

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA**

INGENIERO EN INFORMÁTICA

Síntesis musical humanizada con notación musical simbólica

Realizado por

RAFAEL DE VICENTE MILANS

Dirigido por

**FRANCISCO J. VICO VELA
DAVID D. ALBARRACÍN MOLINA**

Departamento

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA

MÁLAGA, Junio de 2013.

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERO EN INFORMÁTICA

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente/a Dº/Dª. _____

Secretario/a Dº/Dª. _____

Vocal Dº/Dª. _____

para juzgar el proyecto Fin de Carrera titulado:

Síntesis musical humanizada con notación musical simbólica

realizado por Dº. Rafael De Vicente Milans

tutorizado por Dº. Francisco J. Vico Vela y Dº. David D. Albarracín Molina

ACORDÓ POR _____ OTORGAR LA
CALIFICACIÓN DE _____

Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPARECIENTES
DEL TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga a de de

El Presidente _____ El Secretario _____ El Vocal _____

Fdo: _____ Fdo: _____ Fdo: _____

Síntesis musical humanizada con notación musical simbólica

ÍNDICE

Capítulo 1: Introducción.....	15
Motivación.....	15
Objetivos.....	16
Resumen de capítulos	16
• Capítulo 2: Introducción a la música y su proceso de grabación.....	16
• Capítulo 3: Proyecto Melomics	17
• Capítulo 4: Entorno, programas y formatos.....	17
• Capítulo 5: Arquitectura del software.....	17
• Capítulo 6: Generación	17
• Capítulo 7: Síntesis	18
• Capítulo 8: Pruebas y resultados.....	18
• Capítulo 9: Conclusiones y extensiones futuras	18
Capítulo 2: Introducción a la música y su proceso de grabación	21
2.1 Introducción.....	21
2.2 Características del sonido	21
2.2.1 Altura.....	22
2.2.2 Duración	22
2.2.3 Intensidad	22
2.2.4 Timbre	23
2.3 Elementos de la música	23
2.4 Notación musical	24
2.5 Efectos, articulaciones y adornos	29
2.6 Música por ordenador.....	30
2.7 Grabación musical	31
Capítulo 3: Proyecto Melomics	35
3.1 Introducción.....	35
3.2 Aspectos tecnológicos	35
3.2.1 Música evolutiva	35
3.3 Clúster de computación	36
3.4 De la investigación a la empresa. Melomics Media.	37

Capítulo 4: Entorno, programas y formatos	41
4.1 Entorno	41
4.1.1 Parallel Computing Toolbox	41
4.1.2 Torque.....	42
4.1.3 Maui.....	42
4.2 Programas externos	42
4.2.1 Lilypond	43
4.2.2 Timidity	43
4.2.3 SoX	43
4.2.4 Ecasound.....	44
4.2.5 FluidSynth	44
4.3 Herramientas externas adicionales	44
4.3.1 Extreme Sample Converter.....	44
4.3.2 Translator 6.....	45
4.3.3 Viena.....	45
4.3.4 Mixcraft	45
4.3.5 Synthfont	45
4.4 Formatos	46
Capítulo 5: Arquitectura del software	49
5.1 Introducción.....	49
5.2 Estructura básica.....	49
5.3 Notación musical simbólica.....	50
5.4 Límite de instrumentos	52
5.5 Volúmenes	55
5.6 Optimización	57
5.6.1 Computación paralela.....	58
5.6.2 Estrategias de optimización	60
5.6.2.1 Estrategia 1: Paralelismo en SoX	61
5.6.2.2 Estrategia 2: Paralelismo a nivel de instrumento.....	61
5.6.2.3 Estrategia 3: Paralelismo de instrumentos y mezcla	62
5.6.2.4 Estrategia 4: Paralelismo a nivel de canción	62
Capítulo 6: Generación.....	65
6.1 Introducción.....	65

6.2 Humanizar	65
6.3 Recursos interpretativos	68
6.3.1 Interpretación por duplicación.....	70
6.3.2 Interpretación por desarrollo	71
6.3.2.1 Glissando	71
6.3.2.2 Arpeggio	72
Capítulo 7: Síntesis.....	77
7.1 Introducción.....	77
7.2 Paletas de instrumentos	77
7.3 Mezcla	79
7.3.1 Volúmenes	81
7.3.1.1 Fader	82
7.3.1.2 Compresor-limitador	85
7.3.2 Ecuación.....	86
7.3.3 Panorama	89
7.3.4 Efectos de sonido basados en tiempo	91
7.3.4.1 Retardo simple.....	92
7.3.4.2 Retardos con realimentación	94
7.3.3.3 Reverberación.....	95
Capítulo 8: Pruebas y resultados	99
8.1 Procesado de audio	99
8.1.1 Ancho de banda en la ecuación.....	99
8.1.2 Configuración de la reverberación	101
8.1.3 Efecto eco	102
8.2 Optimización con paralelismo	103
8.2.1 Eficiencia individual.....	103
8.2.2 Máxima cantidad	105
Capítulo 9: Conclusiones y extensiones futuras	107
9.1 Conclusiones.....	107
9.2 Extensiones.....	109
Bibliografía.....	111
A.1 melParams	115
A.2 modPitch y duplicate	116

A.3 Script para lanzamiento en clúster	117
B.1 Generación paso a paso.....	119
B.1.1 Crear archivo sf2.....	119
B.1.2 Edición de archivo sf2	122
B.1.3 Crear una paleta de instrumentos	123

Índice de figuras

Figura 2.1. Pentagrama.....	24
Figura 2.2. Representación de diferentes claves junto a la nota DO central.....	24
Figura 2.3. Compás 2/4 y 3/4.....	25
Figura 2.4. Escala musical	25
Figura 2.5. Indicación metronómica para un tempo de 120 negras por minuto	25
Figura 2.6. Acordes	25
Figura 2.7. Puntillos y puntillos dobles.....	27
Figura 2.8. Alteraciones de bemol y becuadro.....	27
Figura 2.9. Ligadura	27
Figura 2.10. Legato	27
Figura 2.11. Articulaciones.....	27
Figura 2.12. Dinámicas	28
Figura 2.13. Dinámicas de transición	28
Figura 2.14. Pizzicato y tremolo.....	28
Figura 2.15. Glissando	28
Figura 2.16. Trino	29
Figura 2.17. Arpeggio.....	29
Figura 3.1. Iamus, foto de Jorge Dragón	37
Figura 3.2. Reproductor Melomics	37
Figura 3.3. Buscador Melomics.....	38
Figura 3.4. Capturas de la aplicación @life	38
Figura 3.5. Partitura de la obra Terminus compuesta por Iamus.....	39
Figura 5.1. Estructura básica	50
Figura 5.2. Bucle para generación audio final, siendo cDup la cantidad de duplicados de cada instrumento.....	54
Figura 6.1. Curva de Gauss	67
Figura 6.2. Proceso de interpretación por duplicación.....	71
Figura 7.1. Parámetros fundamentales para dar forma a una onda sonora (DAHDSR)	78
Figura 7.2. Ecualizador gráfico de 31 bandas.....	86
Figura 7.3. EQ general de los estilos Heavy-Metal, Jazz y Electrónica.....	88
Figura 7.4. Colocación de los instrumentos de una orquesta sinfónica.....	90
Figura 8.1. Análisis espectral. (A) Ruido Blanco, (B, C, D, E) EQ con anchos de banda de 1, 10, 50 y 1.000 Herzios respectivamente	100
Figura B.1. Generación de preset, Extreme Sample Converter	119
Figura B.2. Generación de preset, ESC, imagen 2.....	120
Figura B.3. Generación de preset, ESC, imagen 3.....	121
Figura B.4. Generación de preset, ESC, imagen 4.....	122
Figura B.5. Generación de preset, Viena.....	123
Figura B.6. Generación de preset, Viena, imagen 2	124
Figura B.7. Generación de preset, Viena, imagen 3.....	124
Figura B.8. Generación de preset, Viena, imagen 4.....	124

Índice de tablas

<i>Tabla 2-I. Figuras musicales y su relación de tiempos</i>	26
<i>Tabla 2-II. Figuras musicales del silencio</i>	26
<i>Tabla 6-I. Ejemplo de distribución normal con parámetro de entrada 6</i>	68
<i>Tabla 7-I. Dinámicas creadas con el equipo clasificadas por emociones</i>	81
<i>Tabla 7-II. Clasificación de instrumentos según volumen aparente ordenados de mayor sonoridad (1) al menor (6)</i>	83
<i>Tabla 7-III. Relación MIDI-Nota-Frecuencia</i>	87
<i>Tabla 7-IV. Rangos de frecuencias</i>	88
<i>Tabla 7-V. Tempo vs Retardo</i>	92
<i>Tabla 7-VI. Rapidez del cerebro</i>	93
<i>Tabla 8-I. Reverb en electrónica</i>	101
<i>Tabla 8-II. Reverb en música de audiovisuales</i>	102
<i>Tabla 8-III. Reverb para quartet</i>	102
<i>Tabla 8-IV. Configuraciones para el efecto de eco</i>	103
<i>Tabla 8-V. Tiempos de obtención de un fichero MP3</i>	104
<i>Tabla 8-VI. Tiempos de obtención de 4 piezas musicales</i>	105

Capítulo 1: Introducción

Motivación

La música es, según su definición tradicional, el arte de organizar sensible y lógicamente una sucesión coherente de sonidos y silencios utilizando los principios fundamentales de la melodía, la armonía y el ritmo; donde intervienen complejos procesos psico-anímicos del autor. Por todo ello la música siempre ha sido considerada un arte solo al alcance del hombre, pero con el transcurrir del tiempo y la tecnología, la informática y los ordenadores han evolucionado poco a poco dentro del campo de la música llegando a poder componer obras musicales indistinguibles de las realizadas por un humano. Por otra parte las computadoras comenzaron como simples medios de reproducción y actualmente son una utilidad prácticamente indispensable para el compositor instrumental.

En el proyecto de investigación Melomics en el que participa el Grupo de Estudios en Biomimética (GEB) de la universidad de Málaga, y al cual está adjunto este Proyecto de Fin de Carrera, se ha conseguido la generación de música con el computador Iamus. Esta máquina, mediante algoritmos evolutivos que tiene implementados, genera melodías y contenido musical que almacena en estructuras informáticas. El objetivo de este PFC es proporcionar a Iamus un software de síntesis musical con el que, además de componer piezas musicales, pueda también interpretarlas, es decir, transformar en un archivo de audio estándar la información musical almacenada internamente. El sistema dotará a dichos archivos de salida de efectos, adornos y otros recursos interpretativos, tanto generales, como específicos de instrumentos y estilos, de manera que la emulación automática de la composición se aproxime a la ejecución de un intérprete humano.

Para llevar a cabo todo ello se hace uso del protocolo MIDI el cual proporciona la capacidad de hacer audible dicha información musical, siendo el formato más extendido para la edición de composición musical y generación de audio. Dicho formato no es utilizado para la reproducción directa de la obra musical sino mediante el uso de instrumentos virtuales, procesado de audio, mezcla y otros recursos, para obtener archivos de audio convencionales, como WAV o MP3. Todo este proceso se efectuará de una manera automática, buscando su optimización y obtener un audio resultante con la mayor calidad posible actualmente, aprovechando al máximo los recursos disponibles: clúster de computación Iamus y ordenadores multinúcleo en caso de ejecución local.

Objetivos

Los objetivos del proyecto se agrupan en dos fases: generación y síntesis. La generación se centra en manipular la estructura de datos de entrada hasta obtener una pieza musical dotada de toda la información necesaria para emular los recursos interpretativos de cada instrumento en su posterior sintetizado. En la síntesis se transforma la pieza musical obtenida en la generación, mediante técnicas de procesado de audio y conversión, en archivos de audio reproducibles por cualquier usuario, siendo estos archivos de la mayor calidad alcanzable y fieles a la partitura musical.

Principales objetivos de la generación:

- Interpretar e implementar los principales efectos, articulaciones y adornos musicales, como el legato, glissando, etcétera.
- Subsana la limitación en el número de instrumentos en una misma melodía del protocolo MIDI.

Principales objetivos de la síntesis:

- Desarrollar un software que genere distintos formatos de audio de manera automática, entre ellos MP3, WAV y FLAC.
- Realizar una mezcla del audio haciendo uso de las herramientas más comunes en un estudio de grabación.
- Añadir efectos de sonido que doten al audio de mayor calidad y variedad, tales como eco, reverberación, etcétera.
- Construir paletas instrumentales adecuadas para cada instrumento, efecto y estilo musical.

Los objetivos de generación y síntesis están ligados entre sí, ya que aquellas mejoras que se hagan sobre la calidad interpretativa de la composición deben ser luego reproducidas en el audio resultante.

Resumen de capítulos

A continuación se describe el contenido y estructura de este documento:

- **Capítulo 2: Introducción a la música y su proceso de grabación**

En este capítulo se exponen conocimientos relacionados con la música: el sonido, los elementos de la música y la notación musical, así como una breve introducción a las distintas maneras en que se utiliza el ordenador como apoyo para interpretar o crear música. Por último se da a conocer el proceso de

grabación de una canción con especial atención a la mezcla, última fase y que afecta definitivamente a cómo percibimos la composición.

- **Capítulo 3: Proyecto Melomics**

Se pone en contexto el proyecto Melomics para el cual se desarrolló este Proyecto Fin de Carrera, explicando los problemas que intenta solucionar y las opciones que se pretenden ofrecer dentro del proyecto general.

- **Capítulo 4: Entorno, programas y formatos**

Durante el capítulo se dan a conocer el entorno de desarrollo y sus herramientas, las distintas plataformas sobre las que es posible ejecutar el software, los distintos formatos de archivo utilizados durante el desarrollo, así como los múltiples programas que se han usado durante la realización del proyecto.

- **Capítulo 5: Arquitectura del software**

En esta sección se explica la arquitectura de la aplicación y los problemas y soluciones que se consideraron, entre ellos los modelos de computación empleados para acelerar y optimizar la ejecución del programa. Se proporciona también una breve introducción a la computación paralela y las distintas estrategias de optimización que se probaron.

- **Capítulo 6: Generación**

En este capítulo se detallarán los puntos principales de la generación de la melodía, que produce como resultado un archivo MIDI. Se prestará especial atención a la humanización musical y los pasos llevados a cabo para su implementación.

- **Capítulo 7: Síntesis**

En él se tratará la parte más importante y problemática del proyecto, el proceso de mezcla. También se dará a conocer el término paleta de instrumentos y su función.

- **Capítulo 8: Pruebas y resultados**

Durante el capítulo se exponen las pruebas realizadas para las distintas estrategias de optimización, controlando y estudiando su mayor o menor efectividad. También se muestran las pruebas sobre las distintas funciones de procesado de audio y los resultados y conclusiones obtenidos en ellas.

- **Capítulo 9: Conclusiones y extensiones futuras**

En este capítulo se expondrán las conclusiones obtenidas tras la realización del proyecto y se comentarán las posibles extensiones y mejoras a desarrollar para mejorar el software.

Capítulo 2: Introducción a la música y su proceso de grabación

2.1 Introducción

Existen múltiples definiciones de música, a continuación algunas de las que aparecen en el diccionario de la Real Academia Española:

- Melodía, ritmo y armonía, combinados.
- Sucesión de sonidos modulados para recreación del oído.
- Arte de combinar los sonidos de la voz humana o de los instrumentos, o de unos y otros a la vez, de suerte que produzcan deleite, conmoviendo la sensibilidad, ya sea alegre, ya tristemente.

Algo común tanto en estas definiciones como en cualquier otra es que la música es un arte, el de combinar sonidos y, como todo arte, éste tiene una gran parte subjetiva, de ahí que a cada persona le guste un tipo de música u otro. Por ello en el proyecto se centra en dotar a los sonidos de cada instrumento de la mayor calidad posible y del máximo número de recursos interpretativos, aprovechando así toda la información generada por Melomics, y proporcionar métodos para conseguir una ejecución próxima al intérprete humano. Se darán a conocer algunas definiciones y conceptos musicales adicionales a lo largo del capítulo. Entre ellos, qué se considera sonido musical y sus características, los efectos, adornos y articulaciones musicales que hacen que una misma nota musical suene de una manera u otra, así como la notación que se usa para indicar qué sonido se debe reproducir. Por último se abordará en qué consiste el proceso de mezcla qué se lleva a cabo en los estudios de grabación y el cual se intentará simular en el proceso de síntesis automática para obtener los archivos de audio, ya que la calidad de una obra musical no solo depende de lo bueno que sea cada instrumento individualmente, sino de la correcta conjunción de todos los sonidos que intervienen durante la misma.

2.2 Características del sonido

El arte de combinar sonidos y silencios, es una de las definiciones más comunes de música. Por lo que el sonido, y los silencios, son la materia prima de la música.

El sonido es la sensación percibida por el oído al recibir las variaciones de presión generadas por el movimiento vibratorio de los cuerpos sonoros. Se transmite por el medio que los envuelve, que generalmente es el aire de la atmósfera. La ausencia

perceptible de sonido es el silencio, aunque es una sensación relativa, ya que el silencio absoluto no se da en la naturaleza.

El sonido tiene cuatro propiedades básicas que lo definen: altura o tono, duración, intensidad y timbre o color.

2.2.1 Altura

La altura en música es la cualidad que diferencia un sonido agudo de un sonido grave. Se determina por la frecuencia fundamental de las ondas sonoras, medida en ciclos por segundos o hercios (Hz). Mientras más lenta sea la vibración más baja será la frecuencia y más grave el sonido y viceversa, a mayor velocidad de vibración más alta será la frecuencia y más agudo el sonido.

Las ondas sonoras se originan siempre por el movimiento vibratorio de algún cuerpo en contacto con el aire. Cada instrumento suena en un umbral de frecuencias que es capaz de alcanzar, por ejemplo un piano de cola está entre 20 y 4176 Hz. Para que los humanos podamos percibir un sonido, éste debe estar comprendido entre el rango de audición de 20 y 20.000 Hz. Este rango se le denomina rango de frecuencia audible, por debajo están los infrasonidos y por encima los ultrasonidos.

En la música occidental se fueron estableciendo tonos determinados llamados notas, cuya secuencia de 12 (*C, C#, D, D#, E, F, F#, G, G#, A, A#, B*, indicando el símbolo # que la nota es sostenida) se va repitiendo en octavas.

2.2.2 Duración

La duración corresponde al tiempo que duran las vibraciones que producen un sonido. Este parámetro está relacionado con el ritmo.

La duración de la música se representa mediante figuras. La figura que representa la unidad es la redonda y sirve como referencia para conocer el valor del resto de las figuras. Estas figuras se subdividen de tal forma que cada una de ellas vale el doble que el valor siguiente.

2.2.3 Intensidad

La intensidad en música es la cualidad que diferencia un sonido suave de uno fuerte. Depende de la fuerza con la que el cuerpo sonoro sea ejecutado y de la distancia del receptor de la fuente sonora.

Acústicamente la intensidad depende de la amplitud de las vibraciones y se mide comúnmente en decibelios.

La dinámica musical hace referencia a las graduaciones de la intensidad del sonido. En la terminología musical se denomina matiz dinámico a cada uno de los niveles de intensidad en que se pueden interpretar uno o varios sonidos y van desde el pianississimo (*ppp*) al fortississimo (*fff*).

2.2.4 Timbre

El timbre es el matiz característico de un sonido. Los sonidos que escuchamos están compuestos de múltiples ondas simultáneas pero que nosotros percibimos como uno. El timbre depende de la cantidad de armónicos que tenga un sonido y de la intensidad de cada uno de ellos. Gracias a ello suena diferente un 'la' de 440Hz tocado con una flauta del tocado con una trompeta.

2.3 Elementos de la música

La música se basa en la organización de sonidos y silencios combinados según unos parámetros fundamentales: melodía, armonía y ritmo.

- La melodía es un conjunto de sonidos y silencios combinados uno tras otro (concepción horizontal), y que se perciben como uno solo en el tiempo.
- La armonía regula la unión entre sonidos que suenan a la vez y su enlace con sonidos vecinos. A diferencia de la melodía que controla horizontalmente (a lo largo del tiempo) la armonía lo hace verticalmente (en un mismo instante) y su unidad básica es el acorde.
- El ritmo se define como la organización de pulsos y acentos. Consiste en la organización en el tiempo de sonidos y silencios formando un patrón que suele repetirse. La frecuencia de tal repetición puede ser regular o irregular. Un ritmo puede tener un pulso constante, pero también puede tener diferentes tipos de pulsos o tiempos. Algunos tiempos pueden ser más fuertes, más suaves, más largos o más cortos que otros. El ritmo tiene mucho que ver con definir el compás, el tipo de compás que define al acento y a las figuras musicales que la componen. El ritmo no se escribe con pentagrama, sólo con la figura musical definitoria de la duración del pulso. Si al pentagrama se le agregan figuras musicales aparece el sonido. Al sumar los acentos, figuras y un compás definido por el ritmo, surge la melodía.

2.4 Notación musical

La notación musical es un sistema de representación gráfica de una pieza musical, para que ésta pueda ser leída, comprendida y ejecutada por músicos. Gracias a este sistema un músico puede interpretar la música de la manera deseada por el compositor.

Existen muchos tipos de sistemas de notación musical, pero el más común es el sistema gráfico occidental. Este sistema también conocido como notación moderna se representa mediante signos sobre un pentagrama y cuyo elemento básico es la nota. A continuación, una breve descripción de los principales elementos:

- Pentagrama. Sobre él se escribe la música y consiste en cinco líneas y cuatro espacios entre ellas (Figura 2.1). Sobre estas líneas y espacios se dibujan las notas. Cada línea y espacio especifica una entonación, siendo la más grave la más baja y la aguda la más alta. El pentagrama se lee de izquierda a derecha por lo que horizontalmente se representa el tiempo.

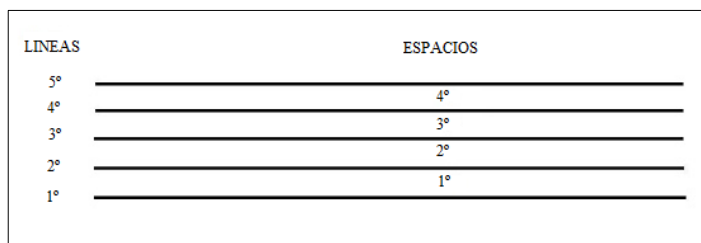


Figura 2.1. Pentagrama

- Clave. Es un signo musical que sirve para determinar el nombre de las notas que se representan en el pentagrama. Este gráfico se coloca al principio del pentagrama y representan las tesituras correspondientes a cada voz. Las claves llevan el nombre de las notas FA, SOL y DO.



Figura 2.2. Representación de diferentes claves junto a la nota DO central

- Compás. Es la unidad de tiempo en la que se divide la música. Las principales divisiones del compás se denominan tiempos y se encuentran representados en el pentagrama mediante una línea vertical que divide los distintos compases. Existen distintos tipos de compases que se reconocen en la partitura a través del indicador de compás que aparece junto a la clave. Dicho indicador es una

fracción numérica cuyo denominador indica en cuantas partes se ha de dividir una semibreve o redonda para obtener una unidad de tiempo y el numerador indica el número de unidades de tiempo en un compás. Por ejemplo un compás 4/4 indica que un compás corresponde a 4 negras o equivalente.



Figura 2.3. Compás 2/4 y 3/4

- Tempo. Indica la velocidad a la que interpretar las notas. Una de las formas de representación es mediante una indicación metronómica, el cual indica el tipo de nota y el número de veces que se debe poder tocar dicha nota en un minuto, situada al inicio del compás encima del pentagrama.

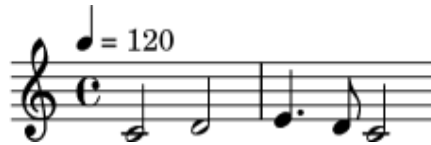


Figura 2.5. Indicación metronómica para un tempo de 120 negras por minuto

- Notas. Los diferentes tonos son llamados mediante letras. En el software se usa la notación americana de estas que es A, B, C, D, E, F, G (La, Si, Do, Re, Mi, Fa, Sol en notación española). Cada línea y espacio del pentagrama representa un tono, de ser necesario se pueden añadir líneas extras al pentagrama para poder alcanzar un tono determinado.

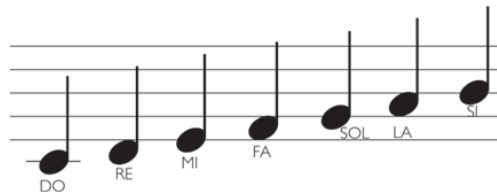


Figura 2.4. Escala musical

- Acorde. Se produce cuando las notas son representadas en la misma vertical del pentagrama, y representan un conjunto de notas que son tocadas a la vez.



Figura 2.6. Acordes

- Figuras musicales. Estas son redonda, blanca, negra, corchea, semicorchea, fusa y semifusa y sirven para indicar la duración de la nota. La duración exacta depende del compás y el tempo. La cantidad de golpes que posee cada figura depende del compás, pero de manera relativa una redonda equivale a una unidad

de tiempo y la blanca la mitad de esta y así sucesivamente durando cada nota la mitad de lo que dura la figura anterior.




Cuadrada		8 Tiempos
Redonda		4 Tiempos
Blanca		2 Tiempos
Negra		1 Tiempo
Corchea		1/2 Tiempo
Semicorchea		1/4 Tiempo
Fusa		1/8 Tiempo
Semifusa		1/16 Tiempo

Tabla 2-I. Figuras musicales y su relación de tiempos

- Silencios. Al igual que las figuras musicales indican la duración, aunque esta vez del espacio sin interpretar música. A diferencia de las notas los silencios se sitúan siempre en el centro del pentagrama.








SILENCIOS	
REDONDA	
BLANCA	
NEGRA	
CORCHEA	
SEMICORCHEA	
FUSA	
SEMIFUSA	

Tabla 2-II. Figuras musicales del silencio

- Puntillo. Indica que la duración de la nota se incrementa en la mitad de la duración de dicha nota. También puede añadirse otro puntillo incrementándose la duración un cuarto de la duración original extra.



Figura 2.7. Puntillos y puntillos dobles

- Alteraciones o accidentes. Son unos signos que se colocan a la izquierda de las notas y modifican su entonación. Podemos encontrar tres alteraciones básicas. El sostenido (#) sube medio tono, el bemol baja medio tono y el becuadro anula las alteraciones anteriores.



Figura 2.8. Alteraciones de bemol y becuadro

- Ligadura. Indica que dos notas unidas entre sí son tocadas como una sola nota.



Figura 2.9. Ligadura

- Legato. Hace que todas las notas cubiertas bajo este signo sean interpretadas sin articular una separación a través de la interrupción de sonido.



Figura 2.10. Legato

- Articulación. Signo o símbolo que se le añade a una nota para indicar cómo debe ser ejecutada. Ejemplos de articulaciones son el staccato, que indica que la nota es tocada de manera más corta que la anotada, normalmente la mitad de esta y el resto es silencio; y el acento, que indica que la nota ha de ser ejecutada con un ataque más intenso.

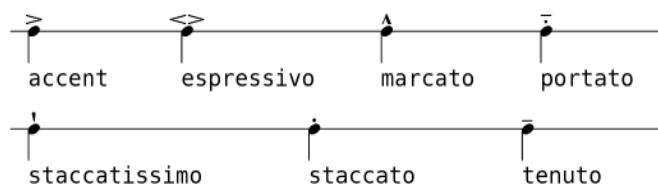


Figura 2.11. Articulaciones

- Dinámica. La intensidad de las notas puede variar durante una pieza musical y estos cambios se indican mediante la dinámica. Estas indicaciones van desde el piano o *p* (intensidad suave), mezzo piano o *mp* (moderadamente suave), mezzo forte o *f* (moderadamente fuerte) y forte o *f* (intensidad fuerte). Tanto a la *f* como la *p* se le pueden añadir más *p* o *f* extra para indicar que se suavice o fortalezca la intensidad de golpeo.



Figura 2.12. Dinámicas

- Crescendo y diminuendo. Son moduladores de dinámica, indican qué la intensidad de uno o más sonidos ha de ser aumentada o disminuida de forma paulatina.



Figura 2.13. Dinámicas de transición

- Pizzicato. Indica que se toca pulsando las cuerdas con los dedos y no frotando con el arco. Se representa usualmente con el texto *pizz.* encima o debajo de la nota donde se inicia el efecto (Figura 2.14).
- Tremolo. Significa que debe haber una fluctuación en la intensidad del sonido, manteniendo la altura constante. En la nota de comienzo se indica explícitamente su uso (*trem.*).

