

# MusicStream: App para artistas emergentes

## Trabajo Fin de Grado - Grado en Ingeniería Informática

José Hurtado García

UMU

9 de Julio de 2025

- 1 Introducción
- 2 Objetivos
- 3 Estado del Arte
- 4 Metodología
- 5 Tecnologías Empleadas
- 6 Diseño
- 7 Implementación
- 8 Pruebas
- 9 Conclusiones
- 10 Futuros Pasos

## Contexto

- Música en el día a día.
- Difícil destacar sin inversión.

## Problema

- Algoritmos favorecen a artistas famosos.
- Visibilidad desigual.

## Solución

- Plataforma justa para emergentes.
- Integración con Spotify.



- **Objetivo General:** Desarrollar una aplicación web para artistas emergentes que facilite su promoción y conexión con audiencias.
- **Objetivos Específicos:**
  - Analizar el estado del arte de plataformas de streaming.
  - Diseñar prototipo funcional.
  - Realizar conectividad con aplicaciones musicales.
  - Escalabilidad y usabilidad mediante tecnologías modernas.



## ● Plataformas Existentes:

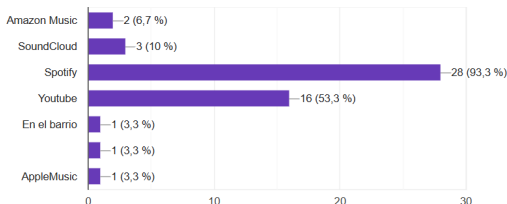
- Spotify.
- Apple Music.
- Amazon Music.
- SoundCloud.

- **MusicStream:** Combina la accesibilidad de SoundCloud con el alcance de Spotify, enfocándose en promoción activa.

## ● Encuesta:

¿En qué plataformas sueles escuchar música?

30 respuestas



- **Fases del Proyecto:**

- ➊ Análisis preliminar: Plataformas y necesidades de usuario.
- ➋ Diseño: Modelado de datos y arquitectura MVC.
- ➌ Implementación: Desarrollo de backend y frontend.
- ➍ Verificación y validación: Pruebas unitarias y funcionales.
- ➎ Despliegue: Uso de Docker y Vercel para entornos escalables.

## Frontend:

- Angular.
- ngx-translate/core

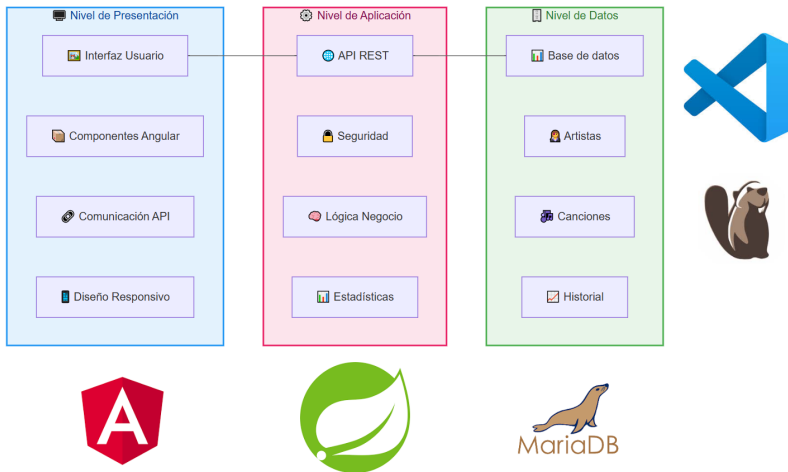
## Backend:

- Spring Boot.
- Spring Security y JWT.
- Maven.
- Spring Boot Mail.

