



ugr

Universidad  
de Granada

TRABAJO FIN DE GRADO  
GRADO DE INGENIERÍA EN INFORMÁTICA

# Aplicación multiplataforma para el aprendizaje del lenguaje musical

---

Subtitulo del Proyecto

**Autor**

Sergio Hervás Cobo

**Directores**

Luis López Escudero

Germán Arroyo Moreno



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, mes de 201









# Aplicación multiplataforma para el aprendizaje del lenguaje musical

---

Subtítulo del proyecto.

## **Autor**

Sergio Hervás Cobo

## **Directores**

Luis López Escudero

Germán Arroyo Moreno



# **Aplicación multiplataforma para el aprendizaje del lenguaje musical: Subtítulo del proyecto**

Sergio Hervás Cobo

**Palabras clave:** palabra\_clave1, palabra\_clave2, palabra\_clave3, .....

## **Resumen**

Poner aquí el resumen.





# **Project Title: Project Subtitle**

Sergio Hervás Cobo

**Keywords:** Keyword1, Keyword2, Keyword3, ....

## **Abstract**

Write here the abstract in English.



---

Yo, **Nombre Apellido1 Apellido2**, alumno de la titulación TITULACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI XXXXXXXXXX, 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: Nombre Apellido1 Apellido2

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)**      **Nombre Apellido1 Apellido2 (tutor2)**



# Agradecimientos

Poner aquí agradecimientos...





# Índice general

|   |           |
|---|-----------|
| <b>1. Introducción</b>                                  | <b>1</b>  |
| 1.1. Motivacion . . . . .                               | 2         |
| 1.2. Objetivos . . . . .                                | 3         |
| <b>2. Estado del Arte</b>                               | <b>5</b>  |
| 2.1. Software a desarrollar . . . . .                   | 5         |
| 2.1.1. Aplicaciones de aprendizaje . . . . .            | 5         |
| 2.1.2. Aplicaciones de aprendizaje musical . . . . .    | 10        |
| 2.2. Desarrollo de Software . . . . .                   | 12        |
| 2.2.1. Flutter . . . . .                                | 12        |
| 2.2.2. React Native . . . . .                           | 12        |
| 2.2.3. Dart . . . . .                                   | 13        |
| 2.2.4. Kotlin . . . . .                                 | 13        |
| 2.2.5. NodeJS . . . . .                                 | 14        |
| 2.2.6. React . . . . .                                  | 14        |
| 2.2.7. Angular . . . . .                                | 15        |
| 2.2.8. MongoDB . . . . .                                | 15        |
| 2.2.9. Pilas MERN & MEAN . . . . .                      | 16        |
| 2.2.10. MariaDB . . . . .                               | 16        |
| <b>3. Especificación de requisitos</b>                  | <b>19</b> |
| 3.1. Introducción . . . . .                             | 19        |
| 3.1.1. Propósito . . . . .                              | 19        |
| 3.1.2. Ámbito del Sistema . . . . .                     | 19        |
| 3.1.3. Definiciones, Acrónimos y Abreviaturas . . . . . | 19        |
| 3.1.4. Referencias . . . . .                            | 20        |
| 3.1.5. Visión General del Documento . . . . .           | 20        |
| 3.2. Descripción General . . . . .                      | 20        |
| 3.2.1. Perspectiva del producto . . . . .               | 20        |
| 3.2.2. Funciones del producto . . . . .                 | 20        |
| 3.2.3. Características de los usuarios . . . . .        | 20        |
| 3.2.4. Restricciones . . . . .                          | 20        |
| 3.2.5. Suposiciones y dependencias . . . . .            | 20        |

|   |           |
|---|-----------|
| 3.3. Requisitos Específicos . . . . .           | 20        |
| 3.3.1. Requisitos funcionales . . . . .         | 20        |
| 3.3.2. Requisitos no funcionales . . . . .      | 20        |
| 3.3.3. Requisitos de información . . . . .      | 20        |
| 3.3.4. Bocetos de interfaz de usuario . . . . . | 20        |
| <b>4. Planificación</b>                         | <b>21</b> |
| <b>5. Análisis del problema</b>                 | <b>23</b> |
| <b>6. Diseño</b>                                | <b>25</b> |
| <b>7. Implementación</b>                        | <b>27</b> |
| <b>8. Pruebas</b>                               | <b>29</b> |
| <b>9. Conclusiones</b>                          | <b>31</b> |

