

1. Introducción del Proyecto

Descripción breve de la temática elegida

El proyecto consiste en el desarrollo de una aplicación cliente-servidor que permite a los usuarios explorar información de álbumes musicales utilizando la API de Spotify y publicar reseñas (reviews) sobre ellos.

Los usuarios podrán puntuar álbumes, dejar comentarios, asignar etiquetas y leer las opiniones de otras personas.

El sistema combina datos externos provenientes de Spotify con información generada por la comunidad de usuarios.

Objetivo general del MVP

Construir un MVP funcional que permita:

- Autenticación y gestión básica de usuarios
- Buscar y visualizar álbumes utilizando la API de Spotify
- Crear, leer y listar reseñas de álbumes
- Mostrar información detallada del álbum (nombre, artistas, fecha de lanzamiento, imagen, cantidad de tracks, etc.) obtenida desde Spotify
- Permitir a cualquier usuario ver las reviews realizadas por otros usuarios

El objetivo es integrar datos externos con contenido propio almacenado en nuestra base de datos, aplicando una arquitectura en microservicios y buenas prácticas de desarrollo.

API externa seleccionada

[Spotify Web API](#)

La API de Spotify permite acceder a información completa sobre artistas, álbumes, pistas, géneros, portadas y más.

En este proyecto se utilizará principalmente para:

- Obtener detalles de álbumes.
- Listar los artistas relacionados.
- Acceder a imágenes oficiales de los álbumes
- Buscar álbumes por nombre, artista o palabra clave.

Estos datos se combinan con las reviews creadas por los usuarios dentro de nuestra propia aplicación.

Motivación

Si bien Spotify ofrece información completa sobre artistas y álbumes, **no permite que la comunidad deje reseñas o comentarios** sobre los álbumes.

Nuestro proyecto surge para cubrir esta necesidad:

- Permite a los usuarios compartir opiniones reales sobre álbumes
- Centraliza puntuaciones, comentarios y tags creados por la comunidad.
- Crea un espacio social donde se pueden descubrir nuevos álbumes según reviews de otros usuarios.
- Combina datos oficiales (de Spotify) con contenido generado por usuarios para enriquecer la experiencia musical

En resumen, el sistema brinda una funcionalidad social y participativa que actualmente no existe dentro de Spotify.

2. Requisitos Funcionales

2.1 Lista de Funcionalidades del MVP

Registro y Gestión de Usuarios

- **RF01:** El sistema debe permitir registrar nuevos usuarios.
- **RF02:** El sistema debe permitir iniciar sesión / autenticarse.
- **RF03:** El sistema debe permitir asociar las reviews realizadas a cada usuario.

Búsqueda de Álbumes (Spotify API)

- **RF04:** El sistema debe permitir buscar álbumes por nombre, artista o palabra clave utilizando la API de Spotify.
- **RF05:** El sistema debe mostrar los resultados obtenidos desde Spotify con información del álbum.
- **RF06:** El sistema debe permitir seleccionar un álbum y ver su información detallada.

Visualización de Detalles de Álbum

- **RF07:** El sistema debe mostrar la información completa del álbum (nombre, artistas, fecha de lanzamiento, tracks, portada) obtenida desde la API de Spotify.
- **RF08:** El sistema debe mostrar todas las reviews asociadas a un álbum.
- **RF09:** El sistema debe permitir visualizar el puntaje promedio de un álbum.

Creación de Reviews

- **RF10:** El sistema debe permitir que los usuarios creen una review sobre un álbum.
- **RF11:** El sistema debe permitir asignar un puntaje numérico a la review.
- **RF12:** El sistema debe permitir agregar un comentario o descripción en la review.
- **RF13:** El sistema debe permitir agregar tags a la review.

2.2 Historias de Usuario

HU01 – Registro de Usuario

Descripción

Como usuario nuevo, quiero poder registrarme en la aplicación para crear una cuenta y acceder a las funcionalidades del sistema.

