de mano izquierda (números de traste en cifras arábigas) están con frecuencia dentro del pentagrama; LilyPond las pone de forma predeterminada por fuera de las líneas, por lo que hay que intervenir forzando que el pentagrama *deje* de ser un obstáculo rígido (para las fórmulas o algoritmos de evitación de las colisiones) y que las cifras puedan colocarse entre las líneas.

Quizá el mayor problema con esta pieza sea el de tomar ciertas decisiones durante la recreación que corregirían fallos de notación en el original. Hay una indicación numérica sobre los compases 6 al 8 de la página seleccionada, en un tipo mayor que el resto y con una línea extensora, que semánticamente corresponde a una indicación del número de cuerda. La forma de indicar los números de cuerda en la guitarra es, según Gould (2011), mediante cifras arábigas dentro de una circunferencia. Por tanto, así como en otros casos se ha seguido la regla de las “intervenciones razonables” sobre el resultado estándar para hacer asimilar una recreación a su modelo para decidir si se hace o no, en este caso la “intervención razonable” se hace para enmendar ligeramente al modelo, y ello según el criterio de una fuente autorizada. Así pues, se crea la indicación como lo determina esta fuente.

Las otras indicaciones numéricas son digitaciones para la mano izquierda y tienen el significado de número de traste. Las podemos reproducir fácilmente. Respecto a las digitaciones de mano derecha, que se indican mediante las letras *p*, *i*, *m*, *a*, (pulgar, índice, medio, anular) la partitura original no tiene ninguna, lo cual es extraño dada la profusión de indicaciones técnicas existentes en esta partitura y que son, por otro lado, habituales en la música impresa para guitarra.

Después de estos ajustes, las diferencias que permanecen son pocas. Por un lado está la indicación de acorde arpegiado, que en el original abarca no solo las cabezas de las notas, sino también la plica y el corchete (en su caso) de la figura, dando como resultado una línea bastante larga; es incorrecto según Gould (2011, p. 131), y ejemplifica la forma en que un gesto inapropiado del ratón, en el manejo de un programa de edición, puede producir una indicación defectuosa. Las dos instancias de este tipo de elemento están en los compases primero

y tercero de la página seleccionada. Mantenemos el resultado alternativo que LilyPond ofrece, y que limita el alcance de la línea a las alturas verticales de las cabezas de las notas que componen el acorde, sin incluir la plica.

Otra diferencia es que en el original se emplean ligaduras curvas para los tresillos. Gould (2011, p. 194) dice que el número del tresillo se debe acompañar de un corchete recto, y no con el arco curvo de las ediciones antiguas, pues podría confundirse con una ligadura. De todas formas, cuando las figuras del grupo están unidas por una barra, la ligadura (o corchete) son innecesarios y LilyPond no los pone. Véase la figura 72 donde se aprecia también que las dos fuentes tipográficas para los números resultan invertidas en su utilización. El original emplea un tipo Times para los dedos y un tipo grueso para las indicaciones de tresillo. La recreación, en cambio, usa el tipo grueso para los dedos.

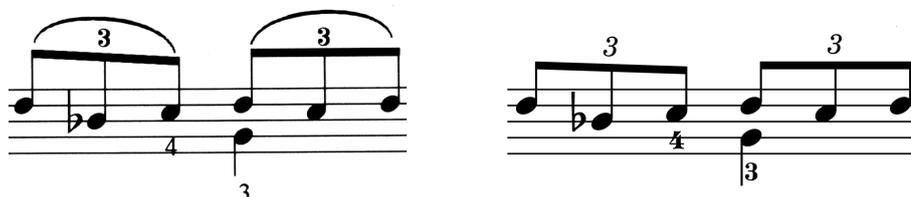


Figura 72: Detalle del modelo 7: ligadura de tresillo en el original (a la izquierda) y ausencia de él en LilyPond (a la derecha).

Cuando la voz superior presenta cifras de digitación, la posición predeterminada es sobre las notas y así se ha establecido a pesar de que el original las sitúa equivocadamente por debajo algunas veces. Esta decisión se encuadra dentro de las diferencias razonables.

Aún existen otras indicaciones, esta vez en números romanos, que llevan línea extensora discontinua, y que tienen el significado de cejillas (son, por tanto, indicaciones para la mano izquierda). Podemos recrearla sin excesivo problema como expresiones textuales comunes, excepto por un detalle en el último compás, y es que allí coinciden *dos* indicaciones textuales distintas con línea extensora, en el mismo instante temporal. Una es una cejilla, como se ha dicho; la otra es un *ritardando*. Para hacer posible esta coincidencia, el único mecanismo de que dispone LilyPond es la creación de una voz fantasma, oculta, a la que puede adjuntarse la segunda expresión textual.

Con el proceso de elaboración de este caso se confirma la sospecha inicial acerca de que las partituras de guitarra son especialmente complejas de tipografiar, a pesar de su aspecto engañosamente sencillo. El resultado final está en la figura 113, página 330.

Queremos hacer una puntualización: de la lectura de los párrafos anteriores podría sacarse como precipitada conclusión que el proceso de recreación es una sucesión de decisiones arbitrarias que tratan de manipular al encuestado para que, consciente o inconscientemente, se decante por elegir la versión recreada. Sin embargo, cuando se observa el caso número 7

con los modelos A y B, original y recreado, lado a lado, no podemos evitar pensar que aquí LilyPond ha sido llevado al límite de sus posibilidades teóricas para conseguir un resultado que es, de una forma muy llamativa, extremadamente similar al modelo *excepto*, quizá, por la diferente inclinación de las barras de corchea.

Tipo 8 - Cuarteto de cuerda

Como ya se ha anunciado, el modelo de partitura del tipo 8 es una realización de muy alto nivel técnico y de gran belleza, labrada con un buril sobre una plancha de metal.

El hecho de que sea un *Adagio molto* ofrece a Beethoven la oportunidad de emplear figuraciones rítmicas de muy diversa rapidez, desde blancas hasta fusas, y al mismo tiempo adicionar ornamentos como trinos con resolución y grupetos circulares de tres y cuatro notas, tipografiados de forma explícita, es decir, prescindiendo de abreviaturas.

Ninguno de estos elementos presenta especial dificultad para LilyPond, utilizando solamente técnicas estándar que no han requerido ningún artificio fuera de lo normal. Tras una primera fase de ajuste de los márgenes como hemos hecho en los casos anteriores, y después de establecer un tamaño adecuado para los pentagramas, el abundante contenido musical de la página se amolda bien al espacio disponible.

Entre las diferencias que persisten mencionaremos que los cambios de clave se hacen en el original mediante un símbolo de igual tamaño que la clave principal, mientras que la práctica común es emplear un tamaño más pequeño.³⁷

En general, LilyPond hace las curvas de las ligaduras de expresión mucho más planas, con una curvatura menos acentuada que en las partituras clásicas. La curvatura ideal se establece mediante un compromiso según el cual la curva debe abarcar todo el grupo de notas pero sus extremos deben estar lo más cerca posible de las notas inicial y final del grupo. A juzgar por los resultados, el compromiso que LilyPond hace es a favor de una mayor distancia de la curva a las notas extremas, lo que deja la curvatura menos pronunciada cuando el hecho de acercarse más supondría forzar la trayectoria, cosa que podría hacerse a mano. No lo hemos hecho en virtud de la doctrina de las intervenciones “razonables”.

El resto de las diferencias son pequeñas. Como es de esperar en un trabajo tan profesional, o quizá esto es precisamente algo que contribuye a tal calificación, el espacio vertical está admirablemente aprovechado sin dar aspecto abigarrado. No hay prácticamente ningún espacio de relleno innecesario entre los pentagramas dentro de un sistema dado; como los márgenes son los mismos y el tamaño de los pentagramas es también igual, por una simple aritméti-

³⁷Según Gould (2011, p. 11), el tamaño de una clave que aparece después del comienzo ha de ser dos tercios del tamaño de la clave inicial.

ca de las medidas verticales el experto grabador puede aún dejar algo de separación entre un sistema y otro, mientras que LilyPond no comprime tanto la dimensión vertical y tiende a colocar los sistemas más cerca para satisfacer la restricción que se le ha impuesto de que conserve la misma extensión total.

Cuando esta cuestión se ha debatido entre los desarrolladores del programa, el problema se ha podido reducir a que para apurar aún más el espacio vertical habría que simular la inteligencia y el juicio visual del grabador humano mediante algún árbol de decisiones que fuese moviendo los elementos uno a uno, buscando una forma de encajarlos que favoreciese la reducción de la altura total. Ya se hace, hasta cierto punto, un proceso parecido para buscar la distancia mínima entre los sistemas trazando sendas “líneas de horizonte” superior e inferior para cada uno de ellos, pero es un proceso ciego y lineal que no implica el ensayo y error de ninguna manera.

El grabador humano, además, se permite la licencia de introducir dentro del pentagrama, aunque sea parcialmente, elementos gráficos o textuales como reguladores dinámicos o expresiones de agógica. LilyPond no lo hace si no se le obliga: por ejemplo, jamás haría que la parte superior de un *sforzando* solapase las primeras líneas del pentagrama como hace el modelo en el cuarto compás de la tercera línea, en el violín primero (pauta superior), o ponerlo completamente dentro, como ocurre en el compás siguiente. Por su poder ilustrativo, incluimos este pasaje en la figura 73 donde se aprecian las atrevidas decisiones que, en el estado actual de la técnica, quizá solamente un humano puede tomar.



Figura 73: Detalle del caso número 8, cuarteto de cuerda (elemento A), donde se ve cómo en ocasiones un grabador humano decide introducir elementos dentro del pentagrama para aprovechar al máximo el espacio vertical.

El proceso de ecualización se completa con una fase de fotocopiado y escaneado en los dos ejemplares. El resultado de la pareja de elementos del caso número 8 puede verse en la figura 114 de la página 331 de los apéndices.

Tipo 9 - Música de cámara

La página seleccionada de esta partitura de música de cámara, que como ya se dijo al presentarla es el guión para el pianista, comienza en el compás número 26. Podemos forzar la numeración del primer compás en LilyPond para que todos los números coincidan.

En la mano derecha del pianista hay pasajes con dos voces. Existen dos enfoques para la organización del código cuando hay que poner más de una voz en el mismo pentagrama. El primero consiste en dedicar bloques de código completamente independientes para cada voz y unirlos en una única estructura dentro del contexto de un pentagrama, en el bloque de la partitura principal. Este método es adecuado cuando toda la pieza, o una parte extensa de ella, está compuesta por varias voces. Hemos usado esta técnica en otro caso anterior, el número 7, correspondiente al tipo de partitura para guitarra.

La segunda forma que describiremos es más conveniente, por su claridad, cuando hay esporádicos fragmentos con más de una voz, como en el ejemplo que nos ocupa. Se trata de abrir contextos de voz simultáneos de forma local en el lugar requerido. La ventaja en este caso es que las partes que ocupan la voz secundaria están perfectamente localizados dentro del texto general, y así es como lo hemos hecho, mientras que de otra forma habría que sincronizar la voz secundaria con la principal mediante silencios ocultos de larga duración u otro artificio.

En el pentagrama inferior del piano hay acordes arpegiados. Si se compara con el mencionado caso 7, de guitarra, se observa que allí las líneas onduladas de acorde arpegiado se prolongan más allá de la cabeza de las notas que forman el acorde, abarcando hasta el final de la plica de las figuras. En este original no ocurre así y LilyPond lo hace de forma predeterminada de un modo similar.

En el capítulo de las diferencias que permanecen después de la recreación, puede decirse lo mismo que en el caso 8 (cuarteto de cuerda) respecto a la curvatura de las ligaduras. En resumen, LilyPond hace unas ligaduras muy planas en comparación, guardando más distancia entre las puntas de la curva y las notas extremas del grupo que abarca la ligadura. En la mayor parte de los casos se ha dejado sin modificar en la versión recreada para no incurrir en un exceso de trucaje, pero en los casos más evidentes hemos ajustado la curvatura de estas ligaduras. El proceso ha sido denominado “embellecimiento” por un usuario de LilyPond y miembro del equipo de desarrollo, Janek Warchoń, que lo utilizó con profusión en una partitura comercial,³⁸ aunque lo calificó de un trabajo sumamente tedioso.

Para la preparación final de los elementos de este caso, tenemos dos opciones ya descritas anteriormente: podemos pasar la copia recreada por el proceso de impresión, escaneado y

³⁸Se trata de la integral de los *lieder* de Oskar Fried (Fried, 2013), una partitura que recibió el premio a la “mejor edición de 2014” de la asociación de editores de música alemanes. La tipografía corrió a cargo de Urs Liska y Janek Warchoń. Véase la sección 5.2 (pág. 98) en que se desarrolla este punto.

nueva impresión para emular el efecto de una fotocopidora, o podemos mejorar el original blanqueando el fondo y limpiando las posibles manchas. El original es de muy buena calidad, es una impresión moderna y el libro es nuevo, por lo que se podría simplemente llevar el nivel de fondo del papel a un blanco absoluto, e incorporarlo al caso del experimento junto a la versión recreada, impresa directamente. Sin embargo, es difícil simular las manchas del escaneado. Puede parecer más realista imprimir el documento y escanearlo para que el escáner aporte las mismas manchas a los dos elementos, con lo que el efecto de fotocopia simulada estará *en los dos lados de la ecuación*. Sin embargo, en teoría esto debe dejar al original en ventaja, ya que una partitura comercial bien impresa en papel siempre será superior a una copia de impresora. Ahora bien, al final hay que imprimir los dos elementos, y ambas impresiones se harán con la misma impresora, por lo que se detiene aquí el proceso (a la espera de lo que puedan más tarde apreciar los jueces validadores) y se estima que las dos copias son razonablemente equivalentes desde el punto de vista de la calidad de impresión y el color del papel.

El resultado final está a la vista en la figura 115, página 332 de los apéndices, tal y como queda listo para la validación.

Tipo 10 - Orquesta sinfónica

El caso número 10 es el elemento de control del grupo 2, de las partituras profesionales o comerciales. Por tanto, su destino es hacer dos copias como elementos A y B y presentar las dos lado a lado. Como caso único, el caso 10 se imprimirá directamente a partir del PDF. La muestra está representada en la figura 116, página 333 del apéndice.

Los elementos de control de cada grupo, los casos 5 y 10, consisten en parejas de elementos iguales sobre los que se fija una línea de base. El resultado de los otros casos será significativo solamente como diferencia respecto del resultado de los casos de control. A partir de la comparación entre los resultados de los elementos de control y la dominancia lateral del participante, podría detectarse un sesgo debido a este dato.

11.5.5. Montaje del material

Los modelos se imprimen en papel blanco DIN A4 de 90 g/cm².

En la figura 74 se ven las medidas definitivas del montaje de los veinte casos: 1a – 10a y 1b – 10b. El soporte tiene 478 × 320 mm. Las hojas estándar DIN A4 tienen unas medidas de 210 × 297 mm. Para realizar una composición de tipo clásico, se determina que los márgenes derecho e izquierdo de cada hoja hasta el borde del soporte sean el doble de los márgenes interiores; de igual manera, el margen inferior es el doble del margen superior. Para hallar

estas medidas se divide la diferencia entre 3 y se redondea al milímetro más cercano. Eso da los márgenes horizontales 19/20/19 y verticales 8/15 que se observan en la figura.

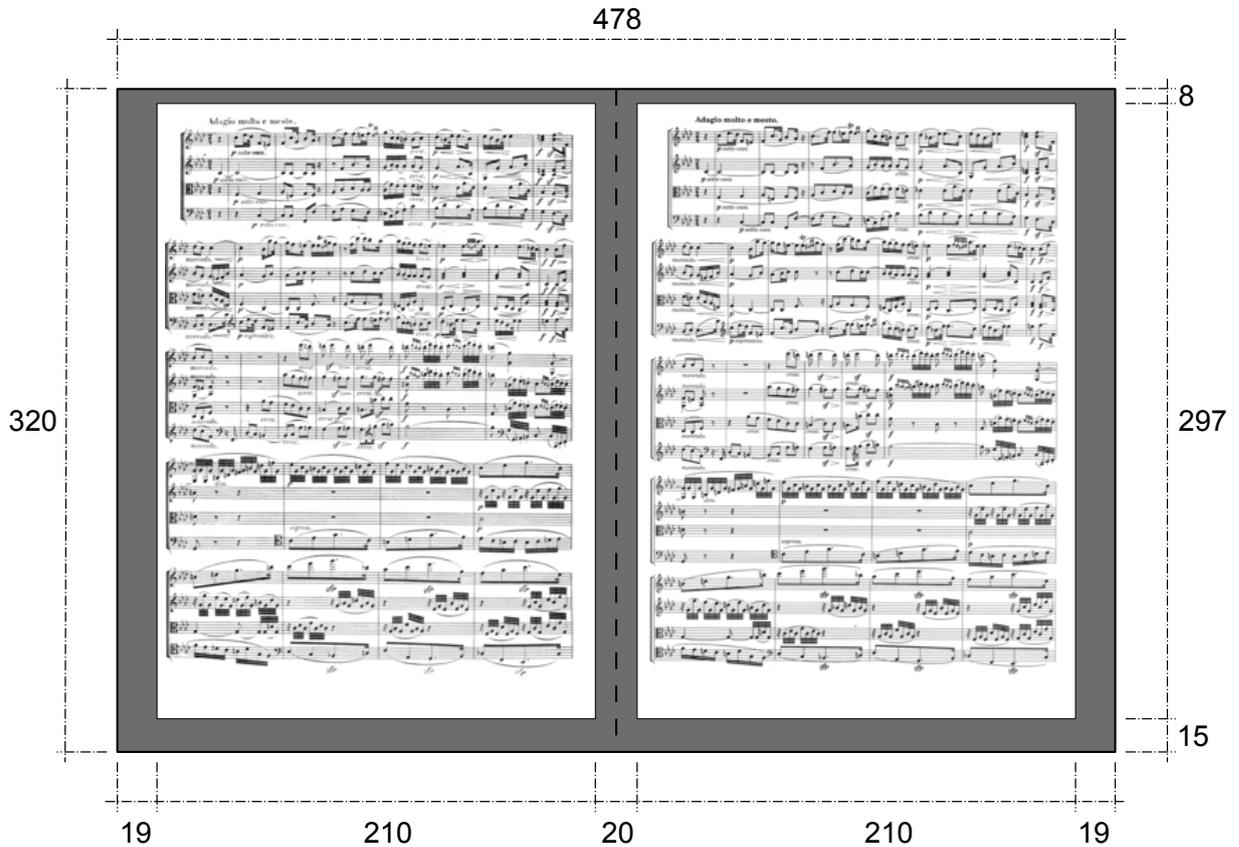


Figura 74: Esquema definitivo del montaje de los casos. Se trata del modelo 8, cuarteto de cuerda. Las imágenes empleadas para este caso pueden verse, más ampliadas, en el apéndice C, pág. 331. Medidas en milímetros.

Las hojas se montan sobre un soporte de cartulina con un peso aproximado de 250 g/m². Están pegadas por detrás en las esquinas, con cinta adhesiva de doble cara según el esquema de la figura 75.



Figura 75: Situación de los adhesivos.

11.6. Validación del cuestionario y del material

Al diseñar el proceso de validación hay que tener en cuenta la observación, ya apuntada, de que algunos músicos reconocen la editorial de una partitura a simple vista, y manifiestan preferencia por ciertas editoriales, en base a su experiencia. Esto puede producir un fenómeno de preferencias cruzadas, porque un músico podría elegir una partitura indirectamente al reconocer en ella a su editorial preferida. Como consecuencia, el proceso de validación es fundamental: entre otros factores, éste tendrá que hacer constar si, a juicio de los validadores, las editoriales son reconocibles, tomando este objetivo como uno de los medios necesarios para alcanzar el objetivo del estudio.

Para validar la encuesta y la idoneidad del material, así como el diseño del experimento, se somete a un panel de expertos que determinarán, entre otros parámetros, la uniformidad de los factores no significativos. Se obtendrá un valor para el coeficiente V de Aiken por secciones y por jueces, y se dará por bueno si supera el umbral de 0,7 en todos los casos.

Ante la consideración de si sería conveniente aumentar el número de preguntas o de opciones, hay que insistir en que el objeto de la evaluación son las partituras, y no el participante. Por ello, se consideran las preguntas sociodemográficas como complementarias o aportadoras de contexto. Ciertamente son variables independientes y podrán, en su caso, obtenerse correlaciones respecto de las variables dependientes, pero sin perder de vista que el objeto del estudio son los resultados de un cierto programa de tipografía musical.

Asimismo, según la forma en que el experimento está diseñado, algunos factores que en principio parecen relevantes para la evaluación del objeto de estudio, como pueden ser las condiciones de iluminación, el hecho de que el encuestado utilice gafas o lentillas, etcétera, debería aparecer en los dos lados de la ecuación cuando se elabore el modelo, y por tanto habrán de anularse en la influencia neta.

La validez del cuestionario y de la presentación del material del experimento será aportada por los jueces respondiendo, para cada una de las cuatro secciones de preguntas, las siguientes cuestiones:

- ¿La pregunta es pertinente?
Se pide al juez validador que puntúe de 0 a 10 si las preguntas de este bloque deben formar parte del cuestionario.
- ¿La redacción es correcta?
Se pregunta si las preguntas de esta sección están bien elaboradas en un grado de 0 a 10.

En la sección cuarta del formulario se encuentran los modelos del experimento. En el cuestio-

nario de validación, para cada uno de estos modelos, se formulan las cuestiones orientadas a determinar la equidad en el aspecto del material.

- ¿Tienen las dos partituras un tamaño de pauta³⁹ similar?
Se trata de eliminar alguna influencia del tamaño de la pauta en la percepción subjetiva de la calidad.
- Los márgenes de los dos modelos ¿son similares?
Si los márgenes son apreciablemente distintos, las partituras parecen muy diferentes. El objetivo es establecer que el participante no proyecta su predilección por unos márgenes más amplios o más estrechos.
- ¿Aprecia alguna diferencia en el color del papel?
Se trata de eliminar la influencia del color del papel en la apreciación de la calidad general de la partitura.⁴⁰
- ¿Aprecia alguna diferencia en la calidad de impresión?
Se trata de eliminar factores como el tipo de impresora utilizado y la resolución con que se ha impreso cada uno de los modelos.
- ¿Encuentra algún signo que permita identificar claramente la editorial de las partituras?
Queremos validar el hecho de que las editoriales no están identificadas expresamente, aunque los participantes muy expertos probablemente sean capaces de determinar la editorial a simple vista.

Por último se solicitará una valoración de 0 a 10 respecto a las cuestiones siguientes para validar el diseño general de la encuesta:

- Si los datos que se piden sobre cada encuestado son adecuados al objetivo de la encuesta.
- Si la cantidad de preguntas es adecuada para el objetivo.
- Si el tipo de preguntas es adecuado.
- Si el diseño de la encuesta responde al objetivo perseguido.

El cuestionario de validación definitivo que se pasa a los jueces puede verse en el apéndice E (pág. 339).

11.6.1. Variables de validación

Del cuestionario de validación surge un conjunto de datos que contiene 64 variables. Aparte de las dos primeras, que son el identificativo del juez validador y la fecha, cada variable de

³⁹Véase el glosario, pág. 351. Se utiliza la palabra “pauta”, más genérica que “pentagrama”, y se reserva esta última para las pautas de cinco líneas.

⁴⁰Tenemos en cuenta la observación que indica que los músicos prefieren, por comodidad visual, las partituras impresas en color crema o crudo, pero no totalmente blanco.

validación hasta la número 10 corresponde a una fila de la tabla 24. Siguen los diez bloques de cinco preguntas cada uno acerca de los modelos, hasta la pregunta núm. 60; finalmente, aparecen las cuatro preguntas de la sección 5 acerca del diseño general.

Tabla 24: Variables del cuestionario de validación.

Núm.	Campo	Tipo
1	Juez	Identificativo
2	Fecha	Fecha/Hora
3	Sección 1: pertinencia	Valor 0 – 10
4	Sección 1: redacción	Valor 0 – 10
5	Sección 2: pertinencia	Valor 0 – 10
6	Sección 2: redacción	Valor 0 – 10
7	Sección 3: pertinencia	Valor 0 – 10
8	Sección 3: redacción	Valor 0 – 10
9	Sección 4: pertinencia	Valor 0 – 10
10	Sección 4: redacción	Valor 0 – 10
11 – 15	Sección 4, modelo 1, preguntas 1 a 5	Valor 0 – 10
16 – 20	Sección 4, modelo 2, preguntas 1 a 5	Valor 0 – 10
...
56 – 60	Sección 4, modelo 10, preguntas 1 a 5	Valor 0 – 10
61	Sección 5: datos adecuados	Valor 0 – 10
62	Sección 5: cantidad de preguntas	Valor 0 – 10
63	Sección 5: tipo de preguntas	Valor 0 – 10
64	Sección 5: diseño general encuesta	Valor 0 – 10

11.6.2. Jueces participantes en la validación

Se ofrecieron la encuesta de validación del apéndice E y una copia del material del experimento a 23 expertos, de los que 19 devolvieron el cuestionario cumplimentado. Los jueces pertenecen a los grupos de profesionales y músicos que aparecen en la tabla 25.

Tabla 25: Composición del panel definitivo de expertos para la validación.

Ocupación del experto	Recuento
Profesores/as de conservatorio	4
Directores/as de coro	3
Cantantes profesionales	2
Licenciados/as en psicología	2
Doctores/as en musicología	2
Doctor/a (otros)	1
Asesor técnico musical	1
Compositor/a	1
Guitarrista y técnico de sonido	1
Musicólogo y profesor de piano	1
Pianista profesional y profesor de conservatorio	1
Total	19

11.6.3. Resultados de la validación: observaciones de los jueces

Los jueces del panel de validación hicieron una serie de aportaciones cualitativas que se resumen a continuación. Por ejemplo, en la pregunta Sexo, las respuestas V/H (Varón/Hembra) fueron objeto de una observación en el sentido de que resultaba «anticuada» y propuso cambiarla por la actual M/F (Masculino/Femenino). Otro juez observó que H/M para Hombre/Mujer era más apropiado. Asimismo, obtuvimos una propuesta en el sentido de preguntar más bien por el «Género» que por el «Sexo». Finalmente, un juez determinó que el género es una propiedad de las palabras y el sexo lo es de las personas, por lo que la redacción definitiva quedó como «Sexo (M/F)».

Un juez expresó dudas sobre la ortografía de la palabra *Ambidiestro* mientras que otro indicó que era igualmente correcto *Ambidiestro* que *Ambidextro*. La redacción definitiva de esta opción de respuesta quedó, pues, como *Ambidextro*.

Varios jueces pidieron especificar lo más detalladamente posible el concepto de partitura al preguntar por la frecuencia de uso. Aunque se añadieron unas notas en el modelo de encuesta delimitando claramente la idea de lo que se considera una partitura en el contexto de la investigación, no se consideró necesario leerlas cada vez, cuando se entiende que el participante tiene una idea formada al respecto que coincide con la práctica común de su entorno. Además se tiene previsto hacer todas las encuestas con el investigador presente cumplimentando el impreso, no de forma autónoma. Algo semejante se produce en la pregunta número 8, donde se pide al participante que cuente el número total de horas que dedica al manejo de partituras musicales durante una semana. Por ejemplo ¿qué se entiende exactamente por “manejo de partituras”? La acción que se tomará es especificar brevemente que hablamos de lectura, escritura, anotación o análisis de partituras musicales de cualquier tipo, como aparece en el modelo de encuesta.

Un juez propuso complementar la pregunta 11 «¿Has recibido alguna vez algún tipo de formación específica para la edición de partituras con medios informáticos?» con otra que, en caso de respuesta negativa, preguntase por qué razón o a qué motivos achaca el participante no haber recibido dicha formación. No se tuvo en cuenta por entender que se alejaba del objetivo de nuestra investigación.

Varios jueces propusieron modificar en la redacción de las preguntas el tratamiento del participante de «usted» a «tú» atendiendo al segmento de edad más numeroso al que va dirigido el estudio.

11.6.4. Resultados de la validación: *V* de Aiken por pregunta

Una vez realizadas las encuestas de validación, se obtiene la puntuación de validez mediante el Coeficiente *V* de Aiken (Aiken, 1980, 1985) a partir de las calificaciones de los jueces. Se

define una función⁴¹ en el programa R para calcular el coeficiente V y se aplica por filas y por columnas, es decir, por jueces y por secciones de preguntas. El intervalo de confianza para cada valor de V se ha estimado por el método *Score* de Wilson (Wilson, 1927, pp. 209-212; Penfield y Giacobbi, 2004; Miller y Penfield, 2006, pp. 450-452) y se ha comparado su límite inferior con un valor mínimo preestablecido de 0,7. Los resultados aparecen en la figura 76 con los límites correspondientes al 90 % y al 99 %.⁴²

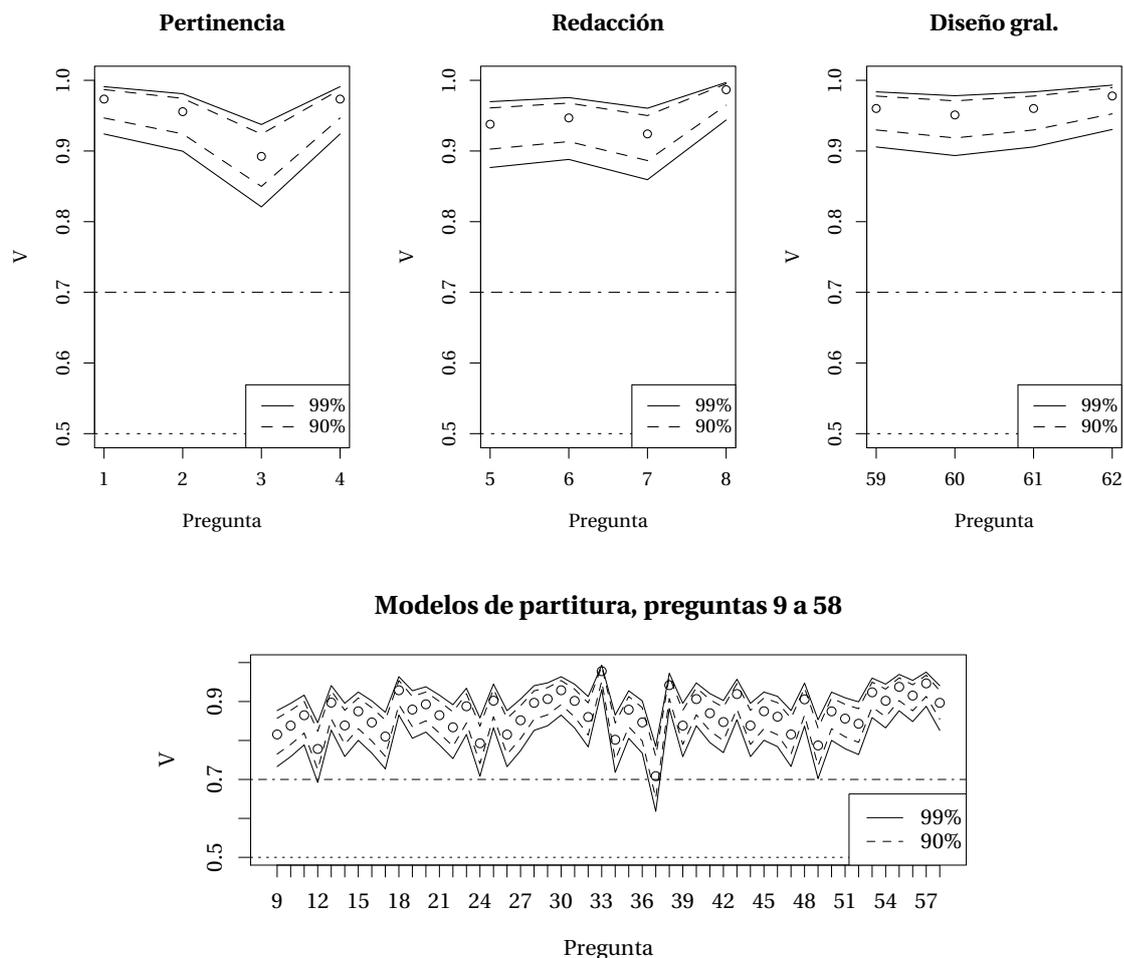


Figura 76: Valores del coeficiente de validación V de Aiken para la pertinencia y redacción de las preguntas de las cuatro primeras secciones de la encuesta, para las cuatro preguntas de la sección 5ª (diseño general de la encuesta) y para para las preguntas 9 a 58, referidas a los parámetros de los modelos de partitura del experimento.

Todos los valores se sitúan por encima del margen de 0,5 que es el umbral mínimo considerado por Cicchetti y Sparrow (1994), pero se observa un pico descendente en la gráfica de V para la pregunta 37. Esta pregunta se refería a la calidad de impresión de la partitura del ti-

⁴¹El código de la función de R que computa los valores de V y los intervalos de confianza están en el apéndice F (pág. 345).

⁴²Por claridad no hemos representado el margen 95 % que estaría entre las dos mostradas del 90 y 99 %.

po “Hoja de acordes con letra” (modelo 6) que se ha repetido con un grosor de línea distinto para asemejarlo más al original. Una vez modificado este caso, los valores están siempre por encima, incluso, del valor conservador mencionado por Charter (2003), que es de 0,7.

11.6.5. Resultados de la validación: V de Aiken por juez

Las figuras 77 y 78 muestran los valores calculados para el coeficiente V de Aiken por jueces para cada sección, y los valores medios de todas las preguntas, por juez.

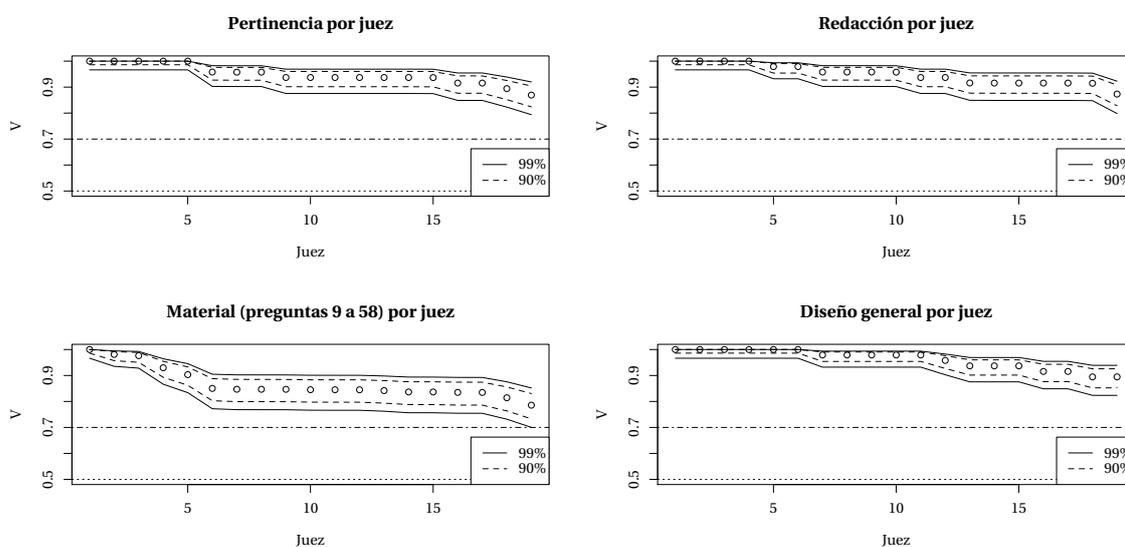


Figura 77: Valores del coeficiente de validación V de Aiken por cada juez. Pertinencia de las cuestiones, redacción, modelos de partitura (preguntas 9 a 58 referidas al material) y diseño general de la encuesta. Ordenado de mayor a menor por el valor de V .

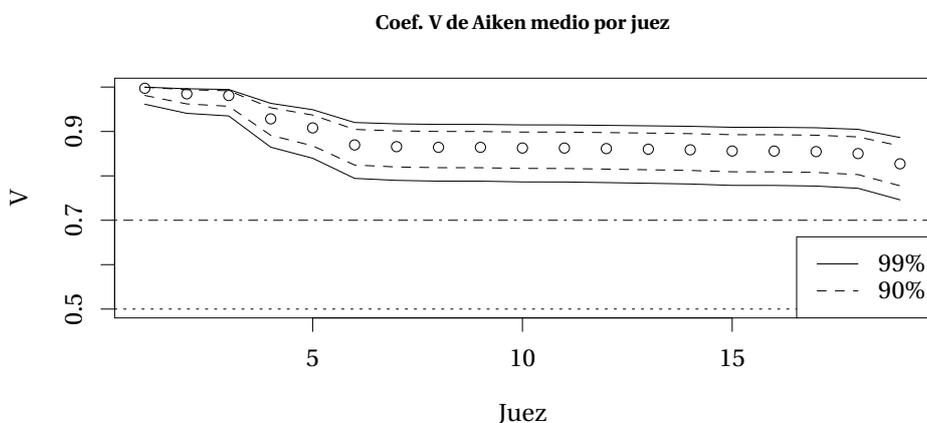


Figura 78: Valores medios, para todas las preguntas, del coeficiente de validación V de Aiken por juez. Ordenado de mayor a menor por el valor de V .

Todas las medias por juez superan el valor conservador de 0,7 que se ha preestablecido como mínimo para aceptar la validación como satisfactoria. Los valores son previos a la corrección efectuada en la pregunta 37. La necesidad de este cambio se detectó al observar los valores de V por pregunta; sin embargo, los valores por juez no acusan este pico. Esto justifica la necesidad de realizar los dos tipos de gráficos o, lo que es lo mismo, aplicar el cálculo tanto a las filas como a las columnas.

11.6.6. Consistencia interna

El valor de la variable dependiente procede de una sola pregunta, que es el experimento. Por tanto, solo tenemos una escala y no es posible evaluar la fiabilidad a través del coeficiente Alfa de Cronbach. El resto de las preguntas son variables independientes de un alcance que es desconocido antes de la realización del experimento. Sin embargo, será útil expresar mediante el Alfa de Cronbach la relación de independencia entre los rangos de estas variables y la métrica del estudio. Los cálculos que siguen computan este coeficiente a partir de la varianza del subgrupo de prueba, compuesto por los 25 primeros participantes.

