

StellarSound



1. Descripción y justificación del proyecto	3
1.1. Descripción del proyecto	3
1.2. Justificación del proyecto	4
2. Alcance del Proyecto	5
2.1. Backend con Laravel Breeze y MySQL	5
2.2. Aplicación Android en Android Studio	5
2.3. Panel de Administración	5
3. Tecnologías Utilizadas	6
4. Objetivos	7
5. Casos de uso	8
5.1. Autenticación de usuario	8
5.2. Exploración de canciones	9
5.3. Gestión de playlists	10
5.4. Marcado de favoritos (likes)	11
5.5. Administración	12
6. Requisitos del sistema	13
7. Guía de estilos	14
8. Prototipado	15
9. Mapa de navegación	16
10. Modelo de la base de datos	22
11. Manual de usuario	23
11.1. ¿Qué es StellarSound?	23
11.2. Requisitos del sistema	23
11.3. Instalación de la aplicación	23
11.4. Primeros pasos	24
12. Retos encontrados en el desarrollo	24
13. Conclusión	25
14. Código fuente	25

1. Descripción y justificación del proyecto

1.1. Descripción del proyecto

StellarSound es una aplicación de streaming de música online, cuyo diseño permitirá a los usuarios explorar, reproducir y gestionar sus propias librerías musicales alojados en un servidor privado. A diferencia de aplicaciones similares como Spotify o Apple Music, StellarSound ofrecerá un entorno personalizado y controlado sin restricciones en la experiencia de escucha musical.

El sistema contará con una API desarrollada en Laravel que servirá como puente entre la app móvil y la base de datos de MySQL, donde se almacenará la información pertinente de toda la aplicación. Además, Laravel Breeze será usado para la autenticación y gestión de inicio de sesión, para garantizar la seguridad de la app. También se integrará un panel administrativo donde los administradores podrán hacer gestión de usuarios, agregar canciones y realizar configuraciones del sistema.

El backend será alojado en un servidor privado, donde se almacenarán tanto los datos como los archivos de audio. Permitiendo a los administradores del sistema un control total del contenido y evitar dependencias de servicios externos.

1.2. Justificación del proyecto

En la actualidad, las plataformas de streaming musical han dominado el mercado con millones de usuarios que acceden a contenido en línea diariamente. Pero estas plataformas presentan las siguientes variaciones de las demás:

- Suscripciones mensuales y dependencias de las mismas, donde los usuarios podrán acceder a funciones avanzadas como la descarga de canciones o la eliminación de anuncios.
- Restricción de contenido, ciertas canciones pueden no estar disponibles por restricción de licencia.
- Falta de personalización, los usuarios deben ajustarse a la interfaz y características de la aplicación.
- Publicidad invasiva, las versiones gratuitas incluyen constantes anuncios que interrumpen la experiencia del usuario.

Con este contexto StellarSound busca ofrecer una alternativa más flexible y personalizable.

Beneficios del proyecto

- Autonomía y control, al usar API y bbdd propias los administradores pueden gestionar contenido sin una dependencia de terceros.
- Accesibilidad, no se requerirá ninguna suscripción para acceder a todo el contenido de la app.
- Personalización total, donde los usuarios pueden gestionar sus bibliotecas sin estar atados a algoritmos limitadores de selección de canciones.

- Privacidad, la información de los usuarios no será compartida con anunciantes de terceros.

2. Alcance del Proyecto

2.1. Backend con Laravel Breeze y MySQL

- Creación de API con Laravel.
- Implementación de Laravel Breeze para autenticación.
- Base de datos relacional en MySQL y Docker con estructura optimizada.
- Gestión de usuarios, roles y permisos (usuario y administrador).

2.2. Aplicación Android en Android Studio

- Interfaz intuitiva y atractiva.
- Consumo de API de autenticación y reproducción de canciones.
- Gestión de playlist y favoritos.
- Reproductor de música integrado con controles básicos.

2.3. Panel de Administración

- Subida de canciones al servidor.
- Gestión de usuarios y roles.
- Visualización de estadísticas y métricas de uso.