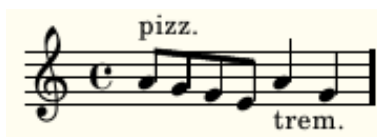


Figura 2.14. Pizzicato y tremolo

- Glissando. Es un adorno, un efecto musical consistente en pasar rápidamente de un sonido hasta otro más agudo o más grave haciendo audibles todos los sonidos intermedios posibles dependiendo de las características del instrumento.



Figura 2.15. Glissando

- Trino. Es una rápida alteración entre la nota específica y el siguiente tono o semitono más alto dentro de su duración. Su representación se ilustra en la figura 2.16.



Figura 2.16. Trino

- Arpeggio. Es la sucesión más o menos acelerada de los sonidos de un acorde. Dicha sucesión se hace normalmente del sonido más grave al más agudo. Para diferenciar un acorde de un arpeggio aparece a la izquierda del acorde una línea zigzagante junto con una indicación de sentido que muestra el orden a seguir.



Figura 2.17. Arpeggio

2.5 Efectos, articulaciones y adornos

Un efecto es una técnica interpretativa que consiste en hacer sonar un instrumento de una manera particular distinta a la habitual. Ejemplos de estas técnicas son el pizzicato, que se consigue pellizcando las cuerdas con las yemas de los dedos en instrumentos como el violín; o el tremolo, que consiste en una variación periódica de la amplitud en el sonido, siendo diferente la ejecución según el instrumento. Los efectos se aplican cuando aparecen indicados en la partitura en forma de abreviatura sobre la nota de inicio del efecto correspondiente, por ejemplo *pizz.* para pizzicato o *trem.* para trémolo.

Las articulaciones definen las diferentes posibilidades en las que se pueden conectar entre sí las notas que componen una melodía. Estas posibilidades se diferencian básicamente en función de tres elementos: el ataque de cada nota, la caída de cada nota y el grado de interrupción o continuidad del sonido existente entre las distintas notas. Ejemplos de articulaciones son el legato o ligado que indica que las notas se tocan una tras otra sin interrupción del sonido, y el staccato o picado en el que hay una interrupción total del sonido entre nota y nota. Cada articulación tiene su propia representación en la notación musical.

Los adornos u ornamentos musicales son florituras que no son necesarias para llevar la línea general de la melodía (o armonía), pero que sirven para decorar esa línea. Se trata de recursos que pueden ser utilizados en las composiciones con el objeto de imprimirles a éstas expresión, ornamento, variedad, gracia o vivacidad. Ejemplos de adornos son el trino, que consiste en una rápida alternancia entre dos notas adyacentes, y el mordente,

que generalmente indica que la nota principal (la nota escrita en la partitura) debe ser tocada con una única alternancia rápida de dicha nota con la inmediatamente superior o inferior en la escala musical. La manera de representación cambia según el adorno, que puede ser un símbolo propio o la abreviatura sobre la nota.

2.6 Música por ordenador

Con el término música por ordenador se entiende toda aquella música creada por o con ayuda de uno o más computadores. El uso del ordenador se puede clasificar en dos categorías dependiendo de la forma de procesado de la información:

- Tiempo real.
- Tiempo de compilación.

En el uso en tiempo real el ordenador proporciona una respuesta inmediata a un estímulo, en este caso existen dos claros ejemplos de uso:

- Para crear y manipular sonidos creados durante una interpretación musical en directo, por ejemplo modificando una voz para que suene robótica.
- Composición interactiva. Implica la creación de un software que responda a unos estímulos predeterminados de una ejecución en vivo. Mediante el uso de este software se modifican de manera total o parcial alguno de los sonidos existentes en la composición, incluso sustituir o añadir nuevos instrumentos. El rol que desempeñe el ordenador puede ser modificado durante la interpretación. Un lenguaje de programación usado para la interactividad es *Pure Data*^[8].

Cuando existe un tiempo de compilación, en ocasiones importante, se distinguen dos usos:

- El ordenador como instrumento, es decir como la fuente generadora del sonido o herramienta de síntesis. El método más común para realizar esta edición de audio es mediante la tecnología de estudio virtual, o VST (siglas en inglés) como es más conocido, que es una interfaz desarrollada por *Steinberg*^[9] para conectar plugins de efectos de sonido y sintetizadores a programas de edición y grabación de audio. VST necesita de aplicaciones que soporten dicha tecnología para ser ejecutado. La mayoría de aplicaciones de edición y grabación de audio soportan esta la tecnología, algunos ejemplos son *Cubase*, *Mixcraft*^[10], *Rosegarden*^[11] o *Synthfont*^[12]. A través de estas aplicaciones se puede interactuar con el formato MIDI^[13], pudiéndose así editar o componer cualquier obra y generar sonidos para cada instrumento, ya sea mediante *VSTi* (*VST instrument*), archivos *sf2* (*SoundFont*) o creándolos directamente mediante algún plugin de síntesis de ondas como *Zynaddsubfx*^[14]. Adicionalmente se pueden

insertar efectos como reverb o retardos mediante funciones de la aplicación u otros plugins.

- El ordenador como compositor o ayudante del mismo, conocido como composición algorítmica. Existen diversos métodos y algoritmos de composición, algunos de ellos son:
 - Composición automática. Es una forma de composición algorítmica que involucra elementos mecánicos o digitales de toma de decisiones. Algunos ejemplos se encuentran en la tesis de Andrés Eduardo Coca Salazar^[15], y en el proyecto *Muphic*^[16] de composición automática basada en imágenes.
 - Composición por autómata celular. Es un modelo de computación, compuesto por una rejilla discreta de celdas, cada una de las cuales puede estar en un estado dentro de un conjunto dado. En cada iteración, se actualiza el estado de todas las celdas en paralelo de acuerdo con una serie de reglas que tienen en cuenta el estado de la celda o célula y sus vecinas. El ejemplo más conocido del uso de este modelo es *WolframTones*^[17].
 - Composición evolutiva. Es aquella que hace uso de algoritmos evolutivos, los cuales son métodos de optimización y búsqueda de soluciones inspirados en la evolución biológica. Se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de forma que las más aptas prevalecen a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez. La computación evolutiva es una rama de la inteligencia artificial y un ejemplo de su uso en composición musical es Melomics (ver capítulo 3).
 - Composición basada en restricciones. Es una forma de composición algorítmica que utiliza reglas para gobernar las opciones posibles que pueden aplicarse a cualquier parámetro de una composición musical. Una restricción o limitación es una condición o regla que debe ser cumplida. Un ejemplo sería el entorno *Rainforest IV*^[18].

2.7 Grabación musical

Una parte importante de la música en la actualidad es su proceso de grabación y edición. Generalmente la grabación de un disco u obra musical se lleva a cabo en un estudio de grabación con ayuda de micrófonos para obtener el sonido de los instrumentos. Situar y configurar los distintos micrófonos para registrar los sonidos y efectos particulares deseados es un proceso complejo. Tras grabar independientemente el sonido de cada instrumento, se realiza la mezcla usando hardware específico como los ecualizadores.

La mezcla es una de las fases más importantes de la grabación y síntesis musical, ya que de ella depende directamente cómo son percibidos los sonidos de la composición. En

general se persigue hacer sentir al oyente de la canción como si estuviera escuchando la música en directo. Para ello, en la mezcla se hace uso principalmente de cuatro elementos:

- Volúmenes. En una composición musical normalmente hay varios instrumentos y cada uno de ellos con un sonido característico (timbre) que influye en el volumen percibido. Sonidos de distintos instrumentos al mismo nivel de volumen son percibidos con distinta intensidad. Por lo cual en la mezcla hay que establecer una relación entre ellos para conseguir simular adecuadamente la sensación de profundidad. Un instrumento que esté al fondo de una orquesta se debiera escuchar menos que el que está en la parte frontal, por ejemplo.
- Panorama. Gracias al estéreo se dispone de dos fuentes de sonido, el altavoz izquierdo y el derecho. Con esta técnica se puede provocar en el oyente la sensación de que un instrumento está más a la izquierda o la derecha o justo delante, también se puede obtener el efecto de hacer parecer mayor el espacio donde se interpreta la composición que la disposición real donde se encuentran los altavoces. Si se usan formatos de audio y sistemas como el *surround* (5.1), el panorama tiene todavía más importancia. Por otro lado si se usa un audio monofónico, el panorama carece de interés ya que solo existe una fuente de sonido.
- Ecuación. Mediante ecualizadores se atenúa o amplifica la señal de audio a determinadas frecuencias. Dichas frecuencias y su ancho de banda están determinados por el ecualizador que se use, desde las 5 bandas a las 31 bandas, así como los ecualizadores paramétricos. Haciendo uso de la ecualización se eliminan ruidos de la grabación, se resaltan las frecuencias más características de cada instrumento sin enmascarar las de otros o se atenúan frecuencias molestas.
- Efectos de sonido basados en el tiempo. Estos efectos de sonido se usan para dar sensación de distancias, ya que la onda sonora rebota por toda la habitación. Dependiendo del estilo musical que se trate se hará un mayor o menor uso. Algunos ejemplos son la reverberación, el eco o el flanger. Este tipo de efectos se usan para aportar realismo a la mezcla.

El tipo de mezcla a realizar dependerá en gran medida del estilo musical de la composición y de los instrumentos empleados. Hay estilos muy tradicionales como la música clásica en las que no se debe innovar porque el oyente espera algo determinado y la mezcla no tiene gran importancia. En otros estilos, como la música electrónica, es la parte principal, se pueden probar múltiples variantes e innovar.

Un ejemplo característico es el caso del bombo (*kick drum*) en el que teniendo en cuenta únicamente su volumen en la mezcla cambia en gran medida según el estilo musical. En orquestas, jazz o new age se sitúa a un nivel muy bajo, percibiéndose de fondo. En el transcurso del tiempo el bombo tiene mucha más presencia, pasando a estar en un nivel intermedio en estilos como rock, country o heavy metal. Con estilos más contemporáneos como rap o hip-hop se convierte en un sonido fundamental alcanzando gran potencia. Al emplear ecualización en la mayoría de bombos se atenúan las

frecuencias comprendidas entre 100-300Hz y se amplifican suavemente frecuencias altas, alrededor de los 5000Hz. Sin embargo aquellos bombos que suelen usarse en rap se realiza algo totalmente distinto, se amplifican las frecuencias entre 40-100Hz y se atenúan aquellas situadas en el rango 1000-5000Hz. En relación al panorama todos los estilos musicales coinciden en situar al bombo en el centro.

Debido a la repercusión y variabilidad que puede alcanzar un único instrumento según el estilo musical se reconoce la importancia de que una pieza musical sea considerada de un estilo u otro, ya que ello afecta, además de al tipo de notas que suenan o la velocidad de su ejecución, a los instrumentos utilizados, el sonido que producen y la colocación entre distintos instrumentos, repercutiendo en el desarrollo del proceso de mezcla. Los parámetros relacionados con la mezcla no son estáticos ni están predefinidos, sino que para cada estilo musical existe un rango aceptado para cada elemento de la misma, permitiendo a cada persona introducir su sello personal.

Capítulo 3: Proyecto Melomics

3.1 Introducción

Melomics^[19] es un proyecto de investigación en la creatividad artificial cuyo objetivo es la automatización completa del proceso de composición de la música profesional. Dentro de dicho proyecto se encuadra el desarrollo de este PFC, con el fin de resolver un problema concreto, la síntesis musical para su reproducción en formatos de audio estándares.

Para ello se darán a conocer algunos de los aspectos tecnológicos en los que se fundamenta Melomics, las funciones que desempeñan los dos clústeres de computadores pertenecientes al proyecto, así como el paso de la investigación al mercado y lo que en este pretende o actualmente ya proporciona Melomics Media.

3.2 Aspectos tecnológicos

Melomics, palabra que deriva de las palabras en inglés *genomics* y *melodies* (genómica y melodías), es un sistema computacional propietario para la composición automática (sin intervención humana) de la música, basándose en métodos bioinspirados. Todo siendo producido y explotado por Melomics Media. Dicha tecnología para la composición musical fue desarrollada por el profesor Francisco J. Vico en 2009 y 2010.

Melomics aplica un enfoque evolutivo de la composición musical, es decir, las piezas musicales se obtienen por la evolución simulada. Estos temas compiten por adaptarse mejor a una apropiada función de aptitud, generalmente basada en criterios formales y estéticos, con casos concretos cómo puede ser penalizar composiciones con acordes de seis notas que se deben interpretar con una sola mano o fomentar la polifonía que forman determinados instrumentos. El sistema Melomics codifica cada tema en un genoma, y toda la población de las piezas musicales se somete a una evolución del desarrollo dinámica (conocido informalmente como evo-devo), cómo por ejemplo, piezas imitando un complejo proceso de desarrollo embriológico. El sistema es totalmente autónomo, por lo que una vez se programa se compone música sin intervención humana.

3.2.1 Música evolutiva

La música evolutiva es la contrapartida de audio al arte evolutivo, en el que la música algorítmica es creada usando un algoritmo evolutivo. El proceso comienza con una

población de individuos que por algún medio o audio producido (por ejemplo, una pieza, melodía o bucle), que o bien se inicializa aleatoriamente o sobre la base de música generada por el hombre. A continuación, a través de la aplicación repetida de pasos computacionales análogos a la selección biológica, recombinación y mutación, el objetivo para audio producido es llegar a ser más musical. Síntesis de sonido evolutiva es una técnica relacionada para generar sonidos o instrumentos sintetizados. Música evolutiva se genera típicamente usando un algoritmo evolutivo interactivo en el que la función de adecuación es el usuario o la audiencia, ya que es difícil de captar las cualidades estéticas de la música computacionalmente. Sin embargo, la investigación sobre medidas automatizadas de calidad musical también está activa. Las técnicas evolutivas de computación también se han aplicado a las tareas de armonización y acompañamiento. Las técnicas más utilizadas son la evolución de computación de algoritmos genéticos y programación genética.

3.3 Clúster de computación

El sistema computacional Melomics incluye dos clústeres de ordenadores programados e integrados en el sistema Melomics:

- Melomics109. Es un clúster de computadoras completamente dedicado a componer y sintetizar la música de los géneros musicales esenciales (música popular), los cuales se recogen en un repositorio musical en continuo crecimiento. Conforme esta tecnología se desarrolla, fragmentos musicales más largos y más complejos se generan.
- Iamus. Está dedicado a la música artística, componiendo fragmentos y obras completas de música clásica contemporánea en su propio estilos, con una calidad tan alta que no se pueden diferenciar de las producidas por compositores profesionales. Algunas de las mejores obras de Iamus han sido grabadas por músicos profesionales y compilado en un CD debut (el primero totalmente compuesta por una inteligencia no humana por sí sola). Sobre los hitos conseguidos por Iamus se pueden encontrar multitud de reportajes y artículos relacionados, como los que pueden encontrarse en la revista *Nature* o en el diario *The Guardian*.

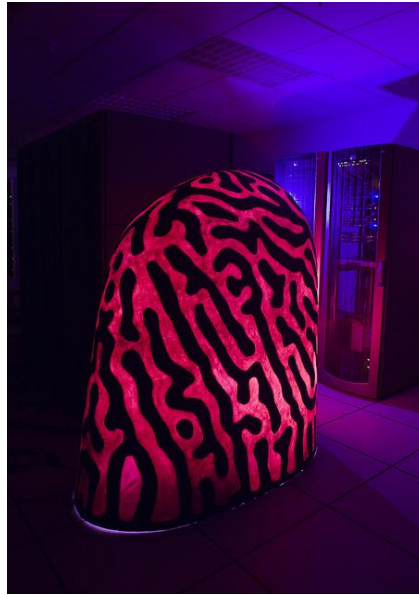


Figura 3.1. Iamus, foto de Jorge Dragón

3.4 De la investigación a la empresa. Melomics Media.

La tecnología Melomics fue transferida a la industria como un *spin-off* académico, Melomics Media, que creó el clúster de computación Melomics109 del que nació una enorme colección de música popular. Dicho repositorio musical es accesible desde el sitio web de Melomics (www.melomics.com) desde cualquier dispositivo, donde la música se encuentra en varios formatos:

- Reproducible en formato MP3.

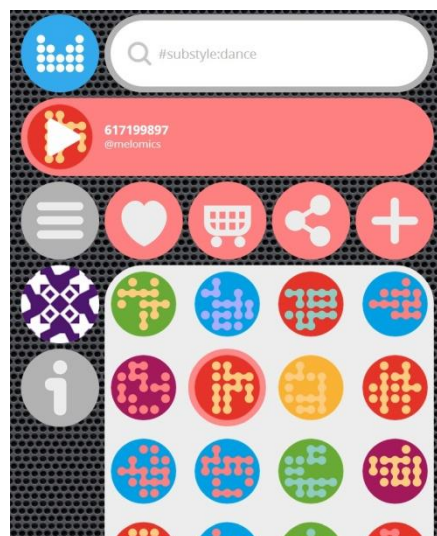


Figura 3.2. Reproductor Melomics

- Editable, disponible en el formato audible MIDI y en formato para intercambio de partituras MusicXML.
- Imprimible, con partituras en formato PDF (Figura 3.3).

Esta diversidad de formatos asegura que las canciones pueden ser reproducidas por los ordenadores y reproductores de medios portátiles, editadas con software profesional de música o interpretada en actuaciones en directo.

Además Melomics crea un mercado de la música en torno a este repositorio, proporcionando aplicaciones móviles y web para buscar, compartir, comprar y distribuir música.

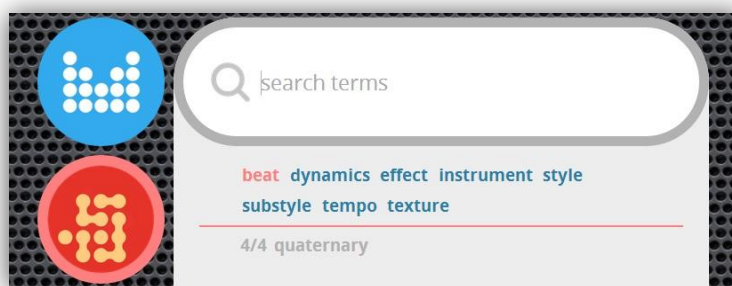


Figura 3.3. Buscador Melomics

Mediante la aplicación *@life* se proporciona un servicio de música en streaming que adapta la música a la actividad o entorno en la que se encuentre. A través de la selección de un escenario se especifica la situación (conduciendo, haciendo jogging o ayudando a dormir a un niño), la actividad será monitorizada por el dispositivo móvil y se



Figura 3.4. Capturas de la aplicación @life

reproducirá la música acorde a ello.

También ha sido desarrollada una API para permitir a empresas externas y usuarios acceder al repositorio musical. Se permite la navegación, acceso y transmisión en streaming de las canciones para cualquier tipo de aplicación musical.



Figura 3.5. Partitura de la obra Terminus compuesta por Iamus

Capítulo 4: Entorno, programas y formatos

4.1 Entorno

El entorno de desarrollo utilizado para la realización del proyecto es MATLAB, un entorno pensado para el desarrollo de algoritmos, la creación de modelos, el análisis de datos, la visualización y el cálculo numérico.

MATLAB^[20] fue elegido para el desarrollo de los algoritmos de Melomics, sobre todo debido a su facilidad de uso para el cálculo matricial y el prototipado rápido. Por las mismas razones y por facilitar la integración del desarrollo correspondiente a este PFC se decidió continuar usando el mismo lenguaje. Otros aspectos que también influyeron en el uso de MATLAB son el útil sistema de ayuda, de fácil uso y con gran cantidad de ejemplos de uso, la posibilidad de compilación y ejecución tanto en Windows como en sistemas Unix y la existencia de multitud de *toolboxes*, propietarias y de la comunidad de desarrolladores, que facilitan la operación sobre matrices, resuelven problemas algorítmicos concretos o facilitan la ejecución de los programas en sistemas de supercomputación.

Los algoritmos de este PFC se ejecutan en sistemas muy diferentes entre sí: ordenadores personales, de un solo núcleo o multinúcleo, con sistemas operativos Windows y Linux, y en clústeres de computación con sistema Debian. Para el desarrollo en computadoras personales se hace uso del *toolbox* de MATLAB para computación paralela. Mientras que para el desarrollo en el clúster se trabaja con el gestor de recursos TORQUE^[21], ya sea mediante llamadas al sistema o a través del *toolbox* de MATLAB, el cual gestiona la comunicación con TORQUE.

4.1.1 Parallel Computing Toolbox

Es un *toolbox* (caja de herramientas) de MATLAB que permite resolver problemas computacionales y de datos intensivos haciendo uso de procesadores multinúcleo, GPUs y clústeres de ordenadores. Permite construcciones de alto nivel, como bucles *for* en paralelo, tipos especiales de matrices, y los algoritmos numéricos paralelizados, que permiten ejecutar en paralelo aplicaciones de MATLAB sin programación MPI o CUDA. El *toolbox* ofrece hasta doce trabajadores (motores computacionales Matlab) para ejecutar aplicaciones de forma local en un ordenador de sobremesa multinúcleo.

Otra característica del *toolbox* es que soporta los planificadores de TORQUE, por lo que se puede comunicar y manejar directamente sin ningún tipo de configuración extra con sistemas manejados con TORQUE.

4.1.2 Torque

Es un gestor de recursos que proporciona control sobre los trabajos por lotes (*batch*) y los recursos de computación distribuida. Es un producto de código abierto basado en el proyecto PBS^[22] original. Su uso incorpora avances significativos en las áreas de escalabilidad, fiabilidad y funcionalidad. Torque puede integrarse junto con Maui^[23] para mejorar la utilización global, la programación y la administración en un clúster.

4.1.3 Maui

Es un planificador avanzado utilizado para mejorar el control y la eficiencia de los recursos del clúster y el supercomputador. El planificador de Maui es un motor de políticas que permite a los sitios control sobre cuándo, dónde y cómo los recursos tales como procesadores, memoria y disco se asignan a los puestos de trabajo. Además de este control, también proporciona mecanismos que ayudan a optimizar de forma inteligente el uso de estos recursos, controlar el rendimiento del sistema, facilitar el diagnóstico de problemas, y en general administrar el sistema.

4.2 Programas externos

Para determinados procesos del software desarrollado el entorno no proporcionaba soluciones suficientemente óptimas o bien se requería de un desarrollo completo, por ello se buscaron programas externos que realizaran estos procesos.

Tras las pruebas pertinentes de cada programa externo éste se descartó o se integró en el código mediante la llamada `system` del entorno. Los siguientes programas que se mencionan son aquellos sobre los que se realizaron las pruebas para determinar su uso.

4.2.1 Lilypond

Como se define en la web del software, LilyPond^[24] es un programa de grabado musical (tipografía musical o edición de partituras), consagrado a la producción de partituras de la calidad más alta posible. Lleva la estética de la música tipografiada de la forma tradicional a las partituras impresas mediante ordenador. LilyPond es software libre y forma parte del Proyecto GNU. Pero lilypond no se reduce solo a crear la partitura, sobre todo desde las mejoras incluidas en las últimas versiones, también se puede usar para obtener un archivo MIDI correspondiente a la partitura.

En el software desarrollado se usa para obtener los archivos MIDI de cada instrumento así como el de la obra completa. Lilypond al ser un programa por línea de comandos que interpreta texto es propicio para la generación de la pieza musical de manera automática.

4.2.2 Timidity

Es un sintetizador software, es decir, su misión es transformar un archivo MIDI en un archivo de onda PCM. Para ello hace uso de archivos *SoundFont* con los sonidos a usar por cada nota e instrumento (paletas de instrumentos, en capítulo 7.2), lo que comúnmente se conoce como síntesis mediante tabla de ondas.

La necesidad de usar un programa como Timidity^[25] es debido a que proporciona multitud de opciones, como son generar archivos de audio estéreo o habilitar y configurar efectos como reverberación o coro según la configuración del *SoundFont*. Además reduce en gran medida el tiempo de desarrollo ya que transformar de archivo de notación musical MIDI a archivo de audio no es trivial.

4.2.3 SoX

Sound Exchange (SoX) es una utilidad multiplataforma por línea de comandos que puede transformar distintos formatos de audio a otros. También puede aplicar una gran cantidad de efectos a estos archivos de sonido, así como reproducir y grabar archivos de audio en la mayoría de plataformas.

Con SoX^[26] se realizan todas las funciones de procesamiento de audio requeridas para realizar la mezcla, como son la ecualización, panorama o reverberación. También se usa para conversión entre formatos de audio.

4.2.4 Ecasound

Es un paquete software diseñado para procesado de audio multicanal. Su funcionalidad es similar a la de SoX, se puede usar desde simplemente como reproductor o grabador a añadir efectos de sonido y mezcla. Es la primera opción que se manejó, pero finalmente se descartó su uso debido a su incompatibilidad con sistemas Windows.

4.2.5 FluidSynth

Es un sintetizador software basado en las especificaciones *SoundFont 2*. Su función es obtener archivos de audio a partir de archivos MIDI. Se descartó su uso debido a que el funcionamiento sobre sistemas Windows y la documentación del software era de inferior calidad a la de Timidity.

4.3 Herramientas externas adicionales

Además de las opciones que proporciona el entorno y de los programas externos usados para determinados procesos del software durante la realización del proyecto, se usaron varios programas adicionales para realizar distintas tareas necesarias como la edición o conversión de archivos *SoundFont*, o para realizar pruebas de mezcla y sonido. A continuación se hace mención a los recursos más importantes utilizados para cada tarea.

4.3.1 Extreme Sample Converter

Es un programa de edición y conversión de formatos de muestras de sonido (*samples*).

Durante el proyecto se usa para la conversión, tanto directa entre distintos formatos (formato destino *.sf2*) como para la conversión *VSTi*, en la cual se programa qué sonidos se necesitan crear en el archivo *sf2* y el software reproduce el instrumento *VSTi* y lo graba haciendo uso del driver de sonido ASIO.

4.3.2 Translator 6

Es un programa similar al *extreme sample converter*^[29], pero a diferencia de éste algunas de sus funciones no están incluidas en la aplicación básica, como la conversión de *VSTi* que debe ser adquirida aparte, motivo por el que se descartó su uso.

4.3.3 Viena

Es un editor de *SoundFont* y con el que se crean y editan las distintas paletas de instrumentos utilizadas por el software desarrollado. Con él se hace corresponder a cada instrumento MIDI unos *samples* de audio para cada nota o rango, así como editar distintos valores propios del formato *sf2* en el que se establece cómo hacer sonar el *sample* para cada situación.

4.3.4 Mixcraft

Mixcraft 6 es un programa para producción musical y grabación multi-pista de gran alcance. Permite grabar audio, organizar bucles, remezclar pistas, componer con instrumentos virtuales y MIDI, añadir efectos, y mezclar y masterizar composiciones musicales.

El uso de esta aplicación se hace para conocer los pasos que se llevan a cabo por un usuario al hacer una mezcla y probar valores de manera manual y gráfica que es el modo más común y extendido de producción.

4.3.5 Synthfont

Programa similar a Mixcraft para edición de audio pero con muchas menos opciones que éste. Las ventajas respecto a Mixcraft es que es gratuito y está especializado en el uso de archivos *sf2*, además de integrar Viena para poder editarlos de manera directa.

Este software es usado para el chequeo de los distintos instrumentos en formato *sf2* aplicando efectos de manera manual y obtener distintas configuraciones válidas para los distintos estilos musicales.

