



ugr

Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA EN INGENIERIA INFORMÁTICA

Sintetizador virtual

Subtitulo del Proyecto

Autor

Miguel García Tenorio

Directores

Carlos Ureña Almagro

Aquí se puede incluir nombre y logo del
Departamento responsable del proyecto



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

—
Granada, Junio de 2021



ugr

Universidad
de Granada

Alternativamente, el logo de la UGR puede
sustituirse / complementarse con uno específico del

Sintetizador Virtual

Subtítulo del proyecto.

Autor

Miguel García Tenorio

Directores

Carlos Ureña Almagro

Título del Proyecto: Subtítulo del proyecto

Nombre Apellido1 Apellido2 (alumno)

Palabras clave: palabra clave1, palabra clave2, palabra clave3,

Resumen

Poner aquí el resumen.

Project Title: Project Subtitle

First name, Family name (student)

Keywords: Keyword1, Keyword2, Keyword3,

Abstract

Write here the abstract in English.

Yo, **Miguel García Tenorio**, alumno de la titulación TITULACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 75576490P, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Miguel García Tenorio

Granada a X de mes de 201 .

D. **Nombre Apellido1 Apellido2 (tutor1)**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

D. **Nombre Apellido1 Apellido2 (tutor2)**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Título del proyecto, Subtítulo del proyecto***, ha sido realizado bajo su supervisión por **Nombre Apellido1 Apellido2 (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 201 .

Los directores:

Nombre Apellido1 Apellido2 (tutor1)
Apellido2 (tutor2)

Nombre Apellido1 Ape-

Agradecimientos

Poner aquí agradecimientos...

Índice

Contenido

Sintetizador virtual	1
Subtítulo del Proyecto	1
Autor	1
Directores.....	1
Informa.....	5
Subtítulo del proyecto.....	5
Autor	5
Directores.....	5
Título del Proyecto: Subtítulo del proyecto	7
Resumen	7
Project Title: Project Subtitle.....	9
Abstract	9
Informan:	13
Los directores:	13
Índice	17
Capítulo 1: Introducción	18
Capítulo 2: Objetivos.....	18
Capítulo 3: Planificación y presupuesto.....	18
3.1. Planificación Inicial	18
3.2. Presupuesto	20
3.2.1. Justificación del presupuesto	20
Capítulo 4: Análisis	21
4.1. Metodología de desarrollo	21
4.2. Análisis del entorno	21
4.2.1. Análisis competitivo.....	21
Capítulo 5: Diseño.....	23
Capítulo 5: Implementación y pruebas	23
5.1. Tecnologías seleccionadas.....	23
5.1.1. Análisis de tecnologías.....	23
5.1.2. Conclusión del análisis de tecnologías	25
Capítulo 6: Conclusiones y vías futuras	25
Capítulo 7: Bibliografía final	25

Capítulo 1: Introducción

Capítulo 2: Objetivos

Capítulo 3: Planificación y presupuesto

3.1. Planificación Inicial

Inicialmente se realiza una planificación en la que se opta por dividir el desarrollo del proyecto en 6 fases o iteraciones, las cuales se dividen en 2 Sprint de dos semanas, excepto la primera iteración y la última que contienen un Sprint, en la que se planifican 8 bloques principales que conllevarán una serie de tareas a realizar.

Se realiza esta división, debido a que se seguirá una metodología ágil de desarrollo, Scrum, que será expuesta en secciones posteriores (ver ...).

El proyecto comienza la semana del 15/02/2021 y termina la semana del 5/07/2021, por lo que se planifican 20 semanas en total. La planificación consta de 8 bloques principales. La [figura 3.1](#) muestra la temporización de la realización de cada uno de los bloques de acuerdo a esta división



Figura 3.1. Diagrama Gantt planificación inicial

De acuerdo con esta planificación inicial se estiman las horas que será necesarias para la realización de cada uno de los bloques. De acuerdo con el Plan de Ordenación Docente del curso 2020/201 de la UGR 1 crédito ECTS se corresponde con 25 horas de trabajo, por lo que 12 créditos ECTS del Trabajo Final de Grado se corresponden con 300 horas de trabajo que hay que planificar. En la [tabla 3.1](#) se muestra la correspondencia en horas para cada uno de los bloques planificados.