3. Tecnologías Utilizadas

Área	Tecnología
Backend	Laravel Breeze (PHP) → Framework potente para desarrollo de API REST rápida y segura. Breeze es un sistema de autenticación básica pero sin sobrecarga técnica. Se usará para el acceso a la base de datos y la gestión de comunicación de Request entre este y Android Studio.
Base de Datos	MySQL y Docker → Base de datos relaciones eficiente y con mucho soporte. Docker permite un fácil despliegue y escalabilidad, reduciendo los problemas de generación de usuarios por IP típicos de MySQL.
Frontend	Android Studio Java → IDE que permite la creación de apps para Android, con una fácil integración de APIs. Se usará para el visionado y gestión de canciones y playlists respectivamente.
API	RESTful API con Laravel → Estilo de arquitectura de API que permite la conexión cliente-servidor mediante peticiones HTTP. Será usado para las peticiones de datos entre Android Studio y Laravel
Control de Versiones	GitHub → Sistema de subida de versiones del proyecto.
Servidor	Ubuntu LXC → Sistema operativo libre y seguro, además de ligero, cuyo alojamiento de datos estará centrado en la base de datos y los archivos multimedia (canciones y portadas).
Autenticación	Laravel Breeze con Sanctum → Permite la autenticación mediante tokens personales, para una comunicación de usuarios entre la app y Laravel.

4. Objetivos

- Desarrollo de app Android intuitiva para exploración y reproducción de música.
- Implementación de un backend robusto con Laravel para gestionar la autenticación, usuarios, roles, canciones y playlists.
- Administración de la base de datos y archivos (canciones y portadas) en un servidor Linux
- Garantizar una arquitectura segura y escalable usando Docker.
- Habilitar una vista de gestión y administración de usuarios, contenido y estadísticas básicas.

5. Casos de uso

5.1. Autenticación de usuario

CU-01	Registro/logueo de un usuario	
Actor	Usuario	
Objetivo	Hacer pasar al usuario por la pasarela de logueo.	
Requisitos Asociados (Dependencias)	Gestión de usuarios y roles	
Descripción	Este caso consiste en que el usuario inicie sesión en la aplicación.	
Precondición	La app debe haber sido instalada previamente.	
Secuencia Normal	PASO	ACCIÓN
	1. Ubicarse en la app	Dependiendo si el usuario quiere iniciar sesión o registrarse deberá dirigirse a su vista correspondiente.
	2. Rellenar campos	El usuario debe rellenar los campos (si se ha registrado debe volver a hacerlo en la vista de logueo)
	3. Pasar login	Cuando ya esten los datos debe pulsar el botón de login para entrar a la aplicación
Postcondiciones o garantía de éxito		
Excepciones	PASO	ACCIÓN
	1	En caso de credenciales incorrectas no pasará de la vista de logueo

5.2. Exploración de canciones

CU-02	Exploración de canciones	
Actor	Usuario	
Objetivo	Permitir al usuario visualizar la lista de canciones disponibles.	
Requisitos Asociados (Dependencias)	Autenticación con token válido API disponible y funcionando correctamente	
Descripción	El usuario puede navegar entre las canciones disponibles en la app filtradas por género, artista o fecha de publicación.	
Precondición	La app debe haber sido instalada previamente El usuario debe haber iniciado sesión correctamente..	
Secuencia Normal	PASO	
	1. Entrar en la vista principal tras el login	
	2. Dirigirse a la vista de búsqueda	
	3. Buscar en la el barra de búsqueda y visualizar las canciones	
Postcondiciones o garantía de éxito	La app muestra correctamente la lista de canciones buscadas.	
Excepciones	PASO	ACCIÓN
	1	En caso de no existir esa canción en la base de datos no se darán resultados