4.4 Formatos

A lo largo del proceso del software desarrollado se van haciendo uso u obteniendo distintos archivos en distintos formatos entre los que destacan:

- MusicXML^[31]. Es un formato de notación musical abierto basado en XML. Fue desarrollado por Recordare LLC y fue diseñado para el intercambio de partituras. Es soportado por multitud de programas de notación musical. En nuestro software es el primer archivo en un formato estándar que se genera desde la entrada. La extensión de archivo es *.xml*.
- Lilypond. Software de notación musical para creación de partituras y reproducción de estas en formato MIDI. La extensión de los archivos de Lilypond es *.ly* y estos archivos se obtienen mediante un conversor proporcionado por el propio programa de *.xml* a *.ly*. Estos archivos obtenidos son editados posteriormente para obtener los archivos MIDI adecuados en cada caso.
- MIDI (Interfaz digital de instrumentos musicales). Es un protocolo de comunicación serial estándar para comunicar y compartir información para la generación de sonidos. En este formato se escribe la pieza musical completa permitiendo su audición mediante sonidos sintetizados y pudiéndose editar desde cualquier programa de edición musical, así como individualmente cada instrumento para obtener el archivo de audio final. La extensión de los archivos MIDI suele ser *.mid* en sistemas Windows y *.midi* en sistemas Unix.
- OSC^[32] (*Open Sound Control*). Es un protocolo diseñado para compartir información musical entre instrumentos, ordenadores y otros dispositivos multimedia a través de la red. Hizo aparición con la intención de sustituir al viejo formato MIDI, ya que sus capacidades son superiores en muchos aspectos. A pesar de todo, el protocolo MIDI sigue ostentando una repercusión mucho mayor a nivel de usuario, motivo por el cual se descartó la implementación de dicho formato.
- SoundFont. Son archivos que contienen muestras grabadas de audio de diversos instrumentos musicales. *SoundFont* es una marca registrada de *E-mu Systems*^[33] y la licencia exclusiva para volver a formatear y administrar el contenido *SoundFont* histórico ha sido adquirida por *Digital Sound Factory*^[34]. La extensión de archivo es *.sf2* y estos ficheros son usados para convertir archivos MIDI con sonido sintetizado a archivos WAV con sonidos grabados de mayor calidad.
- WAV. Formato de audio digital normalmente sin compresión que se utiliza para almacenar sonidos en el ordenador. Admite archivos mono y estéreo a diversas resoluciones y velocidades de muestreo. En nuestro software es el primer archivo de sonido que obtenemos al conjuntar cada instrumento en formato MIDI con la paleta de instrumentos (*sf2*). Sobre estos archivos de extensión *.wav*

se realizan las modificaciones y efectos de sonido, al no tener pérdida de información.

- MP3. Es un formato de compresión de audio digital que usa un algoritmo con pérdida de información para conseguir archivos de menor tamaño. Actualmente es el formato de audio más extendido, existiendo reproductores para automóviles, ordenadores y dispositivos móviles. Con el MP3 se consigue una reducción de tamaño de aproximadamente diez veces respecto al WAV. La extensión de estos archivos es *.mp3*.
- FLAC (*Free Lossless Audio Codec*). Como sus siglas indican es un códec libre de compresión de audio sin pérdida de calidad. La diferencia de tamaño del archivo FLAC respecto al WAV es de entre la mitad y tres cuartos del tamaño original. La extensión de archivo de este formato es *.flac*.

Capítulo 5: Arquitectura del software

5.1 Introducción

Durante el desarrollo del capítulo se explicará la arquitectura del software, es decir, el diseño de la estructura del sistema al más alto nivel. Para ello se mostrará la estructura básica y los motivos principales de ésta, así como los principales problemas que se abordaron durante el desarrollo y que afectaron en mayor o menor medida al desarrollo de la arquitectura final. Para terminar se comentarán los distintos pasos llevados a cabo para la optimización del programa y su motivación, junto con una serie de conceptos previos necesarios.

5.2 Estructura básica

La estructura básica del programa es la que puede contemplarse en la figura 20. Como se observa, la estructura se divide en dos grandes bloques: la generación y la síntesis. Cada uno con un objetivo marcado, el primero obtener el archivo MIDI que representa a la pieza musical y el segundo obtener el archivo de audio final de dicha pieza musical. Para ello se crea una subestructura que realiza estas dos tareas de forma individual para cada instrumento. Esta división por instrumentos es provocada por la forma en que debe realizarse una mezcla de audio, en la que se deben configurar primeramente los instrumentos de forma individual y posteriormente unirlos y tratarlos en conjunto.

La estructura del software está determinada por el proceso de interpretar la notación musical simbólica proporcionada como entrada de la música a reproducir. Esta interpretación se realiza en la fase de generación, en la que se llevan a cabo una serie de transformaciones, como la humanización, previas a la obtención del archivo MIDI. Una vez obtenidos los archivos MIDI se pasa al bloque de síntesis donde se realiza toda la edición de sonido y mezcla. Ambos bloques se detallarán en sus respectivos capítulos posteriormente.

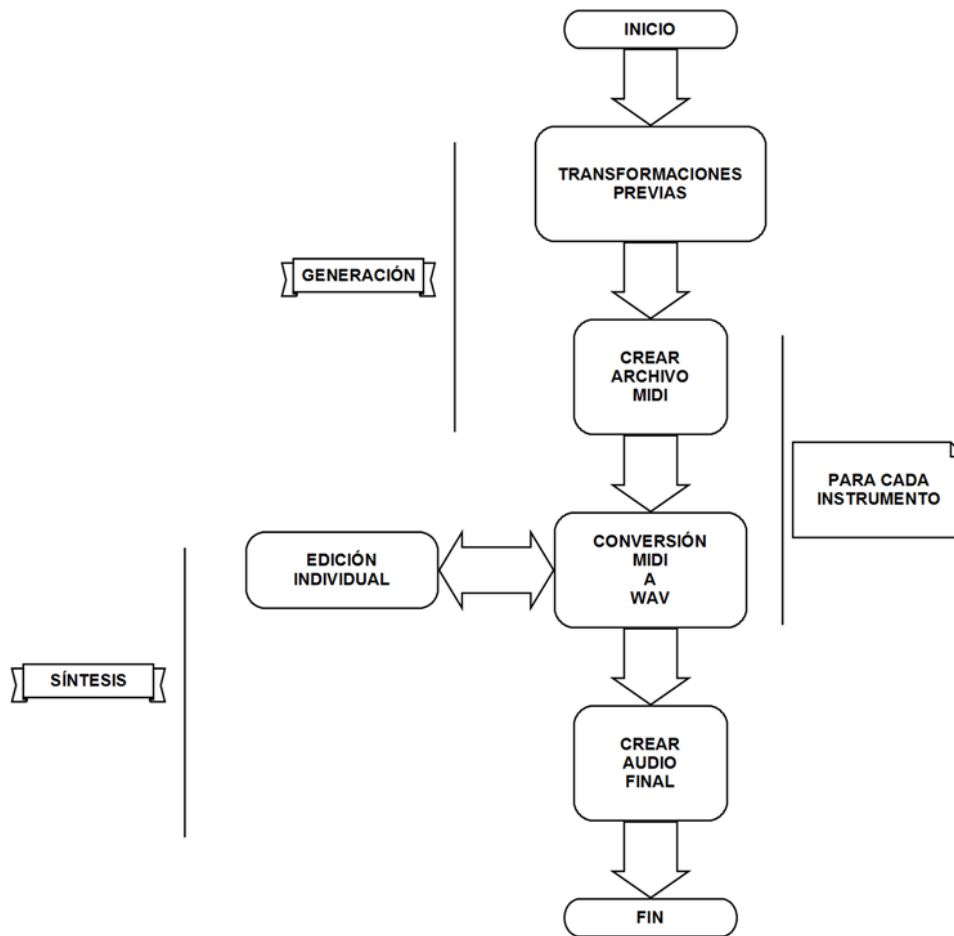


Figura 5.1. Estructura básica

5.3 Notación musical simbólica

Para describir la pieza musical a interpretar se usa una notación musical simbólica, con la cual se suministra al software toda la información necesaria, desde el instrumento a usar a la duración y altura de cada nota. Con esta notación se representan las características del sonido descritas en el capítulo segundo: intensidad, duración, altura y timbre. A estos valores se le añade información sobre efectos, articulaciones, tempo y otras especificaciones necesarias. Toda esta información está parametrizada en una serie de matrices en las que cada fila correspondería a cada instrumento y cada columna a cada nota a lo largo del tiempo interpretada por cada instrumento. Todo ello conforma la notación musical simbólica que el software interpreta para reproducir en sonido la pieza musical acorde a ello.

La información de entrada al programa a la que llamamos notación musical simbólica es aquella que aparece en la llamada a la función principal de la aplicación:

`emulateSong(theme, durations, figures, tags, nMeasures, pitches, velocities, instruments, refDur, insGroup, melParams)`

- `pitches` o matriz de tonos. En ella aparecen números entre 0 y 127 que indican la nota a tocar, es decir, se especifica la propiedad de altura. Cada valor del rango equivale a una nota, alteración y octava según el estándar MIDI.
- `velocities` o matriz de intensidad. Especifica el valor de dinámica, es decir, la intensidad del sonido. Al igual que con las alturas la intensidad se establece con un valor entre 0 y 127 siendo el 0 silencio y 127 la máxima intensidad posible. Teniendo en cuenta que en música la intensidad se declara por niveles de dinámica, estos son traducidos a un valor dentro del rango para cada nivel y estableciendo el porcentaje de volumen. Por ejemplo *f* o *forte* es el nivel base con valor 90 de velocidad o intensidad que corresponde a un 70% del volumen máximo y se considera el nivel de volumen normal.
- `durations` o matriz de duraciones. Indica la duración de una nota y si ésta es activa, es decir, suena o bien es un silencio. Los valores de esta matriz son limitados ya que en la música cada nota tiene una duración determinada y una representación. Estos valores se consiguen a partir del valor de referencia que corresponde a la duración de la negra, y a partir de ahí se construyen todas las duraciones posibles según están definidas en la teoría musical. El valor numérico de la duración está indicado en divisiones que es una medición simbólica, en lugar de en segundos que es una medida dependiente del tempo. Para saber el tiempo real en segundos de una nota se divide el tiempo de dicha nota en divisiones y se divide por el tempo multiplicado por diez.
- `instruments` o vector de instrumentos. Dicho vector representaría la última propiedad del sonido necesaria, el timbre. Lo hace mediante un número que identifica el instrumento que suena.
- `figures` o matriz de figuras. En ella se proporciona la cadena de texto identificativa de cada símbolo de duración de la nota musical.
- `insGroup` o vector de grupo. Indica el grupo instrumental al que pertenece cada instrumento. Es un dato usado para reconocer instrumento de un mismo grupo en una obra orquestal. Por ejemplo los instrumentos de cuerda o de viento metal.
- `nMeasures` o vector de compases. Proporciona el número de compases que individualmente tiene cada instrumento. Con ello se conoce cuándo se puede dejar de analizar un instrumento ya que a partir de dicho compás sólo existirá silencio hasta que finalice la interpretación de todos los instrumentos.
- `tags` o matriz de etiquetas. Se indican los distintos efectos, articulaciones o adornos musicales activos, el tempo, los inicios y finales de compás y demás información extra necesaria para cada nota y momento.
- `refDur`. Duración de referencia a partir de la cual se han construido los valores de duración.

Los datos con mayor incidencia a la hora de interpretar de forma correcta una pieza musical son los que conforman las características del sonido, duración, timbre,

intensidad y altura, junto con las etiquetas indicadoras de efectos relacionados con la ejecución específica de una técnica interpretativa así como los valores de tempo.

En la llamada de la función principal existen dos valores que no han sido comentados, `theme` que indica el nombre dado a la pieza musical y a los archivos de audio finales, y `melParams` que es una estructura de datos de configuración como pueden ser el estilo musical o los archivos de audio que se quieren obtener. Los distintos parámetros de dicha estructura serán explicados en detalle en el apéndice.

Los valores que se indican a la entrada no son los valores finales ya que algunas etiquetas hacen modificar estos valores, como pueden ser el *crescendo* que modifica las velocidades o el *glissando* que añade notas extras.

5.4 Límite de instrumentos

Uno de los problemas que había al empezar el proyecto y objetivo a solucionar por el mismo es que el emulador que se estaba utilizando, al trabajar directamente sobre el protocolo MIDI, no podía conseguir archivos de melodías con más de 16 instrumentos (en realidad 15+1 de percusión). Esto hacía inviable obtener los archivos de sonido correspondientes a obras de orquesta, que contaban con unos treinta instrumentos de media, además del problema de si había más de un instrumento de percusión.

Primeramente se buscó si existía algún protocolo distinto al MIDI para la escritura musical. Un sustituto sopesado fue OSC (*Open Sound Control*). Éste protocolo carecía del problema con el límite de instrumentos, pero su menor difusión provocaba un problema posterior, la conversión a archivo de audio, encontrándose solo software disponible para conversión de OSC a MIDI perdiéndose por lo cual las ventajas que se pudieran obtener del mismo. Además los programas más comunes de edición de sonido trabajan sobre MIDI. Dichos motivos produjeron su descarte.

Tras asumir el empleo del protocolo MIDI existían 2 cuestiones a resolver: hallar la mejor forma de obtener un MP3 con más de los quince instrumentos que permite MIDI y cómo crear dichos MIDI. Para la segunda cuestión existían tres opciones de desarrollo:

- Realización desde cero de un programa que escribiera directamente el MIDI. El principal problema de ésta opción era el desconocimiento del protocolo a bajo nivel, por lo que el tiempo de desarrollo sería muy elevado.
- Mejorar conversor MusicXML a MIDI. Éste pertenece a un reproductor de código abierto en Java con una opción para la conversión, a la cual se accede a partir de un *hack* que permite el uso de dicha opción mediante línea de comandos. El problema de esta opción es que el reproductor carece de soporte

en la actualidad por lo que habría que adaptarlo a las versiones de MusicXML y MIDI que se requieren.

- Lilypond. Dicho software era usado para crear las partituras y en su última versión se proporcionaba una nueva opción para obtener directamente un archivo MIDI de la partitura, así como una librería para remodelar la partitura (dejaba el *pdf* de la misma inservible) mejorando sustancialmente la calidad del MIDI, haciendo posible la reproducción de algunos adornos y articulaciones. Tras algunas pruebas se observó que la calidad de estos MIDI superaban a los del resto de opciones, por ejemplo con el conversor de MusicXML a MIDI solo se podían obtener notas normales sin ningún adorno o técnica. Por todo ello pasó a ser la opción prioritaria.

Una vez decidida la continuación en el uso del protocolo MIDI, quedaba el problema de generar archivos MIDI de más de 15+1 instrumentos. La solución fue resuelta por la propia generación de MIDIs con Lilypond, debido a que éste permite crear MIDIs con cualquier número de instrumentos. El procesado para obtener estos MIDIs con gran cantidad de instrumentos es excesivamente lenta y además incumple el estándar MIDI lo que provocaba que no pudiera asegurarse el uso del MIDI por otros programas. La solución fue crear varios MIDI y sus respectivos audios y unirlos una vez eran convertidos a audio, ya que existían varios programas que permitían la opción de mezclar archivos de audio. Para ello hubo que decidir cómo dividir dichos archivos lo que daría lugar a la estructura básica del software, que concluyó con la creación de MIDIs con el máximo número de instrumentos posibles. Esta opción creaba entre dos y tres archivos MIDI y su consiguiente archivo de audio para la mezcla posterior. Fue la primera opción que se probó y resolvió el problema, pero el proceso de conversión de MIDI a audio así como la creación del archivo MIDI seguía siendo demasiado lenta. Surgió otro problema al interpretar algunos efectos como el pizzicato, cuya única manera de reproducirlos es tratar dichas notas como otro instrumento, lo que se hace duplicando dicho instrumento (proceso `dupChannel`, en capítulo 6.3.1), lo que dificultaba la elección de en qué instrumento establecer el límite, así como elevando el número de archivos MIDI y WAV, aumentando el tiempo de proceso. El principal motivo de que no terminara siendo la solución definitiva es que era muy complicado añadir ciertos efectos de procesado de audio (ver capítulo 7) que se querían añadir a cada instrumento para aumentar la calidad del audio. Para ello se modificó la estructura obteniéndose un archivo de audio de cada instrumento. Este archivo de cada instrumento se consigue usando la estructura desarrollada anteriormente, es decir, duplicar tantas veces el instrumento como sea necesario teniendo en cuenta que si se sobrepasan los quince de tope del archivo MIDI hay que dividir igualmente, aunque aquí ya es menos probable. La figura 5.2 muestra una representación gráfica de la estructura descrita.

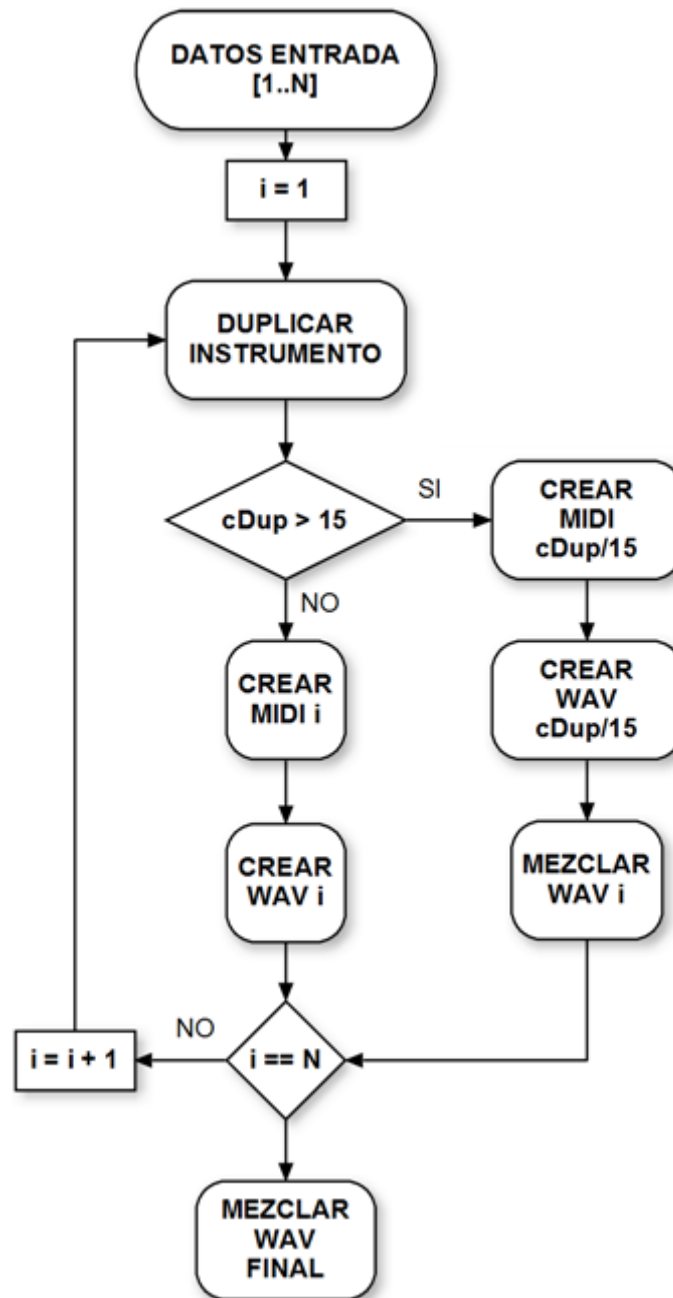


Figura 5.2. Bucle para generación audio final, siendo cDup la cantidad de duplicados de cada instrumento

Sólo puede haber un instrumento de percusión por MIDI. Los instrumentos de percusión se pueden clasificar en dos categorías:

- De altura definida. Aquellos que producen notas identificables. Es decir, aquellos cuya altura de sonido está determinada. Algunos de ellos son el timbal, el xilófono, el vibráfono o la campana.
- De altura indefinida. Aquellos cuyas notas no son identificables, es decir, producen notas de una altura indeterminada. Entre ellos están el bombo, la caja, las castañuelas o la batería.

La manera normal de representar los instrumentos de percusión de altura indefinida o sin afinar es como un set de percusión en el que el sonido particular de cada instrumento se asocia a una nota, por ejemplo un set podría estar representado por un bombo en la nota Do de la segunda octava y en el Re de la cuarta octava los platillos, interpretándose estos sonidos cada vez que se ejecute dicha nota. Estos son aquellos definidos en el estándar MIDI como percusión, pero no todos están presentes, por lo que habría que sustituirlos por alguno de ellos. Igualmente para aquellos de altura definida o que puedan producir más de un sonido no podrían ser representados por lo que la solución no sería completa. Otra opción es considerar cada instrumento de percusión como un instrumento normal, opción que recomienda Lilypond para obtener mejores resultados, y limitar la tesitura al número de sonidos posibles con dicho instrumento, teniendo así instrumentos con una sola nota. Para solucionar esto se modifica la altura del instrumento de percusión para hacerla acorde al archivo *sf2* que lo representa y así conseguir el sonido deseado. Otro problema que aparece para poder llevar a cabo dicha opción es que a la entrada del software pueden aparecer instrumentos de percusión separados o bien unidos en un set. Esto es debido a que en las partituras se suelen escribir unidos en un set porque los ejecuta un mismo músico o porque varios sonidos de percusión componen el instrumento (batería), así que se identifica el instrumento y si es un set se separa en instrumentos individuales.

Toda la estructura establecida coincide con los pasos que se aplican durante el proceso de mezcla en un estudio de grabación, los cuales se comenzaron a estudiar con posterioridad para la mejora sustancial de la calidad del audio creado.

5.5 Volúmenes

El volumen es un parámetro que depende de diversos factores. En un principio se obtiene del valor de velocidad o intensidad de cada nota, definido por el estándar MIDI, que puede tener un valor entre cero y ciento veintisiete. El valor de cero es común a cualquier instrumento, ya que equivale al silencio, pero el volumen máximo generado por un instrumento u otro no es el mismo. Esto se debe a que aunque se interpreten dos instrumentos a la misma intensidad, cada uno puede producir un nivel de decibelios diferente, acorde a las características tímbricas de cada uno. Por ello la obtención del sonido está dividida en dos fases, el de cada nota respecto a otra en un mismo instrumento, controlado por los datos de intensidad o velocidad y el de cada instrumento respecto a los otros observado en los decibelios o voltios que genera cada uno.

La primera fase se desarrolló aplicando los matices dinámicos (forte, mezzo piano,...) que aparecen tanto en las partituras como en la notación simbólica, determinando un valor en la matriz de velocidad. El desarrollo de esta fase estuvo muy definido por el desarrollo y aplicación del crescendo y disminuyendo, incremento o disminución

progresiva de la intensidad. Primeramente se probó la aplicación del crescendo y disminuyendo desde dos ángulos distintos, cada uno con sus inconvenientes:

- Directamente con la definición de Lilypond. Para obras con notas simples su funcionamiento era correcto, sin embargo con determinadas notas y efectos se creaban de forma incorrecta requiriendo edición manual. Además en los casos en los que intervenían en un mismo crescendo notas normales y notas que requerían de una duplicación del canal (por ejemplo pizzicato), hacían este método de desarrollo inviable.
- Modificando directamente el sonido del archivo de audio. Esto se hace con una función de MATLAB para leer archivos WAV (`wavread`) en la que se muestrea la señal y se obtiene un vector de valores de amplitud del archivo. Para ello previamente se creó el WAV teniendo todas las notas del crescendo o disminuyendo implicado al valor máximo de intensidad y se almacena en una variable si se trata de un crescendo o un disminuyendo, los valores máximos y mínimo de intensidad y el momento de inicio y fin. Para calcular dichos momentos de inicio y fin hay que tener en cuenta el tempo que tiene la obra en dicho instante. Con estos datos se crea un vector de crescendo que tendrá el tamaño correspondiente según la frecuencia de muestreo. El punto del vector donde se inicia el crescendo o disminuyendo se calcula multiplicando el instante de tiempo inicial por el tempo, obteniéndose la muestra exacta donde comenzar. De la misma forma se calcula el fin, y la diferencia entre ellos para la longitud. Con ello el vector crescendo será un incremento progresivo desde el valor mínimo de volumen al máximo, concretamente como se quiere obtener el valor proporcional que se debe atenuar la onda en cada momento irá desde el cociente entre valor mínimo y máximo de intensidad a 1, ya que la nota más fuerte no requiere atenuación ninguna. Si fuera un disminuyendo se aplica el vector crescendo invertido. El resultado de este proceso era de muy buena calidad pero con la mejora de las paletas instrumentales e introducción de diversos adornos y articulaciones musicales, al convertir del archivo MIDI al WAV este podía crecer unos segundos en tiempo debido a distintos efectos que alargan las colas del sonido, lo que provocaba que se descompasaran los distintos instrumentos.

Se había llegado a un punto muerto en el desarrollo, y además había que sumar la opción de humanizar dichos valores de intensidad. Por ello se buscó una fórmula totalmente distinta en la que pudieran aplicarse los crescendos y disminuendos, y además poder humanizar la música. Tras estudiar más en detalle las opciones de Lilypond se halló una forma para declarar nuevos valores dinámicos y asignarles un valor. Para ello se construyó una función que se aplicaba a la estructura matricial en la que se añadía a la matriz de intensidades o velocidades el incremento o decremento progresivo correspondiente y posteriormente con la humanización un pequeño error en cada nota (ver capítulo 6.2). Tras esto se tenía cada nota con un valor de intensidad real por lo que se añadieron al archivo de MusicXML etiquetas de dinámicas para cada valor distinto de intensidad (de 0 a 127). Estas etiquetas eran creadas automáticamente al pasar al

archivo de Lilypond, pero salvo las etiquetas estándar de dinámica (*f*, *p*, *mf*, ...) con un valor por defecto de velocidad, el resto de nuevas etiquetas de dinámica no poseían un valor de intensidad concreto. Para solucionar esto se edita el archivo *ly* añadiendo una función en la que se proporciona a cada etiqueta nueva un valor de dinámica concreto. Con esto se consigue tanto tener una música humanizada en la que no suenan todas las notas exactamente igual y la existencia de crescendo y decrescendos en las obras, todo ello aplicado en el archivo MIDI, lo que permite poder usar archivos *sf2* de mayor calidad, que no solo contengan sonidos distintos para cada nota sino también para valores de *velocity* distintos.

El código de la función mediante la cual se definen las etiquetas y sus valores de *velocity* es:

```
#(define (myDynamics dynamic)
  (if (equal? dynamic "z") 0
      (if (equal? dynamic "pp") 0.39
          (if (equal? dynamic "jp") 0.44
              (if (equal? dynamic "p") 0.47
                  (if (equal? dynamic "f") 0.71
                      (if (equal? dynamic "fzz") 0.73
                          (if (equal? dynamic "fff") 0.79
                              (default-dynamic-absolute-volume dynamic))))))))))

z = #(make-dynamic-script "z")
jp = #(make-dynamic-script "jp")
fzz = #(make-dynamic-script "fzz")
```

Como se puede observar, no es necesario crear las etiquetas estándar (*f*, *fff*, *p*, ...), pero sí se asignan valores a todas, ya que éste no tiene por qué coincidir con el valor por defecto que les asigna Lilypond, aunque si no se expresa este cambio explícitamente se usará el valor por defecto.

5.6 Optimización

Uno de los objetivos del proyecto es conseguir archivos de audio con la mayor eficiencia y brevedad posible. Debido a la forma en que se realiza la mezcla, donde primero se trata cada instrumento por separado y posteriormente se combinan, y a que es la parte lenta de todo el proceso, se consideró paralelizar esta fase para reducir los tiempos. Para ello se investigó sobre los distintos sistemas en los que sería ejecutado el

software y la manera de obtener un uso óptimo de los mismos. Estos sistemas son procesadores multinúcleo y el clúster de computación Iamus.

En los siguientes subapartados se introducirán una serie de conceptos básicos sobre computación paralela y los distintos tipos de paralelismo y de sistemas paralelos existentes. También se mostrarán las distintas opciones de optimización que se probaron.

5.6.1 Computación paralela

En la computación paralela, a diferencia de la tradicional computación en serie, se ejecutan instrucciones de manera simultánea, en paralelo. Para ello se requieren de sistemas hardware con arquitecturas de más de un procesador.

Dentro de estas arquitecturas existen dos categorías:

- Sistemas fuertemente acoplados: son sistemas donde los procesadores comparten su información mediante memoria compartida, externa y ajena a todos los procesadores.
- Sistemas débilmente acoplados: son sistemas donde no existe memoria compartida y los procesadores necesitan comunicarse los unos con los otros para compartir la información de su memoria local.

La programación paralela es la técnica de programación usada en la computación paralela. En la programación paralela se divide un problema en varias partes independientes, de manera que cada proceso pueda realizar la ejecución de las instrucciones que se le han asignado sin depender de los demás procesos del programa. De esta forma, un algoritmo se puede ejecutar de forma más rápida y eficiente que si se ejecutara en un solo proceso.

La programación paralela trae consigo nuevos problemas como son las condiciones de carrera (problemas de ejecución de instrucciones en las cuales no se puede avanzar en la ejecución hasta que otra instrucción actualice un recurso requerido). Además, las aplicaciones son más difíciles de implementar ya que la sincronización y la comunicación son más complejas. Pero, a pesar de ello, la mejora en cuanto a incremento de velocidad y eficiencia es considerable, y hace que sea necesaria y muy utilizada en la actualidad.