# Índice de figuras

|  |    |
|--|----|
| 2.1. Logo de Duolingo . . . . .  | 5  |
| 2.2. Ejercicio de Duolingo donde el usuario debe adivinar qué palabra corresponde al audio . . . . .                           | 6  |
| 2.3. Menú de selección de las lecciones de Artly clasificadas en movimientos artísticos . . . . .                              | 7  |
| 2.4. Ciudad de BMath donde se seleccionan los niveles y se construyen los edificios para mostrar el progreso del usuario . . . | 8  |
| 2.5. Ejercicio de BMath sobre cálculo matemático (sumas y restas) con calculadora, con papel y mentalmente . . . . .           | 9  |
| 2.6. Lista de lecciones que ofrece la aplicación Curso de Lenguaje Musical . . . . .   | 10 |
| 2.7. Logo de Flutter . . . . .   | 12 |
| 2.8. Logo de React Native . . . . .  | 12 |
| 2.9. Logo de Dart . . . . .  | 13 |
| 2.10. Logo de MongoDB . . . . .  | 15 |
| 2.11. Pilas MEAN y MERN . . . . .  | 16 |
| 2.12. Logo de MariaDB . . . . .  | 16 |



# Capítulo 1

## Introducción

Durante toda la historia de la humanidad, la música ha sido siempre una parte muy importante de la vida de la mayoría de personas y, a día de hoy, es una pieza clave y fundamental en nuestra sociedad y tradiciones. Escuchamos canciones a todas horas: mientras andamos por la calle, cuando cocinamos, mientras hacemos ejercicio, conduciendo en el coche, etc. La música está presente en gran parte de nuestros hábitos y de nuestra cultura y es por esto que, además de ser una de las siete bellas artes, mucha gente se dedica a estudiarla y a aprender sobre esta.

Sin embargo, por desgracia, a lo largo de la historia el aprendizaje del lenguaje musical ha sido elitista y la mayoría de personas con pocos recursos económicos no ha podido acceder a gran parte de este conocimiento, ya que antiguamente solo las familias nobles y adineradas podían darse el lujo de disfrutar de las melodías clásicas. Puede parecer que no pero, incluso en la actualidad, la formación musical a veces puede ser vista como un privilegio reservado para unos pocos, pues solo quienes pueden permitirse ir a una escuela de música o conservatorio pueden adquirir esos conocimientos.

Por otro lado, pese a ser un aspecto fundamental en nuestras vidas, el aprendizaje de esta disciplina es a menudo visto como algo aburrido debido a clases desactualizadas, poco motivadoras y enfocadas únicamente a un género musical en específico: la música clásica.

Todo esto plantea las siguientes cuestiones, ¿es posible democratizar el aprendizaje musical? ¿Ha habido un cambio en la forma en la que se estudia y/o practica música con la aparición de los dispositivos móviles, de internet y de las aplicaciones software? ¿Es posible aprender música de forma autodidacta, divertida y gratuita?

## 1.1. Motivacion

El surgimiento de nuevas aplicaciones que ofrecen formas de aprendizaje lúdicas y divertidas en distintos ámbitos como idiomas, matemáticas, arte, geografía, etc, me ha hecho pensar en lo útiles que pueden llegar a ser estas para personas con pocos recursos que buscan una forma rápida, barata y sencilla de aprender sobre algún tema en específico. Además, el acceso a la música es algo fundamental que todas las personas deberían tener al alcance y siempre lo he creído así.