5.3. Gestión de playlists

CU-03	Gestión de playlists										
Actor	Usuario										
Objetivo	Permitir crear o eliminar playlists personales										
Requisitos Asociados (Dependencias)	Autenticación con token válido API disponible y funcionando correctamente										
Descripción	El usuario puede organizar canciones en listas personalizadas.										
Precondición	La app debe haber sido instalada previamente El usuario debe haber iniciado sesión correctamente.										
Secuencia Normal	<table border="1"> <tr> <th colspan="2">PASO</th></tr> <tr> <td colspan="2">1. Navegar a la sección de playlists</td></tr> <tr> <td colspan="2">2. Crear una nueva playlist indicando nombre y su imagen</td></tr> <tr> <td colspan="2">3. Añadir canciones a la playlist creada</td></tr> <tr> <td colspan="2">4. Eliminar playlist creada</td></tr> </table>	PASO		1. Navegar a la sección de playlists		2. Crear una nueva playlist indicando nombre y su imagen		3. Añadir canciones a la playlist creada		4. Eliminar playlist creada	
PASO											
1. Navegar a la sección de playlists											
2. Crear una nueva playlist indicando nombre y su imagen											
3. Añadir canciones a la playlist creada											
4. Eliminar playlist creada											
Postcondiciones o garantía de éxito	Playlists reflejan correctamente las acciones del usuario..										
Excepciones	<table border="1"> <tr> <th>PASO</th><th>ACCIÓN</th></tr> <tr> <td>1</td><td>En caso de no dar una imagen, la playlist no se guardará.</td></tr> </table>	PASO	ACCIÓN	1	En caso de no dar una imagen, la playlist no se guardará.						
PASO	ACCIÓN										
1	En caso de no dar una imagen, la playlist no se guardará.										

5.4. Mercado de favoritos (likes)

CU-04	Marcado de favoritos	
Actor	Usuario	
Objetivo	Permitir al usuario marcar una canción como favorita.	
Requisitos Asociados (Dependencias)	Autenticación con token válido API disponible y funcionando correctamente	
Descripción	El usuario pulsa un botón de "like" en una canción para marcarla como favorita..	
Precondición	La app debe haber sido instalada previamente El usuario debe haber iniciado sesión correctamente.	
Secuencia Normal	PASO	
	1. Reproducir una canción	
	2. Pulsar el botón de like en una canción	
	3. El backend registra la acción	
	4. Se actualiza visualmente el estado del botón	
Postcondiciones o garantía de éxito	La canción queda marcada como favorita para ese usuario.	
Excepciones	PASO	
	ACCIÓN	
	1	En caso de reproducir la canción desde la playlist, la función de like no funcionará.

5.5. Administración

CU-05	Administración	
Actor	Administrador	
Objetivo	Gestionar usuarios desde el panel web	
Requisitos Asociados (Dependencias)	Rol de administrador Acceso al panel web de Laravel	
Descripción	El administrador puede crear, editar y eliminar usuarios.	
Precondición	Sesión iniciada como administrador. Usuarios existentes en la base de datos.	
Secuencia Normal		
	PASO	
	1. Acceder al panel de administración	
	2. Consultar la lista de usuarios	
	3. Crear nuevo usuario o modificar/borrar existentes	
Postcondiciones o garantía de éxito	La base de datos refleja correctamente los cambios	
Excepciones		
	PASO	ACCIÓN

6. Requisitos del sistema

6.1. Funcionales

- Registro, login y autenticación de usuarios.
- CRUD de canciones (administrador).
- CRUD de playlists (usuario).
- Reproducción de canciones.
- Likes de canciones
- Diferenciación de roles y middleware de acceso.

6.2. No funcionales

- Escalabilidad para futuras ampliaciones.
- Alta disponibilidad y rendimiento de streaming.
- Seguridad: cifrado de contraseñas, protección CSRF en backend.

6.3. Requisitos de interfaz

- App Android simple, responsiva, moderna y minimalista.
- Navegación fluida y mínima carga visual.
- Backend con endpoints bien documentados.

7. Guía de estilos

Guía de Estilos

Proyecto Final-StellarSound

Logotipo



Colores Primarios

#000000

#FFFFFF

#9933CC

#CCFF33

Colores Secundarios

#AAAAAA

#CCCCCC

Tipografía

Título 1 (H1) - Inter - 40px
Título 2 (H2) - Inter - 24px
 Texto mediano (H4) - Inter - 18px
 Texto pequeño (H5) - Inter - 16px
 Texto enano (H6) - Inter - 11px