Teóricamente, si uno dobla el número de procesadores, el tiempo de ejecución debería reducirse a la mitad, y si se dobla el número de procesadores sucesivamente, entonces se debería acortar a la mitad el tiempo de ejecución. Pero todo programa consta de una parte que no se puede paralelizar por lo que difícilmente se alcanzan estos valores ideales. Para averiguar la aceleración que se puede alcanzar en un programa existen dos leyes muy ligadas entre sí. La ley de Amdahl dice que la mejora obtenida en el

rendimiento de un sistema debido a la alteración de uno de sus componentes está limitada por la fracción de tiempo que se utiliza dicho componente, cuya fórmula original corresponde a la ecuación 1.

$$F = F_a \left((1 - F_m) + \frac{F_m}{A_m} \right) \quad \text{Ecuación 1}$$

$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}} \quad \text{Ecuación 2}$$

Donde F es el tiempo de ejecución mejorado y F_a es el tiempo de ejecución antiguo. La fórmula de la ley de Amdahl también puede reescribirse como en la ecuación 2 haciendo uso de la definición de incremento de velocidad ($A = F_a/F$). Donde A es la ganancia en velocidad conseguida en el sistema completo debido a la mejora de uno de sus subsistemas, A_m el factor de mejora que se ha introducido en el subsistema mejorado y F_m la fracción de tiempo que el sistema utiliza el subsistema mejorado.

La otra ley de computación es la ley de Gustafson en la que indica que el incremento de velocidad con P procesadores es:

$$S(P) = P - \alpha \cdot (P - 1)$$

Donde P es el número de procesadores, S es el *speedup*, y α la parte no paralelizable del proceso.

Existen varias clasificaciones de tipos de paralelismo según qué característica se tenga en cuenta para hacer la clasificación. Si lo clasificamos según la frecuencia con que sus subtareas se sincronizan o comunican entre sí, existen tres tipos:

- Paralelismo de grano fino. Sus subtareas deben comunicarse muchas veces por segundo.
- Paralelismo de grano grueso. Las subtareas se comunican pocas veces por segundo.
- Vergonzosamente paralelo. Nunca o casi nunca se tienen que comunicar la subtareas.

También existen tres modelos de paralelismo atendiendo a la arquitectura utilizada para la solución del problema. Estos paralelismos son:

- Paralelismo por resultado. Se descompone el resultado que se pretende de un programa en varios subresultados, permitiendo que cada parte pueda calcular los resultados de forma simultánea e independiente. Se centra en la forma en la que se termina el resultado que, en ocasiones, tiende a ser una compleja estructura cuyos elementos se pueden calcular en paralelo.

- Paralelismo por especialización. A cada trabajador disponible se le encarga llevar a cabo una clase de tarea, realizando todos ellos su trabajo en paralelo, y siguiendo las restricciones que imponga el problema. Se centra en la composición del equipo de trabajo de la aplicación.
- Paralelismo por agenda. Cada trabajador ayuda con la tarea actual de la agenda, y se ejecutan en paralelo simultáneamente con las restricciones que imponga el problema. Se centra en una lista de tareas a ser realizadas por cualquiera de los procesos trabajadores. Dichos trabajadores no son especialistas en un determinado tipo de tarea, sino que son todos iguales (realizan una tarea, la resuelven y devuelven el resultado).

Existen muchos tipos de computadoras paralelas, pero nos centraremos únicamente en los dos tipos a los que se ha tenido acceso en el desarrollo del proyecto y para los cuales se han realizado pruebas y optimizaciones. Estos dos tipos son:

- Procesadores multinúcleo. Son aquellos que combinan más de una unidad central de procesado por chip. En estas unidades es donde se leen y ejecutan las instrucciones de programa. Esto permite que el multinúcleo o *multicore* pueden ejecutar instrucciones de manera simultánea, pudiéndose realizar paralelismo a nivel de hilos (*thread-level parallelism* o TLP). Este tipo de paralelismo en un solo chip se conoce como multiprocesamiento a nivel de chip (*chip level multiprocessing* o CMP).
- Clúster. Es un tipo de ordenador distribuido, formado por un conjunto de ordenadores débilmente acoplados y unidos mediante una red de alta velocidad. Un clúster puede ser de diferentes tipos atendiendo a los equipos que lo conforman:
 - Homogéneo. Todos los equipos tienen la misma configuración de hardware y sistema operativo.
 - Semihomogéneo. Los equipos tienen diferente rendimiento pero tienen arquitecturas y sistemas operativos similares.
 - Heterogéneo. Los equipos tienen hardware y sistemas operativos diferentes.

En un clúster se espera que haya una combinación de servicios entre alto rendimiento, alta disponibilidad, balanceo de carga y/o escalabilidad.

5.6.2 Estrategias de optimización

Buscando la manera de optimizar la ejecución, se probaron cuatro opciones de optimización usando paralelización, cada una con sus ventajas y desventajas. El objetivo es minimizar el tiempo empleado en obtener un número determinado de canciones sintetizadas. A continuación se relatan cada una de estas estrategias,

describiendo en detalle su funcionamiento y repercusión. En el capítulo de pruebas se muestran los resultados obtenidos con cada una de las opciones.

5.6.2.1 Estrategia 1: Paralelismo en SoX

Durante la ejecución se lanzan múltiples llamadas mediante `system` para invocar al programa SoX y realizar procesado de sonido, ya sea para mezclar los instrumentos, ecualizar o añadir algún efecto. Por ello se actualizaron las llamadas a SoX añadiéndole la etiqueta `multi-thread` la cual permite a SoX realizar el procesado de audio haciendo uso de todos los procesadores disponibles. Esto crea una mejora considerable cuando el procesado de audio requerido es pesado, lo cual solo ocurre en el proceso de mezcla de una cantidad considerable de instrumentos, como puede ser en obras de orquesta, o instrumentos con gran cantidad de etiquetas de interpretación que provocan la duplicación del instrumento.

5.6.2.2 Estrategia 2: Paralelismo a nivel de instrumento

La parte más lenta y crítica de la ejecución es obtener cada instrumento por separado y configurarlo de la manera requerida, llevándose a cabo mediante un bucle del tipo vergonzosamente paralelo, debido a que los instrumentos no necesitan información de los restantes. Por lo cual se cambia el bucle `for` por un bucle `parfor` haciendo uso de una llamada previa a `matlabpool`. Esta opción obtiene un beneficio siempre y cuando haya más de un instrumento. El uso de `matlabpool` acarrea un problema cuando se usa en el clúster debido a que hace que Torque, gestor de colas del clúster, desconozca qué procesadores están siendo usados en el `parfor`, incurriendo en que a un mismo procesador pueda llegar más de una tarea ralentizando el proceso general. Por ello hay que realizar el bucle haciendo uso de Torque ya sea directamente mediante `qsub` o a través del gestor de tareas de matlab.

En realidad el bucle no es vergonzosamente paralelo debido a que la información sobre el tempo se encuentra en el primer instrumento, por lo que para poder aplicar la estrategia se probaron dos métodos diferentes:

- Sin modificar la información. Se ejecuta el primer instrumento donde está la información del tempo y se proporciona esta información al resto de instrumentos los cuales se ejecutan en paralelo.
- Modificación previa. Se aplica previamente al bucle donde se construyen los instrumentos una función llamada `repairTempo` donde se obtienen los distintos cambios de tempo que ocurren y el compás en el que se producen del primer

instrumento, añadiéndose dicha información en los inicios de compás correspondientes del resto de instrumentos.

Este segundo método convierte el bucle de instrumentos en vergonzosamente paralelo y provoca mejores resultados, por lo que se implementó en la versión final.

5.6.2.3 Estrategia 3: Paralelismo de instrumentos y mezcla

Se usan la estrategia uno y dos en conjunto. Esta opción añade las mejoras de cada una de las opciones, ya que se pueden ir ejecutando varios instrumentos independientes en cada procesador y a su vez obteniendo mejoras al procesar el audio con SoX. En esta opción se observó que en algunas de las ejecuciones el proceso se ralentizaba en vez de acelerarse, siendo causado a que tanto MATLAB en el bucle `parfor` como SoX necesitan hacer uso del mismo procesador lo que producía un retardo por la comunicación extra generada al tener que compartirse. Se solucionó eliminando la opción `multi-thread` de SoX para los instrumentos individuales, dejando esta opción sólo para la mezcla final y el procesado sobre el audio final.

5.6.2.4 Estrategia 4: Paralelismo a nivel de canción

La estrategia se centra en la obtención de una gran cantidad de resultados lanzando el programa completo en paralelo. Esta opción es muy útil al ejecutarse en un clúster debido a que la gran cantidad de procesadores, hace que cada uno obtenga un resultado, el cual es independiente del obtenido en otro procesador. Estos lanzamientos se realizan mediante la llamada `qsub`, ejecutándose en cada procesador una ejecución completa del programa totalmente serializado, es decir, sin hacer uso de ninguna de las opciones anteriores. Al igual que en la estrategia 2 cuando no se está en el clúster esta estrategia se aplica mediante un bucle `parfor` dentro del cual se realiza la llamada para la generación de un tema musical diferente.

Capítulo 6: Generación

6.1 Introducción

Al ser uno de los principales bloques del software, durante este capítulo se explicarán los procesos más importantes que se llevan a cabo en él.

El bloque de generación abarca desde que se proporciona la información sobre la composición de la pieza musical al programa hasta que se obtiene el archivo MIDI con la melodía, tanto los individuales de cada instrumento necesarios para la conversión a audio, como el archivo MIDI completo de la canción solicitada requerido a la salida. Dicho bloque define el procesado previo y generación de la música a reproducir, ya sea humanizando parámetros, modificando notas para crear los adornos y técnicas musicales que se desean o preparando cada instrumento para su correcta sintetización posterior.

En el transcurso del proceso de generación se obtienen archivos *xml* (MusicXML) y *ly* (Lilypond) retocados para una generación de archivo MIDI y síntesis posterior correcta, siendo eliminados dichos archivos una vez concluida su función. Una vez concluido todo el procesado se crea un archivo MIDI de la pieza musical completa.

6.2 Humanizar

La Real Academia Española define la acción de humanizar como, "hacer humano, familiar y afable a alguien o algo". En nuestro caso ese algo es la música. El motivo de querer humanizar la música está marcado por el objetivo de conseguir un audio lo más parecido a la interpretación humana. La música en sí a la que está acostumbrado el ser humano es imperfecta. Esto es así debido a que el propio humano es el que la ejecuta y aunque como dice el dicho popular "la práctica hace al maestro", hasta el mejor de los maestros difícilmente es capaz de interpretar una obra completa ejecutando cada nota siempre al mismo nivel exacto de intensidad, por ejemplo. Por ello lo que convierte en mejor o peor a un músico es el número de errores grandes que comete, tendiendo este número a cero conforme mejor sea el músico y pudiéndose ver esto comparando un músico experimentado con uno novel. Lo que diferencia un músico de otro son la multitud de pequeños errores que cometen y asumen de manera natural. Una manera sencilla de observar dicha situación es en una actuación en vivo de cualquier grupo musical, donde las horas de práctica consigue que la ejecución carezca de errores notables, pero donde siempre aparecerán pequeños fallos al llevar a cabo una nota más corta o larga o reproducirla un poco más fuerte que otra, definiendo el conjunto de

imperfecciones la interpretación musical humana. La música ejecutada o interpretada directamente por un ordenador se percibe de forma demasiado perfecta o artificial para el oído humano, motivo por el cual existe el proceso de humanización de la música por ordenador.

La manera de humanizar, con el fin de que el sonido reproducido por el ordenador se perciba más humano, es introducir pequeñas imperfecciones y cambios. Un método para averiguar dónde insertar los errores y en qué cantidad consistiría en la comparación entre una grabación MIDI de una interpretación humana, objetivo a emular, y otra compuesta por ordenador. Los fallos o diferencias que se observaran con mayor probabilidad serían:

- *Timing o note on*. Se produce cuando comienzan las notas, ya que es poco probable que se inicien en el momento exacto siempre. Además si son varios instrumentos, éstos también deben acompañarse.
- Duración de la nota. Para un ordenador la reproducción exacta de una nota durante el tiempo indicado por la notación musical se realiza con gran precisión, siendo más variable en la interpretación humana.
- Tono. Este fallo en un buen músico sería muy complicado de percibir ya que difícilmente reproduciría un tono superior o inferior. Aunque hay instrumentos en los que ésta probabilidad es mayor por su complejidad de ejecución.
- Intensidad. El parámetro MIDI que lo representa es el *velocity* y mide la fuerza con la que se interpreta una nota. Cometer fallos en intensidad es muy común en los instrumentos de percusión, por la dificultad de golpear el instrumento con la misma fuerza.

Hay multitud de plugins para los editores de composición musical que trabajan sobre algunos de estos cuatro parámetros. Normalmente se humaniza el *velocity*, en algunos casos también se juega con el *timing* y la duración y raramente con el tono.

La humanización de la composición musical se crea sumando un error aleatorio al valor original. La generación de números aleatorios se realiza mediante curvas de distribución, aunque las más comunes son:

- Curva uniforme. Es la forma más simple y se caracteriza porque todos los números tienen la misma posibilidad.
- Curva de Gauss o de campana. Es el método más próximo a la realidad de una interpretación humana, debido a que los pequeños errores son muy comunes, mientras que es inusual cometer un fallo grave.

Todos los programas encontrados y probados fueron plugins de distintos editores de composición o aplicaciones externas, los cuales se caracterizan por poseer un entorno gráfico de trabajo, dificultando su adaptación para su uso de manera automática desde nuestro software y sin poder asegurar un buen resultado. Por ello se decidió crear una función que realizara una humanización de los parámetros como modificación previa a la información de la composición musical que se proporciona como entrada del

software, viéndose reflejada la humanización directamente en el archivo MIDI y a través de éste en el audio final.

La función de humanizar desarrollada en el software realiza modificaciones sobre los parámetros:

- Intensidad, con un parámetro de entrada para regular el límite del error. Dicha función da un resultado bastante bueno debido a que se crea un MIDI en el que cada nota posee un valor de velocidad o intensidad. La función añade un error a cada dato de la matriz de *velocity*, salvo en los casos de silencio y nunca dejando en silencio una nota activa.
- Duración. Al igual que para la intensidad se añade un pequeño error limitado por un parámetro de entrada para modificar la duración de una nota. Implícitamente al modificar la duración se modifica el *timing* ya que si se hace durar más a una nota la siguiente comenzara a sonar mas tarde. Esto afecta en sobremanera en obras con más de un instrumento ya que se descompasan con facilidad resultando el audio similar a si tocara la obra una banda sin experiencia ni practica. Debido a ello quedó postergada su mejora para futuras versiones de desarrollo y siendo utilizada únicamente en casos muy concretos.
- Tono. Esta opción finalmente no fue implementada en el software debido a que el fallo de tono no es común en músicos experimentados o profesionales.

Los errores cometidos por un músico pueden cuantizarse y aproximarse mediante funciones de distribución. Por lo que si se conociera dicha distribución de error de algún músico en particular podría usarse en la función de humanización simulando así que dicho músico interpreta la obra o pieza musical. Existen varios estudios y publicaciones sobre cómo distribuir dichos errores y cómo medirlos, pero actualmente la mayoría del software existente suele seguir por defecto una distribución normal, que suele ser la que más se acerca a la realidad. Por ello en nuestro software los errores provocados siguen una curva de Gauss.

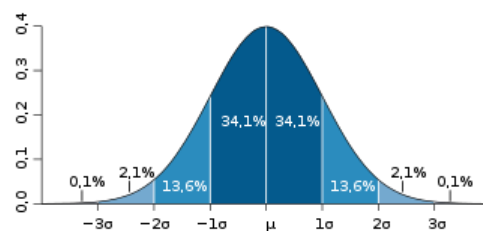


Figura 6.1. Curva de Gauss

A efectos prácticos es una distribución normal con media cero, no cometer error, y estableciendo cómo valor de desviación típica la tercera parte del parámetro de entrada. Un ejemplo de uso sería el de la tabla 6-I.

Rangos	% de los datos
$(-2, 2)$	68,26
$[-2, -4) \cup [2, 4)$	27.18
$[-4, -6) \cup [4, 6)$	4.3
$[-6, -\infty) \cup [6, \infty)$	0.26

Tabla 6-I. Ejemplo de distribución normal con parámetro de entrada 6

Al ser el *velocity* un parámetro que debe ser entero se decidió utilizar solo la parte entera de los resultados, para provocar menores fallos y que éstos fueran los más pequeños posibles.

Para dar más realismo a la interpretación, previamente a la aplicación de la humanización de la intensidad, se aplica una función que modifica dichos parámetros de intensidad para simular valores más cercanos a los que se producen cuando se interpreta un instrumento. Esto se debe a que generalmente al ejecutar un instrumento se sigue una pauta o secuencia en las que se reproduce una nota fuerte y la siguiente algo más débil o viceversa, repitiéndose la secuencia durante toda la pieza musical. La llamada a dicha función consta de los siguientes campos:

- *velocity*. Vector con los valores de intensidad, los cuales son modificados y se devuelven como resultado.
- *durations*. Vector con las duraciones de cada nota y un marcador de nota activa.
- *max* y *min*. Rango máximo y mínimo de intensidad a usar en el instrumento.
- *sequence*. Vector numérico que indica la secuencia de intensidad a seguir, correspondiendo el valor 1 con el rango máximo y el mayor valor del vector con el mínimo.
- *time*. Duración de la secuencia, repitiéndose nuevamente la secuencia transcurrido dicho tiempo si la pieza musical no ha concluido.

La función se divide en dos partes, en la primera se crea un vector de intensidades con todos los valores posibles dentro del rango establecido por *max* y *min* siguiendo la secuencia de entrada, y en la segunda se establece el valor de intensidad de cada nota según su duración y el momento en que se ejecuta, siempre sin modificar los silencios y reiniciando la secuencia una vez transcurrido el tiempo proporcionado a la entrada hasta la conclusión de la interpretación de cada instrumento de la pieza musical.

6.3 Recursos interpretativos

Llamamos recursos interpretativos a toda indicación de la notación musical de cómo tocar una nota o serie de notas durante el transcurso de una pieza musical. Pueden distinguirse claramente tres tipos de indicaciones que comunican al intérprete que debe ejecutar una nota de una manera distinta a la habitual:

- Articulación que indica la forma en que se produce el cambio de un sonido a otro o bien sobre la misma nota. Se trata del conjunto de elementos que definen las diferentes posibilidades en las que se pueden conectar entre sí las notas que componen una melodía. Estas posibilidades se diferencian básicamente en función de tres elementos: el ataque de cada nota, la caída de cada nota y el grado de interrupción o continuidad del sonido existente entre las distintas notas.
- Adorno u ornamento musical que son florituras que no son necesarias para llevar la línea general de la melodía (o armonía), pero que sirven para decorar esa línea. Se trata de recursos que pueden ser utilizados en las composiciones con el objeto de imprimirles a éstas expresión, ornamento, variedad, gracia o vivacidad.
- Efecto que es el sonido que se consigue al emplear una técnica interpretativa que hace sonar al instrumento de una manera particular distinta al sonido habitual.

Se han contemplado tres formas de conseguir la ejecución y efecto sonoro de las articulaciones, adornos y efectos:

- Interpretación directa. Son las técnicas más sencillas de llevar a cabo ya que Lilypond es capaz de interpretarlas y proporcionar un MIDI con el recurso en cuestión ejecutado, teniendo únicamente que asegurar la correcta escritura del recurso en la notación de Lilypond. La inclusión de la biblioteca de Lilypond “articulate” aporta un buen número de nuevas interpretaciones directas. Un ejemplo de interpretación directa puede ser la ejecución de legato, que indica que las notas se toquen sin interrupción.
- Interpretación por duplicación. Se lleva a cabo para aquellos recursos que producen un sonido peculiar o propio, y que para poder ser llevado a cabo necesitan un sonido distinto o modificado del original. Para ello se necesita un archivo *sf2* con el instrumento ejecutando el efecto sonoro considerado, y aplicándose únicamente a las notas que correspondan, duplicando el instrumento. En estos casos puede ocurrir dos cosas, que una leve modificación del sonido del *sample* (sonido de la nota grabado) se obtenga el *sf2* necesario para conseguir el efecto buscado, como ocurre en el acento, en el que se ataca la nota con mayor intensidad que de normal, o bien, que sea necesario tener un conjunto de *samples* propios al igual que con el sonido estándar para poder reproducirlo.
- Interpretación por desarrollo. Son casos en los que la notación del efecto comunica una base desde la que desarrollar la ejecución. Para llevar a cabo estos recursos se detecta el momento en que se producen y se modifica la información de entrada para que aparezca en ella el recurso interpretativo ya desarrollado según corresponda, en lugar de las indicaciones necesarias para ser ejecutado. Ejemplos de este caso son el arpeggio que provoca que un acorde se ejecute como notas individuales en lugar de todas las notas al mismo tiempo, y el glissando que se ejecuta reproduciendo una determinada sucesión de notas, siendo indicadas en la notación únicamente la nota de inicio y fin del mismo.

6.3.1 Interpretación por duplicación

Estas técnicas que generan un sonido característico dependen para su aplicación, en su mayoría, de poseer una paleta de instrumentos con dicho efecto para el instrumento en concreto correspondiente. En el siguiente capítulo se entrará más en profundidad sobre paletas instrumentales, aquí nos centraremos en los cambios necesarios en la generación para que en la síntesis se pueda hacer un correcto uso de ellas y así mejorar el audio final.

Para poder hacer uso de estos efectos hay que comunicar al programa de qué efectos se posee en la paleta para cada instrumento, información que se transmitirá en el archivo de parámetros (`melParams.emuTemplate = {etiqueta, id instrumento, id instrumento duplicado}`) que se proporciona como entrada. Tras esto el emulador para cada instrumento comprobará si existe alguna etiqueta de efecto, articulación y/o adorno musical que posea *samples* de sonido propios. De ser así buscará si en algún momento de la obra el instrumento hace uso de ella y duplicará el instrumento en caso afirmativo, es decir, duplicará la fila correspondiente al instrumento en todas las matrices de la estructura. Con éste único cambio se interpretará el instrumento por duplicado, en un caso con el sonido normal y en otro con el sonido con el efecto. Evidentemente este no es el objetivo. Para ello se silenciaron las notas con efecto en el instrumento original y las notas normales en el instrumento con efecto. Este cambio reproducía el sonido deseado en cada nota, pero perdiéndose la relación entre las notas normales y las que tenían el efecto ya que se habían convertido en dos instrumentos distinguibles. Para evitar esto en lugar de sustituir las notas por silencios, se cambia su intensidad, parámetro *velocity*, a cero, por lo que la nota existe pero no se ejecuta. La diferencia radica en que ahora sí se tiene en cuenta el resto de efectos que pueden indicar su relación con las notas colindantes, quedando dos instrumentos exactamente iguales sonando en uno las notas con un sonido y en el otro las restantes con otro sonido diferente. La sensación que se consigue es la existencia de un solo instrumento ejecutado por un único intérprete, el objetivo buscado.

En la función de duplicación existe la posibilidad de cambiar el instrumento a utilizar de forma automática y así poder realizar pruebas de depuración reproduciendo de forma rápida una obra con distintos instrumentos de una misma paleta instrumental sin tener que crear una nueva, tarea mucho más lenta.

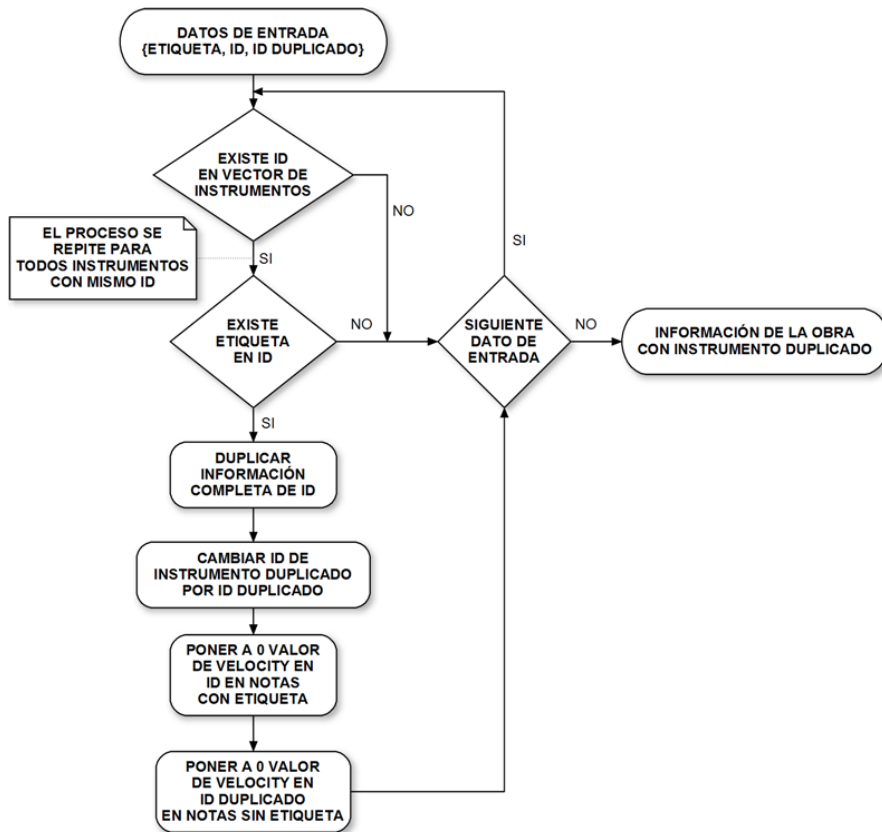


Figura 6.2. Proceso de interpretación por duplicación

6.3.2 Interpretación por desarrollo

Aquellos recursos interpretativos que requieren de un cambio en la información de la composición para que ésta aparezca con el efecto traducido, pudiéndose tratar como notas simples o mediante el uso o combinación de otros recursos que no necesitan de un desarrollo previo. Los recursos que han sido implementados mediante este método son el glissando y el arpeggio.

6.3.2.1 Glissando

El glissando es un adorno musical que significa resbalar y consiste en transitar rápidamente de un sonido hasta otro más grave o viceversa, ejecutándose todos los

sonidos intermedios. Estos sonidos intermedios dependerán del instrumento en sí. En la información de la composición musical que se recibe como entrada viene señalado en la matriz de etiquetas (tags) con la etiqueta 'gliss' indicando el momento en el que iniciar un glissando y finalizando en la siguiente nota, a partir de la cual se continúa de manera normal. Para ello la duración de la nota marcada con glissando se divide en tantas partes como tonos existan entre ellas. Primeramente hay que hallar las notas que se ejecutan para cada instrumento. Por ejemplo, si el instrumento carece de aquellos tonos correspondientes a notas 'negras' en un teclado correspondientes a las notas sostenidas, éstas se eliminan. En cambio en el caso de que el instrumento posea ambos tipos, 'blancas' y 'negras', entonces la nota marcada establece el tipo a usar, si la marcada corresponde a una tecla 'blanca' se eliminan todas las 'negras' y viceversa. Tras ello se añaden todas las notas intermedias que existan entre la marcada u origen y la nota adyacente o destino, estableciendo como duración de la nota origen y las intermedias el valor obtenido de dividir la duración original por el número de notas que conforman el glissando.

A continuación hay que asegurar que la primera nota del glissando se ejecuta respecto a la nota anterior de la manera correcta. Para poder hacerlo hay que comprobar si existe legato, portato, ligaduras o continuación de algún otro efecto como por ejemplo el tremolo. Tras ello, para simular que se tocan todas las notas seguidas rápidamente, se establecen como si estuvieran en legato desde el origen al destino, es decir se ejecutarán sin pausa una tras otra. Por último queda asegurar, al igual que con la nota origen del glissando, que la nota destino también se comporta correctamente con la siguiente. Para ello se han ido pasando todos los efectos que había en la nota origen a las notas intermedias y se han aplicado también y al llegar a la nota destino solo hay que tratarla como una nota normal teniendo en cuenta el legato extra añadido al glissando para simular correctamente su funcionamiento.

6.3.2.2 Arpeggio

El arpeggio, derivado del italiano arpeggio, es una forma de ejecutar un acorde. Un acorde son varias notas ejecutadas simultáneamente. Sin embargo, en el arpeggio se reproducen dichas notas en una sucesión rápida.

