



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de un instrumento virtual del timble con  
tecnología VST: síntesis digital y muestreado.

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e  
Imagen

AUTOR/A: Luis Mesa, Néstor

Tutor/a: Sansaloni Balaguer, Trinidad M<sup>a</sup>

CURSO ACADÉMICO: 2023/2024

*A mis abuelos, siempre.*

Quisiera agradecer a todas las personas que han hecho posible este proyecto:

En primer lugar, agradecer a Juani, Teo y Mario,  
a mi familia, por siempre apoyarme.

Agradecer a Diego y a Gara, por su compañía y apoyo durante todo este proceso.  
A ellos y a todas las amistades que me han ayudado a conseguirlo.

A Alberto por la compañía y las fotografías.

A Haridian por haberme hecho llegar hasta aquí,  
no podría haber sido de otra manera.

Agradecer a mi tutora, Trini, por confiar en el proyecto, cuando yo no lo hice.  
Y a Rubén Picó por su gran ayuda.

A Dani Armijo por su ayuda con su instrumento en las sesiones de grabación.  
Y a Mario Martín por su sabiduría y por cederme su timple.

A José Luis Henández y Adrián Pérez por sus consejos  
y por allanar el camino, gracias.

A Jorge por no dejar que me rindiera  
y a Dios por la fortaleza.

A todos, gracias de corazón.

## Resumen

Este Trabajo Final de Grado tiene como objeto el desarrollo de un instrumento virtual de timple, utilizando dos de las técnicas más comunes, por un lado, la síntesis digital mediante modelado espectral aditivo del instrumento y, por otro lado, la creación de un banco de sonidos MIDI mediante el muestreo del instrumento.

El trabajo propone un esquema para el adecuado microfoneado del timple, y explica el proceso de grabación de muestras de las notas y dinámicas necesarias para la confección del plugin. En la edición, posprocesado, mezcla y automatización de las muestras captadas, se ha utilizado el programa *Logic Pro* y para su mapeado y empaquetado en un plugin VST, el programa *Maize Sampler*.

El sintetizador por modelado espectral del tipo aditivo se ha desarrollado utilizando el *add-on* de *Matlab AudioToolbox*. A partir de las muestras, se extraen las características necesarias para, basándose en la serie de Fourier, reconstruir la señal.

Finalmente, se ha realizado un estudio calificativo de la calidad del sonido que consigue el plugin a través de encuestas de opinión.

## Palabras Clave

Timple, VST, Instrumento Virtual, *Sampler*, Sintetizador de audio

## Abstract

This work aims to develop a virtual timple instrument, using the two most common techniques, on the one hand, digital synthesis through physical modeling of the instrument and on the other hand, the creation of a MIDI sound bank through sampling of the instrument.

The project proposes a scheme for properly microphoning the timple and explains the process of recording samples of the necessary notes and dynamics for the plugin's creation. The captured samples were edited, post-processed, mixed, and automated using Logic Pro. They were then mapped and packaged into a VST plugin using Maize Sampler.

The additive spectral modeling synthesizer was developed using Matlab's AudioToolbox add-on. From the samples, the necessary characteristics are extracted to reconstruct the signal based on the Fourier series.

Finally, a qualitative study of the sound quality achieved by the plugin was conducted through opinion surveys.

## Keywords

Timple, VST, *Virtual Instrument*, *Sampler*, *Audio synthesizer*

## **Glosario de acrónimos**

**ADSR:** *Attack, Decay, Sustain, Release.*

**DFT:** Discrete Fourier Transform. Transformada Discreta de Fourier.

**FFT:** *Fast Fourier Transform.* Transformada Rápida de Fourier.

**MIDI:** *Musical Instrument Digital Interface.* Interfaz digital para instrumentos musicales.

**ORTF:** *Office de Radiodiffusion Télévision Française.*

**TF:** Transformada de Fourier.

**VST:** *Virtual Studio Technology.*

---

# Índice

<b>Glosario de acrónimos .....</b>	<b>IV</b>
<b>1. Introducción.....</b>	<b>1</b>
<b>1.1. Motivación y justificación del trabajo .....</b>	<b>1</b>
<b>1.2. Estado del arte.....</b>	<b>1</b>
<b>1.3. Objetivos .....</b>	<b>2</b>
O.P. Objetivo principal.....	2
O.S. Objetivos secundarios.....	2
<b>1.4. Metodología.....</b>	<b>2</b>
<b>2. Marco Teórico .....</b>	<b>4</b>
<b>2.1. El Timple.....</b>	<b>4</b>
2.1.1. El timple y su origen .....	4
2.1.2. Características del instrumento y tipos de timple .....	5
2.1.3. El sonido del timple.....	8
<b>2.2. Instrumentos virtuales.....</b>	<b>9</b>
2.2.1. <i>Virtual Studio Technology (VST)</i> .....	9
2.2.2. <i>Musical Instrument Digital Interface (MIDI)</i> .....	10
<b>2.3. Samplers y Sintetizadores de Sonido.....</b>	<b>13</b>
2.3.1. Procesamiento de muestras grabadas ( <i>sampling</i> ).....	15
2.3.2. Modelado espectral: síntesis aditiva .....	16
<b>3. Desarrollo práctico .....</b>	<b>21</b>
<b>3.1. Herramientas utilizadas .....</b>	<b>21</b>
3.1.1. Instrumentos musicales .....	21
3.1.2. Equipos y materiales de grabación .....	21
3.1.3. Software utilizado .....	25
<b>3.2. Sampler .....</b>	<b>26</b>
3.2.1. Proceso de grabación .....	26
3.2.2. Procesado de muestras .....	28
3.2.3. Implementación del instrumento virtual .....	32
3.2.4. Pruebas y ajustes .....	35
<b>3.3. Síntesis Aditiva .....</b>	<b>37</b>
3.3.1. Desarrollo .....	37
3.3.3. Pruebas y ajustes .....	43
<b>4. Validación de resultados .....</b>	<b>44</b>
<b>4.1. Planteamiento de la encuesta de evaluación.....</b>	<b>44</b>
<b>4.2. Resultados Obtenidos.....</b>	<b>44</b>

<b>5. Conclusiones .....</b>	<b>47</b>
<b>6. Trabajo Futuro .....</b>	<b>48</b>
<b>6.1. Mejora del Plugin de Reproducción de Muestras Pregrabadas.....</b>	<b>48</b>
<b>6.2. Desarrollo de Prototipos de Síntesis Aditiva .....</b>	<b>48</b>
<b>6.3. Comercialización y Usabilidad .....</b>	<b>48</b>
<b>Referencias .....</b>	<b>50</b>
<b>Índice de Figuras.....</b>	<b>52</b>
<b>Índice de Tablas.....</b>	<b>54</b>
<b>Índice de Ecuaciones .....</b>	<b>54</b>
<b>Índice de Anexos.....</b>	<b>54</b>

# 1. Introducción

## 1.1. Motivación y justificación del trabajo

Este Trabajo Final de Grado (TFG) nace del interés de llevar un instrumento como el timple a los entornos de composición y producción musical digitales.

La idea surge a partir de un proyecto personal de carácter audiovisual que se encuentra en proceso de ideación, el cual se centra en explorar el sonido del instrumento a través de un viaje por las Islas Canarias. En dicho proyecto, es indispensable la incorporación del timple en la banda sonora para capturar la esencia cultural y musical de la región. Sin embargo, para poder presentar el proyecto a empresas y entidades que quieran colaborar en su financiación, es crucial disponer previamente de una maqueta que incluya este instrumento.

El timple es un instrumento musical de cuerda pulsada derivado de la guitarra barroca originario de las Islas Canarias. Esta herramienta considerada imprescindible del folclore local de las islas es protagonista tanto de melodías, como del acompañamiento en piezas musicales en la música popular canaria, y se ha ido introduciendo en otros géneros como el *pop* y el *hip-hop*.

En el proceso de composición musical, muchas veces se necesita escuchar la pieza o generar maquetas para distribuidores o productores musicales. Una opción que abarata costes y lo permite es la simulación mediante instrumentos virtuales ya sea en programas de notación musical o con el uso de DAWs mediante tecnología MIDI.

A pesar de la creciente disponibilidad de instrumentos virtuales en las estaciones de trabajo de audio digital (DAWs), no existe en la actualidad una opción que reproduzca de manera realista y auténtica el sonido característico del timple. Esta carencia dificulta la creación de maquetas realistas que capturen su timbre y resonancia únicos.

En este contexto, compositores y arreglistas principalmente canarios se encuentran con la dificultad de encontrar algún instrumento virtual de sonido similar al timple y suelen recurrir a instrumentos sin conseguir el sonido deseado.

## 1.2. Estado del arte

En el inicio del desarrollo de este trabajo, no existían soluciones disponibles para integrar el timple, un instrumento musical tradicional de Canarias, en los entornos digitales de producción musical o composición. A pesar de los avances tecnológicos en la música digital, este instrumento seguía sin ser representado en las herramientas y *plugins*<sup>1</sup> que utilizan músicos y productores para crear y editar música.

Aún hoy en día, no se ha desarrollado un instrumento virtual específico para modelar el sonido del timple y poder usarlo en las estaciones de trabajo de audio digital (DAWs). Sin embargo, en mayo de 2024, se ha dado un paso significativo en su integración digital. Adrián Pérez ha logrado llevar el timple al entorno de *MuseScore*, un software gratuito y de código abierto de notación musical ampliamente utilizado por arreglistas y compositores.

---

<sup>1</sup> Componente de software que agrega una función específica a un programa existente.



La inclusión del timple en *MuseScore* por Pérez consistió en añadir los parámetros del instrumento en el programa para que apareciera en el listado de instrumentos por defecto. “Aunque *MuseScore* permite a los usuarios parametrizar instrumentos de cuerda como la guitarra (número de cuerdas, afinación, etc.), hacerlo manualmente es un proceso para usuarios avanzados” (A. Pérez, comunicación personal, 2024). Adrián simplificó este proceso añadiendo el timple directamente en el archivo `instruments.xml` y ajustando la estructura interna del programa para que el sonido del timple (basado en la modificación de la guitarra de cuerdas de nylon ya incorporada) se reproduzca correctamente. “Lo que he hecho simplemente es añadir los parámetros necesarios para reproducir el sonido que le he asignado al timple (guitarra de cuerdas de nylon)”.

Este trabajo implicó compilar *MuseScore* en diferentes sistemas operativos: *Linux*, *Windows* y *macOS*. A la fecha, los cambios realizados por González están pendientes de ser incorporados oficialmente al proyecto *MuseScore* mediante un *Pull Request (PR)*.

Con todo esto se puede considerar que el resultado de este proyecto será el primer VSTi que simule el timple generado hasta la fecha.

### **1.3. Objetivos**

Los objetivos que se pretenden alcanzar en este Trabajo Final de Grado son los siguientes.

#### **O.P. Objetivo principal**

Crear un instrumento virtual de timple para su uso en estaciones de trabajo de audio digital.

#### **O.S. Objetivos secundarios**

**O.S.1.** Descubrir las características constructivas y sonoras, así como tipos y materiales del timple canario.

**O.S.2.** Proponer un esquema de microfoneado del instrumento para su grabación en estudio.

**O.S.3.** Grabar, editar, procesar y mapear muestras de audio para la creación de un instrumento virtual por reproducción de muestras pregrabadas.

**O.S.4.** Sintetizar por medio de modelado espectral aditivo la señal de audio del timple canario.

**O.S.5.** Validar y evaluar de manera comparativa de ambas soluciones.

### **1.4. Metodología**

En el desarrollo de este trabajo, se distingue una parte teórica, de recopilado de información, y una parte práctica, de implementación.

En la primera fase del proyecto, se ha realizado una contextualización teórica a través de una revisión bibliográfica de los temas que se consideran de importancia para el desarrollo práctico. En primer lugar, se profundiza en el origen, características y sonido del instrumento. A continuación, se estudian las tecnologías más comunes en el control de *plugins*: MIDI y VST. Tras ello se realiza una aproximación teórica sobre la síntesis digital y la toma de muestras que se utilizan para crear instrumentos virtuales. Para finalizar se realiza una revisión del estado del arte en este tipo de soluciones.

En la segunda parte, para la implementación de los instrumentos virtuales, se han utilizado dos técnicas: la síntesis aditiva y el muestreo del instrumento para crear un banco de sonidos MIDI. La elección de estas técnicas se basa en su capacidad para capturar y reproducir de manera fiel las características sonoras únicas del timble y su viabilidad de implementación en el contexto de un TFG.

La toma de muestras se realiza en el estudio de doblaje de la Escuela Politécnica Superior de Gandía (EPSG) donde se selecciona el material disponible adecuándolo a las necesidades. Posteriormente se utilizan las muestras de notas y dinámicas necesarias para acabar con el mapeo de estas en un software de creación de *plugins* VST.

El segundo método, la síntesis digital, se basa en el análisis de las muestras grabadas, con ellas se realizó un análisis frecuencial mediante la transformada de Fourier y de su envolvente de amplitud mediante la transformada de Hilbert con el fin de extraer los datos necesarios para poder realizar su síntesis mediante la suma de senoidales simples.

Ambas técnicas fueron evaluadas para determinar cuál ofrecía una representación más precisa y funcional del timble. También se realizaron pruebas de audición con músicos y personas familiarizadas con el sonido del instrumento para valorar la fidelidad y calidad sonora de los modelos, y con la retroalimentación recibida, se pudieron realizar los últimos ajustes.

## 2. Marco Teórico

### 2.1. El Timple

#### 2.1.1. El timple y su origen

El timple es todo un emblema cultural de las Islas Canarias, reflejando su espíritu y rica tradición. Desde sus orígenes humildes, ha evolucionado significativamente como se estudia en González Casal (2017), con cambios en el tamaño de su caja, el tiro de cuerda útil y la incorporación de cuerdas prefabricadas, lo que ha mejorado su sonido. Como expone el Gobierno de Canarias (2023), este pequeño instrumento de cinco cuerdas, similar a una guitarra, es fundamental en la música folclórica canaria, especialmente como acompañamiento en las parrandas.

Este instrumento tiene un origen incierto y diversas influencias europeas y americanas. Aunque históricamente su uso fue más común en las islas orientales, hoy está presente en todo el archipiélago.

La primera referencia que existe del uso del timple en Canarias la encontramos descrita en el diario personal del obispo Antonio Tavira en 1792. Tavira describe una visita realizada a Fuerteventura, “donde el “tipilillo” servía de elemento sonoro que acompañó a la famosa “danza de las espadas”” (Casa Museo del Timple, s. f.). A partir de ahí se suceden las citas, siendo mucho más abundantes desde la segunda mitad del s. XIX. En este siglo el término evoluciona a timple y a mediados del mismo comienza la construcción de manera más sistemática en Lanzarote, con el considerado el primer constructor de timples Simón Morales Tavío.



**Figura 1.** *Timple Canario.* Nota: Foto de Josemchi. Extraído de: Adobe Stock.

Usada bajo licencia de Adobe Stock.

[https://stock.adobe.com/es/search?k=timple&search\\_type=usertyped&asset\\_id=651150986](https://stock.adobe.com/es/search?k=timple&search_type=usertyped&asset_id=651150986)

González Casal (2017) explica que, aunque no existen documentos que confirmen su procedencia exacta, está muy clara y aceptada la relación del timple con la familia de las guitarras y vihuelas de la época del renacimiento y barroco europeo. Por su carácter estratégico geográficamente hablando, Canarias era una zona de paso para las embarcaciones europeas que cruzaban el Atlántico hacia América. Los europeos también llevaban entre sus pertenencias instrumentos musicales, por lo que se encuentran coincidencias en antecedentes, tanto en Europa como en América. Así, se localizan instrumentos como el “guitarrillo”, el tiple o la mandolina, el charango en Argentina o

Perú, el cavaquinho en Brasil o el cuatro en Venezuela. Esta conexión pone de manifiesto que nuestro actual timple es el resultado de la evolución de diversos instrumentos de cuerda y del contacto entre todos estos pueblos.

Inicialmente, era un instrumento de acompañamiento tocado rasgueando las cuerdas, pero con el tiempo, los timplistas actuales han desarrollado una mezcla de punteos, arpeggios y rasgueos, demostrando su versatilidad y evolución (Viera Ruíz, 2015).

Dada su función, en un principio eran más sencillos en cuanto a tamaño y número de trastes, por lo que no requerían de una gran habilidad técnica. Actualmente su desarrollo ha permitido su utilización como papel solista en distintas formaciones y estilos de música. Ejemplos de ello son grupos que utilizan el timple como instrumento solista como Guigan y timplistas que han consolidado su carrera como solistas, incluso estrenando obras con orquesta sinfónica como Domingo “El Colorao”, José Antonio Ramos o Benito Cabrera.

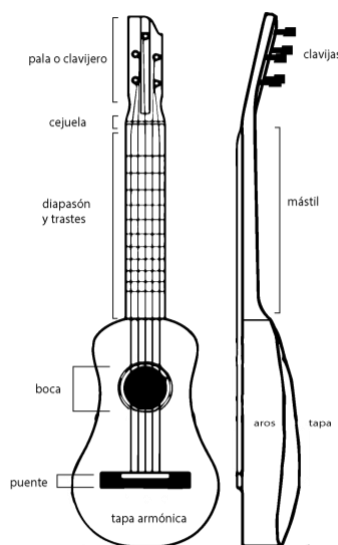
### 2.1.2. Características del instrumento y tipos de timple

#### *Características generales*

“Se puede considerar y clasificar al timple dentro de la rama del grupo de cordófonos compuestos” (González Casal, 2017). Las cuerdas están fijadas a los puentes y tensadas mediante la rotación de las clavijas. Las vibraciones se transmiten por la madera y son amplificadas por la caja de resonancia.

#### *Partes del timple*

El timple se compone de diversas partes esenciales para su funcionamiento y estética (véase Figura 2 **Figura 2.** Partes del timple. Figura de elaboración propia.). La pala o clavijero alberga las clavijas que tensan las cuerdas para afinarlas. La cejuela es una pieza suelta en el mástil que eleva las cuerdas. El diapasón y los trastes, ubicados en el mástil, permiten tocar las notas presionando las cuerdas. La tapa armónica facilita la vibración del sonido; incluye la boca, un orificio para la salida del sonido, y el puente, donde se amarran las cuerdas. El mástil debe ser plano para facilitar la ejecución. Los aros y la tapa posterior forman el cuerpo del timple, contribuyendo a su sonoridad.



**Figura 2.** Partes del timple. Figura de elaboración propia.

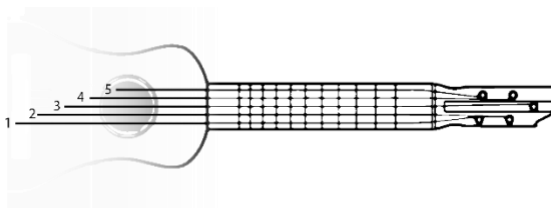
### *Materiales*

El timple es un instrumento artesanal, construido por maestros de la luthería. Esto influye significativamente en su precio, que ronda los 250 euros en los tipples más básicos y de iniciación, y puede llegar hasta cerca de los 1000 euros. Sin embargo, se traduce en instrumentos únicos entre sí y de mayor calidad. Se construye con una variedad de maderas seleccionadas por sus propiedades acústicas, estéticas y facilidad de trabajo, influenciadas por la disponibilidad geográfica y las preferencias del luthier.

Según el estudio de materiales realizado por Viera Ruíz (2015) las maderas más utilizadas para las tapas armónicas se encuentran el cedro canadiense y el pino abeto alemán, conocidas por su excelente propagación del sonido. Para la tapa posterior y aros, se prefieren maderas más oscuras y duras como el palo santo de la India y de Río, que aportan durabilidad y calidad acústica, aunque enfrentan problemas de escasez. Otros materiales incluyen ciprés, arce y maderas frutales como el nogal y el peral. Innovaciones recientes han incorporado materiales como la fibra de carbono, destacada por su resistencia y proyección de sonido superior, ejemplificando la evolución constante en la luthería del timple.

### *Afinación*

Como se ha comentado en apartados anteriores, desde sus orígenes el timple se ha consolidado como un instrumento de 5 cuerdas. Generalmente se numeran las cuerdas de manera ascendente, colocando el timple en posición de tocar, siendo la llamada 1ª cuerda la inferior (véase figura 3). Se considerará esta numeración para el resto del trabajo.



**Figura 3.** Numeraciones de las cuerdas del timple. Figura de elaboración propia.

La afinación tradicional y más extendida, de 1ª a 5ª cuerda es “RE, LA, MI, DO, SOL” (véase Figura 4), tal y como se muestra en la figura. “En la que el MI es más grave que el DO y el SOL” (Izquierdo, 2018).



**Figura 4.** Afinación de las cuerdas del timple. Figura de elaboración propia.

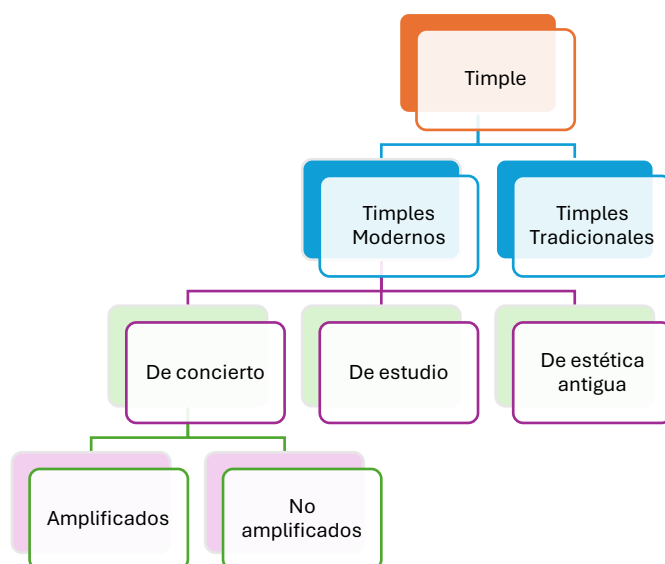
Hoy en día hay agrupaciones que prueban nuevas disposiciones y varían la afinación establecida o que “octavan”<sup>2</sup> alguna de sus cuerdas. También, y gracias al desarrollo de los luthieres, se están fabricando variaciones del timple que le permiten desarrollar notas en registros más graves o con diferente timbre. Tal es el caso del Timple - Bajo creado por David Sánchez.

En el transcurso de este trabajo se tendrá en cuenta el timple en su afinación general.

### *Tipos de timple*

Como todo instrumento con historia, el timple ha sufrido cambios y evoluciones, debido a esto se siguen encontrando modelos muy variados entre sí. Se han introducido cambios en cuanto a tamaño, número de trastes y apariencia con el objetivo de facilitar su construcción, musicalidad y mejorar su sonido.

Siguiendo la clasificación propuesta por González Casal (2017) el timple se puede clasificar principalmente en dos tipos: timples tradicionales o timples modernos. Y los modernos a su vez pueden clasificarse en timples de concierto, de estudio o de estética antigua (véase Figura 5).