Iconografía



Botones y Componentes

Entrar +

At

Para

Buscar

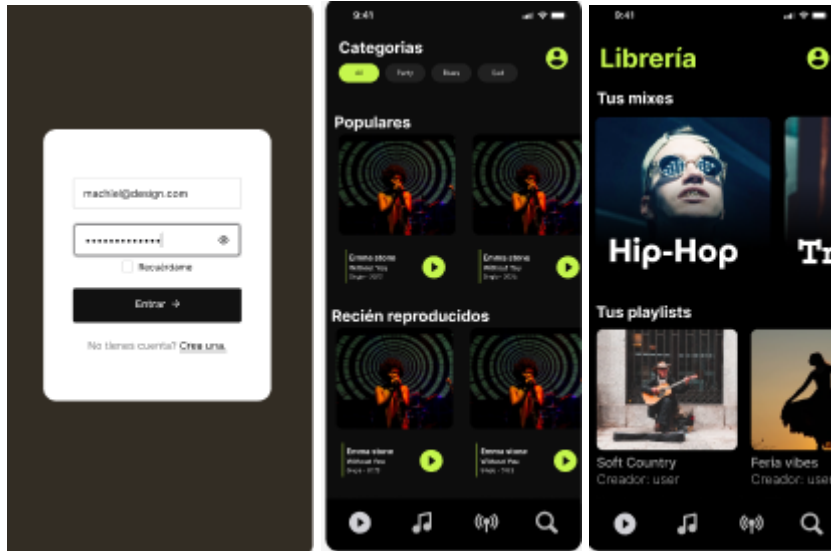
Salir

8. Prototipado

LoginActivity

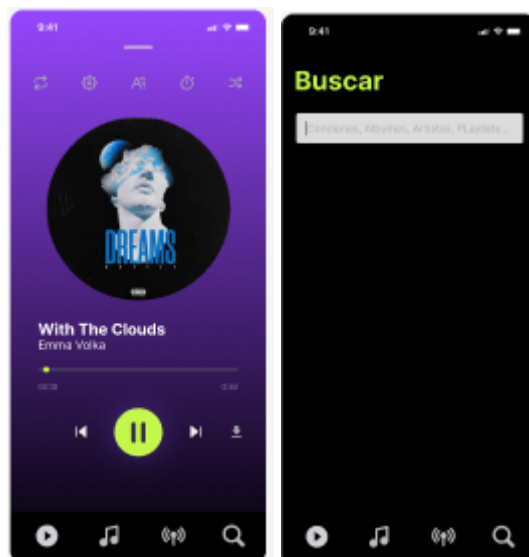
HomeFragment

PlaylistFragment



PlayFragment

SearchFragment

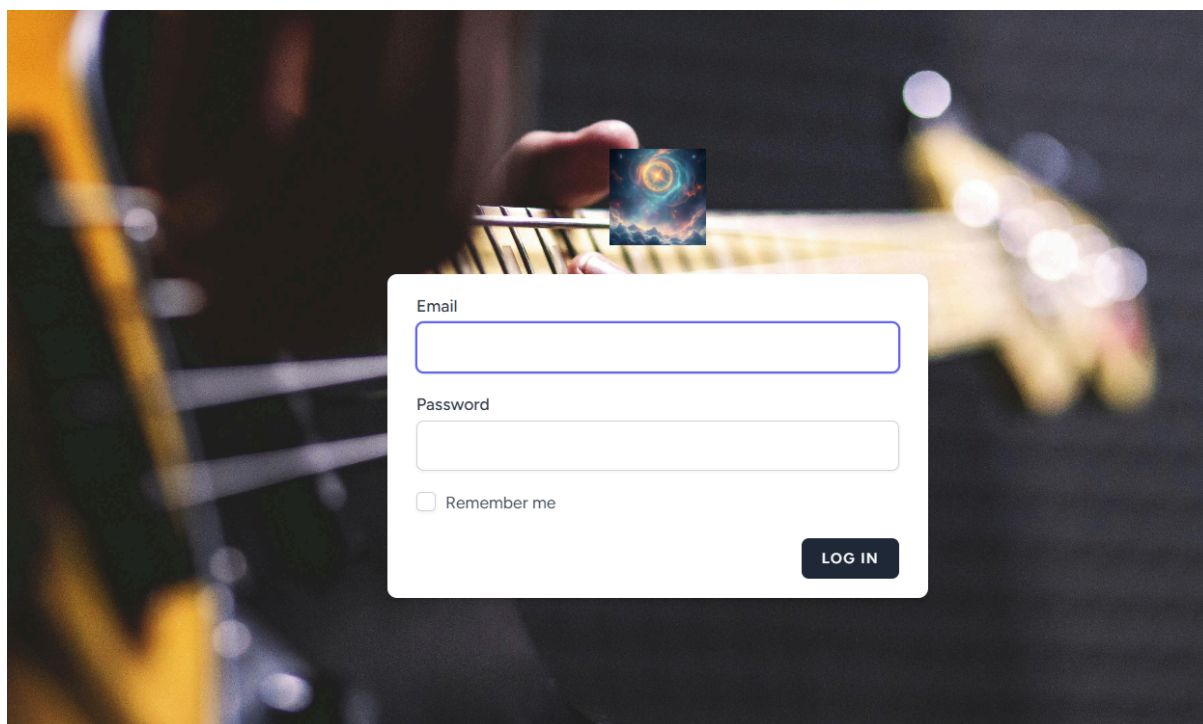


9. Mapa de navegación

Los usuarios deberán loguearse en LoginActivity, luego pulsar el respectivo botón de logueo, una vez dentro podrán acceder a las 4 vistas de la app, a través de los botones de navegación inferiores, siendo el primero para HomeFragment, en el que se encontrarán de manera automática nada más loguear, el segundo sería PlaylistFragment donde se almacenan y pueden reproducir sus