El arpeggio es un recurso que se utiliza frecuentemente en instrumentos de cuerda pulsada como la guitarra, aunque se usa también con otros instrumentos. Es un recurso muy común en el arpa, de ahí el nombre de arpeggio. El arpeggio se usa en diferentes estilos musicales, en algunos como en electrónica pueden llegar a ser el recurso básico. Esto provoca que en la mayoría de sintetizadores para música electrónica contengan un arpegiador, el cual crea arpeggios aleatorios a partir de un acorde y diversas opciones, como la duración o restricciones sobre los cambios de octava. Imitando esta funcionalidad se desarrolló un arpegiador para nuestro software.

Los arpeggios más básicos se interpretan reproduciendo todas las notas del acorde, del más grave al agudo o viceversa, pero hay muchas más formas de arpeggios con un mismo acorde.

El arpegiador desarrollado en el proyecto estaría situado, según la estructura básica, en la parte de modificaciones o transformaciones previas, a diferencia del glissando, cuya transformación se efectúa directamente al analizarse la etiqueta indicadora del mismo durante la creación de los archivos necesarios para obtener el MIDI. Esto se debe a que los arpeggios sencillos sí se pueden escribir con una notación específica y son fáciles de añadir tanto a la partitura como al MIDI, pero al haberse añadido más opciones de arpeggios, estos se deben escribir desplegados en la notación, que es una forma equivalente de escribirlos, y necesaria para su correcta interpretación en el archivo MIDI, como ocurría en el glissando.

Los arpeggios vienen codificados en la matriz de etiquetas. Una nota puede tener en sus etiquetas correspondientes las etiquetas 'chord' y 'arpegio' cada una seguida de un número. El número de 'chord' identifica las notas del acorde, lo normal suelen ser tres notas de la misma octava, y el de arpegio apunta a un arpegio codificado en un archivo de texto plano auxiliar. Estos arpeggios tienen una estructura determinada:

- Tienen una duración comprendida entre un cuarto de compás y un compás completo (600-2.400 divisiones).
- Pueden existir hasta doce notas diferentes y se permiten las repeticiones.
- Solo pueden existir corcheas, semicorcheas y/o tresillos de ambas. La corchea y la semicorchea son las siguientes notas menores a la negra, por lo que duran la mitad y un cuarto de la duración de ésta respectivamente (parámetro de entrada `refDur` indica la duración de una negra). Un tresillo son tres notas interpretadas en la duración de solo dos de ellas, dando una impresión de rapidez.
- Las notas pueden ir con acento, es decir al interpretarse se atacan con más intensidad que de normal.
- Algunas notas pueden ser silencios, aunque esto es poco probable y nunca pueden ser consecutivos.
- Las notas pueden estar una octava por encima o por debajo, aunque si esto ocurre en una nota, en las adyacentes no ocurre.

Toda esta información viene codificada en el archivo auxiliar, por lo que se seleccionará el arpeggio correspondiente y se decodificará. En la decodificación surgen dos cuestiones ya que la duración y el número de notas diferentes del arpeggio pueden ser distintos a la duración del arpegio en la obra y al número de notas del acorde a usar. La cuestión de la diferencia de notas tiene una solución simple, ya que están permitidas las repeticiones de notas y además también puede ocurrir que no se usen todas las notas del acorde, así que se hacen corresponder varias notas del arpegio a cada nota del acorde. Para solucionar el problema de las duraciones hay que tener en cuenta que la reproducción de una estructura musical debe ser repetible, es decir no puede obtenerse un audio distinto en cada ejecución de una misma obra, y que los tresillos son indivisibles. Por ello si el

arpegio es de menos duración que lo deseado se replica el arpegio y si es mayor se corta. Si al cortar o replicar la última nota es mayor que lo deseado se reduce su duración, se cambia de corchea a semicorchea, y si acaba en una nota intermedia de un tresillo se selecciona la primera nota del mismo y se transforma en corchea o semicorchea según corresponda.

Lo último que queda es organizar los campos de la matriz de etiquetas, ya que por ejemplo se puede tocar el arpegio usando trémolos, por lo que hay que añadir las etiquetas según corresponda a cada efecto, para que se sepa dónde empieza y acaba el efecto según se relacione con notas anteriores o posteriores. También hay que añadir la etiqueta de legato para que las notas del arpegio se ejecuten sin pausas como indica la definición de un arpegio. Para terminar se añaden las etiquetas de triplete o tresillo para que puedan aparecer tanto en la partitura como en el MIDI con su duración correcta. Dos tresillos consecutivos es un seisillo por lo que si esto ocurre se escribe como tal, aunque a efectos prácticos son equivalentes y suena totalmente igual.

En la definición de arpegio se decía que tenían que ejecutarse rápidamente, esto se consigue permitiendo solo notas de duración corta. Si se desea mayor velocidad pueden aparecer arpegios con solo semicorcheas y tresillos de semicorcheas, lo que da una sensación de mayor velocidad.

Capítulo 7: Síntesis

7.1 Introducción

La síntesis es el segundo bloque en el que se ha dividido el software, en él se traduce la información musical generada a un archivo de audio, haciendo uso de la técnica de tabla de ondas y posteriormente se mezclan los distintos instrumentos según el estilo musical para conseguir un archivo de audio de la mejor calidad y sonoridad posible.

El punto principal de la síntesis es la mezcla, de ésta depende que el resultado final sea una canción o melodía y no un conjunto de sonidos superpuestos y consecutivos, por ello se le dedicó gran parte del tiempo al proceso de investigación. En el apartado correspondiente se explicará qué características influyen en la mezcla y cómo se ha conseguido aplicar cada característica al audio.

7.2 Paletas de instrumentos

Las paletas de instrumentos se usan para la conversión de los archivos MIDI a audio digital usando la técnica de tabla de ondas. Son bancos de instrumentos donde cada uno tiene asociado un número MIDI (en MIDI se definen 128 instrumentos, del 0 al 127) y posee uno o más archivos asociados a una nota o rango de ellas. Al convertir de MIDI a audio digital (WAV) se usan estos sonidos pregrabados como una tabla de ondas en los que se hace corresponder cada nota con el sonido correspondiente. A esta conversión se le llama síntesis mediante tabla de ondas.

Uno de los formatos más comunes para las paletas es el *SoundFont* (traducido fuentes de sonido). Dicho formato fue creado por *E-mu Systems* y *Creative Labs* para usarlas con las tarjetas de sonido *Sound Blaster*. Actualmente dicho formato es compatible con todas las tarjetas y se puede usar en la mayoría de programas de composición musical, principalmente de software libre. También se pueden encontrar multitud de archivos *sf2* en internet, de autores y compositores que han grabado distintos instrumentos y los han compartido de manera libre y gratuita^[44]. El resto de formatos tanto abiertos como propietarios se usan a través del sistema *VSTi*, permitiendo su uso a través de programas de edición de entorno gráfico que actúan de servidores. Estos instrumentos *VSTi* no pueden ser usados de forma automática mediante línea de comandos, debido a que requieren del entorno creado específicamente para su uso. Estos motivos hicieron que se optara por la conjunción de *Timidity* y *SoundFont*. También existen conversores de paletas para transformar de un formato a otro, pudiendo así utilizar gran cantidad de sonidos virtuales que de otra forma quedarían apartados.

Durante el desarrollo del proyecto se pudo contar con el programa de edición musical Kontakt, el cual posee multitud de paletas de gran calidad y que se pueden convertir a formato *.sf2* usando un programa llamado ESC (*Extreme Sample Converter*). En el apéndice se muestra un manual de cómo hacer esta conversión.

Los archivos *SoundFont*, al igual que la mayoría de formatos para paletas instrumentales, no se limitan únicamente a relacionar cada nota con un ejemplo de audio, sino que poseen muchas más variables y parámetros para su configuración. Entre ellas destacan los osciladores. Existen osciladores de varios tipos, de envolvente o de bajas frecuencias (LFO) por ejemplo, ambos tanto de volumen como de modulación. Con ellos se informa de cómo debe reproducirse el audio usando seis parámetros fundamentales que dan forma a la onda. Estos parámetros son similares en todos los osciladores. Se detalla ahora su funcionamiento para la envolvente de volumen:

- Delay. Es el retraso previo que debe transcurrir antes de ejecutarse el sonido.
- Attack. Cuando el 'delay' concluye comienza a subir el volumen hasta su volumen máximo. Esta subida se produce en el tiempo indicado por el parámetro de ataque o 'attack'. Debido a que el oído humano es sensible a las subidas de volumen este parámetro es usualmente bajo (50-300ms) lo que provoca que sea susceptible a filtros como el de corte.
- Hold. Con este parámetro se indica cuánto tiempo se debe mantener el sonido a máximo volumen tras el tiempo de ataque.
- Sustain + Decay. Son dos parámetros distintos con una funcionalidad ligada. El

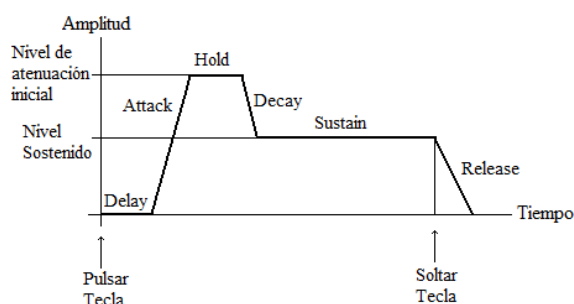


Figura 7.1. Parámetros fundamentales para dar forma a una onda sonora (DAHDSR)

parámetro 'decay' indica el tiempo durante el cual el volumen decae, pasado el tiempo de 'hold', desde el máximo valor al nivel de sostenimiento, es decir al nivel de decibelios marcado por el parámetro 'sustain'. Cuando el volumen llega al valor de 'sustain' se mantiene en el mientras se esté reproduciendo la nota.

- Release. Cuando se libera la tecla del instrumento, entra en juego la fase de liberación, es decir el volumen baja desde el valor de volumen actual hasta el valor cero en el tiempo marcado por el parámetro 'release'. Esta fase empieza cuando se libera la tecla en un piano, se finaliza de frotar la cuerda de un violín, o en el caso de un MIDI cuando concluye la duración de la nota, por lo que el resto de parámetros dejan de intervenir si se estaban aplicando o no se aplican.

La envolvente de modulación se usa para cambiar dinámicamente el tono (*pitch*) o el filtrado de frecuencias (*filter*). El oscilador de bajas frecuencias posee los mismos modos de funcionamiento siendo usado en mayor medida en instrumentos de percusión.

Existen también otras variables para configuración más específicas, en concreto para afinar los tonos o notas y de efectos. Entre los de efectos hay parámetros para filtrado de frecuencias, reverberación, coro y panorama.

Otra posibilidad de la que se dispone es la creación de bucles o *loops* para cada onda de sonido. Con esta opción se puede reproducir una onda de 4 segundos de duración de forma ilimitada, ya que lo que se le indica al programa con ello es un punto de inicio del bucle y de final y en el caso de que la onda llegase al punto final de bucle pero el mensaje MIDI indicara que la nota a reproducir es de mayor duración, el periodo comprendido por el bucle se repetirá el número necesario hasta completarse la duración. El uso de bucle está muy extendido en el caso de instrumentos realizados con sintetizadores.

Todos los parámetros y opciones se pueden manejar desde un editor de *SoundFont*. Para construir y configurar los *preset* o paletas de instrumentos necesarios durante el proyecto, se usó el programa Viena. En el apéndice de la memoria se proporciona un manual de cómo crear un *preset* de sonidos.

7.3 Mezcla

Una posible definición de mezcla es el proceso por el cual material multipista es balanceado, tratado y combinado en un formato multicanal, normalmente doble canal o estéreo. En nuestro software este material multipista serían los archivos de audio (WAV) creados a partir de archivos *SoundFont* correspondientes a cada instrumento, siendo tratados, balanceados y combinados hasta obtener un archivo estéreo final.

Con la mezcla se intenta conseguir dar dinamismo al sonido para provocar en el oyente diferentes sensaciones. Existen multitud de libros sobre mezcla y cómo llevarla a cabo, pero la mayoría coinciden en explicar los distintos dispositivos hardware y técnicas de procesado de audio para conseguir que el sonido produzca las sensaciones que se estén buscando. Para simular dichos sistemas y técnicas se hace uso de un programa de procesamiento de audio llamado SoX con la intención de cumplir dichas expectativas.

La manera elegida para realizar la mezcla en nuestro software, divide el proceso en cuatro partes o características y a su vez cada una de éstas en tres niveles de aplicación. Las características son:

- Volúmenes
- Panorama
- Ecuación

- Efectos de sonido basados en el tiempo

En los subapartados siguientes se detallará cómo afectan y se aplican cada una de ellas en cada uno de los tres niveles posibles de aplicación. Estos niveles están ordenados de menor a mayor según la intensidad de las sensaciones que provocan y son:

- Nivel 1 o configuración individual. Es la diferencia entre cada nivel individual o la configuración de cada equipamiento. Cualquier cambio que se realice sobre los sonidos de una mezcla crearán una dinámica o energía que afectarán en menor medida que un cambio sobre el patrón general. Ejemplos pertenecientes al nivel serían aumentar o disminuir el volumen de la voz principal, aplicar reverb sobre un tambor, o situar al bajo en una posición centrada con respecto al oyente o totalmente a la derecha.
- Nivel 2 o patrón general. Es el patrón que se obtiene al configurar una de las características en toda la mezcla, es decir la suma de todas las configuraciones individuales para cada característica. Por ejemplo, crear una diferencia mayor o menor entre el máximo y mínimo volumen de los instrumentos de una mezcla provoca una sensación de dinamismo mayor que elevar o disminuir un único instrumento. Otro ejemplo es construir una mezcla con multitud de efectos o con ninguno, lo cual crea también un cambio mayor de intensidad que si un único sonido tiene efectos o no.
- Nivel 3 o configuración dinámica. Es el movimiento creado al modificar la configuración de una característica durante la mezcla. Ejemplos del nivel son cambiar el panorama de un instrumento durante un periodo de tiempo creando sensación de movimiento, o modificar la relación de volúmenes de varios sonidos cambiando así los roles de cada instrumento en el transcurso de una canción. Estos cambios producen gran dinámica, pero no siempre son deseables debido a que provoca una presencia mayor de la mezcla en la canción.

Dependiendo de hasta qué nivel de aplicación e intensidad se haya alcanzado se definen dos tipos de mezcla:

- Mezcla invisible. Se usa solo el primer nivel y se utiliza en estilos en los que se requiere una mezcla transparente, no audible, como en el jazz acústico.
- Mezcla visible. Se alcanza al aplicar el tercer nivel y provoca que la mezcla sea un componente más de la canción. Es muy común en estilos como dance o techno y grupos como Pink Floyd la llevaron al extremo.

En la siguiente tabla se muestra una interesante perspectiva de cómo se clasifican las distintas dinámicas a crear según el efecto emocional que producen a través de la música.

Tipos de sentimientos creados	Tipos de dinámicas que pueden crearse	Características	Tipos de dinámicas que pueden crearse	Tipos de sentimientos creados
<u>Emociones Suaves</u> ordenado estructurado divertido gótico estable normal romántico equilibrado simple	Poca variación entre sonidos, sucesivos y de distintos instrumentos EQ natural entre todos los instrumentos, encajan bien sin modificar Equilibrado, colocación simétrica	Volumen Ecuilización Panorama	Fuerte variación entre instrumentos y secciones de la canción EQ interesante entre los instrumentos, se centra en resaltar un sonido peculiar Desbalanceado, colocación anti simétrica	<u>Emociones Intensas</u> interesante excitante salvaje divertido creativo loco nuevo desequilibrado complejo
<u>Valores positivos</u> calor paz amor seguridad ambiente centrado	Seco, EQ clara, instrumentos sin efectos, espacio entre los sonidos	Efectos	Húmedo, EQ completa, instrumentos con retardos y engordados, muro de sonido	<u>Valores positivos</u> diversión catarsis creatividad intriga perspectiva aparente
<u>Valores negativos</u> aburrido banal comercial simple	Uso de compresión para limitar el movimiento de los sonidos	Estabilización vs Movimiento	Cambios de panorama, volumen, EQ y efectos	<u>Valores negativos</u> frenético loco molesto psicótico anormal

Tabla 7-I. Dinámicas creadas con el equipo clasificadas por emociones

7.3.1 Volúmenes

Simplemente situando cada instrumento en un nivel de volumen diferente o cambiando el volumen durante la mezcla, se pueden crear dinámicas musicales y emocionales totalmente diferentes. Para conseguirlo se hace uso del fader, el compresor-limitador y las puertas de ruido (*noise gate*). Este último es usado para controlar el ruido existente al grabar los distintos instrumentos, en nuestro caso, al usar instrumentos virtuales, se carece de dicho problema y no se ha implementado el dispositivo. En los siguientes subapartados se detallará el funcionamiento tanto del fader como del compresor-limitador, los distintos niveles de aplicación y cómo se llevó a cabo su implementación en el software.

7.3.1.1 Fader

En ingeniería de sonido se conoce como *fade* al incremento o decremento gradual de la señal de audio. El fader es el dispositivo para llevarlo a cabo y consiste en un controlador de voltaje con el que amplificar o atenuar la señal.

En el nivel 1 de configuración con los fader se establece la cantidad de volumen para cada instrumento en relación al resto de los sonidos en la mezcla. La dinámica musical creada al situar los sonidos a un volumen u otro es compleja, debido a que normalmente no se espera o desea percibir todos los sonidos al mismo valor. Generalmente se establece un instrumento más alto que otro, algunos de fondo y otros en un valor intermedio.

La situación de cada instrumento depende en gran medida del instrumento en cuestión y el estilo musical de la obra. Hay estilos como el jazz donde los niveles de volumen están muy definidos, sin embargo en otros como dance o rap se posee mayor libertad.

Teniendo en cuenta el volumen como una cantidad de decibelios, entonces podríamos hablar de más de cien niveles diferentes de volumen, por lo que se dividen estos niveles a un número más manejable, y se establece a cada instrumento o sonido en uno de estos niveles o rangos. La clasificación utilizada para definir los volúmenes en nuestro software posee seis niveles, siendo el primero el más fuerte y el sexto el más débil. Los volúmenes son relativos o aparentes. Por ejemplo una motosierra sonará con mayor fuerza que una flauta aunque ambos estén situados al mismo nivel. El volumen aparente es el nivel de sonido al que parecerá estar el sonido al oírse. La siguiente tabla muestra una clasificación de distintos instrumentos o sonidos por rango, aunque muchos instrumentos dependiendo de la pieza musical o estilo pueden situarse en distintos rangos.

Los distintos rangos de volumen aparente y lo que se espera encontrar o conseguir en cada uno de ellos es:

- Rango 1. Sonidos extremadamente altos, lo que motiva que sea muy extraño situar un sonido a este nivel. Generalmente solo sonidos de muy corta duración se sitúan en el, como determinados efectos especiales. Situar un sonido en este rango durante un periodo prolongado resultará molesto y destruirá el resto de la mezcla.
- Rango 2. Principalmente se espera encontrar a este volumen los instrumentos y voces principales. En rap, hip-hop y música dance puede ser habitual encontrar el bombo, característico de dichos estilos. El nivel está diseñado para ser el más audible de la mezcla y todo instrumento o voz situado en dicho rango suele ser el foco de atención de la pieza musical, es decir aquí se encuentran aquellos sonidos que por su calidad o importancia se les quiera dotar de un mayor grado de notoriedad.

- Rango 3. En él están los sonidos que componen la parte rítmica, como los distintos instrumentos de percusión, bajos, guitarras y teclados. En muchas ocasiones en rock&roll la voz principal se encuentra a este nivel.
- Rango 4. Los sonidos incluidos son los encargados de los ritmos de fondo y acordes sostenidos de acompañamiento, instrumentos como los pianos de fondo, teclados o guitarras. A menudo las voces en segundo plano, las cuerdas y el reverb también se sitúan aquí.
- Rango 5. Provoca que los sonidos no sean fáciles de distinguir. Ejemplos a este nivel son el bombo en jazz o gran cantidad de efectos, con el objetivo de que solo se oigan si se escucha detenidamente. A menudo se sitúan aquí algunos sonidos únicamente para rellenar la mezcla.
- Rango 6. Son tan suaves los sonidos en este rango que es difícil detectarlos. Algunos grupos insertan mensajes subliminales o susurros, como por ejemplo Pink Floyd. Los sonidos a este nivel pueden llegar a ser muy efectivos en la mezcla siempre y cuando añadan valor al patrón general de mezcla.

1	2	3	4	5	6
Despertador	Voces principales	Ritmo principal	Camas de ritmo	Efectos	Susurros
Gritos	Instrumentos principales	Timbales	Acordes de relleno	Bombo (Jazz)	Ruido
Explosiones	Bombo (rap)	Bombo (Heavy-metal)	Cuerdas	Zumbido	
	Conjunto de trompas	Voz principal (Rock)	Voces de fondo		
		Tambor (dance)	Percusión (Jazz)		

Tabla 7-II. Clasificación de instrumentos según volumen aparente ordenados de mayor sonoridad (1) al menor (6)

Estos instrumentos son algunos de los más comunes en cada rango, pero dependiendo de la canción y del estilo musical los sonidos pueden saltar de un rango a otro. Un ejemplo claro de ello es el bombo (*kick drum*) que con el transcurso de los años y la aparición de nuevos estilos musicales ha ido ganando importancia dentro de la mezcla.

El siguiente nivel de intensidad se aplica al patrón general y define la relación general entre los distintos niveles de volumen, es decir, establece si las diferencias entre los volúmenes de los distintos sonidos serán más o menos amplias. Existen dos posibles estilos de mezcla atendiendo a la diferencia de variación entre el sonido más fuerte y el más débil:

- Variación pequeña. Se da en estilos como rock alternativo o en canciones de amor. Con este estilo de mezcla se usan como máximo tres niveles de volumen de los seis definidos, lo que dificulta en gran medida la mezcla y el conseguir que los sonidos no se enmascaren entre ellos.

- Variación grande. En este estilo de mezcla se usa todo el rango de volúmenes simplificando la mezcla y otorgando la posibilidad de innovar y hacer cambios. Estilos musicales que se mezclan con dicha variación son entre otros el rock&roll, el rap y las obras de orquesta.

El tercer nivel de aplicación son los cambios que se llevan a cabo durante la mezcla y en el caso de cambios de volumen se llevan a cabo casi exclusivamente con los fader. Ciertos estilos como jazz acústico o blues no permiten estos cambios en los volúmenes durante el proceso por tradición, los sonidos se establecen al comienzo de la pieza musical y no cambian hasta que termina. Sin embargo en otros estilos como la música electrónica es común las subidas y bajadas de volumen de uno u otro sonido creando una dinámica imposible de conseguir haciendo uso únicamente de los niveles anteriores.

Para simular el comportamiento del fader primero se aplicó al archivo de audio de cada instrumento una ganancia negativa para atenuar cada uno en la cantidad deseada según el estilo musical y posteriormente se realizaba la mezcla de todos los instrumentos. Pero el resultado no era satisfactorio por lo que se investigaron y probaron otros métodos. Finalmente se realizó la mezcla directa de todos los instrumentos sin modificar el volumen previamente, indicando en SoX un valor de volumen en Voltios a aplicar a cada instrumento a la hora de mezclar, permitiendo aplicar un mapa de volúmenes para la mezcla. Esto se lleva a cabo estableciendo un valor en dB a cada identificador de instrumento según el estilo musical. Dicho valor es aleatorio dentro de uno de los 6 rangos de volumen aparente creados o estático. Para utilizar la información de forma directa se transforma la cantidad de decibelios correspondientes al rango de volumen de cada instrumento al porcentaje de voltios (valor 1 no modifica el instrumento y 0 se silencia).

Mediante la siguiente llamada a SoX,

```
sox --combine mix -v volumenInstrumento1 instrumento1.wav  
-v volumenInstrumento2 instrumento2.wav mezcla.wav
```

donde se indican los volúmenes en Voltios de cada instrumento así como los archivos de audio de cada instrumento para la mezcla y el fichero de salida, se simula la relación de volúmenes entre los distintos instrumentos y si la variación entre estos es mayor o menor, simulando así los dos primeros niveles de configuración. El nivel uno se instaura previamente mediante la paleta del instrumento (archivo *sf2*) la cual se proporciona para crear el audio del instrumento a un nivel preestablecido, aunque dicho valor puede ser ignorado y reconfigurado durante la mezcla. El tercer nivel de configuración correspondiente a las variaciones durante la mezcla no está implementado, debido a que depende en gran medida de la composición y dificulta la obtención de buenos resultados si se procede a su automatización.

7.3.1.2 Compresor-limitador

La utilización de compresores y limitadores suele ser motivada por causa de razones técnicas o como el fader para conseguir crear un componente musical o emocional.

La compresión consiste en establecer un rango dinámico menor para la señal que se comprime, es decir, disminuir la diferencia entre el pico máximo y mínimo de la señal.

En el nivel uno, configuración individual, se comprimen aquellos instrumentos o voces en las que hay una gran diferencia entre su pico de señal más alto y bajo. Cuando se requiere una mezcla en la que la variación de volumen entre los instrumentos sea pequeña y sobre todo cuando hay muchos instrumentos en la mezcla, se suele realizar la compresión. También se lleva a cabo la compresión del instrumento más fuerte y más suave de la mezcla para así conseguir evitar enmascaramiento y dotarlos de más presencia.

Una vez realizada la mezcla pasa a aplicarse el segundo nivel de configuración en el que suele realizarse una compresión o limitación de los sonidos más fuertes de la mezcla, sobre todo cuando se quiere realizar un CD o se quieren hacer varias canciones que se reproducirán de forma consecutiva y así todas ellas posean el mismo rango dinámico consiguiendo gran estabilidad.

La limitación consiste en no permitir que se superen unos valores de decibelios achatando aquellos picos de señal que los sobrepasen, evitando así que al realizarse la mezcla aparezcan picos que distorsionen el sonido.

En nuestro software para simular dicho proceso se establece un limitador con la siguiente entrada:

```
limiterEffect(theme, gain)
```

en dicha función se aplica al archivo indicado (*theme*) el valor de ganancia a la entrada (*gain*) sin que se produzca el efecto *clipping* o recorte de señal. Con esto se simula la limitación evitando picos no deseados que molesten al escuchar la pieza musical.

La opción de compresión de señal se probó mediante la función `compand` de SoX, pero no fue necesaria su implementación debido a que se realiza directamente sobre los instrumentos virtuales utilizados (*SoundFont*) en caso de que lo necesiten. Es probable su implementación en versiones futuras.

7.3.2 Ecuación

Ecuación es atenuar o potenciar la amplitud de señal de una frecuencia en particular en un sonido. Debido a la gran cantidad de frecuencias posibles en el rango audible, de 20 a 20000Hz, la ecuación es la parte más compleja de la mezcla, ya que amplificar o atenuar una frecuencia determinada de un sonido puede cambiar la estructura básica del sonido en sí mismo.

Hay varios tipos de ecualizadores:

- Gráficos. Es el ecualizador más común y se definen por el número de bandas que poseen. Cada banda está definida por una frecuencia (frecuencia central) y ancho de banda predefinido. La calidad del ecualizador radica en el número de bandas que posee y en cómo afecta a las frecuencias cercanas al amplificar o atenuar una banda concreta. Los ecualizadores de este tipo van desde las 5 bandas a las 31 bandas. Con el de 31 bandas se consiguen mejores resultados que con el de 5 debido a que los anchos de banda y forma de onda de la ecuación son más precisos. Estos valores vienen predefinidos por el fabricante del ecualizador y cada uno tiene sus propios valores.

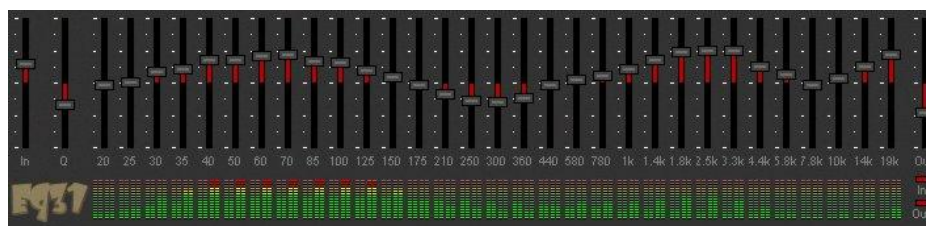


Figura 7.2. Ecualizador gráfico de 31 bandas

- Paramétricos. En estos tipos de ecualizadores se permite definir todas las variables, frecuencia central y ancho de banda, por lo que se tiene control sobre las frecuencias a las que se va a afectar cada vez que se aplica ecuación.
- Filtros. No son ecualizadores en sí pero suelen usarse junto a ellos. Los más comunes son el filtro paso bajo, el cual solo deja pasar las frecuencias menores al valor del filtro, y el paso alto, que permite pasar solo las frecuencias mayores al valor del filtro.

