



DEDICATORIA

Dedico este trabajo, fruto de años de esfuerzo y dedicación, a mi familia, cuyo apoyo incondicional ha sido fundamental en mi trayectoria universitaria. Sin su respaldo, este camino, que hoy se acerca a su conclusión, no habría sido posible.

También quiero expresar mi profundo agradecimiento a mi pareja, quien en innumerables ocasiones me ayudó a aliviar el estrés y la ansiedad previos a los exámenes, así como en aquellos momentos en los que la frustración por no encontrar una solución parecía insuperable.

Por último, una mención especial para mi gato, Tiger, cuya compañía silenciosa y reconfortante me ha recordado, incluso en los momentos de mayor caos, la importancia de tomar pausas y cuidar de mí mismo.

A todos ustedes, mi más sincero agradecimiento.

PROLOGO

El camino del desarrollo de software es tanto un desafío técnico como un proceso de aprendizaje constante. A lo largo de mi formación, este proyecto ha representado no solo la culminación de mis estudios, sino también una oportunidad para aplicar, consolidar y ampliar mis conocimientos. **SoundTribe** nace con la intención de ofrecer una plataforma en la que artistas y oyentes puedan interactuar de manera orgánica, generando un ecosistema en el que la música no solo se comparte, sino que también se valora y se convierte en un motor de crecimiento para quienes la crean.

Este trabajo es el resultado de innumerables horas de desarrollo, investigación y resolución de problemas, donde cada decisión tomada ha estado respaldada por fundamentos técnicos, pero también por una visión clara de lo que significa construir software con impacto real.

A través de estas páginas, compartiré el proceso de diseño, arquitectura y desarrollo de SoundTribe, explicando las decisiones tecnológicas adoptadas y los desafíos que enfrenté en el camino.

PREFACIO

Este documento surge como parte de mi **práctica supervisada**, un trabajo final que marca el cierre de mi formación académica en desarrollo de software. A lo largo de estos años, he adquirido habilidades en diversas tecnologías y metodologías, y este proyecto es el reflejo de ese aprendizaje aplicado a una problemática real.

SoundTribe busca solucionar un problema clave dentro de la industria musical independiente: brindar a los artistas una plataforma para compartir su música, conectar con su comunidad y monetizar su trabajo, al mismo tiempo que los oyentes pueden descubrir contenido relevante y apoyar a sus creadores favoritos.

El enfoque de desarrollo se basó en la implementación de **microservicios**, con una arquitectura robusta y escalable, utilizando **Java con Spring Boot en el backend, Angular en el frontend y PostgreSQL como base de datos**. Además, se incorporaron sistemas de autenticación, almacenamiento distribuido y gestión de pagos.

Este documento no solo expone los aspectos técnicos del desarrollo, sino que también refleja el esfuerzo y la dedicación invertidos en llevar esta idea desde una simple conceptualización hasta un software funcional.

INTRODUCCION

El desarrollo de plataformas digitales que conectan creadores de contenido con su audiencia ha crecido exponencialmente en los últimos años. En este contexto, **SoundTribe** surge como una solución destinada a artistas y oyentes, proporcionando un espacio en el que la música no solo se disfruta, sino que también se convierte en una oportunidad para fortalecer comunidades y generar ingresos.

El presente documento detalla el proceso de desarrollo de SoundTribe, desde su concepción hasta su implementación. Se abordarán aspectos clave como el diseño de la arquitectura, la selección de tecnologías, la construcción de microservicios y la integración de funcionalidades esenciales para su funcionamiento.

Este proyecto representa el cierre de mi etapa formativa y la transición hacia el mundo profesional. A través de esta experiencia, he podido enfrentar desafíos reales, tomar decisiones críticas en el desarrollo y consolidar mis habilidades en ingeniería de software.

A lo largo de los siguientes capítulos, se expondrán los fundamentos del proyecto, las decisiones técnicas adoptadas y los aprendizajes obtenidos en el proceso.