Criterios de aceptación

- *Debo poder ingresar nombre, apellido, email y contraseña*
- *Si el email ya está registrado, el sistema debe mostrar un mensaje de error.*
- *La cuenta debe crearse correctamente y almacenarse en la base de datos.*

Escenario / Caso de uso

1. *El usuario abre la pantalla de registro.*
2. *Completa el formulario con sus datos.*
3. *Presiona “Registrarme”.*
4. *El sistema valida los datos y crea el usuario.*
5. *El sistema muestra un mensaje de confirmación o inicia sesión automáticamente.*

HU02 – Buscar álbumes

Descripción

Como usuario, quiero buscar álbumes por nombre o artista para encontrar el álbum que deseo revisar o explorar.

Criterios de aceptación

- Debe existir un campo de búsqueda en la app
- La búsqueda debe consultar la API de Spotify.
- Debe mostrar una lista de álbumes con título, artista y portada.
- Si no hay resultados, debe mostrar un mensaje claro.

Escenario / Caso de uso

1. El usuario ingresa un texto en el campo de búsqueda
2. El sistema llama a la API de Spotify.
3. Se muestran los álbumes coincidentes.
4. El usuario selecciona uno para ver más detalles.

HU03 – Crear review de un álbum

Descripción

Como usuario autenticado, quiero escribir una review sobre un álbum para compartir mi opinión con otros usuarios.

Criterios de aceptación

- El usuario debe estar logueado.
- Debo poder asignar un puntaje.
- Debo poder escribir un comentario.
- Debo poder agregar tags (opcional).
- La review debe guardarse asociada al usuario y al álbum.
- Después de crearla, debe aparecer en la lista de reviews del álbum.

Escenario / Caso de uso

1. El usuario abre un álbum.
2. Presiona “Agregar review”.
3. Completa puntaje, comentario y tags.
4. Presiona “Enviar”.
5. El sistema guarda la review y la muestra en la sección de reviews del álbum.

HU04 – Ver reviews de un álbum

Descripción

Como usuario, quiero ver las reviews que otros usuarios hicieron de un álbum para conocer sus opiniones.