Para poder ecuación se necesita conocer la frecuencia correspondiente a cada nota (tabla 7-III) y los distintos rangos de frecuencias, así como qué representan en cada sonido. Existen distintas clasificaciones por rangos de frecuencias, uno de los más comunes es el siguiente:

MIDI Note	Freq.	MIDI Note	Freq.	MIDI Note	Freq.	MIDI Note	Freq.	MIDI Note	Freq.					
25	C#	34.648	51	D#	155.563	77	F	698.456	103	G	3135.963	127	G	12543.854
24	C	32.703	50	D	146.832	76	E	659.255	102	F#	2959.955	126	F#	11839.822
23	B	30.868	49	C#	138.591	75	D#	622.254	101	F	2793.826	125	F	11175.303
22	A#	29.135	48	C	130.813	74	D	587.330	100	E	2637.020	124	E	10548.082
21	A	27.500	47	B	123.471	73	C#	554.365	99	D#	2489.016	123	D#	9956.063
20	G#	25.957	46	A#	116.541	72	C	523.251	98	D	2349.318	122	D	9397.273
19	G	24.500	45	A	110.000	71	B	493.883	97	C#	2217.461	121	C#	8869.844
18	F#	23.125	44	G#	103.826	70	A#	466.164	96	C	2093.005	120	C	8372.018
17	F	21.827	43	G	97.999	69	A	440.000	95	B	1975.533	119	B	7902.133
16	E	20.602	42	F#	92.499	68	G#	415.305	94	A#	1864.655	118	A#	7458.620
15	D#	19.445	41	F	87.307	67	G	391.995	93	A	1760.000	117	A	7040.000
14	D	18.354	40	E	82.407	66	F#	369.994	92	G#	1661.219	116	G#	6644.875
13	C#	17.324	39	D#	77.782	65	F	349.228	91	G	1567.982	115	G	6271.927
12	C	16.352	38	D	73.416	64	E	329.628	90	F#	1479.978	114	F#	5919.911
11	B	15.434	37	C#	69.296	63	D#	311.127	89	F	1396.913	113	F	5587.652
10	A#	14.568	36	C	65.406	62	D	293.665	88	E	1318.510	112	E	5274.041
9	A	13.750	35	B	61.735	61	C#	277.183	87	D#	1244.508	111	D#	4978.032
8	G#	12.978	34	A#	58.270	60	C	261.626	86	D	1174.659	110	D	4698.636
7	G	12.250	33	A	55.000	59	B	246.942	85	C#	1108.731	109	C#	4434.922
6	F#	11.562	32	G#	51.913	58	A#	233.082	84	C	1046.502	108	C	4186.009
5	F	10.913	31	G	48.999	57	A	220.000	83	B	987.767	107	B	3951.066
4	E	10.301	30	F#	46.249	56	G#	207.652	82	A#	932.326	106	A#	3729.310
3	D#	9.723	29	F	43.654	55	G	195.998	81	A	880.000	105	A	3520.000
2	D	9.177	28	E	41.203	54	F#	184.997	80	G#	830.609	104	G#	3322.438
1	C#	8.662	27	D#	38.891	53	F	174.614	79	G	783.991			
0	C	8.176	26	D	36.708	52	E	164.814	78	F#	739.989			

Tabla 7-III. Relación MIDI-Nota-Frecuencia

- Menor de 40Hz. Se suele encontrar en bombos y bajos. Es complicado saber discernir qué nota suena en este rango. Estas frecuencias es típico su acentuación en películas en momentos con terremotos o explosiones.
- 40-100Hz. Se corresponde con el botón de “bass” de los reproductores de audio. Es muy común que se amplifique este rango hasta los 80Hz, más allá puede enturbiar el sonido.
- 100-800Hz. Es una zona crítica, debido a que si se amplifica demasiado puede hacer que el sonido cause fatiga e irritabilidad. Pero por el otro lado si se corta (atenúa) demasiado puede quedarse un mezcla muy endeble y sin sentido.
- 800-5000Hz. Es el rango que define la presencia de un sonido en la mezcla. Si se debilitan las frecuencias de este rango normalmente el sonido parecerá más lejano o distante. En este rango se sitúa nuestra habla, por lo que amplificar 1dB en este rango es como atenuar 3dB en cualquier otro. Hay que ser extremadamente cauteloso al hacer cambios en esta frecuencia sobre todo con las voces, ya que podrían dejar de sonar naturales. En este rango están también las frecuencias más irritantes posibles como el sonido de una motosierra o el de arañar una pizarra, que están alrededor de los 4000Hz.
- 5000-8000Hz. Se corresponde con el botón “treble” que encontramos en los reproductores de audio. Es muy común su amplificación sobre la mezcla porque provoca una sonoridad más brillante y presente.
- Mayor de 8000Hz. Suelen encontrarse los platos y los armónicos mayores. En algunos instrumentos se amplifica un poco esta zona para conseguir una mayor calidad de sonido, pero no en demasía porque podría volverse irritante.

Nombre característico	Rangos de frecuencias en Hz
Ultra altas (<i>Hi highs</i>)	>8.000
Altas (<i>Highs</i>)	5.000-8.000
Rango Medio (<i>Midranges</i>)	800-5.000
Zona crítica (<i>Oohzone</i>)	200-800
Bajo (<i>Bass</i>)	40-200
Sub-bajo (<i>Low bass</i>)	<40

Tabla 7-IV. Rangos de frecuencias

En el proceso de ecualización no suele haber grandes experimentos debido al gran cambio que puede provocar sobre un sonido, por lo que por norma general se modifican las distintas frecuencias de forma controlada y limitada, sobre todo para eliminar frecuencias que producen sonidos indeseados o potenciar aquella frecuencia que caracteriza al sonido.

Para la configuración individual de los instrumentos y sonidos en la ecualización existen dos formas de ecualizar o estilos:

- Ecualización natural. En ella se busca el objetivo primordial de la ecualización, hacer que el sonido suene lo más natural posible. Por ello si el sonido ha sido bien grabado no suele ser necesario apenas leves retoques, simplemente para dar más presencia a un sonido o dotarlo de viveza.
- Ecualización interesante. Interesante puede ser simplemente que no se hace ecualización natural. Otra forma de definirlo sería ecualizar para conseguir que un sonido muestre la máxima complejidad posible. Esto significa usar la ecualización para nivelar los picos del sonido. Si un sonido tiene varios picos en determinadas frecuencias, éstas serían las que se escucharían en principio, por lo que si se atenúan las frecuencias en las que se producen dichos picos entonces al oír el sonido el espectro percibido será mayor, más complejo, más interesante.

A pesar de poder hacer un instrumento más natural o interesante lo que realmente importa es conseguir que el conjunto de sonidos de la mezcla compaginen. Por ello lo normal es ecualizar los sonidos de forma nivelada para facilitar la relación entre ellos. De todos los niveles y características posibles de ajustar, el patrón general de ecualización es la más importante pues su objetivo es que la relación entre todos los sonidos sea correcta y no haya ninguna frecuencia o pico de señal de un instrumento que estropee la relación general.

Según el tipo de canción o estilo musical se crea una ecualización general para resaltar las partes más importantes de cada estilo. Por ejemplo en heavy metal suele amplificarse las frecuencias altas y en rap las bajas.

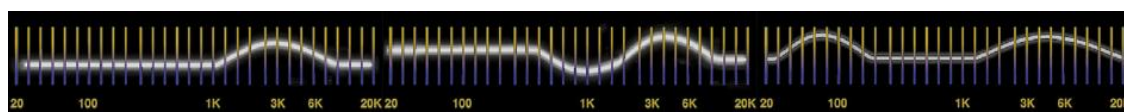


Figura 7.3. EQ general de los estilos Heavy-Metal, Jazz y Electrónica

El nivel dinámico en ecualización no suele aplicarse, ya que es muy inusual cambiar la configuración de ecualización durante la mezcla. Sin embargo en estilos relativamente novedosos como dance o techno esto está cambiando.

Para simular toda la ecualización se ha desarrollado una función que se encarga de ello. En ella se hace uso del ecualizador paramétrico de SoX y se implementa de tal forma que simula uno de 31 bandas, por lo que se definen treinta y una frecuencias centrales y un ancho de banda para cada una. Según el instrumento y el estilo musical se selecciona una ecualización u otra. La ecualización se aplica individualmente a cada instrumento y posteriormente a la mezcla según el estilo musical.

La función creada para realizar la ecualización es `equalizer(theme, instrument, style)` donde los parámetros de entrada corresponden al archivo sobre el que realizar la ecualización, el identificador del instrumento al que pertenece dicho archivo y el estilo musical respectivamente. En el caso de querer realizar una ecualización general sobre el audio final se establece como identificador de instrumento el valor -1. Dependiendo del instrumento y estilo recibido se seleccionan las bandas sobre las que realizar la ecualización y la cantidad de decibelios a atenuar o amplificar para cada una de ellas. Las frecuencias centrales que establecen las bandas en el ecualizador de 31 bandas que se simula son una media entre los valores encontrados en los distintos ecualizadores gráficos de varios editores de sonido y VST, mientras que el ancho de banda correspondiente a cada frecuencia central depende de la frecuencia superior e inferior de manera que al atenuar o amplificar una banda no se interfiera en las adyacentes.

7.3.3 Panorama

Se entiende por panorama mover un sonido a la izquierda o derecha dentro del estéreo, es decir dotar de más potencia a un instrumento en el altavoz izquierdo o en el derecho, para ser exactos disminuir la potencia en uno de ellos.

Al configurar los distintos instrumentos y sonidos individualmente suelen crearse centrados, salvo al crear un set de batería que a veces se crean simulando su colocación en una batería real.

En el panorama la dinámica se crea en el patrón general mediante la relación entre los distintos sonidos de la mezcla casi de forma exclusiva. Hay diversas formas de crear un patrón general con el panorama que son:

- Panorama natural. En este caso se elige dónde se sitúa cada instrumento, de izquierda a derecha, basándose en dónde se situarían en un concierto. Nuevamente el estilo musical determina el panorama si se desea que sea natural.

Por ejemplo en hip-hop existe una libertad total, sin embargo en una orquesta sinfónica está definido dónde se sitúa cada instrumento.

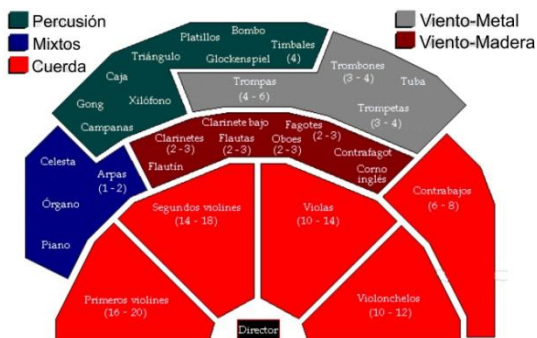


Figura 7.4. Colocación de los instrumentos de una orquesta sinfónica

- Panorama balanceado o enfrentado. Actualmente es más aceptado que el método natural debido a que el objetivo es reproducir el audio en unos altavoces, por lo que el mejor objetivo es obtener el máximo rendimiento de ello. Un panorama simétrico o balanceado es apropiado para baladas o canciones de amor y anti simétrico o enfrentado para canciones en las que se quiera crear tensión. Normalmente al crear una mezcla balanceada se suelen equilibrar los instrumentos de cada rango de frecuencias. En una mezcla enfrentada se hace justo lo contrario, se establecen todos los sonidos de un rango de frecuencias a un lado y en el contrario las del siguiente rango.
- Panorama de control de multitud. Se busca situar cada sonido en un lugar distinto, con lo que se gana claridad ya que los instrumentos no se solapan entre sí, o bien colocando los sonidos cercanos para conseguir crear lo que se conoce como muro de sonido, obteniéndose una mezcla más cohesiva. Con este método se elige entre claridad frente a cohesión. En estilos como jazz acústico, blues o incluso rap se busca claridad, mientras que estilos como heavy metal, rock alternativo o dance buscan más la cohesión.

Si se mezcla en un formato de audio *surround* entonces el espacio para el sonido se multiplicaría enormemente y el panorama ganaría mucha más importancia.

En el último nivel de intensidad se encuentra el panorama cambiante con el que se crea un efecto en el que un sonido se mueve de izquierda a derecha, muy llamativo sobre todo al usar unos auriculares. Este efecto se utiliza en ocasiones especiales y concretas. Los movimientos de panorama durante la mezcla son muy intensos, por lo que es llevado a cabo solo en canciones y estilos muy concretos. En música electrónica, house o trance entre otros es muy común.

Para implementar el panorama se ha creado la función:

```
panEffect(theme, left, right)
```

Dentro de la cual destaca la llamada a SoX:

```
system(['sox --single-threaded ' theme '.wav ' theme 'b.wav  
remix 1v' left ' 2v' righth]);
```

La cantidad de voltaje a aplicar en cada altavoz, izquierdo y derecho, se calcula de forma lineal, siendo el valor 1 sin cambios y 0 silencio en el lado correspondiente. El panorama se aplica modificando el archivo de audio generado para cada instrumento individualmente según el valor que le corresponda en la relación con los demás instrumentos y el estilo musical, situándose al mezclarse todos los instrumentos en el lugar adecuado.

Según el estilo se ha elegido hacer el panorama por un método u otro. Por ejemplo para orquesta se ha realizado un panorama natural, similar al de la figura 40, mientras que en el resto se suele buscar un panorama balanceado.

7.3.4 Efectos de sonido basados en tiempo

En este punto se mostrarán los efectos de sonido que se consiguen al añadir a un sonido la misma señal tras una demora en el tiempo. Según la duración de esta demora o retardo y el número de retardos que se añadan, se consigue un efecto u otro. Existen tres tipos básicos diferentes de efectos de sonido basados en el tiempo:

- Retardo simple.
- Retardo con realimentación.
- Reverberación.

Todos estos efectos y retardos pueden ser a su vez modificados en volumen, ecualización y panorama como si de otro instrumento o sonido de la mezcla se tratara.

Con la gran cantidad de efectos existentes se pueden crear multitud de sensaciones. Según la sensación que se quiera transmitir o que se quiera crear con cada sonido, se le añade a cada instrumento un efecto u otro. Por ejemplo con el reverb se crea más espacio mientras que con ecos y retardos largos una sensación de tranquilidad. Hay que tener en cuenta que cada vez que se añade un efecto basado en tiempo es como añadir uno o más sonidos a la mezcla lo que hace que esté más saturada. Según la cantidad de efectos usados sobre la mezcla se definen dos tipos de mezclas:

- Mezcla dispersa. Se añaden muy pocos efectos y se mantiene la mezcla clara, con mucho espacio entre cada sonido. Para muchos estilos, como música folk o jazz, se busca la claridad natural de los instrumentos por lo que este tipo de mezcla es idónea.
- Mezcla completa. Se usan muchos efectos para conseguir que el sonido de la mezcla sea mayor y más completo, como en un muro de sonido. Muchos estilos de música, como heavy metal o rock alternativo, usan con frecuencia multitud de efectos para rellenar todos los espacios entre los sonidos. En algunos casos

incluso la canción puede llegar a ser casi con exclusividad sobre los efectos, como puede ocurrir en hip-hop, techno o electrónica.

Al igual que con los otros tres parámetros de la mezcla, volumen, ecualización y panorama, los efectos también pueden ser cambiados durante la mezcla, lo que dota al efecto de mayor intensidad. Algunos ejemplos muy comunes son añadir un retardo al final de una palabra o añadir reverb al final de un solo. Uno de los mejores efectos posibles es eliminar un efecto de un sonido, lo que provoca un gran impacto debido a que se está acostumbrado a escuchar ese sonido con el efecto y no al revés.

Para simular y emplear todos estos efectos se han diseñado distintas funciones Matlab en las que todas tienen en común una llamada a SoX a un efecto concreto y una configuración. Dichas funciones pueden ser aplicadas a un instrumento individualmente o a la unión de todos los instrumentos, con lo que se simula los dos primeros niveles de intensidad. Al igual que en el resto de características el tercer nivel dinámico no ha sido implementado debido a que su uso suele ser subjetivo y muy específico del tema musical.

A continuación se definirán los tres tipos de retardos existentes mencionados anteriormente y datos sobre su uso e implementación.

7.3.4.1 Retardo simple

En el retardo simple se añade a la señal original su propia señal tras un periodo de tiempo. Los distintos rangos de tiempo de un retardo y la sensación que provoca o motivo por el que se usa son:

- Más de 100ms. Los retardos dentro de este rango se les conoce como eco. Cuando se usa un retardo de este tipo es imprescindible configurar la duración del retardo para que vaya acorde en tiempo con el tempo de la canción, o se destruirá la sincronización de la obra. El retardo puede estar en una fracción exacta o en un múltiplo del tempo. La siguiente tabla muestra algunos tiempos de retardo según el tempo.

Golpes por minuto (BPM)	Tiempo entre golpes
60 bpm	1.000 ms
90 bpm	666,6 ms
120 bpm	500 ms
150 bpm	400 ms
180 bpm	333,3 ms
240 bpm	250 ms

Tabla 7-V. Tempo vs Retardo

Cualquier múltiplo de estos valores también sería válido y no romperían la sincronización. Por ejemplo para un tempo de 60bpm además de 1.000ms también serian validos los tiempos de 2.000ms o de 500ms. Estos retardos son válidos debido a que cada uno corresponde a la duración de una figura musical y por eso también se los conoce con dichos nombres. En este caso por ejemplo el de 1.000ms sería un retardo a negras, el de 2.000ms a blancas y el de 500ms a corcheas.

Un retardo de más de 100ms suele usarse en canciones con tempos lentos, donde hay espacio para el sonido adicional. Sin embargo conforme más instrumentos y notas haya en la mezcla menos frecuente es el uso de este retardo.

- 60 a 100ms. Es conocido como bofetada o *slap*, y es utilizado para conseguir que un sonido fino o molesto sea más completo, especialmente en voces. De hecho puede ser usado para ocultar un sonido malo o conseguir que una voz suene menos personal.
- 30 a 60ms. Añadir un retardo de este tamaño hace que un sonido parezca que está siendo tocado dos veces. Se usa para simular que hay más voces o instrumentos. Este retardo es conocido como duplicador. También es usado para ocultar sonidos en la mezcla. Hay que usarlo con moderación y en sonidos, canciones y estilos determinados.
- 1 a 30ms. Ocurre algo particular empleando este tipo de retardos, conocido como engorde o *fattening*, debido a que con esta duración nuestro cerebro y oídos no son lo suficientemente rápidos para percibir los dos sonidos por separado, sino que se interpreta como un sonido más “gordo”. El umbral entre escuchar uno o dos sonidos depende de la duración del sonido que va a ser demorado, y de lo panoramizado que se encuentre el sonido original respecto del retardo. El umbral también depende en gran medida del timbre del instrumento y la manera de tocarlo. La siguiente tabla muestra algunos de estos umbrales para distintos instrumentos.

Límite aproximado entre escuchar un sonido o dos	
Hi-hat (Charles)	10 ms
Percusión	10 ms
Bombo	15 ms
Piano	20 ms
Cuernos	20 ms
Voces	30 ms
Guitarras	30 ms
Bajos	40 ms
Tubas	40 ms
Cuerdas lentas	80 ms

Tabla 7-VI. Rapidez del cerebro

Tras el reverb es el efecto más usado en los estudios de grabación. Es el principal efecto empleado para convertir un sonido en estéreo, para ello se sitúa el sonido original en un altavoz y el retardo en el otro. Con el *fattening* o engorde se puede mejorar de forma tangible un sonido ya de por si bueno. Este

efecto es muy efectivo para completar sonidos finos o molestos. También se usa para hacer un sonido más presente, especialmente cuando se desea bajar el volumen de un sonido en la mezcla pero siendo distinguible todavía.

El engorde suele usarse cuando hay pocos sonidos y notas en una mezcla debido a sus características, o en estilos como heavy metal en los que se requiere crear un muro de sonido.

- 0 a 1ms. Este tipo de retardo crea cancelación de fase. El principal efecto de la cancelación de fase es que deja de oírse el sonido, por lo que no es un efecto deseado y bastante crítico a la hora de grabar sonidos mediante micrófonos en el estudio.

Para implementar estos distintos retardos simples se hace uso del efecto `echo` (eco) de SoX, a través del cual se indica el periodo de tiempo tras el cual se añade la señal retardada y un valor de decaimiento de dicha señal. La llamada a la función es `echoEffect(theme, delay, decay)` y en él se establece el archivo sobre el que aplicar el eco, retardo base del eco y el porcentaje en el que decae el eco respectivamente. Para crear el efecto de eco propiamente dicho se tiene en cuenta el tempo de la obra para crear el valor de *delay* adecuado. También se puede modificar el número de ecos consecutivos en que cada eco decae en la proporción indicada (*decay*) respecto al anterior.

7.3.4.2 Retardos con realimentación

Si se establece un retardo de menos de 30ms, se le añade realimentación y se genera un barrido se pueden crear tres efectos diferentes dependiendo de la anchura a la que oscile el barrido:

- Entre 9 y 1ms. Se consigue un efecto llamado *flanging* o flanger. El flanger se usa para crear un tipo de emociones más psicodélicas, más mundanas. Es idóneo para simular que los sonidos están bajo el agua.
- Entre 2 ó 3 ms. El efecto conseguido es llamado coro. El efecto de coro tiene unos tiempos similares a los del duplicador o al de engorde. Se emplea para simular coros de personas o instrumentos, en los que siempre alguno se adelanta o atrasa un poco de los demás.
- Entre 0 y 1ms. Se conseguirá el efecto llamado fase. Es muy sutil y si se usa con un barrido muy pequeño (0-0.1ms). Se consigue un efecto en el que el sonido da vueltas alrededor de los altavoces o en el caso de unos cascos alrededor de la cabeza.

Tanto en el flanger, coro y fase además de la anchura del barrido poseen otro parámetro de configuración que es la velocidad a la que se recorre toda la oscilación.

Los tres efectos poseen su función correspondiente en SoX para llevarla a cabo, pero no han sido implementadas en esta versión del software debido a que solo se usan en momentos y estilos muy específicos.

7.3.3.3 Reverberación

La reverberación, o reverb como se conoce generalmente al efecto, es un fenómeno acústico natural que se produce en recintos más o menos cerrados por el cual a la señal original se le van sumando las diferentes ondas reflejadas en las paredes del recinto con un retardo generado por la distancia física entre la fuente de sonido original y las paredes del recinto.

Algunos autores para definir el fenómeno de la reverberación usan el término eco como repetición de un sonido en el tiempo, de forma que la reverberación es un conjunto de ecos producidos por las paredes del recinto que se van sumando a la señal original. En cambio, otros autores más técnicos, prefieren diferenciar claramente ambos conceptos. Y esa diferenciación se basa en una cuestión de percepción auditiva. Nuestro oído posee una característica denominada persistencia acústica por la cual es incapaz de distinguir un sonido de sus reflexiones siempre y cuando la diferencia de tiempo entre ambas sea menor a 1/15 de segundo. También existe el efecto HAAS, por el cual cuando el oído capta unas reflexiones con un retardo inferior a 35ms es incapaz de integrarlos como ecos consecutivos y los asocia a la fuente original reforzándola.

A efectos prácticos, esto se traduce en que nuestro oído no diferencia entre el sonido puro y su reverberación, sino que lo percibe como un todo, un mismo sonido que además presenta unas características diferentes del sonido original.

El reverb es el efecto más usado a la hora de realizar una mezcla. Por ello algunos de sus parámetros son comunes en prácticamente todos los dispositivos existentes para realizar un reverb, tanto hardware como software. Algunos de los más comunes son:

- Tipo de habitación. Dependiendo del sistema puede disponerse de distintos formatos de habitación para elegir, desde una catedral a una cueva, mientras que en otros se tiene la posibilidad de establecer el tamaño de la sala.
- Tiempo de reverb o reverberación. Es lo que tarda el sonido del reverb en desaparecer.
- Retardo previo o *predelay*. Cuando se produce un sonido este viaja hasta las paredes y regresa, este tiempo de silencio que existe hasta que el reverb comienza es el retardo previo. Cuanto mayor es la habitación mayor retardo previo existe. Si se busca un verdadero efecto de reverberación es importante que exista un pequeño tiempo de retardo previo.
- Difusión. Hace alusión a la cantidad de ecos que componen el reverb.

- Amortiguación o *damping*. En él se establece la absorción de unas frecuencias determinadas por las paredes. Existe tanto para altas como para bajas frecuencias, siendo más común para altas, y es básicamente como una equalización del sonido antes de ser rebotado.
- Señal seca y mojada. Más conocido como “dry” y “wet” es la señal original o natural sin reverb en el caso de seco, y en el caso de mojado únicamente la señal de reverberación. Se usa para controlar la pérdida de volumen de la señal al rebotar o para controlar cuanta cantidad de señal mojada es mezclada con la seca.

El efecto de reverb se implementa mediante la función `reverbEffect(theme, reverb, stereo, hfdamping, room, predelay, wet)` donde:

- `theme`, es el nombre del archivo sobre el cual aplicar el efecto.
- `reverb`, es la cantidad de reverberación que se añade al sonido original.
- `stereo`, permite estrechar o ensanchar la imagen estéreo de la reverb aplicada.
- `hfdamping`, controla el decaimiento de las altas frecuencias, ya que estas son más proclives a ser absorbidas por el entorno.
- `room`, define el tamaño de la habitación que produce la reverberación.
- `predelay`, tiempo que transcurre antes de que comience el reverb.
- `wet`, ganancia aplicada a la reverberación únicamente, no al sonido original.

Variando estos parámetros, sobre todo `reverb` y `room`, se consiguen una gran variedad de sensaciones diferentes utilizadas según el instrumento y estilo musical.

Capítulo 8: Pruebas y resultados

8.1 Procesado de audio

Durante el desarrollo del software de síntesis musical se realizaron multitud de pruebas de sonido o sobre el procesado de audio. Estas pruebas se crearon con el objetivo de obtener resultados sobre uno de los siguientes aspectos:

- Chequear el funcionamiento de cada parámetro del efecto o procesado correspondiente para efectuar la implementación más beneficiosa del mismo.
- Obtener la configuración con mejor resultado auditivo para cada estilo musical.

Para mostrar el funcionamiento de los testeos relativos a cada uno de los aspectos descritos se analizarán las pruebas efectuadas para observar el comportamiento del ancho de banda en la ecualización, y las cumplimentadas para la configuración de la reverberación en tres estilos musicales distintos. También se mostrarán los ensayos elaborados para la implementación del efecto eco y la eliminación de silencios al final de pista.

Los archivos de audio correspondientes a todas las pruebas se adjuntan con el código del programa.

8.1.1 Ancho de banda en la ecualización

En la ecualización influyen tres parámetros fundamentales: frecuencia central, ancho de banda y ganancia. Debido a la decisión de simular un ecualizador de 31 bandas, las frecuencias centrales a utilizar vienen definidas por los valores correspondientes a 1/6 de octava. La ganancia a aplicar, ya sea para atenuar o amplificar cierta banda determinada, es diferente según el instrumento y estilo musical, y por ello se efectúan pruebas para verificar los valores que proporcionan una mejor sonoridad. El último parámetro es el ancho de banda, valor fijo y desconocido en la mayoría de los ecualizadores comerciales, tanto hardware como software, al ser un parámetro cuya configuración determina la calidad del ecualizador. Para observar el funcionamiento del ancho de banda se creó una muestra de ruido blanco con la siguiente llamada:

```
sox -b 16 -c 2 -r 44100 -n whitenoise.wav synth 10 whitenoise
```

Posteriormente sobre dicho ruido blanco se aplica ecualización manteniendo invariables los parámetros de frecuencia central (400Hz) y ganancia (-50dB), este último excesivamente exagerado para obtener un resultado visual más claro, a pesar de ser un valor poco común ecualizar, porque equivaldría a silenciar dicha banda de frecuencia.