**Figura 5.** Esquema de catalogación del timple.

Figura de elaboración propia, basado en: (González Casal, 2017)

En este caso, para el desarrollo práctico del trabajo se utilizarán 2 timples modernos de concierto, uno amplificado del luthier David Sánchez y uno sin amplificar del luthier Abraham Marrero (véase figura 6).

---

<sup>2</sup> Bajar la afinación de una cuerda una octava



**Figura 6.** *Timples utilizados para el desarrollo práctico del trabajo en su funda.*

*(izq) Abraham Marrero (dcha.) David Sánchez. Figura de elaboración propia.*

### **2.1.3. El sonido del timple**

En palabras del luthier Alberto Cárdenas: “El sonido del timple es un sonido tradicionalmente agudo y “chillón” o estridente, para su uso en agrupaciones folclóricas. Sin embargo los nuevos timplistas solistas buscan otro sonido, más “dulce” y equilibrado, con más cuerpo” (Cárdenas, 2022). En el análisis acústico realizado por González Casal (2017) se observa como en su respuesta en frecuencia se caracteriza precisamente por ese sonido brillante.

Con la configuración más normal en la actualidad los timples tienen aproximadamente hasta 19 trastes. Teniendo en cuenta la afinación general, su tesitura abarcaría un total de 29 semitonos (véase figura 7), pudiendo tocar las notas desde el MI3 al LA5.



**Figura 7.** *Tesitura del timple en afinación general. Figura de elaboración propia.*

## 2.2. Instrumentos virtuales

Los instrumentos virtuales, también conocidos como instrumentos de software, son programas de computadora diseñados para emular las características y sonidos de instrumentos musicales reales o para crear sonidos nuevos que no pueden ser producidos por instrumentos físicos. “El término “instrumentos virtuales” se aplica a cualquier software que actúa como un instrumento musical” (Collins, 2003). Estos instrumentos permiten a los músicos y productores utilizar una amplia gama de sonidos en sus composiciones sin necesidad de poseer los instrumentos físicos correspondientes.

El desarrollo de los instrumentos virtuales está estrechamente ligado al avance de la tecnología digital y la informática musical. A partir de los años 80 y 90, con la aparición de las computadoras personales y el software de audio digital, se empezaron a desarrollar los primeros instrumentos virtuales que podían ser utilizados en estudios de grabación y producciones musicales.

Estas herramientas son ampliamente utilizadas en la producción musical contemporánea, tanto en estudios profesionales como en configuraciones de grabación amateurs. Se aplican a actividades de:

- **Composición y Arreglo:** Permiten a los compositores experimentar con diferentes sonidos y arreglos sin necesidad de tener o saber tocar instrumentos musicales de manera física.
- **Producción y Grabación:** Facilitan la incorporación de sonidos de alta calidad en producciones musicales, reduciendo los costos y la necesidad de equipos adicionales.
- **Interpretación en Vivo:** Muchos músicos utilizan controladores MIDI junto con instrumentos virtuales para tocar en vivo, aprovechando la versatilidad y portabilidad de estos sistemas.
- **Educación Musical:** Proveen una herramienta accesible para la enseñanza de la música, permitiendo a los estudiantes explorar una amplia gama de sonidos e instrumentos.

Como se detalla en (Yun & Cha, 2013) los instrumentos virtuales ofrecen una calidad de audio excelente y una buena estabilidad. La variedad de parámetros automatizables que pueden aplicarse en tiempo real sea (filtros, envolventes, afinación, etc.) hacen que “En cierta manera el software de hoy en día sea superior a los dispositivos que quisieron emular”

### 2.2.1. Virtual Studio Technology (VST)

Como se explica en VST (*Virtual Studio Technology*), por la EUIT de Telecomunicación(2012), *Virtual Studio Technology* (VST) es una tecnología desarrollada por Steinberg que permite a los usuarios de estaciones de trabajo de audio digital (DAWs) integrar y utilizar una amplia variedad de *plugins* de efectos e instrumentos virtuales dentro de su entorno de producción musical. Dicha interfaz fue introducida en 1996, y ha conseguido revolucionar la producción musical al ofrecer una forma estándar y flexible la incorporación nuevos sonidos y procesadores de audio en un estudio digital.

Dentro de la tecnología VST, se diferencian varios tipos de *plugins*.

Los VSTi (*VST Instruments*) son *plugins* que emulan instrumentos musicales, ya sean acústicos, electrónicos o híbridos. Pueden realizar imitaciones desde pianos y guitarras hasta sintetizadores y baterías. Estos son capaces de generar sonidos a partir de muestras



o mediante síntesis digital. Los usuarios pueden manipular parámetros como el tono, el timbre, la dinámica y otros aspectos para crear y modificar sonidos. Algunos ejemplos de ellos son: Synth1 (sintetizador), Kontakt (*sampler* y sintetizador), Massive (sintetizador avanzado).

Los Efectos de Audio (VST) diseñados para procesar el audio en tiempo real. Esto incluye efectos como reverberación, *delay*, ecualización, compresión y distorsión. Los efectos pueden aplicarse a pistas individuales, a grupos de pistas o a la mezcla completa, permitiendo a los usuarios modelar y mejorar el sonido en diversas formas. Algunos ejemplos que podemos encontrar son; FabFilter Pro-Q (ecualizador), Waves Reverb (reverberación), Soundtoys Decapitator (saturación).

Cada *plugin* VST incluye su propia interfaz gráfica, que puede variar desde simples controles deslizantes hasta complejas representaciones visuales de circuitos y módulos. Los usuarios pueden interactuar con estos controles para ajustar los parámetros del *plugin*, lo que permite una gran flexibilidad en la creación de sonidos y el procesamiento de audio.

Para tratar los audios, estos deben integrarse en las pistas de un DAW como efectos o instrumentos. Los DAWs reconocen los archivos de *plugins* VST y los integran en su entorno de trabajo, permitiendo la automatización de parámetros y el uso de ajustes predefinidos.

Una ventaja de VST es que está habilitada para la mayoría de sistemas operativos, sin embargo, “siempre debe estar compilado para la plataforma donde se vaya a ejecutar, ya que un VST compilado para Mac OS no funcionará en Windows y viceversa” (EUIT de Telecomunicación, 2012).

El uso de VST ofrece numerosas ventajas. Proporciona flexibilidad, permitiendo a los usuarios expandir las capacidades de su DAW con nuevos efectos e instrumentos sin necesidad de hardware adicional. Los *plugins* VST son más económicos que los equipos de hardware tradicionales, haciendo la producción musical más accesible. Además, la tecnología VST está en constante evolución, facilitando la experimentación con nuevos sonidos y técnicas de procesamiento, lo que fomenta la creatividad. Los desarrolladores de *plugins* pueden lanzar actualizaciones periódicas para mejorar sus productos y agregar nuevas características.

Las versiones actuales de VST incluyen VST 2, que ya no se desarrolla activamente, y VST 3, que ofrece mejoras significativas en el manejo de memoria y procesamiento de audio en tiempo real.

### **2.2.2. Musical Instrument Digital Interface (MIDI)**

MIDI (*Musical Instrument Digital Interface* o Interfaz Digital de Instrumentos Musicales) “es la especificación para un conjunto de códigos digitales destinados a transmitir información de control y sincronización musical en tiempo real, así como para la interfaz de hardware a través de la cual se transmiten estos códigos” (Moog, 1986).

Esta tecnología ha sido un pilar fundamental en la evolución de la música digital, especialmente en la creación y utilización de instrumentos virtuales. Como se especifica en el documento de *MIDI: Musical Instrument Digital Interface* (Moog, 1986), las ideas básicas para un estándar de interconexión digital de instrumentos musicales (originalmente llamado Interfaz Universal de Sintetizadores) fueron propuestas por Dave Smith y Chet Wood en la 70ª Convención de la Sociedad de Ingeniería de Audio en noviembre de 1981. Para finales de 1982, el nombre cambió a Interfaz Digital de

Instrumentos Musicales, y varios fabricantes de instrumentos musicales electrónicos acordaron una especificación técnica integral. El primer enlace documentado de MIDI entre instrumentos de dos fabricantes diferentes se realizó en enero de 1983

Desde su desarrollo en 1983, MIDI ha permitido que los músicos y productores conecten y controlen diferentes dispositivos electrónicos, facilitando la comunicación entre sintetizadores, ordenadores, secuenciadores y otros equipos musicales. La versatilidad de MIDI no solo se limita a instrumentos físicos, sino que también se extiende a los instrumentos virtuales, conocidos como VST (Virtual Studio Technology) y *plugins*.

Los instrumentos virtuales son software que emulan el sonido de instrumentos acústicos o electrónicos, y gracias a MIDI, estos pueden ser controlados con alta precisión. MIDI transmite mensajes que indican qué nota se debe tocar, la intensidad (velocidad) de la nota, cuánto debe durar y otros parámetros de control como el cambio de tono y la modulación.

Esta información puede ser enviada desde un teclado MIDI, un controlador MIDI, o incluso secuenciada directamente en un programa de producción musical (DAW, Digital Audio Workstation).

La integración de MIDI con instrumentos virtuales ha democratizado la producción musical, permitiendo que cualquier persona con un ordenador y un controlador MIDI tenga acceso a una amplia gama de sonidos e instrumentos que de otro modo serían costosos o difíciles de obtener.

The significance of MIDI is that it permits a wide variety of equipment from many different manufacturers to work together in a single system. Each piece of equipment is adhering to the same rules of information management specified by MIDI so that they have a common system of communication (Rothstein, 1995).

Los mensajes MIDI son la base del protocolo para la comunicación entre dispositivos musicales electrónicos. Estos mensajes permiten la interpretación y control de instrumentos virtuales o hardware. Los principales tipos de mensajes MIDI son “NOTE ON” y “NOTE OFF”. A continuación, se detalla el contenido de dichos mensajes siguiendo lo explicado en de Oliveira & de Oliveira (2017) puesto que se utilizarán para el desarrollo práctico del instrumento.

“NOTE ON”: Se envía cuando el intérprete presiona una tecla en un teclado (físico o virtual) musical. Este mensaje incluye parámetros que especifican el tono (*pitch*) y la velocidad (*velocity*) de la nota, que corresponde a la intensidad con la que se toca la tecla. Al recibir este mensaje, el sintetizador empieza a reproducir la nota con el tono y la intensidad especificados.

“NOTE OFF”: Se envía cuando la tecla se suelta, indicando al sintetizador que deje de reproducir la nota correspondiente.

Cada mensaje MIDI se compone de uno o más *bytes*. El primer *byte* es el *byte* de comando (*command byte*), seguido de uno o dos *bytes* de datos (*data bytes*) dependiendo del tipo de mensaje.

El *byte* de comando para “NOTE ON” se estructura como 1001cccc. El *byte* de comando para “NOTE OFF” se estructura como 1000cccc. En ambos casos cccc se sustituye por el número de canal MIDI (de 0 a 15).

El primer *byte* de datos (*data byte 1*) especifica el tono de la nota (0ppppppp), donde el valor de tono varía de 0 a 127 siendo DO4 el valor 60 (se pueden consultar en (InspiredAcoustics, s. f.)) y siendo cada unidad un semitono. El segundo *byte* de datos (*data byte 2*) especifica la velocidad (intensidad) de la nota (0vvvvvvv), con un rango de 0 a 127.

Con el objetivo de definir la forma en que se crean los mensajes se expone a continuación un ejemplo en que el cual se toca una nota LA4 a una velocidad de 90. La nota LA4 coincide con el valor de tono 69, y 90 en hexadecimal equivale a 5A por lo que, suponiendo que no hay ningún canal ocupado, el comando “NOTE ON” sería: 0x90 0x45 0x5A.

Algunas de las ventajas que ofrece MIDI es una mejora en la eficiencia, flexibilidad, por lo que permite cambiar sonidos y ajustes sin regrabar y la interoperabilidad que asegura la compatibilidad entre distintos dispositivos

En el escenario, los intérpretes de instrumentos electrónicos utilizan MIDI para conectar sus sintetizadores, secuenciadores y cajas de ritmos. En el estudio, las computadoras personales equipadas con software sofisticado basado en MIDI están transformando la forma en que se compone música electrónicamente. MIDI también ha encontrado uso en iglesias, salas de estar y estudios caseros, permitiendo a músicos profesionales y aficionados producir música de alta calidad con equipos modestamente preciosos y disponibles fácilmente (Moog, 1986).

### 2.3. Samplers y Sintetizadores de Sonido

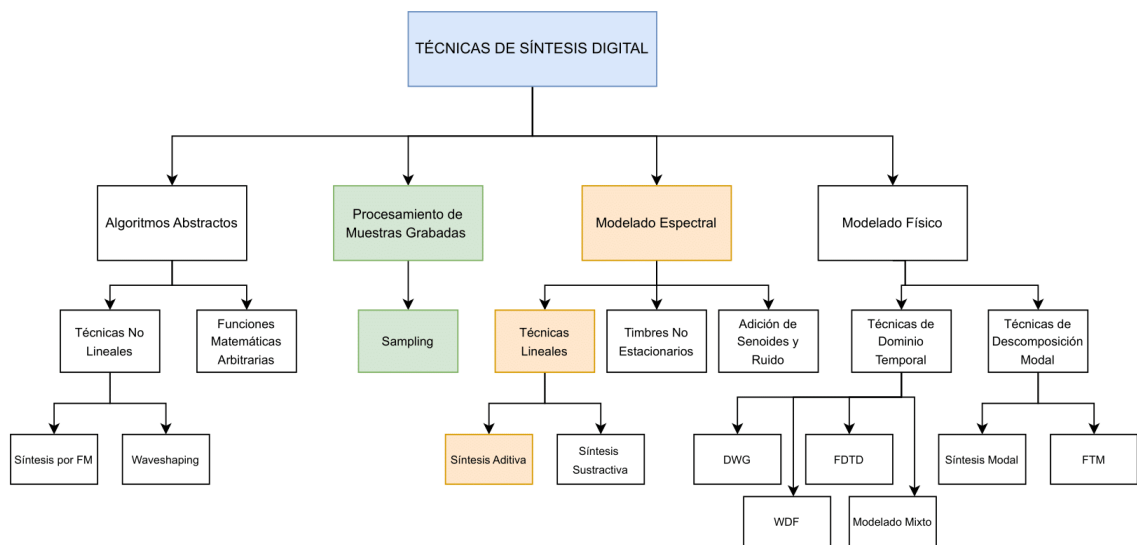
Según el Gran Diccionario Enciclopédico Ilustrado de la Reader's Digest Association, la síntesis es la "composición de un todo por la reunión de sus partes" (Reader's Digest Association, 1981, p. 3530). En un análisis más detallado de su definición, Russ (1996) añade:

El proceso de síntesis es, por lo tanto, una acción de juntar, y la "composición de un todo" es significativa porque implica algo más que un simple ensamblaje aleatorio (...). Aunque un sintetizador puede ser capaz de producir variedades casi infinitas de salida, controlarlas y elegirirlas requiere intervención y habilidad humanas (p. 3).

Así mismo existen, varios tipos de síntesis en los que los más comunes son la química y la síntesis de sonido. En el caso que atañe a este trabajo, se centrará en la síntesis de sonido:

Sound synthesis is the process of producing sound. It can reuse existing sounds by processing them, or it can generate sound electronically or mechanically. It may use mathematics, physics or even biology; and it brings together art and science in a mix of musical skill and technical expertise (Russ, 1996, p. 4).

En cuanto a los esquemas de síntesis de sonido, se clasifican según la técnica empleada. La figura 8 resume los métodos más utilizados en la síntesis digital de sonido.



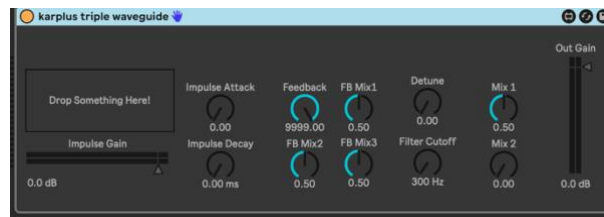
**Figura 8.** *Técnicas de Síntesis Digital.*

Figura de elaboración propia, basado en Villacorta & Marchiaro (2006)

Además de la clasificación de las técnicas utilizadas para lograrlos, el sonido sintetizado se puede dividir en imitativo o sintético, siendo el imitativo trata de lograr un sonido que se identifique como un sonido ya existente proveniente por ejemplo de un instrumento musical.

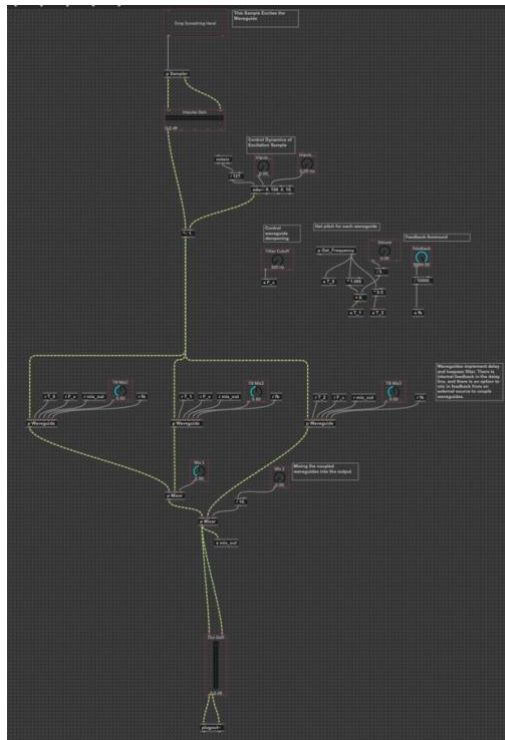
Dentro de la síntesis se consideran sintetizadores a aquellos aparatos que "pueden sintetizar" o "hacer síntesis" (Reader's Digest Association, 1981, p. 3530). En concreto,

los sintetizadores de sonido son aquellos usados para crear y procesar sonidos y música. Estos se componen principalmente de dos bloques: la interfaz de control, a través de la que se controlan los parámetros que definen el sonido final; y el motor de síntesis, que interpreta dichos parámetros para producir el sonido.



**Figura 9.** Ejemplo de interfaz de control.

*Nota:* Captura de pantalla que muestra la interfaz de control del sintetizador karplus triple waveguide desarrollado por John Janiczek. Elaboración propia.



**Figura 10.** Ejemplo de motor de síntesis.

*Nota:* Captura de pantalla que muestra el motor de síntesis del sintetizador Karplus Triple Waveguide desarrollado por John Janiczek. Elaboración propia.

En la mayoría de los casos existe un grado de abstracción entre la interfaz de control y el propio motor de síntesis. Esto se debe a que la complejidad del proceso de síntesis suele ser muy alta, y a menudo es necesario reducir la complejidad aparente del control mediante el uso de algún tipo de modelo conceptual más simple. Esto permite al usuario del sintetizador utilizarlo sin necesidad de un conocimiento detallado del funcionamiento interno. En las las 9 y 10 respectivamente se puede observar la interfaz de control y el motor de síntesis del sintetizador *Karplus Triple Waveguide* desarrollado por John

Janiczek en 2024. Se trata de un sintetizador basado en el modelo de Karplus-Strong<sup>3</sup> que utiliza una triple guía de onda para conseguir el sonido de guitarra. Como se observa está desarrollado en Max 4 Live<sup>4</sup>.

En este proyecto, se propone la creación de un instrumento virtual utilizando dos técnicas de síntesis digital distintas para su desarrollo, marcadas en verde y naranja respectivamente en la *Figura 8*.

En primer lugar, se empleará la técnica de procesamiento de muestras grabadas, conocida como *sampling*, debido a su amplia utilización en la industria musical y su capacidad para producir resultados que se asemejan mucho a la realidad. El *sampling* para instrumentos virtuales imitativos implica la grabación de notas individuales de un instrumento real y su posterior reproducción en el instrumento virtual, lo que permite captar las características tonales y dinámicas del instrumento original.

En segundo lugar, con el objetivo de explorar la posibilidad de lograr un mayor realismo en la relación entre notas, se propone utilizar la síntesis aditiva. Esta técnica se enmarca en las técnicas lineales del modelado espectral. La síntesis aditiva construye el sonido sumando múltiples ondas sinusoidales, lo que permite un control detallado sobre los armónicos y la estructura del sonido para su recreación.

El uso de estas dos técnicas permitirá evaluar sus respectivas ventajas y desventajas en la creación de un instrumento virtual, para determinar cuál de ellas ofrece mejores resultados en términos de realismo y fidelidad sonora.

### 2.3.1. Procesamiento de muestras grabadas (*sampling*)

Corresponden a esta categoría los llamados *samplers*, “cuyo principio de funcionamiento consiste en grabar, procesar y reproducir sonidos” (Russ, 1996). Aunque estrictamente hablando esta técnica no crea sonidos desde cero, la misma ofrece una variedad infinita de posibilidades, al grabar digitalmente sonidos reales o sintéticos, filtrarlos o editarlos, y combinarlos con otras señales, hasta el punto de no distinguir el sonido original.



**Figura 11.** Diagrama de flujos de Sampling.

Figura de elaboración Propia

A continuación, se detallan los pasos fundamentales en el proceso de creación y utilización de un *sampler* que se pueden ver de forma esquemática en el diagrama de flujos de la figura 11.

---

<sup>3</sup>

<sup>4</sup> Max 4 Live es una interfaz de programación por nodos del DAW Ableton Live para blabla

### 1. Grabación de Muestras

En esta etapa, es como presupuesto proceso implica la captura de cada nota del instrumento a diferentes intensidades (velocidades de golpeo o dinámica) y técnicas de ejecución (por ejemplo, *pizzicato*, *legato*). Es crucial utilizar equipos de grabación de alta calidad y un entorno controlado acústicamente para garantizar la fidelidad de las muestras.

### 2. Digitalización y Edición

Una vez grabadas, las muestras son digitalizadas y editadas. La digitalización convierte las señales analógicas en datos digitales mediante un convertidor analógico-digital (ADC). Durante la edición, se eliminan ruidos no deseados, ajustándose los puntos de inicio y fin de cada muestra, y aplican técnicas de normalización para mantener una coherencia de volumen entre todas las muestras.

### 3. Procesamiento y Efectos

Los *samplers* avanzados permiten aplicar efectos de procesamiento adicionales, como *reverb*, *delay*, *chorus* y *EQ*, para mejorar y personalizar el sonido. Estos efectos se pueden ajustar para replicar las características acústicas del entorno original o crear nuevas texturas sonoras.

### 4. Mapeo de Muestras

Se define el mapeo como el proceso de asignar cada muestra grabada a una tecla o rango de teclas específico en el teclado del *sampler*. Esto permite que, al pulsar una tecla en el teclado controlador, se reproduzca la muestra correspondiente. El mapeo también incluye la asignación de diferentes capas de velocidad, permitiendo variaciones dinámicas al tocar con diferentes intensidades.