En la sección Análisis estadístico se definen dos baremos diferentes para la puntuación del experimento y se comprueba que ambos miden exactamente lo mismo. En este caso, el valor del coeficiente Alfa de Cronbach para las escalas de los dos baremos alternativos es igual a la unidad. No tiene sentido el cálculo del coeficiente para otras variables como la iluminación de la escena o los descriptores sociodemográficos, que no son escalas ni guardan una relación lógica con la variable dependiente.

El Alfa de Cronbach para la escala “Número de veces que el participante prefirió el modelo de LilyPond” frente a la escala utilizada en la prueba definitiva, fue de 0,87. De estas dos escalas, la primera no tiene en cuenta la respuesta indiferente, mientras que la escala definitiva contabiliza medio punto por cada respuesta indiferente. Se interpreta la alta correlación entre ambas escalas, medida por el coeficiente Alfa, en el sentido de que haber ignorado la respuesta indiferente, contabilizando solamente las respuestas favorables, habría dado un resultado semejante.

Para otras combinaciones de variables, tomadas de dos en dos,⁴³ se obtuvo también un valor cercano a cero, siempre a partir de las varianzas del subgrupo de prueba.

El conjunto de escalas correspondiente a todas las variables desglosadas de la encuesta “Uso de partituras” arroja un coeficiente Alfa de 0,58, que es alto teniendo en cuenta que no se

⁴³Las parejas de variables examinadas son: la colocación Izquierda o Derecha (I/D) de la partitura seleccionada por el participante en los modelos de control; I/D de control frente a la dominancia lateral diestro/ambidextro/zurdo (D/A/Z), I/D excluidos los casos de control frente a D/A/Z, I/D totales frente a D/A/Z, y si necesitan graduación para la vista frente a las horas semanales dedicadas.

esperaba una alta consistencia entre este conjunto de escalas, puesto que responden a los usos de distintos tipos de partitura. El valor cuando se comparan dos a dos resulta de 0,72 entre las escalas “Uso de partitura original” y “Uso de partitura escaneada o fotocopiada”, que son las que tienen la puntuación más alta. La consistencia entre otras parejas de escalas no tiene interés, a pesar de dar un valor positivo del coeficiente, porque su puntuación respectiva es menor.⁴⁴

11.6.7. Estabilidad del instrumento: fiabilidad test/retest

Se efectuó la encuesta por segunda vez al mayor número posible de participantes, resultando un subgrupo de prueba compuesto por 25 personas con una separación temporal mínima de dos semanas respecto de la primera. Como son datos repetidos, es de aplicación el análisis ANOVA⁴⁵ (Fisher, 1954; Tejedor, 1999; Spiegel y col., 2007, pp. 335-371) que se obtiene a partir de la varianza de las mediciones.

Restringimos el análisis de la fiabilidad test/retest a la puntuación del experimento (pregunta 12) según el baremo establecido, que tiene una escala de valores 0 a 8 en pasos de medio punto. Los dos grupos para el análisis ANOVA son los resultados de la primera y segunda vez. Se analiza el modelo con el programa R y se obtiene una tabla con los datos que se muestran en la tabla 26.

Tabla 26: Análisis de varianza.

Parámetro	Intra grupos	Residual
Gl	1	24
Suma C.	0.04	31.92
Media C.	0.042	1.45
Valor-F	0.029	
Pr(>F)	0.867	

donde:

- Gl = grados de libertad
- Suma C. = desviaciones intra grupos y residual
- Media C. = varianzas intra grupos y residual
- Valor-F = valor del estadístico F de Fisher, calculado como (varianza intra grupo)/(varianza residual)
- Pr(>F) = valor p , probabilidad de que estas varianzas correspondan a la misma distribución

⁴⁴Por ejemplo: si el uso de partitura manuscrita sobre papel tiende a presentar una frecuencia semanal (no diaria), y lo mismo puede decirse del uso de partitura escrita en pizarra, no sorprende que la consistencia entre estas escalas sea aparentemente alta como la que existe entre los usos de partitura original y fotocopiada, que son de frecuencia diaria en el subgrupo de prueba.

⁴⁵ANOVA (del inglés *Analysis of variance*) es una generalización de la prueba t de Student para parejas de datos. No se aplica el Coeficiente de Correlación Intraclase (CCI) porque solamente es válido para la fiabilidad de mediciones continuas, según Candela (1992).

Como se obtiene un valor p elevado, $0,867 > 0,05$, se acepta la hipótesis nula H_0 : los resultados del experimento repetido para el subgrupo de prueba son estadísticamente iguales y por tanto la escala es estable en el tiempo.

11.7. Puesta en práctica

Por las características del material empleado para la prueba, no cabe en este caso la realización *online* de la encuesta y se descarta el envío de todo el material a los participantes para que cumplimenten el cuestionario de forma autónoma.

Aunque esta restricción limita el tamaño de la muestra, las ventajas derivadas del hecho de que todas las pruebas de medida estén realizadas personalmente por el investigador compensa la referida limitación. A ello contribuye también la premisa anteriormente expuesta de que el objeto de la evaluación son las partituras, no los voluntarios. Los parámetros de los voluntarios son variables (en su mayor parte nominales, excepto la edad) que en principio son independientes de las partituras que eligen.

Condiciones de iluminación

Se registran las condiciones de iluminación de la escena para cada experimento.⁴⁶ Las dos alternativas son utilizar un fotómetro de luz incidente, o uno de luz reflejada. Todas las cámaras fotográficas incorporan un fotómetro de luz reflejada para el cálculo de la exposición correcta. La medida de las condiciones de iluminación a partir de la luz reflejada en lugar de hacerlo a partir de la luz incidente está justificada siempre que se emplee la misma hoja de calibración para todas las medidas, y éstas se realicen con el mismo ángulo de visión que el participante respecto a la hoja. Manteniendo constantes estos factores, el empleo de un fotómetro de luz incidente no habría aportado ninguna ventaja derivada de una mayor precisión.

Se ajusta la sensibilidad de la cámara a 100 ISO y se fotografía la hoja patrón blanca de calibración desde la posición del punto de vista del encuestado. La hoja está situada en el mismo lugar que los modelos que se presentan al participante, y con la misma inclinación. La cámara se ajusta en el modo de programa (modo "P"), de tal forma que se seleccionan, de acuerdo con la luz reflejada por la hoja patrón, una velocidad de obturación y una apertura del diafragma que la cámara registra junto a la fecha y la hora de la exposición en los metadatos del archivo de imagen. Cuando posteriormente se examina la imagen y se extraen estos metadatos, puede aplicarse la fórmula del valor de exposición para obtener una medida indirecta de

⁴⁶El registro efectuado una vez puede servir para un grupo de experimentos, en caso de que se realice con luz artificial, lo que da para un local determinado unas condiciones de iluminación repetibles, como ya se refirió.

las condiciones de iluminación que reinaban en la escena en el momento de la realización de la prueba.

El valor de exposición se calcula, según la escala estándar que fue definida por Ray (2000, p. 318), como $EV = \log_2 \frac{N^2}{t}$, donde N es la abertura relativa del diafragma (el número f), y t es el tiempo de exposición en segundos. El tiempo de exposición viene expresado como una fracción cuyo numerador es la unidad, de forma que $t = \frac{1}{p}$. En la hoja de recogida de datos aparecen los valores medidos del denominador p y de f , así como el valor EV calculado.⁴⁷

Tiempo de respuesta

Se barajaron las siguientes opciones para registrar o controlar el tiempo de respuesta de los voluntarios, en la última fase de la encuesta.

- El participante tiene un tiempo fijo (por ejemplo, 10 segundos) para observar los dos modelos y después un máximo de 5 segundos para expresar su preferencia, en cada pareja.
- Se mide el tiempo que el participante tarda en decidirse por una de las dos. Este tiempo será una variable que podría indicar la dificultad que el encuestado experimenta en función del parecido entre ellas.
- No hay límite de tiempo: los voluntarios pueden observar los modelos tranquilamente y emitir un juicio razonado.

Sin pretender introducirnos en el campo de la psicología, es previsible que algunos participantes deseen mostrar una gran intuición y criterio firme, contestando lo más rápidamente posible. Sin embargo, dado que en situaciones reales, el tiempo que una persona se tomará para expresar su preferencia por uno u otro modelo dependerá de su carácter y experiencia, y puede ser un factor muy variable, optamos por dejar a los voluntarios que se tomen todo el tiempo necesario antes de responder.

Desarrollo del experimento

Según se anunció en la sección Procedimiento, este apartado detalla los pasos que se siguen durante la sección cuarta del cuestionario, el experimento comparativo. Para cada participante:

- Se selecciona una combinación de posiciones A/B de la lista y se marca como utilizada.

⁴⁷En la hoja de cálculo, la fórmula correspondiente para EV a partir de las columnas p y f es $\text{LOG}((f^2 * p)) / \text{LOG}(2)$.

- Se prepara un juego de casos A/B según la combinación seleccionada. Los casos sobrantes forman otro juego A/B complementario. Se puede usar una vez cada juego de casos, para participantes diferentes.
- Se mide la iluminación de la escena con el fotómetro de la cámara, ajustada a 100 ISO, sobre la hoja de calibración blanca, que es siempre la misma e igual a la que se ha usado para imprimir el material del experimento.⁴⁸ Se anotan los valores f y T. El valor EV puede calcularse más tarde.⁴⁹
- Un solo participante realiza el experimento cada vez, de forma aislada. De esta forma se evita cualquier presión debida a la presencia de otras personas.
- El participante se sienta enfrente del investigador. Se selecciona para él un identificador aleatorio de la lista y se marca como utilizado. Se anota el identificador junto a su nombre real en una lista confidencial.
- Después de las 11 primeras preguntas, si el participante ha manifestado en la pregunta número 4 usar habitualmente corrección para la vista (gafas o lentillas) para el uso de partituras, le pedimos que se las ponga.
- El participante elegirá a suerte uno de los dos juegos, o se usará el complementario del juego utilizado con el participante inmediatamente anterior.
- Para el experimento (pregunta 12), después de formulada la pregunta, los diez casos del juego elegido se van abriendo de uno en uno frente al participante, que va seleccionando el preferido, diciendo “izquierda”, “derecha” o “indiferente”. Se pide una respuesta del tipo “izquierda” o “derecha”, que es preferible a que el participante señale una de las dos, para evitar errores, porque si el investigador está sentado frente al participante, su propia izquierda es la derecha del otro, y viceversa. Se anota la respuesta del participante, literalmente, y después se compara con la etiqueta “A” o “B” que tiene cada modelo, para determinar cuál ha sido la elección. No es posible preguntar directamente al entrevistado si prefiere el modelo original “A” o el recreado “B” porque éste no sabe cuál es cuál. Se anotan las observaciones que puedan ser de interés.
- Durante el experimento se cuida de que los soportes estén siempre planos sobre la mesa y que estén igualmente iluminados (por ejemplo, que la cabeza no bloquee la luz).
- Se aparta la hoja de recogida de datos hasta que se registren las respuestas. Después se archiva en orden del identificador.

En la figura 79 se representa esquemáticamente la situación de los elementos durante el desarrollo del experimento.

⁴⁸Para ajustar el balance de blancos de un escáner se utiliza una hoja de calibración blanca que habitualmente acompaña al aparato. Sin embargo, esta hoja suele ser de un tamaño demasiado pequeño para fotografiarla de manera que ocupe todo el campo del objetivo. Por ello se ha empleado un papel en formato DIN A4 igual a los empleados para el material del experimento.

⁴⁹Los valores f y T también pueden recuperarse más tarde de los metadatos del archivo de imagen.

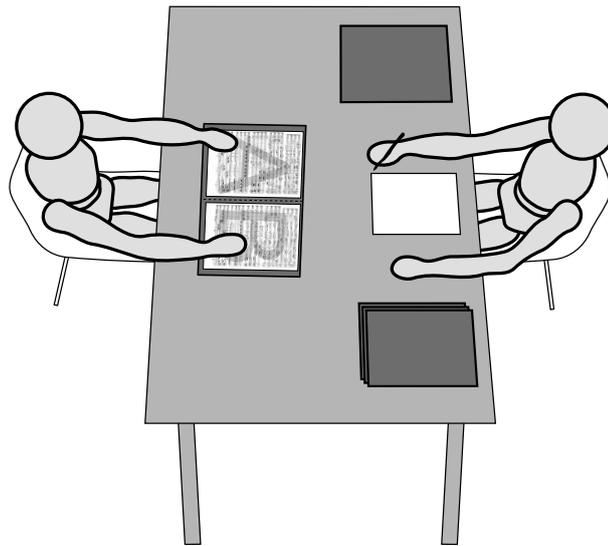


Figura 79: Esquema de la situación de los elementos del experimento. El participante (a la izquierda), observa uno a uno los diez casos que el investigador (a la derecha) le va presentando, y se pronuncia sobre su preferencia por uno de los elementos de cada caso, A o B, o manifiesta su indiferencia, de acuerdo con la pregunta.

Registro y seguimiento

Los datos se consignan en una hoja de cálculo que se guarda en un formato de texto. Esto permite un seguimiento de las modificaciones y un procedimiento de copia de seguridad basado en la teoría de las versiones. Los detalles están en la sección Software utilizado (pág. 258).

11.7.1. Fase piloto

La experiencia piloto, previa a la puesta en práctica, permite refinar algunas preguntas y reducir la ambigüedad. Se ha probado con una versión preliminar del cuestionario y del material del experimento aplicados a un grupo reducido.

La actividad declarada por el participante es una pregunta que cambió del tipo respuesta única a respuesta múltiple, ya que los participantes deseaban reflejar que se veían a sí mismos como músicos además de como profesores o estudiantes, en su caso. La posibilidad de incluir cualquier combinación obliga a desglosar esta respuesta en sendas variables nominales.

Se ha comprobado que la iluminación de la escena, medida a través del fotómetro de una cámara *reflex* enfocada a la hoja patrón blanca de calibración, puede medirse con la cámara de un teléfono móvil con una exactitud tal que a partir del tiempo de exposición y la apertura de diafragma, a 100 ISO, la discrepancia en los valores EV calculados es menos de una unidad. La única diferencia es que en el caso del teléfono móvil, la apertura de diafragma es fija.

En la pregunta número 7, el apartado *b*) pide “¿Con qué frecuencia utilizas partituras escaneadas o fotocopiadas?”. En la número 10 se pregunta si el participante ha instalado algún programa de edición de partituras. Son preguntas delicadas que el voluntario puede interpretar como que se le trata de extraer una confesión sobre alguna práctica ilícita. Esto confirma la necesidad de implantar un esquema de anonimización para todos los participantes.⁵⁰

Un participante dijo durante la prueba piloto que la cuestión determinante para preferir una partitura había sido plantearse por cuál de ellas daría dinero. Esta formulación de la pregunta no estaba inicialmente en el texto de la encuesta, pero se incorpora para el modelo definitivo.

Otro aspecto que la experiencia piloto detectó es la conveniencia de recordar al participante que siempre tiene la opción de responder «indiferente», ya que se observa una cierta tendencia a olvidarlo. Cuando se confirmó que el participante había olvidado que disponía de la opción de respuesta indiferente, se repitió la prueba desde el principio. Este y otros casos similares, descritos en la sección 12.2, dan cuenta de esta situación. En todo caso, ya se ha establecido que la indiferencia no es determinante para el contraste de la hipótesis del estudio.

11.8. Análisis estadístico

11.8.1. Baremo para la puntuación del experimento

Para contabilizar el éxito de cada experimento se consideran las siguientes alternativas:⁵¹

- Contabilizar la diferencia entre los casos favorables y los desfavorables (un punto positivo por “B” y un punto negativo por “A”), ignorando las respuestas indiferentes.
- Contar las respuestas favorables y sumar la mitad de las indiferentes (un punto por “B” y medio punto por “X”).

Ahora bien, estas dos soluciones son completamente equivalentes. Al analizar las frecuencias de las puntuaciones obtenidas con cada uno de estos baremos, se puede comprobar que los histogramas de $B - A$ y de $B + X/2$ (salvo por el rango de valores del eje de la variable) tienen una forma idéntica, porque para cada valor v de $B + X/2$, hay exactamente un valor de $B - A$

⁵⁰Los originales en papel suelen ser preferibles; sin embargo, la fotocopia en sí no es ilegal (aunque muchos originales tratan de disuadir de esta práctica en la primera página), y paga un canon a entidades como CEDRO.

⁵¹La nomenclatura A/B del baremo de puntuación corresponde a la que hemos utilizado para denominar los elementos original y recreado durante la elaboración del material. En el contexto de los datos del experimento, “B” es la puntuación favorable (veces que el participante prefirió LilyPond) y “A” es la desfavorable (veces que prefirió el original).

que es $2v - 8$, lo que hace que ofrezcan la misma distribución.

Para probarlo, se observa en primer lugar que, como estamos excluyendo los dos casos de control, se cumple que

$$A + B + X = 8$$

donde

A = número de veces que el participante prefirió el modelo original.

B = número de veces que el participante prefirió el modelo recreado con LilyPond.

X = número de respuestas indiferentes.

El primer baremo es

$$B - A$$

Se despeja A de la primera expresión:

$$A = 8 - B - X$$

Se sustituye en el primer baremo, y se obtiene

$$\begin{aligned} B - A &= B - 8 + B + X \\ &= 2B + X - 8 \\ &= 2(B + X/2) - 8 \end{aligned}$$

que es la mencionada función lineal $2v - 8$ sobre el segundo baremo $B + X/2$.

La media y la mediana también están, obviamente, en la misma proporción $2v - 8$. El valor del coeficiente Alfa de Cronbach para la combinación de los dos sistemas de puntuación es igual a la unidad, lo que significa que la métrica consistente en contabilizar un punto positivo por cada vez que el participante prefiere el modelo recreado y medio punto por cada respuesta indiferente, equivale de forma lineal a la métrica consistente en restar simplemente a las veces que el participante prefirió el modelo recreado con LilyPond (elecciones positivas) un punto por cada elección de las negativas (o del modelo original).

Esta relación sencilla es posible debido a que no se ha considerado la posibilidad de registrar una cuarta respuesta (en adición a las tres utilizadas A , B y X), que significaría "ninguna de las dos" como algunos participantes, a quienes no les gustaba ninguno de los modelos, sugirieron durante la fase de prueba piloto.

En resumen, lo expuesto significa que frente a la tarea de diseñar un baremo que sirva de puntuación para el experimento, se puede elegir uno cualquiera de los dos sistemas (manteniéndolo a lo largo de toda la encuesta) y obtener el mismo resultado en ambos casos al contrastar la hipótesis.

11.8.2. Estadísticos y software empleado

Para el análisis de la asociación entre el uso de distintos tipos de partitura se emplean χ^2 de Pearson (Pearson, 1900), el coeficiente ϕ (González, Lise y Felpeto, 2013), el coeficiente de contingencia (Martel y Vegas, 1996), la V de Cramér (Lorenzo, 2007), el test exacto de Fisher (Gómez, 2008) y el cómputo de los residuos tipificados corregidos (Castejón, Lechuga y Martínez, 2015). Otras referencias más generales ya han aparecido en el compendio bibliográfico (capítulo 1).

Se elabora un modelo loglineal (Christensen, 1997b; Simkiss y col., 2012) para producir un gráfico de asociación de Cohen y Friendly (Cohen, 1980; Friendly, 1992).

Las reglas de asociación (Borgelt y Kruse, 2002) para el uso de partituras se ordenan por el valor *lift* en función de las medidas de apoyo y confianza de cada regla.

Para la relación entre las horas de manejo de partituras y la edad del participante, se examina el coeficiente de correlación del modelo lineal de mínimos cuadrados y el valor p para evaluar el grado de significación. Para encontrar la edad a la que se produce el máximo de horas, se optimiza la función resultante de un ajuste polinómico (Cleveland, Grosse y Shyu, 1992).

El contraste de los baremos de puntuación alternativos, por participante, en el experimento comparativo, se efectúa mediante una prueba t de Student pareada (Gutiérrez, 2007) con 105 grados de libertad y se examina el valor p para ponderar la significación. Se da un intervalo de confianza del 99 % para la media.

Para efectuar el contraste del baremo con la media teórica μ_0 se emplea una prueba t de Student de una cola con 105 grados de libertad y valor p (Fisher y col., 1925). Se calcula un intervalo de confianza del 99 % para la media.

Los modelos lineales para el efecto de la edad sobre la elección de A, B e indiferencia, dan un coeficiente y un valor p para cada uno de los casos considerados: Original solo, LilyPond solo, Indiferencia sola y Baremo con indiferencia.

El análisis de los subgrupos que rechazan los distintos modelos, o se muestran indiferentes, se realiza gracias al empleo de árboles de decisión basados en ANOVA (Berlanga, Rubio y Vilà, 2013; Calle y Sánchez-Espigares, 2007; Rojo, 2006). Para simplificar los modelos y evitar la saturación, se efectúa una «poda» de los árboles en función del error resultante.

El mecanismo de registro, control de versiones y copia de seguridad se efectúa con el sistema Git (Greaves y col., 2010; Torvalds y Hamano, 2005). La hoja de cálculo se guarda como CVS,⁵² que es un formato de texto. El seguimiento de las modificaciones (o “versiones”) de

⁵²CVS son las siglas en inglés de *Comma separated values* (valores separados por comas).

archivo de texto aprovecha las funciones de Git como sistema de control concurrente de las versiones. Es concurrente porque permite modificar el mismo archivo en varios ordenadores diferentes y reconciliar posteriormente todas las modificaciones. También es descentralizado, lo que significa que la reconciliación se puede efectuar en cualquiera de los ordenadores, para después propagarse al resto. Al mismo tiempo, el registro histórico de estas versiones sirve como copia de seguridad distribuida cuando la reconciliación se efectúa sobre uno o varios soportes de memoria externos.

Los datos se trasladan después, en formato CVS, al paquete de cálculos matemáticos R (R-CDT, 2000) para su análisis estadístico.

12. Estudio 2. Resultados y discusión

Nuestro experimento sobre la aceptación de LilyPond como herramienta para la producción de partituras, por parte de los músicos, es la pregunta número 12 de una encuesta. En la aplicación de este instrumento han colaborado 106 participantes anonimizados, a los que no se evaluaba como objeto de la investigación, sino que se les preguntaba por su preferencia personal sobre las diez parejas de modelos de partitura que se les presentaban. La mitad de estos modelos estaban tomados de situaciones reales y la otra mitad vueltas a producir mediante LilyPond a partir de aquéllos. Se enfrentó cada original con el modelo recreado y se pidió a cada participante que seleccionara uno de ellos en cada pareja según su criterio de lo que consideraba “una edición preferible” o “una partitura más hermosa”. A continuación se resumen y analizan las respuestas a las preguntas 1 a 11 del cuestionario; el resultado del experimento puede verse en la sección 12.5 (pág. 280) y los resultados comparativos por modelos en el apartado 12.5.10 (pág. 291). El análisis estadístico se realizó con el programa R (R-CDT, 2000).¹

12.1. Iluminación de la escena

De los valores de apertura de diafragma y velocidad de obturación a 100 ISO registrados para cada participante a partir de la fotografía de una hoja de papel blanco, se han calculado los valores de exposición EV, dando como resultado las frecuencias del histograma de la figura 80.

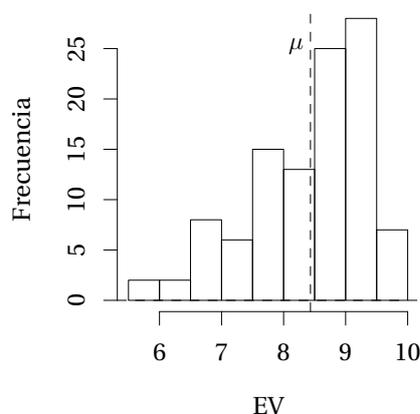


Figura 80: Histograma de los valores de exposición EV para la iluminación de la escena.

¹Los datos de las tablas proceden del programa y utilizan el punto decimal para los valores fraccionarios; en el texto se ha usado la coma decimal según la norma española.

El valor EV medio fue de 8,4. En el histograma aparecen todos los valores excepto un valor extremo de EV = 16, que corresponde a la luz directa del sol. La referencia para estos valores de EV frente a los locales tipo en que se realizaron las encuestas, aparecen en la tabla 27.

Tabla 27: Algunos locales utilizados, con sus iluminaciones.

EV	Modelo de local	Tipo de iluminación
5.6	Despacho	Natural, suave
6.2	Salón	Natural, suave
6.5	Sala de estar	Natural, suave
6.6	Despacho	Natural
7.1	Aula	Natural
7.3	Despacho	Fluorescente
8.3	Aula	Fluorescente
8.6	Tribuna de un salón de actos	Fluorescente, intensa
8.8	Sala de reuniones	Natural, intensa
9.0	Aula	Natural, intensa
9.7	Aula	Fluorescente, muy intensa

La iluminancia recomendada por Pangert y Loock (2004) para la lectura de música es de 300 lux o preferiblemente 500 lux, similar a la recomendada para una oficina; el primer valor coincide con los valores más bajos medidos en el estudio.² En la misma referencia se indica también que al aire libre los valores de iluminancia mayores no causan ningún problema, gracias a la adaptación de la pupila del observador. Asimismo, observan unas necesidades de luz que crecen con la edad, afirmando que los requisitos de luz de una persona de 65 años para la lectura de un libro bien impreso son 2,66 veces mayores que los de una persona de 25 años.

Este informe técnico es una importante referencia respecto a la ergonomía de la lectura de música. Acerca de la calidad de los originales y copias de material de notación musical, lista una serie de criterios que incluyen, entre otros, los siguientes:

- El tamaño de la notación ¿es el adecuado para su visión cómoda a la distancia normal de acuerdo con el instrumento de que se trate?
- ¿Tienen los símbolos un cuerpo de tamaño suficiente?
- La impresión de inyección de tinta ¿es lo bastante negra?
- El papel utilizado ¿es blanco, o está ligeramente teñido?
- ¿El papel es mate, o produce reflejos?
- ¿Se aprecian claramente los huecos entre las barras de semicorchea?
- El contraste de la copia ¿es tan grande que faltan líneas del pentagrama?

El valor medio observado corresponde a 640 lux. A la vista de esta tabla y del histograma de la figura 80, puede afirmarse que las condiciones medias de iluminación fueron buenas o muy buenas, para los participantes de todas las edades.

²Convirtiendo los valores de exposición a iluminancia en unidades de lux. Una conversión similar, usando una tarjeta de gris 18 %, puede verse en la guía de Kodak (Eastman Kodak Co., 1999)

El posible impacto de la iluminación de la escena sobre los resultados del experimento se ven en el apartado 12.5.12. Los músicos necesitan unas buenas condiciones de iluminación para la lectura de partituras. Debès y col. (2004) registran un porcentaje de insatisfacción del 53 % en un estudio en el que exploran los problemas de salud de los músicos de orquesta. Las reclamaciones acerca de la calidad de impresión de las partituras daba un 26 % de insatisfechos. Encontró que había asociación ($p < 0,05$) entre las quejas por la sensación de incomodidad visual y aquellas derivadas de la calidad de impresión de las partituras. A su vez, detectó asociación ($p < 0,01$) entre la incomodidad visual y una fatiga general inusual. En su conjunto, estos resultados sugieren que la buena calidad de las partituras y unas buenas condiciones de iluminación son factores que pueden ayudar a evitar la fatiga general, sin limitarse a la fatiga visual. Es importante señalar que en el estudio mencionado, la prevalencia de quejas relacionadas con la visión resultaron ser independientes de la edad, el sexo, el peso o la talla.

12.2. Participantes

Todas las encuestas fueron personales, lo que amplía la proporción de respuestas completas y disminuye los casos dudosos o imprecisos. Se mencionan dos casos de participantes que ofrecieron dudas en cuanto a la inclusión de sus resultados.

Un participante escogió el modelo de la derecha en todos los casos (excepto los de control). Se le hizo el retest inmediatamente con el juego complementario, que tiene los modelos B en el lado opuesto, y volvió a elegir el de la derecha en los cuatro primeros casos. Antes de llegar al primer caso de control se decidió eliminar a este participante. Dado que el participante tomó los soportes en su mano, formando un pequeño ángulo en lugar de permanecer completamente planos sobre la mesa, la sospecha es que había una pequeña diferencia de iluminación que le hacía elegir siempre el del mismo lado, o quizá el participante tiene un lado preferente muy marcado. La iluminación de la escena en este caso era algo inferior a la media del resto de las encuestas.

En ocasiones, las encuestas son de una validez dudosa por circunstancias diversas. Por ejemplo, una encuesta se hizo a pleno sol. Al medir la luz, se obtuvo una exposición con un tiempo inferior a $1/10000$ s. y un índice EV de 16,0. Existe la sospecha de que en estas condiciones extremas el experimento no es fiable; sin embargo Debès y col. (2004) indican que gracias a la adaptación de la pupila no existe ningún problema con el manejo de partituras musicales al aire libre.

En las observaciones finales se da cuenta de algunas otras situaciones que ofrecieron duda. Teniendo en cuenta el caso eliminado y el dudoso, que se mantiene, el tamaño final de la muestra es $n = 106$.

12.2.1. Datos sociodemográficos

Edad de los participantes

La tabla 28 muestra el resumen de las edades de los participantes.

El 50 % de los participantes tienen entre 22 (la edad más repetida) y 42 años de edad. La edad media es de 32 años. La mediana toma el valor 30,5, que es fraccionario porque la muestra tiene un tamaño par y los valores centrales son 30 y 31 años.

Tabla 28: Edades de los participantes.

Parámetro	Mínima	1er. cuartil	Moda	Mediana	Media	3er. cuartil	Máxima
Edad	11	22	22	30.5	32.36	42	63

Sexo de los participantes

El reparto de los participantes por sexos puede verse en la tabla 29.

Tabla 29: Sexo de los participantes.

Sexo	Recuento	Proporción
Mujeres	53	50.0 %
Hombres	53	50.0 %

Aunque la muestra es circunstancial y no se ha diferenciado entre ambos grupos, las proporciones de hombres y mujeres coinciden exactamente. Para 106 participantes cuyo sexo fuese aleatorio, la probabilidad de que ambos estén igualmente representados³ es del 7,7 %.

Dominancia lateral

La tabla 30 recoge la condición declarada como zurdo/diestro/ambidextro por los participantes. Los no diestros son menos de un 15 % en esta muestra.

³La fórmula general para el cálculo de la probabilidad de obtener exactamente q resultados en n intentos, donde cada intento puede producir dos resultados, es $n!/((q!)^2 \times 2^n)$.

Tabla 30: Lado preferente de los participantes.

D/Z/A	Recuento	Proporción
Diestros	91	85.8 %
Zurdos	9	8.5 %
Ambidextros	6	5.7 %

Graduación para la vista

Los participantes que declararon necesitar graduación para la vista durante el manejo de partituras musicales son 60 (el 56,6 %) frente a 46 (el 43,4 %) que no la necesitan.

Según el sexo de los participantes, el mayor porcentaje de personas que necesitan graduación para la vista está entre las mujeres, con un 62,2 % frente al 37,7 % que no la necesitan, mientras que entre los hombres está equilibrado. No se aprecia asociación significativa entre estas variables ($\chi^2 = 1,38$, $p = 0,24$, V de Cramér = 0,11).

Si se considera la ocupación musical de los participantes, se observa que el mayor porcentaje de personas que usan graduación para la vista es el 70 % de los compositores, seguido del 63 % de los estudiantes. El menor porcentaje es el 50 % que corresponde a “Otros” seguido en orden creciente por el 55 % de los directores y los aficionados. Las variables no guardan ninguna asociación ($\chi^2 = 0,10$, $p = 0,99$, V de Cramér = 0,12).

Según el nivel de estudios musicales, el mayor porcentaje de personas que necesitan graduación para la vista se encuentra entre los que han completado estudios de conservatorio de nivel profesional (principalmente estudiantes de grado superior), con un 58,6 %, y entre los que han completado estudios superiores (principalmente profesores), con un 58,8 %. No existe asociación ($\chi^2 = 2,08$, $p = 0,84$, V de Cramér = 0,14).

Actividad declarada por los participantes

La figura 81 presenta la actividad o actividades que los participantes manifestaban profesar (con posibilidad de respuesta múltiple) dentro del campo musical. Se muestran los totales por cada respuesta. El 73 % de los participantes se declaran “músicos”, seguido de un 60 % de estudiantes y un 50 % de profesores. Están menos representados los directores de alguna agrupación (19 %), los compositores (9 %) y los que se autodenominan como aficionados (8 %). En el apartado “Otros” se cuentan 1 archivero/a musical y 1 coordinador/a educativo/a. Solo el 4,7 % de los participantes se reconocían exclusivamente como “músicos” y no profesores, estudiantes o alguna de las demás actividades. La caracterización de los participantes como “músicos” presenta, como se ve, unos límites flexibles.

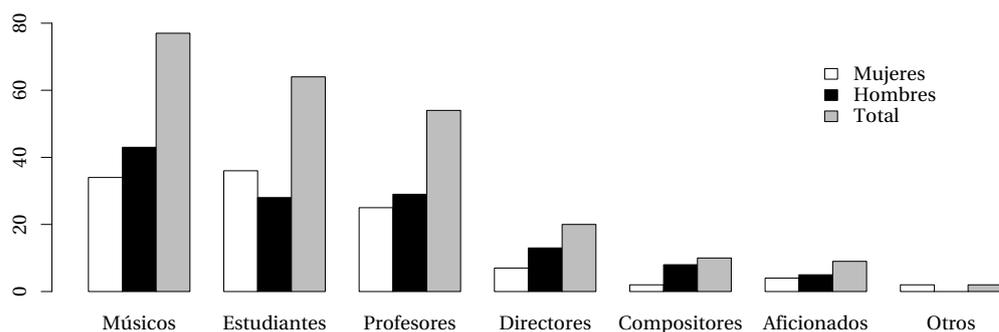


Figura 81: Actividades declaradas por los participantes, con distinción de su sexo. Posibilidad de respuesta múltiple. Ordenado por frecuencias decrecientes.

A lo largo de toda la memoria se emplea el término “músicos” a pesar de que no todos los encuestados se adscribieron a él en la respuesta. Las categorías ofrecidas en la pregunta parecen cubrir aceptablemente el abanico, ya que solamente dos personas optaron por autodenominarse dentro del apartado “otros”.

Al distinguir por sexo, dado que ambos están equilibrados, podría pensarse que el exceso de mujeres estudiantes compensaría al exceso de hombres en el resto de las ocupaciones. Sin embargo, esto no ocurre así porque la pregunta admite respuestas múltiples y cada participante puede pertenecer a más de una categoría.

La tesis de Ribeiro (2010) sobre el uso de programas de tipografía musical encuestó a 13 personas. Las opciones en la pregunta sobre relación con la música eran: aficionado / estudiante / profesional, con recuentos de 9 / 1 / 3.

Nivel de estudios musicales completados por los participantes

La muestra se compone de un 55 % de estudiantes de grado superior, un 32 % de profesores titulados y un 13 % de titulados elementales y otros. El resumen aparece en la tabla 31.

Tabla 31: Nivel de estudios musicales completados por los participantes.

Respuesta/Estudios	Recuento	Proporción
a / Ninguno	1	0.94 %
b / Academia	2	1.89 %
c / Elementales	9	8.49 %
d / Profesionales	58	54.72 %
e / Superiores	34	32.08 %
f / Doctorado	2	1.89 %

A diferencia del dato de la autodenominación, el de los estudios completados es objetivo. Los participantes sin estudios formales de música se incluyen sin ninguna reserva porque todos

ellos superan el número establecido de años de experiencia diaria en el uso de partituras musicales.

12.3. Uso de partituras

12.3.1. Valores globales

Al preguntar sobre la frecuencia de uso de los distintos tipos de partitura, ningún encuestado señaló la opción *h* (otros). Se elimina esta pregunta en los análisis; los tipos de partitura ofrecidos como respuestas posibles han resultado suficientes para cubrir el espectro.

Las dos primeras opciones de respuesta (uso diario, uso semanal) para algún tipo de partitura ya acumulan el 100 % de las observaciones como se aprecia en la tabla 32.

Tabla 32: Participantes que utilizan algún tipo de partitura sobre una base diaria o semanal.

Uso	Diario	Semanal
Recuento	87	82
Porcentaje, aislado	82.08 %	77.36 %
Porcentaje, acumulado	82.08 %	100.00 %

La estadística simple de las variables “Uso de partituras” arroja las medidas que aparecen en la tabla 33. Aunque expresa cinco valores ordenados de frecuencia, no tiene sentido el valor medio ya que es una variable nominal.

Tabla 33: Uso de partituras. Recuento por tipos y valores más repetidos.

	Originales	Fotocopias	Manuscr.	Pizarra	Editadas	Propias	Pantalla
Diario	39	82	4	4	11	4	10
Semanal	30	20	15	15	26	6	30
Mensual	18	4	22	10	41	20	19
Anual	13		34	10	15	22	13
< Anual	6		31	67	13	54	34
Moda	Diario	Diario	Anual	< Anual	Mensual	< Anual	< Anual
Mediana	Diario	Semanal	Mensual	Anual	Semanal	Anual	Semanal

Todos los cuestionarios se cumplimentan en su totalidad, por lo que la suma por columnas es 106 para todos los tipos de partitura.

Los histogramas están en la figura 82, donde las frecuencias de uso más bajas están señaladas con la cifra 0, con el significado de “nunca o menos de una vez al año” y las más altas con el 4, que corresponde a “diariamente”.