	HORAS	PORCENTAJE TOTAL
Planificación	22	7,3
Formación	21	7
Investigación	7	2,3
Análisis	40	13,3
Diseño	41	13,6
Documentación	32,5	10,83
Implementación	109	36,3
Revisión y Pruebas	27,5	9,17
TOTAL	300	100

Tabla 3.1. Horas planificadas

3.2. Presupuesto

Se estima que el precio del proyecto ronde los 5490€

Elementos	Coste	Total
Ordenador	780 €	780 €
Teclado MIDI	100 €	100 €
Auriculares estudio	60 €	60 €
Espacio de coworking (4 meses)	100 € / mes	400 €
Trabajo autónomo (4 meses)	13,5 € / hora	4050 €
Bibliografía	100 €	100 €
Total		5490 €

Tabla 3.2. Presupuesto del proyecto

3.2.1. Justificación del presupuesto

A continuación, se justifica cada elemento del presupuesto:

- **Ordenador:** Será necesario un ordenador de gama media/alta ya que se requiere de capacidad computacional para ejecutar el entorno de programación, lanzar el servidor para realizar pruebas y ejecutar el programa.
- **Teclado MIDI:** Se requiere un teclado MIDI para probar alguna de la

funcionalidad que ofrece el sistema. Basta con un teclado MIDI de gama media/baja para realizar las pruebas que rondan entorno a los 100 €

- **Auriculares de estudio:** Es preciso comprar unos auriculares de estudio de gama media para tener un sonido claro y limpio de los sonidos producidos por el software que se esta desarrollando. Este tipo de auriculares rondan los 60€ en el mercado.
- **Espacio de coworking:** Se supone que no se dispone de espacio de trabajo, por lo que se opta por alquilar una plaza en un espacio coworking en la que se dispondrá de electricidad, agua, calefacción, internet y sala de reuniones entre otros. Aproximadamente la tarifa ronda entre los 100 € mensuales.
- **Trabajo autónomo:** Como se planificaron 300 horas de trabajo (ver [Tabla 3.1](#)), y el salario de un Ingeniero informático Junior ronda los 13,5 € la hora y el proyecto tiene una duración de 4 meses, se estiman unos 4050 € de trabajo autónomo.
- **Bibliografía:** Serán necesarias fuentes que no se encuentran en Internet y no se encuentran en bibliotecas públicas (salvo en la de la UGR) que conllevarán un coste adicional.

Capítulo 4: Análisis

4.1. Metodología de desarrollo

4.2. Análisis del entorno

Antes de entrar en el análisis de los requerimientos del sistema o las historias de usuario, es necesario analizar el entorno para conocer cómo se comportan los usuarios al utilizar un sintetizador virtual y lograr un software más adaptado a los usuarios.

4.2.1. Análisis competitivo

En primer lugar, se realiza un análisis de la competencia para encontrar puntos fuertes y débiles de estos sistemas.

Antes de empezar con el análisis competitivo, hay que destacar que los softwares de síntesis digital más populares están diseñados para ser extensiones de programas DAW, por lo que muchos de estos no se pueden utilizar sin un programa de este tipo, o bien se pueden utilizar, pero con una funcionalidad limitada. A pesar de esto, podemos encontrar otras opciones que no necesitan de un DAW para su funcionamiento.

De esta manera, podemos clasificar los sintetizadores en dos grandes bloques:

- Aquellos que necesitan de un DAW para su utilización
- Aquellos que no necesitan un DAW para su utilización.