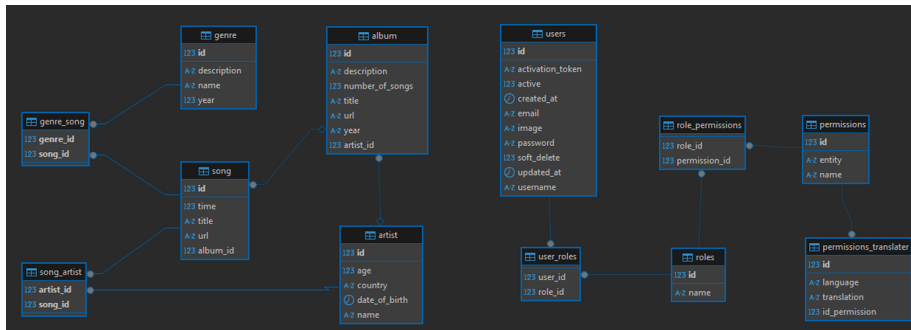
## Base de datos:

- MariaDB.
- DBeaver y HeidiSQL como gestores





- **Patrón MVC:** Separa modelo (datos), vista (interfaz) y controlador (lógica).



- **Tablas Clave:** song, genre, artist, album, role\_permissions, permissions\_translator.
- **JPA:** Mapeo de entidades a base de datos.
- **Internacionalización:** Tabla permissions\_translator para traducción de permisos.

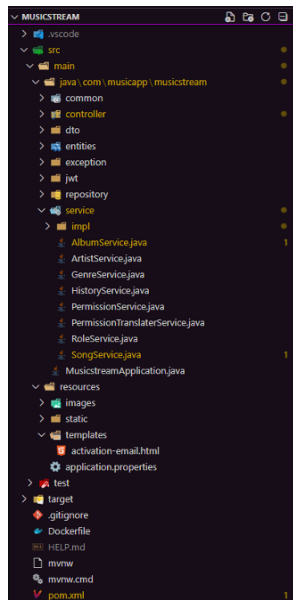
- **Estructura:**

- MusicStreamApplication.java: Punto de entrada.
- service: Lógica de negocio (ArtistService, SongService, etc.).
- controller: Endpoints REST (GET, POST, PUT, DELETE).
- entity: Mapeo de datos con JPA.

- **Seguridad:** JWT para autenticación, Spring Security para autorización.

- **Configuración:**

application.properties para conexión a MariaDB.



- **Persistencia:**

- Entidades
- DTOs
- Repositorios

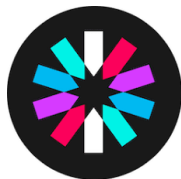
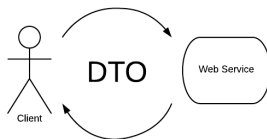
- **Capa de Servicio**

- **Implementación del servicio**

- **Controlador**

- **Seguridad**

- Spring Security
- JWT



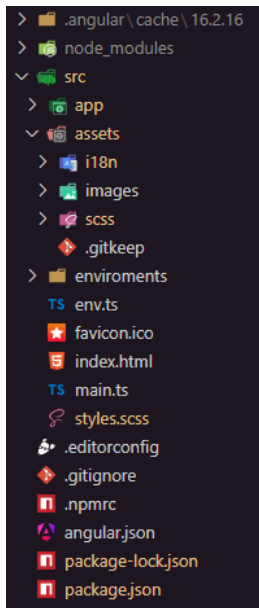
- **Estructura:**

- node\_modules
- src/: Componentes, estilos y servicios.
- package.json

- **Funcionalidades:**

- Registro y login con verificación por correo.
- Reproducción de canciones aleatorias.
- Generación de bingos musicales personalizados.

- **Internacionalización:** Soporte multi-idioma con JSON.