### 5. Implementación en Software

Finalmente, las muestras y sus configuraciones se integran en un formato compatible con el software de producción musical (DAWs) y los *samplers* virtuales. Formatos comunes incluyen “WAV” para las muestras y archivos de configuración específicos para *samplers* como Kontakt, EXS24, y SFZ. Estos archivos de configuración contienen toda la información sobre el mapeo de las muestras, las envolventes, los filtros y los efectos aplicados.

## 2.3.2. Modelado espectral: síntesis aditiva

El modelado espectral, tal y como indica su nombre, “se basa en la información espectral del sonido (...) un sonido se analiza y describe en términos de sus componentes espectrales” (O’Leary & Griffith, 2003). Este tipo de técnicas se basan en la teoría de Fourier, desarrollada por el matemático francés Jean-Baptiste Fourier a principios del siglo XIX. En 1807 Fourier demostró que cualquier forma de onda periódica puede ser representada como la suma de una serie de ondas sinusoidales simples, cada una con su propia frecuencia, amplitud y fase.

En este modelado el primer paso es analizar el sonido a modelar, estas señales pueden ser examinadas e interpretadas utilizando diferentes tipos de técnicas. “Una de las herramientas más comunes y usadas es la Transformada de Fourier (TF) la cual pasa una señal del dominio temporal y llevándola al dominio frecuencial, lo que permite analizar las diferentes frecuencias presentes dentro de la señal” (Giraldo C. & Quintero M., 2014).

Para analizar el contenido espectral de un sonido, es común utilizar la Transformada Rápida de Fourier (FFT, por *Fast Fourier Transform*). La FFT es un algoritmo eficiente para calcular la Transformada Discreta de Fourier (DFT, por *Discret Fourier Transform*) y su inversa, permitiendo descomponer una señal en sus componentes de frecuencia. La DFT de una secuencia  $x[n]$  de  $N$  puntos está dada por:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, k = 0, 1, \dots, N - 1 \quad \dots (1)$$

donde  $X[k]$  representa la amplitud de la componente sinusoidal en la frecuencia  $k$ . La FFT mejora la eficiencia de este cálculo, lo que es importante para aplicaciones en tiempo real y análisis detallados de señales complejas.

### *Síntesis aditiva*

La síntesis aditiva es una técnica de síntesis de sonido basada en esta premisa de que cualquier sonido puede ser creado sumando una serie de ondas sinusoidales de diferentes frecuencias. Estas frecuencias pueden ser múltiplos enteros de una frecuencia fundamental (armónicos) o pueden ser inarmónicas. En términos de síntesis aditiva, son conocidas por el término “armónicos” las frecuencias formantes pese a que sean inarmónicas.

Las ondas sinusoidales se suman para formar una forma de onda compleja. La complejidad de la forma de onda final depende del número y la naturaleza de las ondas sinusoidales que se suman. Por ejemplo, una onda sinusoidal pura tiene solo un componente de frecuencia, mientras que una onda cuadrada puede descomponerse en una serie de sinusoidales con frecuencias impares.

Para cada armónico en un sintetizador aditivo, típicamente se controlan varios parámetros:

- Frecuencia ( $f_k$ ): Puede ser fija (armónica) o variable (inarmónica).
- Fase ( $\Phi_k$ ): La posición inicial de la onda sinusoidal.
- Amplitud ( $A_k$ ): El volumen de la onda sinusoidal.
- Envolvente: Define cómo cambia la amplitud con el tiempo y puede ser una envolvente simple (como una envolvente ADSR) o una envolvente segmentada más compleja.

La fórmula general para una onda sintetizada aditivamente es:

$$x(t) = \sum_{k=1}^N A_k(t) \cdot \sin(2\pi f_k t + \Phi_k) \quad \dots (2)$$

donde:

- $N$  es el número de armónicos.
- $A_k(t)$  es la amplitud del  $k^o$  armónico en el instante de tiempo  $t$ .



- $f_k$  es la frecuencia del  $k^{\circ}$  armónico.
- $\Phi_k$  es la fase del  $k^{\circ}$  armónico.

### Envolventes

En la práctica, las amplitudes y frecuencias de los armónicos pueden cambiar con el tiempo. Para representar estos cambios, se utilizan envolventes de amplitud y frecuencia que varían de forma continua. En síntesis, aditiva avanzada, es común usar envolventes de frecuencia para hacer que los armónicos no sean simplemente múltiplos enteros de la frecuencia fundamental, sino que varíen de manera controlada a lo largo del tiempo. Esto permite la creación de sonidos más realistas y dinámicos.

Las envolventes de amplitud son utilizadas para modelar el comportamiento dinámico de cada armónico en el tiempo, mientras que las de frecuencia son utilizadas para modelar variaciones en la afinación o para crear efectos de vibrato, portamento o glissando. En el caso de este proyecto esta segunda aplicación no nos compete por lo que nos centraremos en las envolventes de amplitud. Lo que si interesa conseguir es simular la envolvente de amplitud para lograr un comportamiento dinámico de la señal más realista.

Una técnica común para simplificar la representación y manipulación de los datos es aproximar estas envolventes con segmentos lineales. Una envolvente de amplitud lineal por tramos, también conocida como envolvente segmentada linealmente, es un tipo de envolvente de amplitud que se compone de segmentos de línea recta conectados entre sí. Cada segmento puede tener una pendiente (cambio de amplitud) diferente, y los puntos de conexión entre los segmentos se denominan nodos. Las envolventes de amplitud lineales por tramos  $A'_k(t)$  son aproximaciones a las amplitudes  $A_k(t)$  originales.

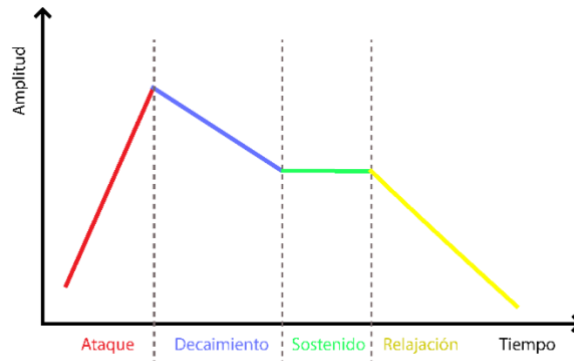
Utilizando estas envolventes, la fórmula de síntesis aditiva se modifica para incluir la integración de la frecuencia en función del tiempo:

$$x(t) = \sum_{k=1}^N A'_k(t) \cdot \sin \left( 2\pi \int_0^t f'_k(u) du + \Phi_k \right) \quad \dots (3)$$

donde:

- $A'_k(t)$  representa la envolvente de amplitud lineal por tramos del  $k^{\circ}$  armónico.

Un ejemplo clásico de una envolvente de amplitud lineal por tramos es la envolvente ADSR (*Attack, Decay, Sustain, Release* o Ataque, Decaimiento, Sostenido, Relajación), ampliamente utilizada en síntesis de sonido.



**Figura 12.** Esquema gráfico de curva ADSR.

*Figura de elaboración propia.*

La envolvente se divide en 4 tramos lineales de diferente pendiente:

- **Ataque:** La fase durante la cual la amplitud de la señal aumenta desde cero hasta su valor máximo.
- **Decaimiento:** La fase durante la cual la amplitud disminuye desde el valor máximo hasta el nivel de sustain.
- **Sostenido:** La fase durante la cual la amplitud se mantiene constante mientras se sostiene la nota.
- **Relajación:** La fase durante la cual la amplitud disminuye desde el nivel de sustain hasta cero después de que se suelta la nota.

### *Transformada de Hilbert*

La transformada de Hilbert es una herramienta matemática utilizada para obtener la señal analítica de una señal real y de esta manera extraer la envolvente de amplitud de esta. La señal analítica es una señal compleja cuya parte real es la señal original y cuya parte imaginaria es la transformada de Hilbert de la señal original.

Para una señal  $x(t)$ , la transformada de Hilbert, denotada como  $\mathcal{H}\{x(t)\}$ , se define como:

$$\mathcal{H}\{x(t)\} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(u)}{t-u} du \quad \dots (4)$$

La señal analítica  $z(t)$  de  $x(t)$  se define como:

$$z(t) = x(t) + j\mathcal{H}\{x(t)\} \quad \dots (5)$$

donde  $j$  es la unidad imaginaria.

Por lo tanto, se podría escribir la señal analítica  $z(t)$  en forma polar como:

$$z(t) = A(t)e^{j\Phi(t)} \quad \dots (6)$$

donde:

$A(t) = |z(t)| = \sqrt{x(t)^2 + j\mathcal{H}\{x(t)\}}$  es la envolvente de amplitud

$\Phi(t) = \arg(z(t))$  es la fase instantánea

Los gráficos de espectro se utilizan comúnmente para mostrar el contenido armónico de las formas de onda sintetizadas. Un espectro muestra la relación entre las frecuencias y sus amplitudes relativas en una señal de audio.

La síntesis aditiva es especialmente potente para recrear timbres complejos y evolucionar sonidos a lo largo del tiempo. Sin embargo, la complejidad de controlar un gran número de armónicos simultáneamente puede hacer que los sintetizadores aditivos sean difíciles de manejar y programar. Esto ha llevado a que la mayoría de los sintetizadores aditivos sean implementaciones digitales, donde las herramientas computacionales pueden gestionar mejor la generación y el control de los numerosos parámetros necesarios.

### 3. Desarrollo práctico

#### 3.1. Herramientas utilizadas

Las herramientas escogidas para la realización práctica del proyecto han sido elegidas en base a la disponibilidad y la calidad de resultados que pueden ofrecer.

##### 3.1.1. Instrumentos musicales

Para el desarrollo de la síntesis digital a través de ambas técnicas se necesita de muestras de sonido del instrumento. Es por ello por lo que se ha decidido muestrear dos timple. De esta manera, se consiguen dos resultados y así poder compararlos. El primer timple es un timple del luthier Abraham Gutiérrez del año 2021. Es un timple de aros de moral, con fondo de arce y ojo de perdiz con un sonido más dulce de lo normal en este instrumento. Y el segundo un timple electrificado del luthier David Sánchez del año 2012 con un sonido brillante y más “chillón”.



**Figura 13.** Timples utilizados para la síntesis por procesado de muestras en su funda.

*Nota:* (izq.) Abraham Gutiérrez, (dcha.) David Sánchez. Elaboración propia.

##### 3.1.2. Equipos y materiales de grabación

Para la grabación de las muestras se utilizó el estudio de doblaje de la Escuela Politécnica Superior de Gandía (EPSG), el cual tiene una calidad de aislamiento y acondicionamiento acústico favorable para conseguir una representación fiel del sonido.



**Figura 14.** Vista del estudio de doblaje de la EPSG, sala de control y cabina desde la cabina de grabación.

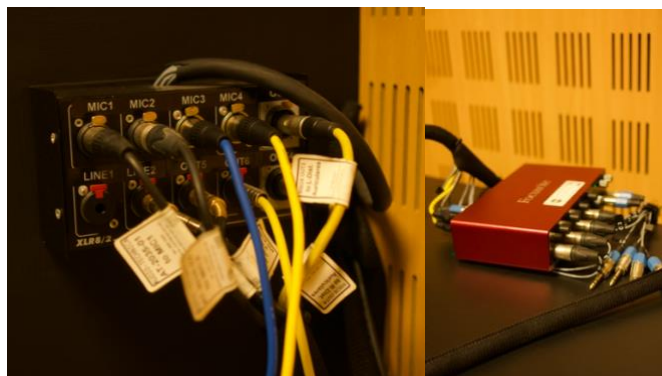
Figura de elaboración Propia.

A continuación, se detalla el listado de material utilizado.

**Tabla 1.** *Desglose de material utilizado para toma de muestras.*

	UNIDADES	MARCA	MODELO
<b>Micrófono de condensador de diafragma grande (Multipatrón)</b>	1	The t-Bone	Retro Tube II
<b>Micrófono dinámico de diafragma mediano</b>	1	Shure	SM57
<b>Micrófono de condensador de diafragma pequeño</b>	2 (Par Estéreo)	The t-bone	SC 140 Stereo set
<b>Interfaz de audio</b>	1	Focusrite	Scarlett 18i8 3rd Gen
<b>Controlador de faders MIDI</b>	1	MACKIE	MCU Pro
<b>Ordenador</b>	1	Apple	iMac

Para el conexionado de los equipos se utilizaron cables XLR-3 de la marca *the T-bone*. El enrutado de las señales entre la cabina y la sala de control de grabación fue realizado a través de una caja de conexiones instalada en dicho estudio para tal función. Estas señales se conectaron directamente a una interfaz de audio *Focusrite Scarlett 18I8* donde se convierten en señales digitales. Finalmente, las señales de cada una de las entradas se encaminan a través de la interfaz hasta el ordenador. Para una mayor comodidad en el control de los *faders* controles de ganancias de entrada y de efectos se utiliza un controlador MIDI *MCU Pro*, instalado en el estudio para ello.

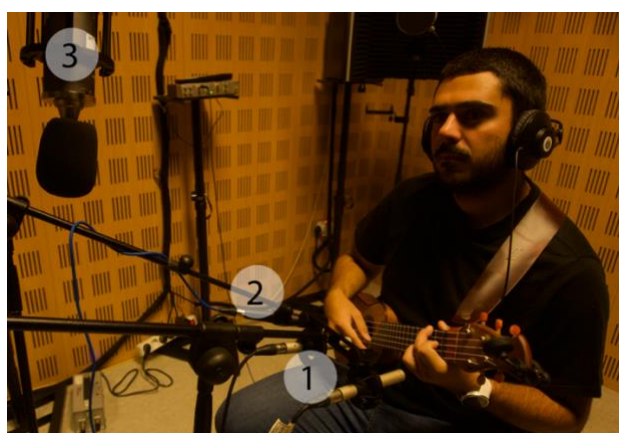


**Figura 15.** Enrutado de señales de audio mediante caja de conexiones (izq.)  
a la interfaz de audio digital (dcha.). Figura de elaboración propia.

Dado que el timple es un instrumento de carácter folclórico y local, no existen referencias sobre el microfoneado para estudio de este instrumento. Por lo que para la realización del presente trabajo, se ha tenido que idear y poner en práctica una configuración propia. Para ello se ha revisado la bibliografía pertinente sobre instrumentos de tesitura, tamaño y timbre similares.

Principalmente se han seguido las indicaciones del libro *The Recording Engineers Handbook* (Owsinski, 2007), ejemplar de referencia para la grabación de estudios de instrumentos. Teniendo en cuenta las indicaciones para instrumentos de cuerda como la guitarra de cuerdas de nylon, se ha tomado la decisión de grabar utilizando un conjunto de cuatro micrófonos, de manera que se recoja todo el espectro y matices en los que actúa el instrumento.

Para la captura principal del sonido del timple se utiliza un par estéreo en configuración A-B compacta, con una distancia de unos 20 cm entre ellos, apuntado hacia la zona de la boca por el lado izquierdo y a los trastes por el lado derecho (véase figura 16 (1)). Al tratarse de un instrumento pequeño se ha modificado el par estéreo a esta configuración ya que con la configuración *ORTF* sugerida por Owsinski para la guitarra de nylon no se utiliza un ángulo en que se logre un sonido envolvente realista.

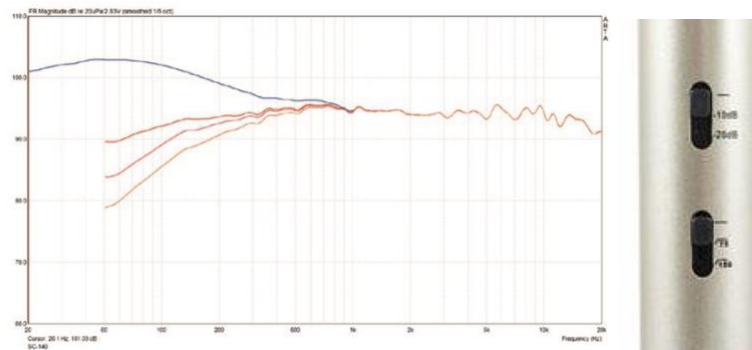


**Figura 16.** Disposición de micrófonos en sala de grabación.

*Nota:* donde (1) es el par estéreo en configuración A-B compacta, (2) es el micrófono dinámico de diafragma mediano y (3) es el micrófono multipatrón de diafragma grande.

Figura de elaboración propia.

Para ello se ha escogido el par estéreo *SC 140 Stereo set* de *the t-bone*. Este micrófono fue seleccionado por su respuesta en frecuencia bastante plana (véase figura 17 izq.) que permite captar el sonido de manera fidedigna. Para lograr esto se tuvo que asegurar que el filtro paso bajo incorporado en el micrófono permaneciera apagado (véase figura 17 parte derecha de la figura).



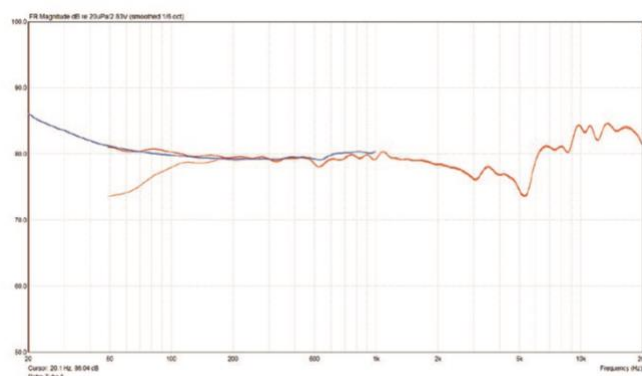
**Figura 17.** Respuesta en frecuencia del micrófono *SC 140* de *the t-bone* (izq.) y la configuración de interruptores escogida (dcha.).

*Nota:* Extraído de hoja de especificaciones oficial disponible en: [https://www.thomann.ae/the\\_tbone\\_sc140\\_stereoset.htm](https://www.thomann.ae/the_tbone_sc140_stereoset.htm)

Para completar y enriquecer este sonido, se ubica un segundo micrófono, en este caso dinámico, de diafragma mediano en la parte inferior derecha para “recoger las frecuencias más graves que resuenan en la zona de la caja de resonancia” (véase figura 16 (2) ). Para esta función se ha escogido el micrófono *shure sm57*, un micrófono dinámico muy popular en grabaciones de estudio.

Y, por último, para agregar el sonido del ambiente que “genera el instrumento en la sala” se ha escogido un micrófono de cardioide de diafragma grande y patrón de directividad variable colocado a una altura de “8 a 18 pulgadas”, lo que serían de 20 a 45 cm aproximadamente, por encima del instrumento (véase figura 16 (3) ).

En este caso el micrófono utilizado fue el *Retro Tube II* en su patrón de directividad omnidireccional para captar el ambiente en su totalidad. Este patrón nos ofrece una respuesta en frecuencia bastante plana para las frecuencias bajas y medias y acentúa ligeramente las altas por lo que es ideal para la función prevista.



**Figura 18.** Respuesta en frecuencia del micrófono *Retro Tube II* en su patrón de directividad omnidireccional.

*Nota:* Extraído de hoja de especificaciones oficial disponible en: [https://www.thomann.ae/the\\_tbone\\_sc140\\_stereoset.htm](https://www.thomann.ae/the_tbone_sc140_stereoset.htm)

### 3.1.3. Software utilizado

A lo largo de todo el desarrollo fue necesaria la utilización de diversos programas informáticos.

**Tabla 2.** Listado de software utilizado en la realización práctica del proyecto.

	<i>Sampler</i>	<i>Síntesis</i>
<i>Logic</i>	X	X
<i>Maize</i>	X	
<i>Matlab</i>		X

En primer lugar, para la captura y procesamiento de las muestras de audio se utilizó *Logic Pro X*, un software profesional de producción musical desarrollado por *Apple*. Esta herramienta fue seleccionada por su calidad y precisión en la grabación de audio, así como por su amplio conjunto de herramientas de edición y procesamiento. Este software permitió realizar grabaciones de alta fidelidad del timple, eliminar ruidos no deseados y normalizar las muestras para asegurar una calidad consistente. La elección de *Logic Pro X* se fundamenta también en la familiaridad con su interfaz lo que ahorró tiempo en el tratamiento del audio.

Para la creación del banco de sonidos MIDI, se utilizó *Maize Sampler*, una aplicación específica para el desarrollo de instrumentos virtuales con tecnología VST, que permite su exportación tanto para *Windows* como para *Apple*. El mayor beneficio y por lo que destaca es por su interfaz gráfica que da una facilidad de uso muy importante y su capacidad para integrar muestras de audio en un formato accesible para estaciones de trabajo de audio digital (DAWs). Esta herramienta permitió organizar y mapear las muestras de timple a notas MIDI, facilitando su uso por músicos y productores en distintos contextos musicales.

Para la síntesis aditiva, se emplearon diversas herramientas de software para lograr una representación fiel y precisa del sonido del timple. Se utilizó el entorno de programación *MATLAB* debido a su capacidad avanzada para el procesamiento de señales y su flexibilidad para implementar algoritmos de síntesis personalizados. Los *add-ons*<sup>5</sup> de *Audio Toolbox* permitieron desarrollar y ajustar algoritmos que replican los componentes frecuenciales del timple, con un control detallado sobre los armónicos.

Por último, durante todo el proceso fue imprescindible el uso de un controlador MIDI para ir comprobando el resultado de cada cambio. En este caso se utilizó un teclado MIDI *AKAI MPK mini 2*.

---

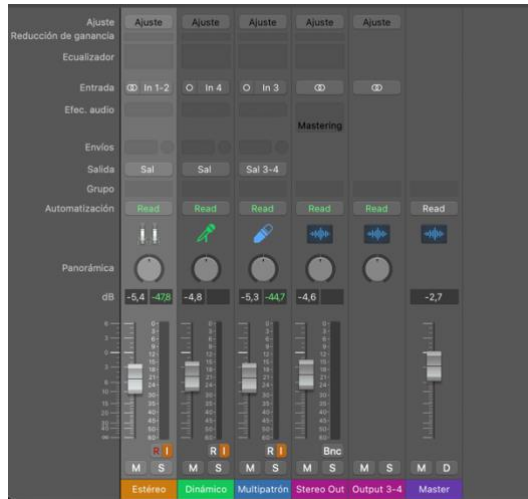
<sup>5</sup> Complemento o extensión que añade funciones adicionales a un software principal.



## 3.2. Sampler

### 3.2.1. Proceso de grabación

Para la grabación se ha configurado una sesión multipista; una pista de tipo mono por micrófono salvo para el par estéreo en que se combinan dos pistas en una estéreo; en el software de audio *Logic Pro-X*. Para facilitar la identificación se asignan nombres de pista, iconos y colores a cada una de ellas (véase Figura 22).



**Figura 19.** Mezclador de señales de audio de micrófonos para toma de muestras en sesión multipista de *Logic Pro X*. Figura de elaboración propia.

Para la captura de las muestras, en primer lugar, se comprobó el ajuste de ganancias de cada pista teniendo en cuenta la intensidad máxima y mínima a la que se va a tocar el instrumento. Se ajustó la ganancia de manera que estas intensidades quedaran entre los -20 dB a -3 dB. Este ajuste asegura que las señales no sean ni demasiado bajas, lo que podría perderse en el ruido de fondo, ni demasiado altas, lo que podría causar distorsión o *clipping*.