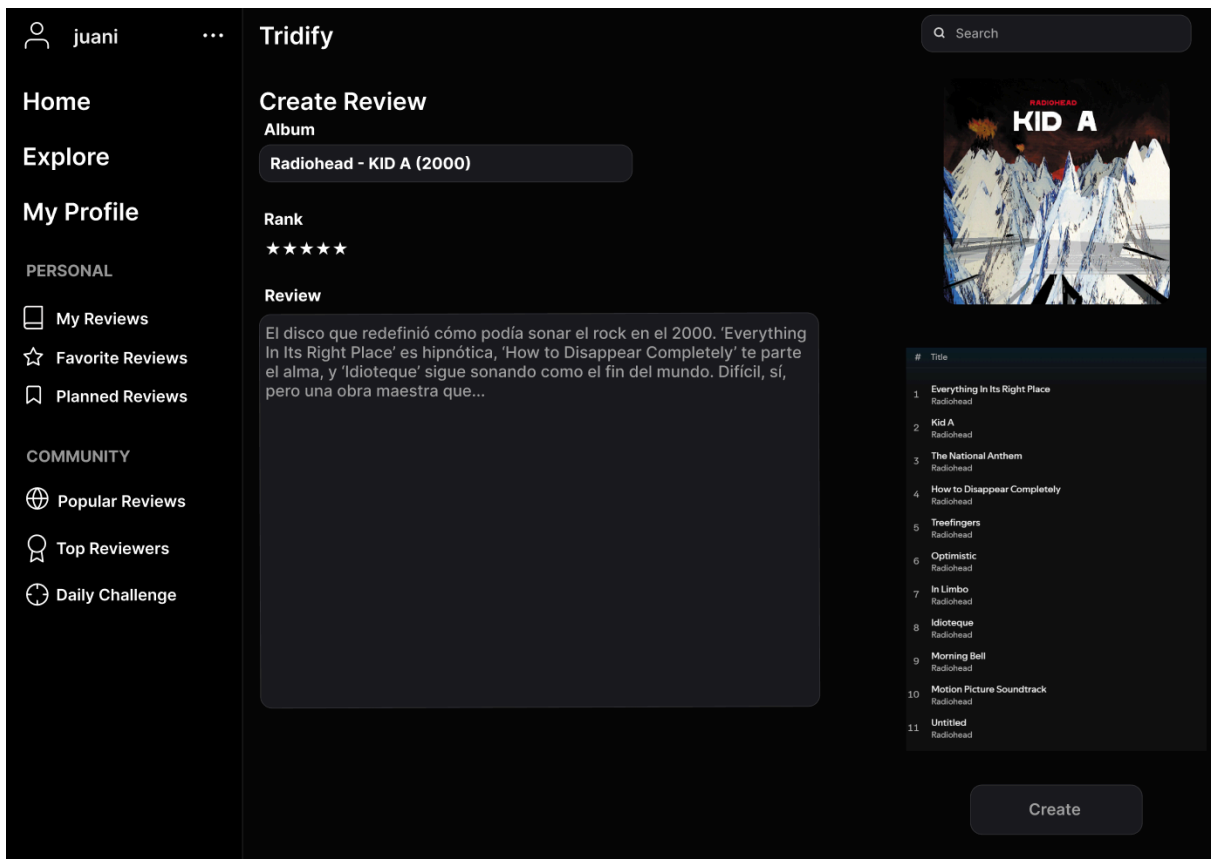
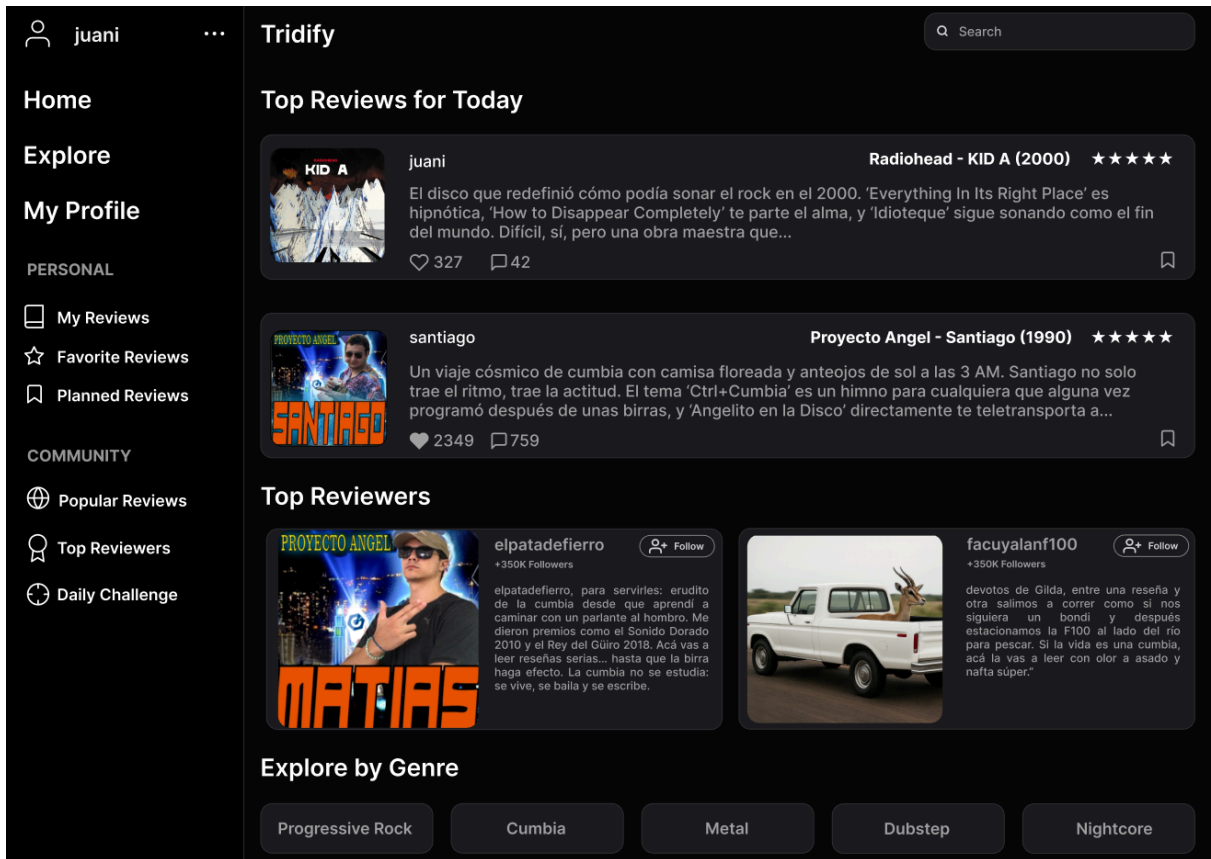
Criterios de aceptación