En cuanto al aprendizaje musical, al estar relacionado con la música clásica y tradicional, suele ser visto como algo aburrido, por lo que muchos no se llegan a interesar por esto o suelen abandonar su formación en mitad. Sin infravalorar las clases de música tradicionales, el mundo está cambiando y existe una sensación de que la docencia musical se ha quedado atrás y no se ha adaptado a las necesidades de los jóvenes ni de la sociedad actual. Habiendo pocas opciones que ofrezcan un aprendizaje de la música de forma divertida y lúdica, muchas personas pierden una oportunidad valiosa de llegar a un mundo muy valioso.

Por otro lado, mi experiencia estudiando en una escuela de música durante 6 años me ha hecho entender que el aprendizaje musical, pese a ser un proceso complejo y exigente, es gratificante y satisfactorio; y que cualquier persona, tenga o no recursos económicos, debería tener la oportunidad de poder aprender música y disfrutar de ella sin impedimentos. La música y la formación en esta debería ser accesible para cualquier persona.

Al unir el surgimiento de distintas aplicaciones software educativas que ofrecen una visión novedosa y entretenida del aprendizaje con mi experiencia en el conservatorio de música, se me ocurrió la idea de desarrollar un sistema de aprendizaje musical que permita a cualquier persona, independientemente de su nivel original de conocimientos, aprender conceptos básicos e intermedios sobre lenguaje musical, así como la posibilidad de instruirse para empezar a tocar instrumentos musicales con un enfoque didáctico, divertido y progresivo.

Todo esto y la falta de aplicaciones similares y de calidad en el mercado respecto al conocimiento musical, han convertido en una necesidad el desarrollo de este proyecto.

## 1.2. Objetivos

Se pretende desarrollar una aplicación software que permita al usuario aprender conceptos de lenguaje musical de forma autodidacta y sencilla, proporcionando un enfoque que anime a las personas a involucrarse en la adquisición de conocimientos y a entender la mayoría de conceptos que la envuelven, desde el lenguaje musical hasta la práctica de un instrumento.

También se desea un sistema que proporcione una metodología de enseñanza mediante gamificación; es decir, con contenidos lúdicos y divertidos y con un progreso organizado en logros y niveles que permita a los usuarios sentirse satisfechos con su progreso y motivados a seguir aprendiendo. Los usuarios podrán ver su progreso mediante un sistema de cuentas de usuario que permitirá también la posibilidad de compartir sus logros con otros usuarios.

Por último, otro de los objetivos será incorporar distintos tipos de ejercicios y actividades que faciliten este aprendizaje, como por ejemplo la inclusión de un sistema que detecte las notas musicales tocadas por el usuario o la incorporación de un teclado virtual simple y sencillo que permita al usuario practicar y aprender a tocarlo.





## Capítulo 2

# Estado del Arte

### 2.1. Software a desarrollar

Para tener una perspectiva más amplia y clara de lo que se pretende desarrollar, se ha realizado una búsqueda de software similar al de este proyecto, tanto de aprendizaje musical como de otros temas que utilicen las técnicas de gamificación y de aprendizaje lúdico.

#### 2.1.1. Aplicaciones de aprendizaje

En la actualidad existen numerosas aplicaciones en el mercado que ayudan al aprendizaje de conceptos y a la adquisición de conocimientos sobre distintos temas, como los idiomas, las matemáticas o el arte, entre otros. Es tan grande el auge de este tipo de herramientas que muchas de ellas están siendo recomendadas por el colectivo docente como apoyo a los contenidos que se dan en clase. A continuación se detallarán las aplicaciones más destacadas encontradas:

##### Duolingo

Duolingo es una de las aplicaciones más populares que existen para aprender idiomas. Esta aplicación se basa en el método de aprendizaje por inmersión y en la gamificación: el usuario estudia el idioma mediante una serie de actividades lúdicas. Aunque su principal motivación es la enseñanza de hablar y escribir en un idioma, también se incluyen actividades de comprensión auditiva, de lectura y de vocabulario.



Figura 2.1: Logo de Duolingo

La aplicación se divide en lecciones, las cuales contienen una serie de ejercicios (Figura 2.2) que el usuario debe completar para ir avanzando por las distintas secciones.