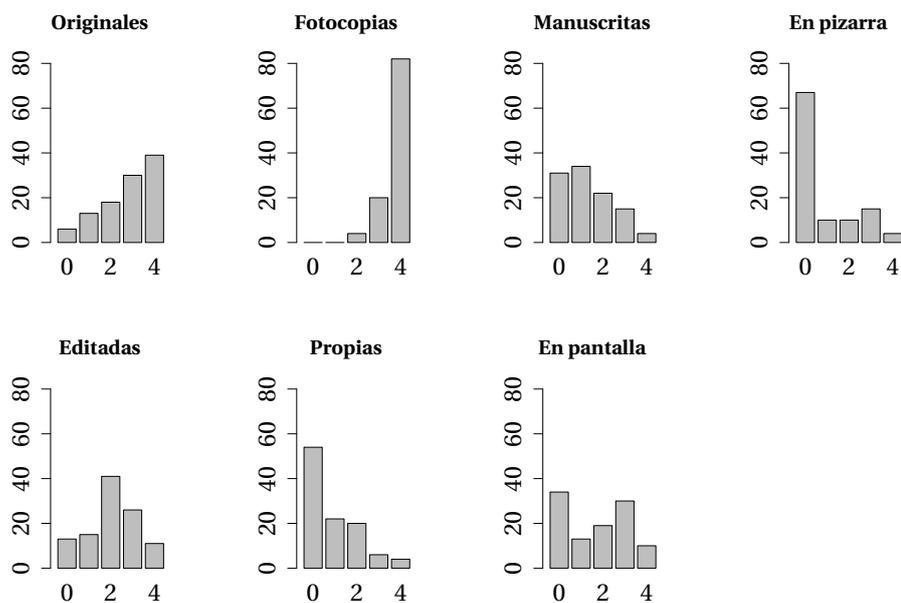


Figura 82: Uso de partituras: histograma de cada respuesta

Atendiendo a las medianas, este resultado se puede resumir diciendo que la frecuencia de uso que el entrevistado tipo declara hacer de las partituras, es la siguiente:

1. Diariamente, (a) partituras originales.
2. Más de una vez a la semana: (b) partituras fotocopias o escaneadas, (g) en pantalla de ordenador, y (e) hechas por otra persona mediante un programa.
3. Mensualmente, (c) manuscritas.
4. Con menor frecuencia, (d) dibujadas en una pizarra, y (f) hechas por sí mismo mediante un programa.

Por orden absoluto, las respuestas más frecuentes son las que aparecen en la tabla 34 después de excluir las respuestas nulas.

Tabla 34: Frecuencias absolutas más elevadas de las respuestas al uso de partituras

Respuesta	Recuento	Proporción
Utilizo partituras fotocopias diariamente	82	77.36 %
Utilizo partituras editadas por ordenador una o más veces al mes	41	38.70 %
Utilizo partituras originales diariamente	39	36.79 %
Utilizo partituras manuscritas entre una vez al año y una vez al mes	34	32.08 %
Utilizo partituras vistas en una pantalla una o más veces a la semana	30	28,30 %

Aunque es la quinta respuesta más repetida, llama la atención que 30 entrevistados utilicen, con una frecuencia semanal, partituras vistas en una pantalla.

Si contabilizamos solamente un uso medio diario o semanal, los tipos de partitura más utilizados aparecen en la tabla 35, que corresponde a la suma de las dos primeras filas de la tabla 33.

Tabla 35: Tipos de partitura más frecuentemente utilizados sobre una base diaria o semanal.

Tipo	Recuento	Proporción
Partitura escaneada o fotocopiada	102	96.23 %
Partitura original, en papel	69	65.09 %
Partitura vista directamente de la pantalla de un ordenador o tablet	40	37.74 %
Partitura editada por algún programa de ordenador	37	34.91 %
Partitura manuscrita, o copia de ella	19	17.92 %
Partitura leída de una pizarra	19	17.92 %
Partitura propia, elaborada con algún programa informático	10	9.43 %

Dada esta frecuencia de uso y después de los dos tipos más utilizados, el tercero son partituras vistas directamente en pantalla, 40 casos que hacen un 37,74 %; una utilización tan frecuente de partitura electrónica no habría sido imaginable hace algunos años.

12.3.2. Asociaciones entre frecuencias de uso

Buscaremos relaciones entre las siete respuestas a la pregunta de la encuesta referida al uso de partituras, utilizando los siguientes enfoques:

- Los estadísticos de asociación χ^2 de Pearson, el coeficiente ϕ , el coeficiente de contingencia y la V de Cramér.
- Modelo loglineal (Agresti, 2007; Christensen, 1997b) y el correspondiente gráfico de asociación de Cohen y Friendly (Cohen, 1980; Friendly, 1992; Meyer y col., 2005)
- El algoritmo *a priori* de Borgelt (Borgelt y Kruse, 2002) para calcular reglas de asociación.
- Un mosaico de teselas en falso color que representan las 49 tablas de contingencia, según Friendly (1994, 1999), Hartigan y Kleiner (1981) y Hofmann y col. (2000).

Las dos respuestas con mayor incidencia en cuanto al uso de partituras son “Uso de material original” y “Uso de partitura escaneada o fotocopiada”. La tabla de contingencia se ve en el cuadro 36; las cifras corresponden a la tesela número 8 (segunda fila, primera por la izquierda) del mosaico de la figura 84. En esta tabla destaca que usan a diario originales y fotocopias 35 personas (un tercio exacto de los participantes), y más de una vez por semana 65 participantes (dos tercios del total). Asimismo, se observa que no existen casos de usos de fotocopia nulos o anuales, aunque sí para los originales, con 19 casos de personas que no las utilizan casi nunca.

Tabla 36: Tabla de contingencia para los usos de partitura más frecuentes.

		Partitura original				
		< Anual o nunca	Anual	Mensual	Semanal	Diaria
Fotocopias	Diaria	1	6	16	24	35
	Semanal	4	5	2	5	4
	Mensual	1	2	0	1	0
	Anual	0	0	0	0	0
	< Anual o nunca	0	0	0	0	0

Estadísticos de asociación

El grado de asociación observado entre los dos tipos de medio utilizados con más frecuencia, que son la partitura original y la escaneada o fotocopiada, se refleja en la tabla 37.

Tabla 37: Estadísticos de asociación entre el uso de los tipos de partitura Original y Fotocopiada.

ϕ	0.506
Coefficiente de contingencia	0.452
V de Cramér	0.358
	χ^2 gl $P(>\chi^2)$
Razón de verosimilitud	24.514 8 0.0019
Pearson	27.151 8 0.0007

gl = grados de libertad.

Para reducir la tabla a unas dimensiones de 2×2 creamos una variable dicotómica que toma un valor unidad tanto para la respuesta “uso diario” como para “uso semanal”, y cero en caso de una frecuencia menor. El resultado de la prueba de dependencia que corresponde según Agresti (2007), Presnell (2000) y Venables y Ripley (1997), es la que aparece en la tabla 38 (véase la página 271), con un resultado marginal, aunque la matriz de residuos tipificados corregidos no muestra ningún valor absoluto fuera de 1.96. Otros pares de respuestas a esta pregunta dan relevancias menores.

Como alternativa a la prueba χ^2 con la corrección de continuidad de Yates está el test exacto de Fisher con todos los datos de la tabla aplicando la función `fisher.test()` de R, que resulta en $p = 0,0004$ lo que indica una alta significación estadística de la dependencia entre estas variables.

El resultado apoya la afirmación previa de que la práctica de fotocopiar partituras está asociada al uso de material original.

Tabla 38: Prueba de dependencia χ^2 de Pearson para la variable dicotómica que indica un uso de partituras frecuente (más de una vez por semana) para los dos tipos más relevantes: originales y fotocopias. gl = Grados de libertad. YCC = Corrección de continuidad de Yates.

		Frec. uso partitura fotocopiada		Total	
		1 o más por semana	<1 por semana		
Frec. uso partitura original	1 o más por semana	Recuento	68	1	69
		Prop. fila	99 %	1 %	
		Prop. col.	67 %	75 %	
		RTC	1.715	-1.715	
	<1 por semana	Recuento	34	3	37
		Prop. fila	92 %	8 %	
		Prop. col.	33 %	25 %	
		RTC	-1.715	1.715	
	Total		102	4	106
			gl	YCC	χ^2
		1	sí	5.27	0.02

Gráfico de asociación

La asociación se analiza con un modelo loglineal (Agresti, 2007; Simkiss y col., 2012; Tabachnick y Fidell, 2007), y permite elaborar un gráfico de asociación de Cohen y Friendly que presenta la desviación entre el valor observado y el valor esperado en caso de independencia y según dicho modelo, para cada elemento de la tabla de contingencia.⁴ Dicha desviación representa la contribución (con signo positivo o negativo) de ese elemento al valor total de χ^2 de Pearson. El rectángulo que corresponde a una casilla tiene una altura proporcional a esta contribución y una anchura proporcional a la raíz cuadrada del valor esperado, de tal manera que el área del rectángulo es proporcional a la diferencia entre el valor observado y el esperado. La línea de base en cada fila indica la independencia de las variables (contribución nula). Si la frecuencia observada es mayor que la esperada, el rectángulo se sitúa por encima de la línea de base, y en este caso aparece en color negro. En caso contrario, aparece por debajo y en color gris.

El gráfico de asociación de la figura 83 presenta, en la columna de las periodicidades diaria, semanal y mensual para el uso de partituras originales, y en la fila de periodicidad diaria para el uso de partituras fotocopiadas, frecuencias mayores de las que cabría esperar si estas variables fueran independientes, lo que sugiere una asociación entre ambas. A su vez, dicha asociación es un indicativo suficiente de que los músicos que son usuarios de material fotocopiado tienden a ser, al mismo tiempo, los que emplean material original.

⁴La desviación es $(O - E)/\sqrt{E}$ donde O es el valor observado y E es el valor esperado.

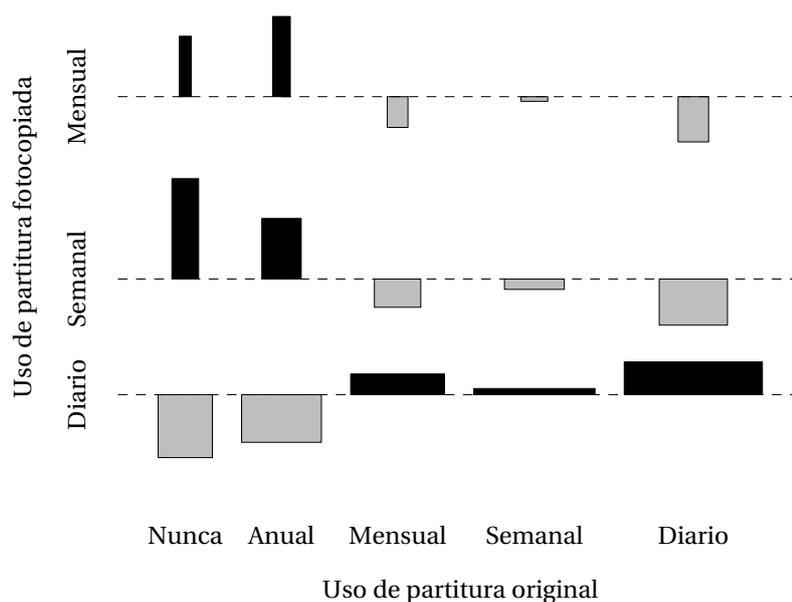


Figura 83: Gráfico de Cohen y Friendly de la asociación entre el uso de partituras originales y fotocopiadas. No aparecen las filas correspondientes a las frecuencias “Anual” y “Menor que anual o nunca” para el uso de fotocopias porque su contingencia es nula; véanse los recuentos en la tabla 36 (pág. 270)

Reglas de asociación

La búsqueda exhaustiva de reglas de asociación es impráctica en general, pero se puede acelerar mediante el algoritmo conocido como “apriori”, así llamado porque se restringe el conjunto de reglas a aquellas que cumplen ciertas observaciones de los datos. Los indicadores de una regla son el apoyo, la confianza y el *lift*, con el siguiente significado:

- El Apoyo es el soporte que recibe la regla según los casos que la cumplen respecto al total.
- La Confianza de la regla crece al aumentar el número de casos que la cumplen respecto a los casos en que se cumple el antecedente.
- *Lift* o elevación: depende de los dos indicadores anteriores. Indica la proporción en que la regla se cumple, comparado con el soporte teórico suponiendo que son independientes. Las mejores reglas tienen el indicador *lift* más elevado y tienen un mayor poder predictivo.

Si la elevación es 1, la regla no predice nada porque se cumple en la misma proporción que si las variables fuesen independientes. Si la elevación es mayor de uno, la regla se cumple con mayor probabilidad de lo que se esperaría si fueran independientes. Si la elevación es menor de la unidad, la regla es negativa: ese conjunto aparece menos veces de lo esperado en caso de haber sido independientes.

Después de restringir las reglas a aquellas que tienen en el término izquierdo o derecho uno de los campos referidos al uso de partituras, y ordenarlas según su valor de *lift*, vemos que las dos primeras tienen un valor mayor de la unidad y son las que aparecen en la tabla 39, donde

- lhs = término izquierdo de la regla o antecedente, contiene una afirmación sobre una combinación de valores de variables.
- rhs = término derecho o consecuente.

Tabla 39: Reglas de asociación halladas para “uso diario de partituras originales” en el término izquierdo.

Nº	lhs		rhs	Apoyo	Confianza	Lift
1	Frec.original = Diaria	⇒	Frec.fotocopiada = Diaria	0.365	0.912	1.157
2	Frec.original = Diaria	⇒	Frec.manuscrita = Nunca	0.271	0.676	1.045

Se lee: “lhs implica rhs”. Las dos reglas encontradas se interpretan como indicativos de que las personas que usan partituras originales diariamente, también usan diariamente partituras fotocopiadas. Al mismo tiempo, recibe un apoyo considerable la regla según la cual las personas que usan originales diariamente no utilizan nunca música manuscrita.

Mosaico de tablas de contingencia

Mediante el empleo de mosaicos, pueden representarse gráficamente las relaciones de asociación entre variables nominales. No es un método numérico, sino visual, y puede ser de utilidad para identificar qué parejas de variables están asociadas. En este ejemplo se utiliza para visualizar la relación entre el uso frecuente de partituras originales y fotocopiadas.

Cada una de las siete preguntas sobre el uso de partituras establecía unas respuestas graduadas de 0 a 4 en función de la frecuencia, desde 0 (nunca o menos de una vez al año) hasta 4 (diariamente). Para encontrar la relación entre estas siete preguntas para los distintos participantes, sería necesario elaborar 49 tablas de contingencia de 5×5 . La relación directa entre pares de variables vendría dada por valores altos en la fila y columna correspondientes a las respuestas de frecuencia alta de uso de partituras.

La técnica del mosaico consiste en traducir los números de las tablas a colores de gris de una intensidad proporcional al valor de la frecuencia correspondiente. El resultado es un cuadrado de $7 \times 7 = 49$ teselas, cada una de las cuales representa la tabla de contingencia para una pareja de variables nominales, que en nuestro caso expresan usos de partitura distintos. Véase la figura 84.

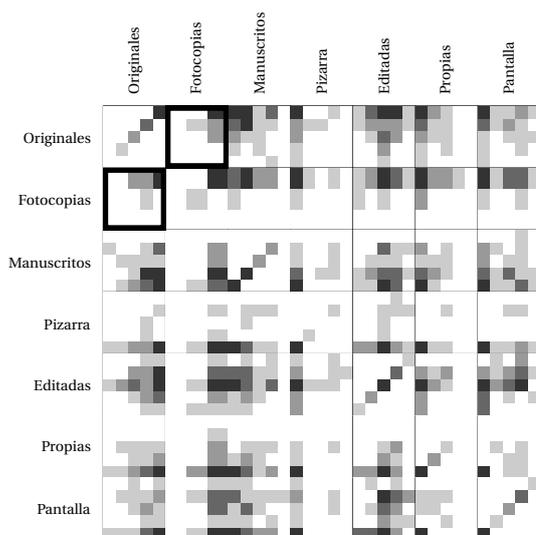


Figura 84: Uso de partituras (pregunta número 7): representación en falso color de las 49 tablas de contingencia posibles para los siete usos de partitura tomados de dos en dos.

Las dos teselas enmarcadas en negro sugieren la existencia de asociación por contener un punto de color oscuro situado arriba y a la derecha. Son irrelevantes las teselas de la diagonal, que expresan las asociaciones triviales de una variable consigo misma.

La asociación encontrada corresponde a las personas que declaran utilizar diariamente tanto partituras originales como fotocopias.

Uso de partituras por actividad declarada

Al distinguir entre las distintas actividades que los participantes declaran ejercer en relación con la música, se obtiene la tabla 40 que muestra un resumen de los medios de notación musical que reciben las mayores frecuencias de uso de partituras, y con mayor proporción dentro de cada actividad. Se muestran por orden de mayor a menor proporción, y no aparecen las frecuencias nulas o que se dan en proporción menor al 30 %.

El dato que destaca en la tabla 40 es el uso de partitura electrónica, que aparece en el grupo de actividad de los aficionados y no en el resto de los grupos. Las partituras electrónicas tienen ventajas e inconvenientes. La mayor ventaja es que no consumen papel, ni apenas espacio físico para el volumen de música que se puede almacenar. Entre los problemas que presentan está que pueden dejar de responder, o hacerlo con más lentitud que el papel.

Tabla 40: Resumen de frecuencia de uso de partituras según la actividad declarada por el participante.

Actividad	Medio	Frecuencia	Proporción
Músico profesional	Partitura fotocopiada o escaneada	Diaria	82 %
	Partitura editada con ordenador	Mensual	42 %
Director de orquesta o coro	Partitura fotocopiada o escaneada	Diaria	88 %
	Partitura original	Diaria	41 %
	Partitura editada con ordenador	Mensual	41 %
Compositor, arreglista	Partitura fotocopiada o escaneada	Diaria	88 %
	Manuscritos	Mensual	50 %
Profesor	Partitura fotocopiada o escaneada	Diaria	84 %
	Partitura original	Diaria	42 %
Estudiante	Partitura fotocopiada o escaneada	Diaria	85 %
	Partitura editada propia	Mensual	38 %
	Partitura original	Diaria	34 %
Aficionado	Partitura fotocopiada o escaneada	Semanal	75 %
	En pantalla de ordenador o tablet	Semanal	50 %

Reglas de asociación para partituras y actividad

El análisis de reglas de asociación que incluyan términos de actividad declarada por el participante y uso de partituras que hace el mismo, ofrece una gran cantidad de reglas con un amplio nivel de soporte. Las dos primeras reglas de entre las más apoyadas por los casos observados (eliminadas las reglas triviales⁵), tienen en el término derecho una expresión que representa el uso diario de partituras fotocopiadas. Las reglas son las que aparecen en la tabla 41.

Tabla 41: Reglas de asociación halladas que implican el uso diario de partituras fotocopiadas.

Nº	lhs	rhs	Apoyo	Confianza	Lift	Lectura
1	Profesional = sí	\implies Frec.fotocopia = Diaria	0.576	0.817	1.036	Los músicos profesionales usan partituras fotocopiadas diariamente.
2	Estudiante = sí	\implies Frec.fotocopia = Diaria	0.541	0.836	1.061	Los estudiantes usan partituras fotocopiadas diariamente.

Otras reglas con un alto valor de confianza no aparecen aquí porque no gozan de un apoyo suficiente al referirse a los colectivos de compositores y directores, que están relativamente poco representados. Es decir: aunque las reglas halladas indican que el uso de fotocopias diario es el más destacado entre todos los participantes, podrían existir otras relaciones menos manifiestas, que los estadísticos no encuentran porque se dan en menor número de casos.

⁵Las reglas triviales son de la forma $\{Compositor = no\} \implies \dots$ y no indican nada porque el término izquierdo, en este caso "El participante no se identifica como compositor o arreglista", tiene una incidencia muy elevada.

Los resultados anteriores sobre consumo de partituras musicales tienen un referente en López-Sintas y col. (2014) que se desarrolla en el ámbito del consumo de música digital. El estudio analiza la manera en que están estructurados socialmente los patrones de acceso a la música, encontrando que son factores de clasificación la edad y la educación, aunque sugiere que un marco de análisis individual es insuficiente para entender los patrones de acceso. Los autores exponen en dicho estudio que los clasificados como “compradores” son el 4,5 %, los que descargan son un 13 % y los copiadore-compradores son un 9 %.

En el mismo contexto de los patrones de consumo de música, Liebowitz y Watt (2006) indican que es difícil saber si el consumo de música en CD y en un concierto son realmente sustitutos o complementos, y proponen entre otras medidas limitar drásticamente el precio de los originales para reducir la copia.

Horas semanales dedicadas al manejo de partituras

Los participantes dedican una media de 23 horas a la semana al manejo directo (lectura, análisis o anotación) de partituras. Solo uno de cada cuatro encuestados dedica menos de 15 horas a la semana; véase la tabla 42.

Tabla 42: Resumen de las horas semanales dedicadas por los participantes al manejo de partituras

Parámetro	Mínimo	1er. cuartil	Mediana	Media	3er. cuartil	Máximo
Horas	1	15	20	23.07	30	60

El máximo de 60 horas semanales declarado por algunos participantes supone cerca de 9 horas al día, por término medio.

Horas en función de la edad

Para el análisis, en función de la edad, del número medio de horas dedicadas por los participantes al manejo de partituras a lo largo de una semana, se ha efectuado un modelo lineal y después se ha hecho una aproximación polinómica.

El modelo líneas se ve en la figura 85, que muestra la relación entre la edad del participante y el número de horas. El coeficiente lineal del modelo es negativo ($-0,21$) con una significación de 0,05. Esto significa que los músicos de más edad tienen a dedicar menos horas semanales al manejo de partituras; sin embargo, el coeficiente de correlación es $\rho = -0,20$, lo que indica que los datos están muy dispersos y la disminución de horas con la edad no se cumple de manera estricta.

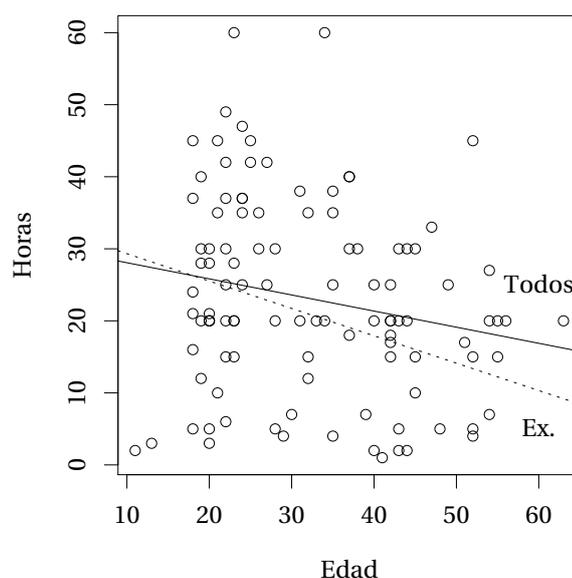


Figura 85: Efecto de la edad sobre las horas dedicadas al manejo de partituras y modelos lineales para el total de los participantes (en línea continua) y para un subgrupo que excluye a los titulados superiores (en línea de puntos, rotulada como “Ex.”).

Al repetir el cálculo del modelo para un subgrupo de las observaciones que excluya a los titulados superiores, se obtiene un mejor ajuste. El coeficiente es $-0,35$ (un descenso más acusado) con $p = 0,03$, y la correlación es algo mejor que en el caso anterior, $\rho = -0,29$. La media de edades en este subgrupo desciende de 32 a 27 años, pero la media de horas no varía. En ambos casos, el centro de los datos se sitúa en un/una participante de unos 30 años que dedica 23 horas a la semana al manejo (lectura, análisis o anotación) de partituras musicales.

Una interpretación de este resultado es que existe un grupo amplio de estudiantes de grado superior de Conservatorio, de alrededor de 22 años de edad, que dedican muchas horas a la semana a tratar directamente con partituras musicales. Los titulados superiores mantienen esta elevada cantidad de horas; en comparación, los participantes no titulados que están dentro del mismo rango de edades que estos titulados superiores, contabilizan un número de horas semanales menor, lo que produce un descenso en la relación horas/edad.

El modelo expuesto de regresión lineal simple no ofrece una indicación de a qué edad los músicos dedican más tiempo al manejo de partituras. Para encontrar este dato, se ha empleado un ajuste polinómico que permite encontrar el máximo. El resultado se muestra en la figura 86. Las dos curvas corresponden a los mismos grupos que se han empleado para el modelo lineal. Se ha utilizado el procedimiento LOESS (Linton, 2008) y se ha optimizado la función para encontrar el máximo. El resultado es similar para los dos grupos y es un/una participante que dedica al manejo de partituras 26 horas a la semana y tiene una edad de 27 años. Es interesante que, en comparación con el promedio de los datos, la edad es menor y el número de horas es mayor.

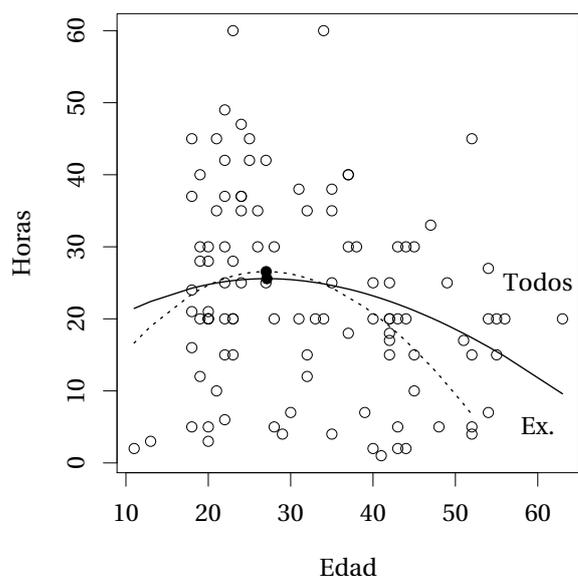


Figura 86: Modelo polinómico para estimar la edad en que se produce el máximo número de horas dedicadas al manejo de partituras, para todos los participantes (en línea continua) y para todos excepto los titulados superiores (en línea de puntos, rotulada como “Ex.”).

En cuanto a la tendencia, se observa que al prescindir de los titulados superiores el declive en el manejo de partituras es más pronunciado al aumentar la edad.

12.4. Uso de software de edición de partituras y formación en el mismo

12.4.1. Uso de software

Se preguntó a cada participante por el programa o programas que utiliza para la producción de partituras musicales, y con qué frecuencia usa cada uno. Los resultados globales aparecen en la tabla 43.

Tabla 43: Uso de software de edición de partituras: recuento global.

Uso	Finale	Sibelius	Encore	MuseScore	LilyPond	Online	Otro
Diario	4	3	1	0	1	0	0
Semanal	16	8	4	0	0	2	4
Mensual	15	5	4	0	1	0	1
Anual	21	18	10	4	7	1	3
Nunca	50	72	87	102	97	103	98

Entre los participantes que declaran haber usado alguna vez al menos uno de los programas (las cuatro primeras filas de la tabla), el programa más utilizado es Finale (52,8 % de los participantes) seguido de Sibelius (32,1 %), Encore (17,9 %), LilyPond (8,5 %), MuseScore (3,8 %)

y aplicaciones online o *apps* de tableta (2,8 %). La respuesta “Otros” acumula un 7,5 %, pero cada opción individual no supera el 2,8 % de Apple Notion, seguido de Page, Noteflight, Cubase, Guitar Pro, Notator y “No recuerda” con un 0,9 % cada uno. La suma supera el 100 % porque muchos participantes utilizan más de un programa.

Un total de 79 participantes (el 70 %) utilizan con una frecuencia no nula algún programa para generar partituras musicales, y el 30 % restante nunca los utilizan. De los 79 que utilizan algún programa, 71 (el 90 %) utilizan Finale o Sibelius; éstos constituyen el 67 % del total. 19 entrevistados manifiestan utilizar los dos programas (24 % de los músicos usuarios de software, 18 % de todos los músicos). 5 usuarios utilizan los tres programas más populares.

Por comparación, en Ribeiro (2010) hay un 77 % de usuarios de Encore.

El programa más usado diariamente por los participantes es Finale (4 casos) seguido de Sibelius, con 3 casos. En tercer lugar están igualados Encore y LilyPond, con un caso cada uno.

Todos los usuarios de programas de edición de partituras declaran haberlos instalado en su sistema, o que alguien se los ha instalado. Dos participantes manifiestan tener instalado un programa y haberlo comprado: un caso de Steinberg Notator y un caso de Notion para iPad. Es decir: de los 75 participantes que utilizan software propietario, solamente 2 (el 2,7 %) declaran haber adquirido el programa. Otro participante afirma tener conocimiento de que su institución educativa ha adquirido e instalado un programa (Finale). Los usuarios de MuseScore y LilyPond no han comprado el programa (ambos son de descarga gratuita).

12.4.2. Formación recibida en software

Ante la pregunta “*¿Has recibido alguna vez algún tipo de formación específica para la edición de partituras con medios informáticos?*”, 37 participantes respondieron afirmativamente (un 35 %), sin una asociación significativa por sexos.

La respuesta abierta a la pregunta de cuál fue el programa para el que recibieron formación, dio un resultado único en 25 ocasiones, otros 9 dieron una respuesta múltiple, y 3 no lo recordaban. Una vez cerradas las respuestas, el resultado es el que aparece en la tabla siguiente, donde la suma de porcentajes es superior al 100 % a causa de las respuestas múltiples. Resumidamente, a partir de los porcentajes de la tabla, se puede destacar lo siguiente en números redondos:

- Uno de cada tres músicos en general ha recibido formación en preparación de partituras con medios informáticos.
- Uno de cada cuatro músicos en general ha recibido alguna formación en el programa Finale.

- Del total de los músicos que han sido formados en cualquier programa de edición de partituras, dos tercios lo ha sido en Finale, y un cuarto en Sibelius.

Véase la tabla 44.

Tabla 44: Formación específica en software de edición de partituras.

Programa	Recuento	Proporción sobre las respuestas afirmativas	Proporción sobre el total
Finale	25	67.60 %	23.6 %
Sibelius	9	24.32 %	8.5 %
LilyPond	4	10.81 %	3.8 %
Encore	2	5.41 %	1.9 %
MuseScore	2	5.41 %	1.9 %
Notator	1	2.70 %	0.9 %
<i>No recuerda</i>	3	8.11 %	2.8 %
Suma	46		

Los estadísticos de asociación no indican ninguna relación entre la formación recibida y el resto de las respuestas, excepto en dos combinaciones, siendo la primera de ellas bastante obvia (por referirse al programa más popular):

- Los músicos que utilizan diariamente el programa Finale han recibido formación para este programa; estos cuatro casos dan un residuo estandarizado de Pearson de 2,2. La asociación entre la formación recibida y el uso de Finale es poco significativa ($\chi^2 = 9,85$; $p = 0,04$, V de Cramér = 0,31).
- 8 de los 9 usuarios de LilyPond han recibido alguna formación sobre edición de partituras (en cualquier programa). El residuo estandarizado de Pearson para estos casos es 2,9; esta asociación entre el uso de LilyPond y el haber recibido formación es muy significativa ($\chi^2 = 16,53$; $p < 0,001$, V de Cramér = 0,40).

12.5. Resultado del experimento comparativo

12.5.1. Recuentos global y medio

La pregunta número 12 presentaba diez casos o parejas de partituras, una original y la otra recreada con LilyPond, a cada participante. Éstos debían mostrar su preferencia por una de ellas en cada caso, o expresar su indiferencia. Durante toda la exposición de resultados se sigue la siguiente nomenclatura: para los diez modelos de partitura, “A” es el recuento de las veces que el participante seleccionó el modelo original, “B” es el de las elecciones del modelo recreado con LilyPond, y “X” es el número de veces que expresó indiferencia. Así, de cada

experimento se obtienen 10 datos: cada uno de ellos es A, B o X, los tres en el rango de 0 a 10. La suma es siempre 10 para cada participante.

Antes de aplicar cualquier baremo, se presenta en la figura 87 muestra el recuento global resultante de sumar los diez datos de cada encuestado, para todos ellos.

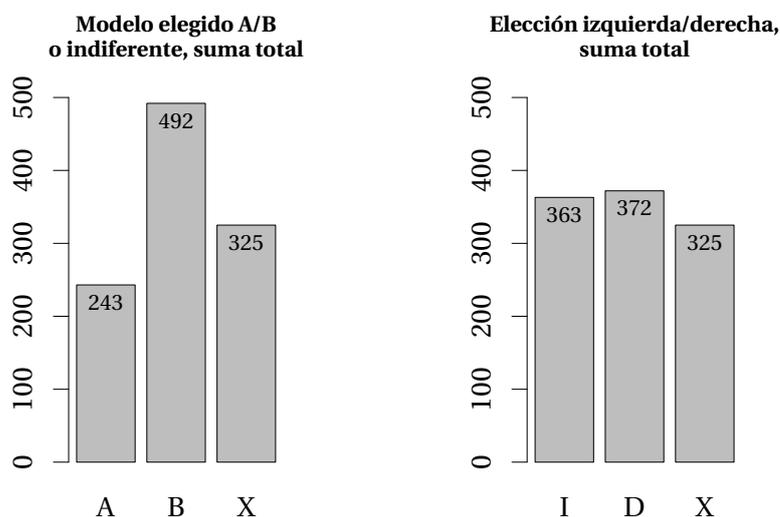


Figura 87: Recuento general del número de veces que los participantes prefirieron las partituras de LilyPond (B) frente a los originales que sirvieron como modelo (A), y número de veces que esta elección se encontraba a la izquierda o a la derecha. Los resultados X son respuestas indiferentes.

El dato X de indiferencia se repite en los dos gráficos porque es inseparable de la respuesta Original/LilyPond o Izquierda/Derecha. La suma en los dos casos es Original + LilyPond + Indiferentes = Izquierda + Derecha + Indiferentes = 1.060, que es el número total de respuestas de los 106 participantes a los 10 modelos del experimento.

Los recuentos medios por participante aparecen en la tabla 45 (véase la página 282). La barra B de la figura 87, que muestra un recuento de más del doble de resultados que la barra A, así como el dato medio para B de 4,64 que aparece en la tabla, frente a 2,30 para A, indican que en promedio para el conjunto de los tipos y las procedencias de las partituras propuestas, los participantes prefirieron las partituras de LilyPond a los modelos de los que estas partituras reelaboradas procedían. Respecto a la parte derecha de la figura 87, el equilibrio en el número total de elecciones izquierda y derecha indica que el proceso de aleatorización se diseñó adecuadamente para eliminar el sesgo que podría producirse por la colocación de los modelos.

Tabla 45: Resultados del experimento: recuento medio por participante.

Respuesta: modelo preferido	Media por participante
A - Partitura original	2.30
B - Partitura recreada con LilyPond	4.64
X - Indiferente	3.06
Total	10.00
Original (excl. control)	2.02
LilyPond (excl. control)	4.43
Indiferente (excl. control)	1.56
Total	8.00
Original + $\frac{1}{2}$ indiferentes	2.80
LilyPond + $\frac{1}{2}$ indiferentes	5.20
Total	8.00
Izquierda	3.43
Derecha	3.51
Izquierda (excl. control)	3.18
Derecha (excl. control)	3.26

El estudio de Purchase y col. (2002) es un antecedente con ciertas similitudes. En este trabajo se examinaron unos diagramas trazados por ordenador siguiendo ciertos algoritmos y se hicieron una serie de medidas de las preferencias estéticas sobre estos modelos por parte de los encuestados. En una serie de seis sub-experimentos, las preferencias por un diagrama sobre otro se recogieron como información cuantitativa; las razones expuestas para justificar la elección se recogieron como información cualitativa.

Entre las diferencias se encuentra la existencia de estas medidas cualitativas, que no hemos efectuado. Es interesante el propósito del estudio, que es deducir reglas estéticas para la producción de mejores diagramas. Sin embargo, no entra en detallar la descripción del método de presentación del material durante los experimentos. Tampoco se ocupa de comparar diagramas existentes o de evaluar el sistema de recreación de dichos diagramas.

12.5.2. Frecuencia de las puntuaciones

Se ha visto en el apartado anterior que en cada experimento se obtienen tres resultados A, B y X que corresponden al recuento de las respuestas “Original”, “Recreado con LilyPond” e “Indiferente”. Cada uno de los valores puede oscilar entre 0 y 8, y cumplen que $A + B + X = 8$.

Se hace un recuento de estos resultados y se clasifican por su frecuencia. Los histogramas resultantes de cada una de estas respuestas para el total de los participantes se muestran en la figura 88.

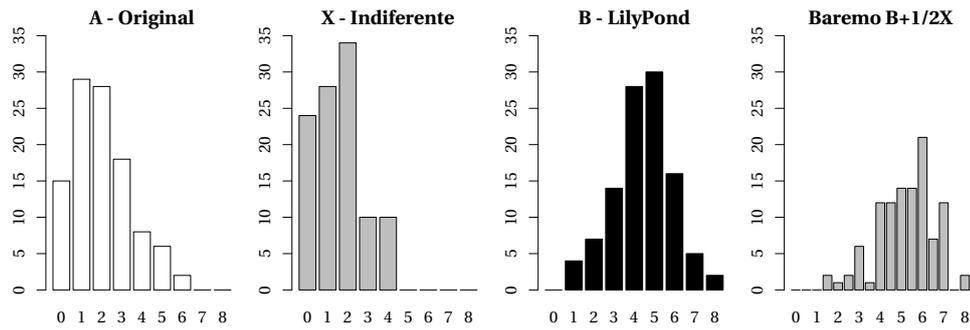


Figura 88: Frecuencias de los resultados del experimento. Recuento del número de veces, dentro del conjunto de los modelos, que los participantes prefirieron el original, se mostraron indiferentes o prefirieron la versión recreada con LilyPond. El último gráfico resulta de añadir al recuento de los resultados de la opción “LilyPond” la mitad de los indiferentes.

Los mismos resultados se muestran en la figura 89 en forma de curvas de densidad, que son histogramas a los que se ha reducido el ancho de banda para suavizar la curva.