De acuerdo con esta clasificación, para cada grupo, se analiza el sintetizador más popular:

- **Serum:**
 - Esta dentro del primer grupo (no necesita un DAW)
 - Diseñado por la empresa Xfer Records

- Implementado en C++ usando JUCE ¹ como framework
- Puntos fuertes
 - En cuanto a síntesis digital, es de los mas sofisticados en el mercado
 - Tiene dos osciladores
 - Interfaz bastante sencilla e intuitiva
 - Posibilidad de elegir distintos tipos de onda
 - Ofrece la posibilidad de modificar la onda de audio de cada oscilador a nuestro antojo
 - Sistema de modulación bastante versátil, permite arrastrar el componente de modulación a lo que se quiera modular e incluso cuenta con una matriz de modulación
 - Permite guardar configuraciones de sonidos
 - Posee un banco de configuraciones preestablecidas, es decir, de sonido ya diseñados que se pueden buscar y filtrar por categorías u otros criterios
 - Cuenta con gran variedad de efectos que se pueden aplicar a la vez a un mismo sonido
 - En la parte inferior de la interfaz hay una simulación de un piano que nos va indicando que nota estamos pulsando encada momento
 - Posee gráficos que se sincronizan con el sonido producido
 - Permite insertar muestras de sonido (simples) para su sintesis
- Puntos débiles
 - Solo se puede utilizar a través de un DAW, es decir, es un plugin
 - Tiene bastantes parámetros que se pueden modificar por lo que se requiere un conocimiento alto en síntesis digital
 - Debido a que es un plugin sus funciones se limitan a crear sonidos y enviarlos al DAW.
 - Elevado coste, actualmente \$189 USD

- **Midi.city:**

- Esta dentro del segundo grupo (no necesita un DAW)
- Es un sintetizador en la nube, que puede usarse en cualquier dispositivo que disponga de un navegador web
- Constituye una comunidad, en la que puedes crearte una cuenta para participar en foros, votar para nuevas características, acceder al registro de cambios, reportar algún fallo etc....
- Principalmente se basa en JavaScript usando ToneJS como Framework
- Puntos fuertes:
 - Se puede utilizar en cualquier dispositivo con navegador web
 - Es open source, su utilización no requiere comprar ninguna licencia
 - Posee todas las octavas
 - Posee gráficos que se sincronizan con el sonido producido
 - Los sonidos generados se pueden reproducir, con el teclado, pulsando con el ratón en la nota o conectando un teclado midi.
 - Cuenta con un metrónomo
 - Es capaz de secuenciar archivos midi
 - Podemos reproducir una secuencia de percusión a la vez que reproducimos los sonidos sintetizados

¹ JUCE: Framework basado en C++ para el desarrollo de Plugins para DAWS

- Cuanta con un banco de sonidos bien clasificados
- Puntos débiles:
 - No se pueden crear nuevos sonidos, los sonidos se limitan a los que vienen por defecto
 - No se pueden grabar piezas musicales tocadas con los sonidos sintetizados
 - No se pueden modificar los parámetros de los sonidos, la síntesis es invisible al usuario se realiza en un segundo plano.
 - En ocasiones presenta latencia

Capítulo 5: Diseño

Capítulo 5: Implementación y pruebas

5.1. Tecnologías seleccionadas

5.1.1. Análisis de tecnologías

Para llevar a cabo la implementación del sistema software que se requiere, se necesita una tecnología que nos ofrezca lo siguiente:

- Soluciones para desarrollar un sistema de tiempo real (multiprocesamiento, temporizadores, semáforos, señales de tiempo real, entrada/salida síncrona y asíncrona etc..)
- Métodos y variables para comunicarnos con el sistema de audio del computador
- Métodos eficientes para el procesamiento de señales digitales (DSP)
- Documentación para el tratamiento de audio
- Facilidades para implementar una interfaz de usuario
- Facilidades para implementar gráficos complejos

Múltiples lenguajes de programación como Java o Python, ofrecen parcialmente soluciones a los puntos comentados, pero debido a la complejidad del procesamiento de señales digitales es preciso utilizar Frameworks y librerías nativas que simplifiquen esta tarea.