Algunos de los ejercicios que presenta la aplicación son:

- Traducción de palabras o frases con el teclado
- Traducción seleccionando bloques de palabras
- Pronunciación de palabras o frases mediante el micrófono del dispositivo
- Selección de imagen para el vocabulario

Además, Duolingo ofrece en cada lección, antes de los ejercicios, una breve explicación de la gramática o vocabulario con imágenes y definiciones sencillas, que ayudarán al usuario a entender el temario antes de comenzar la evaluación de la lección. En cuanto al diseño, la herramienta ofrece una interfaz sencilla y fácil de usar, con botones intuitivos y coloridos que mejoran la experiencia de usuario, algo primordial en este tipo de aplicaciones. Por último, cabe destacar que esta aplicación es gratuita (aunque ofrece funcionalidad adicional de pago dentro de esta) y está disponible para dispositivos móviles (Android, iOS y Windows Phone).

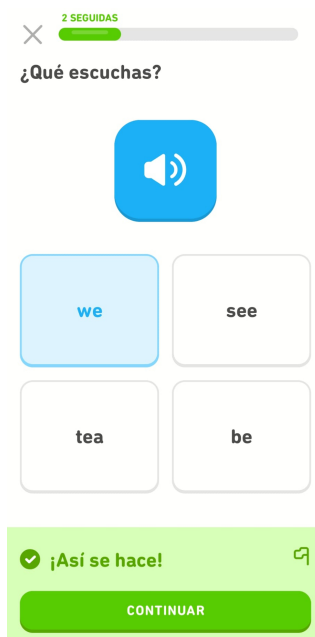


Figura 2.2: Ejercicio de Duolingo donde el usuario debe adivinar qué palabra corresponde al audio

## Artly

Artly es una aplicación móvil que permite aprender cientos de obras de arte de forma interactiva. De forma parecida a Duolingo, cuando el usuario abre la aplicación, se encuentra una lista de secciones (Figura 2.3) correspondientes a los distintos movimientos artísticos. Estos apartados se irán desbloqueando uno a uno a medida que el usuario vaya completando las lecciones y ejercicios.

En cada uno de los apartados el usuario podrá ver una lista de cuadros y esculturas de los distintos artistas que pertenecen a ese movimiento. Al seleccionar uno de ellos, se abrirá una pantalla con la imagen del cuadro o escultura, en la que el usuario podrá ver la obra más de cerca, junto con el título, el autor y una descripción. Artly incorpora además una serie de preguntas sobre las obras vistas en dicho apartado (seleccionar el autor, seleccionar el título de la obra...). Al finalizar las preguntas, el usuario sumará un progreso en el tema que permitirá desbloquear otros apartados y conseguir ciertos logros en su perfil. La aplicación es gratuita pero tiene mejoras de pago que te permitirán avanzar con más facilidad y sin anuncios.

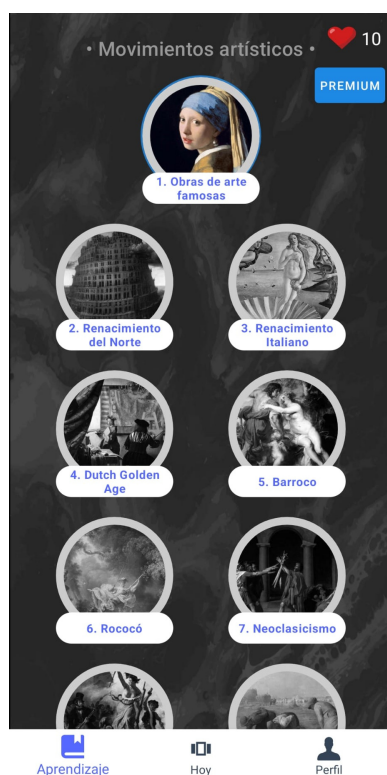


Figura 2.3: Menú de selección de las lecciones de Artly clasificadas en movimientos artísticos

## BMath

BMath es un software para móviles para el aprendizaje de conceptos matemáticos mediante gamificación. Esta aplicación crea un programa personalizado para el usuario en función del curso académico en el que esté, adaptándose a su conocimiento y a su nivel. Pese a ser gratuita, el tiempo diario que te permite el modo básico es de 5 minutos, mientras que si pagas te permitirá ampliarlo en 10 minutos cada día.