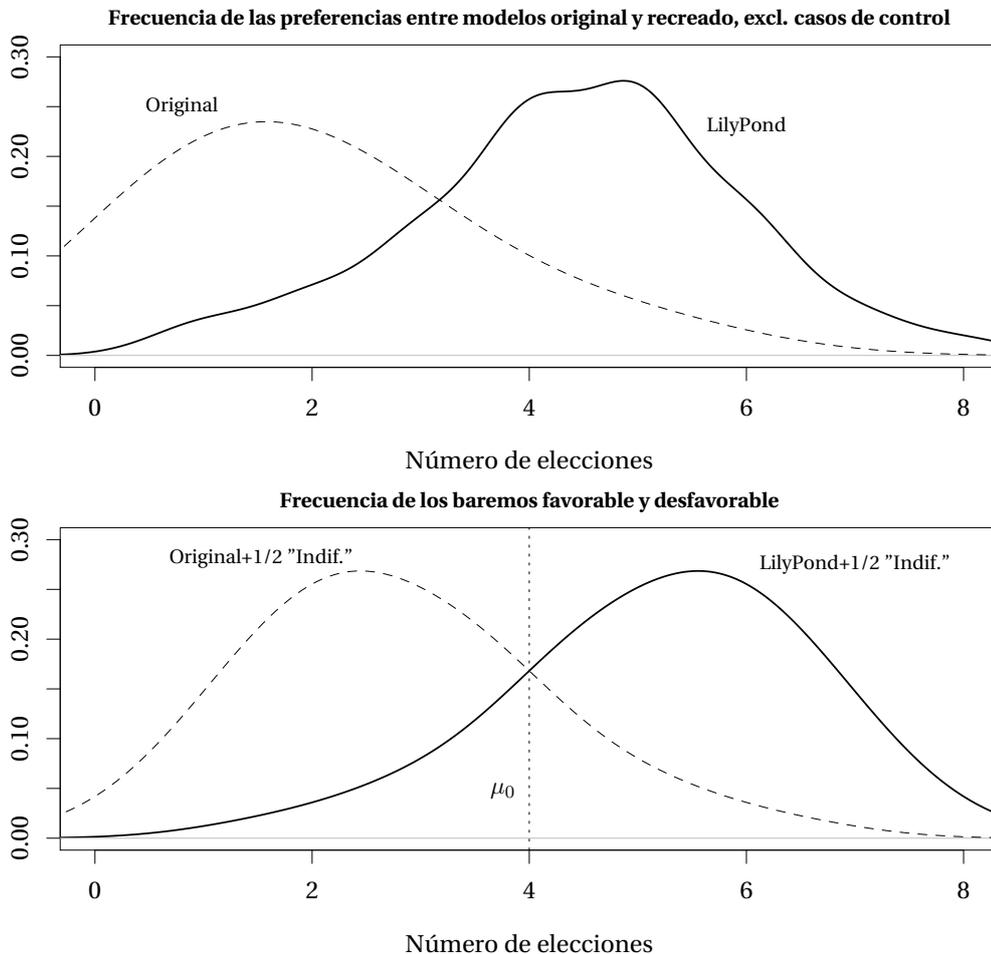


Figura 89: Arriba, curvas de densidad para los recuentos de A (partitura original) y B (partitura de LilyPond). Abajo, curvas de densidad para las métricas “ $A + \frac{1}{2}X$ ” y “ $B + \frac{1}{2}X$ ”. Dentro de cada gráfico, la diferencia entre las curvas muestra que los participantes prefirieron las partituras recreadas con LilyPond a los modelos originales. μ_0 = media teórica del baremo.

Las curvas “Original” y “LilyPond” del gráfico superior, indican la frecuencia de las respuestas sin contabilizar la respuesta indiferente. En el gráfico inferior se representa la curva de densidad del “baremo favorable” que contabiliza las elecciones del modelo de LilyPond más la mitad de las respuestas indiferentes, y se ha añadido, por comparación, la curva del “baremo desfavorable” que añade también la mitad de las respuestas indiferentes al recuento de las elecciones del modelo original.

Las curvas indican que:

- Las puntuaciones que incluyen el recuento de la indiferencia, llevan a la simetría perfecta a las curvas de puntuaciones simples que no tienen en cuenta las respuestas de indiferencia.
- La simetría que presentan, lo es respecto de un eje en el valor “4” de la coordenada “Número de elecciones”, que es la media teórica del baremo.

La simetría se explica porque la suma de los dos baremos es 8 en cada observación.⁶ El contraste formal entre estas distribuciones se calcula en el apartado siguiente, pero las curvas “LilyPond” y “LilyPond + $\frac{1}{2}$ Indif” de la figura 89 situadas apreciablemente más a la derecha que las curvas “Original” y “Original + $\frac{1}{2}$ Indif” indican que los participantes prefirieron, en promedio sobre todos los modelos de partitura, las partituras elaboradas con LilyPond a las partituras modelo originales, apartándose en sentidos opuestos de la media teórica, que es un número de elecciones igual a 4.

12.5.3. Comparación de medias para los baremos Favorable y Desfavorable

Se observa una diferencia entre los recuentos medios de las elecciones A (Original) y B (LilyPond), añadiendo en ambos casos la mitad de las respuestas indiferentes, pero es necesario hacer un análisis de contraste de dos colas porque en función de la varianza, las colas podrían superponerse y producir un falso positivo.

El tamaño de las muestras y sus varianzas son iguales y se trata de una prueba t de Student pareada porque se comparan los valores de los dos baremos (favorable y desfavorable) calculados para cada participante. La prueba se calcula con el programa R mediante la función `t.test()`.

Véase el resultado en la tabla 46, que es la salida del programa R al emplear la función `t.test()`. En la tabla, `gl` = Grados de libertad. `IC` = Intervalo de confianza.

⁶Sin los casos de control, la suma de las dos puntuaciones es $A + \frac{X}{2} + B + \frac{X}{2} = A + B + X = 8$.

Tabla 46: Contraste para el resultado “Preferencia por el original” y “Preferencia por LilyPond” mediante la prueba t de Student.

Prueba t pareada	
t	9.21
gl	105
p	<0.001
IC 99 %	(1.72 , 3.09)
media de las diferencias	2.41

El valor p es cercano a cero: se rechaza la hipótesis nula, “la diferencia entre las medias es cero”, o “las medias son iguales”.

No se rechaza la hipótesis alternativa: “hay una diferencia entre las medias”. Este resultado confirma con un alto nivel de significación la hipótesis de que, en igualdad del resto de los factores visuales y para el conjunto de partituras dado, los músicos consideran que las elaboradas con LilyPond son preferibles a las elaboradas por otros medios, aun cuando en el baremo empleado en el diseño de la prueba los participantes tenían la opción de expresar indiferencia entre ambas.

Si se efectúa la misma prueba para el recuento simple de las elecciones de A y de B, sin tener en cuenta la respuesta indiferente, se obtiene exactamente el mismo resultado de la tabla 46.⁷

12.5.4. Comparación del baremo con la media teórica

Después de comparar entre sí las puntuaciones simples y los baremos que incluyen el factor de indiferencia, con el mismo resultado, utilizamos R para comparar los datos de baremo observados con su media teórica $\mu_0 = 4$. La media teórica del baremo es 4 porque, estando teóricamente distribuidos de manera uniforme, el caso más desfavorable es cero puntos (ningún caso de indiferencia) y el más favorable es 8 puntos si excluimos los casos de control (tampoco hay lugar para los casos de indiferencia en el extremo positivo). Observamos que el resultado es similar (figura 47, pág. 286).

La hipótesis nula es “no hay diferencia, la media es 4”; el valor p es cercano a cero; se rechaza.

No se rechaza la hipótesis alternativa “hay diferencia, la media no es igual a 4”.

⁷Se probó en el apartado de metodología que la puntuación *diferencial* B-A equivale, a efectos del contraste, al baremo favorable que computa la mitad de las indiferencias.

Tabla 47: Contraste para el baremo que incluye el factor de indiferencia, con su media teórica $\mu_0 = 4$, mediante la prueba t de Student.

Prueba t de una muestra	
t	9.21
gl	105
p	<0.001
IC 99 %	(4.86 , 5.55)
media estimada	5.20

La media verdadera estimada es 5,20. La media teórica con un valor de 4 está fuera del intervalo (4,86, 5,55) con el 99 % de confianza. Este resultado supone una evidencia suficiente que demuestra que, en igualdad de condiciones y manteniendo otros factores visuales dentro de unos márgenes de similitud, tomando una serie de modelos de partitura de procedencias diversas y recreándolas con el programa LilyPond, los músicos encuentran preferibles las versiones recreadas, con una significación estadística muy elevada.

12.5.5. Efecto de la dominancia lateral en la elección Izquierda/Derecha

La figura 90 muestra una ligera tendencia del subgrupo de los participantes que se declaran “zurdos” a elegir el modelo de la derecha. La coordenada representada en el eje horizontal es la diferencia entre los recuentos de elecciones de los modelos situados a la derecha y a la izquierda.

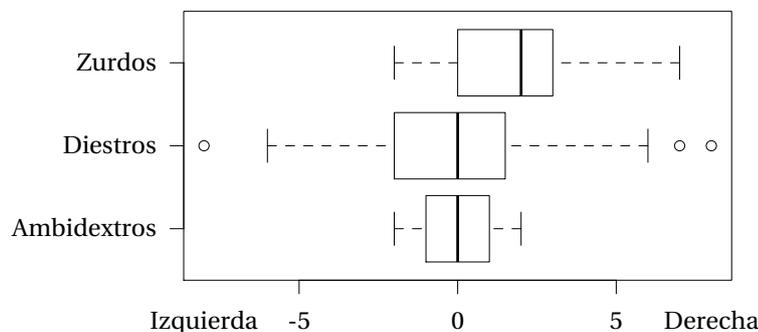


Figura 90: Efecto de la dominancia lateral sobre la elección del modelo de la izquierda o de la derecha. El eje horizontal es la diferencia Derecha-Izquierda.

La media de este valor para los zurdos es 1,78. Esto representa que, en promedio, los participantes zurdos han elegido el modelo de la derecha un número de veces que es 3/4 mayor que si hubieran sido diestros o ambidextros (con un resultado centrado en cero). Sin embargo, la significación de este dato es reducida ($p = 0,09$) porque el subgrupo formado por los 9 zurdos del estudio constituyen solamente el 8,5 % del total de participantes. Como referencia, Borod, Caron y Koff (1984) presentaban, en un estudio centrado en medidas de preferencia sobre 53 zurdos y 93 diestros, un resultado significativo para los zurdos en el sentido de que

estaban menos lateralizados y las medidas eran más dispersas en ellos. Entre las diferencias vemos que este estudio contiene una proporción de zurdos mucho mayor y que se refiere a la preferencia para realizar acciones, en lugar de la predilección visual por modelos situados a un lado u otro. Por su parte, Levy (2002) estudió la preferencia estética en relación a la dominancia lateral y halló un sesgo perceptual en los zurdos al examinar material fuertemente asimétrico y sus imágenes especulares correspondientes, expresando la diferencia como que “los resultados de un grupo de diestros pudieron predecir los de otro grupo de diestros, pero no los del grupo de zurdos”.

Se observa que los estudios que relacionan la dominancia lateral con la preferencia estética suelen emplear material asimétrico como fotografías en las que aparecen personas (Christman y Pinger, 1997). Por la asimetría esencial de la notación musical, en general no tiene sentido emplear imágenes especulares de modelos como si se tratase de una fotografía.

12.5.6. Efecto de la iluminación y la graduación para la vista en la elección A o B

La iluminación de la escena, según se midió para cada experimento, no está asociada claramente a la puntuación en el baremo, aunque algunos resultados muy desviados pueden hacer pensar lo contrario. En función de la iluminación, un solo caso propició un residuo de Pearson de 7,1 por la alta aceptación del modelo original (6,5 sobre 8) con luz muy intensa, pero este resultado tiene un apoyo mínimo en el recuento ($\chi^2 = 216,11$, $p = 0,11$, V de Cramér = 0,41).

La proporción de personas que necesitan graduación se muestra en la figura 91 frente a las puntuaciones del baremo empleado.

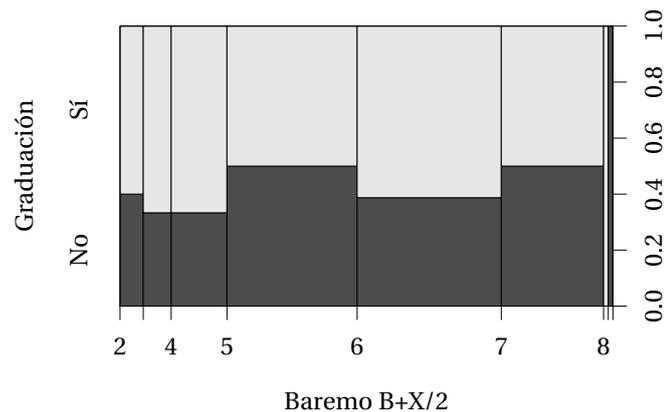


Figura 91: Efecto de la corrección visual: proporciones en función de la puntuación del baremo. La frontera aproximadamente horizontal indica que esta variable no influye en la puntuación.

La parte superior de color claro tiene mayor superficie y corresponde a las personas con graduación en la vista, que son más numerosas; la frontera horizontal entre las dos áreas de color indica que la puntuación en el baremo no está afectada por el hecho de utilizar graduación.

Los estadísticos de asociación para el impacto de la graduación para la vista sobre la puntuación del experimento aparecen en la tabla 48. Aparecen 13 grados de libertad porque las puntuaciones del baremo son una variable numérica discreta, con valores cada medio punto.

Tabla 48: Estadísticos de asociación entre la variable Corrección visual del participante y la predilección por LilyPond. gl = grados de libertad.

ϕ	0.4		
Coefficiente de contingencia	0.371		
V de Cramér	0.4		
	χ^2	gl	$P(>\chi^2)$
Razón de verosimilitud	19.596	13	0.10575
Pearson	16.923	13	0.20284

La significancia es poco intensa y similar a la que presenta el impacto de la iluminación.

Las tablas de contingencia de los resultados A y B por participante y por grupos según necesitaran graduación para la vista o no, se muestran en la figura 92. No se aprecia que los valores más repetidos sean distintos, pero sí se detecta más dispersión en el caso de los participantes que usan gafas o lentillas. En ausencia total de indiferencia, todos los valores con alguna contingencia estarían sobre la diagonal, ya que $A + B = 8$. Los valores que se observan fuera de la diagonal se deben a la indiferencia.

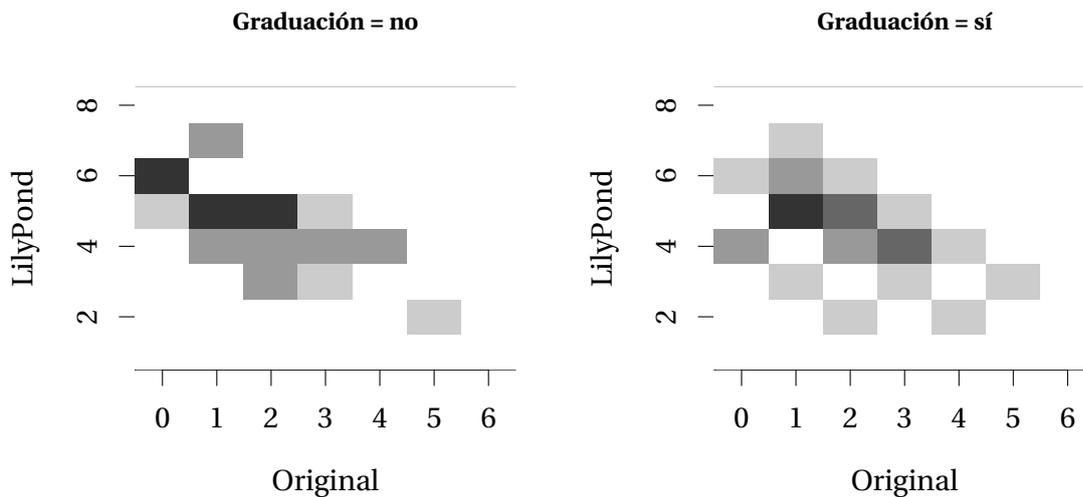


Figura 92: Efecto de la corrección visual sobre la indiferencia: tablas de contingencia para los resultados A (original) y B (LilyPond) con distinción del uso de corrección. El color más oscuro está en proporción a la frecuencia del valor. Todos los valores fuera de la diagonal se deben a la indiferencia.

12.5.7. Impacto del sexo sobre la elección A/B

La relación entre la variable Sexo y el resultado del experimento según la métrica utilizada viene expresada por los estadísticos de la tabla 49. El resultado indica una asociación muy débil entre los dos factores, lo que implica que la predilección por LilyPond no depende del sexo del participante.

Tabla 49: Estadísticos de asociación entre la variable Sexo del participante y la predilección por LilyPond. gl = grados de libertad.

ϕ	0.35		
Coefficiente de contingencia	0.331		
V de Cramér	0.35		
	χ^2	gl	$P(>\chi^2)$
Razón de verosimilitud	15.196	13	0.29525
Pearson	13.016	13	0.44659

En todo caso, la ligera tendencia que se observa en la figura 93 sugiere que a las 53 mujeres les gustan más las partituras de LilyPond que los originales, dado que la proporción de color oscuro que le corresponde en el gráfico es más elevada con las puntuaciones altas del baremo.

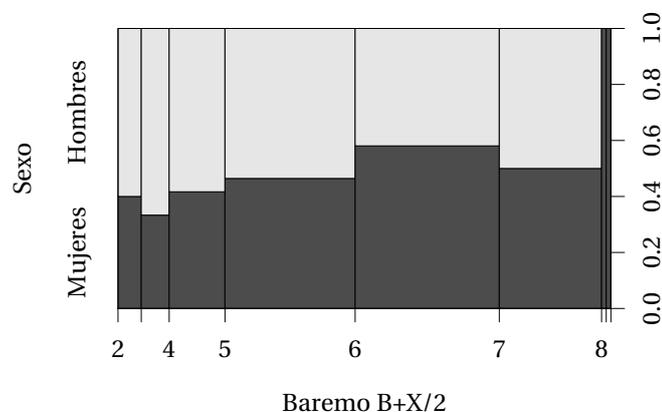


Figura 93: Efecto del sexo de los participantes: proporciones en función de la puntuación del baremo. La frontera ligeramente ascendente conforme se eleva la puntuación, sugiere una tendencia por parte de las mujeres a preferir las partituras de LilyPond.

12.5.8. Efecto de la edad sobre la elección A/B

Al mostrar la elección media de A y B por edades y elaborar un modelo lineal por mínimos cuadrados, se obtiene la imagen 94 que indica cuatro cosas:

- La elección de A no depende de la edad.

- La elección B, excluida la indiferencia, crece ligeramente con la edad.
- La indiferencia disminuye con la edad.
- El baremo que tiene en cuenta la indiferencia compensa las dos últimas tendencias, haciendo que el resultado sea independiente de la edad.

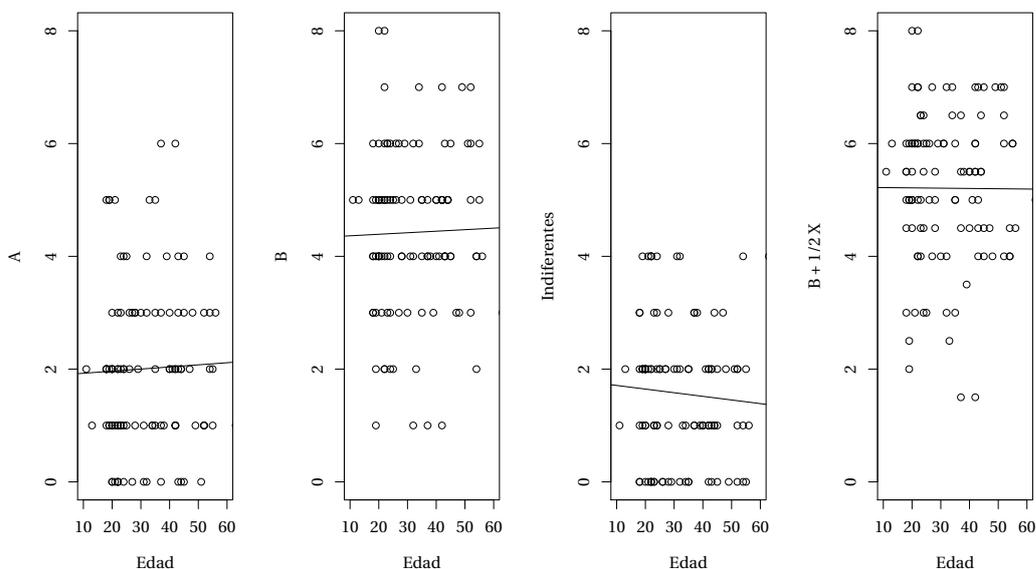


Figura 94: Modelos lineales para las puntuaciones medias por edad de “Original (A)”, “LilyPond (B)”, “Indiferente” y la métrica “ $B + \frac{1}{2}X$ ”.

Los coeficientes de los respectivos modelos aparecen en la tabla 50. Los valores p indican que estas tendencias no son significativas.

Tabla 50: Coeficientes de los modelos lineales para la aceptación de LilyPond en función de la edad.

Modelo	Coef.	p
Original (A) ~ Edad	0.0022	0.860
LilyPond (B) ~ Edad	0.0079	0.518
Indiferencia (X) ~ Edad	-0.0101	0.307
Baremo con indiferencia ($B + \frac{X}{2}$) ~ Edad	0.0029	0.800

12.5.9. Otros factores

La influencia del nivel de estudios completados por los participantes sobre su predilección hacia las partituras recreadas, no es determinante debido al reducido número de casos distintos de las categorías de los niveles profesional y superior, que juntas suman un 87 %. Sin embargo, la gráfica de la izquierda de la figura 95 sugiere un ligero aumento de la aceptación media de los resultados de LilyPond en función directa de los estudios completados. Con

la misma reserva debe hablarse del impacto de la dominancia lateral del participante, que con un 6 % de ambidextros es muy poco determinante aunque la gráfica de la derecha en la figura 95 sugiere una fuerte tendencia de éstos a inclinarse por LilyPond.

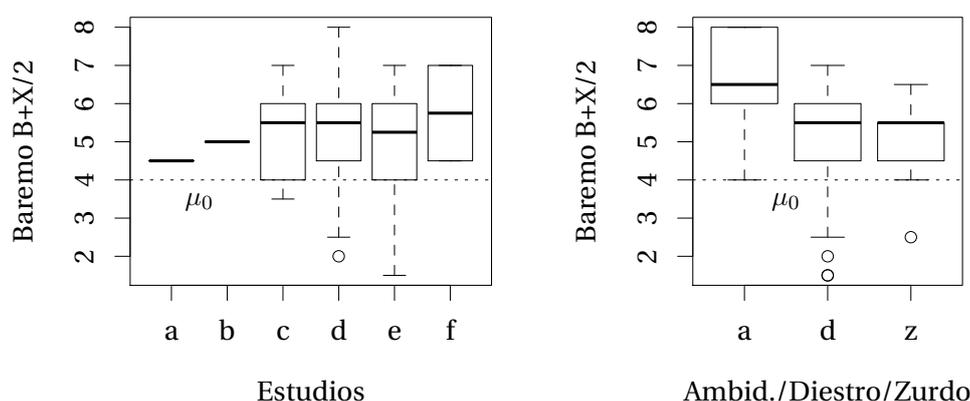


Figura 95: Resultado global de la métrica que contabiliza la preferencia por LilyPond en función de los estudios completados por los participantes (a la izquierda; desde a = sin estudios musicales formales hasta f = doctorado) y de la dominancia lateral (a la derecha; a/d/z = ambidextro/diestro/zurdo) . La media teórica de esta métrica es 4.

12.5.10. Resultado por modelos

La tabla 51 muestra la cuenta global de casos en que los participantes prefirieron cada uno de los modelos. Corresponde a la suma simple de todas las respuestas a la pregunta número 12.

Tabla 51: Recuento del resultado del experimento, por modelos. Las columnas 5 y 10 son los dos casos de control: tipos de partitura Percusión y Orquestal.

Núm. del modelo	1	2	3	4	5(Ctrl)	6	7	8	9	10(Ctrl)
A (Original)	8	24	32	19	13	37	46	15	23	13
B (Recreada)	87	60	36	57	12	62	35	78	34	9
X (Indiferente)	6	17	33	25	76	2	20	8	44	79

La figura 96 presenta los mismos resultados en forma de gráficos de barras.

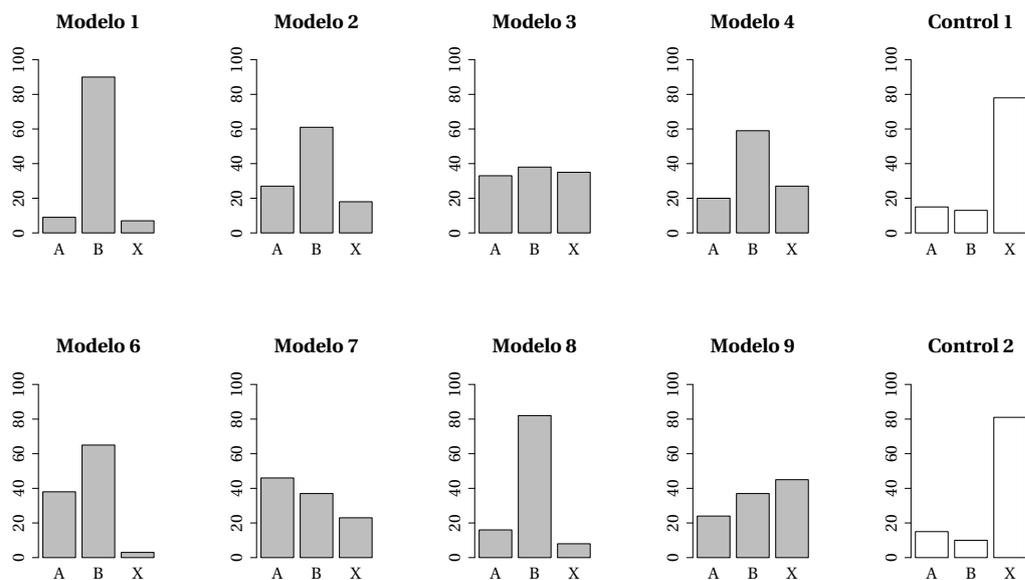


Figura 96: Resultado del experimento por casos: número de participantes que eligieron el modelo original (A), el recreado con LilyPond (B) o mostraron indiferencia (X). Los casos 1, 2, 4, 6 y 8 (violoncello solista, piano, coral, hoja guía de acordes, cuarteto de cuerda) son ganadores; los casos 3 y 9 (*lied* y música de cámara) son dudosos y con alta indiferencia; el caso 7 (partitura de guitarra) es perdedor.

De esta representación visual de los resultados se desprende el siguiente resumen:

- En los casos 1, 2, 4, 6 y 8 los participantes se han decantado claramente por el modelo recreado con LilyPond sobre el original. En el caso número 1 (del tipo “particella”, grupo 1) la proporción es 10 a 1; en el caso 8 (del tipo “cuarteto de cuerda”, grupo 2), LilyPond resulta elegido seis veces más que el original. En otros casos no es tan pronunciada, como en el caso número 6 (del tipo “hoja guía de acordes”, partitura comercial, grupo 2), donde la proporción es 3 a 2 aproximadamente. En estos casos el nivel de indiferencia es bajo.
- En los casos 3 y 9, que son la del tipo “partitura para voz y piano” (el *lied* de Schubert) y la partitura del tipo “música de cámara” (el trío de Debussy), hay un equilibrio relativo entre A y B. Los músicos no prefieren claramente LilyPond y muestran un alto nivel de indiferencia en estos casos. Sumar la mitad de los indiferentes a cada opción no altera este equilibrio.
- En el caso número 7, la partitura del tipo “guitarra” (la *habanera* de Pérez Ribes), se produce la excepción: LilyPond no parece convencer a los músicos con este modelo. Los niveles de preferencia por A o B están relativamente equilibrados pero en posición de pérdida para nuestro ejemplar recreado con LilyPond, en favor del original. Solo un 37 % de los encuestados la prefirieron frente a un 43 % que eligieron el original. La indiferencia es del 20 %. Véase más adelante (apartado 12.5.14) un análisis de los participantes que componen este subgrupo mediante el uso de árboles de regresión.

El número de elecciones izquierda/derecha está equilibrado para todos los modelos, lo que prueba la efectividad del procedimiento de aleatorización seguido.

12.5.11. Resultado por modelos, según sexo

El gráfico correspondiente a la predilección por cada modelo en función del sexo del participante, puede verse en la figura 97. Los gráficos de mosaico están contemplados por Emerson (1998) y Huh (2004).

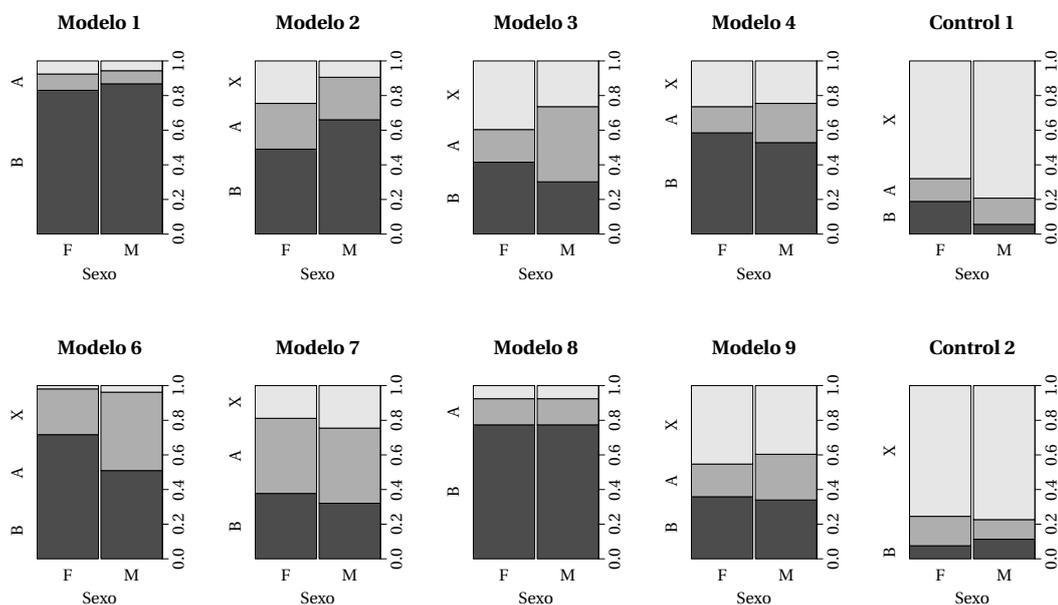


Figura 97: Aceptación de cada modelo en función del sexo de los participantes. Proporción de personas (en el eje vertical) que eligieron cada modelo de LilyPond (B), en color oscuro, y cada modelo original (A), en gris medio, para cada sexo. El color más claro indica la proporción de indiferentes.

En función del sexo, solo los modelos números 2 y 6 muestran alguna pequeña diferencia. Al parecer son más los participantes de sexo masculino que prefieren LilyPond en la partitura para piano (caso 2), por 35 a 26, mientras que en el caso de la canción con acordes y posiciones de guitarra (caso 6), las participantes de sexo femenino que prefieren LilyPond son más numerosas, 38 frente a 27. En el primer caso los hombres indiferentes son menos, por 5 a 13; en cambio en el segundo caso hay solamente tres indiferentes. Concluimos entonces que la prevalencia de B entre los hombres se hace a costa de una disminución de su indiferencia ante los ejemplares del modelo 2. En cambio, el modelo 6 no deja indiferente a nadie y la prevalencia de las mujeres que eligen LilyPond es, por tanto, más clara.

El modelo que presenta un resultado más equilibrado en cuanto al sexo, quizá el único capaz

de poner de acuerdo a hombres y mujeres, es el número 8 (cuarteto de cuerda), curiosamente el más laborioso y también uno de los que más preferencia muestran por la partitura de LilyPond.

12.5.12. Resultado por modelos, según el uso de graduación para la vista y la iluminación de la escena.

No hay diferencias significativas en los niveles de aceptación de las partituras de uno u otro tipo en función de si el participante necesita corrección visual para el manejo de partituras, ni de acuerdo al nivel de iluminación de la escena del experimento según se midió para cada encuesta realizada.

12.5.13. Resultado por modelos, según estudios

En la figura 98 pueden verse, en el eje vertical, las proporciones en que los participantes que habían completado cada uno de los niveles de estudios considerados, eligieron los modelos hechos con LilyPond (B) o los originales (A).

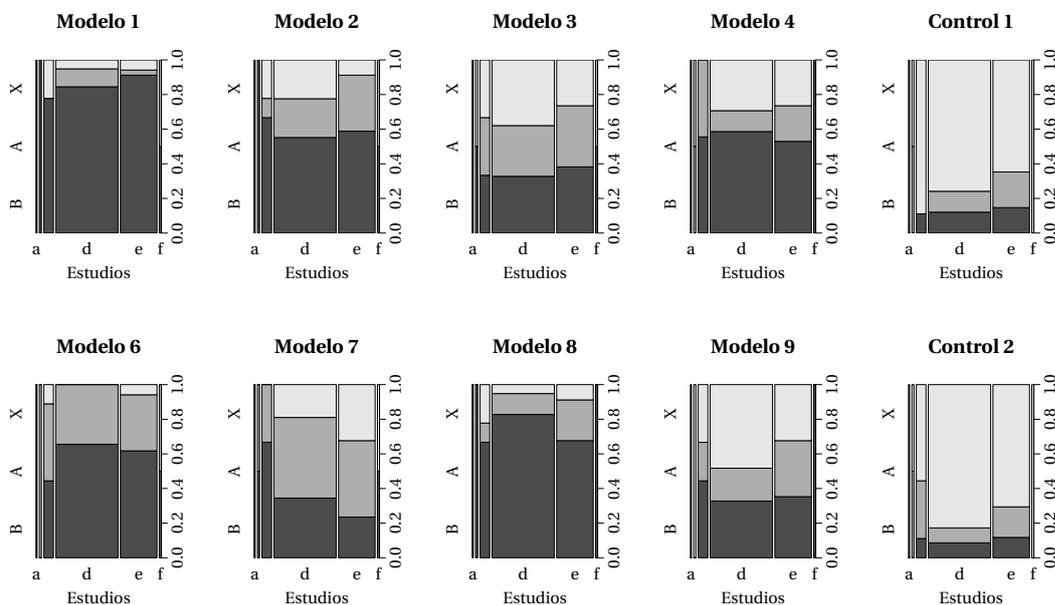


Figura 98: Aceptación de cada modelo en función de los estudios completados por el participante. Proporción de personas (en el eje vertical) que eligieron cada modelo de LilyPond (B), en color oscuro, y cada modelo original (A), en gris medio, para las seis categorías de estudios completados *a a f*. El color más claro indica la proporción de indiferentes.

Solamente los apartados *c*, *d* y *e* están representados suficientemente como para mostrar al-

gún tipo de influencia. Corresponden a los estudios elemental, medio y superior. Quedan en minoría y poco apreciables los colectivos sin estudios formales o con estudios no reglados, y el de personas que han superado el nivel de doctorado.

Las observaciones dignas de mención que pueden hacerse a partir de este análisis son las siguientes:

- Para los modelos 2, 4 y 9, que son la pieza pianística, la coral y la de música de cámara, la proporción de titulados superiores que prefieren el modelo original sobre el de LilyPond es mayor que la de estudiantes de grado superior, a costa de un menor grado de indiferencia.
- En el modelo 7 (partitura para guitarra), la aceptación de los ejemplares de LilyPond desciende al aumentar el nivel de estudios, y aumenta el grado de indiferencia.

En los modelos 1, 3, 6 y 8 no hay una diferencia marcada en función de los estudios.

Los modelos de control muestran poca distinción según los estudios del participante, pero obviamente el nivel de indiferencia es mucho más elevado que en el resto de los modelos. Hay una ligera tendencia a la disminución de la indiferencia (es decir, a decantarse por uno de los modelos) a pesar de que los modelos eran iguales, al considerar el grupo *e* de personas con estudios superiores. Esto tiene probablemente una explicación psicológica que escapa al objetivo de este estudio.

12.5.14. Uso de árboles de regresión para la clasificación por modelos

Calle y Sánchez-Espigares (2007), citando a Therneau, Atkinson y col. (2014), utilizan árboles de regresión con 146 casos⁸ en el programa R. En nuestro estudio lo utilizamos con $n=106$ para obtener una indicación aproximada de la combinación de factores que condicionan los resultados, eliminando el mayor número posible de variables predictoras.

El algoritmo utilizado para el desarrollo del árbol es `rpart` de R (Therneau, Atkinson y col., 2014), que es una implementación de las mismas ideas que el algoritmo C&RT (o “CART”) de AnswerTree para SPSS. El procedimiento incorpora estimación del riesgo (porcentaje de casos mal clasificados) mediante validación cruzada. Sólo se debe especificar la validación cruzada para conjuntos de datos pequeños, según Rojo (2006). La razón es que para muestras grandes, una validación de este tipo es prohibitiva en términos computacionales.

⁸Berlanga y col. (2013) recomiendan el uso de árboles de regresión para muestras no inferiores a 1.000 casos en un artículo sobre el uso de SPSS.

Rechazo del modelo 7

El modelo de partitura de guitarra resultó desfavorecido en los resultados. Aunque no podemos saber por qué, ya que no disponemos de datos que reflejen los motivos de los encuestados, sí podemos clasificar a los participantes en grupos que den una respuesta media próxima a la que se considera, y examinar los valores de las variables que definen a estos grupos.

Si se toma una variable objetivo (como la puntuación del baremo $B + \frac{X}{2}$) y una serie de variables nominales que son susceptibles de explicar dicho objetivo, el empleo de árboles de regresión es una técnica de la llamada “minería de datos” que puede ser de utilidad. El árbol explora mediante ANOVA la variabilidad del objetivo en función de los factores, y separa las observaciones en dos grupos cada vez, sucesivamente, que presenten medias diferenciadas. El criterio de división de cada rama está expresado en el punto de ramificación. Puede verse el caso considerado en la figura 99 en que se ha fijado como objetivo una expresión que toma el valor verdadero cuando el participante elige el modelo original sobre el recreado, en el caso número 7. Podemos entonces averiguar a qué grupo de participantes no hemos logrado convencer con la partitura de guitarra recreada con LilyPond.