DEFINICIÓN DEL PROYECTO.....	4
Objetivo.....	4
Visión.....	4
Misión.....	4
Roles de Usuario.....	4
Visitante.....	4
Oyentes.....	4
Artistas.....	5
Administrador.....	5
Funcionalidades Principales.....	5
Página Principal.....	5
Sistema de Puntuación de Canciones.....	5
Monetización y Comercio.....	6
Arquitectura y Tecnologías.....	6
Arquitectura General.....	6
Tecnologías Utilizadas.....	6
Microservicios.....	7
Gestión de Usuarios.....	7
Subscripción a Artistas.....	7
Gestión de Música y Contenidos.....	7
Gestión de Posteos.....	7
Gestión de Ventas y Pagos.....	7
Seguridad y Moderación.....	7
DEFINICIÓN TÉCNICA.....	8
Diagrama de Clase y de Base de Datos.....	8
Diagrama de Estado.....	9
Pedido.....	9
Máquina de estados de un pedido físico.....	9
Casos de revisión y disputas.....	9
SoundTribe.....	1

DEFINICIÓN DEL PROYECTO

A continuación, compartiré los deseos y objetivos que tengo en relación con esta aplicación, tanto a nivel técnico como de desarrollo. Sin embargo, debo aclarar que estos aspectos no necesariamente serán considerados dentro del alcance del proyecto actual. Como desarrollador comprometido, me esforzaré por incorporar todos los elementos posibles que menciono a continuación, pero si alguna de estas ideas resulta lo suficientemente compleja como para afectar el avance de un sprint o el cumplimiento de un periodo de entrega, consideraré posponer su implementación para una futura actualización.

Objetivo

Este proyecto tiene como objetivo desarrollar una plataforma digital que permita a los artistas compartir su música, interactuar con su comunidad y comercializar productos digitales y físicos. Al mismo tiempo, brinda a los oyentes la posibilidad de descubrir nueva música, apoyar a sus artistas favoritos y realizar compras de contenido exclusivo.

Visión

Convertirse en un ecosistema integral para la industria musical independiente, proporcionando herramientas innovadoras para la distribución de contenido, monetización y conexión con la audiencia.

Misión

Facilitar el acceso a la música y el crecimiento de artistas emergentes mediante una plataforma inclusiva, segura y eficiente.

Roles de Usuario

Visitante

Los visitantes son usuarios no registrados que pueden explorar la plataforma con funcionalidades limitadas. Se les permite buscar y reproducir música, pero no pueden participar

en interacciones sociales ni realizar compras. Se les incentivará a registrarse para acceder a todas las funcionalidades.

Oyentes

Los oyentes son usuarios registrados que pueden interactuar con la música y los artistas. Sus principales funcionalidades incluyen:

- Buscar y reproducir música.
- Comentar en publicaciones públicas y privadas (si son seguidores del artista).
- Suscribirse a artistas para recibir actualizaciones.
- Comprar merchandising digital y físico.
- Realizar donaciones a artistas que tengan habilitada esta función.

Artistas

El rol de artista es central en la plataforma, ya que generan el contenido que consumen los oyentes. Sus principales funcionalidades incluyen:

- Subir música a la plataforma.
- Crear publicaciones con texto, encuestas y contenido multimedia.
- Comentar en sus propios posteos y en los de otros artistas.
- Vender productos digitales y físicos.
- Configurar la privacidad de sus publicaciones (públicas o privadas para suscriptores).

Administrador

El administrador es responsable de moderar la plataforma y gestionar conflictos entre usuarios. Sus tareas incluyen:

- Suspender temporalmente a artistas en caso de incumplimiento de normas.
- Gestionar denuncias (por ejemplo, reembolsos a oyentes en caso de incumplimiento en la entrega de productos).
- Velar por el cumplimiento de las políticas de la plataforma.

Funcionalidades Principales

Página Principal

La página principal mostrará un resumen de la actividad en la plataforma, incluyendo:

- **Top 10 canciones más gustadas.**

- **Top 10 canciones más recientes.**
- **Búsqueda por etiquetas:** Permite filtrar por género, subgénero y estilo.