Es un ejemplo de aprendizaje por gamificación ya que la aplicación es prácticamente un videojuego (de hecho, se consideran así), pues muestra una ciudad (Figura 2.4) por la que puedes mover la cámara y donde construyes distintos edificios. Para construirlos, deberás superar una serie de pruebas sobre álgebra, geometría, etc. También obtendrás puntos de experiencia al superar los ejercicios (Figura 2.5) y, al subir de nivel, recibirás nuevos edificios y decoraciones para tu ciudad. Con esto, se pretende que el usuario tenga la motivación de realizar ejercicios y superarlos correctamente para lograr que su ciudad prospere. Como alumno también puedes elegir tu propio personaje animado que te identificará en el juego.

Los diseños del software están muy cuidados y posee colores muy llamativos para que el usuario se sienta motivado y atraído. Además, la aplicación proporciona una experiencia de usuario agradable y motivadora, ayudando al alumno en todo momento a solucionar los ejercicios y enseñándole técnicas para resolverlos.

La aplicación también posee otras funciones como un apartado donde añadir recordatorios para acceder a la aplicación, preguntas para conocer cómo se siente el usuario tras cada sesión de estudio, sección parental, etc.



Figura 2.4: Ciudad de BMath donde se seleccionan los niveles y se construyen los edificios para mostrar el progreso del usuario



Figura 2.5: Ejercicio de BMath sobre cálculo matemático (sumas y restas) con calculadora, con papel y mentalmente

### 2.1.2. Aplicaciones de aprendizaje musical

Centrándonos ya en el tema que nos ocupa (el aprendizaje musical) existen numerosas aplicaciones que ayudan a la adquisición de conocimientos musicales o a la ayuda a aprender a tocar un instrumento musical. A continuación se muestran algunas de ellas.

#### Curso de Lenguaje Musical

Esta herramienta para móvil tiene el objetivo de enseñar los conceptos básicos de lenguaje musical. La aplicación proporciona una forma de aprendizaje muy tradicional y que no está basada en la gamificación, ya que no dispone de ningún tipo de seguimiento del progreso ni de ejercicios para realizar sobre las lecciones aprendidas. La app contiene una lista de lecciones (Figura 2.6) y cada una de ellas muestra un vídeo de un docente explicando el contenido del temario.

Por último, en cuanto a la interfaz, la aplicación es muy sencilla y tiene un diseño poco atractivo, dando una simple lista con enlaces a los vídeos de las lecciones sin ningún tipo de decoración o imagen.

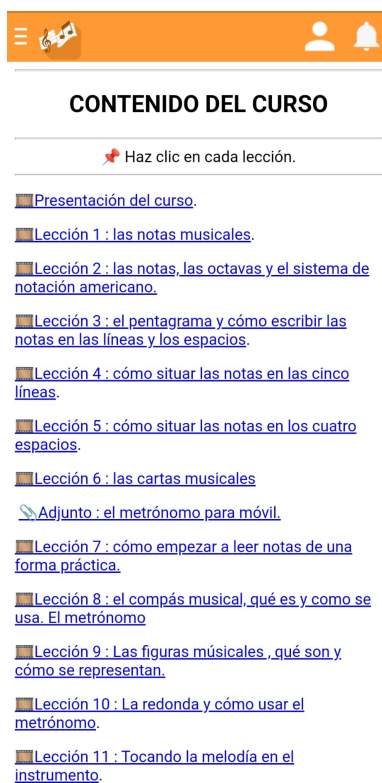


Figura 2.6: Lista de lecciones que ofrece la aplicación Curso de Lenguaje Musical

## Sonid

Sonid consiste en una aplicación para el aprendizaje de piano a través de la gamificación. La aplicación se divide en distintas lecciones (Figura 2.7b) que el usuario irá desbloqueando al pasarlas correctamente. Cada lección explica unos conceptos básicos de música y piano y, al finalizar, el usuario deberá realizar una serie de ejercicios para comprobar que ha entendido y aprendido el contenido del temario.