Para la grabación de las muestras se contó con la ayuda del músico Daniel Armijo, con una amplia experiencia en instrumentos de cuerda y en grabaciones de estudio lo que facilitó mucho el trabajo. Fue fundamental la comunicación en todo momento para mantener un control de la parte en que la grabación se encontraba y tratar de dar indicaciones para la mejora de los errores cometidos. Se envió la señal de uno de los micrófonos al músico a través de auriculares para permitir que él mismo supiera que tomas debía repetir y a la vez escuchase qué sonidos indeseados se colaban en la grabación.



**Figura 20.** Imagen de la grabación de muestras en el estudio.

Figura de elaboración Propia.

Para aportar variaciones según la velocidad o intensidad a la que se toque la nota, se dividió la dinámica en tres grabaciones por nota; *piano*, *mezzo* y *forte*. Esto se realiza para obtener la relación de armónicos correspondiente a cada intensidad. Por ejemplo, en notas con mayor intensidad el sonido es más brillante que en notas más suaves. Para llevar a cabo una interpretación con un nivel relativamente similar y conseguir la dinámica adecuada del instrumento, se acordaron unos márgenes de niveles para cada dinámica con el músico. De esta forma conocería entre que niveles debería estar cada nota. Para reducir el margen de error se mantuvo el medidor de nivel en una pantalla auxiliar en la cabina de grabación en todo momento para que el músico supiera como debería mejorar cada toma. Para facilitar la grabación y edición de la sesión, se realizó la captura de las diferentes notas y tomas de cada una de ellas en un archivo único por dinámica.

Dada la condición de artesanalidad del instrumento, y su pequeño tamaño, el timple en sus trastes más agudos presenta una clara desafinación provocada por su morfología. Para facilitar el uso del instrumento virtual se mantenía también el control de afinación cromática y se procedía a su corrección en cada nota que fallara.



**Figura 21.** Daniel Armijo realizando ajustes de afinación durante toma de muestras agudas.

Figura de elaboración Propia

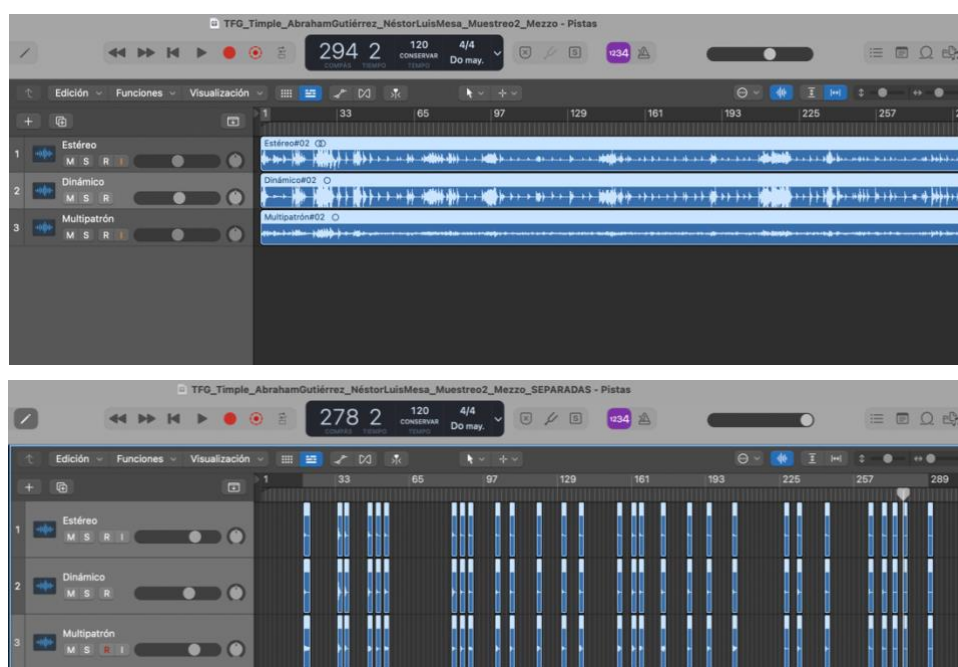
Una vez capturadas todas las notas de la tesitura del timble en las tres dinámicas definidas se pasa a repetir el proceso completo en el otro timble.

### 3.2.2. Procesado de muestras

Una vez obtenidas las muestras necesarias de cada instrumento, nota e intensidad, se procede a su procesado para obtener el mejor resultado posible. Este fase consistirá en la edición de errores, el procesado del audio, la mezcla de los diferentes micrófonos y la automatización del nivel. Finalmente se hará una breve descripción de los parámetros de exportación escogidos.

#### *Edición*

En la edición de audio existen dos subdivisiones: la correctiva y la selectiva. En este trabajo, se desarrolla una “edición selectiva” (Izhaki, 2003) del material. De tal manera que de las grabaciones realizadas de cada intensidad se acaba recortando y seleccionando una única toma de cada nota. En este proceso de selección se eliminaron las tomas con errores causados por afinación, ruido de fondo, errores humanos y de nivel.



**Figura 22.** Comparativa de antes (arriba) y después (abajo) del proceso de edición selectiva de un ejemplo. Figura de elaboración Propia.

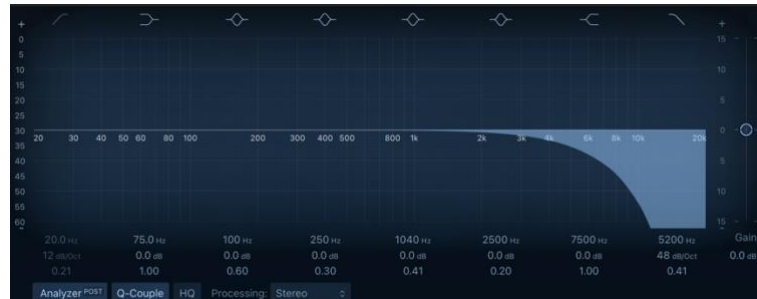
Realizada esta selección, se obtuvieron finalmente 30 archivos de audio por micrófono para el Abraham Gutiérrez y 29 para el David Sánchez. En el Anexo II, se puede observar un desglose de los archivos de proyecto utilizados, y cuantas tomas fueron necesarias en cada caso.

#### *Posprocesado de audio*

A continuación, se detallan los efectos y filtros aplicados a cada una de las pistas de audio para lograr un sonido lo más realista y libre de imperfectos posible.

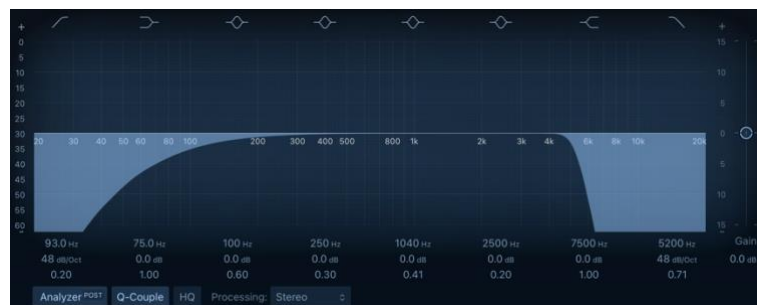
## Ecualización

En la parte de ecualización, se realizó un filtrado selectivo de la señal en la que se elimina el ruido eléctrico no deseado presente en altas frecuencias. Para ello se utilizó un filtro paso bajo con frecuencia central 5200 Hz y una atenuación de 48dB/oct. Este filtro se aplicó tanto al multipatrón como al micrófono dinámico.



**Figura 23.** Configuración del ecualizador paramétrico aplicado a la pista del micrófono dinámico. Figura de elaboración propia.

Sin embargo, el micrófono multipatrón, dado que se trata de un micrófono de diafragma más grande captaba ruido indeseado de baja frecuencia, causado principalmente por el movimiento del timplista y del instrumento. Es por ello por lo que se necesitó también de un filtro paso alto de frecuencia central 93 Hz y 48 dB/oct de atenuación.



**Figura 24.** Configuración del ecualizador paramétrico aplicado a la pista del micrófono multipatrón. Figura de elaboración propia.

Gracias a este filtrado se consiguió un sonido libre de imperfecciones sin afectar a las características del sonido del instrumento.

## Dinámica

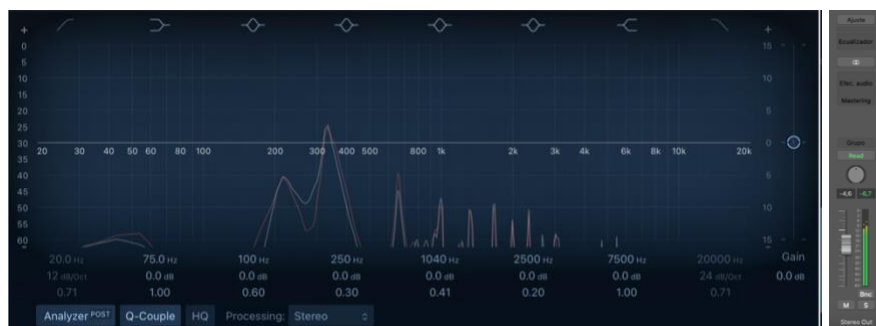
En cuanto a la dinámica, se aplicó una puerta de ruido tratando de eliminar los sonidos no deseados. De esta forma evitamos los sonidos que se pueden colar en silencios por los movimientos propios del músico o las indicaciones que se le designan. Para ello se configuró el umbral según la intensidad de la señal y los valores de *atack*, *hold* y *release* para que la reducción fuera natural y lo suficientemente rápida en actuación.



**Figura 25.** Ejemplos de puerta de ruido aplicadas a los canales del micrófono multipatrón (arriba) y dinámico (abajo). Figura de elaboración Propia.

### Mezcla

Una vez aplicados los efectos necesarios para que las pistas de manera individual tengan un sonido consistente, limpio y realista, se procede a la mezcla de las 3 pistas para obtener el sonido final de la muestra. Para ello, se emplea el análisis espectral de la señal en tiempo real mediante la opción *POST-Analyzer* del Ecualizador Paramétrico (véase Figura 28 (parte izquierda de la figura)) y se comprueba la señal estéreo conjunta (véase Figura 28 (parte derecha de la figura)).



**Figura 26.** *POST-Analyzer* frecuencial de la señal combinada de los 3 canales (izq). Balance estéreo de salida de la señal cominada de los 3 canales (dcha.) Figura de elaboración propia.

Como se observa, la respuesta en frecuencia es la deseada para el tipo de muestras de las que se trata. Para mejorar la respuesta en frecuencias medias y graves se decide reducir ligeramente la ganancia del micrófono dinámico en la mezcla. La mezcla final se puede observar en la figura 29.



**Figura 27.** Ajuste de ganancias de cada canal para mezcla final de muestras.

Figura de elaboración Propia.

Finalmente, para obtener el archivo de audio mezclado de cada muestra se realiza un *Bounce in Situ* del conjunto de las tres grabaciones.

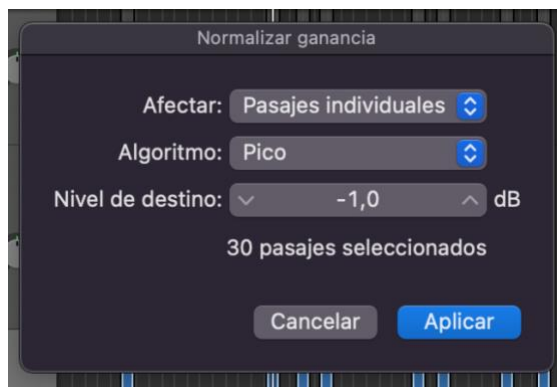


**Figura 28.** Pista de muestras combinadas de la mezcla de micrófonos. Figura de elaboración Propia.

### *Automatización*

Dado que los archivos de audio que se van a exportar son para su utilización en un instrumento virtual, se necesita que todas las notas tengan la misma intensidad para la misma velocidad MIDI. Por este motivo, a la hora de exportar, se debe realizar un proceso de normalización de los archivos de audio, de manera que tengan la misma intensidad.

Para este fin se utiliza la opción de “normalizar ganancia de pasaje” perteneciente al grupo de herramientas avanzadas de *Logic Pro*. En el panel de control se puede escoger de qué manera afecta la normalización, el algoritmo utilizado y el nivel de destino de la señal como se observa en la figura 31.



**Figura 29.** Panel de control de la herramienta "Normalizar ganancia de pasaje".

Figura de elaboración propia.

En una primera versión, se intentó simular el nivel de intensidad normalizando cada dinámica realizada (*piano*, *mezzo*, *forte*). Para ello, observando el rango dinámico que se utiliza en instrumentos virtuales de timbre similar (véase Anexo III), se dedujo que los cambios de dinámica en pico se suelen situar entre los -9 dB los más bajos, y -5 dB los más altos. A la vista del resultado, para el instrumento virtual se tomó la decisión de normalizar las muestras de las 3 variaciones de dinámicas dentro de este rango como se muestra en la tabla 3.

**Tabla 3.** Niveles de normalización escogidos (primera opción).

Dinámica	Nivel de normalización (dB)
<i>piano</i>	-10
<i>mezzo</i>	-7
<i>forte</i>	-5

En una segunda versión se ha corregido un error cometido en este paso, que se explica en profundidad en el apartado “3.2.4. Pruebas y ajustes”.

### Exportación

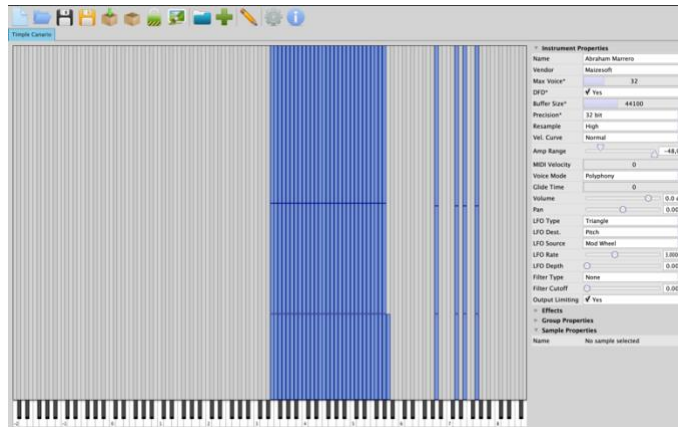
En el momento de la exportación, los archivos fueron nombrados para facilitar la colocación en el instrumento virtual, con el nombre de la nota en notación inglesa. Posteriormente, se llegaría a la idea de numerarlas según el número de nota MIDI, esto ayudaría a la presentación en carpetas para su utilización en el *sampler*, pero también para su escaneado para la síntesis del apartado 3.3.

### 3.2.3. Implementación del instrumento virtual

Con las muestras ya exportadas individualmente podemos proceder a la implementación del instrumento virtual. Para lo cual se necesita de un software de enrutado de muestras a notas MIDI, estos softwares son llamados *samplers*. En este caso se utiliza la aplicación *Maize Sampler*, un software de pago desarrollado por *Maizesoft*. Este programa permite la creación de instrumentos virtuales con tecnología VST de una



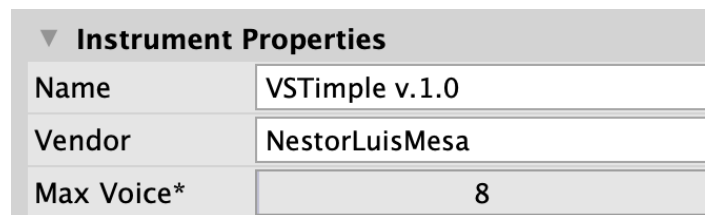
manera práctica e intuitiva a través de su interfaz, generando el código VST automáticamente.



**Figura 30.** Interfaz Maize Sampler. Figura de elaboración Propia.

### Configuración del instrumento

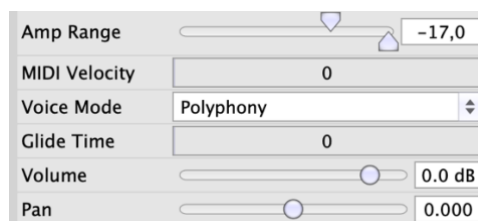
Comenzando por la configuración del “instrumento”, que es la configuración general del funcionamiento del *plugin*. Entre todas las posibles modificaciones, en primer lugar, le ponemos un nombre y el nombre del desarrollador, tal y como se aprecia en la figura 33 para este caso.



**Figura 31.** Configuración de propiedades del instrumento (i) en Maize Sampler.

Figura de elaboración Propia.

En cuanto a configuración del funcionamiento, se selecciona el mínimo de voces posibles (ocho) ya que no se puede escoger cinco que es el número de cuerdas (véase Figura 34). Además, ajustamos la variación de amplitud a los valores deseados, en este caso se ha escogido de -17dB a 0dB. Se selecciona el modo de voz polifonía y el tiempo de *glissando* de 0ms que son los valores más utilizados para instrumentos de cuerda. En cuanto al volumen y paneo de la señal se deja en los valores por defecto.



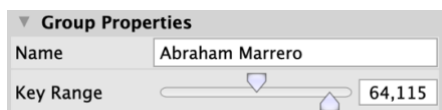
**Figura 32.** Configuración de propiedades del instrumento (ii) en Maize Sampler.

Figura de elaboración Propia.

La pestaña de efectos se deja sin aplicar ya que se pretende que la señal sea natural. En la pestaña de propiedades del grupo, se configura el funcionamiento de cada mapeado de muestras, en este caso de cada timple (véase Figura 35). Luego se le coloca el nombre



de cada timple y se ajusta el rango de teclas utilizadas, que en este caso abarca de la nota 64 a la 115.

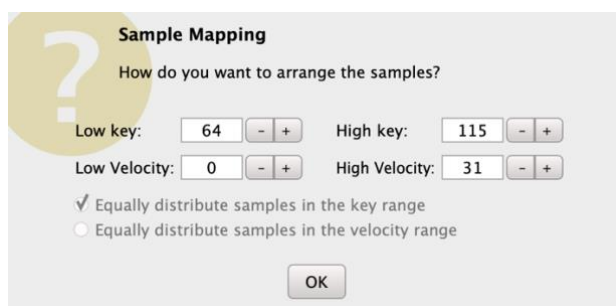


**Figura 33.** Configuración de propiedades del grupo en Maize Sampler.

Figura de ejemplo del Abraham Marrero. Figura de elaboración propia.

### Enrutado de samples

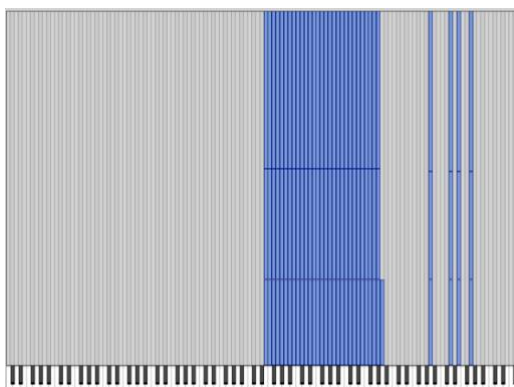
Para la importación y gracias a la numeración de los archivos se utiliza el botón “Add samples...” que abre el siguiente cuadro de diálogo.



**Figura 34.** Cuadro de diálogo “Sample Mapping” para el enrutado de muestras en Maize Sampler. Figura de ejemplo de enrutado de notas “piano”. Figura de elaboración propia.

Como se observa en la figura 36, podemos introducir varios archivos, seleccionando el rango de velocidad y de notas que se desea cubrir y escoger de qué forma distribuirlas. En el caso de ejemplo presentado en la figura, se elige la opción de distribuir en el rango de teclas, seleccionando de la nota 64 (MI3) a la 115 (LA5), y el rango de velocidades establecido para las muestras *piano*.

Repitiendo el proceso para todas las notas, se consigue la disposición de todas ellas sobre el teclado. Por último, los armónicos se disponen como se suele hacer en otros instrumentos virtuales, en la octava superior a la nota más aguda en sus correspondientes notas, y se tiene como resultado, el mapeo de las muestras con sus notas y velocidades.



**Figura 35.** Disposición de muestras finales en teclas y amplitudes. Figura de elaboración propia.

Una vez logrado el enrutado de muestras, se repite el proceso para el segundo timple.

## Interfaz Gráfica

Para el diseño de la interfaz gráfica del *plugin* se simuló el resultado deseado mediante una inteligencia artificial de manera que se pudiera comprobar cómo quedaría la fotografía ideada. Para ello se empleó la inteligencia artificial generativa de imágenes de *Canva*. Una vez obtenida la fotografía se colocó en *Illustrator* para agregar el nombre del *plugin* “VSTimple” y las etiquetas de las perillas que se desean añadir. Por un lado, para el control de la ganancia del instrumento “*volume*” y por otro lado, para el control de la cantidad de “*reverb*” del mismo.

Ya acabada la imagen de base se introdujo en el editor de interfaz gráfica de *Maize Sampler* agregando los campos de “*Instrument*” y “*Group*” además de las perillas ya mencionadas. El resultado se puede apreciar en la siguiente figura.



**Figura 36.** Editor de interfaz gráfica de *Maize Sampler*. Nota: Se puede apreciar el diseño elaborado para el *plugin*. Figura de elaboración Propia.

### Exportación en formato VST

Una vez acabado el *plugin* podemos realizar su exportación en formato VST. La forma de funcionar de *Maize Sampler* es mediante instrumentos y *plugins*. El *plugin* puede cargar varios instrumentos. Primero exportamos el instrumento en formato “.mse” y a continuación en el botón “export plugin” podemos generar un nuevo *plugin* con nuestra apariencia de interfaz gráfica.

### 3.2.4. Pruebas y ajustes

Instalando el *plugin* en un ordenador se confirma que todo funciona y responde correctamente a través de un DAW. Se comprueba que funcionan los controles de *reverb* y *volume* así como las teclas en todas sus dinámicas y armónicos. La instalación se comprobó en varios softwares como *Ableton Live*, *Logic Pro* y *Reaper*, y tanto en *macOs* como en *Windows*. En el Anexo IV se puede encontrar una guía del proceso de instalación en ambos sistemas operativos.

En este proceso de comprobación se ha encontrado un error, ya que el programa *Maize Sampler* permite realizar una curva de amplitudes a partir de un rango que el desarrollador determina. De esta forma normaliza cada nota para que, según la velocidad, responda a una intensidad en ese rango. A raíz de esto, se ha tenido que repetir el proceso de normalización de los archivos de audio finales para que estos sean todos iguales (véase tabla 4), dando lugar a una segunda versión que difiere de la vista en el apartado de “Automatización”. Gracias a esto, las amplitudes varían progresivamente según la velocidad.