# Implementación: Frontend

Activa tu cuenta de **Music Stream** Recibidos x

**pepeke2000@gmail.com**  
para irte

mar 1 jul, 19:32 (hace 6 días) ☆ 😊 ↶ ⋮



## Activa tu cuenta

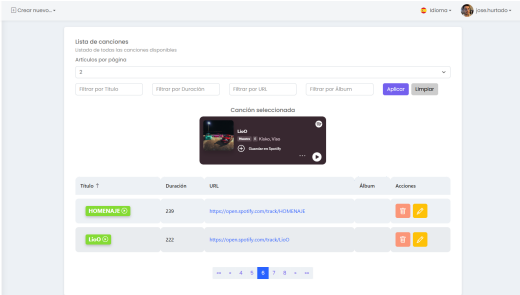
Hola jose hurtado garcia,

Gracias por registrarte con nosotros. Para activar tu cuenta, por favor haz clic en el botón de abajo:

[Activar Cuenta](#)

o haz clic en el siguiente enlace: [http://localhost:3000/activar?token=8c33284\\_2cca-47fa-b43b-e67dedf6e4af](http://localhost:3000/activar?token=8c33284_2cca-47fa-b43b-e67dedf6e4af)

Si no solicitaste esto, por favor ignora este correo electrónico.



Music Stream

Crear nuevo...


Inicio • jose.hurtado

Lista de canciones  
Usando de todos las canciones disponibles  
Artículos por página

2

Filtrar por Título   Filtrar por Duración   Filtrar por URL   Filtrar por Álbum   [Aplicar](#) [Limpiar](#)

Canción seleccionada



Título ↑	Duración	URL	Álbum	Acciones
<a href="#">HOMENAJE</a>	2:30	<a href="https://open.spotify.com/track/HOMENAJE">https://open.spotify.com/track/HOMENAJE</a>		<a href="#">🗑️</a> <a href="#">🔗</a>
<a href="#">Liao</a>	2:22	<a href="https://open.spotify.com/track/Liao">https://open.spotify.com/track/Liao</a>		<a href="#">🗑️</a> <a href="#">🔗</a>

⏮️ ⏪ ⏩ ⏭ 🔍 ⏮️ ⏪ ⏩ ⏭

## 👤 Registrarse

👤 Nombre de usuario

✉ Correo electrónico

🔑 Contraseña

🖼 Foto de perfil

[Seleccionar archivo](#) FotoPerfil.jpg

[👤 Registrarse](#)

¿Ya tienes una cuenta?

[🔑 Iniciar sesión](#)

# Implementación: Frontend

The screenshot shows the MusicStream web application. The top navigation bar includes a logo, a 'Crear nuevo...' button, and a user profile 'jose.hurtado'. A left sidebar contains navigation links: 'Estadísticas', 'Artistas', 'Canciones' (highlighted), 'Canciones aleatorias', 'Bingo', 'Álbumes', and 'Sobre nosotros'. The main content area is titled 'Lista de canciones' and displays a list of songs. A dropdown menu shows '2' items per page. Filter buttons for 'Título', 'Duración', 'URL', and 'Álbum' are present, along with 'Aplicar' and 'Limpiar' buttons. A 'Canción seleccionada' section shows a preview of the song 'Perdido' by Kiko, Evil G. Below this is a table with columns: 'Título ↑', 'Duración', 'URL', 'Álbum', and 'Acciones'. The table lists two songs: 'NO TE VALORA' and 'Perdido'. The 'Perdido' row is highlighted in blue. At the bottom, a pagination bar shows the current page is 4 out of 5.

Music Stream

Crear nuevo...