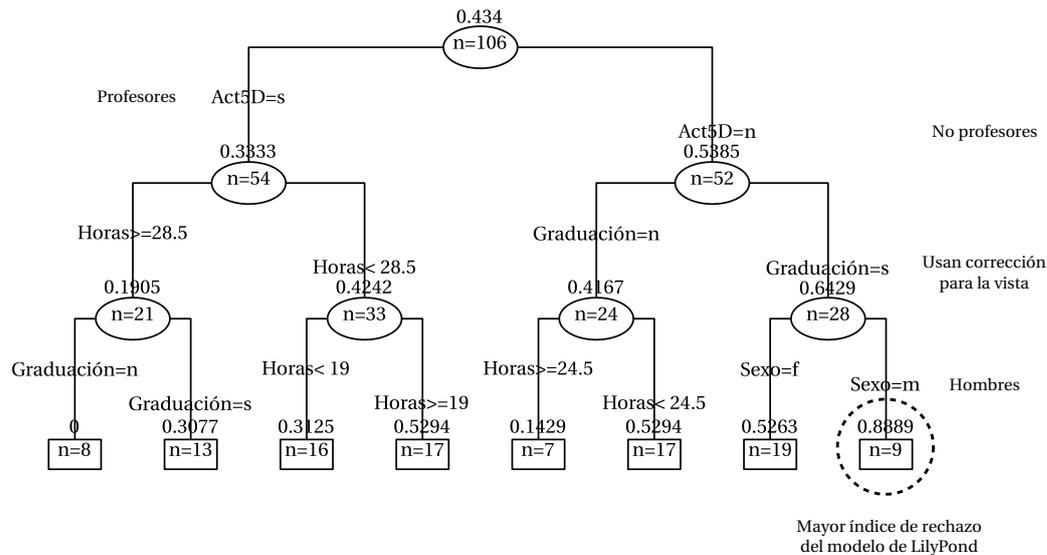


Figura 99: Árbol de regresión para el resultado desfavorable obtenido en el caso 7, partitura de guitarra. En este caso, el grupo de 9 participantes que aparece en el extremo derecho está formado principalmente por los que eligieron el modelo original en el caso de la partitura de guitarra.

Para conseguir variables nominales con mayor representatividad en la muestra a partir del conjunto de variables nominales existentes, donde cada uno de ellos tenía una representatividad reducida, se ha creado una variable agrupada que acumula los niveles de estudios Superior y Doctorado frente a otros niveles de formación, y otra para el uso de partituras Diario y Semanal (en cada tipo de partitura) frente a las frecuencias menores.

Al contrastar la respuesta objetivo contra este conjunto de variables nominales, el resultado

es que aparece un grupo de 9 participantes con una alta incidencia en la respuesta “A” para el modelo 7. La media del grupo para el valor de esta respuesta es 0,889, cercano a la unidad, lo que significa que este grupo rechazó mayoritariamente el modelo de LilyPond en la partitura de guitarra.

Los valores de las variables relevantes para este grupo, que definen a los participantes que lo componen y que han propiciado la ramificación, son: Actividad musical distinta de “Profesor”, Necesita graduación para la vista, Sexo masculino. El grupo está destacado en la representación del árbol de la figura 99.

Es deseable un modelo lo más sencillo posible. Para podar el árbol de forma óptima, evitando el sobreajuste y minimizando el riesgo (el porcentaje de casos mal clasificados), se ha empleado la regla 1-SE, que solamente se puede usar para árboles pequeños como el nuestro.

Para aplicar la regla 1-SE se tiene en cuenta que en cada ramificación disminuye el parámetro de complejidad, pero aumenta el error por validación cruzada. Se confecciona una tabla del error por validación cruzada frente al parámetro de complejidad, con expresión del error estándar para cada árbol. El resultado es la tabla 52.

Tabla 52: Tabla de errores por validación cruzada para el árbol de regresión correspondiente al rechazo del modelo 7.

Árbol	P. de complejidad	Núm. de ramific.	Error rel.	Error por val. cruzada	Error estándar
1	0.043	0	1.000	1.010	0.026
2	0.028	1	0.957	1.326	0.081
3	0.027	4	0.873	1.391	0.095
4	0.018	5	0.846	1.415	0.101
5	0.015	6	0.828	1.431	0.104
6	0.010	7	0.813	1.447	0.104

Cualquier riesgo que se encuentre dentro de un error estándar relativo de una unidad a partir del mínimo que se ha conseguido, se considera equivalente al mínimo. En este caso, todos excepto el primero lo superan. A continuación se elige el árbol más sencillo que supera este margen. En la tabla anterior se selecciona el árbol 2, que tiene una sola ramificación.

Según el modelo simplificado que se obtiene de la poda, puede decirse que la propiedad más determinante para que un participante elija (o no) LilyPond en el modelo de partitura de guitarra, es que sea un profesor, en cuyo caso pertenece a un grupo de 54 personas que presenta una media de 0,333, o que no sea profesor, en cuyo caso el grupo tiene 52 miembros con una media de 0,539 en la variable lógica que expresa la preferencia por el modelo original frente al de LilyPond. Este resultado indica que la baja puntuación en la partitura de guitarra hecha con LilyPond se debe a la contribución de los no profesores, que tienden a elegir el modelo original en su lugar.

El árbol resultante después de la poda aparece en la figura 100.

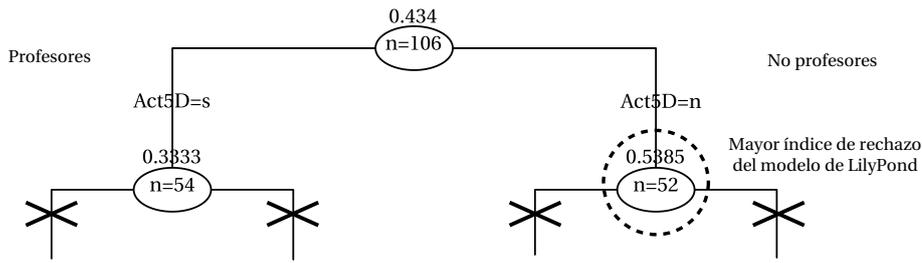


Figura 100: Árbol de regresión para el rechazo del modelo 7 (partitura de guitarra), después de la poda del árbol de la figura 99.

Indiferencia

El árbol de regresión obtenido para la variable objetivo «número de indiferencias, excluidos los casos de control», se presenta en la figura 101.

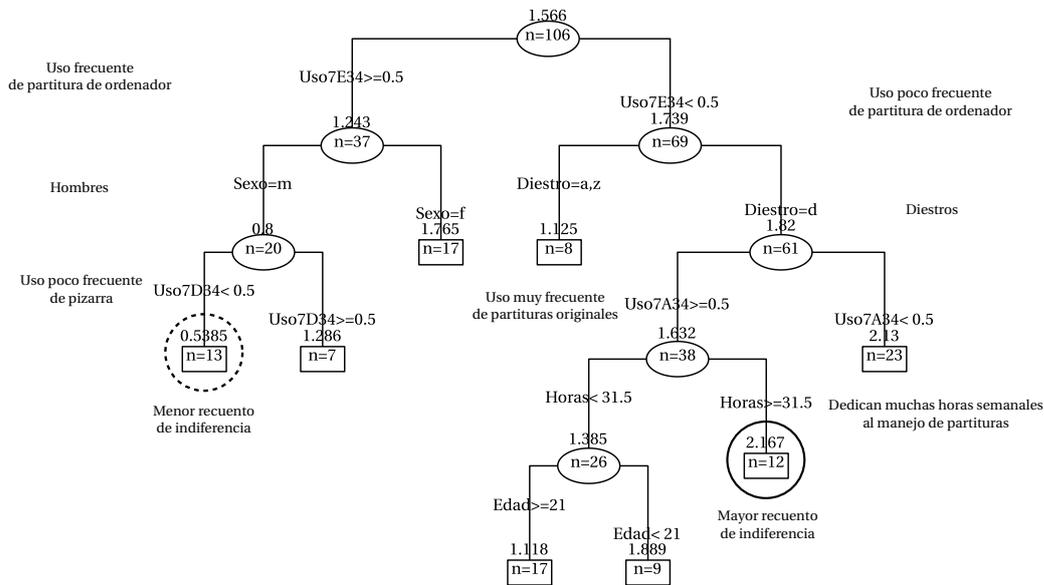


Figura 101: Árbol de regresión para la indiferencia. El grupo destacado en línea discontinua presenta la menor media, 0,539, y el destacado en línea continua presenta la mayor media con 2,167, en el recuento de las veces que los participantes respondieron con indiferencia.

Este árbol da los mayores valores en un grupo de 12 participantes, todos ellos diestros, que usan partituras editadas mediante ordenador con menor frecuencia que la media, y partituras originales en papel con mayor frecuencia que la media, y que dedican semanalmente más de 31,5 horas al manejo de partituras musicales. La cantidad media de respuestas indiferentes para de este grupo es 2,167, frente a la media global observada que es de 1,566.

La menor indiferencia, con una media de 0,539, la presenta un grupo de 13 participantes, todos ellos hombres, con elevada frecuencia de uso de partituras de ordenador y reducida frecuencia de uso de música manuscrita sobre pizarra. Los dos grupos mencionados se en-

cuentran destacados mediante una línea continua y una línea de puntos, respectivamente.

La poda óptima mantiene las dos primeras ramificaciones; los errores de árbol son los que aparecen en la tabla 53.

Tabla 53: Tabla de errores por validación cruzada para el árbol de regresión de la indiferencia.

Árbol	P. de complejidad	Núm. de ramific.	Error rel.	Error por val. cruzada	Error estándar
1	0.047	0	1.000	1.012	0.117
2	0.026	2	0.906	1.109	0.137
3	0.023	5	0.828	1.223	0.144
4	0.017	6	0.805	1.234	0.146
5	0.010	7	0.789	1.232	0.145

Según esta tabla, la poda se efectúa para el parámetro de complejidad 0,026 del árbol 2, que tiene dos ramificaciones.

El árbol resultante de aplicar la poda, que corresponde al modelo simplificado, aparece en la figura 102.

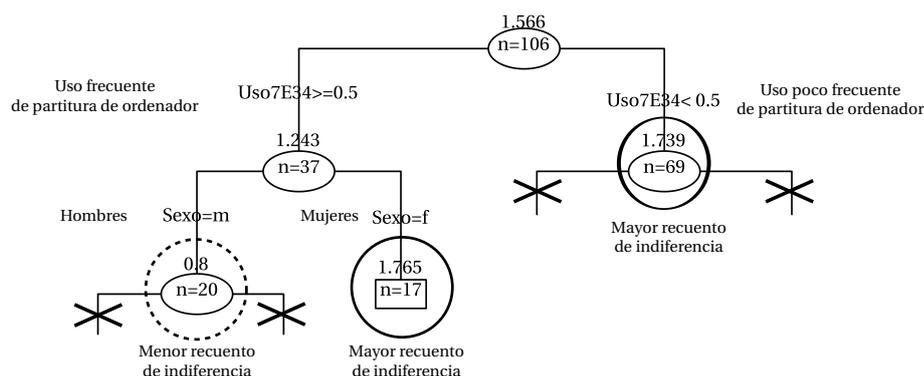


Figura 102: Árbol de regresión para el índice de indiferencia, tras la poda del árbol de la figura 101.

El modelo resultante de simplificar el árbol mediante el procedimiento de poda óptima, expresa que los menos indiferentes son 20 hombres que pertenecen a un grupo que presenta un uso muy frecuente de partituras de ordenador, y los participantes más indiferentes son 17 mujeres del mismo grupo anterior, unidos a 69 participantes de ambos sexos que presentan un uso de partituras de ordenador reducido. La comparación de las medias en el término de indiferencia, que son de 0,8 y 1,8, indica que la indiferencia entre los participantes del segundo grupo es el doble que entre los del primero.

12.6. Observaciones

Al margen de los resultados cuantitativos, pueden ofrecer interés para la repetibilidad ciertos casos particulares dentro del desarrollo de la encuesta.

Es interesante, por ejemplo, que algunos voluntarios con mucha experiencia declaraban tratar de evitar a toda costa la respuesta “indiferente”, lo que les llevó a preferir un ejemplar determinado de las parejas de control, a pesar de que eran iguales.

No se ha hecho un registro de este extremo, pero se detectó una tendencia de los participantes a olvidar que en cada uno de los casos disponen de la opción “indiferente” como respuesta, cuando ninguno de los dos modelos del caso es claramente preferible para ellos. En dos ocasiones no se produjo ninguna respuesta indiferente y el investigador preguntó al final si el participante había tenido presente que podría haber dicho “indiferente” en cada uno de los diez casos, y en las dos ocasiones el participante manifestó no haberlo tenido presente. En lugar de eliminar a estos participantes, quedaron registradas solamente las respuestas de las preguntas 1 a 11 y se registraron dos semanas más tarde los resultados del experimento (pregunta 12) durante la fase de retest.

En el extremo opuesto está el caso de un encuestado que expresó su indiferencia por cualquiera de los modelos, en todos los casos. Otro participante con estudios elementales manifestó indiferencia en los primeros casos, hasta que comenzó a detenerse largo tiempo en cada uno. Al terminar la prueba, se le presentaron de nuevo los primeros casos y manifestó inmediatamente su preferencia clara por un ejemplar en cada uno de ellos. Esto indica que en pocos minutos se puede entrenar la vista para reconocer detalles de la notación musical y encontrar pequeñas diferencias a las que habitualmente no se presta atención.

Mostramos a continuación algunas de las observaciones realizadas por los participantes en siete de los diez casos propuestos. Son interesantes porque pueden reflejar ciertos detalles de las partituras que se pasaron por alto durante la preparación del material.

En el caso número 1, que es una parte de orquesta para violoncello, un participante muy experimentado manifestó su disgusto por los dos ejemplares, que debido a su gran tamaño consideraba solo apto “para niños”. Varios participantes observaron que el original A presentaba algún fallo de distribución, como compases muy anchos con una sola figura musical y otros en cambio relativamente estrechos pese a contener cuatro o más figuras. En el sentido opuesto, hubo quien indicó que un compás de larga duración siempre requiere una anchura mayor. Debe recordarse que en la selección de modelos para el material se incluyeron expresamente partituras de todo tipo.

El caso número 2 (sonata para piano de Dussek) dio lugar a algún comentario negativo sobre el modelo original, A, por la notación excesivamente apretada en pasajes de corcheas que tienen los compases algo más estrechos de lo normal. La distribución de la música sobre las líneas, así como el número total de sistemas, es un factor que durante la preparación del material se dejó tal y como el programa LilyPond había determinado automáticamente, sobre seis sistemas, frente a cinco del original. En defensa del modelo original hay que decir que la distribución de la música, aunque tiene una apariencia algo forzada, responde exactamente

a la de la edición original de la partitura de Dussek de la que está extraída.

El modelo recreado con LilyPond del caso 3, el *lied* de Schubert, fue criticado en más de una ocasión porque faltaba el guión separador de las sílabas del texto, allí donde las notas correspondientes están muy juntas. Es una funcionalidad del programa que no es del agrado de todo el mundo. El caso 3 es de los que ofrecieron un resultado más incierto, pues los participantes se repartieron en tres grupos casi exactos en cuanto a sus respuestas, como se vio en la exposición del recuento de los datos; en este caso, la recreación produjo una partitura muy similar al original.

La pieza coral de F. Guerrero, caso número 4, suscitó el comentario de un participante con miopía que prefería las notas “más grandes” de la recreación de LilyPond. Sin embargo, el tamaño de los pentagramas es idéntico al del original. Esto indica que la tipografía “Feta”⁹ es capaz de introducir una cabeza de nota *aparentemente* más grande en el mismo espacio de pentagrama. La figura 103 presenta lado a lado el detalle de las tipografías de Finale y de LilyPond. Martin (1998) trata el tema del diseño de las cabezas de nota para aprovechar el espacio disponible. Smithers (2000), en “Diseñar un tipo para probar una teoría” estudia el impacto del diseño de una fuente tipográfica sobre el mensaje que transmite.

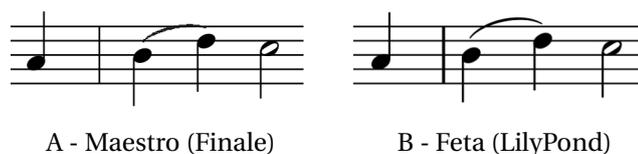


Figura 103: Comparación del tamaño aparente de las cabezas de nota para el mismo tamaño de pentagrama en el caso 4 del experimento, la partitura coral *Ibant Magi* de F. Guerrero. A la izquierda, el tipo Maestro de Finale; a la derecha, el tipo Feta de LilyPond.

Lo interesante de este caso es la forma en que los participantes manifiestan una apreciación clara a partir de un factor que tiene un fundamento óptico muy sutil, como la forma en que el óvalo de las figuras cubre el grosor de las líneas del pentagrama. El motivo de incluir la reseña de este caso en las observaciones de los resultados, y no en la metodología de la confección del modelo 4, es que no se detectó el fenómeno descrito hasta que los participantes lo comentaron durante la encuesta y cuando la fase piloto ya había concluido.

El caso 6 tiene diagramas de posiciones de acordes (que algunos participantes llamaban “tablatura”). Los diagramas de LilyPond fueron explícitamente criticados por sus líneas demasiado delgadas.

Los modelos A y B del caso 7 (pieza para guitarra) fueron elogiados y criticados de forma alternativa por presentar cada uno de ellos ventajas e inconvenientes. En el original se observó una inclinación de las barras más horizontal y menos “angulosa”; en el modelo recreado se

⁹Las fuentes tipográficas de LilyPond llevan nombres de quesos: *Feta*, *Parmesan*, *Emmentaler*.

apreciaron positivamente las indicaciones del número de cuerda encerradas en un círculo y las digitaciones en un tipo más grueso.

El modelo B del caso 8 (cuarteto de Beethoven) recibió varias observaciones en el sentido de que los sistemas estaban muy juntos entre sí. La regla teórica establece que la separación entre sistemas debe ser siempre mayor que entre los pentagramas dentro del mismo sistema. En este caso no había ningún margen de maniobra debido a la gran densidad de la notación, pero es cierto que ciertos elementos que pertenecen a un sistema, por un estrecho margen casi llegan a colisionar con el sistema siguiente.

13. Estudio 2. Conclusiones

13.1. Conclusiones respecto de los objetivos

LilyPond es un programa de producción de partituras musicales desarrollado por un equipo de voluntarios en su tiempo libre, pero con altas aspiraciones. Cuando se planteó un estudio objetivo del grado de aceptación relativo que tendrían las partituras elaboradas con un programa de este tipo entre las personas que se dedican a la música, todas las posibilidades estaban abiertas.

Se ha diseñado, validado y puesto en práctica un instrumento para la medida del grado de desempeño de un programa editor de partituras, en cuanto a la calidad de sus resultados en el campo de la notación musical dentro de un contexto educativo: todas las partituras empleadas como modelos estaban en circulación en el ámbito de la enseñanza como material de estudio.

El experimento propuesto fue validado para garantizar la imparcialidad de las pruebas, y el coeficiente de validez V de Aiken superó el umbral prefijado para los 19 jueces y para todas las preguntas, por lo que el proceso de validación mediante un panel de jueces, cuantificado numéricamente, ha sido satisfactorio. Los procedimientos de anonimización y aleatorización empleados en el experimento han mostrado una alta fiabilidad y adecuación a los objetivos marcados.

Asimismo, el instrumento presentó estabilidad en el tiempo comprobada mediante test / re-test. Se cumple con todo ello el objetivo de *diseñar y validar un instrumento que permita objetivar este grado de aceptación.*

La mitad de los 106 músicos (principalmente profesionales, profesores y estudiantes) entrevistados dedica habitualmente entre 15 y 30 horas a la semana al manejo, lectura, análisis o anotación de partituras musicales.

La información obtenida a resultas del experimento apoya la teoría previamente no contrastada de que las partituras de LilyPond son, en general, visualmente preferibles a las de otros sistemas, y que son perfectamente válidas para trabajos de distintos niveles de complejidad.

Se cumple también el objetivo de *averiguar qué formación específica han recibido los músicos en edición de partituras.* Los resultados muestran que uno de cada tres músicos ha recibido

alguna formación en esta materia, que de ellos dos tercios lo fueron al menos en Finale y que un cuarto lo fue al menos en Sibelius. Uno de cada cuatro músicos en general está formado en el programa Finale. Los usuarios de LilyPond estaban formados en su gran mayoría.

El objetivo de *averiguar qué frecuencia de uso de las partituras hacen los músicos, y diferenciar entre los principales soportes como libros, fotocopias, música manuscrita y partituras vistas en una pantalla* se cumple a través de los resultados de la encuesta, que indican que casi 9 de cada 10 músicos emplea partituras musicales todos los días, y que el 100 % lo hace una o más veces a la semana. Por tipos de partitura, los más utilizados son fotocopias de originales (3/4 de ellos diariamente, la práctica totalidad diaria o semanalmente), seguido de los propios originales (más de 1/3 diariamente, 2/3 diaria o semanalmente). Se ha detectado una asociación estadísticamente significativa entre el hecho de utilizar cada uno de estos dos medios como fuente de notación musical.

Se efectuó el cálculo estadístico de la asociación entre los factores sociodemográficos de los participantes y su preferencia por las partituras de LilyPond. De esta forma se da por cumplido el objetivo de *establecer qué asociación existe entre la preferencia por las partituras elaboradas con LilyPond, sobre otros sistemas, y los factores sociodemográficos (relacionados con la música o no) de la población*. Sin embargo, los resultados permiten negar de modo general la existencia de tal asociación o, dicho de otro modo, la preferencia por estas partituras no depende de ningún factor conocido asociado a las características del entrevistado.

Siete de cada diez músicos usan programas informáticos para elaborar partituras musicales. Atendiendo solamente a estos usuarios, en el proceso encaminado a cumplir los objetivos secundarios de *investigar qué herramientas están utilizando los músicos para satisfacer la necesidad de elaborar partituras musicales* y el de *encontrar los programas de edición de partituras más utilizados*, se ha encontrado que más de la mitad de ellos usan Finale, un tercio usan Sibelius, 9 de cada 10 usan Finale o Sibelius, y casi 2 de cada 10 usan ambos. El abanico de preguntas cerradas pudo cubrir casi todos los programas; algunos participantes utilizaron, con menor frecuencia, seis casos de programas que se recogieron en el apartado “otros”, cinco de ellos empleados por una sola persona cada uno.

En cuanto al origen de los programas que los músicos utilizan para elaborar partituras, se ha detectado un pequeño porcentaje de usuarios de software privativo que lo han adquirido. Se cumple de esta forma el objetivo de *determinar cuál es este origen*.

Desde el punto de vista práctico, no limitado a la música, se ha comprobado que es posible cuantificar el grado de idoneidad de una herramienta de software que produce un resultado gráfico, a partir de un proceso de comparación simple y globalizado de dicho resultado con el de otros programas.

El objetivo de *dar una respuesta a la cuestión de si el software libre de edición de partituras*

está a la altura de las necesidades de los músicos y/o de la industria de la producción de música impresa está cumplido, pues LilyPond es un ejemplo de software libre en este campo y los resultados muestran que la tarea mencionada tiene en este programa una opción que puede medirse con las exigencias de los músicos y los profesionales, o superarlas.

Todo ello indica que el objetivo principal: *Averiguar qué grado de aceptación tienen, entre los músicos, las partituras musicales hechas con LilyPond, en comparación con los originales que han servido de modelo para éstas*, se cumple con la investigación que motiva esta Memoria.

13.2. Conclusiones respecto de las hipótesis

Hipótesis 1: *Los músicos emplean partituras musicales diariamente en el ejercicio de su actividad habitual, y acumulan un elevado número de horas semanales dedicado al manejo de éstas.*

Se confirma la hipótesis para el 90 % de los músicos según el resultado de la encuesta. Si se atiende a un período semanal, se cumple que todos ellos utilizan partituras. La idea principal que se puede transmitir a partir de este resultado es que la partitura musical es un material que continuamente se emplea en dicha actividad, y por tanto en su enseñanza, algo que es obvio para los músicos pero cuyo respaldo carecía hasta el momento de objetividad.

Respecto al número de horas semanales dedicados a esta actividad, como se ha visto en la exposición de resultados, es un dato muy disperso, en ocasiones muy elevado, pero con amplias variaciones de una persona a otra. La muestra incluía un importante número de estudiantes, que presentan una media elevada en cuanto al tiempo de uso de partituras. Entre los músicos de más edad, los titulados superiores y los profesionales, se ha detectado un ligero descenso en términos generales. El recuento medio permite dar una cifra de alrededor de 23 horas a la semana en el manejo de partituras por parte de un músico.

Hipótesis 2: *Los músicos, en su actividad ordinaria, emplean partituras fotocopiadas o escaneadas con una frecuencia que iguala o supera al de partituras originales.*

La segunda hipótesis se confirma, pues la fotocopia es el medio más prevalente utilizado por los músicos como fuente de material de notación, para todos los tipos de músico de la muestra, con treinta puntos porcentuales de diferencia media respecto al material original, el segundo en frecuencia. La diferencia alcanza los cuarenta puntos porcentuales en los músicos profesionales, y los cincuenta puntos porcentuales en los estudiantes.

La partitura electrónica es utilizada frecuentemente por un tercio de los participantes, y por uno de cada dos del grupo de los aficionados.

Hipótesis 3: *En los músicos, existe asociación entre la frecuencia de uso de partituras musicales originales y la de partituras fotocopiadas o escaneadas.*

Se confirma la hipótesis con un alto nivel de significación estadística. El resultado se puede expresar diciendo que las personas que usan frecuentemente material fotocopiado tienden a ser las mismas que usan, asimismo frecuentemente, partituras originales.

Hipótesis 4: *La actividad de producir partituras mediante algún programa informático es común entre los músicos.*

Se ha encontrado que siete de cada diez músicos producen partituras con medios informáticos, usando mayoritariamente tres programas de edición privados muy populares, aunque hemos contado doce programas distintos que los participantes efectivamente utilizan.

Se confirma la hipótesis número cuatro, y junto a los resultados anteriores, la idea de que la partitura musical está presente continuamente en la práctica musical del docente, el estudiante y el profesional, en sus dos vertientes como consumidores y productores, parece confirmarse sólidamente.

Hipótesis 5: *En igualdad de otros factores visuales, si se toma una partitura musical existente y se elabora de nuevo mediante LilyPond, los músicos la encuentran preferible o más atractiva.*

El experimento confirma parcialmente la quinta hipótesis. Para su preparación se tuvo la precaución de “ecualizar” o igualar los factores visuales que pudieran hacer uno de los ejemplares preferible al otro por causas ajenas al propio mérito de la notación musical. El cuestionario y el material empleados fueron adecuadamente validados, con un resultado satisfactorio, por encima de un umbral conservador.

Por un lado, los resultados generales muestran, con un alto grado de significación estadística, que las partituras elaboradas con el programa LilyPond a partir de una serie de modelos eran preferibles a los propios modelos.

Por otro lado, el resultado puede depender del tipo de partitura. Si se distingue entre los resultados obtenidos por cada uno de los modelos, se observa un resultado negativo en que la versión recreada no recibió el mismo grado de preferencia que el resto. El nivel de significación estadística fue relativamente bajo y el resultado fue positivo en los siete casos restantes. La conclusión lógica es que la medida de la preferencia en este tipo de pruebas comparativas es sensible al tipo de material: la herramienta puede ofrecer un mayor nivel de competencia dependiendo de la situación, dado el amplio abanico de tipos de notación musical existente.

Hipótesis 6: *El grado de preferencia por el resultado visual de LilyPond por parte de un músico no está determinado por su nivel de estudios o por otras variables sociodemográficas.*

Se registraron una serie de factores descriptivos de cada participante, que permitieron elaborar subgrupos y así tratar de explicar los resultados. Los factores “Edad”, “Sexo”, “Ocupación”, “Graduación para la vista”, “Nivel de estudios”, “Iluminación de la escena”, y “Dominancia lateral” no han supuesto ninguna influencia clara en el resultado. Se confirma así la sexta hipótesis.

En los casos en que se aprecia alguna desviación, el bajo número de casos impide sacar conclusiones; por ejemplo: los ambidextros obtienen, en el baremo que mide la predilección por LilyPond, un punto más (sobre 8) que los diestros y los zurdos, pero no alcanzan la proporción de 6 de cada cien.

Hipótesis principal: *Las partituras musicales elaboradas con la herramienta de tipografía musical de software libre “LilyPond” presentan un nivel de desempeño comparable al de otros sistemas privativos de edición musical en una selección razonablemente amplia de situaciones reales dentro de los terrenos educativo y profesional.*

Los resultados generales del experimento confirman con un nivel de significación alto la hipótesis principal en la que se afirmaba que el desempeño de una solución libre al problema de la tipografía se podría comparar con éxito a las soluciones más populares, que son privativas. Dicha afirmación se apoya en las confirmaciones de las cuatro primeras hipótesis, que muestran que el consumo y elaboración de partituras es muy común entre los músicos, y de las hipótesis números 5 y 6 que miden el grado de desempeño del programa LilyPond respecto a la calidad de su notación musical, evaluada por los músicos sin distinción por los factores examinados.

El contraste del resultado del experimento con su media teórica ha dado, con un nivel de significación superior al 99 %, una preferencia de los participantes por los modelos recreados con el programa frente a los originales sobre los que se hicieron las recreaciones.

Debe apuntarse que los ejemplares usados como modelo procedían tanto de la práctica docente como de ediciones profesionales. LilyPond ha hecho un trabajo satisfactorio al emular los distintos casos, aunque el investigador no es un profesional de la edición de partituras. Teniendo en cuenta que este programa es Software Libre, es difícil no recomendar su uso basándose, al menos, en los resultados.

La conclusión es que, a la hora de producir partituras musicales como material para la enseñanza o para uso profesional, cuando se opone LilyPond a otras soluciones de uso común, algunas de ellas privativas (es decir: no libres), el software libre puede medirse con éstos y superar una prueba ciega de aceptación por parte de un amplio rango de músicos, ya sean profesionales de la interpretación, profesores o estudiantes.

13.3. Limitaciones y prospectiva

Los modelos de partitura se han seleccionado según una serie de criterios debidamente documentados. Sin embargo, no se ha tenido en cuenta el criterio de simultaneidad temporal. Los modelos de partitura utilizados como referencia no se han sometido a ninguna restricción previa en cuanto a la fecha de su producción, mientras que las recreaciones usadas para ser sometidas a comparación se han elaborado con la última versión del programa LilyPond en el momento de la confección del material.

Una de las partituras de referencia, probablemente aquella que presenta un nivel técnico más alto, está publicada en el s.XIX y pertenece a la tradición clásica del grabado manual sobre plancha de metal. Hay que decir que el diseño de las formas tipográficas de la fuente predeterminada que LilyPond utiliza, tiende precisamente a imitar las formas de las partituras clásicas grabadas a mano. Por contra, ciertas ediciones más recientes emplean un tipo más moderno, mientras que las versiones recreadas se hicieron siempre con la fuente predeterminada denominada “Feta”, lo que ha sido una restricción autoimpuesta para sistematizar el proceso.

El experimento se ha efectuado para todos los participantes de forma presencial e individualizada, lo que limita el tamaño del conjunto de respuestas, aunque el peso del cuestionario se ha mantenido intencionadamente breve buscando un equilibrio que permitiese acceder a un número mayor de participantes. Entre otras opciones para facilitar el acceso a una muestra mayor están la encuesta *online* y el envío del material al domicilio del encuestado. A su vez, la primera variante del instrumento que se ha mencionado tendría limitaciones: el experimento *online* implicaría comparar dos imágenes de música sobre la baja resolución de una pantalla de ordenador. Es posible que en este caso no se estuviera evaluando correctamente el resultado visual del programa, que en principio está orientado a la producción de notación musical impresa. Con la encuesta no personal no habría sido posible controlar las condiciones.

Con frecuencia, cuando los participantes han manifestado su preferencia en el transcurso del experimento, han añadido comentarios que hubiera sido interesante recoger, pero no estaba previsto un análisis de este tipo. Aunque los resultados son confirmatorios de la hipótesis principal y la investigación ha seguido un enfoque cuantitativo, queda por descubrir cuál o cuáles son las propiedades que los participantes aprecian en los modelos que eligieron. La pregunta solicitaba solamente que el encuestado se decantara por uno de los modelos según una mezcla de criterios: cuál le parecía *más hermosa*, o *mejor edición*, o una notación *más atractiva visualmente*, obligándole de alguna manera a ponderar todos los criterios y a ofrecer una respuesta en la que la balanza se inclinara hacia uno de los lados, sin más matizaciones que la opción de manifestar su indiferencia. Con una encuesta más elaborada hubiera sido posible desglosar los criterios e ir preguntando por todos ellos, uno a uno, ofreciendo al entrevistado a motivar de alguna manera sus elecciones. Algunos de los comentarios más

interesantes vienen de personas con lentillas o gafas que afirman preferir la partitura de Lily-Pond “porque es más grande”, aunque se tomaron todas las precauciones para que el tamaño del pentagrama fuese igual en los dos ejemplares de cada caso y todos los modelos superaron la validación. Quizá se pueda interpretar este comentario en un sentido más sutil. Es posible que lo que algunos músicos perciban como una música impresa más “grande” sea en realidad una notación con mayor peso visual, quizá debido al mayor diámetro horizontal de la cabeza de las figuras, o al grosor de las líneas del pentagrama y las plicas de las notas.

Otros comentarios iban orientados a justificar una elección sobre otra a pesar de que los dos modelos tenían “ventajas e inconvenientes”. Lo que nos recuerda este comentario es que las partituras musicales son objetos visuales complejos en los que interviene un gran número de factores.

El material empleado para el experimento se puede mejorar. En todos los casos, los modelos A y B eran impresiones láser sobre hojas DIN A4, con los mismos márgenes. Pero los músicos comentan algo en lo que parece haber un cierto consenso: las partituras fotocopiadas (o impresiones de láser) son muy inferiores a los originales comerciales. El contraste visual de las fotocopias es excesivo: una impresión demasiado negra sobre un papel demasiado blanco. El color crema del papel de las ediciones comerciales es preferible. Respecto al tamaño, los márgenes de los libros de música (p.ej. de piano) son más grandes, normalmente 2 cm. en el interior y 2,5 cm. en el borde externo, con una anchura de línea de 19 cm., pero estos mismos márgenes en una hoja DIN A4 dejan poco espacio para la música, algo más de 16 cm. de ancho. Si se imprime la música al mismo tamaño, los márgenes prácticamente desaparecen. Por tanto, *el tamaño DIN A4 es pequeño para la música*. Un estudio posterior deberá tener todo esto en cuenta. Sin embargo, ignoramos cuál sería el coste de una tirada única en offset sobre papel crema en tamaño 235 × 310 mm. Una posibilidad intermedia sería imprimir con láser en el centro de hojas DIN A3 y guillotinar después por las marcas de corte.

La medición de la luz en cada encuesta es tediosa, pero no vemos una manera mejor de dejar constancia de que las condiciones de iluminación fueron las necesarias y suficientes. Un experimento en condiciones completamente controladas es posible, aunque ello reduciría aún más el número de participantes.

Respecto a las diferencias observadas por tipo de modelo, pueden tomarse dos clases de medidas:

- Por un lado, estas diferencias justifican la conveniencia de ofrecer un abanico de tipos de partitura distintos en una experiencia de esta clase, en la que participan músicos con distintas especialidades que están acostumbrados al manejo de música impresa con diferentes características. En otras palabras, emplear un material más diverso que se adapte a músicos de trasfondo más dispar, para distintos instrumentos o para distintos niveles en el campo de la educación.

- Por otro lado, esta especialización de los músicos en un cierto tipo de partitura sugiere otro criterio para la selección del material, más adaptado a la población diana. Esto supondría restringir la especialidad de los músicos participantes y ofrecerles solamente material de su propia especialidad. El único grupo con un número suficiente de miembros serían los pianistas, a quienes se les mostrarían partituras de piano.

Pensamos que aún sería posible un enfoque mixto, que consistiría en la preparación de materiales específicos para instrumentos diversos, y utilizando con cada participante solamente el material de su tipo correspondiente. El objeto es no restringir aún más el tamaño de la muestra.

La limitación autoimpuesta a diez casos o tipos diferentes de partitura, parece estar en el límite de la cantidad que puede presentarse a un voluntario sin llegar a importunarlo en exceso con una larga entrevista, que además requiere un esfuerzo de concentración.

Para mejorar el grado de aprovechamiento de los diez casos del experimento sin aumentar el tiempo dedicado a la encuesta, se puede considerar eliminar uno o ambos casos de control y sustituirlos por casos reales con su recreación correspondiente, que entrarían en el baremo junto a los ocho restantes. La razón es que sobre los tipos de partitura correspondientes a los casos de control empleados, no podemos afirmar nada en el sentido de la hipótesis principal, dado lo cual sería preferible emplearlos como casos normales. La justificación se fundamentaría en dos resultados de nuestro estudio: el equilibrio entre izquierda y derecha en las respuestas, y la independencia de la dominancia lateral del participante, que combinados hacen que se reduzca el interés en este tipo de empleo de los casos de control.