Las distintas ecualizaciones realizadas son:

```
sox whitenoise.wav eq_400_1.wav gain -5 equalizer 400 1h -50
sox whitenoise.wav eq_400_10.wav gain -5 equalizer 400 10h -50
sox whitenoise.wav eq_400_50.wav gain -5 equalizer 400 50h -50
sox whitenoise.wav eq_400_1000.wav gain -5 equalizer 400 1000h -50
```

Como se puede observar los distintos parámetros de ancho de banda aplicados son: 1, 10, 50 y 1.000 Hertzios. Empleando un analizador de espectro sobre cada uno de los archivos de audio generados se obtienen los siguientes resultados.

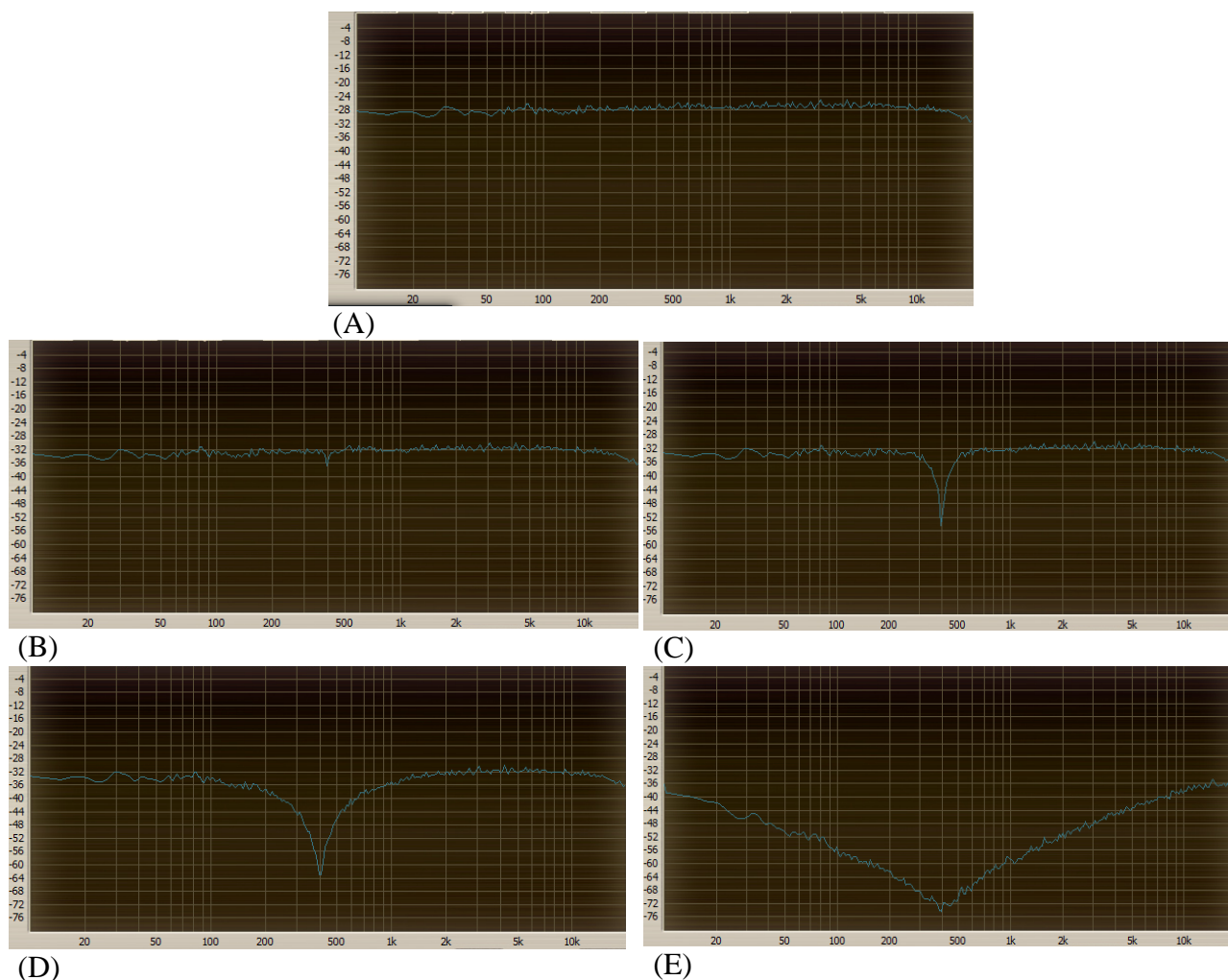


Figura 8.1. Análisis espectral. (A) Ruido Blanco, (B, C, D, E) EQ con anchos de banda de 1, 10, 50 y 1.000 Hertzios respectivamente

Al examinar las imágenes se comprueba que si el ancho de banda es muy reducido, no solo no provoca que la mayoría de frecuencias de la banda no se modifiquen, sino que además la atenuación o amplificación llevada a cabo es muy inferior a la deseada sin apenas modificarse el sonido original. En el caso de los 1000Hz ocurre justo lo contrario, influye en prácticamente todo el rango de frecuencias. A pesar de ello para ecualización de altas frecuencias cuya distancia entre bandas es superior a los 2 o 3KHz, se emplean anchos de banda de dicho tamaño para la configuración de dichas bandas.

En el caso concreto de los 400Hz teniendo en cuenta que las bandas anterior y posterior tienen como valores de frecuencia central 370 y 415Hz respectivamente, el valor utilizado es el de 10Hz de ancho de banda ya que no influye en las bandas adyacentes. Haciendo uso de pruebas similares, teniendo en cuenta las frecuencias centrales anterior y posterior, se configuran los distintos anchos de banda para cada banda, siempre buscando evitar el solapamiento entre las distintas bandas.

8.1.2 Configuración de la reverberación

El efecto de reverberación es el más utilizado de todos, por ello se emplea en mayor o menor medida en casi todos los estilos musicales implementados. Para configurar los parámetros que le dan forma se efectuaron pruebas modificando dichos valores, y con la ayuda del grupo de músicos pertenecientes al proyecto Melomics se escogieron aquellos que proporcionaban mejor sonoridad.

	1	2	3	4	5	6	7	8	9	10
Reverb	50	30	30	30	20	10	5	1	1	1
Stereo	50	50	50	50	50	50	50	50	30	70
Hfdamping	50	40	40	40	40	20	10	10	10	10
Room	100	80	80	80	70	60	64	64	64	64
Predelay	0	0	50	30	30	30	30	30	30	30
Wet	0	0	-3	-1	-1	-1	-1	-1	-1	-1

Tabla 8-I. Reverb en electrónica

Uno de los primeros estilos musicales con los que se trabajó fue la música electrónica. Se le comenzó aplicando el reverb por defecto de SoX, pero producía un efecto solapamiento sobre todo el audio percibiéndose exclusivamente éste. Por ello se redujeron todos los parámetros en general. Los datos de *predelay* y *wet* usados para electrónica se han reutilizado en el resto de estilos, debido a que simulan con precisión la realidad de una sala. Finalmente para música electrónica se empleó la configuración número 8, con un valor medio para el tamaño de habitación y la profundidad del estéreo. La cantidad de reverberación añadida así como de decaimiento de las altas frecuencias se disminuyó drásticamente, motivado por la búsqueda de una mezcla clara donde poder diferenciar los distintos sonidos. El aumento de la reverberación provocaba una sonoridad confusa por el solapamiento que se producía entre algunos instrumentos poseedores de reverberación individual que los dota de una mayor presencia.

	1	2	3	4	5
Reverb	10	30	50	40	40
Stereo	50	50	50	50	50
Hfdamping	10	10	10	10	10
Room	64	50	40	80	60
Predelay	30	30	30	30	30
Wet	-1	-1	-1	-1	-1

Tabla 8-II. Reverb en música de audiovisuales

El siguiente estilo tratado fue música para audiovisuales, caracterizada por ser una música lenta con notas largas en las que se busca un sonido que acompañe de fondo a imágenes. Caracterizado por presentar un menor número de instrumentos que en electrónica se fue aumentando la cantidad de reverberación añadida con el objetivo de rellenar las zonas con menos instrumentos y que la música fuera más presente. La configuración seleccionada fue la número 5, pero con un tamaño de habitación variable entre 50 y 80 para abarcar los distintos espacios que puedan mostrarse en las imágenes a las que pueda acompañar la música.

	1	2	3	4	5
Reverb	10	40	50	60	70
Stereo	50	50	60	70	80
Hfdamping	10	10	10	10	10
Room	64	64	64	64	64
Predelay	30	30	30	30	30
Wet	-1	-1	-1	-1	-1

Tabla 8-III. Reverb para *quartet*

El estilo llamado *quartet* es un estilo particular en el que se usa un cuarteto de cuerda, permitiéndose sustituir alguno de los instrumentos por instrumentos de viento o de teclas como la celesta o el xilófono, acompañado por un bajo y en algunos casos percusión o coros. En él la reverberación está muy ligada al efecto de eco, junto al cual busca dar una sensación de tranquilidad alargando los sonidos. Motivado por este objetivo y los instrumentos concretos utilizados se aumentó la cantidad de reverberación añadida así como de la anchura del estéreo, hasta llegar a la configuración número 5 utilizada.

8.1.3 Efecto eco

Con el efecto de eco se busca alargar el sonido y llevar a cabo su extinción paulatinamente. Para ello se utiliza en conjunción con el reverb generalmente. A diferencia de con otros efectos, como el reverb, en el eco no solo se configuran los parámetros de entrada de la función SoX utilizada, sino también valores como el número de ecos a producir o el retardo extra a añadir a cada eco. La siguiente tabla define distintas configuraciones probadas.

	1	2	3	4	5	6	7	8	9
Delay	Blancas	Blancas	Blancas	Negras	Negras	Negras	Corcheas	Corcheas	Corcheas
Extra delay	0	0	1	1	2	1	1	1	1
Decay	10	20	20	20	20	30	10	20	10
Nº de ecos	1	2	2	2	3	4	3	3	3

Tabla 8-IV. Configuraciones para el efecto de eco

La configuración final de *extra delay* y número de ecos es de 1sg y 3 ecos respectivamente. Con el parámetro de tiempo extra de retardo se provoca la sensación de un espacio mayor, de ahí que a la duración de la figura correspondiente se le añada 1sg extra a cada eco consecutivo. El número de ecos se establece de forma que el último todavía sea audible al decaer cada eco un porcentaje respecto al anterior, o del sonido original en el caso del primero. El porcentaje de decaimiento de la señal se estableció aleatoriamente entre un 10%, 20% o 30%, a causa de los resultados satisfactorios con esas cantidades. De igual manera sucede con los retardos a negras y corcheas por lo que se elige aleatoriamente uno de los dos valores, quedando descartado el retardo a blancas por provocar un retardo excesivo.

Al igual que en el caso de la reverberación y de otros efectos, estos valores y resultados fueron seleccionados de acuerdo a la opinión y ayuda de expertos en el campo.

8.2 Optimización con paralelismo

Para comparar las distintas estrategias de optimización usando paralelismo se construyeron dos pruebas diferenciadas por el objetivo marcado en cada una:

- Obtener una canción de la forma más rápida posible.
- Obtener el máximo número de canciones en el menor tiempo posible.

Las pruebas llevadas a cabo para ambos objetivos fueron realizadas sobre un ordenador de 4 núcleos y una pieza musical de 6 instrumentos con una duración de 1 minuto.

El primer objetivo es buscado tanto en la fase de desarrollo y pruebas, cómo al implementar nuevos estilos musicales, para obtener resultados cuanto antes. El segundo objetivo es el objetivo final y se lleva a cabo haciendo uso del clúster de computación.

8.2.1 Eficiencia individual

Con estas pruebas se busca conseguir el archivo de audio final en el menor tiempo posible. Para ello se emplean las estrategias de optimización descritas en el capítulo 5 y

se comparan los resultados obtenidos. Los distintos lanzamientos se realizan efectuando modificaciones en el bucle principal donde es generado cada instrumento, realizándolo tanto de forma secuencial como paralela (*for* y *parfor*, respectivamente), y en SoX cambiado el modo en que es procesado el audio, en un único procesador o en todos los núcleos (etiquetas *single-thread* y *multi-thread*). Las distintas combinaciones, que identificamos según el tipo de bucle usado y modo de uso de SoX (*for*, *parfor*, *single*, *multi*), dan lugar a las distintas estrategias previamente mencionadas. Dichas pruebas son:

- For + Single. Versión original sin optimización en la que el código se ejecuta de forma secuencial.
- For + Multi. Se usa la opción *multithread* de SoX para acelerar el proceso. Se corresponde con la estrategia 1.
- Parfor + Multi. Se intercambia el bucle *for* principal por un bucle *parfor*, con el que se ejecuta en paralelo la creación de cada instrumento individualmente. Además se mantiene activa la opción *multi-thread* de SoX. Esta combinación se corresponde con la primera etapa de la estrategia 3.
- Parfor + Single. Se emplea únicamente un bucle paralelo para la creación de los instrumentos y se corresponde con estrategia 2 descrita anteriormente.
- Híbrida. Se utiliza un bucle paralelo como en las dos combinaciones anteriores, pero dentro del mismo no se activa la opción *multi-thread* de SoX mientras en la parte independiente a este donde se realiza la mezcla y procesado de la misma sí se activa.

Los resultados de tiempo obtenidos de las distintas combinaciones son las siguientes:

For + Single	For + Multi	Parfor + Multi	Parfor + Single	Híbrida
43.912s	43.759s	33.510s	27.665s	28.579s

Tabla 8-V. Tiempos de obtención de un fichero MP3

Del análisis de los tiempos obtenidos se extraen varias conclusiones, destacando la mejora del 60% aproximadamente al paralelizar el bucle principal en el cual se lleva a cabo la construcción de cada instrumento de la pieza musical, siendo ésta la opción que se lleva a cabo generalmente. Dicha mejora se llevó a cabo debido a que el tiempo de procesado de dicho bucle de modo secuencial es del 62% del tiempo total de procesado. La opción *multi* de SoX solo aporta una mejora de décimas de segundo en cada uso y no en todas las funciones, como se puede observar de los tiempos obtenidos en la opción híbrida, ya que de no ser así dicho tiempo debería haber sido el mejor de todos. Esto sí se cumple si el número de instrumentos crece de forma que se superen los 20 ó 30 instrumentos, como ocurre en obras de orquesta con hasta 43 instrumentos distintos y aproximadamente 30 de media, con los que el tiempo de procesado de la mezcla de todos los instrumentos individuales crece de los 1.5s a los 10-20s, donde la opción híbrida es la mejor. Otro dato importante que muestran los test es que al utilizarse todos los núcleos del procesador, tanto con el bucle *parfor* como con la opción *multi*, éstas

entran en conflicto y se generan tiempos de espera, ralentizando el proceso. Esto último se observa con claridad en las pruebas para obtener más de una pieza musical del siguiente apartado.

8.2.2 Máxima cantidad

En este caso se intenta conseguir el máximo número de piezas musicales en el menor tiempo posible. Para realizar estas pruebas generamos el mismo tema cuatro veces (el mismo que núcleos posee el ordenador) por cuatro métodos diferentes, combinando la estrategia que provoca máxima velocidad de generación de un solo tema (Parfor + Single) junto con la paralelización o no de los cuatro temas musicales a generar, y que posteriormente comparamos. Dichas pruebas son:

- For + For. Versión original sin optimización en el que todo el código se ejecuta de forma secuencial.
- For + Parfor. Se crean las piezas musicales de forma secuencial empleando la estrategia que mejores tiempos obtuvo en las pruebas individuales.
- Parfor + Parfor. Se añade una paralelización en la creación de cada tema musical.
- Parfor + For. Únicamente se paraleliza el lanzamiento de la construcción del audio de cada pieza musical, generándose estas secuencialmente en cada núcleo. Dicha opción coincide con la estrategia 4 descrita en el capítulo 5.

Los resultados de tiempo obtenidos de las distintas combinaciones son los siguientes:

For + For	For + Parfor	Parfor + Parfor	Parfor + For
169.724s	112.314s	97.126s	76.845s

Tabla 8-VI. Tiempos de obtención de 4 piezas musicales

Comparando los resultados se observa una mejora del 45% al paralelizar la emulación completa, por lo que al realizar lanzamientos en el clúster de computación esta estrategia proporciona el mayor número de piezas musicales en un tiempo definido. Se detecta cómo se ralentiza la emulación al paralelizar doblemente, al solicitarse el uso de un mismo procesador por distintos procesos generando tiempos de espera y comunicación extra. La paralelización en el clúster se lleva a cabo a través de Torque para poder obtener el máximo rendimiento.

Capítulo 9: Conclusiones y extensiones futuras

9.1 Conclusiones

La realización del proyecto ha sido una gran experiencia debido a que se ha tenido la oportunidad de realizarlo dentro de un grupo de investigación y trabajando diariamente con el personal que lo componía. Durante el proceso se ha leído y consultado multitud de artículos, publicaciones y libros sobre materias tan distintas como la teoría musical o la programación paralela. Esto ha provocado la adquisición de conocimientos totalmente ajenos a la carrera pero necesarios para poder comprender el motivo y el camino a seguir para el desarrollo tecnológico e informático a realizar. En el transcurso de las investigaciones se ha comprobado que una de las habilidades más importantes adquiridas durante la carrera es la capacidad para asimilar conceptos nuevos y aplicarlos a lo aprendido previamente. También al tener que elaborar un software que debía convertirse en una parte de una infraestructura mayor se ha comprobado la importancia de implementar un código bien documentado y modulado, debido a que su ausencia provoca retrasos, fallos inesperados e ineficiencia. Probablemente el hecho de no solo poder asentar y ejecutar lo aprendido durante la carrera, sino el haber podido con el desarrollo del PFC acercarme e iniciar la carrera profesional en la que se ha observado la importancia del trabajo diario y aprovechar los conocimientos y experiencias de compañeros de trabajo. En este tiempo se ha constatado la importancia de una buena gestión en la realización de un proyecto o trabajo y control de versiones, incrementándose proporcionalmente al número de personas implicadas, así como de todas las fases de un desarrollo software, desde la implementación a las pruebas, aptitudes y conocimientos también adquiridos durante la carrera. Durante el transcurso del proyecto se ha podido trabajar con personas especializadas tanto en el campo de la música o el sonido como con un amplio número de informáticos de gran experiencia, de los que se ha logrado obtener experiencias y conocimientos a los que son imposibles de acceder a través de libros y publicaciones o trabajando en solitario. Por último se ha de mencionar la oportunidad de trabajar en un clúster de computación pudiendo aplicar los conocimientos de programación paralela y distribuida adquiridos previos así como desarrollarlos en un entorno único.

El proyecto partió con el objetivo de desarrollar un software capaz de crear archivos de audio con una interpretación fiel de las partituras, en concreto de la información compositiva de las piezas musicales creadas en el proyecto Melomics. Tras la fase de desarrollo y múltiples pruebas efectuadas se puede asegurar que la aplicación desarrollada cumple los objetivos de reproducción fiel de la composición y soluciona problemas como la limitación de instrumentos y reproducción de recursos interpretativos como el arpeggio o el pizzicato.

La música es un campo inmenso con multitud de estilos diferentes, donde la subjetividad de cada persona aporta puntos de vista distintos, por lo que una gran parte del trabajo se ha centrado en la investigación de las distintas formas de materializar y alcanzar en el audio determinadas características y sensaciones, para así poder trasladar este conocimiento al software desarrollado, cumpliendo así otro de los objetivos marcados: conseguir la mayor similitud entre el audio reproducido y la interpretación humana. Para ello se ha dotado al software de una gran parte de los recursos de grabación musical, disponiendo la aplicación de las herramientas necesarias para mezclar y sintetizar estilos tan diferentes entre sí como la música clásica contemporánea y la música electrónica, ambos implementados, y permitiendo agregar en un futuro cualquier otro estilo musical que se considere. El software además dispone de una función de humanización, desarrollada sobre todo en el campo de la intensidad y el volumen, que unido al uso de paletas instrumentales y de efectos como la reverberación, construye archivos de audio que difieren de lo típicamente conocido como música por ordenador, demasiado perfecta, extraña al oído humano. Estos archivos se encuentran disponibles en múltiples formatos.

En el transcurso de la realización del proyecto se ha tenido muy en cuenta la optimización de todo el desarrollo de síntesis, debido al objetivo de integración dentro de la estructura del proyecto Melomics, aun siendo solo una pequeña parte. Se considera este objetivo cumplido también, como puede comprobarse con la multitud de archivos que se encuentran en el sitio web del mismo, siendo la gran mayoría de estos creados mediante distintas versiones del software desarrollado en el éste PFC.

9.2 Extensiones

A continuación se comentarán algunas de las extensiones y mejoras detectadas a lo largo del proyecto y se consideran que podrían resultar significativas para futuras versiones del software.

Nuevos estilos musicales

Como se ha comentado en el apartado de conclusiones la música es un campo enorme y en el desarrollo del software solo se han tratado algunos de estos campos, por lo que la aplicación permanecerá en continua evolución, añadiendo los distintos estilos y subestilos musicales que no se incluyen, y ampliando los existentes con formas diferentes de sintetizar aportando así una mayor riqueza y variabilidad.

Entorno gráfico

Una extensión muy útil será un interfaz gráfico que permita usar el emulador y así poder crear fácilmente nuevas configuraciones o modificar las existentes. Pudiendo añadir y cambiar los instrumentos a usar o incluso aplicar cambios en la melodía original.

Tercer nivel de dinámica

Se ha explicado previamente los distintos niveles de dinámica que se pueden conseguir al realizar la síntesis, siendo el tercero y último aquel que cambia la configuración durante la pieza musical y añade o elimina efectos extra en momentos concretos. Este desarrollo no se implementó debido a que es muy dependiente de cada pieza y estilo musical y bastante subjetivo, por lo que se considera que su desarrollo junto al entorno gráfico sería muy beneficioso para la mejora del proceso de síntesis.

Bibliografía

1. Bazil, E. (2008). *EQ:Uncovered*. MPC-SAMPLES.
2. Gibson, D. (2005). *The Art of Mixing*. Thomson Course Technology PTR.
3. Harley, R. (2010). *The Complete Guide to High-End Audio*. Acapella Publishing.
4. Izhaki, R. (2008). *Mixing Audio*. Elsevier.
5. Moreno Arcas, F. (2011). *Desarrollo de un sistema de modelado de partituras musicales en notación profesional a partir de un generador de estructuras musicales*. PFC en E.T.S. Ingeniería informática, Universidad de Málaga.
6. Recordare LLC. (2011) . *MusicXML 3.0 Tutorial*. Recordare LLC.
7. Sokolow, F. (1995). *The dictionary of Strum & Picking Patterns*. Hal Leonard.
8. Puckette, M. *Lenguaje de programación Pure Data* [Sitio web]. Disponible en: <http://puredata.info/> [Consulta: junio de 2013]
9. Steinberg. *Catálogo de productos VST de Steinberg* [Sitio web]. Disponible en: <http://www.steinberg.net/en/products/vst.html> [Consulta: junio de 2013]
10. Acoustica. *Software de edición musical Mixcraft* [Sitio web]. Disponible en: <http://www.acoustica.com/mixcraft/> [Consulta: junio de 2013]
11. Rosegarden. *Software de edición musical Rosegarden* [Sitio web]. Disponible en: <http://www.rosegardenmusic.com/> [Consulta: junio de 2013]
12. Rundt, K. *Software de edición musical Synthfont* [Sitio web]. Disponible en: <http://www.synthfont.com/index.html#> [Consulta: junio de 2013]
13. Asociación de fabricantes MIDI. *Especificaciones del protocolo MIDI* [Sitio web]. Disponible en: <http://www.midi.org/techspecs/> [Consulta: junio de 2013]
14. Nasca, P. *Sintetizador software Zynaddsubfx* [Sitio web]. Disponible en: <http://zynaddsubfx.sourceforge.net/> [Consulta: junio de 2013]
15. Coca, A. (2009). *Composición automática de fragmentos musicales con sistemas dinámicos caóticos y bifurcaciones*. Tesis, Universidad Nacional de Colombia. Disponible en: <http://www.bdigital.unal.edu.co/3518/> [Consulta: junio de 2013]
16. Bertrand J, Catalán C. y Georgiev V. (2012). *Muphic: composición musical automática basada en imágenes*. Disponible en: <http://eprints.ucm.es/15932/> [Consulta: junio de 2013]
17. Wolfram, S. *Especificaciones y composiciones de WolframTones* [Sitio web]. Disponible en: <http://tones.wolfram.com/about/faqs/howitworks.html> [Consulta: junio de 2013]
18. Tudor, D. *Entorno Rainforest IV* [Sitio web]. Disponible en: <http://www.cieweb.net/rainforest/> [Consulta: junio de 2013]
19. Melomics. *Repositorio musical de Melomics* [Sitio web]. Disponible en: <http://melomics.com/> [Consulta: junio de 2013]

20. Mathworks. *Software de cálculo MATLAB* [Sitio web]. Disponible en: <http://www.mathworks.es/> [Consulta: junio de 2013]
21. Adaptive Computing. *Gestor de recursos TORQUE* [Sitio web]. Disponible en: <http://www.adaptivecomputing.com/products/open-source/torque/> [Consulta: junio de 2013]
22. PBS Works. *Proyecto PBS* [Sitio web]. Disponible en: <http://www.pbsworks.com/> [Consulta: junio de 2013]
23. Adaptive Computing. *Planificador de tareas MAUI* [Sitio web]. Disponible en: <http://docs.adaptivecomputing.com/maui/mauistart.php> [Consulta: junio de 2013]
24. Lilypond. *Software y lenguaje musical lilypond* [Sitio web]. Disponible en: <http://www.lilypond.org/manuals.es.html> [Consulta: junio de 2013]
25. Izumo, M. *Sintetizador software Timidity++* [Sitio web]. Disponible en: <http://timidity.sourceforge.net/> [Consulta: junio de 2013]
26. Sox. *Documentación de uso del software para procesamiento de audio SoX* [Sitio web]. Disponible en: <http://sox.sourceforge.net/sox.html> [Consulta: junio de 2013]
27. Ecasound. *Software para procesamiento de audio Ecasound* [Sitio web]. Disponible en: <http://eca.cx/ecasound/index.php> [Consulta: junio de 2013]
28. Fluidsynth. *Sintetizador software Fluidsynth* [Sitio web]. Disponible en: <http://sourceforge.net/apps/trac/fluidsynth/> [Consulta: junio de 2013]
29. Grabowski, W. *Software conversor de audio Extreme sample converter* [Sitio web]. Disponible en: <http://www.extranslator.com/> [Consulta: junio de 2013]
30. Chicken system. *Software conversor de audio Translator 6* [Sitio web]. Disponible en: <http://www.chickensys.com/downloads2/translator.html> [Consulta: junio de 2013]
31. Makemusic. *Lenguaje musical musicXml* [Sitio web]. Disponible en: <http://www.musicxml.com/> [Consulta: junio de 2013]
32. Open Sound Control. *Protocolo de comunicación OSC* [Sitio web]. Disponible en: <http://opensoundcontrol.org/introduction-osc> [Consulta: junio de 2013]
33. Creative. *Especificaciones del formato SoundFont* [Sitio web]. Disponible en: <http://connect.creativelabs.com/developer/SoundFont/sfspec21.pdf> [Consulta: junio de 2013]
34. Digital sound factory. *Catálogo de bancos de instrumentos sf2* [Sitio web]. Disponible en: <https://www.digitalsoundfactory.com/> [Consulta: junio de 2013]
35. Vico, F. *Proyecto Melomics del grupo de estudios en biomimética* [Sitio web]. Disponible en: <http://geb.uma.es/melomics/melomics.html> [Consulta: junio de 2013]
36. Teoria.com. *Portal sobre el estudio de la teoría musical* [Sitio web]. Disponible en: <http://www.teoria.com/indice.php> [Consulta: junio de 2013]
37. Ctorteka, B. (2011). *Invention: Humanizing Music Sequence* [Sitio web]. Disponible en: <http://blog.max-planck-innovation.de/inventions/humanizing-music-sequence> [Consulta: junio de 2013]

38. Woodring, T. (2011). *How to humanize your music* [Sitio web]. Disponible en: <http://songwritingbusiness.blogspot.com.es/2011/08/how-to-humanize-your-music.html> [Consulta: junio de 2013]
39. Henning, H. (2011). *The nature and perception of fluctuations in human musical rhythms* [Sitio web]. Disponible en: <http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0026457> [Consulta: junio de 2013]
40. Blogger. *Blog sobre audio y síntesis* [Sitio web]. Disponible en: <http://audiosintesis.blogspot.com.es/> [Consulta: junio de 2013]
41. Astor mastering. *Cursos y publicaciones sobre mezcla y mastering* [Sitio web]. Disponible en: http://www.astormastering.com.ar/blog/?page_id=106 [Consulta: junio de 2013]
42. Hispasonic. *Tutoriales sobre síntesis musical* [Sitio web]. Disponible en: <http://www.hispasonic.com/tutoriales> [Consulta: junio de 2013]
43. Pyckett, C. (2006). *How synthesizers works* [Sitio web]. Disponible en: <http://www.pykett.org.uk/synths.htm> [Consulta: junio de 2013]
44. Homemusician.net. *Colección de bancos de instrumentos en formato sf2* [Sitio web]. Disponible en: <http://soundfonts.homemusician.net/> [Consulta: junio de 2013]

Apéndice A: Implementación

A.1 melParams

Uno de los parámetros de entrada es la estructura de datos melParams mediante la cual se determinan o configuran distintos valores del software. Los datos que se hallan y su utilidad son:

- `style`. Estilo musical de la obra. Es empleado para seleccionar la configuración de instrumentos y efectos.
- `subStyle`. Subestilo musical dentro del estilo. Por ejemplo el género dance dentro de la música electrónica. Es usado también para la configuración de instrumentos y efectos.
- `baseDir`. Dirección relativa a donde se situarán los archivos a generar.
- `emuOFormats`. Vector de celdas con los formatos de audio a obtener. Estos pueden ser MIDI, WAV, MP3 y FLAC en la versión actual.
- `emuTemplate`. Parámetro para la obtención de efectos musicales mediante el método de duplicación de canal.
- `emuSoundFont`. Dirección de los archivos de paletas de instrumentos.
- `emuHumanize`. Pares de valores que definen el tipo de humanización (*velocity*, *durations*) y el valor para construir la curva de error.
- `maxVelocity`. Parámetro para la configuración del campo *velocity*. Define la intensidad máxima de golpeo para cada instrumento.
- `minVelocity`. Parámetro para la configuración del campo *velocity*. Define la intensidad mínima de golpeo para cada instrumento.
- `sequenceVelocity`. Parámetro para la configuración del campo *velocity*. Define la secuencia de intensidades de golpeo para cada instrumento.
- `timerVelocity`. Parámetro para la configuración del campo *velocity*. Define el tiempo de repetición de la secuencia de golpeo.
- `preset`. Parámetro que define el archivo *sf2* y la localización del mismo para cada instrumento.
- `equalInstrumentChord`. Parámetro utilizado en el estilo audiovisual que define si los tres instrumentos que forman el acorde son iguales o no.
- `instrumentsMidiSet`. Vector que define el número del instrumento MIDI a utilizar en el archivo final de dicho formato.
- `duplicate`. Parámetro para definir si se quiere duplicar un instrumento, siendo el valor indicado el incremento o decremento de altura del instrumento duplicado. En caso negativo el valor sería NaN (*not a number*).