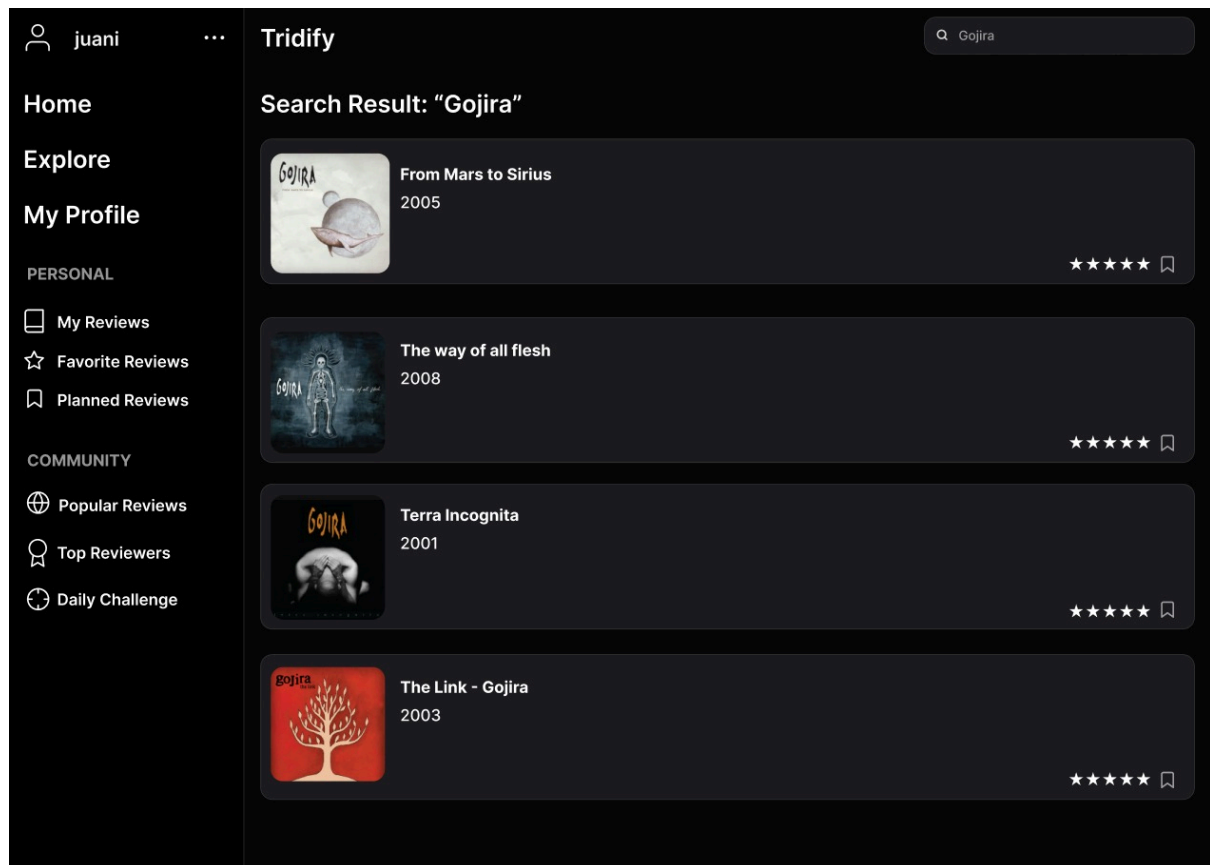
- Debe mostrarse una lista de reviews para cada álbum
- Cada review debe incluir usuario, comentario, puntaje y tags.
- Debe actualizarse automáticamente cuando se agregan nuevas reviews.
- Debe calcularse el puntaje promedio.