Para evaluar otro aspecto del desempeño de LilyPond en comparación con otros programas, sería muy interesante alterar el punto de vista del participante como *consumidor* de partituras musicales en favor de otro como *productor*, para lo que habría que seleccionar cuatro conjuntos de voluntarios: dos grupos diferenciados, compuestos por aprendices y expertos, respectivamente, en un determinado programa de amplia difusión como Finale o Sibelius; otros dos grupos estarían compuestos por usuarios de LilyPond (también aprendices y expertos por separado). A los cuatro grupos se les ofrecería realizar, parcial o totalmente, una misma serie de partituras de diversos grados de complejidad a partir de unos modelos existentes. Dado un término fijo de tiempo (como media hora), se medirían los fallos cometidos y la cantidad de música producida. La principal diferencia en el modo de manejo entre ambos programas es que LilyPond está orientado a documentos de texto, mientras que la mayoría de los programas de edición de partituras tienen una interfaz gráfica. El mismo tipo de comparación se produce entre un procesador de textos como Word (que tiene interfaz gráfica) y un sistema de preparación de documentos como \LaTeX (que es un sistema por lotes orientado a código). A este respecto, hay un experimento similar reciente (Knauff y Nejasmic, 2014) en el que se ha comparado Word con \LaTeX , con resultados muy interesantes: el rendimiento de-

pende mucho del tipo de documento. Word resultó ser más eficiente para la producción de documentos sencillos y \LaTeX para los más complejos como libros con muchas fórmulas matemáticas, bibliografía y referencias cruzadas. Después de medir objetivamente la eficiencia en la composición de textos, los voluntarios cumplieron un cuestionario de *usabilidad* para medir el cansancio, la frustración y aspectos emocionales como la satisfacción en la tarea. Sin embargo, no se consideró el aspecto estético, que en cambio centra nuestro estudio sobre LilyPond. Lo que este estudio tiene en común con el presente trabajo es que, dependiendo del tipo de material, los resultados pueden modificarse o incluso invertirse.

Otro estudio (Brischoux y Legagneux, 2009), esta vez cuantitativo, preguntó a los editores de varias publicaciones científicas la prevalencia de cada formato en los manuscritos recibidos; \LaTeX domina totalmente en las áreas de Matemáticas (97 %), Estadística (89 %) y Física (74 %), seguidos por los grupos de Informática (46 %) y Astronomía/Astrofísica (35 %), con el resto de las áreas completamente dominadas por Microsoft Word. Por su parte, quizá el futuro de LilyPond esté relacionado con su idoneidad para ciertos tipos de partitura cuya naturaleza está aún por determinar.

Los trabajos de Percival (Percival, 2008; Percival, Anders y Tzanetakis, 2008, 2007) sobre generación de material para ejercicios con Strasheela (Anders y col., 2005) podrían aplicarse a la producción de ejemplos con vistas a un experimento comparativo que no dependiera de partituras existentes. Strasheela es un sistema de generación de fragmentos musicales basado en reglas de restricción. Estas reglas bien pueden estar fundamentadas en la práctica común, es decir, pueden provenir del análisis de partituras reales. Otro trabajo que ofrece una perspectiva interesante para la generación de música y que podría utilizarse para producir experimentos comparativos es el de Pestana y Pestana (2012).

13.4. Repetibilidad

El material empleado para el experimento se publicará en forma electrónica, de manera que el experimento sea repetible por terceros; sin embargo, las justificaciones aportadas en el capítulo de metodología respecto al proceso de selección y recreación de los modelos deberían hacerse también si los modelos fueran otros distintos. Prácticamente todo lo dicho allí (véanse las págs. 223 y ss.) está orientado a la repetibilidad del estudio.

El material original que se ha empleado como modelo no estaba en su totalidad dentro del dominio público. Las cinco partituras del grupo 2 eran material comercial: con la excepción del elemento de control, todas ellas han sido adquiridas en los canales oficiales y están sujetas a copyright. Sin embargo, se han utilizado en la confianza de que el uso académico entra dentro de los supuestos exentos de sujeción a las leyes y acuerdos internacionales de protección de la propiedad intelectual, y con la certeza de que el uso que se les ha dado no entra en

competencia con los derechos de su explotación comercial, que pertenecen exclusivamente a los propietarios del copyright. De la eventualidad de caer en violaciones del copyright se derivan una serie de problemas potenciales que aconsejan el uso exclusivo de material de dominio público, o bien publicado bajo una licencia que permita el uso libre, como por ejemplo las licencias Creative Commons. Los cinco modelos del grupo 1 eran material no sujeto a restricciones de copyright.

El hecho de usar hojas de papel en formato DIN A4 plantea el problema de los márgenes reducidos en comparación con los formatos comerciales, más grandes, lo que habrá de tenerse en cuenta para la repetibilidad.

Para el cálculo del índice V de Aiken en el apartado de validación, se ha empleado el código de R que aparece en el apéndice F.

13.5. Aplicabilidad

Aplicabilidad del método y el diseño del material

Dado que se encuentra adecuadamente documentado, el diseño general del material empleado puede usarse como base para estudios similares cuyo objetivo sea estudiar de forma comparativa distintos modelos de generación de documentos de carácter gráfico, sin limitarse a notación musical. La tipografía en general se presta también a este tipo de análisis. Los conceptos, descritos en el apartado de metodología, de «Ajuste de máximos» y «Ajuste de mínimos» para la recreación de los modelos, se pueden trasladar a otros materiales existentes, extraídos del uso cotidiano, para formar conjuntos de experimentos comparativos como el que se ha empleado.

Aplicabilidad de los resultados

Los resultados apoyan la promoción del uso de software libre en la educación musical, aunque no se han limitado a este ámbito. Los participantes incluían tanto a estudiantes y profesores como a músicos de otras ocupaciones, y las partituras pertenecen tanto a los niveles básicos como a los más avanzados.

La importancia de la notación como elemento omnipresente en la actividad del músico, está demostrada y apoya un énfasis por parte de las administraciones educativas en la promoción de la enseñanza de la edición de partituras en los centros de educación musical.

Para la producción de partituras musicales, no está justificado el gasto en programas priva-

tivos que tienen un elevado coste y limitan severamente las opciones de uso del propietario. En cambio, el uso de software libre no limita la descarga y redistribución gratuitas, sino que las fomenta, por lo que estos factores por sí solos no constituyen una razón para preferir las soluciones privativas más populares.

Los resultados del estudio se refieren solamente a la calidad visual, legibilidad e idoneidad para el propósito de la notación, expresada a través de la preferencia de los músicos hacia las partituras recreadas. No se dice nada acerca de la facilidad de manejo o de la adecuación a otras tareas como la composición, la confección de arreglos, el análisis y otras disciplinas de la música en su conjunto.

13.6. Conclusiones finales

En educación y en aplicaciones relacionadas con la transcripción y la recuperación del patrimonio musical, ya sea escrito o de tradición oral, y tanto de música seria como popular, el software libre ofrece un valor para la tipografía musical del que no goza ninguna otra solución de carácter privativo: su disponibilidad y sus posibilidades de instalación y uso ilimitados dentro de la legalidad.

No puede entenderse que un trabajo de transcripción puesto en el dominio público, lo esté como documento en un formato cerrado o que solo se pueda modificar con un programa no libre. Además, los desarrollos libres permiten a la institución que los utiliza efectuar correcciones, mejoras y traducciones que nunca podría hacer de otro modo.

El software libre, según muestran los resultados de los dos estudios, presenta propiedades que lo hacen adecuado para su empleo como medio de producción de material de notación en educación musical, en investigación y en usos profesionales, con unos resultados que los músicos aprecian como de alta calidad y se adecúan de manera satisfactoria a los diversos tipos de partitura que éstos utilizan.

Según indican las medidas de uso realizadas, las aplicaciones libres de tipografía musical y relacionadas como LilyPond, Frescobaldi, y MuseScore son minoritarias entre los usuarios. Sin embargo, eso puede cambiar dado que ya se están utilizando de forma muy extensa en grandes proyectos y en multitud de trabajos del más alto nivel técnico. Las autoridades en materia de Educación y la propia comunidad educativa tienen mucho que ganar acercándose a estas soluciones.

Apéndices

Apéndice A.

Lista completa de los desarrolladores de LilyPond

1: (6387) Han-Wen Nienhuys	52: (16) Dan Eble	103: (1) Vegard Nossum
2: (3225) Graham Percival	53: (15) Aleksandr Andreev	104: (1) Tristan Jelacic
3: (2476) Jan Nieuwenhuizen	54: (13) Paul Morris	105: (1) Trevor Bača
4: (1545) Francisco Vila	55: (13) Patrick Schmidt	106: (1) Travis Briggs
5: (1355) David Kastrup	56: (12) Joseph Wakeling	107: (1) Torsten Hämmerle
6: (1311) John Mandereau	57: (11) Ralph Palmer	108: (1) Thomas Klausner
7: (917) Reinhold Kainhofer	58: (11) Heikki Tauriainen	109: (1) Seafire
8: (785) Joe Neeman	59: (9) Michael Käßler	110: (1) Rutger Hofmann
9: (675) Jean-Charles Malahieude	60: (8) Matthias Kilian	111: (1) Richard Gay
10: (621) Trevor Daniels	61: (8) Han-Wen Nienhuys & Jan Nieuwenhuizen	112: (1) Paul Jarc
11: (614) Werner Lemberg	62: (7) Pavel Roskin	113: (1) Owen Tuz
12: (527) Neil Puttock	63: (7) Jeffrey B. Reed	114: (1) Nathan Reed
13: (518) Carl D. Sorensen	64: (7) Glen Prideaux	115: (1) Michael Ellis
14: (486) Till Rettig	65: (6) Carl Peterson	116: (1) Matthias Neeracher
15: (480) Phil Holmes	66: (5) Ralf Wildenhues	117: (1) Matthew Rowles
16: (415) Heikki Junes	67: (5) Peter Chubb	118: (1) Masamichi Hosoda
17: (357) James Lowe	68: (5) Mikemoral	119: (1) Markus Oehme
18: (352) Patrick McCarty	69: (5) Matthieu Jacquot	120: (1) Mark H Weaver
19: (302) Mike Solomon	70: (5) Eryolf Østrem	121: (1) Mark Hindley
20: (286) Mats Bengtsson	71: (5) Christian Hitz	122: (1) Lukasz Czerwiński
21: (245) Keith OHara	72: (5) Anthony Fok	123: (1) Lukas Pietsch
22: (208) Mark Polesky	73: (4) Zefram	124: (1) Kurt Kroon
23: (179) Valentin Villenave	74: (4) Xavier Scheuer	125: (1) Knut Petersen
24: (176) Julien Rioux	75: (4) Thomas Morgan	126: (1) Kevin Barry
25: (164) Nicolas Sceaux	76: (4) Stefan Weil	127: (1) Keaton Mowery
26: (136) Janek Warchoł	77: (4) Simon Bailey	128: (1) Karin Hoethker
27: (113) Jürgen Reuter	78: (4) Rodolfo Zitellini	129: (1) Johannes Rohrer
28: (74) Erlend Aasland	79: (4) Laura Conrad	130: (1) Jay Anderson
29: (74) Erik Sandberg	80: (4) Boris Shingarov	131: (1) Jan-Peter Voigt
30: (67) Federico Bruni	81: (4) Benjamin Peterson	132: (1) Hu Haipeng
31: (66) Marc Hohl	82: (3) Wilbert Berendsen	133: (1) Hajo Dezeliski
32: (53) Rune Zedeler	83: (3) Tineke de Munnik	134: (1) Guy Stalnakar
33: (52) Yoshiki Sawada	84: (3) Milan Zamazal	135: (1) Graham Breed
34: (48) Dénes Harmath	85: (3) Michael Welsh Duggan	136: (1) frederic gohier
35: (44) Pavel Fric	86: (3) Martin Tarenskeen	137: (1) Felix Janda
36: (41) Benkő Pál	87: (3) Kieren MacMillan	138: (1) Felipe Gonçalves Assis
37: (38) Bertrand Bordage	88: (3) Kevin Dalley	139: (1) Felipe Castro
38: (37) Pedro Kroger	89: (3) Arnold Theresius	140: (1) Ed Catmur
39: (36) Colin Campbell	90: (3) Alexander Kobel	141: (1) Dieter Grollmann
40: (33) David Nalesnik	91: (2) Sven Axelsson	142: (1) David Pounder
41: (30) Walter Garcia-Fontes	92: (2) Risto Vääräniemi	143: (1) Daniel Tonda Castillo
42: (29) Jonathan Kulp	93: (2) Piers Titus van der Torren	144: (1) Daniel Johnson
43: (23) Urs Liska	94: (2) Maximiliano García de Veas	145: (1) Daisuke Yamashita
44: (23) Thomas Morley	95: (2) Luca Rossetto Casel	146: (1) Choan Gálvez
45: (21) Adam Spiers	96: (2) Jean-Jacques Gerbaud	147: (1) Carsten Steger
46: (20) Frédéric Bron	97: (2) James E. Bailey	148: (1) Carsten Block
47: (19) Maximilian Albert	98: (2) Henning Hraban Ramm	149: (1) Bernard Hurley
48: (18) Ian Hulin	99: (2) Damien Heurtebise	150: (1) Arjan Bos
49: (18) Andrew Hawryluk	100: (2) Chris Snyder	151: (1) Andrew Wilson
50: (17) Devon Schudy	101: (2) Anders Pilegaard	152: (1) Alberto Simoes
51: (16) Tom Cato Amundsen	102: (1) Wol	153: (1) Abraham Lee

Figura 104: Lista completa de los 153 desarrolladores del proyecto LilyPond desde su creación hasta abril de 2015, en orden inverso del número de contribuciones individuales al código del proyecto o de su documentación.

Apéndice B.

El epitafio de Sícilo

La partitura musical completa más antigua que ha llegado hasta nuestros días es esta inscripción funeraria del s.I en una columna de mármol que se conserva en el Museo Nacional de Dinamarca, en Copenhague (Burkholder, Grout, Palisca y Torrellas, 2011; Solomon, 1984; de Souza, 2012). Los tratadistas la atribuyen a Sícilo, hijo de Euterpe, que la dedica a su esposa.

El propósito de este apéndice es demostrar una de las características más interesantes del software de tipografía musical LilyPond: su soporte para el conjunto de caracteres Unicode.

La codificación Unicode contempla un modo de representación compacto conocido como UTF-8. Al emplear este esquema para codificar los documentos de LilyPond, es posible producir partituras multilingües siempre que la fuente tipográfica contenga los glifos correspondientes a los códigos de carácter requeridos.

El ejemplo es reproducible a partir del código de LilyPond de la figura 106.

Los signos alfabéticos que aparecen sobre el texto son las notas musicales. Las que tienen una raya tienen duración doble; las que tienen una raya y un punto tienen duración triple. La transcripción procede de Reinach y Agostini (1984) y aparece también en de Candé (1981). Se incluye una transcripción fonética del texto. La traducción aproximada es la siguiente:

*Mientras vivas, brilla;
no te entristezcas más de lo necesario,
porque la vida es corta
y el tiempo se cobra su tributo.*

Ἐπιτάφιος τοῦ Σεικίλου
Epitafio de Sículo

Ὅσον ζῆς, φαί - νου,
 Ὄ - son dsés, fé - nu,
 μη - δὲν ὄ - λως σὺ
 mē - den o - lōs si
 λυ - ποῦ· πρὸς ὀ - λί -
 li - pú. Pros o - lí -
 γον ἐσ - τί τὸ ζῆν
 gon es - tí to dsen,
 τὸ τέ - λος ὀ χρό -
 to té - los o jró -
 νος ἄ - παι - τεῖ.
 nos a - pe - tei.

Figura 105: El epitafio de Sículo. A la izquierda, el original; la letra aparece en gris; la música en negro. A la derecha, la transcripción de Reinach, realizada en LilyPond por el autor. Las líneas respectivas, en orden de arriba a abajo, se corresponden.

```

\version "2.19.9"

\header {
  title = "Ἐπιτάφιος τοῦ Σεικίλου"
  subtitle="Epitafio de Sículo"
}

seikilos = \relative f {
  \key d \dorian
  \time 6/8
  g'8 d'4 d4.
  b8( c d) c4. \break
  b4 c8 d c( b)
  a8 \bar "" \break g4 a8( f4)
  g8 b d \bar "" \break c b( c)
  b8 g4 a8( f4) \break
  g8 b a c d \bar "" \break b
  g8 g4 g8( e d) \bar "||"
}

letraGriego = \lyricmode {
  Ὄ -- σον ζῆς, ψαί -- νου,
  μη -- δὲν ὄ -- λως σὺ λυ -- ποῦ.
  πρὸς ὁ -- λί -- γον ἐσ -- τὶ τὸ ζῆν
  τὸ τέ -- λος ὁ χρό -- νος
  ἀ -- παί -- τεῖ.
}

letraFonetica = \lyricmode {
  Ó -- son dsês, fé -- nu,
  mē -- den o -- lōs si li -- pú.
  Pros o -- lí -- gon es -- tí to dsen,
  to té -- los o jró -- nos
  a -- pe -- tei.
}

{
  \new Staff { \seikilos }
  \addlyrics { \letraGriego }
  \addlyrics { \letraFonetica }
}

\layout { indent = 0 }
\paper { line-width = 6\cm }

```

Figura 106: Código de LilyPond de la transcripción del epitafio de Sículo que aparece a la derecha en la figura 105.

Apéndice C.

Modelos A y B de los casos del experimento

En cada uno de los diez casos de las páginas siguientes, el elemento de la izquierda es el original, modelo A. El de la derecha, elemento B, es siempre la versión recreada mediante Lily-Pond, el programa objeto del experimento que forma parte del estudio 2. Los casos 5 y 10 son de control y en ellos los elementos A y B son iguales.

Durante la realización del experimento se aleatoriza la posición izquierda o derecha de los modelos A y B independientemente para cada caso y para cada sujeto.

Debe tenerse en cuenta que la reducción al 50 % y la calidad de impresión de una impresora estándar pueden producir diferencias entre los modelos de cada pareja que no aparecerían en condiciones ideales y a tamaño real.

The image displays a musical score for Viola, titled "Caresse sur l'océan". The score is presented in two columns of staves. The left column contains staves 1 through 32, and the right column contains staves 6 through 29. The music is written in bass clef with a key signature of one flat (B-flat) and a 4/4 time signature. The piece begins with a piano (*p*) dynamic. The notation includes various note values such as quarter, eighth, and sixteenth notes, as well as rests and fermatas. The score concludes with a final fermata on a whole note.

Figura 107: Caso nº 1. Tipo: partitella, para violoncello. Título: *Caresse sur l'océan*.

217

222

227

232

238

240

217

222

227

232

238

240

con espres

pp

Figura 108: Caso nº 2. Tipo: pieza para piano. Título: Sonata nº 3 en Do menor, Op.35, de J.L. Dussek.

The image displays two versions of a musical score for the song "Im Haine" by Franz Schubert. The score is presented in two columns, labeled Model A and Model B. Each column contains a vocal line and a piano accompaniment. The tempo is marked "Mässig" and the dynamics include "pp". The lyrics are: "Sonnen-strah-len durch die Tan-nen, wie sie fal-len, zieh'n von dan-nen al-le Schmer-zen, und-im Her-zen woh-net rei-ner Frie-de nur, zieh'n von". The piano accompaniment features a steady eighth-note bass line and a more active treble line with chords and melodic fragments.

Figura 109: Caso nº 3. Tipo: *Lied* con acompañamiento. Título: *Im Haine*, D.738, de F. Schubert.

The image displays two systems of musical notation for a drum set, arranged in two rows. Each system consists of eight staves. The notation is written on a five-line staff with various rhythmic symbols, including eighth notes, quarter notes, and rests. There are also dynamic markings such as accents (^) and a forte (f) marking. The measures are numbered at the bottom of each staff: 18, 22, 26, 30, 34, 38, 42, and 46. The notation is complex, with many notes and rests, indicating a detailed drum set part.

Figura 111: Caso n° 5. Tipo: percusión. Parte de batería de *How You Remind Me*, de Daniel Adair.

♩ = 77
A

A A

Verse 1: You stood be-fore cre-a-tion, for-e-ver with-in Your hand-

5

D D

F#m F#m

8 You spoke all life in-to mo-tion, my soul now to-

B B

A A

11 Verse 2: You stood be-fore my fail-
Verse 3: I'll walk u-pon sal-va-

D D

14 and car-ried the cross for my shame. My
Your Spi-rit a-live in me. D My

F#m F#m

17 sin weighed u-pon my soul now to stand
life to de-clare Your pro-mise. My soul now to stand

♩ = 77
A

A A

Verse 1: You stood be-fore cre-a-tion, for-e-ver with-in Your hand-

5

D D

F#m F#m

8 You spoke all life in-to mo-tion, my soul now to-

B B

A A

11 Verse 2: You stood be-fore my fail-
Verse 3: I'll walk u-pon sal-va-

D D

14 and car-ried the cross for my shame. My
Your Spi-rit a-live in me. D My

F#m F#m

17 sin weighed u-pon my soul now to stand
life to de-clare Your pro-mise. My soul now to stand

Figura 112: Caso nº 6. Tipo: *Lead Sheet* (hoja-guía de acordes). Título: *The Stand*.

The image displays two versions, A and B, of a guitar piece. The notation is on a single staff with a treble clef and a key signature of one sharp (F#). The piece consists of 12 measures. The first measure is a whole note chord. The second measure is a quarter note chord. The third measure is a quarter note chord. The fourth measure is a quarter note chord. The fifth measure is a quarter note chord. The sixth measure is a quarter note chord. The seventh measure is a quarter note chord. The eighth measure is a quarter note chord. The ninth measure is a quarter note chord. The tenth measure is a quarter note chord. The eleventh measure is a quarter note chord. The twelfth measure is a quarter note chord. The score includes various musical notations such as slurs, ties, and dynamic markings like 'mf' and 'rit.'. There are also some handwritten annotations in the original image, such as 'III', 'V', 'II', and 'II-'.

Figura 113: Caso nº 7. Tipo: pieza para guitarra. Título: *Habanera*, de Juan Pérez Ribes.

The image displays two versions of a musical score, labeled A and B, for the piece 'Premier Trio en Sol' by Claude Debussy. The score is arranged in two systems, each containing three staves (piano, violin, and cello). The first system covers measures 26 to 38, and the second system covers measures 43 to 55. The notation includes various musical symbols such as notes, rests, dynamics (p, f), and articulation marks (accents, staccato). The two versions, A and B, are presented side-by-side for comparison.

Figura 115: Caso nº 9. Tipo: música de cámara. Título: *Premier Trio en Sol*, de C. Debussy.

Apéndice D.

Modelo del cuestionario del estudio 2

Las páginas siguientes contienen el cuestionario que se utilizó para la encuesta y el experimento del estudio 2. El material empleado para la sección cuarta son los diez modelos que aparecen reducidos al 50 % en el apéndice C.

El cuestionario de validación correspondiente aparece en el apéndice E.

Encuesta sobre el uso de partituras musicales en educación y el uso de aplicaciones de tipografía musical

Sección primera: datos demográficos y acerca de tu ocupación relacionada con la música

1. Edad:
2. Sexo: M F
3. Diestro/a Ambidextro/a Zurdo/a
4. ¿Utilizas graduación para la vista en tus estudios o durante tu trabajo? Sí No
5. Por favor, indica qué actividad o actividades te definen mejor en el campo de la música.
Posibilidad de respuesta múltiple.
 - a) Soy músico, instrumentista o cantante.
 - b) Director/a de orquesta, banda, coro, conjunto instrumental o vocal.
 - c) Compositor/a o arreglista.
 - d) Profesor/a de música, en cualquier nivel.
 - e) Estudiante de música, en cualquier nivel.
 - f) Soy aficionado/a a la música.
 - g) Otro tipo de relación con la música.

¿Cuál?

6. ¿Qué nivel de estudios musicales has completado?

Indica el nivel más alto que hayas completado solamente.

- a) No tengo estudios formales de música.
- b) Estudios de música no reglados (escuela municipal, academia...).
- c) Estudios de conservatorio de nivel elemental.
- d) Estudios de conservatorio de nivel profesional.
- e) Estudios superiores de música.
- f) Posgrado o doctorado en música.

Sección segunda: uso habitual de partituras

7. En tu actividad musical o en la enseñanza ¿con qué frecuencia utilizas estos tipos de partitura?

Incluye: libros o partituras didácticas, piezas sueltas, estudios, ejercicios, exámenes, pequeños fragmentos, particellas, guiones o partituras de solista, orquestales y similares.

No incluye: musicogramas, pictogramas, lenguaje de signos.

Escoge para cada tipo de partitura entre: 0 (nunca o menos de una vez al año), 1 (una o más veces al año, y menos de una vez al mes), 2 (una o más veces al mes, y menos de una vez a la semana), 3 (una o más veces por semana, pero no diariamente), 4 (normalmente una o más veces al día)

- a) 0 1 2 3 4 Partituras comerciales originales, en papel.
- b) 0 1 2 3 4 Partituras escaneadas o fotocopiadas.

- c) 0 1 2 3 4 Partituras realizadas a mano, en papel (originales o copias).
- d) 0 1 2 3 4 Partituras realizadas a mano en pizarra tradicional o electrónica.
- e) 0 1 2 3 4 Partituras editadas mediante ordenador, hechas por otra persona.
- f) 0 1 2 3 4 Partituras editadas mediante ordenador, hechas por mí mismo.
- g) 0 1 2 3 4 Partituras proyectadas o en una pantalla de cualquier tipo.
- h) 0 1 2 3 4 Algún tipo de partitura no incluido en ninguno de los anteriores grupos.

¿Cuál?

8. Durante una semana de tiempo en tu actividad habitual ¿cuántas horas dedicas a la lectura, escritura, anotación o análisis de partituras musicales, de cualquier tipo?

Sección tercera: uso de software de edición de partituras y formación en el mismo

9. Si has elaborado alguna vez partituras musicales mediante ordenador, ¿qué programa o programas has utilizado, y con qué frecuencia?

Escoge para cada programa entre: 0 (nunca o menos de una vez al año), 1 (una o más veces al año, y menos de una vez al mes), 2 (una o más veces al mes, y menos de una vez a la semana), 3 (una o más veces por semana, pero no diariamente), 4 (normalmente una o más veces al día)

- a) 0 1 2 3 4 Finale.
- b) 0 1 2 3 4 Sibelius.
- c) 0 1 2 3 4 Encore.
- d) 0 1 2 3 4 MuseScore.
- e) 0 1 2 3 4 LilyPond.
- f) 0 1 2 3 4 Servicios online o aplicaciones de tableta o "smartphone".
- g) 0 1 2 3 4 Otro.

¿Cuál?

10. Tú, tu empresa o tu centro educativo ¿habéis adquirido o instalado algún programa de edición de partituras? No. No lo sé. Sí.

¿Cuál?

11. ¿Has recibido alguna vez algún tipo de formación específica para la edición de partituras con medios informáticos? No. Sí.

¿Para qué programa o programas?

Sección cuarta: evaluación comparativa de modelos de partitura y sistemas de tipografía musical

En este momento debes ponerte las gafas si manifestaste necesitarlas en la pregunta nº 4.

12. En cada uno de los grupos siguientes, si tuvieses que trabajar con una de estas dos partituras, ya sea para leerlas, estudiarlas o analizarlas, ¿cuál de ellas preferirías?

Selecciona la que te parece estéticamente más agradable, o mejor edición, o de lectura más fácil, o “mejor hecha” en general, y en caso de duda piensa cuál comprarías si te hubiesen encargado gastar dinero en una de ellas. No elijas una solamente por su calidad de impresión. Contesta diciendo “**izquierda**”, “**derecha**” o “**soy indiferente**” si ninguna de ellas es claramente preferible. No hay límite de tiempo.

Combinación nº:

Fecha:

1	2	3	4	5	6	7	8	9	10
<input type="checkbox"/> A <input type="checkbox"/> B									
<input type="checkbox"/> I <input type="checkbox"/> D									

Control 1

Control 2

RETEST. Fecha:

1	2	3	4	5	6	7	8	9	10
<input type="checkbox"/> A <input type="checkbox"/> B									
<input type="checkbox"/> I <input type="checkbox"/> D									

Control 1

Control 2

Apéndice E.

Cuestionario de validación del estudio 2

En las páginas siguientes se muestra el cuestionario para la validación de la encuesta del estudio 2 que aparece en el apéndice D.

Los jueces tuvieron a su disposición el mismo material que se utilizó con los participantes, y que aparece a una escala del 50 % en el apéndice C.

Nombre del juez validador:

Cuestionario de validación de la “encuesta sobre el uso de partituras musicales en educación y el uso de aplicaciones de tipografía musical”

Para cada bloque de preguntas de la encuesta, se le pedirá que, por favor, valore de 0 a 10 las afirmaciones que se hacen sobre las preguntas de dicho bloque, donde 0 significa “Estoy en completo desacuerdo” y 10 significa “Estoy completamente de acuerdo”.

Sección primera: datos demográficos y sobre tu ocupación relacionada con la música

1. *Edad*
2. *Sexo*
3. *Diestro/Ambidextro/Zurdo*
4. *¿Utilizas graduación para la vista en tus estudios o durante tu trabajo?*
5. *Por favor, indica qué actividad te define mejor en el campo de la música.*
6. *¿Qué nivel de estudios musicales has completado?*

Por favor, valore de 0 a 10 en qué grado se cumplen estas afirmaciones:

Pertinencia: las preguntas de la sección *Datos demográficos y sobre tu ocupación relacionada con la música* deben formar parte de este cuestionario. 0 1 2 3 4 5 6 7 8 9 10

Redacción: las preguntas de la sección *Datos demográficos y sobre tu ocupación relacionada con la música* están bien elaboradas. 0 1 2 3 4 5 6 7 8 9 10

Observaciones

Sección segunda: uso que haces habitualmente de material de notación musical (partituras)

7. *En tu actividad musical o en la enseñanza ¿con qué frecuencia utilizas estos tipos de partitura?*
8. *Durante una semana de tiempo en tu actividad habitual ¿cuántas horas dedicas a la lectura, escritura, anotación o análisis de partituras musicales, de cualquier tipo?*

Por favor, valore de 0 a 10 en qué grado se cumplen estas afirmaciones:

Pertinencia: las preguntas de la sección *Uso que haces habitualmente de material de notación musical (partituras)* deben formar parte de este cuestionario. 0 1 2 3 4 5 6 7 8 9 10

Redacción: las preguntas de la sección *Uso que haces habitualmente de material de notación musical (partituras)* están bien elaboradas. 0 1 2 3 4 5 6 7 8 9 10

Observaciones

Sección tercera: uso de software de edición de partituras y formación en el mismo

9. *Si has elaborado alguna vez partituras musicales mediante ordenador, ¿qué programa o programas has utilizado con más frecuencia?*

10. Tú, tu empresa o tu centro educativo ¿habéis adquirido o instalado algún programa de edición de partituras?

11. ¿Has recibido alguna vez algún tipo de formación específica para la edición de partituras con medios informáticos?

Por favor, valore de 0 a 10 en qué grado se cumplen estas afirmaciones:

Pertinencia: las preguntas de la sección *Uso de software de edición de partituras y formación en el mismo* deben formar parte de este cuestionario. 0 1 2 3 4 5 6 7 8 9 10

Redacción: las preguntas de la sección *Uso de software de edición de partituras y formación en el mismo* están bien elaboradas. 0 1 2 3 4 5 6 7 8 9 10

Observaciones

Sección cuarta: evaluación comparativa de modelos de partitura y sistemas de tipografía musical

12. En cada uno de los grupos siguientes, si tuvieses que trabajar con una de estas dos partituras, ya sea para leerlas, estudiarlas o analizarlas, ¿cuál de ellas preferirías?

Por favor, valore de 0 a 10 en qué grado se cumplen estas afirmaciones:

Pertinencia: las preguntas de la sección *Evaluación comparativa de modelos de partitura y sistemas de tipografía musical* deben formar parte de este cuestionario. 0 1 2 3 4 5 6 7 8 9 10

Redacción: las preguntas de la sección *Evaluación comparativa de modelos de partitura y sistemas de tipografía musical* están bien elaboradas. 0 1 2 3 4 5 6 7 8 9 10

Modelo 1

Las dos partituras presentan un tamaño de pauta similar. 0 1 2 3 4 5 6 7 8 9 10

Los márgenes de las dos partituras son similares. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan un color similar del papel. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan una calidad de impresión similar. 0 1 2 3 4 5 6 7 8 9 10

Las partituras no son de una editorial claramente identificable. 0 1 2 3 4 5 6 7 8 9 10

Modelo 2

Las dos partituras presentan un tamaño de pauta similar. 0 1 2 3 4 5 6 7 8 9 10

Los márgenes de las dos partituras son similares. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan un color similar del papel. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan una calidad de impresión similar. 0 1 2 3 4 5 6 7 8 9 10

Las partituras no son de una editorial claramente identificable. 0 1 2 3 4 5 6 7 8 9 10

Modelo 3

Las dos partituras presentan un tamaño de pauta similar. 0 1 2 3 4 5 6 7 8 9 10

Los márgenes de las dos partituras son similares. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan un color similar del papel. 0 1 2 3 4 5 6 7 8 9 10

Las dos partituras presentan una calidad de impresión similar. 0 1 2 3 4 5 6 7 8 9 10

Las partituras no son de una editorial claramente identificable. 0 1 2 3 4 5 6 7 8 9 10

Modelo 4

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 5

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 6

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 7

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 8

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 9

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Modelo 10

- Las dos partituras presentan un tamaño de pauta similar.
- Los márgenes de las dos partituras son similares.
- Las dos partituras presentan un color similar del papel.
- Las dos partituras presentan una calidad de impresión similar.
- Las partituras no son de una editorial claramente identificable.

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 8 9 10

Observaciones

Sección quinta: valoración general de la encuesta

El objetivo de la encuesta es medir el grado de aceptación de un determinado método de producción de partituras musicales, por parte de una serie de sujetos del campo de la música. Por favor, puntúe las siguientes afirmaciones generales sobre el cuestionario:

Los datos que se piden sobre cada sujeto son adecuados al objetivo de la encuesta. 0 1 2 3 4 5 6 7 8 9 10

La cantidad de preguntas es adecuada para el objetivo. 0 1 2 3 4 5 6 7 8 9 10

El tipo de preguntas es adecuado para el objetivo. 0 1 2 3 4 5 6 7 8 9 10

El diseño de la encuesta responde al objetivo perseguido. 0 1 2 3 4 5 6 7 8 9 10

Observaciones

Apéndice F.

Cálculo del coeficiente de validez V de Aiken

El código de R que aparece a continuación se usó para calcular los valores del coeficiente V de Aiken (Aiken, 1980, 1985; Penfield y Giacobbi, 2004) en la fase de validación del estudio 2 (secciones 11.6.4 y 11.6.5, y figuras 76, 77 y 78; páginas 247 a 249).

El valor se calcula como como $V = \frac{\bar{X}-l}{k}$, donde \bar{X} es la media de las calificaciones de los jueces en la muestra, l es la calificación más baja posible y k es el rango de los valores posibles. Los límites inferior y superior del intervalo de confianza para V según el método *Score* de Wilson (Wilson, 1927), son $L, U = \frac{2nkV+z^2 \pm z\sqrt{4nkV(1-V)+z^2}}{2(nk+z^2)}$, donde z es el valor normalizado de la confianza y n es el número de jueces.

Los datos se leen a partir de un archivo CSV que contiene solamente las calificaciones de cada juez, en las filas, y las de cada pregunta en las columnas. El programa produce dos gráficos que representan los valores medios de V por pregunta y por juez con los márgenes del 90 % y 99 % de validez, y las rectas de umbral para $V = 0,5$ y $V = 0,7$. La figura 117 muestra un ejemplo de la salida del programa para un supuesto de 15 preguntas y 20 jueces.

```
datos <- read.csv("datosValidacion.csv", header = TRUE, sep=';', dec=',')
# datos<-datos[3:length(datos)] # opcional: eliminar las columnas nombre, fecha

l <- 0 # l = calificación más baja posible
k <- 10 # k = rango de valores posibles
n <- dim(datos)[1] # n = número de filas (número de jueces)
z90 <- qnorm(0.9495) # 1.64
# z95<-qnorm(0.975) # 1.96 ; percentil 97.5
z99 <- qnorm(0.9949) # 2.57

V <- function(x) (mean(x)-l)/k # V de Aiken

# Opcional: ordenar jueces por el valor de V
orden <- rev(order(apply(datos,1,V))) # vector de ordenación en función de V
datos <- datos[orden,] # ordenamos los datos

# Límites del intervalo de confianza por el método Score de Wilson
L <- function(x) (2*n*k*VV+x^2-x*sqrt(4*n*k*VV*(1-VV)+x^2))/(2*(n*k+x^2)) # inferior
U <- function(x) (2*n*k*VV+x^2+x*sqrt(4*n*k*VV*(1-VV)+x^2))/(2*(n*k+x^2)) # superior
```

Apéndice F. Cálculo del coeficiente de validez V de Aiken

```
graficoV <- function(valores,titulo,etiquetaX) {
  plot(valores,main=titulo,
       ylim=c(0.5,1),      # rango vertical
       xlab=etiquetaX,     # Pregunta o Juez
       ylab="V")
  lines(L(z99),lty=1)      # curvas inferior y superior para el 99%, en línea continua
  lines(U(z99),lty=1)
  # lines(L(z95),lty=1)    # curvas intermedias para el 95%, no son necesarias
  # lines(U(z95),lty=2)
  lines(L(z90),lty=2)     # curvas inferior y superior para el 90%, en línea discontinua
  lines(U(z90),lty=2)
  abline(,0.5,lty=3)      # recta umbral puntuación V = 0.5
  abline(,0.7,lty=4)      # recta umbral puntuación V = 0.7
  legend("bottomright",c("99%","90%"),lty=c(1,2)) # leyenda
}

par(mfrow=c(1,2))        # dos gráficos lado a lado

VV <- apply(datos,2,V)   # 2 = por columnas (por preguntas)
graficoV(VV,"Coef. V de Aiken por pregunta","Pregunta")

VV <- apply(datos,1,V)   # 1 = por filas (por jueces)
graficoV(VV,"Coef. V de Aiken por juez","Juez")
```

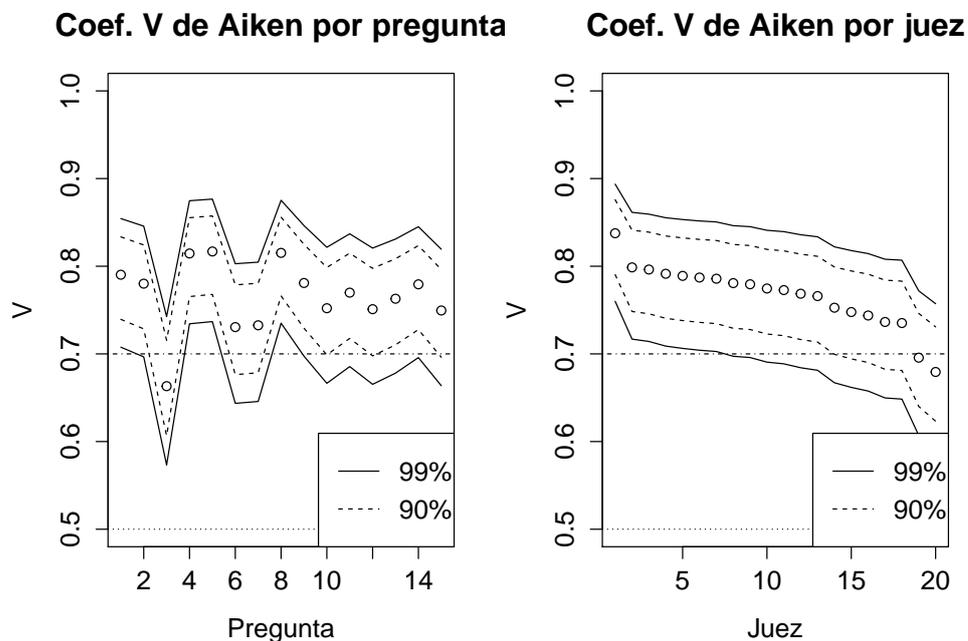


Figura 117: Ejemplo de salida del programa.