Algunos de los ejercicios que posee esta aplicación son:

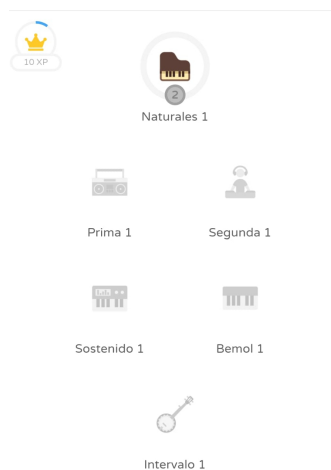
- Seleccionar la tecla en el piano que corresponde a una nota musical. (Figura 2.7a)
- Responder si la tecla que se ha pulsado es una nota musical o no.
- Decidir qué nota musical corresponde a una tecla del piano que se ha pulsado.

Además de las lecciones, el usuario también puede practicar con escalas y acordes del piano, así como aprenderlas en la wiki y en el diccionario que posee la aplicación. También dispone de un foro para que toda la gente pueda compartir sus dudas.

Por último, en cuanto al seguimiento, el usuario puede ver su progreso y sus estadísticas en su perfil, donde aparece el número de lecciones completadas, el número de errores que ha tenido, la experiencia que tiene. Además, el usuario consigue logros al completar las lecciones y comparte una clasificación global con otros usuarios que también usan la aplicación.



(a) Ejercicio de Sonid para conocer las notas de un piano



(b) Menú de selección de lecciones de Sonid dividido por escalas

## 2.2. Desarrollo de Software

En cuanto a las tecnologías y herramientas que existen para el desarrollo de aplicaciones, existen numerosas opciones que se pueden utilizar. En este apartado se muestran algunas de ellas.

### 2.2.1. Flutter

Flutter es un kit de desarrollo software (framework) de código abierto destinado al desarrollo de aplicaciones móviles multiplataforma. Está programado en Dart, fue creado por Google y su primera versión estable se lanzó en 2018. Se emplea para desarrollar aplicaciones para móvil, web y escritorio desde una sola base de código, lo cual permite mucha más agilidad y consistencia.



Figura 2.7: Logo de Flutter

Flutter ofrece tres ventajas respecto a otros frameworks utilizados para el desarrollo de aplicaciones multiplataformas:

- **Compilación en nativo**
- **Flexibilidad** para crear interfaces gráficas
- **Desarrollo veloz**, permitiendo ver el resultado del código al instante.

Flutter está basado en el concepto de widgets, que son los elementos que componen la interfaz y la interacción del usuario. Estos widgets pueden ser modificados, añadidos o eliminados dinámicamente, permitiendo una gran flexibilidad a la hora de crear interfaces gráficas.

### 2.2.2. React Native

Framework de código abierto creado por Meta Platforms (Facebook) y destinado al desarrollo de aplicaciones multiplataforma para Android, iOS, Web, Windows, etc. Está basado en Javascript como lenguaje de programación y en React como herramienta para la interfaz de usuario. La diferencia con React es que, en lugar de trabajar con el navegador, trabaja con las plataformas móviles al transformar los componentes en nativos en función de la plataforma en la que se esté ejecutando la aplicación.

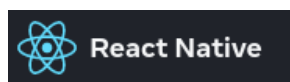


Figura 2.8: Logo de React Native



Esta tecnología permite construir aplicaciones móviles usando solamente Javascript, con el mismo diseño que utiliza React permitiendote realizar una interfaz de usuario completa mediante componentes.