playlists, el tercero sería el PlayFragment, donde pueden mirar la canción que están reproduciendo (si y sólo si hay alguna canción reproduciéndose), el último botón es para el SearchFragment, donde se creará una lista de las posibles canciones similares a la búsqueda del usuario.

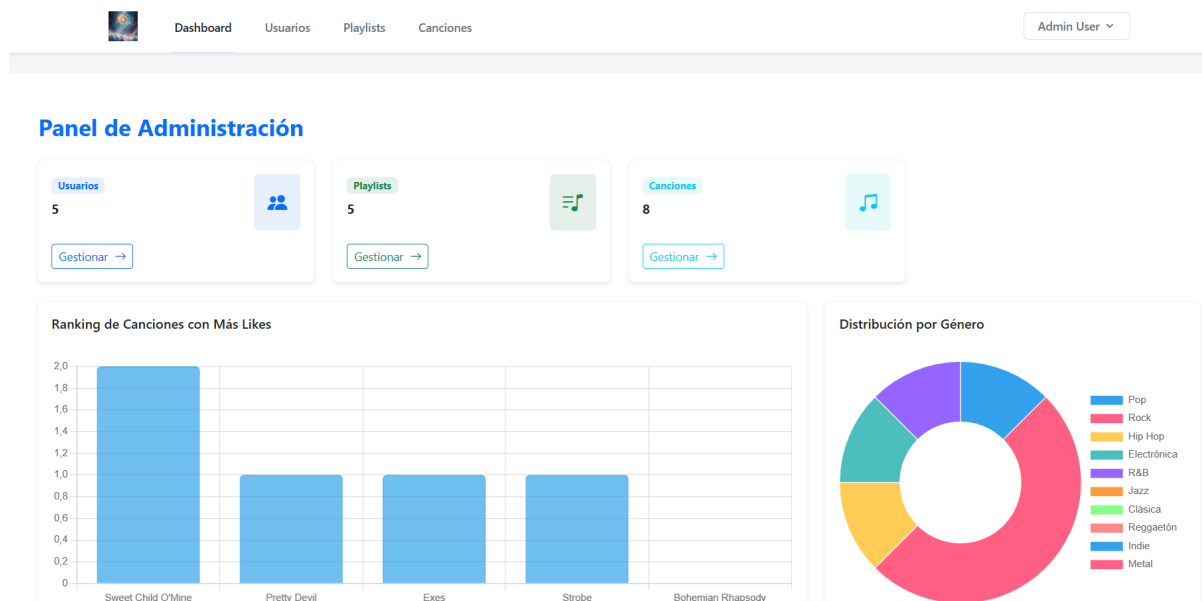
Para la web de administradores, tenemos lo siguiente:

Vista de login



Donde los administradores podrán loguearse dentro de la web.

Vista de dashboard



Donde se podrán ver las estadísticas de las canciones y una visión de datos generales, también hay una cabecera compartida entre todas las vistas para la navegación, siendo la primera la de esta misma vista.