**Tabla 4.** Niveles de normalización para cada dinámica tras la corrección.

<b>Dinámica</b>	<b>Nivel de normalización (dB)</b>
<i>piano</i>	-1
<i>mezzo</i>	-1
<i>forte</i>	-1

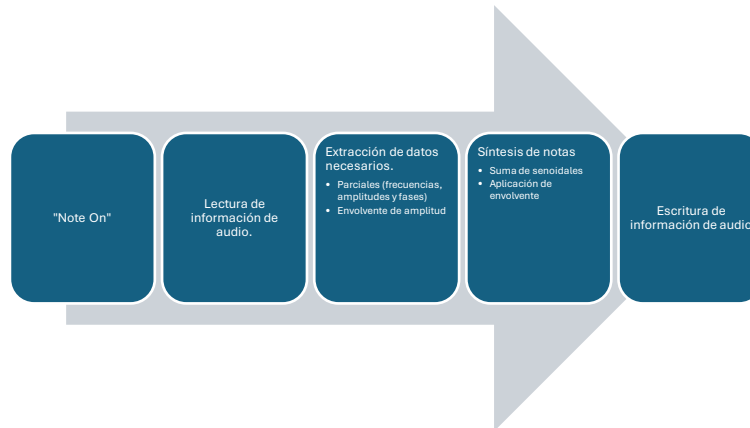
Una vez conformes con el resultado se realiza una exportación de un pasaje MIDI de un fragmento de la obra para timple “*Chipude*” de José Antonio Ramos, para la comparativa entre los distintos prototipos logrados y para su evaluación. Tanto este como los pasajes generados para su comparativa, así como la grabación de timple original se pueden escuchar en el enlace de Anexo V.

### 3.3. Síntesis Aditiva

Para el desarrollo de un prototipo de instrumento virtual de síntesis aditiva se ha escogido el software *Matlab*. A través del *add-on* de *Audio Toolbox* se obtiene acceso a las herramientas necesarias para poder leer entradas y escribir en salidas de audio.

#### 3.3.1. Desarrollo

La síntesis de audio una vez se recibe el mensaje de “NoteON” se ejecutará en los pasos que marca el diagrama de flujos siguiente:



**Figura 37.** Diagrama de bloques funcionamiento del código de síntesis aditiva.

Figura de elaboración propia.

Para comprobar el funcionamiento de la síntesis aditiva y la configuración correcta se implementó un código que realiza todo el proceso con una sola muestra de audio (véase Anexo VI: Código de fuente 1). Una vez comprobado el funcionamiento se fragmentó en varios *scripts* con el objetivo de conseguir disminuir la latencia que proporcionaba esta función (de hasta de 10 segundos) entre la recepción del mensaje MIDI y la escritura de audio. Para lograrlo se eliminaron los pasos intermedios de lectura de la información de la muestra y extracción de los datos, pues son los pasos que más potencia computacional y tiempo requieren y se desarrollaron varios *scripts* que realizaran estos pasos de alta latencia solamente una vez, y almacenan los datos necesarios en matrices en un directorio local. Esto también facilitaría su futuro desarrollo como VST pues dejamos parametrizados los datos estrictamente necesarios.

#### *Lectura de información de audio*

En este primer paso, se ha programado un *script* llamado `extractAndSaveAudioData` (véase Anexo VI: Código de fuente 2) en el que a partir de la carpeta contenedora y un archivo al que escribir la información se extrae la información relevante de las muestras de audio y se genera una matriz con ella.

Para esta función se desarrolla un bucle, mostrado en el código de a continuación, que recorre la carpeta en que se encuentran las muestras, y del nombre de archivo extrae el número de nota que se ha colocado para saber en qué posición guardarlo. Haciendo uso de la función `audioread` de *Matlab* se escribe la información en una matriz junto con la frecuencia de muestreo.

```

        % Procesar cada archivo

for i = 1:length(files)

    % Obtener el nombre del archivo

    fileName = files(i).name;

    % Extraer el número de nota MIDI del nombre del archivo

    [~, name, ~] = fileparts(fileName); % Obtener el nombre sin la extensión

    noteNumber = str2double(name(1:end-1)); % Extraer el número de nota, asumiendo que el número está al
principio y es seguido por una letra (por ejemplo, '64M')

    % Leer el archivo de audio

    [audio, fs] = audioread(fullfile(audioFolder, fileName));

    % Almacenar los datos en la celda correspondiente

    if noteNumber >= 1 && noteNumber <= maxNoteNumber

        audioData{noteNumber} = audio; % Almacenar el audio en la celda correspondiente a la nota

        sampleRates(noteNumber) = fs; % Almacenar la frecuencia de muestreo correspondiente a la nota

    else

        warning('Número de nota %d fuera del rango esperado.', noteNumber);

    end

end
end

```

Finalmente la matriz se guarda en un archivo “.mat” mediante la función *save* de Matlab para su uso posterior en el sintetizador.

```
save('AudioData.mat','AudioData')
```

### *Extracción de datos necesarios*

Para la extracción de los datos se han desarrollado dos *scripts* uno para la información relativa a los parciales, y otro para la información de la envolvente de amplitud de cada muestra.

#### Parciales (frecuencias, amplitudes y fases)

A partir del *script* `processAndStoreFormants` (véase Anexo VI: Código fuente 3) se puede almacenar la información de los parciales de la señal analizada a partir de la carpeta contenedora de los archivos de audio.

Este proceso se realiza mediante otra función desarrollada para ello (`extractPartials` (véase Anexo VI: Código fuente 4). Con ella se obtiene la información de amplitudes,

frecuencias y fases a partir del archivo y valores umbral. Esta función es llamada como se expone en el código siguiente.

```
%% Extraer parciales del archivo de audio

[ frecuencias, amplitudes, phases ] = extractPartials(fullfile(audioFolder, fileName),
30, 1000);
```

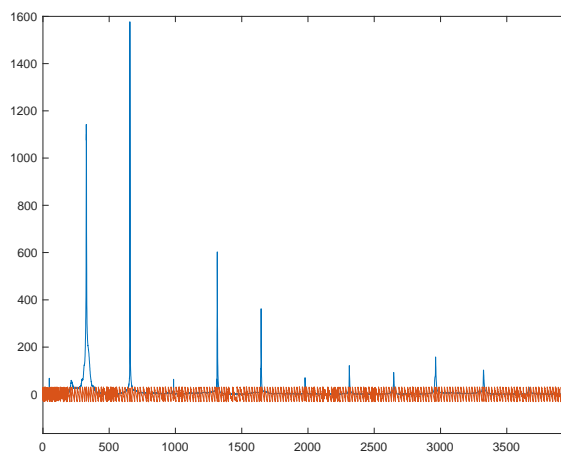
Para entender el funcionamiento de estas dos funciones debemos analizarlo por pasos y gráficamente. En primer lugar, se extrae la información frecuencial de las grabaciones de audio mediante la transformada rápida de Fourier (*fft* en *Matlab*).

```
% Aplicar la transformada de Fourier

N = length(y);

Y = fft(y);

f = (0:N-1) * (fs / N);
```



**Figura 38.** *Espectro frecuencial de la nota MI3 del timble Abraham Marrero.*  
*Nota: Se representa en azul magnitud y en rojo fase de la transformada rápida de Fourier.*

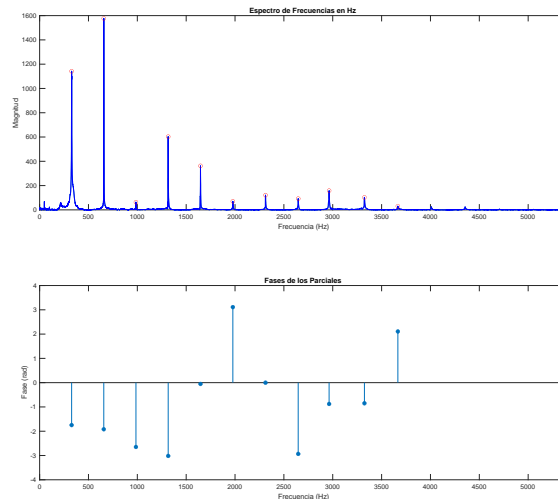
*Figura de elaboración Propia.*

La figura 40 muestra la FFT de la nota MI3 del timble Abraham Marrero, en la gráfica se distinguen los parciales (formantes) claramente diferenciados. Son las frecuencias donde se sitúan los máximos de la señal y que se corresponden a los armónicos de la nota tocada. Para poder identificarlos de manera automática se emplea la función *findpeaks* de *Matlab* especificando unos valores de umbral y distancia mínimas.

```
% Identificar los parciales (picos en el espectro)

[pks, locs] = findpeaks(magnitud, 'MinPeakHeight', threshold, 'MinPeakDistance',
minDistance);
```

Mediante pruebas y observando los gráficos de cada paso, se determinan los valores adecuados de estos valores de umbral y distancias mínimas. En la gráfica de la figura 41, se observa la detección de formantes realizada, de la que se extrae la frecuencia, amplitud y fase de cada uno. Esta información servirá para el sintetizado de las notas del timble.



**Figura 39.** (arriba) Gráfica de espectro frecuencial de nota MI3 de timble en la que se señalan parciales. (abajo) Gráfica de fases de los parciales señalados.

Figura de elaboración Propia.

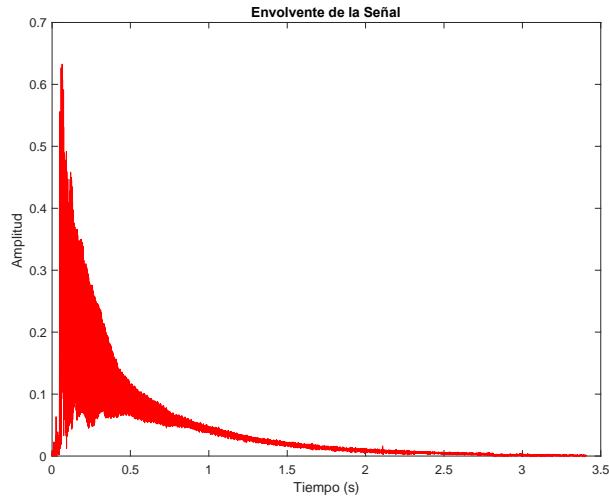
### Envolvente de amplitud

Con todo lo elaborado hasta ahora se conseguiría un sonido muy similar al emitido por el timble, pero este sería constante en el tiempo en cuanto a amplitud. Para poder lograr la caída de intensidad con respecto al tiempo de manera realista se debe extraer la envolvente de amplitud de la señal. Tal y como se explica en el apartado de Síntesis de este TFG, esto se puede lograr a través de la transformada de Hilbert.

El proceso descrito se ejecuta por un *script* independiente llamado *extractAndSaveEnvelopes* (véase Anexo VI: Código de fuente 5) que tiene por objeto realizar la transformada de Hilbert y extraer las envolventes de una serie de grabaciones para guardar los resultados en una matriz. En el siguiente extracto del código de la función se realiza la transformada y se extrae la envolvente *envelope*.

```
% Extraer la envolvente usando la transformada de Hilbert
```

```
analyticSignal = hilbert(audio);  
  
envelope = abs(analyticSignal);
```



**Figura 40.** Gráfica de envoltura extraída de muestra de nota MI3 de timple.

Figura de elaboración propia.

### *Sintetizado de notas*

Con toda la información recopilada de las muestras de audio de cada nota del timple en datos, se puede hacer el código del sintetizado de notas. Se realiza en la función *synthesizeSoundFromData* (véase Anexo VI: Código fuente 6).

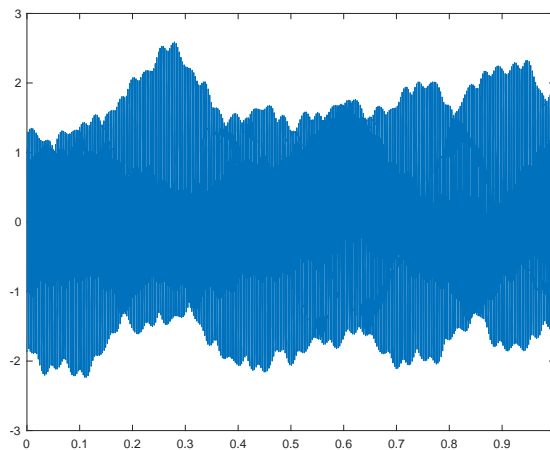
En primer lugar, con los datos de amplitudes, frecuencias y fases de los formantes se generan y se hace la suma de las senoidales a través de un bucle *for*. La señal resultante tiene un aspecto como el de la figura 43.

```
% Generar la señal sintetizada usando un rango de frecuencias
```

```
for i=1:length(F)
```

```
    y = y + amplitude*A(i)*sin(2*pi*F(i)*t + P(i)); end
```

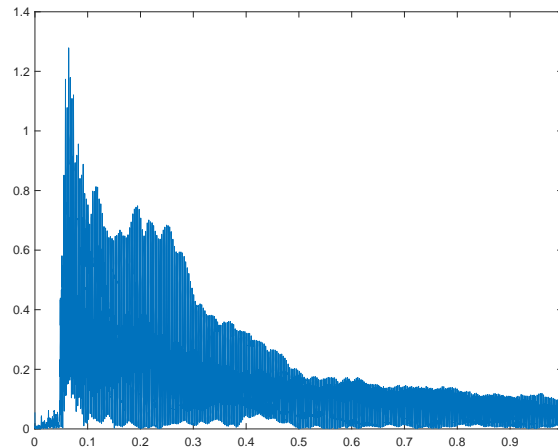
Aplicando este proceso a la nota MI3 vemos como la señal generada es como en la figura 43. Se pueden observar las distintas señales que conforman la señal sintetizada superpuestas.





**Figura 41.** Gráfica que muestra la señal sintetizada de nota MI3 de timple respecto al tiempo. Figura de elaboración Propia.

Una vez sintetizada la señal se aplica la envolvente y se obtiene el resultado final acercándose considerablemente a la señal original (véase figura 44).



**Figura 42.** Gráfica que muestra la señal sintetizada de la nota MI3 de timple con la envolvente de amplitud aplicada frente al tiempo. Figura de elaboración Propia.

#### *Escritura de información de audio*

Con todos los datos recopilados y la función de sintetizado desarrollada se puede finalmente desarrollar la función del sintetizador completa. Esta función está compuesta principalmente por la lectura de mensajes MIDI, la llamada a la función de sintetizado y la escritura de audio.

Primeramente, se deben cargar todos los ficheros que contienen la información que se utilizará para la síntesis digital; estos son: ficheros de frecuencias, de amplitudes, de fases, de envolvente, y el que contiene la información de las grabaciones.

```
%%IMPORTACIÓN DE VARIABLES

envelopesdata=load('EnvelopesData.mat');

    envelopes = envelopesdata.envelopes;

loadedPhaseData = load('phaseData.mat');

    phases = loadedPhaseData.allPhases;

loadedAmpData = load('ampData.mat');

    amplitudes = loadedAmpData.allAmplitudes;

loadedFreqData = load('freqData.mat');

    frecuencias = loadedFreqData.allFrequencies;

matFile="AudioData.mat";
```

Tras esto se inicializa la conexión MIDI, y una vez leído el mensaje de “NoteOn” se llama directamente a la función de sintetizado.

```
% Generar y reproducir sonido sintetizado
```

```
[synthesizedSound] = synthesizeSoundFromData(note, amplitudes, phases, frequencies, envelopes, duration, fs, matFile, amplitudes, phases, frequencies, envelopes);  
  
% Enviar el sonido al dispositivo de audio  
  
sound(synthesizedSound,fs);
```

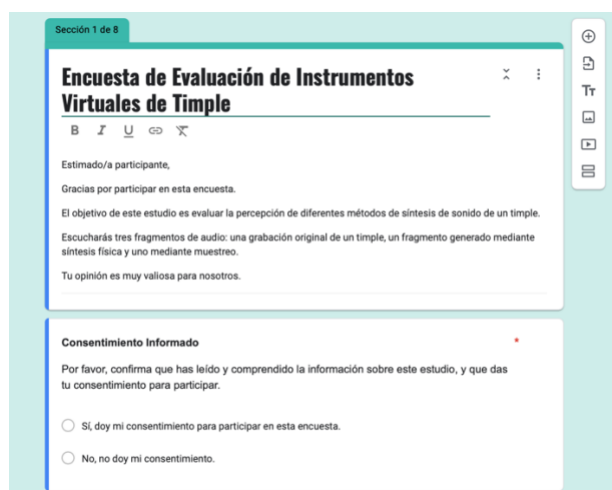
### 3.3.3. Pruebas y ajustes

Finalmente se puede comprobar como gracias a la parametrización de datos para evitar pasos redundantes en la aplicación, se ha conseguido una reducción de la latencia, siendo esta de unos dos segundos como máximo.

El resultado es un sonido coherente en las muestras individuales. Para poder comprobar cómo se desenvuelve en fragmentos con varias notas, dado que no está integrado en un DAW se ha implementado una última versión (véase Anexo VI: Código fuente 7). El objetivo era que leyera archivos MIDI utilizando para ello la función *readmidi* del usuario Valerio Biscione disponible en los *add-on* de *Matlab* que permite leer un archivo MIDI, extrayendo los datos necesarios para su reproducción con el sintetizador. De esta forma se ha podido realizar la comparativa con el resto de las opciones implementadas.

## 4. Validación de resultados

Para comprobar la validez de los modelos en la práctica, así como para contemplar posibles vías de mejora, se ha realizado una encuesta de evaluación. La intención era recoger la opinión y sensaciones que desprende la experiencia auditiva de cada modelo en personas familiarizadas con el sonido del timple y a ser posible que tengan experiencia tocándolo.



**Figura 43.** Captura de pantalla de la encuesta de opinión realizada para la evaluación de los instrumentos desarrollados. Figura de elaboración Propia.

### 4.1. Planteamiento de la encuesta de evaluación

Durante la realización de la encuesta se trataron de crear preguntas sin posibilidad de generar sesgos en el encuestado siendo lo más claras posibles. Para la confección de las posibles respuestas se siguió la escala de evaluación Likert, utilizando 5 opciones siendo una de estas neutral tal y como recomiendan algunos expertos (Matas, 2018). Se utilizó la herramienta de Google “Google Forms” por su versatilidad en la recopilación de datos en formato de tabla y gráficos.

En primer lugar, se crearon tres fragmentos de audio de la misma pieza musical, concretamente de *Chipude (2009)* del famoso timplista José Antonio Ramos. Los 3 fragmentos se basan en el mismo fragmento MIDI como entrada siendo este implementado en el *sampler* del timple del luthier de David Sánchez, el *sampler* del timple de Abraham Marrero y en el sintetizador aditivo.

Cada fragmento de audio fue evaluado por los encuestados en las preguntas de evaluación en términos de: “Calidad del Sonido”, “Realismo del Sonido” y “Preferencia Personal”. Por otra parte, se permitía dejar comentarios en formato de texto para poder dar una evaluación más detallada y sugerencias de mejora. Además, se incluyó una sección para comparar directamente los fragmentos de audio, donde se pidió a los participantes que indicaran cuál consideraban más realista y cuál les gustó en general.

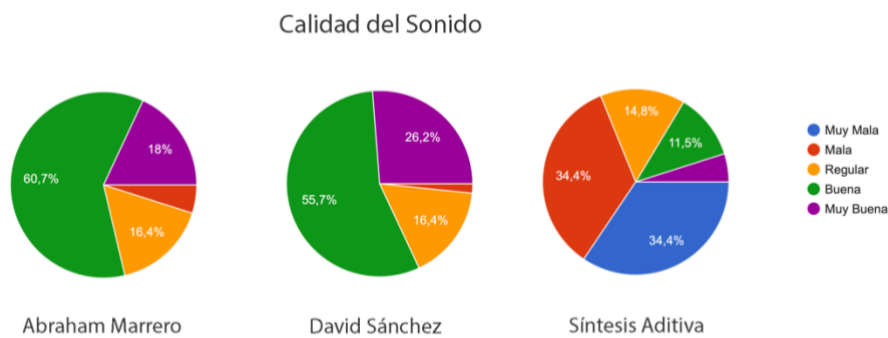
### 4.2. Resultados Obtenidos

La encuesta fue completada por un número representativo de participantes con conocimiento previo del sonido particular del instrumento. En concreto participaron del estudio 61 personas, siendo prácticamente equivalente la cantidad de mujeres y hombres. El conjunto de población estudiado es, en su mayoría, oyentes no profesionales, siendo el porcentaje de músicos de un 21,3%. Además, un 14,8% del total tiene experiencia tocando

el timble. Todos los resultados de la encuesta se pueden revisar de manera gráfica en el “Anexo VII. Representación gráfica de los resultados de encuesta evaluativa.”

En cuanto a las preferencias personales de los encuestados se ha percibido una inclinación a favor del banco de sonidos del timble de David Sánchez, con un 72,1% de los votos, aunque en los comentarios adicionales la mayoría de las personas comentan que han tenido dificultades en la distinción de características entre los dos *samplers*. Esto puede indicar un mal tratamiento de la señal en que se perdieran características claves para la caracterización distintiva. Por otra parte, los comentarios también hacen referencia al gusto personal y, en este sentido, ambos modelos de muestreo serían válidos.

Las gráficas de la 46, muestran la distribución de las respuestas respecto a la calidad del sonido del instrumento virtual según el modelo evaluado. La mayoría de los encuestados calificaron la calidad del sonido del *sampler* del timble de Abraham Marrero como "Buena" (60,7%) o "Muy Buena" (18%). Por su parte, el modelo realizado sobre el timble de David Sánchez muestra unos resultados algo mejores con una mayoría menor considerando una calidad de sonido “Buena” (55,7%) pero con un 26,2% de los votos a “Muy Buena”. Sin embargo, tal y como se muestra en la figura 51, en la evaluación de la calidad del sonido de la síntesis aditiva, la mayoría de encuestados coincidieron en que la calidad era “Mala” (34,4%) o “Muy Mala” (34,4%).

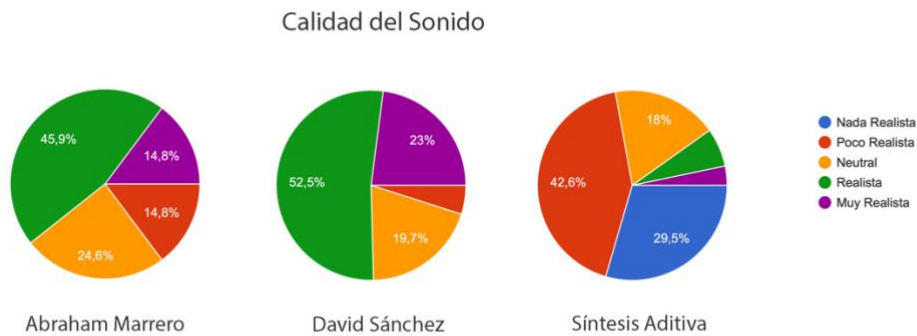


**Figura 44.** Representación gráfica de la calidad de sonido percibida por los encuestados con respecto al modelo evaluado. Figura de elaboración propia.

Se ha comprobado como los encuestados coinciden en una evaluación de calidad muy similar en los dos modelos de reproducción de muestras pregrabadas siendo esta “Buena” y “Muy Buena” para la gran mayoría de respuestas. Además, muy pocos encuestados (menos de un 5%) consideraron este sonido “Malo”.