Sistema de Puntuación de Canciones

El sistema de puntuación ayudará a destacar las canciones más populares y mejorar las recomendaciones para los oyentes.

Criterios de Puntuación:

- **Si se escuchan 30 segundos:** +2 puntos.
 - Si se da "me gusta" o "favorito": +2 puntos cada uno.
 - Si se da "no me gusta": -3 puntos.
- **Si se escucha completa:** +3 puntos.
 - Si se da "me gusta" o "favorito": +3 puntos cada uno.
 - Si se da "no me gusta": -4 puntos.
- **Si no se escucha, pero se da "me gusta" o "favorito":**
 - Si se da "me gusta" o "favorito": +0.5 puntos cada uno.
 - Si se da "no me gusta": -0.5 puntos.

Monetización y Comercio

- **Venta de productos digitales y físicos:** Los artistas pueden vender álbumes digitales y discos físicos.
- **Pagos y comisiones:** La plataforma retiene un 5% de comisión por cada transacción.
- **Control de entregas:** La plataforma gestiona el pago y solo libera los fondos cuando el producto ha sido entregado al comprador. En caso de una compra digital (album digital) el producto se entrega automáticamente en el mail del comprador

Arquitectura y Tecnologías

Arquitectura General

La plataforma estará basada en una arquitectura de microservicios para garantizar escalabilidad y mantenimiento eficiente.

Tecnologías Utilizadas

- **Backend:** Java, Spring Boot, con implementación de microservicios.
- **Frontend:** Angular para una experiencia de usuario moderna y responsiva.
- **Base de Datos:** PostgreSQL para almacenamiento estructurado de datos.
- **Almacenamiento:** MiniO para almacenamiento distribuido de archivos.

Microservicios

La plataforma se dividirá en varios microservicios independientes:

Gestión de Usuarios

- Registro, inicio de sesión y autenticación.
- Manejo de perfiles.
- Configuración de preferencias y seguridad.

Subscripción a Artistas

- Seguimiento de artistas.
- Notificaciones de nuevos lanzamientos y publicaciones.

Gestión de Música y Contenidos

- Subida y almacenamiento de música.
- Streaming de canciones.
- Sistema de puntuación y recomendaciones.

Gestión de Posteos

- Creación de publicaciones y encuestas.
- Comentarios y respuestas.

Gestión de Ventas y Pagos

- Venta de productos físicos y digitales.
- Procesamiento de pagos.
- Control de entregas.

Gestión de Chat

Seguridad y Moderación

- **Protección de datos personales** con cifrado de contraseñas y medidas de seguridad avanzadas.

- **Moderación automatizada y manual** para prevenir abusos y estafas.

DEFINICIÓN TÉCNICA

Como ya mencionamos, este proyecto se basará en tecnologías con las que estoy familiarizado y que me resultan de interés, lo suficiente como para seguir aprendiendo más sobre ellas. Si deseas conocer en detalle las bases de las herramientas y tecnologías que estoy utilizando, te invito a consultar el apartado de **Arquitectura y Tecnologías** (pág. N).

En este capítulo, me dedicaré a explicar mis principales diagramas y reflexiones sobre cómo deberé implementar el desarrollo, basándome en diagramas UML y documentación que pueda servir para un entendimiento público. Es probable que, durante el desarrollo de la aplicación, esta página reciba actualizaciones constantes, ya que pueden surgir errores en los diagramas o posibles malentendidos sobre cómo deberían funcionar ciertos aspectos. Prefiero hacerlo de este modo para que el lector siempre encuentre información fresca y actualizada acerca de cómo funciona el código y de cómo planteo la arquitectura y el funcionamiento.

Aquel lector que lea este capítulo después de la finalización del desarrollo será premiado por no tener que revisar constantemente este apartado para entender mi proceso mental y, sobre todo, por no tener que leer mis divagaciones plasmadas en forma de documentación.

