# **DOCUMENTACIÓN DE PROYECTO INTERDISCIPLINARIO (SoundVibe)**

CATIVA RODRIGO; SILVA BENJAMÍN; VARGAS LUCAS

Escuela Técnica N. º36 D.E. 15

## Definiciones y especificación de requerimientos

### **SoundVibe** es una aplicación de gestión musical que permite a los usuarios organizar su biblioteca de canciones, crear listas de reproducción personalizadas y reproducirlas de manera continua. Ofrece una experiencia de usuario intuitiva y centrada en la personalización y la música.

* 1. Definición general del proyecto de software

El principal objetivo de la aplicación es facilitar la gestión de bibliotecas musicales de los usuarios mediante una interfaz atractiva y funcional. La aplicación permitirá a los usuarios añadir, eliminar y organizar canciones dentro de su biblioteca, crear listas de reproducción personalizadas y reproducirlas según sus preferencias. Además, **SoundVibe** buscará ofrecer una experiencia de usuario optimizada.

### Especificación de requerimientos del proyecto

Los requerimientos técnicos del proyecto es el uso del lenguaje de programación Java y otro lenguaje a elección para complementar el desarrollo, además de la implementación de una base de datos para gestionar y almacenar información sobre canciones, listas de reproducción y usuarios. La aplicación debe contar con una interfaz gráfica de usuario (GUI) que sea coherente y fácil de navegar.

El proyecto de SoundVibe es independiente y no forma parte de un sistema preexistente. No se trata de una nueva versión o una derivación de otro software.

## Requerimientos Funcionales

La aplicación **SoundVibe** deberá cumplir con los siguientes requerimientos funcionales:

1. El sistema permitirá a los usuarios añadir canciones a la biblioteca a través de la importación de archivos locales
2. Los usuarios podrán crear, modificar y eliminar listas de reproducción personalizadas dentro de la aplicación. Estas listas se podrán organizar manualmente o mediante género, artista o fecha de adición.
3. El sistema ofrecerá una funcionalidad de reproducción continua de música, permitiendo la reproducción en segundo plano mientras el usuario realiza otras actividades en su dispositivo.
4. **SoundVibe** incluirá una interfaz de usuario que permita la búsqueda rápida de canciones, artistas o listas de reproducción dentro de la biblioteca.
5. El sistema almacenará de manera persistente las canciones, listas de reproducción y configuraciones del usuario para que puedan acceder a ellas en futuras sesiones.

## Product List

### **Aplicación de reproductor de música**

Software de aplicación para reproducir música, disponible en dispositivos móviles y de escritorio.

#### **Interfaz de usuario (UI)**

* + - * Pantalla de inicio
      * Pantalla de exploración
      * Pantalla de reproducción
      * Pantalla de biblioteca

#### **Backend (Servidor)**

* + - * Servicio de manejo de música
      * Gestión de datos

#### **Base de datos**

* + - * Estructura para canciones
      * Estructura para álbumes
      * Estructura para artistas
      * Estructura para listas de reproducción
      * Estructura para usuarios

#### **Funcionalidades principales**

* + - * Carga de música
      * Reproducción de música
      * Búsqueda y exploración
      * Gestión de listas de reproducción
      * Autenticación de usuarios

### **Plataforma de sincronización en la nube**

Servicio en la nube para sincronizar la música y las listas de reproducción entre diferentes dispositivos del usuario.

#### **Integración con aplicación SoundVibe**

* + - * Sincronización de listas de reproducción
      * Sincronización de canciones

#### **Gestión de usuarios**

* + - * Sincronización de preferencias y datos de usuario

#### **Servicios en la nube**

* + - * Almacenamiento en la nube (utilizando Firebase o AWS)

### **Algoritmo de recomendaciones**

Sistema de recomendación de música basado en el historial de escucha y preferencias del usuario.

#### **Módulo de análisis de historial de escucha**

#### **Módulo de preferencias del usuario**

#### **Generador de recomendaciones**

#### **Integración con pantallas de inicio y exploración en la UI**

### **Sistema de gestión de usuarios**

Módulo para gestionar el registro, inicio de sesión, y perfiles de usuarios.

#### **Autenticación de usuarios**

* + - * Registro
      * Inicio de sesión

#### **Gestión de perfil de usuario**

* + - * Actualización de información personal
      * Configuración de preferencias

#### **Sincronización de datos entre dispositivos**

### **Base de datos del proyecto**

Estructura de base de datos para almacenar y gestionar toda la información relevante de la aplicación.:

#### **Base de datos para canciones**

#### **Base de datos para álbumes**

#### **Base de datos para artistas**

#### **Base de datos para listas de reproducción**

#### **Base de datos para usuarios**

## Product Backlog

### **Aplicación de reproductor de música**

Versión inicial de la aplicación que permite la reproducción de música y la gestión básica de listas de reproducción, sin funcionalidades avanzadas ni enfoque en la estética.

#### Backend

#### **Manejo de música**

* + - * Carga de música
      * **Reproducción de música**

### **Base de Datos**

#### **Estructura para Canciones**

* + - * Tabla de Canciones: Crear una tabla sencilla para almacenar la información básica de cada canción (título, artista, archivo de música).

#### **Estructura para Listas de Reproducción**

* + - * **Tabla de Listas de Reproducción: Crear una tabla para almacenar las listas de reproducción del usuario, con referencias a las canciones.**

### **Funcionalidades Principales**

#### **Carga de Música**

* + - * **Subir Canciones: Permitir a los usuarios cargar canciones desde sus dispositivos.**

#### **Reproducción de Música**

* + - * **Reproducir y Pausar Canciones: Implementar la funcionalidad para reproducir y pausar canciones.**

#### **Gestión de Listas de Reproducción**

* + - * **Crear y Guardar Listas de Reproducción: Permitir a los usuarios crear listas de reproducción básicas y guardar canciones en ellas.**

## Prototipo