Idioma - jose.hurtado

Estadísticas

Artistas

Canciones

Canciones aleatorias

Bingo

Álbumes

Sobre nosotros

Lista de canciones

Listado de todas las canciones disponibles

Artículos por página

2

Filtrar por Título

Filtrar por Duración

Filtrar por URL

Filtrar por Álbum

Aplicar

Limpiar

Canción seleccionada

Perdido

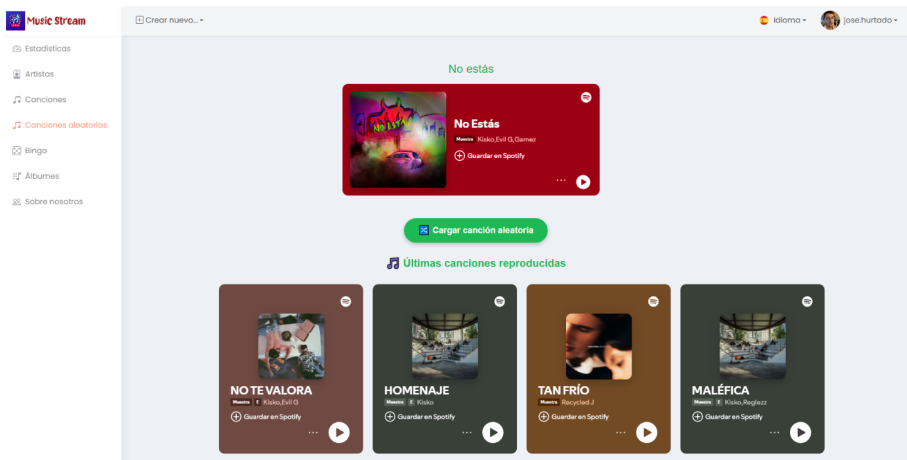
Álbum: Kiko, Evil G

Guardar en Spotify

Título ↑	Duración	URL	Álbum	Acciones
NO TE VALORA	251	<a href="https://open.spotify.com/track/NO TE VALORA">https://open.spotify.com/track/NO TE VALORA</a>		
Perdido	200	<a href="https://open.spotify.com/track/Perdido">https://open.spotify.com/track/Perdido</a>		

1 2 3 4 5

# Implementación: Frontend

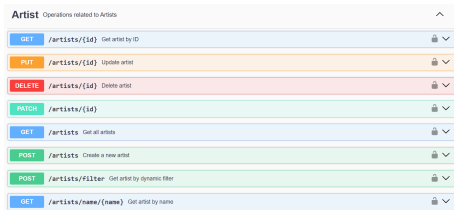




## • Tipos de Pruebas:

- Unitarias: Verificación de controladores (ej., ArtistController).
- Funcionales: Pruebas con Postman para endpoints REST.

```
public class ArtistControllerTest {  
  
    @Mock  
    private ArtistRepository artistRepository;  
  
    @Mock  
    private DTOUtils dtoUtil;  
  
    @InjectMocks  
    private ArtistController artistController;  
  
    private List<Artist> artists;  
    private List<ArtistDTO> artistDTOs;  
    private Artist artist;  
    private ArtistDTO artistDTO;  
  
    @BeforeEach  
    public void setUp() {
```



- **Herramientas:**

- Vercel: Despliegue del frontend
- Docker: Contenerización para portabilidad.
- Render: Despliegue y la gestión de aplicaciones backend
- Supabase: Herramientas y servicios para construir aplicaciones

- **Beneficios:** Escalabilidad, consistencia y facilidad de mantenimiento.

- **Prototipo Funcional:** Aplicación web completa con integración a Spotify.
- **Visibilidad Mejorada:** Herramientas para artistas emergentes (canciones aleatorias y bingo musical).
- **Usabilidad:** Interfaz intuitiva y multi-idioma.
- **Escalabilidad:** Arquitectura modular y despliegue con Docker.
- **Aprendizajes:** Competencias en full-stack, diseño de bases de datos, y gestión ágil.
- **Demo:** [pepehurtado.vercel.app](https://pepehurtado.vercel.app).

- Migración a entornos cloud más robustos (AWS, Google Cloud).
- Implementación de algoritmos de recomendación personalizados.
- Estrategia de marketing digital para captar usuarios y artistas.
- Plan de negocio para monetización y colaboraciones estratégicas.

- A mi tutor, Francisco García Sánchez, por su apoyo y orientación.
- A mi familia y amigos por su soporte incondicional.
- A todos los que respondieron la encuesta, aportando valiosas perspectivas.

¡Gracias por su atención! ¿Preguntas?