En segundo lugar, los participantes han evaluado en base a su percepción el realismo del sonido reproducido, en cuanto a cómo se parece al instrumento real según el modelo de síntesis. Como se puede apreciar en la figura 47, para el instrumento basado en el Abraham Marrero la gran mayoría consideran que se trata de un modelo “Realista” (45,9%) y un pequeño porcentaje lo señala como “Muy Realista” (14,8%). Al igual que con la calidad del sonido, los resultados son muy parecidos para el instrumento basado en el timble de David Sánchez, ya que como ilustra la figura, el 52% de los participantes lo consideraron “Realista”, y además se alza con un 23% de votos para “Muy Realista”. Para acabar se puede observar una tendencia hacia la calificación negativa del realismo en la síntesis llevada a cabo mediante modelado espectral. Esto se puede deber debido al fragmento escogido en que no se desenvuelve tan bien. La mayoría de los encuestados

consideraron el sonido “Poco realista” (42,6%) y casi un 30% lo consideró “Nada Realista” (29,5%).



**Figura 45..** Representación gráfica del realismo del sonido percibida por los encuestados con respecto a la reproducción de distintos modelos de síntesis. Figura de elaboración propia.

De los datos recopilados, se muestra que existe un cierto grado de consenso en la validez de los modelos realizados mediante bancos de sonidos. Este método ha sido bien recibido por los usuarios, aunque algunos han hecho comentarios señalando "excesivo *reverb* y *sustain*". También se ha visto que el modelo aditivo desarrollado en Matlab ha sido clasificado como "nada realista". Gracias a estos resultados, se han podido establecer vías de trabajo futuro realistas y consecuentes para mejorar el *plugin* en cuanto a calidad y realismo del sonido.

Tanto los datos expuestos en la presente memoria, como datos adicionales sobre los resultados de la encuesta evaluativa pueden encontrarse en el Anexo VII: “Resultados de encuesta evaluativa”.

## 5. Conclusiones

Este trabajo ha cumplido los objetivos planteados, logrando la creación de un instrumento virtual de timple que reproduce fielmente el sonido característico del instrumento. El resultado es una herramienta útil para la composición y producción musical digital.

Gracias a la validación del modelo llevada a cabo a través de una encuesta de evaluación a usuarios amateurs, músicos y profesionales del sector, se puede afirmar que el modelo realizado por reproducción de muestras pregrabadas es de una solución de calidad, profesional y aplicable a las necesidades que trata de cubrir. Además, gracias a estos resultados, se han podido establecer vías de trabajo futuro realistas y consecuentes para mejorar el *plugin*.

El modelo de síntesis aditiva no resulta adecuado para la realización del *plugin*. A las limitaciones asociadas a la elevada latencia generada por el procesado, se suma la mala calidad del sonido obtenido. Según lo recogido en la encuesta, es con diferencia, el peor de los tres sonidos evaluados. A raíz de esto se ha propuesto una alternativa para su implementación. Esta alternativa consiste en generar un archivo de audio con la información recopilada en un buffer, como se detalla en el Anexo VI: Código fuente 8. Esta solución ofrece un sonido mucho más realista y demuestra el potencial de la síntesis aditiva cuando se aplica correctamente.

Durante todo el proceso, se ganaron conocimientos de gran valor para el desarrollo de instrumentos virtuales; se adquirió un profundo conocimiento sobre las características constructivas y sonoras del timple canario; se ampliaron los conocimientos sobre técnicas avanzadas de grabación, muestreo y síntesis aditiva de audio llevándolos a la práctica en el desarrollo del *plugin*, y se desarrollaron competencias en la síntesis digital, específicamente en el uso de transformadas de Fourier y de Hilbert para el modelado espectral aditivo. Por último, la validación del instrumento virtual proporcionó experiencia en la evaluación de soluciones de síntesis y muestreo, así como en la recolección y análisis de retroalimentación de usuarios.

En este proyecto me he enfrentado a desafíos significativos, como la falta de documentación técnica sobre la grabación del instrumento, por lo que se requirió la ideación y puesta en práctica de un esquema de microfoneado propio. La captura precisa del timbre y la resonancia únicos del timple, fue una tarea que requirió de una implementación técnica precisa y que demandó ajustes iterativos para lograr el resultado más adecuado.

En resumen, el timple virtual desarrollado en este proyecto funciona de forma satisfactoria y es la primera herramienta disponible para el modelado del timple. Este modelo establece una sólida base para futuros desarrollos más realistas del timple, poniendo en valor el potencial de las tecnologías de síntesis y muestreo para recrear sonidos musicales de manera eficaz y precisa.

## 6. Trabajo Futuro

Una vez concluido este Trabajo Final de Grado, se pueden establecer diversas vías de trabajo futuro para realizar posibles mejoras o ampliaciones del resultado final antes de su comercialización. A continuación, se detallan algunas de estas posibles líneas de desarrollo.

### 6.1. Mejora del Plugin de Reproducción de Muestras Pregrabadas

Del análisis del cuestionario realizado, se ha podido identificar como el comportamiento dinámico del *plugin* es un área con potencial de mejora. Las siguientes modificaciones son propuestas específicas para abordar estas áreas de mejora.

1. Ajustar los parámetros de reverberación para conseguir un sonido más natural y realista.
2. Optimizar la curva ADSR para mejorar la respuesta dinámica del instrumento virtual.
3. Simplificar el proceso de instalación y configuración del *plugin*, reduciendo la necesidad de múltiples componentes y mejorando la usabilidad y compatibilidad con diferentes estaciones de trabajo de audio digital (DAWs).

### 6.2. Desarrollo de Prototipos de Síntesis Aditiva

El prototipo de síntesis aditiva ha mostrado resultados prometedores, pero también ha dejado claro que hay espacio para la mejora. Las siguientes vías se presentan como futuras líneas de investigación en esta área.

1. Investigar y desarrollar distintos métodos de síntesis alternativos, que puedan ofrecer modelos más realistas y que permitan un control de parámetros en tiempo real.
2. Mejorar la simulación de la interacción entre las notas que se tocan, de manera que se asemeje más a la realidad física del instrumento.
3. Por ello, se establece como siguiente paso la creación de un modelo físico que simule todos los aspectos de la generación de sonido en el timbre. Este enfoque requerirá de un conocimiento técnico avanzado, pero podría resultar en un modelo mucho más eficiente y realista, capturando las sutilezas del comportamiento del instrumento real.

### 6.3. Comercialización y Usabilidad

Con el objetivo de poder lanzar el instrumento virtual como producto al mercado aún se deberían mejorar ciertos aspectos de este y conseguir solventar varias carencias. Las tareas que se exponen a continuación se consideran necesarias para contribuir a su lanzamiento.

1. Realizar pruebas controladas de usuario en entornos de uso reales con músicos profesionales y amateurs para recibir retroalimentación y realizar los ajustes adicionales antes de la comercialización. Estas pruebas ayudarán a identificar posibles áreas de mejora y asegurarán que el instrumento virtual cumpla con las expectativas de los usuarios.
2. Desarrollar una documentación exhaustiva y recursos de soporte técnico para facilitar el uso del *plugin* por parte de usuarios finales. Una documentación bien elaborada ayudará a los usuarios a entender mejor el funcionamiento del *plugin*,

mientras que los recursos de soporte técnico ofrecerán asistencia en caso de problemas o dudas, mejorando así la experiencia del usuario.



## Referencias

Cárdenas, A. (Director). (2022). *El Timple – El sonido de la música canaria* [Entrevista]. <https://youtu.be/7XxqONA4qhk>

Casa Museo del Timple. (s. f.). *Orígenes*. Casa Museo del Timple. Recuperado 21 de enero de 2024, de <https://casadeltimple.org/origenes/>

Collins, M. (2003). *A professional guide to Audio Plug-ins and Virtual Instruments*. (1ªed.). Focal Press.

de Oliveira, H. M., & de Oliveira, R. C. (2017). *Understanding MIDI: A Painless Tutorial on Midi Format*.

UIT de Telecomunicación. (2012). *VST (Virtual Studio Technology)* [Educativa]. Ingeniatic. <https://www.etsist.upm.es/estaticos/ingeniatic/index.php/tecnologias/item/659-vst-virtual-studio-technology.html>

Giraldo C., D., & Quintero M., O. L. (2014). *Análisis de señales de audio utilizando la transformada de Gabor*.

Gobierno de Canarias: Consejería de Educación, Formación Profesional, Actividad Física y Deportes. (2023). Timple. En *CanariWiki*. <https://www3.gobiernodecanarias.org/medusa/wiki/index.php?title=Timple>

González Casal, P. (2017). *Estudio acústico y organológico del timple. Parámetros y tipologías*. [Trabajo de Fin de Grado]. Universitat Politècnica de Valencia.

InspiredAcoustics. (s. f.). *MIDI note numbers and center frequencies*. InspiredAcoustics. Recuperado 2 de junio de 2024, de [https://inspiredacoustics.com/en/MIDI\\_note\\_numbers\\_and\\_center\\_frequencies](https://inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies)

Izhaki, R. (2003). *Mixing and audio: Concepts, practices and tools*. (3ªed.). Focal Press.

Izquierdo, P. (2018). *Cómo afinar un timple*. Pedro Izquierdo. <https://www.pedroizquierdo.com/como-afinar-un-timple/>

Matas, A. (2018). Diseño del formato de escalas tipo Likert: Un estado de la cuestión. *Revista electrónica de investigación educativa*, 20(1), 38-47.

Moog, B. (1986). MIDI: Musical Instrument Digital Interface. *Audio Engineering Society Journal*, 34(5).

O’Leary, S., & Griffith, N. J. L. (2003, September -11). *A Hybrid Approach To Timbral Consistency in a Virtual Instrument*. 6ª Int. Conference on Digital Audio Effects (DAFX-03), London, UK.

Owsinski, B. (2007). *The Recording Engineer’s Handbook* (2ª ed.). Course Technology, a Part of Cengage Learning.

Pérez, A. (2024). *Timple para Muscore*. Comunicación Interpersonal (N. Luis Mesa) [E-mail].

Reader’s Digest Association. (1981). Síntesis. En *Gran Diccionario Enciclopédico Ilustrado* (12ª ed., Vol. 11). Reader’s Digest Association.

Rothstein, J. (1995). *MIDI: A comprehensive introduction*. (Vol. 7). AR Editions,

Inc..

Russ, M. (1996). *Sound Synthesis and Sampling* (3<sup>a</sup> ed.). Focal Press.

Viera Ruíz, M. E. (2015). *Artesanía y Tecnología: Aplicación a un instrumento musical: El Timple* [Trabajo de Fin de Grado]. Universidad de Las Palmas de Gran Canaria.

Villacorta, N. L., & Marchiaro, C. A. (2006). *Síntesis digital de sonido en tiempo real por modelado físico de instrumentos virtuales de cuerda percutida* [Trabajo de Fin de Grado]. Universidad Nacional de Rosario.

Yun, Y., & Cha, S. (2013). Designing Virtual Instruments for Computer Music. *International Journal of Multimedia and Ubiquitous Engineering*, 8 (5), 173-178. <https://doi.org/10.14257/ijmue.2013.8.5.16>

## Índice de Figuras

<b>FIGURA 1.</b> TIMPLE CANARIO.....	4
<b>FIGURA 2.</b> PARTES DEL TIMPLE.....	5
<b>FIGURA 3.</b> NUMERACIONES DE LAS CUERDAS DEL TIMPLE.....	6
<b>FIGURA 4.</b> AFINACIÓN DE LAS CUERDAS DEL TIMPLE. ....	6
<b>FIGURA 5.</b> ESQUEMA DE CATALOGACIÓN DEL TIMPLE. ....	7
<b>FIGURA 6.</b> TIMPLES UTILIZADOS PARA EL DESARROLLO PRÁCTICO DEL TRABAJO EN SU FUNDA.....	8
<b>FIGURA 7.</b> TESITURA DEL TIMPLE EN AFINACIÓN GENERAL.. ....	8
<b>FIGURA 8.</b> TÉCNICAS DE SÍNTESIS DIGITAL. ....	13
<b>FIGURA 9.</b> EJEMPLO DE INTERFAZ DE CONTROL.....	14
<b>FIGURA 10.</b> EJEMPLO DE MOTOR DE SÍNTESIS. ....	14
<b>FIGURA 11.</b> DIAGRAMA DE FLUJOS DE SAMPLING. ....	15
<b>FIGURA 12.</b> ESQUEMA GRÁFICO DE CURVA ADSR. ....	19
<b>FIGURA 13.</b> TIMPLES UTILIZADOS PARA LA SÍNTESIS POR PROCESADO DE MUESTRAS EN SU FUNDA.....	21
<b>FIGURA 14.</b> VISTA DEL ESTUDIO DE DOBLAJE DE LA EPSG, SALA DE CONTROL Y CABINA DESDE LA CABINA DE GRABACIÓN.....	21
<b>FIGURA 15.</b> ENRUTADO DE SEÑALES DE AUDIO MEDIANTE CAJA DE CONEXIONES.....	23
<b>FIGURA 16.</b> DISPOSICIÓN DE MICRÓFONOS EN SALA DE GRABACIÓN. ....	23
<b>FIGURA 17.</b> RESPUESTA EN FRECUENCIA DEL MICRÓFONO SC 140 DE THE T-BONE (IZQ.) Y LA CONFIGURACIÓN DE INTERRUPTORES ESCOGIDA (DCHA.).....	24
<b>FIGURA 21.</b> RESPUESTA EN FRECUENCIA DEL MICRÓFONO RETRO TUBE II EN SU PATRÓN DE DIRECTIVIDAD OMNIDIRECCIONAL.....	24
<b>FIGURA 22.</b> MEZCLADOR DE SEÑALES DE AUDIO DE MICRÓFONOS PARA TOMA DE MUESTRAS EN SESIÓN MULTIPISTA DE LOGIC PRO X.....	26
<b>FIGURA 23.</b> IMAGEN DE LA GRABACIÓN DE MUESTRAS EN EL ESTUDIO.....	27
<b>FIGURA 24.</b> DANIEL ARMIJO REALIZANDO AJUSTES DE AFINACIÓN DURANTE TOMA DE MUESTRAS AGUDAS. ....	27
<b>FIGURA 25.</b> COMPARATIVA DE ANTES (ARRIBA) Y DESPUÉS (ABAJO) DEL PROCESO DE EDICIÓN SELECTIVA DE UN EJEMPLO.. ....	28
<b>FIGURA 26.</b> CONFIGURACIÓN DEL ECUALIZADOR PARAMÉTRICO APLICADO A LA PISTA DEL MICRÓFONO DINÁMICO.....	29
<b>FIGURA 27.</b> CONFIGURACIÓN DEL ECUALIZADOR PARAMÉTRICO APLICADO A LA PISTA DEL MICRÓFONO MULTIPATRÓN.....	29
<b>FIGURA 28.</b> EJEMPLOS DE PUERTA DE RUIDO APLICADAS A LOS CANALES DEL MICRÓFONO MULTIPATRÓN (ARRIBA) Y DINÁMICO (ABAJO). ....	30
<b>FIGURA 28.</b> POST-ANALYZER FRECUENCIAL DE LA SEÑAL COMBINADA DE LOS 3 CANALES (IZQ). BALANCE ESTÉREO DE SALIDA DE LA SEÑAL COMINADA DE LOS 3 CANALES (DCHA.) FIGURA DE ELABORACIÓN PROPIA.....	30
<b>FIGURA 29.</b> AJUSTE DE GANANCIAS DE CADA CANAL PARA MEZCLA FINAL DE MUESTRAS.....	31
<b>FIGURA 30.</b> PISTA DE MUESTRAS COMBINADAS DE LA MEZCLA DE MICRÓFONOS. FIGURA DE ELABORACIÓN PROPIA. ....	31
<b>FIGURA 31.</b> PANEL DE CONTROL DE LA HERRAMIENTA "NORMALIZAR GANANCIA DE PASAJE".....	32
<b>FIGURA 32.</b> INTERFAZ MAIZE SAMPLER.. ....	33

<b>FIGURA 33.</b> CONFIGURACIÓN DE PROPIEDADES DEL INSTRUMENTO (I) EN MAIZE SAMPLER. ....	33
<b>FIGURA 34.</b> CONFIGURACIÓN DE PROPIEDADES DEL INSTRUMENTO (II) EN MAIZE SAMPLER. ....	33
<b>FIGURA 35.</b> CONFIGURACIÓN DE PROPIEDADES DEL GRUPO EN MAIZE SAMPLER. ....	34
<b>FIGURA 36.</b> CUADRO DE DIÁLOGO “SAMPLE MAPPING” PARA EL ENRUTADO DE MUESTRAS EN MAIZE SAMPLER. FIGURA DE EJEMPLO DE ENRUTADO DE NOTAS “PIANO”. ....	34
<b>FIGURA 37.</b> DISPOSICIÓN DE MUESTRAS FINALES EN TECLAS Y AMPLITUDES. ....	34
<b>FIGURA 38.</b> EDITOR DE INTERFAZ GRÁFICA DE MAIZE SAMPLER. NOTA: SE PUEDE APRECIAR EL DISEÑO ELABORADO PARA EL PLUGIN.....	35
<b>FIGURA 39.</b> DIAGRAMA DE BLOQUES FUNCIONAMIENTO DEL CÓDIGO DE SÍNTESIS ADITIVA. ....	37
<b>FIGURA 40.</b> ESPECTRO FRECUENCIAL DE LA NOTA MI3 DEL TIMPLE ABRAHAM MARRERO. NOTA: SE REPRESENTA EN AZUL MAGNITUD Y EN ROJO FASE DE LA TRANSFORMADA RÁPIDA DE FOURIER. ....	39
<b>FIGURA 41.</b> GRÁFICA DE ESPECTRO FRECUENCIAL DE NOTA MI3 DE TIMPLE EN LA QUE SE SEÑALAN PARCIALES Y FASES. ....	40
<b>FIGURA 42.</b> GRÁFICA DE ENVOLVENTE EXTRAÍDA DE MUESTRA DE NOTA MI3 DE TIMPLE. ....	41
<b>FIGURA 43.</b> GRÁFICA QUE MUESTRA LA SEÑAL SINTETIZADA DE NOTA MI3 DE TIMPLE RESPECTO AL TIEMPO.. ..	42
<b>FIGURA 44.</b> GRÁFICA QUE MUESTRA LA SEÑAL SINTETIZADA DE LA NOTA MI3 DE TIMPLE CON LA ENVOLVENTE DE AMPLITUD APLICADA FRENTE AL TIEMPO.....	42
<b>FIGURA 45.</b> CAPTURA DE PANTALLA DE LA ENCUESTA DE OPINIÓN REALIZADA PARA LA EVALUACIÓN DE LOS INSTRUMENTOS DESARROLLADOS. ....	44
<b>FIGURA 46.</b> REPRESENTACIÓN GRÁFICA DE LA CALIDAD DE SONIDO PERCIBIDA POR LOS ENCUESTADOS CON RESPECTO AL MODELO EVALUADO.....	45
<b>FIGURA 47.</b> REPRESENTACIÓN GRÁFICA DEL REALISMO DEL SONIDO PERCIBIDA POR LOS ENCUESTADOS CON RESPECTO A LA REPRODUCCIÓN DE DISTINTOS MODELOS DE SÍNTESIS.....	46

## Índice de Tablas

<b>TABLA 1.</b> DESGLOSE DE MATERIAL UTILIZADO PARA TOMA DE MUESTRAS.....	22
<b>TABLA 2.</b> LISTADO DE SOFTWARE UTILIZADO EN LA REALIZACIÓN PRÁCTICA DEL PROYECTO. ....	25
<b>TABLA 3.</b> NIVELES DE NORMALIZACIÓN ESCOGIDOS (PRIMERA OPCIÓN). ....	32
<b>TABLA 4.</b> NIVELES DE NORMALIZACIÓN PARA CADA DINÁMICA TRAS LA CORRECCIÓN. ....	36

## Índice de Ecuaciones

<b>ECUACIÓN 1.</b> TRANSFORMADA DISCRETA DE FOURIER .....	16
<b>ECUACIÓN 2.</b> SUMA DE ONDAS PARA SÍNTESIS DE SONIDO .....	17
<b>ECUACIÓN 3.</b> ECUACIÓN DE SÍNTESIS ADITIVA CON APLICACIÓN DE ENVOLVENTE.....	18
<b>ECUACIÓN 4.</b> ECUACIÓN DE TRANSFORMADA DE HILBERT.....	19
<b>ECUACIÓN 5.</b> ECUACIÓN DE LA SEÑAL ANALÍTICA.....	19
<b>ECUACIÓN 6.</b> ECUACIÓN DE LA SEÑAL ANALÍTICA EN FORMA POLAR.....	19

## Índice de Anexos

<b>Anexo I:</b> Relación del trabajo con los ODS	
<b>Anexo II:</b> Desglose de archivos grabados en la toma de muestras	
<b>Anexo III:</b> Estudio Comparativo velocidad MIDI y nivel de pico	
<b>Anexo IV:</b> Guía de Instalación del <i>plugin</i> “VSTimple” en un DAW	
<b>Anexo V:</b> Demostraciones de funcionamiento de los sintetizadores implementados	
<b>Anexo VI:</b> Código fuente del sintetizador de timple mediante síntesis digital	
<b>Anexo VII:</b> Resultados de encuesta evaluativa	



**Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster**  
Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza				X
ODS 2. Hambre cero				X
ODS 3. Salud y bienestar				X
ODS 4. Educación de calidad	X			
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico				X
ODS 9. Industria, innovación e infraestructuras	X			
ODS 10. Reducción de las desigualdades				X
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables				X
ODS 13. Acción por el clima				X
ODS 14. Vida submarina				X
ODS 15. Vida de ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

**Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto**

**ODS 9: Industria, innovación e infraestructuras.**

El desarrollo de un instrumento virtual para el timble promueve la innovación en la industria musical y la creación de nuevas infraestructuras digitales para la preservación y difusión de la cultura musical tradicional.

**ODS 4: Educación de calidad.**

Este proyecto tiene un impacto significativo en la educación musical, proporcionando una herramienta moderna para el aprendizaje y la enseñanza del timble, facilitando el acceso a este instrumento a estudiantes y músicos de todo el mundo.

## ANEXO II. Desglose de archivos grabados en la toma de muestras.

En las siguientes tablas se pueden observar el desglose de los archivos de proyecto utilizados para grabar las muestras necesarias para la implementación de los instrumentos virtuales.

**Tabla 1.** Desglose de archivos de proyecto de Logic Pro grabados en la toma de muestras del timble de Abraham Marrero.