A.2 modPitch y duplicate

Con la función `modPitch` se modifica la altura de un instrumento. La llamada a dicha función es `pitches=modPitch(pitches, instruments, instrumentMod, defaultPitch)`, siendo los parámetros de entrada una matriz de alturas, un vector de instrumentos correspondiente a las alturas, un vector de los instrumentos a modificar y la cantidad a sustraer o añadir a las alturas respectivamente. Esta función es usada para corregir los instrumentos transponedores, aquellos cuya la altura definida en la partitura no corresponde con su sonoridad real.

Como efecto extra dicha función se usa cuando el valor de `duplicate` así lo requiere. En caso de uso, generalmente el valor de `duplicate` es 5 para crear un duplicado de un instrumento y formar con el original una quinta perfecta. Para que sea posible dicha funcionalidad se añade el siguiente código en el interior del bucle principal donde se genera cada instrumento:

```
if ~isnan(melParams.duplicate(k))
    dup=1;
    %% Configuración de panorama %%
else
    dup=0;
end
for j=1:1+dup
    if dup
        themeAux=[num2str(j) 'b' theme];
        if j==2
            pitchBlock=modPitch(pitchBlock,instrumentBlock,...
                instrumentBlock, melParams.duplicate(k))
        end
    else
        themeAux=theme;
    end
    %% Generación del instrumento %%
    if dup && j==2
        panEffect([melParams.baseDir filesep themeAux ...
            int2str(k)], left2, right2);
        mixWavConverter([themeint2str(k)], 1, 2, '*', ...
            instruments(k), melParams);
        instrumentEffect([melParams.baseDir filesep theme ...
            int2str(k)], instruments, melParams, k);
    end
end
```

```

elseif dup && j==1
    panEffect([melParams.baseDir filesep themeAux ...
int2str(k)], left1, righth1);
end
end
end

```

Primeramente se comprueba si se debe realizar duplicación. De no ser así el proceso continuará de la forma normal. En caso afirmativo se establecen unos valores de panorama para usar con el instrumento original y el duplicado. Tras esto se genera el instrumento original y se le aplica el panorama establecido. Posteriormente se crea el instrumento duplicado modificando su altura, mediante la función `modPitch`, para formar una quinta perfecta u otro efecto si el valor difiere de 5. Para concluir se aplica el panorama al duplicado y se mezclan ambos con la misma potencia (0.5) y se aplican el resto de efectos definidos para el instrumento tratando la mezcla del instrumento original y su duplicado como si del instrumento original se tratara respecto al resto de instrumentos en la mezcla.

A.3 Script para lanzamiento en clúster

Para realizar pruebas de lanzamiento en el clúster usando Matlab se uso el siguiente script:

```

base=pwd;
path_src = {base,[base filesep 'emulation'],[base filesep ...
'emulation' filesep 'audio'],[base filesep 'emulation' ...
filesep 'utils'],[base filesep 'emulation' filesep ...
'xml_ly'], [base filesep 'resources'],[base filesep ...
'resources' filesep 'synthw'],[base filesep 'utils'],};
jm = findResource('scheduler', 'type', 'torque');
job = createJob(jm);
set(job, 'PathDependencies', path_src);
for i=1:nodes
    createTask(job, @makeMP3, 3, {pInfo{i}, themeName{i}, ...
fileMat{i}});
end
submit(job);
ok=waitForState(job, 'finished',12000);
out = getAllOutputArguments(job);
destroy(job);

```

Con este script en concreto se realiza la generación de múltiples piezas musicales en paralelo. Para ello se busca el recurso Torque, se crea un trabajo en él y se le asigna el

path donde se encuentra el código necesario para ejecutar las tareas. Posteriormente se crean tantas tareas como se indique con el parámetro `nodes` y asignándoles la función a ejecutar y sus parámetros de entrada. Tras esto se lanza el trabajo y se espera a que éste finalice o transcurra el tiempo de espera establecido. Por último se recogen los resultados (también pueden adquirirse los errores), aunque en este caso la función `makeMP3` no devuelve nada ya que solo crea el archivo MP3, y se destruye el trabajo.

Otra opción es comunicarse directamente con Torque, para ello se ejecutaría un bucle parecido al definido anteriormente para crear tareas.

```
for i=1:nodes
    system(['qsub ' dirMelomics '/resources/scriptMP3.sh -v ...
        "dirMelomics=' dirMelomics ',cDir=' cDir ...
        ',rutaMat=' rutaMat{i} '"']);
end
```

En este bucle se establece como tarea la ejecución del archivo `scriptMP3.sh` que tendría el siguiente código:

```
export
PATH=$PATH:/opt/matlab/bin:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
(echo "cd('${dirMelomics}'); pathMELOMICS; cd('${cDir}');
makeMP3('${rutaMat}');exit;" | matlab)
```

Con este script se determina el *path* de Matlab y se ejecuta una llamada a Matlab donde se realizan las instrucciones indicadas. Para controlar si dichas tareas lanzadas han sido concluidas la función ejecutada en cada nodo crea un archivo de log donde devuelve los resultados o el error obtenido, ficheros que se comprueban desde el nodo central hasta la finalización de todas las tareas o el transcurso de un tiempo de salida.

Estas son dos formas sencillas de lanzamiento en clúster de computación, siendo de gran utilidad para una mayor velocidad de ejecución, sobre todo en un software como éste, el cual genera gran cantidad de archivos temporales, la escritura de estos localmente en cada nodo del clúster de computación y su traslado posterior a la localización principal de los resultados finales.

Apéndice B: Manual de generación de una paleta de instrumentos

B.1 Generación paso a paso

La generación de un *preset* se divide en 3 pasos: grabar los *samples* de sonido y guardarlos como *sf2*, editar los parámetros del *SoundFont* y configurar un *preset* de instrumentos. Algunos de los pasos tienen diferentes formas de actuación o incluso pueden no ser necesarias. A continuación se describe paso a paso cómo generar un *preset* de varios instrumentos utilizando el programa *Extreme Sample Converter* (ESC) para el primer paso y Viena para los 2 siguientes.

B.1.1 Crear archivo sf2

Se realiza usando el programa ESC y existen dos modos para llevar a cabo dicha tarea.

Con el primer modo se crea el archivo *sf2* por conversión directa, y se utiliza siempre que el instrumento a utilizar se encuentre en uno de los formatos de origen soportados, los cuales están indicados en la pestaña ‘Source Format’ (Figura B.1, C). Un ejemplo sería convertir un instrumento de la librería de Kontakt 4, para ello solo hay que localizar el archivo del instrumento en el explorador (Figura B.1, A) situado en la

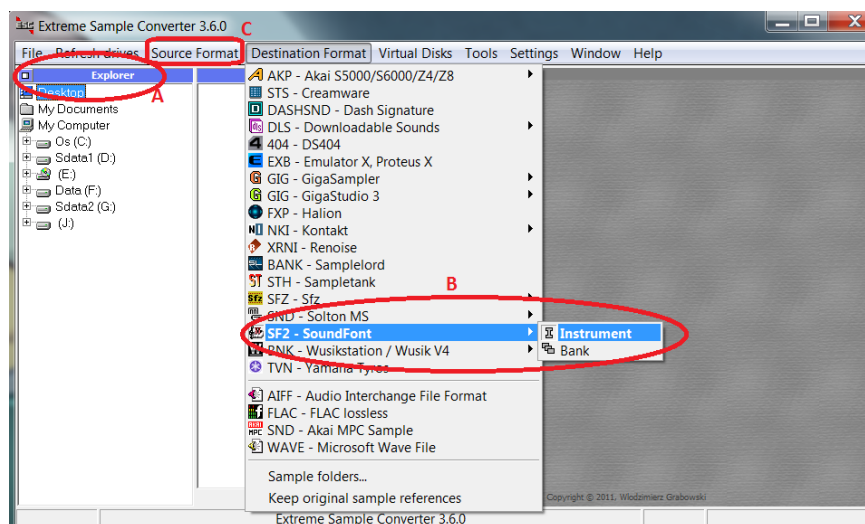


Figura B.1. Generación de preset, Extreme Sample Converter

ventana izquierda, establecer como formato destino *sf2* en “Destination Format → SF2-SoundFont → Instrument” (Figura B.1, B) y pulsar el botón ‘Convert’ (Figura B.4, B) que aparecerá una vez seleccionado un instrumento. Al realizar la conversión de esta

forma se obtendrá un *SoundFont* con los parámetros DAHDSR aplicados, así como los bucles que existieran en el formato origen.

El segundo modo se emplea para *VSTi* de un formato no soportado (Kontakt 5 por ejemplo), para ello se deben seguir los siguientes pasos:

1. Al igual que con el primer modo, al abrir *Extreme Sample Converter* lo primero es seleccionar el formato destino. Para ello se selecciona “Destination Format → SF2-SoundFont2 → Instrument” (Figura B.1, B). Tras esto pulsar en “Source Format → VSTi Converter”, y se abrirá una pantalla de configuración donde crear los *samples* de los sonidos que se seleccionen (Figura B.2).
2. Configurar la carpeta temporal donde se guardarán los archivos temporales. Esto se hace en “Menu → Temp Folder” (Figura B.2, A). Este paso solo hay que realizarlo la primera vez, o si se desea cambiar la carpeta temporal.
3. En la ventana *VSTi Converter* pulsamos sobre “Menu → VST Instruments →

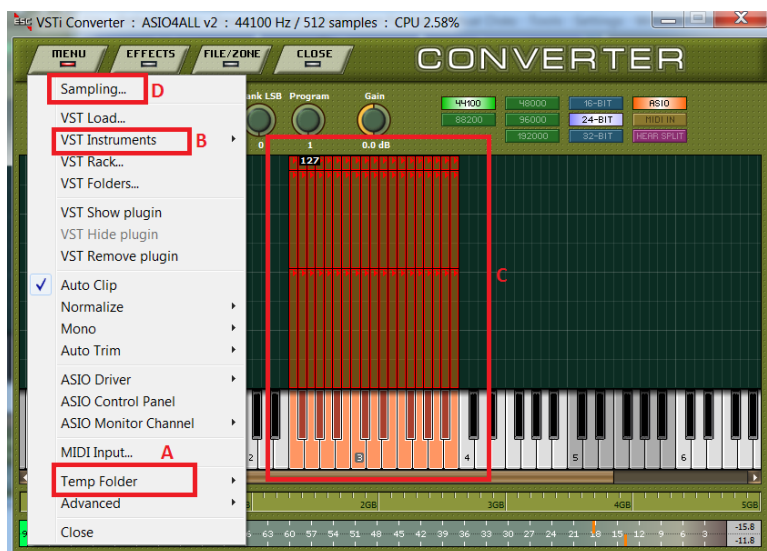


Figura B.2. Generación de preset, ESC, imagen 2

Kontakt 5” (Figura B.2, B) o cualquier otro VST. Dentro del VST elegido se selecciona el instrumento a reproducir y se configura. Esto se hará de distinta forma según el VST usado.

- Una vez seleccionado el instrumento a *samplear* se escogen con el botón izquierdo del ratón las notas cuyos sonidos aparecerán en el *sf2* (Figura B.2, C). También pueden elegirse distintos rangos de *velocity*, para lo cual destacamos las barras a modificar y se arrastra desde la parte superior de la barra (justo encima del triángulo o símbolo de reproducir) hasta el valor de *velocity* deseado. En la pestaña 'FILE/ZONE' (Figura B.3, A) podemos dividir la zona en un número predeterminado de campos de *velocity* diferentes y equidistantes (desde 2 a 32 *splits*). En la misma pestaña tenemos la opción de grabar una zona diseñada o cargar una previa ('Save...' y 'Open...').

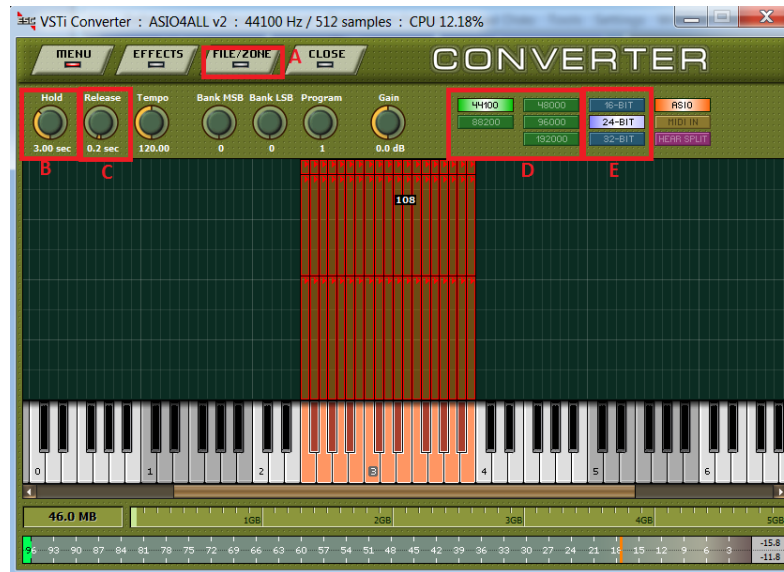


Figura B.3. Generación de preset, ESC, imagen 3

- Configurar el tiempo de grabación para cada *sample* de sonido con las ruedas 'Hold' (Figura B.3, B) y 'Release' (Figura B.3, C). El primero señalará cuánto tiempo mantener el sonido mientras graba y el segundo cuánto grabar de cola (sonido que permanece al soltar la tecla).
- Para terminar con esta pantalla y realizar la grabación se configura la calidad del *sample* seleccionando la frecuencia de muestreo (Figura B.3, D) y el número de bits (Figura B.3, E), 44100 y 16 bits es un valor común. Luego se presiona en "Menu -> sampling ..." (Figura B.2, D) y se muestra un cuadro de dialogo donde establecer el nombre al *sample* ('Sample name' y 'Preset name').
- Una vez generados los WAV, en la pantalla principal de *Extreme Sample Converter* se mostrará un archivo *sfz* (Figura B.4, A) con el nombre indicado al *samplear*. Se selecciona y pulsa sobre el botón 'Convert' (Figura B.4, B) que aparece en la parte inferior, tras lo cual aparece un cuadro de diálogo para indicar la ruta de guardado y nombre del archivo *sf2*. Finalizado el proceso ya estará creado el instrumento en formato *sf2*.

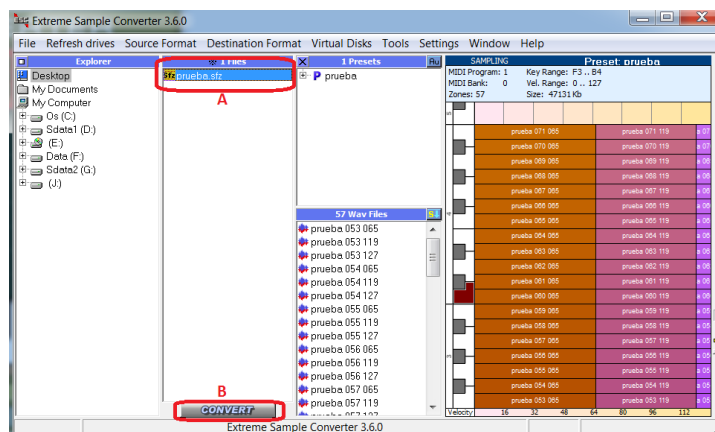


Figura B.4. Generación de preset, ESC, imagen 4

B.1.2 Edición de archivo sf2

Los archivos generados de forma directa se encontrarán configurados, por lo que solo se modifican si se considera necesario para mejorar alguno de los aspectos. Sin embargo, los archivos creados mediante la grabación con el modo *VSTi Converter* aparecen vacíos todos los campos del archivo *sf2* salvo la asignación de teclas y *velocity*. Los parámetros para poder utilizar estos *SoundFont* con una calidad mínima básica y de forma rápida y sencilla basta con configurar los valores de *delay*, *attack*, *hold*, *decay*, *sustain* y *release*, en las envolventes correspondientes. Los pasos a seguir son:

1. Pulsar en el botón 'Open File' (Figura B.5, A) y seleccionar el instrumento *sf2* deseado. Una vez abierto seleccionar la pestaña 'Global' (Figura B.5, B) situada al final de la jerarquía que comienza con 'Instruments'.
2. El primer valor a modificar es 'Vol Env Release' el cual suele estar con el valor 0.1 por defecto. Dicho valor indica durante cuánto tiempo decrecerá el sonido hasta dejar de sonar una vez se deja de pulsar una tecla del instrumento (mensaje MIDI 'Note Off') y así evitar una terminación abrupta. Para facilitar el añadir dichos valores lo habitual es añadirlo en la columna 'Global' (Figura B.5, C), valor que se aplica a todas las columnas sin valor específico, de no ser así se aplica el valor por defecto indicado junto al nombre de cada parámetro. Para modificarlo de esta forma se selecciona el primer recuadro con valor específico (Figura B.5, E) y se pulsa el botón 'Move values to the global split' (Figura B.5, F) con el que el valor más repetido pasará a ser el valor de 'Global' y el cual hay que modificar según se considere adecuado.

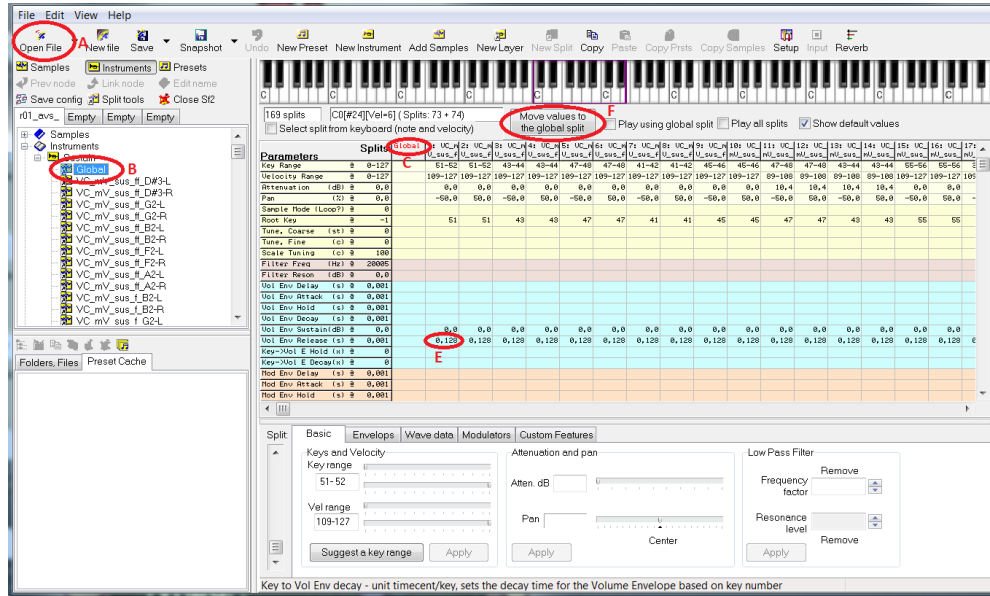


Figura B.5. Generación de preset, Viena

3. Los valores *hold*, *decay* y *sustain* definen cómo se comporta el sonido una vez alcanzada la amplitud máxima. Con *attack* se establece el tiempo hasta alcanzar el volumen máximo y con *delay* cuánto tarda en comenzar a reproducirse el sonido tras pulsar una tecla (mensaje MIDI ‘Note On’). El *decay* indica el tiempo transcurrido hasta alcanzar el valor en decibelios marcado por *sustain* y al cual se mantiene el sonido hasta la aplicación de *release*. Todos estos valores se deben configurar según cada instrumento para obtener un mejor resultado.

En este paso se define cómo se reproducirán los sonidos grabados para cada nota MIDI empleada.

B.1.3 Crear una paleta de instrumentos

Los archivos *sf2* creados hasta ahora son *preset* (paletas) de instrumentos de un único instrumento, por lo que el siguiente paso es unirlos en un único archivo.

1. Pulsar en ‘New File’ (Figura B.6, A) y establecer el nombre para el *preset*.
2. Seleccionar la pestaña contigua a la creada que aparece cómo *Empty* (Figura B.6, B) y abrir uno de los instrumentos a unir al *preset*.

3. Volver a la pestaña del *preset* y pulsar el botón de ‘New Preset’ (Figura B.6, C) de la parte superior. En la nueva ventana seleccionar el archivo abierto

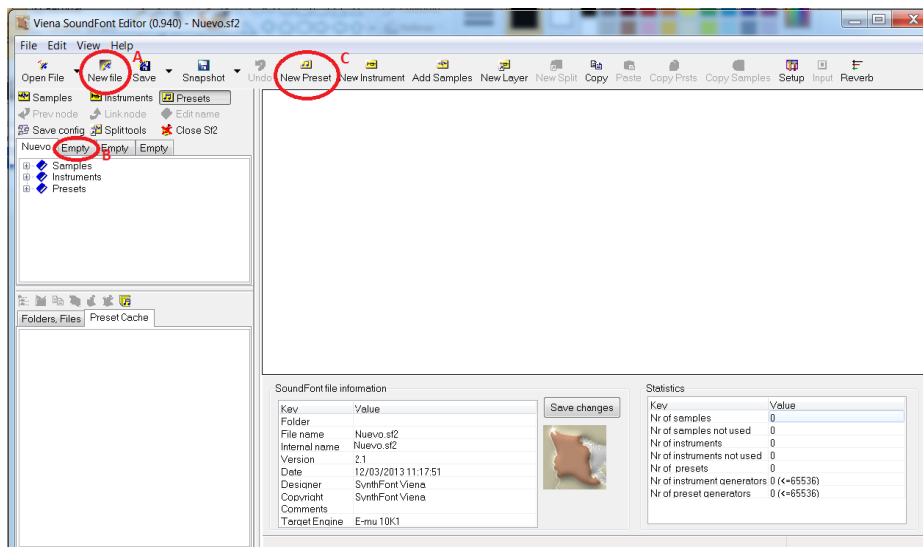


Figura B.6. Generación de preset, Viena, imagen 2

previamente (Figura B.7, A), marcar el instrumento a grabar (Figura B.7, B) y asignarle en el campo ‘Preset’ (Figura B.7, C) el número de instrumento MIDI (0-127). El campo ‘Name’ (Figura B.7, D) también se puede personalizar para una mejor identificación del instrumento.

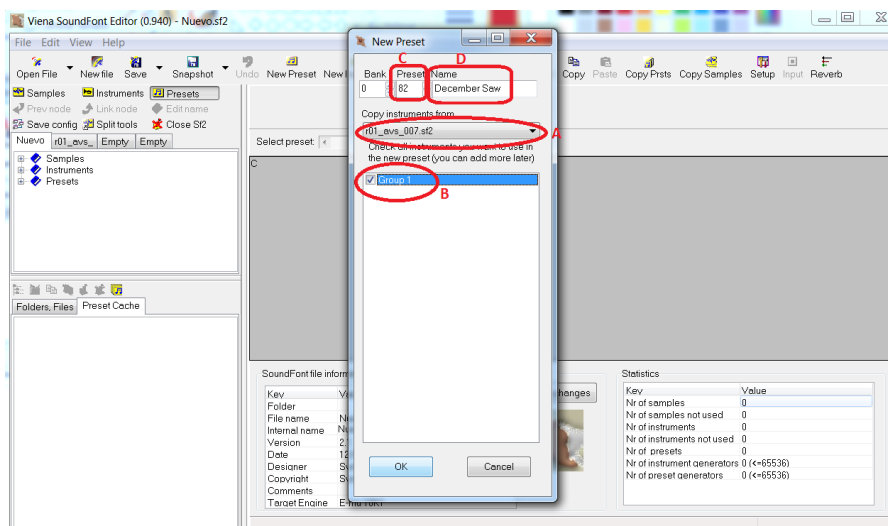


Figura B.7. Generación de preset, Viena, imagen 3

4. Repetir pasos 2 y 3 hasta que tener todos los instrumentos deseados. Para introducir más de 128 instrumentos se crea un nuevo banco dentro del mismo archivo *sf2*, estableciendo un nuevo valor en el campo ‘Bank’ al introducir nuevos instrumentos.

- Una vez agregados todos los instrumentos (Figura B.8, B) pulsar el botón ‘Save’ (Figura B.8, A), elegir una ubicación y la paleta de instrumentos estará disponible para su uso, seleccionando cada instrumento mediante la configuración del archivo MIDI sobre el que se aplique el preset.

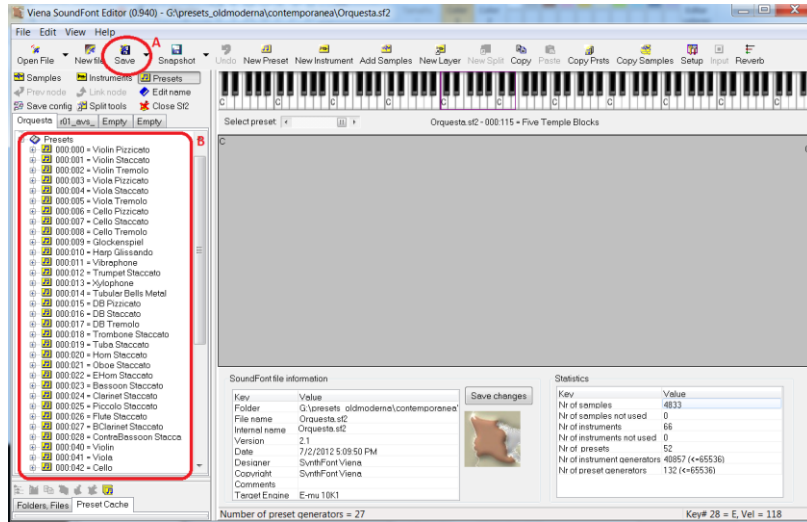


Figura B.8. Generación de preset, Viena, imagen 4