Vista de usuarios


Gestión de Usuarios [Nuevo Usuario](#)

Listado de Usuarios Total: 5

ID	Nombre	Email	Rol	Playlists	Favoritos	Acciones
#1	A Admin User <small>Registrado: Fecha no disponible</small>	admin@musicapp.com	Administrador	0	0	Editar Eliminar
#2	J Juan Pérez <small>Registrado: Fecha no disponible</small>	juan@musicapp.com	Usuario	2	1	Editar Eliminar
#3	M María García Ser <small>Registrado: Fecha no disponible</small>	maria@musicapp.com	Usuario	1	2	Editar Eliminar
#4	C Carlos López <small>Registrado: Fecha no disponible</small>	carlos@musicapp.com	Usuario	2	2	Editar Eliminar
#5	P Pablo <small>Registrado: 12/06/2025</small>	pablo@mail.com	Usuario	0	0	Editar Eliminar

Pulsando el segundo botón de la cabecera, se puede acceder a una vista de usuarios, donde dará la posibilidad de administrarlos, a continuación se verán las vistas de los botones dentro de esta:

Vista de nuevos usuarios


Dashboard
 Usuarios
 Playlists
 Canciones
 Admin User

Crear Usuario

Nombre

Correo electrónico

Contraseña

Rol


Usuario

Cancelar

Crear usuario

Pulsando el botón de nuevo usuario se podrá acceder a esta vista de creación de usuarios, donde una vez creados se redirigirá automáticamente a la vista anterior.

Vista de edición de usuarios


Dashboard
 Usuarios
 Playlists
 Canciones
 Admin User

Editar Usuario

Nombre

Juan Pérez

Correo electrónico

juan@musicapp.com

Nueva contraseña (opcional)

Deja en blanco para mantener la actual

Rol


Usuario

Cancelar

Guardar cambios

Pulsando en el botón de lápiz del usuario deseado se podrá acceder a una vista similar a la de nuevos usuarios solo que tendrá los campos rellenos, y cualquier modificación se hará a dicho usuario, tanto guardar como cancelar llevan de vuelta a la vista de usuarios.

Vista de playlists


[Dashboard](#)
[Usuarios](#)
[Playlists](#)
[Canciones](#)


Admin User ▾

Gestión de Playlists

[Nueva Playlist](#)

El tercer botón de la cabecera superior envía a esta vista de playlist que permite su correspondiente administración, a continuación se verán las vistas de los botones dentro de esta:

Vista de nueva playlist


[Dashboard](#)
[Usuarios](#)
[Playlists](#)
[Canciones](#)

Admin User ▾

Crear nueva Playlist

Nombre de la Playlist

Asignar a Usuario

Selecciona un usuario

Portada (imagen)

Seleccionar archivo
Ningún archivo seleccionado

[Crear Playlist](#)
[Cancelar](#)

Pulsando el botón de nueva playlist, se podrá acceder a esta vista donde te da la posibilidad de elegir el usuario al que atribuirle la playlist entre otras cosas, ambos botones de la vista redirigen de nuevo a la vista de playlists.

Vista de edición de playlists

Dashboard
 Usuarios
 Playlists
Canciones
 Admin User

Editar Playlist

Nombre de la Playlist

Usuario

Portada actual:

Cambiar imagen (opcional)
 Ningún archivo seleccionado

Canciones en la playlist

☒ Pretty Devil — Alessandra
 ☒ Exes — Tate McRae
 ☒ Sweet Child O'Mine — Guns N' Roses
 ☒ Bohemian Rhapsody — Queen
 ☐ SICKO MODE — Travis Scott
 ☒ Strobe — deadmau5
 ☐ Thinkin Bout You — Frank Ocean
 ☐ Find your flame — Kellin Quinn & Tyler Smyth

Pulsando el botón de lápiz de la playlist elegida pasará a esta vista donde se podrán elegir las canciones atribuidas entre otras cosas, ambos botones inferiores redirigen a la vista de playlists.