Nº	Nombre de archivo	Timble	Contenido	Muestras grabadas	Muestras tras edición selectiva
1	AM_Chipude	AM	Canción de muestra	-	-
2	AM_Acordes	AM	Acordes de muestra	-	-
3	AM_Piano	AM	Muestras de cada nota de la tesitura (piano)	82	30
4	AM_Mezzo	AM	Muestras de cada nota de la tesitura (piano)	79	30
5	AM_Forte	AM	Muestras de cada nota de la tesitura (piano)	93	30
6	AM_ArmónicosPiano	AM	Muestras de armónicos (piano)	87	4
7	AM_ArmónicosMezzo	AM	Muestras de armónicos (piano)	68	4
8	AM_ArmónicosForte	AM	Muestras de armónicos (piano)	80	4

*Nota:* AM como acrónimo de Abraham Marrero

**Tabla 2.** Desglose de archivos de proyecto de Logic Pro grabados en la toma de muestras del timble de David Sánchez.

Nº	Nombre de archivo	Timble	Contenido	Muestras grabadas	Muestras tras edición selectiva
1	DS_Chipude	DS	Canción de muestra	-	-
2	DS_Acordes	DS	Acordes de muestra	-	-
3	DS_Piano	DS	Muestras de cada nota de la tesitura (piano)	70	29
4	DS_Mezzo	DS	Muestras de cada nota de la tesitura (piano)	74	29
5	DS_Forte	DS	Muestras de cada nota de la tesitura (piano)	83	29
6	DS_ArmónicosPiano	DS	Muestras de armónicos (piano)	60	4
7	DS_ArmónicosMezzo	DS	Muestras de armónicos (piano)	62	4
8	DS_ArmónicosForte	DS	Muestras de armónicos (piano)	48	4

*Nota:* DS como acrónimo de David Sánchez

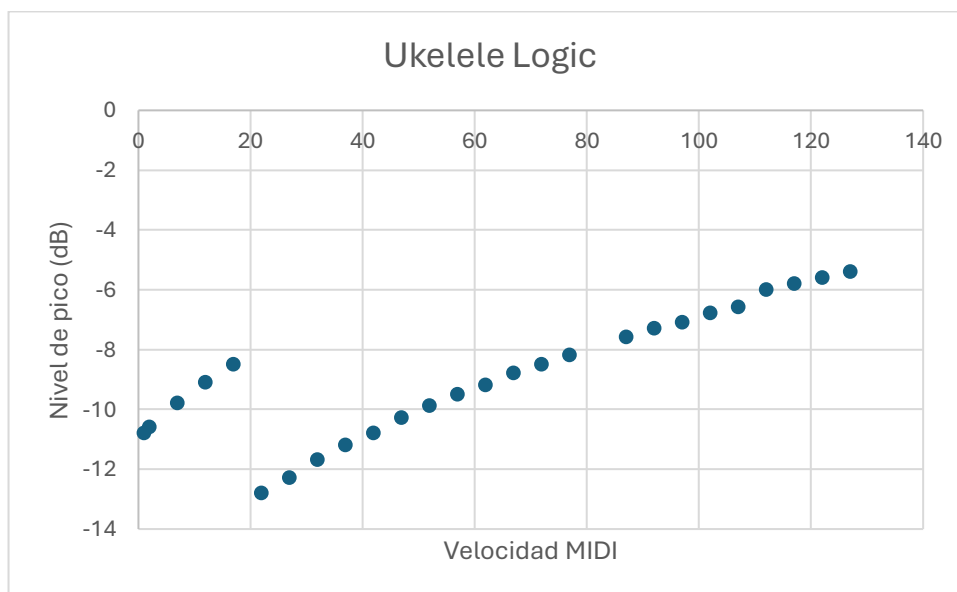
**ANEXO III. Estudio de relación entre nivel de pico (dB)  
y velocidad MIDI en el instrumento de dos instrumentos virtuales.**

Las tablas y gráficas siguientes recogen un estudio realizado para comparar la relación entre nivel de pico y la velocidad MIDI de dos instrumentales virtuales; el *Ukelele* y la *Plucked Nylon Guitar* de Logic Pro.

**Tabla 5.** *Relación entre velocidad MIDI y nivel de pico(dB) del instrumento virtual “Ukelele” del banco de sonidos básico de Logic Pro X.*

Velocidad (1-127)	Nivel de pico (dB)	Velocidad (1-127)	Nivel de pico (dB)	Velocidad (1-127)	Nivel de pico (dB)
1	-10,8	37	-11,2	77	-8,2
2	-10,6	42	-10,8	87	-7,6
7	-9,8	47	-10,3	92	-7,3
12	-9,1	52	-9,9	97	-7,1
17	-8,5	57	-9,5	102	-6,8
22	-12,8	62	-9,2	107	-6,6
27	-12,3	67	-8,8	117	-5,8
32	-11,7	72	-8,5	127	-5,4

*Nota:* Los valores de velocidad son los establecidos por Logic por defecto al cambiar la velocidad con los controles del teclado MIDI.



**Figura 46.** *Gráfico de representación de la comparativa entre el Nivel de pico en dB y la Velocidad MIDI del instrumento virtual “Ukelele” de Logic Pro.*

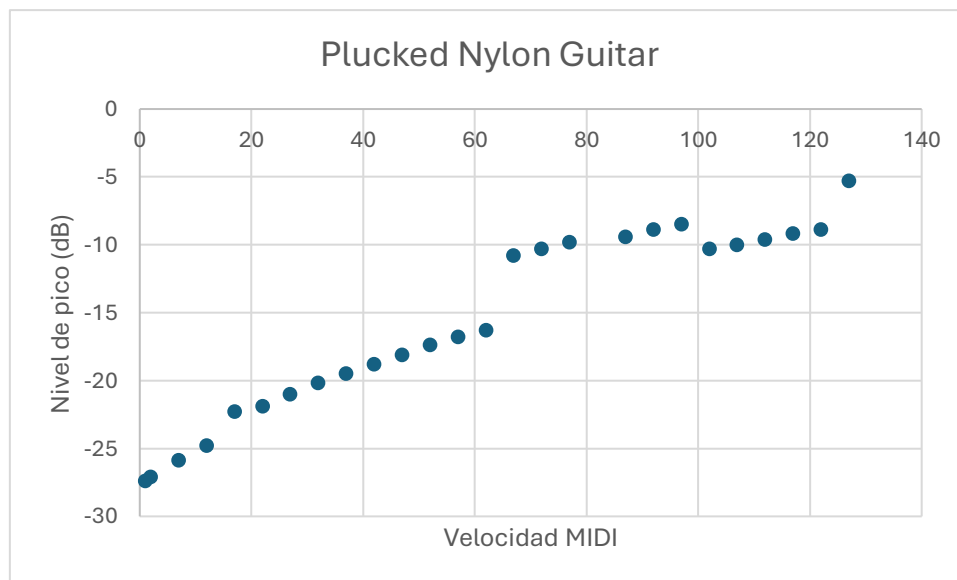
*Nota:* Elaboración propia. Datos extraídos de Tabla 1.



**Tabla 2.** Relación entre velocidad MIDI y nivel de pico (dB) del instrumento virtual “Plucked nylon guitar” del banco de sonidos básico de Logic Pro X.

Velocidad (1-127)	Nivel de pico (dB)	Velocidad (1-127)	Nivel de pico (dB)	Velocidad (1-127)	Nivel de pico (dB)
1	-27,4	37	-19,5	77	-9,8
2	-27,1	42	-18,8	87	-9,4
7	-25,9	47	-18,1	92	-8,9
12	-24,8	52	-17,4	97	-8,5
17	-22,3	57	-16,8	102	-10,3
22	-21,9	62	-16,3	107	-10
27	-21	67	-10,8	117	-9,2
32	-20,2	72	-10,3	127	-5,3

*Nota:* Los valores de velocidad son los establecidos por Logic por defecto al cambiar la velocidad con los controles del teclado MIDI.



**Figura 2.** Gráfico de representación de la comparativa entre el Nivel de pico en dB y la Velocidad MIDI del instrumento virtual “Plucked nylon guitar” de Logic Pro.

*Nota:* Elaboración propia. Datos extraídos de Tabla 1.

## ANEXO IV. Guía de Instalación del *plugin* “VSTimple” en un DAW.

La instalación de un plugin VST desarrollado a partir de Maize Sampler es un proceso sencillo para cualquier ordenador y programa. Para empezar la instalación es común a cualquier DAW, ya que el *plugin* se instala en el ordenador y una vez actualizada la lista de *plugins* aparece en cualquier programa de audio que pueda utilizarlos.

Para instalar el VST en nuestro ordenador debemos tener en cuenta de que sistema operativo se trata. Dependiendo de ello se realizará de una forma u otra.

### macOS

Para ordenadores con sistema operativo *macOS*, el paquete de instalación contiene 3 archivos, “VSTimple.component”, “VSTimple.vst” y “VSTimple.vst3”. Una vez descargados se trasladarán a las carpetas especificadas en la siguiente tabla:

**Tabla 6.** *Desglose de archivos de instalación de “VSTimple” en macOS y carpetas de destino de los mismos.*

Nº	Nombre de archivo	Carpeta de destino
1	VSTimple.component	/Library/Audio/Plug-Ins/Component
2	VSTimple.vst	/Library/Audio/Plug-Ins/VST
3	VSTimple.vst3	/Library/Audio/Plug-Ins/VST3

*Nota:* Copiar el archivo directamente en la carpeta especificada. Se debe elegir entre utilizar vst o vst3.

Una vez instalado el *plugin* se debe instalar el instrumento virtual, en este caso sería el archivo llamado “VSTimple v.1.0.mse”. Este archivo se recomienda instalarlo en la carpeta component en una carpeta dedicada que se puede llamar “instrumentos” para no perder de vista su ubicación. Para su instalación se copia directamente en la carpeta deseada.

Una vez copiados todos los archivos, se procede al ensamblaje del instrumento virtual. Para ello en el DAW deseado se busca el plugin, que en el caso de *macOS* estará en la subcategoría “Instrumentos AU” y se escoge el desarrollador, en este caso “NestorLuisMesa” y el plugin “VSTimple v.1.0.mse”. Una vez abierto el plugin se nos solicita que indiquemos la ubicación del instrumento.

**Figura 1.** *Ejemplo de plugin desconectado del instrumento en el software Reaper.*

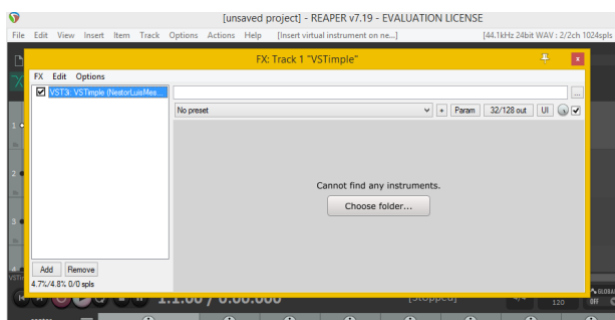


Figura de elaboración propia.

Seleccionamos la carpeta en que se ha pegado el archivo “VSTimple v.1.0.mse” y la instalación habrá finalizado.

## Windows

Para ordenadores con sistema operativo *Windows*, el paquete de instalación contiene 2 archivos, “VSTimple.dll” y “VSTimple.vst3”. Una vez descargados se trasladarán a las carpetas especificadas en la siguiente tabla, escogiendo solo una forma de instalarlo (*dll* o *vst3*):

**Tabla 7.** Desglose de archivos de instalación de “VSTimple” en Windows y carpetas de destino de los mismos.

Nº	Nombre de archivo	Carpeta de destino (Windows 32bits)	Carpeta de destino (Windows 64bits)
1	VSTimple.dll	C:\Windows\System32	C:\Windows\syswow64
2	VSTimple.vst3	C:\Program Files\Common Files\VST3	

*Nota:* Copiar el archivo directamente en la carpeta especificada. Se debe elegir entre utilizar *dll* o *vst3*. Si no existe la carpeta se debe crear con el nombre exacto.

Una vez instalado el *plugin* se debe instalar el instrumento virtual, en este caso sería el archivo llamado “VSTimple v.1.0.mse”. Este archivo se recomienda instalarlo en una carpeta dedicada que se puede llamar “instrumentos” por ejemplo para no perder de vista su ubicación. Para su instalación se copia directamente en la carpeta deseada.

Una vez copiados todos los archivos, se procede al ensamblaje del instrumento virtual. Para ello en el DAW deseado se busca el *plugin*, que en el caso de *macOS* estará en la subcategoría “Instrumentos AU” y se escoge el desarrollador, en este caso “NestorLuisMesa” y el *plugin* “VSTimple”. Una vez abierto el *plugin* se nos solicita que indiquemos la ubicación del instrumento.

**Figura 2.** Ejemplo de *plugin* desconectado del instrumento en el software Reaper.

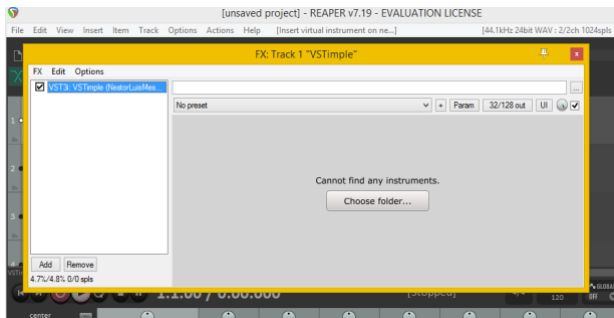


Figura de elaboración propia.

Seleccionamos la carpeta en que se ha pegado el archivo “VSTimple v.1.0.mse” y la instalación habrá finalizado.

## ANEXO V. Demostraciones de funcionamiento de los sintetizadores implementados.

Este anexo contiene una serie de audios con demostraciones del sonido natural del timple, así como de todas las versiones de sintetizadores resultantes de este TFG. Al anexo completo se puede acceder a través de este enlace:

[https://drive.google.com/drive/folders/1i67qXF2i2QxHpJ\\_aPazLQ3Gl6xvk9DPm?usp=sharing](https://drive.google.com/drive/folders/1i67qXF2i2QxHpJ_aPazLQ3Gl6xvk9DPm?usp=sharing)

A continuación, se detalla el contenido de cada fichero de audio que se encuentra contenido en el anexo.

**Audio 1.** *Demostración de sonido de grabación de timple, interpretación de la canción Chipude de José Antonio Ramos por el timplista Daniel Armijo con los timpleros de Abraham Marrero y David Sánchez en ese orden.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

**Audio 2.** *Demostración de funcionamiento del sintetizado mediante reproducción de muestras pregrabadas del timple de Abraham Marrero interpretando la canción Chipude de José Antonio Ramos.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

**Audio 3.** *Demostración de funcionamiento del sintetizado mediante reproducción de muestras pregrabadas del timple de David Sánchez interpretando la canción Chipude de José Antonio Ramos.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

**Audio 4.** *Demostración de funcionamiento del sintetizado síntesis aditiva utilizando Matlab interpretando la canción Chipude de José Antonio Ramos.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

**Audio 5.** *Demostración de funcionamiento del sintetizado síntesis aditiva utilizando Matlab interpretando notas individuales.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

**Audio 6.** *Demostración de funcionamiento del sintetizado síntesis aditiva en su versión mejorada utilizando Matlab interpretando un fragmento de la canción Chipude de José Antonio Ramos.*

*Nota:* Para apreciarlo en su se recomienda escucharlo con auriculares.

## ANEXO VI. Código fuente del sintetizador de timple mediante síntesis digital.

Este anexo recoge los códigos de Matlab implementados para la creación del sintetizador de timple mediante síntesis digital.

### Código fuente 1:

*Código sintetizador de una sola nota a partir de grabación de audio.*

```
%% Listar dispositivos MIDI disponibles

midiDevices = mididevinfo;

if isempty(midiDevices.input)

    error('No se encontraron dispositivos MIDI de entrada.');
```

else

```
    % Mostrar lista de dispositivos MIDI de entrada disponibles

    fprintf('Dispositivos MIDI de entrada disponibles:\n');

    for i = 1:length(midiDevices.input)

        fprintf('%d: %s\n', i, midiDevices.input(i).Name);

    end

    % Solicitar al usuario que seleccione un dispositivo MIDI

    selectedDeviceIndex = input('Seleccione un dispositivo MIDI

(por número): ');

    if selectedDeviceIndex < 1 || selectedDeviceIndex >

length(midiDevices.input)

        error('Selección inválida. Por favor, seleccione un número

válido de la lista.');
```

end

```


    % Seleccionar el dispositivo MIDI elegido por el usuario

    selectedDeviceName =

midiDevices.input(selectedDeviceIndex).Name;

    fprintf('Utilizando el dispositivo MIDI: %s\n',

selectedDeviceName);

    % Crear objeto para recibir mensajes MIDI

    midiIn = mididevice(selectedDeviceName);
```

```

end

%% Crear objeto para la reproducción del sonido

fs = 44100; % Frecuencia de muestreo

audioOut = audioDeviceWriter('SampleRate', fs);

%% Inicializar variables

isNoteOn = false;

note = 0;

frequency = 0;

amplitude = 1;

duration = 1; % Duración de la nota en segundos

%% LECTURA MIDI

fprintf('Presiona una tecla en tu controlador MIDI.\n');

% Función para convertir número de nota MIDI a frecuencia

midiToFreq = @(note) 440 * 2.^((note - 69) / 12);

%bucle infinito esperando notas

while true

    % Leer mensajes MIDI

    midiMsg = midireceive(midiIn);

    % Procesar mensajes MIDI si hay alguno

    if ~isempty(midiMsg)

        for i = 1:length(midiMsg)

            msg = midiMsg(i);

            % Comprobar si el mensaje es Note On

            if strcmp(msg.Type, 'NoteOn') && msg.Velocity > 0

                isNoteOn = true;

                note = msg.Note;

                amplitude = msg.Velocity / 127; % Normalizar la
                velocidad a [0, 1]

                fprintf('Note On: %d, Frequency: %.2f Hz, Velocity:
                %d\n', note, frequency, msg.Velocity);

                % Generar y reproducir sonido sintetizado

                [synthesizedSound] =

```

```

synthesizeSoundFromRecording("1ABH_forte_Estéreo#E3.aif",frequency,amplitude
,duration,fs);

Velocity 0

% Enviar el sonido al dispositivo de audio
    sound(synthesizedSound,fs);

% Comprobar si el mensaje es Note Off o Note On con
elseif strcmp(msg.Type, 'NoteOff') || (strcmp(msg.Type,
'NoteOn') && msg.Velocity == 0)
    isNoteOn = false;
    note = msg.Note;
    fprintf('Note Off: %d\n', note);

% Detener la reproducción si es necesario
    audioOut(zeros(fs * duration, 1));

end end

end

% Pausar un momento para evitar sobrecargar la CPU
pause(0.01);

end

```

*Nota:* Código de elaboración Propia.

### ***Código fuente 2:***

*Script extractAndSaveAudioData para la extracción de la información de muestras de audio.*

```

function extractAndSaveAudioData(audioFolder, outputFile)

% Obtener la lista de archivos en la carpeta especificada
files = dir(fullfile(audioFolder, '*.wav')); % Asumiendo archivos .wav

% Inicializar las variables para almacenar los datos
noteNumbers = [];

maxNoteNumber = 140; % Nota más alta en el rango dado

audioData = cell(1, maxNoteNumber); % Crear una celda para cada nota

sampleRates = NaN(1, maxNoteNumber); % Frecuencias de muestreo para

```

```

cada nota

% Procesar cada archivo

for i = 1:length(files)

    % Obtener el nombre del archivo

    fileName = files(i).name;

    % Extraer el número de nota MIDI del nombre del archivo

    [~, name, ~] = fileparts(fileName); % Obtener el nombre sin la
extensión

    noteNumber = str2double(name(1:end-1)); % Extraer el número de
nota, asumiendo que el número está al principio y es seguido por una letra
(por ejemplo, '64M')

    % Leer el archivo de audio

    [audio, fs] = audioread(fullfile(audioFolder, fileName));

    % Almacenar los datos en la celda correspondiente

    if noteNumber >= 1 && noteNumber <= maxNoteNumber

        audioData{noteNumber} = audio; % Almacenar el audio en la celda
correspondiente a la nota

        sampleRates(noteNumber) = fs; % Almacenar la frecuencia de
muestreo correspondiente a la nota

    else

        noteNumber);

        warning('Número de nota %d fuera del rango esperado.',
end

end

% Guardar los datos en un archivo .mat

save(outputFile, 'noteNumbers', 'audioData', 'sampleRates');

fprintf('Datos guardados en %s\n', outputFile);

end

```

*Nota:* Código de elaboración propia.

***Código fuente 3.***



*Script* processAndStoreFormants para el almacenamiento de la información relevante de los formantes de la señal de audio analizada.

```
function processAndStoreFormants(audioFolder, freqFile, ampFile, phaseFile)

    % Obtener la lista de archivos en la carpeta especificada

    files = dir(fullfile(audioFolder, '*.wav')); % Asumiendo archivos .wav

    % Inicializar variables para almacenar datos

    allFrequencies = [];

    allAmplitudes = [];

    allPhases = [];

    noteNumbers = [];

    % Procesar cada archivo

    for i = 1:length(files)

        % Obtener el nombre del archivo

        fileName = files(i).name;

        % Extraer el número de nota MIDI del nombre del archivo

        [~, name, ~] = fileparts(fileName); % Obtener el nombre sin la
extensión

        noteNumber = str2double(name(1:end-1)); % Extraer el número de
nota, asumiendo que el número está al principio y es seguido por una letra
(por ejemplo, '64M')

        % Verificar si el número de nota es válido

        if isnan(noteNumber)

            warning('Nombre de archivo no válido: %s', fileName);

            continue;

        end

        %% Extraer parciales del archivo de audio

        [frequencies, amplitudes, phases] =
extractPartials(fullfile(audioFolder, fileName), 30, 1000);

        %% Almacenar los datos

        allFrequencies{noteNumber} = frequencies;

        allAmplitudes{noteNumber} = amplitudes;
```

```

    allPhases{noteNumber} = phases;

    noteNumbers = [noteNumbers; noteNumber];

end

%% Guardar los datos en archivos .mat

save(freqFile, 'allFrecuencias', 'noteNumbers');

save(ampFile, 'allAmplitudes', 'noteNumbers');

save(phaseFile, 'allPhases', 'noteNumbers');

%comunicar resultados

fprintf('Datos guardados:\n');

fprintf('Frecuencias en %s\n', freqFile);

fprintf('Amplitudes en %s\n', ampFile);

fprintf('Fases en %s\n', phaseFile);

end

```

*Nota: Código de elaboración propia.*

#### ***Código fuente 4.***

Función extractPartials para la extracción de la información de amplitud, frecuencia y fase de los parciales de una señal de audio.

```

function [frecuencias, amplitudes, phases] = extractPartials(filename, threshold,
minDistance)

    % Leer el archivo de audio

    [y, fs] = audioread(filename);

    % Aplicar la transformada de Fourier

    N = length(y);

    Y = fft(y);

    f = (0:N-1) * (fs / N);

```

```

% Obtener el espectro de magnitudes y fases
magnitudo = abs(Y);
phase = angle(Y);

% Solo considerar la mitad positiva del espectro
magnitudo = magnitudo(1:floor(N/2));
f = f(1:floor(N/2));
phase = phase(1:floor(N/2));

% Identificar los parciales (picos en el espectro)
[pks, locs] = findpeaks(magnitudo, 'MinPeakHeight', threshold, 'MinPeakDistance',
minDistance);

% Extraer frecuencias, amplitudes y fases correspondientes
frecuencias = f(locs);
amplitudes = pks;
amplitudes=amplitudes/max(amplitudes);
phases = phase(locs);

end

```

*Nota:* Código de elaboración propia.