Apéndice G.

Glosario de términos técnicos y musicales

Se ha puesto el énfasis en la brevedad. Los términos en cursiva remiten a su vez a las entradas correspondientes del glosario.

algoritmo

Método o conjunto detallado de los pasos que permiten realizar alguna acción compleja como encontrar la solución a un problema.

articulación

En notación musical, signo o abreviatura que afecta a una nota o un conjunto de ellas y que indica variantes diversas en la forma de atacarlas, producirlas o separarlas entre sí.

AWK

Lenguaje especializado en el proceso de texto, desarrollado por A. Aho, P. Weinberger y B. Kernighan en 1977.

C

Un lenguaje de programación que pone el énfasis en la velocidad, inventado en 1970 por Dennis Ritchie.

C++

Un dialecto del lenguaje C. Desarrollado a principios de los años 80 por Bjarne Stroustrup.

código fuente

Texto que, una vez procesado o compilado (v. *compilación*), produce un documento o un programa ejecutable. El código fuente sigue las reglas de un determinado *lenguaje* de ordenador.

compás

Fragmento breve de notación musical, que representa un determinado número de pulsos de tiempo y está delimitado por líneas verticales. También: especificación simbólica del tipo de agrupación de los pulsos de tiempo.

compilación

Proceso, realizado por un programa compilador, que transforma un texto llamado *código fuente* en un documento o programa ejecutable de ordenador. El compilador está especializado en un *lenguaje* determinado.

copyleft

Cláusula de una *licencia* libre que, en caso de redistribución del material protegido por dicha licencia, obliga a hacerlo manteniendo la propia cláusula, protegiendo así la libertad que aquélla concede.

copyright

Cierto tipo de protección de la propiedad intelectual, que concede al autor o al editor de una obra exclusividad absoluta de derechos sobre ella. Creado originalmente para conceder el monopolio durante un tiempo limitado a los impresores de libros y para promover la actividad de los creadores.

Creative Commons

Conjunto de licencias de protección de obras artísticas y otros documentos, que ofrecen un término medio entre el monopolio y el dominio público. Organización sin ánimo de lucro que las promueve.

distribución

En el contexto de *GNU/Linux*, una distribución es un proyecto que ofrece a los usuarios la fácil instalación y actualización del sistema operativo. Ofrece por lo general una o más versiones del núcleo Linux, aplicaciones diversas y herramientas complementarias.

Dominio Público

Estatus de una obra que ha perdido la protección debida a los plazos del *copyright*.

figura

En notación musical, un símbolo que representa una nota y que expresa su duración mediante la forma, y su altura mediante la posición vertical dentro del *pentagrama*.

FSF

(Free Software Foundation) Fundación para el *Software Libre*. Organización sin ánimo de lucro, con sede en Boston, que promueve el sistema operativo *GNU* entre otros muchos proyectos.

Git

Cierta herramienta de *software* para el control concurrente y descentralizado de versiones de archivos de texto.

GNU

Proyecto de sistema operativo libre iniciado por Richard Stallman y que es la base de los sistemas *GNU/Linux* de todo el mundo. En rigor, el sistema GNU no es completamente autosuficiente sin un núcleo como *Linux*. GNU acoge y apoya otros muchos proyectos de software libre como *LilyPond*.

GNU/Linux

Sistema operativo completo basado en un núcleo *Linux* y un entorno operativo *GNU*. Habitualmente se encuentra en forma de “distribuciones” que facilitan el acceso a amplias bibliotecas de software compatible.

GPL

Licencia pública general de *GNU*, promovida por la *FSF*; aplicada a un programa, lo caracteriza como software libre, y es una licencia con *copyleft*.

intervención

En inglés, “commit”. En el contexto del control de versiones, el registro de un cambio de versión en un archivo o en un conjunto de ellos, por un determinado autor.

Java

Lenguaje de programación orientado a la portabilidad entre sistemas distintos. Desarrollado por James Gosling, de Sun Microsystems, a partir de 1991.

LaTeX

Lenguaje de ordenador para la producción de documentos de texto a partir de un *código fuente*. Basado en *T_EX*; desarrollado por Leslie Lamport en 1984.

lenguaje de ordenador

Conjunto de reglas formales, léxicas y gramaticales, que permiten a un *compilador* (especializado en ese lenguaje) convertir el *código fuente* en un documento o programa ejecutable.

LilyPond

Sistema de *software libre* integrado en *GNU* para la preparación de partituras musicales a partir de un documento de texto. Compatible con sistemas Windows, MacOS y GNU/Linux, fue iniciado a finales de los años 90 por Han-Wen Nienhuys y Jan Nieuwenhuizen.

licencia

Documento que acompaña a una obra, aplicado a ella por su creador, que especifica los términos que comprometen al usuario de dicha obra.

lied

Canción, en alemán.

LinEX

Cierta *distribución* de *GNU/Linux* creada en 2002 por la Junta de Extremadura para su uso en los colegios e institutos de esta comunidad autónoma, con el objetivo de ahorrar en el apartado de las licencias. Pionera en el género de distribuciones autonómicas e institucionales.

Linux

Informalmente: cierto sistema operativo libre. En rigor, Linux es solamente la pieza central o núcleo de un sistema operativo como *GNU*, formando un conjunto denominado *GNU/Linux*. Creado inicialmente por el programador finlandés Linus Torvalds, que le dio nombre, y mantenido por la comunidad.

Lisp

Veterano lenguaje de programación, aún en uso, desarrollado en Stanford a finales de los años 50.

Metafont

Lenguaje para la definición de fuentes tipográficas, desarrollado por Donald Knuth en 1979.

MIDI

“Musical Instrument Digital Interface”. Estándar para la transmisión y almacenamiento de secuencias musicales o “canciones” desde un ordenador hasta un instrumento musical electrónico, y para la comunicación de los instrumentos entre sí.

MusicXML

Un lenguaje para la definición e intercambio de partituras musicales.

notación musical

Lenguaje simbólico que permite fijar y transmitir la ejecución de música. La forma de la notación musical moderna o notación clásica occidental quedó estabilizada en el s.XVII y utiliza *figuras* y *pentagramas*.

Open Source

“Software de fuentes abiertas”. Software que se publica bajo una licencia que comparte ciertos aspectos del software libre, entre ellos que el *código fuente* está disponible para su *compilación* por parte del usuario.

particella

Parte musical escrita que corresponde a un solo músico o sección homogénea de músicos dentro de un conjunto.

partitura

Por extensión: toda notación musical. En su acepción más restringida, la partitura es el guión para el director de una orquesta, que contiene las partes de todos los músicos. Véase *Particella*.

pentagrama

Pauta para la escritura musical, de cinco líneas horizontales. Informalmente puede designar también pautas con un número de líneas distinto.

PDF

Formato de documento creado por Adobe para la descripción de páginas impresas listas para imprimir o visualizar en pantalla, orientado a una alta compatibilidad entre distintos sistemas.

PNG

Formato de imagen que no presenta pérdidas de calidad por compresión y es adecuado para gráficos. Se emplea habitualmente en la web.

Python

Un lenguaje de programación ampliamente difundido, sencillo de aprender y adecuado para prototipos y proyectos rápidos. Desarrollado por Guido van Rossum a partir de 1990.

Scheme

Un dialecto del lenguaje *Lisp*.

software

Programas y datos para un sistema informático, que proveen a una máquina de propósito general la capacidad de servir a una determinada aplicación.

software gratuito

En el contexto del software, “gratis” es el software de descarga y/o distribución exentas de coste, independientemente de si es (o no) *privativo*, *Open Source*, *software libre* o de si está sujeto o no a *copyleft*.

software libre

Software que se distribuye bajo una *licencia* que concede al usuario plena libertad para instalarlo, copiarlo, redistribuirlo, disponer del código fuente, modificarlo o mejorarlo y redistribuir la versión modificada o mejorada. A veces se confunde con el software de “fuentes abiertas” u *Open source*.

software privativo

Software no libre. Aquel que, en virtud de su licencia, se opone a una o más de las libertades que las licencias libres otorgan al usuario, especialmente la de examinar el *código fuente* y poder *compilarlo*.

TEX

Lenguaje para la composición tipográfica de textos, desarrollado por Donald Knuth en 1979.

Texinfo

Lenguaje informático empleado en la documentación del software. Se usa de forma oficial en todos los programas del proyecto GNU.

transcripción

Copiar música que aún no está escrita o que lo está en un sistema obsoleto, a un sistema de notación actualizado, o adaptarla con destino a un instrumento distinto.

versión

En el contexto del desarrollo de *software*, estado en que queda un archivo (normalmente *código fuente*) cada vez que se sufre algún cambio en el curso de su desarrollo. La versión es el objeto del control de estos cambios, la historia de los mismos y su autor o autores.

Apéndice H.

Software utilizado

El documento está elaborado exclusivamente con software libre y este anexo quiere servir como un pequeño homenaje de agradecimiento a los autores y colaboradores que han creado y mantenido los programas. La licencia en todos los casos es GPL u otra compatible con ésta.¹ Se omiten los números de versión por estar sujetos a cambios muy frecuentes. Como autor o autores de cada programa se consignan solamente a los creadores originales, pero dada su condición de software libre, la comunidad ha contribuido en mayor o menor manera en todos los proyectos y en algunos casos ésta es quien mantiene el desarrollo de los proyectos aún después de que los autores originales lo hayan abandonado; por ello, la frase “con aportaciones de la comunidad” debe entenderse como añadida después del nombre del autor o autores, en todos los casos.

Para la extracción de las imágenes de los archivos PDF en los modelos del material se ha usado la utilidad `pdftimages`. Los modelos del material, originales y recreados, están procesados con el GIMP, que es el programa de manipulación de imágenes de GNU, y con `imagemagick`, de ImageMagick Studio LLC.

Las ilustraciones originales están hechas con el lenguaje `TikZ`, de Till Tantau, e Inkscape, de diseño vectorial, de Ted Gould, Bryce Harrington y Nathan Hurst.

Las listas de aleatorización y anonimización se produjeron con guiones del lenguaje Python a través del entorno de desarrollo IDLE. Ambos son de Guido van Rossum. IDLE está mantenido por el SF IDLEfork project.

La recogida de datos se efectuó sobre una hoja de cálculo de LibreOffice Calc, de la Document Foundation. Basado en OpenOffice.org, de Oracle (y actualmente de la Apache Foundation). El cálculo estadístico y los gráficos de resultados están elaborados con el entorno matemático R, de Robert Gentleman y Ross Ihaka y mantenido por el R Core Development Team. El lenguaje R se manejó a través del entorno de desarrollo Rstudio, de J.J. Allaire y Joe Cheng, mantenido por RStudio, Inc.

¹Por ejemplo, Imagemagick se distribuye bajo licencia Apache. Ispell y de Python se publican bajo licencia Open Source y algunas versiones antiguas de Python no eran compatibles con la GPL.

Para la redacción del texto fuente del documento se ha empleado GNU emacs, de Richard Stallman. El corrector ortográfico empleado es ispell, de Geoff Kuenning.

El control de versiones del código fuente del documento y de R, así como las copias de seguridad, son obra de Git, de Linus Torvalds. El mismo programa se empleó para el análisis histórico del código fuente de LilyPond y otros.

La composición tipográfica del documento utiliza X_YT_EX, de Jonathan Kew, basado en L^AT_EX, de Leslie Lamport, basado a su vez en T_EX, de Donald Knuth. La plantilla está basada en la clase `scrbook` del paquete Koma-Script, de Makus Kohm. El control de las dependencias para la maquetación está hecho con la utilidad GNU Make, de Richard Stallman y con `latexmk`, de John Collins.

El texto está compuesto en un tipo Open Type Heuristica, de Andrej Panov, basado en Adobe Utopia, del que es una extensión.

La gestión de las referencias bibliográficas está hecha con Biblatex, de Philipp Lehman, basado en BibT_EX, de Oren Patashnik. Las reglas de formato del módulo APA-spanish para Biblatex fueron preparadas por el autor.

La tipografía musical de las ilustraciones originales se hizo gracias a GNU LilyPond, de Han-Wen Nienhuys y Jan Nieuwenhuizen.

Las utilidades y herramientas auxiliares de GNU para el manejo de la consola pertenecen a la distribución Debian GNU/Linux.

Bibliografía

- Abella, A. y col. (2003). *Libro blanco del software libre en España*. Mérida: Consejería de Educación, Ciencia y Tecnología de la Junta de Extremadura. Asociación de Usuarios Españoles de GNU/Linux.
- Abelson, H. y col. (1998). Report on the algorithmic language Scheme. *SIGPLAN Not.* 33(9), 26-76.
- Adair, D. (2009). How you remind me [Partitura musical]. En *Batterie magazine* (58).
- Adell, J. y Bernabé, Y. (2007). Software libre en educación. En J. Cabero (Ed.), *Tecnología educativa* (pp. 173-195). Madrid: McGraw-Hill.
- Agrawal, R., Imielinski, T. y Swami, A. (1993). Mining association rules between sets of items in large databases. En *Proceedings of the International Conference on Management of Data (SIGMOD-1993)* (pp. 207-216). Washington, D.C.
- Agrawal, R. y Srikant, R. (1994). Fast algorithms for mining association rules. En *Proc. 20th Int. Conf. Very Large Data Bases, (VLDB-94)* (pp. 487-499).
- Agresti, A. (2007). *An introduction to categorical data analysis* (2ª ed.). Nueva York: John Wiley & Sons, Inc.
- Aiken, L. R. (1980). Content validity and reliability of single items or questionnaires. *Educational and Psychological Measurement*, 40, 955-959.
- Aiken, L. R. (1985). Three coefficients for analyzing the reliability and validity of ratings. *Educational and Psychological Measurement*, 45(1), 131-142.
- Allen, W. F., Ware, C. P. y Garrison, L. M. (1867). *Slave songs of the United States* (2ª ed.). Nueva York: A. Simpson & Co.
- Almeida, J. J., Carvalho, N. R. y Oliveira, J. N. (2011). WIKI::SCORE: a collaborative environment for music transcription and publishing. *Information Services and Use*, 31(3), 177-187.
- American Psychological Association y col. (2011). *Publication manual* (6ª ed.). Washington: American Psychological Association (APA).
- Anders, T. (2007). Composing music by composing rules: design and usage of a generic music constraint system.
- Anders, T., Anagnostopoulou, C. y Alcorn, M. (2005). Strasheela: design and usage of a music composition environment based on the Oz programming model. En *Multiparadigm programming in Mozart/Oz* (pp. 277-291). Springer.

- Anson, D., Ezust, A., Casanova, M., Hoyt, J. y Dillon, M. (2013). jEdit (Versión 4.3pre17) [Programa de ordenador]. SourceForge. Recuperado el 20 de marzo de 2015, desde <http://www.jedit.org>
- Apel, W. (1961). *The notation of polyphonic music, 900-1600* (5ª ed.). Cambridge, Mass.: Mediaeval academy of America.
- Arjona, L. (2012). *Translations in libre software* (Tesis de Máster, Universidad Rey Juan Carlos. Máster Universitario en Software Libre).
- Arjona, L. y Robles, G. (2012). Mining for localization in Android. En *Proceedings of the 9th IEEE working conference on mining software repositories* (pp. 136-139). IEEE Press.
- Assayaag, G. y Timis, D. (1986). A toolbox for music notation. En *Proceedings of the 1986 international computer music conference*.
- Association Européenne des Conservatoires, Académies de Musique et Musikhochschulen (AEC) (Ed.). (2007). *Proceedings of the AEC Annual Congress and General Assembly*.
- AT&T. (2006). Graphviz - graph visualization software. Documento en línea. Recuperado el 24 de enero de 2006, desde <http://graphviz.org>
- Avelino, G., Valente, M. T. y Hora, A. (2015). What is the Truck Factor of popular GitHub applications? A first assessment. *PeerJ Preprints* 1233.
- Azevedo, B. M. y Almeida, J. J. (2013). ABC with a UNIX flavor. En *SLATE* (pp. 203-218). Oporto, Portugal.
- Baines, P. y Haslam, A. (2005). *Type and typography*. Londres: Lawrence King Publishing.
- Balaban, M. (1988). A music workstation based on multiple hierarchical views of music. En *Proceedings of the 1988 International Computer Music Conference*. International Computer Music Association. San Francisco.
- Banter, H. (1987). *Akkord lexikon*. Mainz, Germany: Schott's Söhne.
- Barksdale, A. (1957). *The printed note: 500 years of music printing and engraving*. Toledo, Ohio: The Toledo Museum of Art.
- Barton, L. W., Terzopoulos, C. J. y Craig-McFeely, J. (2007). Online access to manuscripts of Byzantine Chant. *Journal of Psaltic Notes*.
- Beethoven, L. (1970). String Quartet No.7, Op.59 No.1 [Partitura musical]. En *Ludwig van Beethoven: complete string quartets*. Reimpresión de (Beethoven, 1862). Nueva York: Dover.
- Beethoven, L. (1862). Quartette für 2 Violinen, Bratsche und Violoncell, Erster Band, Nr.43 [Partitura musical]. En *Ludwig van Beethovens Werke*. 6. Compuesta en 1806. Publicada originalmente en 1808. Breitkopf & Härtel.
- Belkin, A. (1994). Macintosh notation software: present and future. *Computer Music Journal*, 18(1).
- Bell, J. (2002). *Cómo hacer tu primer trabajo de investigación*. Barcelona: Gedisa.

- Bellini, P., Fioravanti, F. y Nesi, P. (1999). Managing music in orchestras. *Computer*, 32(9), 26-34.
- Bellini, P., Nesi, P. y Spinu, M. B. (2002). Cooperative visual manipulation of music notation. *ACM Trans. Comput.-Hum. Interact.* 9(3), 194-237.
- Bello, J., Chew, E. y Turnbull, D. (2008). *ISMIR 2008: Proceedings of the 9th International conference of music information retrieval*. Drexel University.
- Benkler, Y. (2004). Sharing Nicely: On shareable goods and the emergence of sharing as a modality of economic production. *Yale Law Journal*, 114(2), 273-358.
- Berendsen, W. (2008a). Frescobaldi (Versión 2.18) [Programa de ordenador]. Recuperado el 20 de marzo de 2015, desde <http://www.frescobaldi.org>
- Berendsen, W. (2008b). LilyKDE (Versión 0.6.6) [Programa de ordenador]. Recuperado el 20 de marzo de 2015, desde <http://code.google.com/p/lilykde>
- Berendsen, W. (2008c). LilyPond-KDE4 (Versión 0.3) [Programa de ordenador]. Recuperado el 20 de marzo de 2015, desde <http://code.google.com/p/lilykde>
- Berlanga, V., Rubio, M. J. y Vilà, R. (2013). Cómo aplicar árboles de decisión en SPSS. *REIRE: Revista d'Innovació i Recerca en Educació*, 6(1), 65-79.
- Berliner, B. y col. (1990). CVS II: parallelizing software development. En *Proceedings of the USENIX Winter 1990 technical conference* (Vol. 341, p. 352).
- Bielawa, H. (1993). Review of Sibelius 7. *Computer Music Journal*.
- Biggs, J. R. (1973). *Basic typography*. Londres: Faber.
- Blostein, D. y Haken, L. (1991). Justification of printed music. *Communications of the ACM*, 34(3), 88-99.
- Blostein, D. y Haken, L. (1994). The Lime music editor: a diagram editor involving complex translations. *Software — Practice and Experience*, 24(3), 289-306.
- Boehm, L. (1961). *Modern music notation*. Nueva York: G. Schirmer, Inc.
- Booth, W. C., Colomb, G. G. y Williams, J. M. (2001). *Cómo convertirse en un hábil investigador*. Barcelona: Gedisa.
- Borgelt, C. y Kruse, R. (2002). Induction of association rules: apriori implementation. En *15th Conference on Computational Statistics (COMPSTAT 2002, Berlin, Germany)*. Heidelberg: Physica Verlag.
- Borod, J. C., Caron, H. S. y Koff, E. (1984). Left-handers and right-handers compared on performance and preference measures of lateral dominance. *British Journal of Psychology*, (75), 177-186.
- Bouzaiene, N., Le Gall, L. y Saint-James, É. (1998). Une bibliothèque informatique pour la notation musicale baroque. En R. Hersch (Ed.), *Proceedings of the international conference on electronic publishing, document manipulation and typography*. LNCS. St. Malo (Francia).

- Brahms, J. (1927). *Concerto for piano and orchestra no.2, Op. 83* [Partitura musical compuesta originalmente en 1881] (Emil von Sauer, Ed.). Londres: Edition Peters.
- Brischoux, F. y Legagneux, P. (2009). Don't format manuscripts. *The Scientist*, 23(7), 24.
- Burkholder, J., Grout, D., Palisca, C. y Torrellas, G. (2011). *Historia de la música occidental* (8ª ed.). Alianza música. Anaya.
- Button, H. E. (1920). *System in musical notation*. London: Novello & Co.
- Byrd, D. (1974). A system for music printing by computer. *Computers and the Humanities*, 8, 161-72.
- Byrd, D. (1985). *Music notation by computer* (Tesis doctoral, Universidad de Indiana, Bloomington, Indiana).
- Byrd, D. (1994). Music notation software and intelligence. *Computer Music Journal*, 18(1), 17-20.
- Byrd, D. (2005). Music notation, music representation, and intelligence. Presentación para una charla. School of Music, Indiana University.
- Byrd, D. y Crawford, T. (2002). Problems of music information retrieval in the real world. *Inf. Process. Manage.* 38(2), 249-272.
- Byrd, D. e Isaacson, E. (2003a). A music representation requirement specification for academia. *Computer Music Journal*, 27(4), 43-57.
- Byrd, D. e Isaacson, E. (2003b). Music representation in a digital music library. En *Proceedings of the 3rd ACM/IEEE-CS joint conference on digital libraries* (pp. 234-236). JCDL '03. Houston, Texas: IEEE Computer Society.
- Caba, L. (2009). El software libre educativo. *Revista digital Ciencia y Didáctica*, (28).
- Cabero, J. y Llorente, M. (2007). Experiencias educativas mediante la aplicación de software libre.
- Calle, M. L. y Sánchez-Espigares, J. A. (2007). Árboles de clasificación y regresión en la investigación biomédica. *Medicina clínica*, 129(18), 702-706.
- de Candé, R. (1981). *Historia universal de la música*. Madrid: Aguilar.
- Candela, A. (1992). Validación de aparatos y métodos de medida: concordancia sí, correlación no. *Medicina Clínica*, 99(314).
- di Carlo, A. (1998, noviembre). Re: Documentation for new maintainers. Correo electrónico. Archivos de la lista de distribución Debian-mentors. Recuperado el 8 de septiembre de 2011, desde <http://lists.debian.org/debian-mentors>
- Cascales, B., Lucas, P., Mira, J. M., Pallarés, A. y Sánchez-Pedreño, S. (2000). *ΛT_EX, UNA IMPRENTA EN SUS MANOS*. Madrid: Aula documental de investigación.
- Castejón, P., Lechuga, M. y Martínez, Ú. (2015). *Guía práctica de estadística aplicada a la empresa y al marketing*. Madrid: Paraninfo-Universidad.
- Cataldi, L. (2003). *Édition musicale avec PMX*. 2ª ed. Manual técnico.

- Caudwell, A. H. (2010). Gource: visualizing software version control history. En *Proceedings of the ACM international conference on object oriented programming systems, languages and applications*. (pp. 73-74). ACM.
- Chacon, S. (2009). *Pro Git*. Nueva York: Apress.
- Chappell, W. (1973). *Basic typography*. Londres: Faber.
- Charter, R. A. (2003). A breakdown of reliability coefficients by test type and reliability method, and the clinical implications of low reliability. *The Journal of General Psychology*, 130(3), 290-304.
- Chen, J.-W. y Zhang, J. (2007). Comparing text-based and graphic user interfaces for novice and expert users. En *AMIA annual symposium proceedings* (pp. 125-129).
- Chlapik, H. (1987). *Die Praxis des Notengraphikers*. Doblinger.
- Christensen, C. (1997a). *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard's College.
- Christensen, R. (1997b). *Log-linear models and logistic regression*. Springer.
- Christman, S. y Pinger, K. (1997). Lateral biases in aesthetic preferences: pictorial dimensions and neural mechanisms. *Laterality: Asymmetries of Body, Brain and Cognition*, 2(2), 155-175.
- Cicchetti, D. V. y Sparrow, S. S. (1994). Guidelines, criteria and rules of thumb for evaluating normed and standardized assessment instruments in Psychology. *Psychological Assessment*, 6(4), 284-290.
- Cleveland, R. B., Cleveland, W. S., McRae, J. E. y Terpenning, I. (1990). Stl: a seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1), 3-73.
- Cleveland, W. S., Grosse, E. y Shyu, W. M. (1992). Local regression models. En J. Chambers y T. Hastie (Eds.), *Statistical models in S* (Cap. 8). Wadsworth & Brooks/Cole.
- Close, D. B., Robbins, A. D., Rubin, P. H. y Stallman, R. (1989). *The GAWK manual*. Boston: Free Software Foundation.
- Cohen, A. (1980). On the graphical display of the significant components in a two-way contingency table. En *Communications in statistics: theory and methods* (Vol. A9, pp. 1025-1041).
- ColorIuris. (2008). Sitio web de ColorIuris. Recuperado el 1 de abril de 2008, desde <http://www.coloriuris.net>
- Currás, E. (1985). *Documentación y metodología de la investigación científica*. Madrid: Paraninfo.
- Dannenber, R. B. (1989). The Canon score language. *Computer Music Journal*, 13(1).
- Dannenber, R. B. (1993). Music representation: issues, techniques, and systems. *Computer Music Journal*, 17(3).
- Davis, S. y Bostrom, R. (1992). An experimental investigation of the roles of the computer interface and individual characteristics in the learning of computer systems. *International Journal of Human-Computer Interaction*, 4(2).

- Debès, I., Schneider, M. y Malchaire, J. (2004). Etude épidémiologique des problèmes de santé des musiciens d'un orchestre symphonique. *Médecine du travail & Ergonomie*, 40(2).
- Debussy, C. (1986). *Trio in G für Klavier, Violine und Violoncello* [Partitura musical] (1ª ed.). Compuesta en 1880. Edición del autógrafo por Ellwood Derr. Munich: G. Henle Verlag.
- Deutsch, P. (2002). Ghostscript. Recuperado el 2 de agosto de 2002, desde <http://ghostscript.com>
- Devore, R., Kulma, D., Naxer, M., Clark, J. y Kuhns, K. (2010). *How music works: a first-year theory textbook*. Kent: Kent State Library Resources.
- Donato, A. (1963). *Preparing music manuscript*. Englewood Cliffs, NJ: Prentice-Hall.
- Donemus. (1982). *Uitgeven van muziek: Handleiding voor componisten en copisten*. Donemus Amsterdam.
- Droettboom, M. (2000). *Study of music notation description languages*. Inf. téc. Recuperado desde <http://gigue.peabody.jhu.edu/~mdboom/format.pdf>
- Droettboom, M. (2006). pyScore [Programa de ordenador]. SourceForge. Recuperado el 2 de abril de 2015, desde <http://sourceforge.net/projects/pyscore>
- Dunker, R. (2003). Musixlyr—a MusiXTEX extension package for lyrics handling. Manual técnico. Versión 2.1c.
- Dussek, J. L. (2012). *Sonata para piano número 3 en Do menor, Op.35* [Partitura musical] (J.-P. Coulon, Ed.). Compuesta en 1797. Autopublicada por el editor.
- Eastman Kodak Co. (1999). *Estimating luminance and illuminance with reflection-type exposure meters and an 18% neutral test card*.
- Eco, U. (1993). *Cómo se hace una tesis. Técnicas y procedimientos de investigación, estudio y escritura*. Madrid: Gedisa.
- Emerson, J. W. (1998). Mosaic displays in S-PLUS: a general implementation and a case study. *Statistical Computing and Graphics Newsletter*, 9(1), 17-23.
- Erdős, P. (1995). A selection of problems and results in combinatorics. En Matrahaza (Ed.), *Recent trends in combinatorics* (pp. 1-6). Cambridge: Cambridge Univ. Press.
- Erickson, R. F. (1975). The DARMS project: a status report. *Computing in the humanities*, 9(6), 291-298.
- Etapà, J. (2008). Montpellier releases young Villenave's opera. *Ópera Actual*, (116).
- Etapà, J. (2009). Montpellier. Opéra Comédie. Villenave: Affaire Étrangère; absolute premiere. *Ópera Actual*, (118).
- Eunjung-Cha, A. (2002). Europe's Microsoft alternative. Region in Spain abandons Windows, embraces Linux. Documento en línea. Recuperado el 3 de noviembre de 2002, desde <http://www.washingtonpost.com>
- Field-Richards, H. (1993). Cadenza: a music description language. *Computer Music Journal*, 17(4).

- Filgueiras, M. (1996). *Implementing a symbolic music processing system* (Tesis doctoral, LIACC, Universidade do Porto).
- Filgueiras, M. (1998). Implementing a symbolic music processing system. *Software — Practice and Experience*, 28(5), 493-512.
- Filgueiras, M. y Leal, J. P. (1993). Representation and manipulation of music documents in ScEX. *Electronic Publishing*, 6(4), 507-518.
- Fish, S. (2007). The Linus Torvalds bus factor. En *Shlomi Fish – essays – computing*. Recuperado desde shlomifish.org
- Fisher, R. A. y col. (1925). Applications of "Student's" distribution. *Metron*, 5(3), 90-104.
- Fisher, R. A. (1954). *Statistical methods for research workers* (12^a ed.). Edinburgh: Oliver and Boyd.
- Flores, F. (2008, agosto). Abriendo juego, abriendo mundos. Blog. No disponible actualmente. Recuperado el 3 de diciembre de 2012, desde <http://www.fernandoflores.cl>
- Fodor, B. (2010). LilyPondTool (Versión 2.12.869) [Programa de ordenador]. SourceForge. Recuperado el 4 de septiembre de 2010, desde <http://lilypondtool.organum.hu>
- Foxley, E. (1987). Music — a language for typesetting music scores. *Software — Practice and Experience*, 17(8), 485-502.
- Freedman, A. y Morrison, I. (2009). *Computer desktop encyclopedia*. Point Pleasant: The Computer Language Company Inc.
- Fried, O. (2013). *Lieder für eine oder zwei Singstimmen mit Klavierbegleitung (Canciones y dúos con acompañamiento de piano)* (A. Gurdon y U. Liska, Eds.). Nueva edición revisada. Tipografía musical de Urs Liska y Janek Warchoń. Esslingen: Sound-rel.
- Friendly, M. (1992). Graphical methods for categorical data. En *SAS user group international conference proceedings* (17, pp. 190-200).
- Friendly, M. (1994). Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, 89(425), 190-200.
- Friendly, M. (1999). Extending mosaic displays: marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8(3), 373-395.
- Frutiger, A. (2002). *En torno a la tipografía*. Barcelona: Gustavo Gili.
- Fry, B. (2008). *Visualizing data: exploring and explaining data with the Processing environment*. Sebastopol, CA: O'Reilly.
- Fuentes, M. L. y Huidobro, M. (2004, mayo). *Creación de un sistema interactivo: elementos fundamentales de la tipografía* (Tesis de Máster, Departamento de Diseño Gráfico, Escuela de Artes y Humanidades, Universidad de las Américas de Puebla, San Andrés Cholula, Puebla, México).
- Gago, I. A. (2010). ¿Por qué, para qué, utilizo Software Libre? *Comunicación y Pedagogía: nuevas tecnologías y recursos didácticos. Especial Software Libre*, (246), 31-35.

- Galera, M. (2013). Recursos para la enseñanza de la lectura musical: el uso de los editores de partituras en actividades creativas. *Espacio y Tiempo: Revista de Ciencias Humanas*, (27), 95-106.
- Le Gall, L. (1997). *Création d'une police adaptée à la notation musicale baroque* (Tesis de Máster, École Supérieure Estienne, París).
- Gamble, W. (1923). *Music engraving and printing. Historical and technical treatise*. Sir Isaac Pitman & Sons, ltd.
- Gandrud, C. (2013). *Reproducible research with R and RStudio*. CRC Press.
- García, J. (2003). *La pastilla roja*. Madrid: Edit Lin Editorial, S.L.
- García, J. (2008). Juantomás García. Documento en línea. Recuperado el 1 de junio de 2008, desde <http://www.juantomas.net>
- Gerou, T. y Lusk, L. (1996). *Essential dictionary of music notation*. Van Nuys CA: Alfred Publishing.
- Gerou, T. y Lusk, L. (2004). *Diccionario esencial de la notación musical*. Barcelona: Ma Non Troppo, Robinbook.
- Giesecking, M. (2001). *Code-basierte Generierung interaktiver Notengraphik* (Tesis doctoral, Universität Osnabrück).
- Gil, B. (1961). *Cancionero popular de Extremadura* (2ª ed.). Badajoz: Departamento de publicaciones de la Diputación Provincial.
- Gill, E. (1931). *An essay on typography*. Reimpreso en 1988; existe edición en español. Boston: David R. Godine.
- Goldfarb, C. F. (1992). SMDL, standard musical description language. *Cover pages*, (ISO/IEC DIS 10743).
- Gomberg, D. A. (1975). *A computer-oriented system for music printing* (Tesis doctoral, Universidad de Washington).
- Gomberg, D. A. (1977). A computer-oriented system for music printing. *Computing and the Humanities*, 11(3), 63-80.
- Gómez Deck, D. (2005). Squeak: construyendo la libertad. En M. A. Segovia (Ed.), *Libro blanco del software libre en España* (Vol. 2). Sociedad de la Información. Junta de Extremadura.
- Gómez, A. (2008). *Estadística básica con R y R-Commander*. Cádiz: Universidad de Cádiz, Servicio de Publicaciones.
- González, C., Lise, A. y Felpeto, A. (2013). *Tratamiento de datos con R, Statistica y SPSS*. Madrid: Díaz de Santos.
- González-Barahona, J., Robles, G., Herraiz, I. y Ortega, F. (2014). Studying the laws of software evolution in a long-lived FLOSS project. *Journal of Software: Evolution and Process*, 26(7), 589-612.