Vista de canciones


Dashboard
 Usuarios
 Playlists
 Canciones
 Admin User

Gestión de Canciones

#	Título	Artista	Álbum	Género	Acciones
1	Pretty Devil	Alessandra	Best Year Of My Life	Rock	
2	Exes	Tate McRae	THINK LATER	Pop	
3	Sweet Child O'Mine	Guns N' Roses	Appetite for Destruction	Rock	
4	Bohemian Rhapsody	Queen	A Night at the Opera	Rock	
5	SICKO MODE	Travis Scott	ASTROWORLD	Hip Hop	
7	Strobe	deadmau5	For Lack of a Better Name	Electrónica	
9	Thinkin Bout You	Frank Ocean	Channel ORANGE	R&B	
10	Find your flame	Kellin Quinn & Tyler Smyth	SF Soundtrack	Rock	

El cuarto botón del menú de navegación superior, saldrá esta vista de administración de las canciones de la base de datos, a continuación se verán las vistas de los botones dentro de esta:

Vista de nueva canción


Dashboard
 Usuarios
 Playlists
 Canciones
 Admin User

Título

Artista

Álbum


Género
Pop

Portada
Seleccionar archivo Ningún archivo seleccionado

Archivo de audio
Seleccionar archivo Ningún archivo seleccionado

Pulsando el botón de nueva canción se podrá acceder a esta vista de creación de canción, ambos botones irán de nuevo a la vista de canciones.