***Código fuente 5:***

Función `extractAndSaveEnvelopes` para la extracción de la información de amplitud, frecuencia y fase de los parciales de una señal de audio.

```

function extractAndSaveEnvelopes(audioFolder, outputFile)

    % Obtener la lista de archivos en la carpeta especificada

    files = dir(fullfile(audioFolder, '*.wav')); % Asumiendo archivos .wav

    % Inicializar variables para almacenar las envolventes

    maxNoteNumber = 92; % Nota más alta en el rango dado

    envelopes = cell(1, maxNoteNumber); % Crear una celda para cada nota

    sampleRates = NaN(1, maxNoteNumber); % Frecuencias de muestreo para cada nota

    % Procesar cada archivo

    for i = 1:length(files)

        % Obtener el nombre del archivo

        fileName = files(i).name;

        % Extraer el número de nota MIDI del nombre del archivo

        [~, name, ~] = fileparts(fileName); % Obtener el nombre sin la extensión

        noteNumber = str2double(name(1:end-1)); % Extraer el número de nota, asumiendo que el número está al
        principio y es seguido por una letra (por ejemplo, '64M')

        % Leer el archivo de audio

        [audio, fs] = audioread(fullfile(audioFolder, fileName));

        % Convertir a mono si es necesario

        if size(audio, 2) > 1

            audio = mean(audio, 2); % Promedio de los canales para convertir a mono

        end

        % Extraer la envolvente usando la transformada de Hilbert

        analyticSignal = hilbert(audio);

        envelope = abs(analyticSignal);

        % Almacenar la envolvente en la celda correspondiente

```

```

if noteNumber >= 1 && noteNumber <= maxNoteNumber

    envelopes{noteNumber} = envelope; % Almacenar la envolvente en la celda correspondiente a la nota

    sampleRates(noteNumber) = fs; % Almacenar la frecuencia de muestreo correspondiente a la nota

else

    warning('Número de nota %d fuera del rango esperado.', noteNumber);

end

end

% Guardar las envolventes en un archivo .mat

save(outputFile, 'envelopes', 'sampleRates');

fprintf('Envolventes guardadas en %s\n', outputFile);

end

```

*Nota:* Código de elaboración propia.

### ***Código fuente 6***

Función `synthesizeSoundFromData` para la síntesis de sonido a partir de los datos necesarios extraídos de las muestras.

```

function [y,frequencias,phases,amplitudes] = synthesizeSoundFromData(note, amplitude, duration, fs, matFile,
amplitudes,phases,frequencias,envelopes)

    % Cargar datos del archivo .mat

    data = load(matFile);

    % Verificar si el número de nota está dentro del rango de datos

    if note >= 1 && note <= length(data.audioData)

        audio = data.audioData{note}; % Obtener el audio correspondiente

        origFs = data.sampleRates(note); % Obtener la frecuencia de muestreo

    else

        error('No se encontraron datos para la nota MIDI %d.', note);

    end

end

```

```

% Convertir a mono si es necesario

if size(audio, 2) > 1
    audio = mean(audio, 2); % Promedio de los canales para convertir a mono
end

t = (0:length(audio)-1) / origFs;

% Re-muestrea el audio si es necesario

if origFs ~= fs
    audio = resample(audio, fs, origFs);
    fprintf('yata.\n')
end

% Extraer envolvente usando la transformada de Hilbert
envelope = envelopes(note);

% Cargar los datos desde archivos .mat

F = frequencies(note);
A = amplitudes(note);
P = phases(note);

% Ajustar la duración del sonido
numSamples = round(duration * fs);
t = (0:numSamples-1) / fs;

% Inicializar la señal sintetizada
y = zeros(numSamples, 1);

% Generar la señal sintetizada usando un rango de frecuencias
for i=1:length(F)
    y = y + A(i)*sin(2*pi*F(i)*t + P(i)); end

% Aplicar la envolvente

% Asegurarse de que la envolvente tiene el tamaño correcto

```

```

if length(envelope) > numSamples
    envelope = envelope(1:numSamples); % Ajustar el tamaño de la envolvente
end

y = y.*envelope;

% Asegurarse de que la señal sea de una sola dimensión (mono)
if size(y, 2) > 1
    y = y(:, 1); % Convertir a mono si es necesario
end

end

```

*Nota:* Código de elaboración propia.

### ***Código fuente 7.***

Código de sintetizado de nota de timple mediante síntesis aditiva a partir de datos extraídos de muestras por las funciones anteriores.

```

%% LIMPIEZA DE VARIABLES

clear

clc

%% VARIABLES

envelopesdata = load('EnvelopesData.mat');
envelopes = envelopesdata.envelopes;

loadedPhaseData = load('phaseData.mat');
phases = loadedPhaseData.allPhases;

loadedAmpData = load('ampData.mat');
amplitudes = loadedAmpData.allAmplitudes;

loadedFreqData = load('freqData.mat');

```

```

frequencies = loadedFreqData.allFrequencies;

matFile="AudioData.mat";

%% Listar dispositivos MIDI disponibles
midiDevices = mididevinfo;
noteNumbers = loadedFreqData.noteNumbers;

if isempty(midiDevices.input)
    error("No se encontraron dispositivos MIDI de entrada.");
else
    % Mostrar lista de dispositivos MIDI de entrada disponibles
    fprintf("Dispositivos MIDI de entrada disponibles:\n");
    for i = 1:length(midiDevices.input)
        fprintf("%d: %s\n", i, midiDevices.input(i).Name);
    end

    % Solicitar al usuario que seleccione un dispositivo MIDI
    selectedDeviceIndex = input("Seleccione un dispositivo MIDI (por número): ");

    if selectedDeviceIndex < 1 || selectedDeviceIndex > length(midiDevices.input)
        error("Selección inválida. Por favor, seleccione un número válido de la lista.");
    end

    % Seleccionar el dispositivo MIDI elegido por el usuario
    selectedDeviceName = midiDevices.input(selectedDeviceIndex).Name;
    fprintf("Utilizando el dispositivo MIDI: %s\n", selectedDeviceName);

    % Crear objeto para recibir mensajes MIDI
    midiln = mididevice(selectedDeviceName);
end

```



```

%% Crear objeto para reproducir sonido

fs = 44100; % Frecuencia de muestreo

audioOut = audioDeviceWriter('SampleRate', fs);

%% Inicializar variables

isNoteOn = false;

note = 0;

frequency = 0;

amplitude = 1;

duration = 1; % Duración de la nota en segundos

fprintf('Presiona una tecla en tu controlador MIDI.\n');

%% Función para convertir número de nota MIDI a frecuencia

midiToFreq = @(note) 440 * 2.^((note - 69) / 12);

while true

    % Leer mensajes MIDI

    midiMsg = midireceive(midiln);

    % Procesar mensajes MIDI si hay alguno

    if ~isempty(midiMsg)

        for i = 1:length(midiMsg)

            msg = midiMsg(i);

            % Comprobar si el mensaje es Note On

            if strcmp(msg.Type, 'NoteOn') && msg.Velocity > 0

                isNoteOn = true;

                note = msg.Note;

                frequency = midiToFreq(note);

                amplitude = msg.Velocity / 127; % Normalizar la velocidad a [0, 1]

```

```

fprintf('Note On: %d, Frequency: %.2f Hz, Velocity: %d\n', note, frequency, msg.Velocity);

% Generar y reproducir sonido sintetizado

[synthesizedSound] = synthesizeSoundFromData(note, amplitud, duration, fs, matFile,
amplitudes,phases,frequencies,envelopes);

% Enviar el sonido al dispositivo de audio

sound(synthesizedSound,fs);

audiowrite('audio5.wav',synthesizedSound, fs);

% Comprobar si el mensaje es Note Off o Note On con Velocity 0

elseif strcmp(msg.Type, 'NoteOff') || (strcmp(msg.Type, 'NoteOn') && msg.Velocity == 0)

isNoteOn = false;

note = msg.Note;

fprintf('Note Off: %d\n', note);

% Detener la reproducción si es necesario

audioOut(zeros(fs * duration, 1));

end

end

end

% Pausar un momento para evitar sobrecargar la CPU

pause(0.01);

end

```

*Nota:* Código de elaboración propia.

### ***Código fuente 8.***

Código de sintetizado de fragmento de notas de timple extraído de archivo .mid mediante síntesis aditiva a partir de datos extraídos de muestras por las funciones anteriores.

```

clear

clc

% Leer archivo MIDI

midiFile = '/Users/nelstorluismesa/Documents/prueba.mid'; % Cambia esto por la ruta a tu archivo MIDI

midiData = readmidi(midiFile);

midiDevices = mididevinfo;

envelopesdata=load('EnvelopesData.mat');

envelopes = envelopesdata.envelopes;

loadedPhaseData = load('phaseData.mat');

phases = loadedPhaseData.allPhases;

loadedAmpData = load('ampData.mat');

amplitudes = loadedAmpData.allAmplitudes;

loadedFreqData = load('freqData.mat');

frequencies = loadedFreqData.allFrequencies;

noteNumbers = loadedFreqData.noteNumbers;

matFile="AudioData.mat";

startTime = 0;

fullAudio = [];

% Extraer notas

notes = midiInfo(midiData, 0);

% Mostrar información de las notas

fprintf('Notas extraídas del archivo MIDI:\n');

fprintf('Canal\tNota\tVelocidad\tInicio\tDuración\n');

for i = 1:size(notes, 1)

    fprintf('%d\t%d\t%d\t%.2f\t%.2f\n', notes(i, 1), notes(i, 3), notes(i, 4), notes(i, 5), notes(i, 6));

```

```

end

% Convertir notas MIDI a frecuencias

midiToFreq = @(note) 440 * 2.^(note - 69) / 12);

% Crear objeto para reproducir sonido

fs = 44100; % Frecuencia de muestreo

audioOut = audioDeviceWriter('SampleRate', fs);

% Generar y concatenar sonidos basados en las notas del archivo MIDI

for i = 1:10

    note = notes(i, 3);

    frequency = midiToFreq(note);

    amplitude = notes(i, 4) / 127; % Normalizar la velocidad a [0, 1]

    duration = notes(i, 6); % Duración de la nota en segundos

    fprintf('Note On: %d, Frequency: %.2f Hz, Velocity: %d\n', note, frequency, notes(i, 4));

    % Generar sonido sintetizado

    [synthesizedSound] = synthesizeSoundFromData(note, amplitude, duration, fs, matFile,
amplitudes, phases, frequencies, envelopes);

    % Crear un vector de ceros para el tiempo de silencio entre notas (si es necesario)

    silenceDuration = notes(i, 5) - startTime; % Duración del silencio antes de la nota actual

    if silenceDuration > 0

        silence = zeros(round(silenceDuration * fs), 1);

        fullAudio = [fullAudio; silence]; % Añadir silencio al audio concatenado

    end

    % Añadir el sonido sintetizado al audio concatenado

    fullAudio = [fullAudio; synthesizedSound];

```

```
% Actualizar el tiempo de inicio para la próxima nota

startTime = notes(i, 5) + duration;

end

outputFileName = 'NotasConcatenadas.wav';

audiowrite(outputFileName, fullAudio, fs);
```

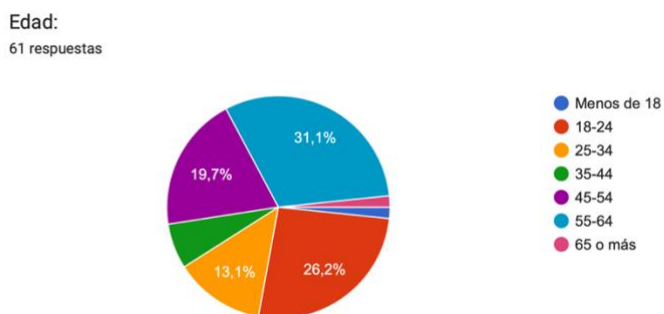
*Nota:* Código de elaboración propia.

## ANEXO VII. Representación gráfica de los resultados de encuesta evaluativa.

### 1.Respuestas:

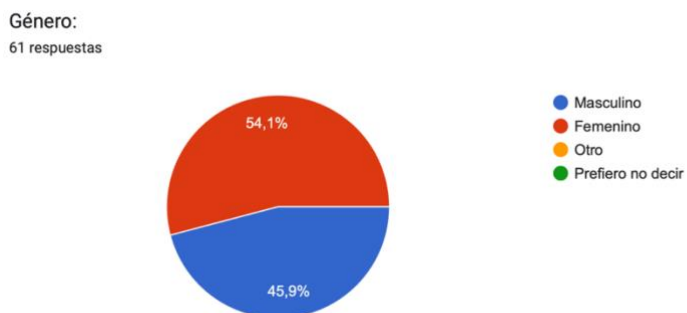
Se han recibido 61 respuestas en el momento de entrega de la presente memoria (29/07/2024).

### 2. Información demográfica:



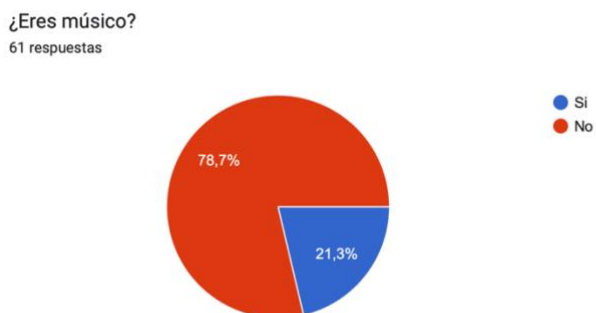
**Figura 8.** Representación gráfica de la edad de los encuestados.

Figura de elaboración propia.



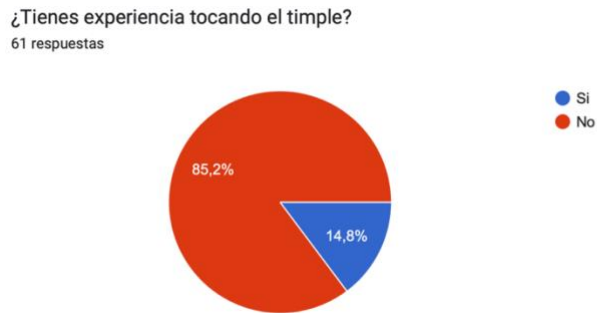
**Figura 2.** Representación gráfica del género de los encuestados.

Figura de elaboración propia.



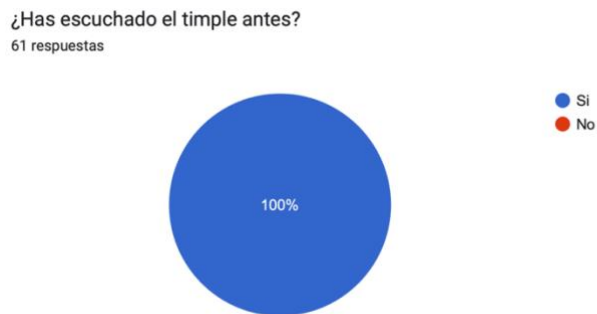
**Figura 3.** Representación gráfica de la experiencia musical de los encuestados.

Figura de elaboración propia.



**Figura 4.** Representación gráfica de la experiencia en el timble de los encuestados.

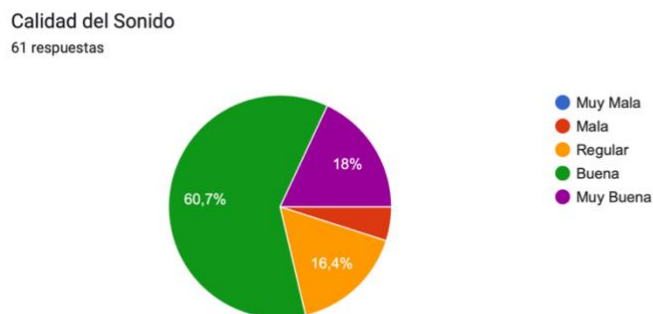
Figura de elaboración propia.



**Figura 5.** Representación gráfica de la experiencia escuchando el timble de los encuestados.

Figura de elaboración propia.

### **3. AUDIO 1**

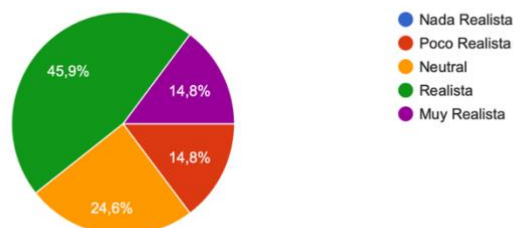


**Figura 6.** Representación gráfica de la calidad de sonido percibida por los encuestados con respecto a la reproducción de muestras pregrabadas del Abraham Marrero.

Figura de elaboración propia.

Realismo del Sonido (¿Qué tan similar te parece al sonido de un timble real?)

61 respuestas

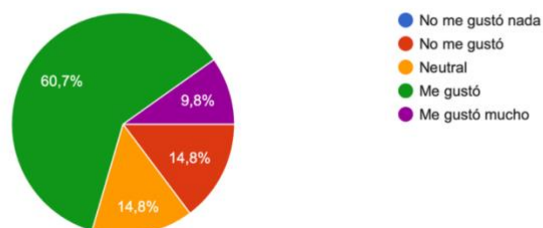


**Figura 7.** Representación gráfica del realismo del sonido percibida por los encuestados con respecto a la reproducción de muestras pregrabadas del Abraham Marrero.

Figura de elaboración propia.

Preferencia Personal (¿Cuánto te gustó este sonido en particular?)

61 respuestas



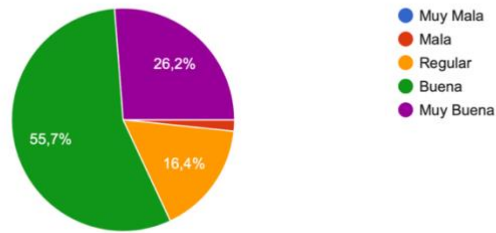
**Figura 8.** Representación gráfica de la preferencia personal de los encuestados con respecto a la reproducción de muestras pregrabadas del Abraham Marrero.

Figura de elaboración propia.

## AUDIO 2



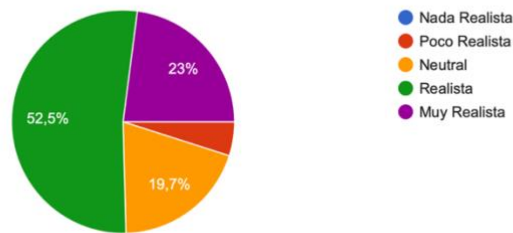
Calidad del Sonido  
61 respuestas



**Figura 9.** Representación gráfica de la calidad de sonido percibida por los encuestados con respecto a la reproducción de muestras pregrabadas del David Sánchez.

Figura de elaboración propia.

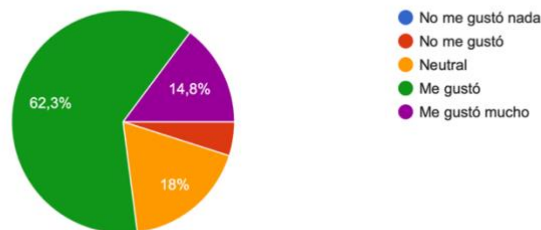
Realismo del Sonido (¿Qué tan similar te parece al sonido de un timble real?)  
61 respuestas



**Figura 10.** Representación gráfica del realismo del sonido percibida por los encuestados con respecto a la reproducción de muestras pregrabadas del David Sánchez.

Figura de elaboración propia.

Preferencia Personal (¿Cuánto te gustó este sonido en particular?)  
61 respuestas

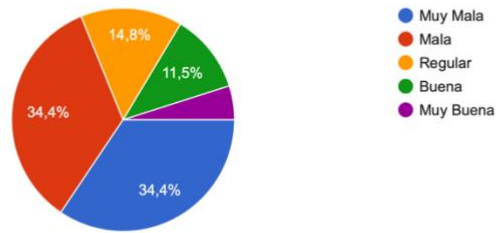


**Figura 11.** Representación gráfica de la preferencia personal de los encuestados con respecto a la reproducción de muestras pregrabadas del David Sánchez.

Figura de elaboración propia.

### **AUDIO 3**

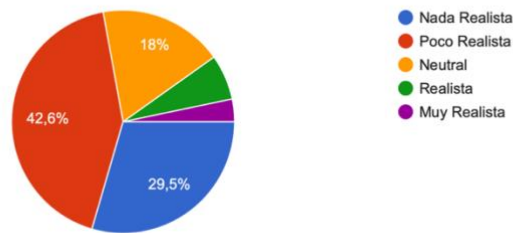
Calidad del Sonido  
61 respuestas



**Figura 12.** Representación gráfica de la calidad de sonido percibida por los encuestados con respecto a la síntesis aditiva implementada.

Figura de elaboración propia.

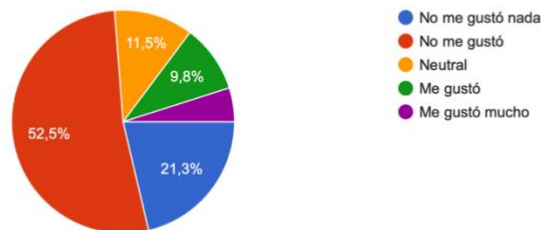
Realismo del Sonido (¿Qué tan similar te parece al sonido de un timble real?)  
61 respuestas



**Figura 13.** Representación gráfica del realismo percibido por los encuestados con respecto a la síntesis aditiva implementada.

Figura de elaboración propia.

Preferencia Personal (¿Cuánto te gustó este sonido en particular?)  
61 respuestas

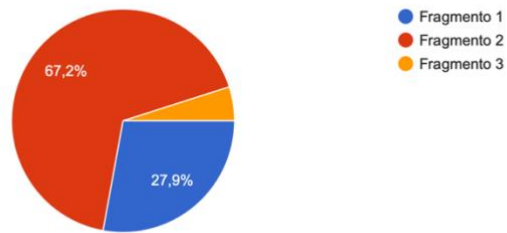


**Figura 14.** Representación gráfica de la preferencia personal de los encuestados con respecto a la síntesis aditiva implementada.

Figura de elaboración propia.

## COMPARATIVA

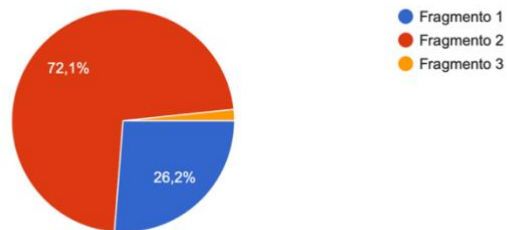
¿Cuál de los tres fragmentos te pareció más realista?  
61 respuestas



**Figura 15.** Representación gráfica de la percepción de los encuestados del fragmento más realista de entre los tres.

Figura de elaboración propia.

¿Cuál de los tres fragmentos te gustó más?  
61 respuestas



**Figura 16.** Representación gráfica de la preferencia personal de los encuestados entre los tres sintetizadores.

Figura de elaboración propia.