Escenario / Caso de uso

1. El usuario abre la página de un álbum.
2. El sistema obtiene todas las reviews del backend.
3. Se muestran ordenadas por fecha o relevancia.
4. El usuario puede leer todas las opiniones.

2.3 Prototipos / Wireframes





3. Requisitos No Funcionales

- Escalabilidad:
 - El sistema debe permitir agregar nuevas funcionalidades a futuro (por ejemplo: likes en reviews de álbumes y chat entre usuarios).
- Rendimiento:
 - La carga de la lista de reseñas de un álbum debe tardar menos de 2 segundos para una cantidad típica de reseñas (100).
- Usabilidad:
 - La interfaz debe ser simple con textos y controles claramente legibles.
 - Los mensajes de error deben ser comprensibles para el usuario e indicar, cuando sea posible, como corregir el problema.
- Seguridad:
 - Toda la comunicación entre frontend y backend debe realizarse mediante protocolo HTTPS para proteger los datos del usuario en tránsito.
- Mantenibilidad:
 - El código debe seguir una estructura de carpetas y convenciones de nombres coherentes, tanto en frontend como en backend, para facilitar el mantenimiento.
- Confiabilidad:
 - Una vez que una reseña se guarda correctamente, el sistema debe garantizar su persistencia y mostrar nuevamente al recargar la página.
 - En caso de fallo al guardar una reseña, el sistema debe informar al usuario.

4. Modelado del Dominio

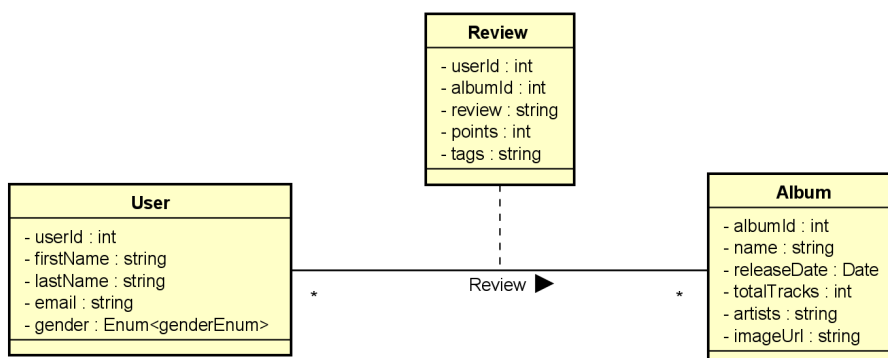
El siguiente modelo de dominio representa las entidades principales del sistema. Cabe destacar que este modelo es el necesario para cumplir el MVP planteado, pero está abierto a la extensión y el agregado de funcionalidades.

4.1 Diagrama de Clases

El modelo está compuesto por tres entidades principales:

- **User**
- **Album**
- **Review**

A continuación se presenta el diagrama de clases correspondiente:



4.2 Descripción de Entidades

Entidad: User

Representa a los usuarios registrados en el sistema, quienes pueden crear reviews de distintos álbumes.

Atributos:

- **userId : int** — Identificador único del usuario.
- **firstName : string** — Nombre del usuario.

- `lastName : string` — Apellido del usuario.
- `email : string` — Correo electrónico utilizado para registrar la cuenta.
- `gender : Enum<genderEnum>` — Género del usuario.

Entidad: Album

Representa los álbumes obtenidos desde la API de Spotify.

Aunque los datos provienen de Spotify, los álbumes que reciben reviews se almacenan localmente para su referencia.

Atributos:

- `albumId : int` — Identificador único.
- `name : string` — Nombre del álbum.
- `releaseDate : Date` — Fecha de lanzamiento.
- `totalTracks : int` — Cantidad total de canciones del álbum.
- `artists : string` — Nombre del artista o artistas.
- `imageUrl : string` — URL de la imagen del álbum.

Entidad: Review

Representa la reseña generada por un usuario sobre un álbum específico.

Atributos:

- `userId : int` — Usuario que realizó la review.
- `albumId : int` — Álbum al que pertenece la review.
- `review : string` — Comentario o texto descriptivo.
- `points : int` — Puntaje asignado por el usuario.
- `tags : string` — Palabras clave o etiquetas asociadas a la review.