Vista de edición de canciones


Dashboard
 Usuarios
 Playlists
 Canciones
 Admin User


Exes

Artista
Tate McRae

Álbum
THINK LATER

Género
Pop

Portada
Seleccionar archivo Ningún archivo seleccionado

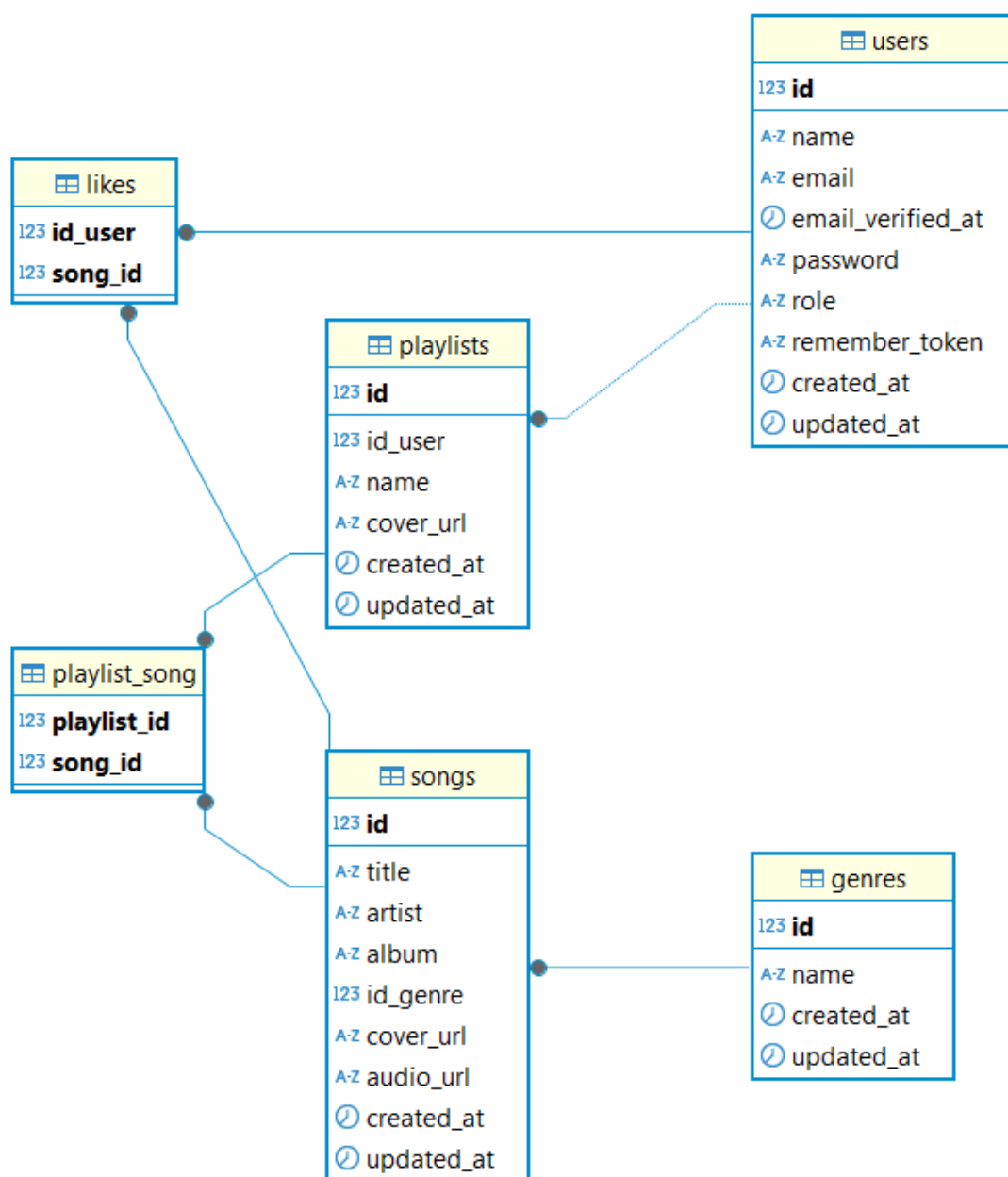


Archivo de audio
Seleccionar archivo Ningún archivo seleccionado

0:00 / 3:07

Pulsando el botón de lápiz de la respectiva canción pasará a esta vista donde estarán puestos los atributos de dicha canción, ambos botones irán de nuevo a la vista de canciones.

10. Modelo de la base de datos



11. Manual de usuario

11.1. ¿Qué es StellarSound?

Es una aplicación móvil que permite a los usuarios explorar una lista de canciones proporcionadas desde un servidor Laravel. Está pensada como un sistema de catálogo musical que se conecta vía API y presenta las canciones con un diseño moderno y funcional.

11.2. Requisitos del sistema

- **Sistema operativo:** Android 8.0 (Oreo) o superior.
- **Conexión a Internet:** Obligatoria para acceder al catálogo.
- **Cuenta de usuario o creación de la misma:** Necesaria para autenticarse.
- **Servidor local:** con docker instalado y una imagen de mysql ejecutándose.
- **Acceso a red local:** El backend debe estar accesible desde la misma red.

11.3. Instalación de la aplicación

- Instalación del proyecto laravel en el servidor local, una vez instalado se crearán las migraciones apuntando a la imagen docker.
- Asegurarse de que tanto el servidor como el móvil están en la misma red, y posteriormente inicia el servicio laravel en la ip del servidor.
- Hacer que la aplicación apunte a la IP del servidor.
- Instalar la apk de la aplicación.

11.4. Primeros pasos

- Generar una cuenta para la aplicación tanto de usuario como de administrador (si es el dueño del servidor).
- Usar la web para crear las primeras canciones o si desea usar el seeder que viene proporcionado en Laravel.

12. Retos encontrados en el desarrollo

1. Consumo de diferentes maneras la api de Laravel y sus métodos en Android Studio, aunque parezca que una vez consigues una conexión satisfactoria y completa sobre un método, puedes aplicarlo a los demás, la verdad es que no se comportan de la misma manera teniendo que hacer cambios importantes por cada uno.
2. El sistema de reproducción de audio, ya debía de hacer uso de cuentas matemáticas para conocer la entidad en reproducción y poder actuar según el estado de la reproducción.
3. Uso de estados en la vista de búsqueda, donde por cada estado de búsqueda la vista cambia dando un mensaje simple de estado de la búsqueda.
4. La capacidad de añadir archivos de audio en la creación de las canciones, ya que PHP no admitía archivos tan grandes.

13. Conclusión

StellarSound representa una solución moderna y eficiente para la gestión y visualización de un catálogo musical en dispositivos Android, integrando de forma segura y escalable un backend Laravel protegido por Sanctum.

Con su autenticación segura, su diseño basado en componentes reutilizables (como RecyclerView con adaptadores personalizados) y su arquitectura cliente-servidor clara, la aplicación no solo proporciona una buena experiencia de usuario, sino que también sienta una base sólida para futuras funcionalidades como reproducción de audio, favoritos o gestión de playlists.

14. Código fuente

<https://github.com/2DAM-pablosal302/Proyecto-Final>