Podemos encontrar dos ventajas en este framework:

- **Código compartido:** tu código puede ser compartido para diferentes plataformas.
- **Comunidad:** Existe una amplia comunidad sobre React y React Native, lo cual permite encontrar soluciones a los problemas que se puedan presentar en el desarrollo.

Como desventaja, encontramos una limitación en los componentes nativos, pues seguramente se necesite escribir algo de código específico para una plataforma en concreto

### 2.2.3. Dart

Dart es un lenguaje de programación de código abierto desarrollado por Google y utilizado por Flutter. Es orientado a objetos y de tipado estático. Surgió en 2011 como alternativa a Javascript.



Figura 2.9: Logo de Dart

Dart posee las siguientes ventajas:

- **Lenguaje fácil y sencillo** de aprender
- **Acceso gratuito**
- **Funciona en todas las plataformas**
- **Programación estructurada y flexible**
- Al ser **orientado a objetos**, facilita la encapsulación y reutilización de código gracias al uso de clases.

### 2.2.4. Kotlin

Kotlin es un lenguaje de programación construido sobre Java, desarrollado por JetBrains y lanzado en 2016. Es orientado a objetos y tiene tipado estático, pero también soporta la programación por procedimientos y funciones.

Algunas de sus ventajas son:

- **Compatibilidad con Java:** permite utilizar código Java en Kotlin y viceversa
- **Lenguaje fácil y sencillo** de aprender y usar por su sintáxis similar a Java
- **Robusto:** Es un lenguaje seguro con los valores nulos
- **Exactitud y claridad:** reduce el código repetido de forma sustancial y disminuye la probabilidad de error.

### 2.2.5. NodeJS

NodeJS es un entorno en tiempo de ejecución de código abierto, del lado del servidor y programado en Javascript. Es asíncrono y dirigido por eventos. Node es utilizado para el desarrollo de aplicaciones de red escalables y rápidas, ofreciendo beneficios en rendimiento, velocidad de desarrollo, etc.

Se utiliza en el desarrollo de aplicaciones web porque permite ejecutar Javascript del lado del servidor en lugar de en el navegador.

Los principales motivos para utilizar NodeJS son:

- **Simultaneidad de peticiones:** NodeJS es capaz de manejar múltiples peticiones al mismo tiempo gracias al modelo dirigido por eventos, lo que permite una mayor escalabilidad.
- **Lenguaje sencillo** basado en Javascript.
- **Gestión de paquetes de calidad** gracias a NPM.

### 2.2.6. React

React es una biblioteca Javascript de código abierto que se utiliza para la creación de interfaces de usuario de forma fácil y sencilla. Es mantenida por una gran comunidad de desarrolladores de software libre y fue lanzada en 2013.

En React se trabaja con componentes, los cuales son elementos o partes de la interfaz de usuario que facilitarán la reutilización de código y la modularidad.

Entre las ventajas de React encontramos:

- **Componentes reutilizables:** React permite crear componentes que pueden ser reutilizados en otras partes de la aplicación.
- **Fácil de aprender:** React es fácil de aprender y de utilizar, además de tener una documentación sólida y una gran cantidad de recursos online gratuitos.
- **Rendimiento:** React es rápido y eficiente gracias al DOM virtual.

### 2.2.7. Angular

Angular es un framework de código abierto para el desarrollo de aplicaciones web y móviles. Está basado en TypeScript y está desarrollado por Google. Esta tecnología también está basada en componentes, lo cual permite la creación de aplicaciones web escalables

### 2.2.8. MongoDB

MongoDB es una base de datos no relacional de código abierto y orientada a documentos. Es de tipo NoSQL, es decir, que no utiliza el modelo relacional de bases de datos tradicionales. Cada registro de la base de datos es un documento que consta de pares clave - valor (muy similar a los objetos JSON). Está escrito en C++ y las consultas se hacen pasando objetos JSON como parámetros.

MongoDB proporciona las siguientes ventajas:

- **Rendimiento:** MongoDB es más rápido porque almacena los datos directamente en un mismo sitio (colección).
- **Simplicidad:** No hay que seguir un esquema ni unir tablas como en SQL.
- **Flexibilidad** para diseñar el esquema y crear las relaciones.