Existen dos lenguajes de programación que nos proporcionan lo expuesto en este punto:

- C++
- JavaScript

A continuación, se describen las ventajas e inconvenientes de estos lenguajes

5.1.1.1. C++

Ventajas

- Dispone de librerías para trabajar con el tiempo, como chrono.
- Ofrece soluciones bastante eficientes para el multiprocesamiento y la sincronización.
- Dispone de librerías para comunicarnos con el sistema de audio del computador.
- Dispone de librerías DSP, como IIPP de Intel.
- La gran mayoría de sistemas software para el procesamiento de audio están implementados en C++ o se basan en este.
- Existen librerías para el procesamiento de gráficos, como Open GL .
- Existen múltiples Frameworks para el desarrollo de interfaces.
- Permite aprovechar al máximo los recursos del ordenador.

Desventajas

- No es portable en nuestro sistema a realizar, necesitaremos realizar una versión para cada SO ya que el sonido se trata de distinta forma en cada uno de estos.
- Los Frameworks existentes no dan soporte a la solución que se busca, ya que están muy limitados a la realización de plugins para DAWS existentes. Con estos Frameworks solo podremos usar nuestro sistema software dentro de un DAW y no podremos realizar una interfaz desde cero.
- Si se usa esta tecnología, el resultado final será de bajo nivel por lo que la dificultad y el tiempo para realizarlo se eleva con creces.
- Debido a la ausencia de Frameworks específicos, se requiere demasiado conocimiento en cuestiones matemáticas y físicas para el tratamiento de las señales de audio digital.

5.1.1.2. JavaScript

Ventajas

- Portable, si se utiliza esta tecnología el software estará disponible para cualquier dispositivo con un navegador.
- Existe un Framework específico, llamado Tonejs, destinado a la creación de sonido digital en el ámbito del navegador, el cual se ajusta bastante bien al sistema a desarrollar, ya que ofrece soluciones a alto nivel de síntesis digital que no requerirían de un excesivo conocimiento en cuestiones matemáticas y físicas. Además, da soporte a la comunicación con el sistema de audio del computador.
- Existen Frameworks para trabajar con gráficos como Three js .
- Existen Frameworks que proporcionan herramientas para el desarrollo de una aplicación en la nube como React.
- En general, hay una alta disponibilidad de librerías.

Desventajas

- Las capacidades de tiempo real son más limitadas, los callbacks no están sincronizadas con precisión. Tonejs es capaz de solucionar este problema.
- El ámbito de ejecución, se limita al navegador, por lo que los recursos estarán limitados a este.
- No existen muchas alternativas a Tonejs.
- Poco software de audio utilizando este lenguaje.
- Da problemas para programas no triviales, al no ser fuertemente tipado.

5.1.2. Conclusión del análisis de tecnologías

Sopesadas las ventajas y desventajas de las tecnologías analizadas en el punto anterior, se llegan a las siguientes conclusiones:

1. Pese a que C++, ofrece más capacidades de tiempo real que JavaScript, Tonejs soluciona el problema de la sincronización y el tiempo real de manera sencilla por lo que se podría llegar a alcanzar el grado estas capacidades usando este Framework
2. El grado de dificultad en la implementación de la interfaz en JavaScript es más bajo
3. La ventaja de la total e inmediata disponibilidad de JavaScript aporta un gran valor añadido al producto final.
4. El Framework Tonejs ofrece muchas soluciones a un nivel más alto que C++, permitiendo que la implementación de este sea más asequible
5. Debido a la alta disponibilidad de librerías para JavaScript, podremos encontrar soporte y soluciones para gran variedad de los requisitos del programa.

A partir de estas primeras conclusiones, se llega a la conclusión final:

Se elige JavaScript como lenguaje de programación para la implementación del sintetizador, utilizando como Frameworks **Tonejs**, para trabajar con las señales de audio, **React** para el Frontend y **Express** para el Backend.

Capítulo 6: Conclusiones y vías futuras

Capítulo 7: Bibliografía final

<https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/ipp.html>
|

<https://xferrecords.com/products/serum> -SERUM

<https://midi.city/> Midi.city

<https://tonejs.github.io/> - Tonejs

<https://es.reactjs.org/> react

<https://expressjs.com/es/> express