- González-Barahona, J., Seoane, J. y Robles, G. (2003). *Introducción al Software Libre*. Fundació per a la Universitat Oberta de Catalunya.
- Goossens, M., Mittelbach, F., Rahtz, S., Roegel, D. y Voß, H. (1997). *The L^AT_EX graphics companion: illustrating documents with T_EX and PostScript*. Addison-Wesley.
- Goossens, M., Mittelbach, F. y Samarin, A. (1994). *The L^AT_EX companion*. Reading, MA: Addison-Wesley.
- Gorhan, F., Hettinger, J., Schulz, J. y Wolter, M. (2012). *Development of a model evaluating the maturity of open source software* (Tesis de Máster, Cooperative State University Stuttgart. School of Business, Stuttgart).
- Gould, E. (2011). *Behind bars: the definitive guide to music notation*. Faber Edition Series. Londres: Faber Music.
- Gourlay, J. S. (1986). A language for music printing. *Communications of the ACM*, 29(5), 388-401.
- Gourlay, J. S. (1987). *Spacing a line of music*, (inf. téc. N° OSU-CISRC-10/87-TR35). Department of Computer and Information Science, The Ohio State University.
- Gourlay, J. S., Parrish, A., Roush, D., Sola, F. y Tien, Y. (1987). *Computer formatting of music* (inf. téc. N° OSU-CISRC-2/87-TR3). Department of Computer and Information Science, The Ohio State University.
- Graham, R., Knuth, D. E. y Patashnik, O. (1988). *Concrete mathematics* (1^a ed.). Reading, MA: Addison-Wesley.
- Mc Grain, M. (1991). *Music notation*. Hal Leonard Publishing Corporation.
- Gram, C. y Cockton, G. (1996). *Design principles for interactive software*. IFIP Advances in Information and Communication Technology. Springer.
- Grande, C. (1995). NIFF6a notation interchange file format.
- Grande, C. (1997). The notation interchange file format: a Windows-compliant approach. En *Beyond MIDI: the handbook of musical codes* (pp. 491-512). Cambridge: MIT Press.
- Greaves, D., Hamano, J., Torvalds, L. y col. (2010). Git documentation.
- Grøver, J. (1989a). *A computer-oriented description of music notation. Part I: the symbol inventory* (inf. téc. N° 133). Department of informatics, University of Oslo.
- Grøver, J. (1989b). *A computer-oriented description of music notation. Part II: two voices sharing a staff, leger line rules, dot positioning* (inf. téc. N° 134). Department of informatics, University of Oslo.
- Grøver, J. (1989c). *A computer-oriented description of music notation. Part III: accidental positioning* (inf. téc. N° 135). Department of informatics, University of Oslo.
- Gruenbaum, L. (2014). The Samchillian Tip Tip Tip Cheeepreeeee: a relativistic keyboard instrument. Documento en línea. Recuperado el 6 de noviembre de 2014, desde <http://www.samchillian.com/nimefinal.pdf>

- Guerrero, F. (1584). *Liber vesperarum Francisco Guerrero Hispalensis Ecclesiae magistro auctore*. Roma: A. Gardano y D. Basa.
- Guerrero, F. (2003). *Ibant Magi (Hostis Herodes impie)* [Partitura musical] (F. G. Jácome, Ed.). Primera edición de la obra original: (Guerrero, 1584). Lugo: Publicaciones de la Coral «Sólo Voces».
- Guisande, C. (2006). *Tratamiento de datos*. Madrid: Díaz de Santos.
- Gutiérrez, J. (2007). *Estadística general aplicada*. Colección Académica. Fondo editorial Universidad EAFIT.
- Hader, K. (1948). *Aus der Werkstatt eines Notenstechers*. Vienna: Waldheim–Eberle Verlag.
- Haken, L. y Blostein, D. (1993). The Tilia music representation: extensibility, abstraction, and notation contexts for the Lime music editor. *Computer Music Journal*, 17(3), 43-58.
- Haken, L. y Blostein, D. (1995, septiembre). A new algorithm for horizontal spacing of printed music. En *International computer music conference* (pp. 118-119). Banff.
- Hardings, J. (2004). *Seminario de software libre*. Departamento de ciencias de la computación, Universidad de Chile.
- Hartigan, J. A. y Kleiner, B. (1981). Mosaics for contingency tables. En *Computer science and statistics: Proceedings of the 13th symposium on the interface* (pp. 268-273). Springer.
- Healy, K. y Schussman, A. (2003). The ecology of open-source software development [Documento de trabajo].
- Hegazy, W. A. (1987). *On the implementation of the MusiCopy language processor*, (inf. téc. N° OSU-CISRC-10/87-TR34). Department of Computer and Information Science, The Ohio State University.
- Hegazy, W. A. y Gourlay, J. S. (1987). *Optimal line breaking in music* (inf. téc. N° OSU-CISRC-8/87-TR33). Department of Computer and Information Science, The Ohio State University.
- Hernández, J., Ramírez, M. y Ferri, C. (2004). *Introducción a la minería de datos*. Madrid: Pearson Educación, S.A.
- Heussenstamm, G. (1987). *The Norton Manual of music notation*. Nueva York: Norton.
- Hewlett, W. B. y Selfridge-Field, E. (1985). *Directory of computer assisted research in musicology*. Menlo Park, California: Center for Computer Assisted Research in the Humanities.
- Hewlett, W. B. y Selfridge-Field, E. (Eds.). (2001). *The virtual score; representation, retrieval and restoration*. Computing in Musicology. Cambridge: MIT Press.
- Ho, D. (2007). Notepad++ (Versión 5.0.3) [Programa de ordenador]. SourceForge. Recuperado el 20 de marzo de 2015, desde <http://sourceforge.net/projects/notepad-plus>
- Hofmann, H., Siebes, A. P. y Wilhelm, A. F. (2000). Visualizing association rules with interactive mosaic plots. En *Proceedings of the sixth ACM SIGKDD international conference on*

- knowledge discovery and data mining* (pp. 227-235). ACM.
- Hofstadter, D. R. (1982). Metamagical Themas. The music of Frédéric Chopin: startling aural patterns that also startle the eye. *Scientific American Magazine*, 246(4), 16-28.
- Hoos, H. H., Hamel, K. A., Renz, K. y Kilian, J. (1998). The GUIDO music notation format: a novel approach for adequately representing score-level music. En *Proceedings of international computer music conference* (pp. 451-454).
- Hoos, H. H., Renz, K. y Görg, M. (2001, octubre). GUIDO/MIR – an experimental musical information retrieval system based on GUIDO music notation. En *Proceedings of the 2nd annual international symposium on music information retrieval* (pp. 41-50). Bloomington, Indiana.
- Houston, J. (2005). *The stand* [Partitura musical]. Sidney: Hillsnog Publishing.
- Huginin, J. (1997). Python and java: the best of both worlds. En *Proceedings of the 6th international python conference* (Vol. 9, pp. 2-18).
- Huh, M. Y. (2004). Line mosaic plot: algorithm and implementation. En *COMPSTAT 2004: Proceedings in computational statistics* (pp. 277-285). Springer.
- Huron, D. (1993). The Humdrum toolkit: software for music researchers [Programa de ordenador].
- Hyndman, R. J. y Athanasopoulos, G. (2013). *Forecasting: principles and practice*. OTexts.
- Hyndman, R. J. y Khandakar, Y. (2007). *Automatic time series for forecasting: the forecast package for R*. Monash University, Department of Econometrics y Business Statistics.
- Ignatzek, K. (1995). *Die Jazzmethode für Klavier I. Voicings, Akkorde, Begleitung*. Mainz: Schott.
- ISK (Interkerkelijk Stichting voor het Kerklied) (Ed.). (1973). *Liedboek voor de Kerken*. 491 canciones para coro y órgano. Zoetermeer (Holanda): Boekencentrum Uitgevers.
- ISK (Interkerkelijk Stichting voor het Kerklied) (Ed.). (2013). *Nieuwe Liedboek voor de Kerken*. Nueva edición. Zoetermeer (Holanda): BV Liedboek.
- JabRef Development Team. (2015). *JabRef* [Programa de ordenador].
- Jaschinski, A. (Ed.). (2000). *Notation*. Kassel: Bärenreiter Verlag.
- Jofré, J. (2003). *El lenguaje musical*. Barcelona: Ma non Troppo, Robinbook.
- Johnson, H. (1946). *How to write music manuscript*. Nueva York: Carl Fischer, Inc.
- Kane, J. (2005). *Manual de tipografía*. Barcelona: Gustavo Gili, S.A.
- Knauff, M. y Nejasmic, J. (2014). An efficiency comparison of document preparation systems used in academic research and development. *PloS one*, 9(12).
- Knopke, I. (2008, septiembre). The Perlhumdrum and Perllilypond toolkits for symbolic music information retrieval. En *Proceedings of the 9th international conference on music information retrieval* (pp. 147-152). Filadelfia.
- Knuth, D. E. (1979). Mathematical typography. *Bulletin of the American Mathematical Society (new series)*, 1(3), 337-372. Publicado también como un informe CS de Stanford, STAN-

CS-78-648.

- Knuth, D. E. (1992). Two notes on notation. *Amer. Math. Monthly*, 99, 403-422.
- Könemann (Ed.). (1994). *Edward Grieg: Lyric Pieces for piano*. Köln: Könemann Music Budapest.
- Kopka, H. y Daly, P. W. (1995). *A guide to L^AT_EX 2_ε: document preparation for beginners and advanced users* (2^a ed.). Addison-Wesley.
- Kostylo, J. (2008). Commentary on Ottaviano Petrucci's music printing patent (1498). En L. Bently y M. Kretschmer (Eds.), *Primary sources on copyright (1450-1900)*.
- Ladiray, D. y Quenneville, B. (2012). *Seasonal adjustment with the X-11 method*. Springer Science & Business Media.
- Lagrange, M., Percival, G. y Tzanetakis, G. (2007, septiembre). Adaptive harmonization and pitch correction of polyphonic audio using spectral clustering. En *International conference on digital audio effects*. Burdeos, Francia.
- Lamb, A. (1996). The University of Colorado music engraving page. No disponible actualmente. Recuperado el 1 de octubre de 2007, desde cc.colorado.edu
- Lamport, L. (1994). *L^AT_EX: a document preparation system. User's guide and reference manual* (2^a ed.). Reading, MA: Addison-Wesley.
- Lancho, J. (2012). La cláusula mariposa [Sitio web en formato de blog]. Recuperado el 12 de junio de 2013, desde clausulamariposa.es
- de Landsheere, G. (1986). *La recherche en éducation dans le monde*. París: Presses universitaires de France.
- Langston, P. S. (1990). Unix music tools at Bellcore. *Software—Practice and Experience*, 20(S1), 47-61.
- Laurie, D. (2005). *M-Tx: music from text*. Manual técnico. Versión 0.60. Oak Ridge National Laboratory.
- Lehman, P. y col. (2006). The Bibl_AT_EX package [Programa de ordenador].
- Lehmann, J. y Wobst, A. (2015). PyX documentation. Documento en línea. SourceForge. Recuperado el 15 de febrero de 2015, desde <http://pyx.sourceforge.net>
- Lerdorf, R., Tatroe, K. y MacIntyre, P. (2006). *Programming PHP*. Sebastopol, CA: O'Reilly Media, Inc.
- Levy, J. (2002). Lateral dominance and aesthetic preference. *Neuropsychologia*, 14(4), 431-445.
- Ley 23/2006, de 7 de julio, por la que se modifica el texto refundido de la Ley de Propiedad Intelectual, aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril. (2006). BOE núm. 162, de 8 de julio de 2006.
- Liebowitz, S. y Watt, R. (2006). How to best ensure remuneration for creators in the market for music? Copyright and its alternatives. *Journal of Economic Surveys*, 20(4), 513-545.

- Lincoln, H. B. (1977). Encoding, decoding and storing melodies for a data base of renaissance polyphony: a progress report. En *VLDB '77: proceedings of the third international conference on very large data bases* (pp. 277-282). Tokyo, Japan: VLDB Endowment.
- Lingas, A. (2014). *Chant in the orthodox tradition*. School of Arts. City University London.
- Linton, O. B. (2008). *Local regression models*. Palgrave Macmillan.
- Lizarán, I. J. (2010). Edición de partituras con software libre. *Papeles del Festival de música española de Cádiz*, (5), 99-121.
- Loeliger, J. (2009). *Version control with Git: powerful tools and techniques for collaborative software development*. Sebastopol, CA: O'Reilly.
- López García, J. L. (1977). *Conjunto coral. Cuestionario teórico adaptado a la asignatura de conjunto coral*. Murcia: Editorial Desconocida.
- López, J., Ros, J., Orera, L., Fernández, M. T. y Prat, J. (2005). *Las tesis doctorales. Producción, evaluación y defensa*. Madrid: Fragua.
- López-Sintas, J., Cebollada, À., Filimon, N. y Gharhaman, A. (2014). Music access patterns: a social interpretation. *Poetics*, 46, 56-74.
- Lorenzo, J. (2007). *Estadística descriptiva*. Madrid: Paraninfo.
- Loy, G. y Abbott, C. (1985). Programming languages for computer music synthesis, performance, and composition. *ACM Comput. Surv.* 17(2), 235-265.
- Lozano, D., Bello, J., Cancino, J. y Martínez, P. (2013). Lilypond, software accesible para la edición de partituras por parte de músicos ciegos. En *Música, inclusión y discapacidad visual. guía de orientaciones didácticas*. Bogotá: Ministerio de Cultura. República de Colombia.
- Luidl, P. (2004). *Tipografía básica*. Campgràfic editors, s.l.
- MacLachlan, J. (1998). *Learn guitar: from beginner to your 1st band*. Beginners Series. R&C Gregory Publishing.
- Manaris, B., Vaughan, D., Wagner, C., Romero, J. y Davis, R. (2003). Evolutionary music and the Zipf-Mandelbrot law: developing fitness functions for pleasant music. En S. Cagnoni, C. Johnson, J. Cardalda, E. Marchiori, D. Corne, J.-A. Meyer, ... y E. Hart (Eds.), *Applications of evolutionary computing* (Vol. 2611, pp. 522-534). Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Mariano, A., Nienhuys, H.-W. y Nieuwenhuizen, J. (1999). Mudela 1.0.7 / LilyPond 1.1.8 reference manual. January.
- Márquez, P. (2007). Distribuciones GNU/Linux de las comunidades autónomas. Documento en línea. No disponible actualmente. Recuperado el 14 de julio de 2007, desde <http://observatorio.cnice.mec.es>
- Martel, P. y Vegas, F. (1996). *Probabilidad y estadística matemática: aplicaciones en la práctica clínica y en la gestión sanitaria*. Madrid: Díaz de Santos.

- Martin, G. (1998). *Music notation font design: technical specifications for note head font design*. Seattle, Washington: Sunhawk Corporation.
- Mas i Hernández, J. (2005). *Software libre: técnicamente viable, económicamente sostenible y socialmente justo*. Barcelona: Infonomía.
- Mele, G. (2004). Los orígenes de la imprenta musical. *Goldberg Magazine*, (12).
- Meyer, D., Zeileis, A. y Hornik, K. (2005). *The strucplot framework: visualizing multi-way contingency tables with vcd* (inf. téc. N° 22). Department of Statistics y Mathematics, Wirtschaftsuniversität Wien. Research Report Series. Viena.
- Microsoft Corporation. (2008). Documentos, términos de uso. Documento en línea. Recuperado el 2 de octubre de 2008, desde <http://download.microsoft.com/documents>
- Miller, J. M. y Penfield, R. D. (2006). Using the Score method to construct asymmetric confidence intervals: an SAS program for content validation in scale development. *Behavior Research, Methods, Instruments, and Computers*, (37), 209-212.
- Molina, E. y Pérez-Ribes, J. (1995). *Estudios y obras para guitarra* [Partitura musical]. Prólogo de Carmelo Solís. Badajoz: Departamento de publicaciones de la Diputación Provincial.
- Montel, D. (1997). La gravure de la musique, lisibilité esthétique, respect de l'œuvre. En Genevois & Orlarey (Ed.), *Musique & notations*. Lyon.
- Montero, I. y León, O. G. (2007). A guide for naming research studies in psychology. *International Journal of Clinical and Health Psychology*, 7(3), 847-862.
- Moody, G. (2002). *Rebel Code: The Inside Story of Linux and the Open Source Revolution*. Perseus Pub.
- Müller, G. (1990). *Interaktive Bearbeitung konventioneller Musiknotation* (Tesis doctoral, Eidgenössischen Technischen Hochschule Zürich).
- Murphy, J. (1999). *Technical analysis of the financial markets*. Nueva York: Prentice Hall Press.
- Music Publisher's Association. (1996, diciembre). *Standard music notation specifications for computer programming*. MPA.
- Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5), 323-351.
- Nienhuys, H.-W. (2006). LilyPond, automated music formatting and the art of shipping. En *Forum internacional software livre (FISL 7.0)*. Porto Alegre.
- Nienhuys, H.-W. y Nieuwenhuizen, J. (1995, abril). LilyPond. Código fuente de un programa de ordenador en forma de repositorio de versiones público. Copias descentralizadas. Recuperado el 30 de septiembre de 2014, desde <http://git.sv.gnu.org/r/lilypond.git>
- Nienhuys, H.-W. y Nieuwenhuizen, J. (2003, mayo). LilyPond, a system for automated music engraving. En *Actas* (pp. 167-172). XIV Colloquium on Musical Informatics (CIM). Florencia.

- Nienhuys, H.-W., Nieuwenhuizen, J. y Percival, G. (2015a). *LilyPond: manual de aprendizaje, versión 2.19.18* (F. Vila, Trad.).
- Nienhuys, H.-W., Nieuwenhuizen, J. y Percival, G. (2015b). *LilyPond: referencia de la notación, versión 2.19.18* (F. Vila, Trad.).
- Nienhuys, H.-W., Nieuwenhuizen, J. y Percival, G. (2015c). *LilyPond: utilización del programa, versión 2.19.18* (F. Vila, Trad.).
- Nieuwenhuizen, J. (2009, febrero). Re: OT: Some git statistics. Correo electrónico. Archivos de GNU de la lista de distribución LilyPond-devel. Recuperado el 29 de septiembre de 2009, desde <http://lists.gnu.org>
- Núñez, M. (2009). El conocimiento libre en la danza: danza 2.0. *Danzaratte: Revista del Conservatorio Superior de Danza de Málaga*, (6), 63-69.
- Ogawa, M. y Ma, K.-L. (2009). Code_swarm: a design study in organic software visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1097-1104.
- Ogawa, M. y Ma, K.-L. (2010). Software evolution storylines. En *Proceedings of the 5th international symposium on software visualization* (pp. 35-42). ACM.
- Oliva, A. (2007). Free software and The Matrix. En *7mas. jornadas regionales de software libre*. Universidad Nacional de Córdoba. Córdoba, Argentina.
- de la Orden, L. (2013). Espadistros. GNU/Linux hecho en España. Documento en línea. Recuperado el 25 de enero de 2013, desde <http://espadistros.wordpress.com>
- Organización Mundial de la Propiedad Intelectual. (1967, 14 de julio). Convenio que establece la Organización Mundial de la Propiedad Intelectual.
- Organización Mundial de la Propiedad Intelectual. (1989, octubre). Convenio de Berna para la protección de las obras literarias y artísticas. Acta de París del 24 de julio de 1971 y enmendada el 2 de octubre de 1979. Ginebra.
- Organización Mundial de la Propiedad Intelectual. (1994). *Glosario de derechos de autor y derechos conexos*. Ginebra: World Intellectual Property Organization.
- Ornstein, S. M. y Turner, J. (1983). *Mockingbird: a composer's amanuensis* (inf. téc. N° CSL-83-2). Xerox Palo Alto Research Center. Palo Alto, CA.
- Ornstein, S. M. y Turner, J. (1984). Mockingbird: a composer's amanuensis. *Byte*, 9(1).
- Page, S. D. (1988). *Computer tools for music information retrieval* (Tesis doctoral, Universidad de Oxford).
- Pangert, R. y Loock, F. (Eds.). (2004). *Musikermmedizin, Musikerarbeitsplätze. Eine Einführung für Orchestermusiker, Musikpädagogen und Studenten*. Normativa sobre seguros de accidente en músicos. München: IAS Institut für Arbeits- und Sozialhygiene Stiftung Gesetzliche.
- Parrish, A. y Gourlay, J. S. (1987). *Computer formatting of musical simultaneities* (inf. téc. N° OSU-CISRC-10/87-TR28). Department of Computer and Information Science, The Ohio State

University.

- Parrish, A., Hegazy, W. A., Gourlay, J. S., Roush, D. K. y Sola, F. J. (1987). *MusiCopy: an automated music formatting system* (inf. téc. N° OSU-CISRC-10/87-TR29). Department of Computer and Information Science, The Ohio State University.
- Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302), 157-175.
- Penfield, R. D. y Giacobbi, P. R. (2004). Applying a score confidence interval to Aiken's item content-relevance index. *Measurement in Physical Education and Exercise*, 8(4), 213-225.
- Percival, G. (2008, agosto). *Computer-assisted musical instrument tutoring with targeted exercises* (Tesis de Máster, Universidad de Victoria, Victoria, Canadá).
- Percival, G. (2010). Sustainability in F/OSS: developers as a non-renewable resource. En *Rencontres mondiales du logiciel libre*. Burdeos, Francia.
- Percival, G., Anders, T. y Tzanetakis, G. (2008, agosto). Generating targeted rhythmic exercises for music students with constraint satisfaction programming. En *ICMC08: Proceedings of the 2008 international computer music conference*. Belfast.
- Percival, G., Wang, Y. y Tzanetakis, G. (2007). Effective use of multimedia for computer-assisted musical instrument tutoring. En *Emme '07: Proceedings of the international workshop on educational multimedia and multimedia education* (pp. 67-76). Nueva York: ACM Press.
- Pérez, E. y Sánchez-Crespo, A. (2006). *La validez jurídica de la licencia GPL en España* (2ª ed.). Sánchez-Crespo Abogados y Consultores, S.L.
- Pérez, E. y Sánchez-Crespo, A. (2009). *La validez jurídica de la licencia GPLv3 en el marco normativo español de los derechos de autor y otras licencias "opensource"*. Sánchez-Crespo Abogados y Consultores, S.L.
- Pérez-Ribes, J. (1989). *Villancicos y canciones extremeñas*. Badajoz: Departamento de publicaciones de la Diputación Provincial.
- Pestana, P. y Pestana, D. (2012). Philip Glass' Façades—a case study on the complexity of music scores. En *Proceedings of the 5th chaotic modeling and simulation international conference, athens* (pp. 459-467).
- Petrucchi, O. (2001). *Harmonice musices odhecaton A*. Critical facsimiles. Facsímil de la tercera edición de 1504. Nueva York: The Broude Trust.
- Pindyck, R. S. y Rubinfeld, D. L. (1998). *Econometric models and economic forecasts*. Boston: Irwin/McGraw-Hill.
- Powell, S. (2007). *Music engraving today* (2ª ed.). Brichtmark Music.
- Prather, R. E. (1996). Harmonic analysis from the computer representation of a musical score. *Communications of the ACM*, 39(12), 239.

- Presnell, B. (2000). *An introduction to categorical data analysis using R*. Reproduce los ejemplos de (Agresti, 2007).
- Proposición de Ley de modificación del Texto Refundido de la Ley de Propiedad Intelectual, de forma que se reconozca al Autor el derecho de explotación de su obra bajo las denominadas Licencias Libres. (2011). BOCG, núm. 18-1, 28 de diciembre de 2011. Presentada por la Asamblea de Extremadura.
- Puig, M. (2003). Discogràfiques a catalunya. la mà de guido. *Catalunya música: Revista musical catalana*, (222), 42-43.
- Purchase, H. C., Allder, J.-A. y Carrington, D. (2002). Graph layout aesthetics in UML diagrams: user preferences. *Journal of Graph Algorithms and Applications*, 6(3), 255-279.
- R Core Development Team. (2000). *An introduction to R*.
- Racine, J. S. (2012). RStudio: a platform-independent IDE for R and Sweave. *Journal of Applied Econometrics*, 27(1), 167-172.
- Rader, G. M. (1996). Creating printed music automatically. *Computer*, 29(6), 61-69.
- Ramos, A. (1991). *Tu segundo manual del Atari ST*. Aarhus: Indigo 2.
- Raskin, J. (1997). An early system for printing music: Lingua Musica pro Machinationibus IV (1966/67). En E. Selfridge-Field (Ed.), *Beyond MIDI: the handbook of musical codes*. Cambridge: MIT Press.
- Rastall, R. (1983). *The notation of western music: an introduction*. J. M. Dent & Sons London.
- Rauterberg, M. (1992). An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. *Behaviour & information technology*, (11).
- Ray, S. F. (2000). *Camera exposure determination* (9ª ed.) (R. E. Jacobson, S. F. Ray, G. G. Atteridge y N. R. Axford, Eds.). Oxford: Focal Press.
- Raymond, E. S. (1999). *The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary*. Sebastopol, CA: O'Reilly Media.
- Read, G. (1969). *Music notation: a manual of modern practice* (2ª ed.). Crescendo book. Nueva York: Allyn and Bacon.
- Read, G. (1978). *Modern rhythmic notation*. Bloomington: Indiana University Press.
- Real Academia Española. (2003). *Diccionario de la lengua española* (22ª ed.). Edición en 2 volúmenes, formato 13 x 19. Espasa-Calpe.
- Reas, C. y Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Cambridge: MIT Press.
- Reinach, T. y Agostini, E. (1984). *La música griega*. Centro de Investigaciones Sociales.
- Renz, K. (2002). *Algorithms and data structures for a music notation system based on GUIDO music notation* (Tesis doctoral, Universität Darmstadt).

- Ribeiro, G. F. (2010). *Usabilidade de interfaces com o usuário de softwares de edição e edição de partituras* (Tesis de post-graduado de Ingeniería de Sistemas, Escola Superior Aberta Do Brasil).
- Richter, C. F. (1958). *Elementary seismology*. (Publicado también por Bailey Bros. & Swinfen Ltd., London). San Francisco: W. H. Freeman & Co.
- Roads, C. (1979). Grammars as representations for music. *Computer Music Journal*, 3(1).
- Robine, M., Percival, G. y Lagrange, M. (2007, agosto). Analysis of saxophone performance for computer-assisted tutoring. En *ICMC07: Proceedings of the international computer music conference* (Vol. 2, pp. 381-384). Copenhagen.
- Robles, G., Arjona, L., Serebrenik, A., Vasilescu, B. y González-Barahona, J. (2014). FLOSS 2013: a survey dataset about free software contributors: challenges for curating, sharing, and combining. En *11th Working conference on mining software repositories (MSR)* (pp. 396-399).
- Rochkind, M. J. (1975). The source code control system. *IEEE transactions on software engineering*, 1(4), 364-370.
- Roelofs, R. (1991). *Een Geautomatiseerd Systeem voor het Afdrukken van Muziek* ("Un sistema automático para imprimir música") (Tesis de Máster, Erasmus Universiteit Rotterdam).
- Roemer, C. (1984). *The art of music copying* (2ª ed.). Descatalogado. Sherman Oaks (CA): Rerrick music Co.
- Rojo, J. M. (2006). *Árboles de clasificación y regresión*. Laboratorio de Estadística. Instituto de Economía y Geografía. Consejo Superior de Investigaciones Científicas. Madrid.
- Rosecrans, G. (1979). *Music notation primer*. Nueva York: Passantino.
- Rosenthal, C. A. (1967). *A practical guide to music notation*. Nueva York: MCA Music.
- Ross, T. (1987). *Teach yourself the art of music engraving and processing*. Miami, Florida: Hansen House.
- Rossetto, L. (2013). *Enea nel Lazio, opera riformata: prima l'azione, poi le parole* (Tesis doctoral, Università degli Studi di Torino, Facoltà di Lettere e Filosofia).
- van Rossum, G. (1991). Python documentation. Documento en línea. Recuperado el 10 de marzo de 2015, desde <http://docs.python.org>
- Rothstein, J. (1993). Review of Passport Designs' Encore music notation software. *Computer Music Journal*, 17(2), 104-107.
- Roush, D. K. (1987). *Using MusiCopy* (inf. téc. N° OSU-CISRC-18/87-TR31). Department of Computer and Information Science, The Ohio State University.
- Rubenstein, W. B. (1987). A database design for musical information. En *SIGMOD '87: proceedings of the 1987 ACM SIGMOD international conference on management of data*. Nueva York: ACM.
- Sádaba, I. (2008). *Propiedad intelectual: ¿bienes públicos o mercancías privadas?* Los libros de la Catarata.

- Salazar, A. (1989). *La música en la sociedad europea*. Madrid: Alianza Música.
- Salcedo, J., Anduiza, E. y col. (2010). *Dilemas de acción colectiva* (Tesis de Máster, Departamento de Ciencia Política y Derecho Público. Universidad Autónoma de Barcelona).
- Sandberg, E. (2006, marzo). *Separating input language and formatter in GNU LilyPond* (Tesis de Máster, Universidad de Uppsala).
- Sanguino, J. (1997). *Iniciación a \LaTeX 2 ϵ , un sistema para preparar documentos*. Madrid: Addison-Wesley.
- Schirmer, G. (2001). *The G. Schirmer manual of style and usage*. Nueva York: The G. Schirmer Publications Department.
- Schott (Ed.). (1973). *Frédéric Chopin: 24 Préludes Op.28*. Viena: Wiener Urtext Edition (Schott / Universal Edition).
- Schubert, F. (1894). Im Haine, Op.56 No.3, D738 [Partitura musical]. En *Franz Schubert's Werke*. XX: Sämtliche Lieder und Gesänge, No.415. Compuesta en ca.1822. Publicada originalmente en 1826. Leipzig: Breitkopf & Härtel.
- Schubert, F. (1933). *Study scores* [Partitura musical]. Reimpresión de (Schubert, 1894). Nueva York: Edwin F. Kalmus.
- Schweer, W. y col. (2012). MuseScore [Programa de ordenador].
- Selfridge-Field, E. (Ed.). (1997). *Beyond MIDI: the handbook of musical codes*. Cambridge: MIT Press.
- Sheahan, M. y Chubb, P. (2009). *The amazing penguin-powered robotic clarinet*. Sydney: Department of Communications, Information Technology and the Arts.
- Shinn, M. (2013). *Instant MuseScore*. Packt Publishing Ltd.
- Siebra, B., Anjos, E. y Rolim, G. (2014). Computational science and its applications. (Cap. Study on the Social Impact on Software Architecture through Metrics of Modularity, Vol. 8583, pp. 618-632). ICCSA.
- Sierra Bravo, R. (2007). *Tesis doctorales y trabajos de investigación científica. Metodología general de su elaboración y documentación*. Madrid: Paraninfo.
- Simkiss, D., Ebrahim, G. J. y Waterston, A. J. R. (2012). *Analysing categorical data: log-linear analysis*. Research methods II: Multivariate analysis. Journal of Tropical Pediatrics.
- Simons, D. (2004). PMX – a preprocessor for MusiX \TeX . Recuperado el 14 de febrero de 2015, desde <http://ctan.org/pub/support/pmx>
- Sinclair, J. y Gilmour, L. (2002). *Collins concise dictionary and concise thesaurus*. Collins.
- Sinclair, S., Droettboom, M. y Fujinaga, I. (2006). Lilypond for pyScore: approaching a universal translator for music notation. En *Proceedings of the 7th international conference on music information retrieval*. Victoria, Canadá.
- Sloan, D. (1993). Aspects of music representation in HyTime/SMDL. *Computer Music Journal*, 17(4).

- Sloboda, J. A. (1984). Experimental studies of music reading: a review. *Music Perception: An Interdisciplinary Journal*, 2(2), 222-236.
- Smith, L. (1967). Score (Versión 2.12.869) [Programa de ordenador]. Westminster, CA: San Andreas Press. Recuperado el 21 de septiembre de 2014, desde <http://www.scoremus.com>
- Smith, L. (1973). *Editing and printing music by computer*.
- Smithers, T. (2000). *Designing a font to test a theory*. Artificial Intelligence in Design '00. Springer.
- Sola, F. (1987). *Computer design of musical slurs, ties and phrase marks* (inf. téc. N° OSU-CISRC-10/87-TR32). Department of Computer and Information Science, The Ohio State University.
- Sola, F. y Roush, D. (1987). *Design of musical beams* (inf. téc. N° OSU-CISRC-10/87-TR30). Department of Computer and Information Science, The Ohio State University.
- Solomon, J. (1984). Towards a history of tonoi. *The journal of musicology*. Citado por Burkholder, Grout, Palisca y Torrellas (2011).
- de Souza, M. M. (2012). *Os aspectos poético-musicais nas obras de Homero: métrica, ritmo e performance (Séc. VIII a. C.)*. (Tesis de Máster, Universidade federal de Goiás. Faculdade de história, Goiânia).
- Spiegel, M. R. (1991). *Estadística* (2ª ed.). Madrid: McGraw-Hill.
- Spiegel, M. R., Schiller, J. y Srinivasan, R. A. (2007). *Probabilidad y estadística* (2ª ed.). México D.F.: McGraw-Hill.
- Stallman, R. (1998). *Free software, free society* (J. Gay, Ed.). Boston, Massachusetts: GNU press.
- Stallman, R. (2004). *Software libre para una sociedad libre*. Madrid: Traficante de sueños.
- Stallman, R. (2007, 7 de febrero). Conferencia Internacional de Software Libre 3.0. Badajoz.
- Stallman, R. y col. (2008). Páginas de filosofía del proyecto GNU. Recuperado desde <http://gnu.org/philosophy>
- Steinbach, A. y Schofer, A. (1987). *Automatisierter Notensatz mit T_EX* (Tesis de Máster, Rheinische Friedrich-Wilhelms Universität, Bonn).
- Stephany, F., Mens, T. y Gîrba, T. (2009). Maispion: a tool for analysing and visualising open source software developer communities. En *Proceedings of the international workshop on smalltalk technologies* (pp. 50-57). ACM.
- Stone, K. (1980). *Music notation in the Twentieth Century*. Nueva York: Norton.
- Swann, A. (1990). *Bases del diseño gráfico*. Barcelona: Gustavo Gili, S.A.
- Swicegood, T. (2008). *Pragmatic version control using Git*. The pragmatic starter kit. Frisco, TX: The pragmatic programmers, LLC.
- Tabachnick, B. G. y Fidell, L. S. (2007). *Using multivariate statistics* (5ª ed.). Nueva York: Allyn and Bacon.

- Tantau, T. (2015). The TikZ and PGF Packages. Documento en línea. SourceForge. Recuperado el 15 de marzo de 2015, desde <http://pgf.cvs.sourceforge.net>
- Taupin, D. (1993). Music \TeX : Using \TeX to write polyphonic or instrumental music. *TUGboat*, 14(3), 212-220.
- Taupin, D., Mitchell, R. y Egler, A. (1995). Musix \TeX : l'écriture de la musique polyphonique avec \TeX . *Cahiers GUTenberg*, 21(6).
- Tejedor, F. J. (1999). *Análisis de varianza*. Madrid: La Muralla, S.A.
- The Gale Group, I. (Ed.). (1998). *How products are made: how is sheet music made?: An illustrated guide to product manufacturing*. Gale / Cengage Learning.
- Therneau, T. M., Atkinson, E. J. y col. (2014). *An introduction to recursive partitioning using the RPART routines* (inf. téc. N° 61). Mayo Foundation.
- Tichy, W. (1982). Design, implementation, and evaluation of a revision control system. En *6th international conference on software engineering*. Tokyo.
- Tichy, W. (1985). RCS — a system for version control. *Software — Practice and Experience*, 15(7), 637-654.
- Torres, A. (2013). *A tecnoutopia do software livre: uma história do projeto técnico e político do GNU* (Tesis de Máster, Universidade de São Paulo. Faculdade de Filosofia, Letras e Ciências Humanas, São Paulo).
- Torvalds, L. y Hamano, J. (2005). Git: fast version control system. [Software de ordenador. Sistema de control de versiones]. Recuperado el 30 de enero de 2015, desde <http://git-scm.com>
- Trier, M. y Thiel, S. (2009). GitPython [Biblioteca de funciones de software]. Recuperado el 10 de septiembre de 2009, desde gitpython.readthedocs.org
- Trondheim, L. (2008). Web oficial de Lewis Trondheim. Documento en línea. Recuperado el 20 de febrero de 2009, desde <http://lewistrondheim.com>
- Tyboni, B. (1994). *Noter handbok i traditionell notering*. Stockholm: Gehrman's Musikförlag.
- Uitdenbogerd, A. L. y Yap, Y. W. (2003, octubre). Was Parsons right? An experiment in usability of music representations for melody-based music retrieval. En *Proceedings of the fourth international conference on music information retrieval* (pp. 75-79). Baltimore (Maryland).
- Vega, D. (2012). ¿Manierismo en Tomás Luis de Victoria? *Revista de Musicología*, 35(1), 131-153.
- Venables, W. N. y Ripley, B. D. (1997). *Modern applied statistics with S-Plus* (2ª ed.). Nueva York: Springer-Verlag.
- Vercelli, A. (2006). *Aprender la libertad. El diseño del entorno educativo y la producción colaborativa de los contenidos básicos comunes*. Buenos Aires: Autopublicado.
- Vila, F. (2001). Asignaturas optativas en el tercer ciclo de grado medio en el Conservatorio de Badajoz. En UEX (Ed.), *Actas de las I Jornadas de educación musical*.

- Vila, F. (2007). Dive into the Lily Pond. Teaching music typesetting with Free Software. En Association Européenne des Conservatoires, Académies de Musique et Musikhochschulen (AEC) (Ed.), *Proceedings of the AEC Annual Congress and General Assembly*.
- Vila, F. (2009, diciembre). *Orígenes y evolución de la tipografía musical*. (Estudio de DEA. Universidad de Extremadura. Facultad de educación. Departamento de Didáctica de la expresión musical, plástica y corporal).
- Vila, F. (2012). Liberación de partituras en el mundo digital. En CEIMUS II (Ed.), *Actas del II Congreso de Educación e Investigación Musical: Innovación, motivación y creatividad en la enseñanza musical*.
- Vila, F. (2013). Notación musical, educación y software libre. En UEX (Ed.), *Actas del I Simposio internacional de educación musical*.
- Villenave, V. (2009a). Au conservatoire supérieur de Badajoz, en Espagne, Francisco Vila, professeur, enseigne Lilypond. *FNAPEC journal*, (54).
- Villenave, V. (2009b). Ópera libre. Repositorio de código fuente. Recuperado el 12 de marzo de 2009, desde http://repo.or.cz/w/opera_libre.git
- Villenave, V. (2010). *Affaire Étrangère (opéra)* [Partitura musical]. París: Autopublicada.
- Vinci, A. C. (1989). *Fundamentals of traditional music notation*. Kent State University Press.
- Vives, J. (2006). Los límites y excepciones a la propiedad intelectual. (10). Biblioteca EPSEVG. Servicio de Propiedad Intelectual (UPC). Grupo de Trabajo BPI (Bibliotecas y Propiedad Intelectual - FESABID) Máster en gestión cultural. Universidad de Sevilla, Universidad de Granada, Junta de Andalucía.
- Walshaw, C. (1993). abc2mtex: an easy way of transcribing folk and traditional music, versión 1.0. Londres: University of Greenwich.
- Wanske, H. (1988). *Musiknotation: von der Syntax des Notenstichs zum EDV-gesteuerten Notensatz*. Mainz: Schott-Verlag.
- Wayner, P. (2000). *Free for all: how Linux and the free software movement undercut the high-tech titans*. Nueva York: Harper Business.
- Weaner, M. y Boelke, W. (1993). *Standard music notation practice*. Edición revisada por A. Broido y D. Dorff. Nueva York: Music Publisher's Association of the United States Inc.
- Wiggins, G., Miranda, E., Smaill, A. y Harris, M. (1993). A framework for the evaluation of music representation systems. *Computer Music Journal*, 17(3).
- Willberg, H. P. y Forssman, F. (2002). *Primeros auxilios en tipografía*. Barcelona: Gustavo Gili.
- Williams, S. (2011). *Free as in freedom: Richard Stallman's crusade for free software*. O'Reilly Media, Inc.
- Wilson, E. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, (22), 209-212.
- Wirth, N. (1975). *Algorithms + Data Structures = Programs*. Englewood Cliffs: Prentice-Hall.

Wolf, J. (1919). *Handbuch der Notationskunde*. Leipzig: Breitkopf & Hartel.

Wright, H. (2014). How to read and write TAB: a guide to TAB notation. No disponible actualmente. Recuperado el 8 de enero de 2014, desde <http://www.guitartabs.cc/tabfaq.shtml>