Diagrama de Clase y de Base de Datos

Es importante tener en cuenta que, al tratarse de un proyecto basado en microservicios, el diagrama de clases estará dividido en varias partes: **Gestión de Usuarios, Suscripción a Artistas, Gestión de Música y Contenidos, Gestión de Posteos, Ventas y Pagos, y Gestión de Chat**. Con esto en mente, comenzaré mencionando que me centraré en plasmar las clases más relevantes. Cabe destacar que, en muchos casos, una clase podrá hacer referencia a otra clase que no aparecerá en la misma imagen. Esto se debe a la naturaleza de la arquitectura de microservicios. Sin embargo, dejaré una aclaración para que la documentación sea más comprensible y el lector pueda seguirla con mayor claridad.

Además, es importante mencionar que la estructura que se mantendrá será la siguiente: **Diagrama de Base de Datos → Diagrama de Clases**.

Diagrama de Estado

Pretendo manejar ciertos aspectos en forma de estados, ya que resulta más fácil visualizarlos y gestionarlos de esta manera. La estructura de este apartado será la siguiente: primero mencionaré una problemática relacionada con una clase en particular, luego explicaré cómo abordaré este problema, acompañado de una imagen que respalde lo expuesto previamente.

Pedido

Es completamente normal querer realizar un pedido, ya sea para comprar merchandising o adquirir un álbum. Sin embargo, la gestión del pedido varía dependiendo de si se trata de una compra digital o física. En el caso de un **pedido digital**, no será necesario utilizar una máquina de estados para su control, ya que el proceso se realiza automáticamente (este tema será abordado en la página N).

Por otro lado, nos enfocaremos en la **máquina de estados de un pedido físico**. Cabe destacar que los pedidos realizados a través de la plataforma **no son gestionados directamente por SoundTribe**, sino por el artista que recibe la compra. SoundTribe proporciona a los artistas un sistema para publicar y vender sus productos, ofreciendo visibilidad y alcance a través de la plataforma. Nuestro rol es facilitar el procesamiento del pago, el control del pedido y, finalmente, la entrega del dinero al artista, descontando una **comisión del 5%**. Este porcentaje es necesario para mantener la plataforma operativa y, a su vez, es lo suficientemente bajo en comparación con la competencia para asegurar que los artistas obtengan la mayor ganancia posible.

Máquina de estados de un pedido físico

El proceso de un pedido físico sigue la siguiente lógica:

1. Un **oyente** realiza un pedido, el cual entra en estado de **ESPERA**, a la espera de la respuesta del artista.
2. Si el artista confirma que preparará el pedido, el estado cambia a **PREPARACIÓN**.
3. Una vez que el artista envía el producto al oyente, el pedido pasa al estado **VIAJANDO**.
4. Cuando el oyente recibe el producto y actualiza el pedido, el estado cambia a **ENTREGADO**, finalizando así el flujo principal del proceso.

Casos de revisión y disputas

Sin embargo, existen situaciones en las que los estados no se actualizan correctamente, ya sea por olvido o porque el pedido **realmente no ha sido procesado**. En estos casos, el pedido entra en estado de **REVISIÓN**.

Un pedido puede ingresar en **REVISIÓN** desde cualquier estado, excepto **ENTREGADO** o **CANCELADO**. Cuando esto ocurre, un **administrador** interviene en una sala de chat donde estarán involucrados los tres actores principales: el **administrador**, el **oyente** y el **artista**. El administrador tendrá la autoridad para hacer cumplir las normas de la comunidad y tomar una decisión que resuelva el conflicto de manera justa y eficiente.

Las posibles resoluciones del estado **REVISIÓN** son:

- **El pedido sigue en pie:** En este caso, el estado del pedido regresa al estado anterior.
- **El pedido es cancelado:** El oyente recibe un **reembolso** total y el artista es **suspendido**.

Un mismo pedido puede entrar en **REVISIÓN** hasta tres veces. Si llega a una tercera revisión, la plataforma cancelará automáticamente el pedido, devolverá el dinero al oyente y aplicará sanciones definitivas al artista involucrado.