Figura 2.10: Logo de MongoDB

### 2.2.9. Pilas MERN & MEAN

MERN es un conjunto de tecnologías que se utilizan para el desarrollo de aplicaciones web. Está compuesto por las siglas de MongoDB, Express (ayuda para crear y gestionar el backend en Node), React y NodeJS. MEAN es igual, pero sustituye React por Angular.

Hay una gran cantidad de beneficios al utilizar una de estas pilas para desarrollar una aplicación web como por ejemplo:

- **Cubre todo el ciclo de desarrollo:** Desde el backend hasta el frontend.
- **Facilita el trabajo** con la arquitectura modelo-vista-controlador.
- **Rápido desarrollo:** El desarrollo de una aplicación web con estas tecnologías es rápido y eficiente.
- **Código abierto:** Todas las tecnologías que componen estas pilas son de código abierto y gratuitas y están respaldadas por la comunidad.
- **Fácil mantenimiento:** El mantenimiento de una aplicación web con estas tecnologías es fácil y sencillo.



Figura 2.11: Pilas MEAN y MERN

### 2.2.10. MariaDB

MariaDB es un sistema de gestión de bases de datos relacional de código abierto. Es un fork de MySQL y está desarrollado por su comunidad de usuarios. Surgió en 2009 como consecuencia de la compra de MySQL por parte de Sun Microsystems. Sigue un modelo relacional, lo cual significa que el contenido de los datos se almacena en tablas con un esquema más rígido que en MongoDB u otra base de datos noSQL.



Figura 2.12: Logo de MariaDB

Pese a haber mencionado solamente MariaDB como ejemplo de sistema de gestión de bases de datos relacional, existen muchos otros como por ejemplo PostgreSQL, MySQL, etc.



## Capítulo 3

# Especificación de requisitos

### 3.1. Introducción

A continuación, vamos a describir y detallar los requisitos del sistema que vamos a desarrollar. En esta sección se describirá qué vamos a construir y cómo lo vamos a hacer, con sus restricciones específicas.

#### 3.1.1. Propósito

En este capítulo se pretende describir de forma clara y precisa las funciones, características y restricciones del sistema que se va a desarrollar. Estas definiciones servirán al equipo de desarrollo para conocer las necesidades del sistema y a los usuarios finales.

Además, este documento servirá como base para el desarrollo de las funcionalidades descritas y como medio de comunicación entre las partes involucradas en el proyecto.

#### 3.1.2. Ámbito del Sistema

El sistema que vamos a desarrollar es una aplicación multiplataforma que permitirá a los usuarios aprender conceptos musicales de una forma amena y divertida.

#### 3.1.3. Definiciones, Acrónimos y Abreviaturas

- **Requisito:** Es una condición o característica que debe cumplir el sistema para satisfacer una necesidad o cumplir una función.

- **Funcionalidad:** Descripción de lo que debe hacer el producto software.
- **Restricción:** Condición que limita la funcionalidad del sistema.
- **Interfaces externas:**
- **Rendimiento:**
- **Usuarios/Cliente:**
- 

#### 3.1.4. Referencias

#### 3.1.5. Visión General del Documento

### 3.2. Descripción General

#### 3.2.1. Perspectiva del producto

#### 3.2.2. Funciones del producto

#### 3.2.3. Características de los usuarios

#### 3.2.4. Restricciones

#### 3.2.5. Suposiciones y dependencias

### 3.3. Requisitos Específicos

#### 3.3.1. Requisitos funcionales

#### 3.3.2. Requisitos no funcionales

#### 3.3.3. Requisitos de información

#### 3.3.4. Bocetos de interfaz de usuario



## Capítulo 4

# Planificacion



## Capítulo 5

# Análisis del problema



## Capítulo 6

# Diseño



## Capítulo 7

# Implementación





## Capítulo 8

# Pruebas



## Capítulo 9

# Conclusiones



