Manual Técnico – The Cassette

Índice

---

1. Introducción

La aplicación “The Cassette” es un software desarrollado en Flutter que permite la búsqueda, almacenamiento y gestión de discos musicales utilizando la API de Discogs. Su objetivo es facilitar a los coleccionistas y amantes de la música el acceso rápido a información de discos, además de permitir la organización de colecciones personales.

2. Análisis del problema

2.a Problemática

Actualmente, los coleccionistas de discos deben gestionar sus colecciones de manera manual o mediante herramientas no especializadas. La falta de integración con bases de datos de música dificulta la obtención de información detallada sobre cada disco.

2.b Clientes potenciales

- Coleccionistas de vinilos y CDs.
- Melómanos interesados en la gestión de su música.
- Tiendas de discos que deseen llevar un registro digitalizado de su inventario.

2.c Análisis DAFO

Debilidades: Dependencia de la API de Discogs, posible limitación de peticiones. Opciones insuficientes en la aplicación.

Amenazas: Competencia con otras aplicaciones similares, incluida la aplicación de Discogs.

Fortalezas: Integración con una de las bases de datos más completas sobre discos. Aplicación sencilla e interfaz intuitivo.

Oportunidades: Expansión a nuevas funcionalidades como recomendaciones personalizadas, lista de deseos, enlaces de compra, etc.

2.d Monetización y beneficios

- Enlace de afiliados para compra de discos online.
- Sincronización con servicios de streaming.
- Venta de discos de segunda mano a través de la app o enlazando con la aplicación de Discogs.

3. Diseño de la solución

3.a Tecnologías elegidas

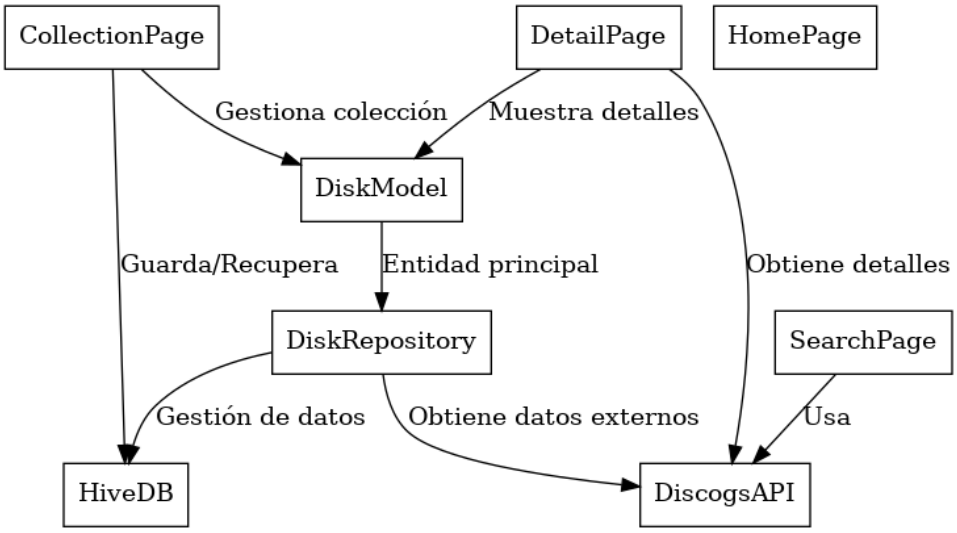
- **Flutter**: Desarrollo multiplataforma.
- **Hive**: Base de datos ligera para almacenamiento local.
- **Discogs API**: Fuente de información sobre discos.

3.b Arquitectura

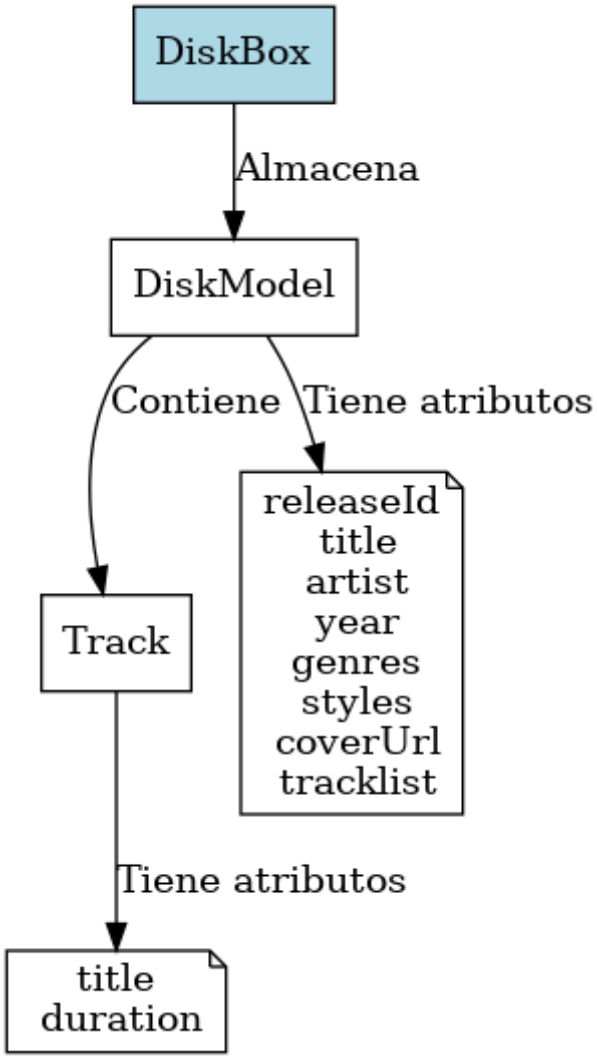
La aplicación sigue una arquitectura basada en **gestión de estados y separación de capas**:

- **Capa de Presentación:** Interfaces de usuario (UI) y widgets.
- **Capa de Negocio:** Repositorios y gestión de datos.
- **Capa de Datos:** API externa y base de datos local.

3.c Diagrama de clases



3.d Diagrama E/R



3.e Consideraciones técnicas

- Optimización de llamadas a la API para evitar límites de peticiones.
- Manejo eficiente del almacenamiento en Hive.
- Implementación de paginación para mejorar la carga de resultados.

4. Documentación de la solución

El código fuente está organizado en el directorio **lib/** y estructurado de la siguiente manera:

- main.dart: Punto de entrada de la aplicación.
- app/api/discogs_api.dart: Módulo de comunicación con la API de Discogs.
- app/database/hive_db.dart: Gestión de almacenamiento local con Hive.
- app/models/disk_model.dart: Definición del modelo de disco.

- app/pages/: Contiene las pantallas principales como búsqueda, colección y detalle.
- app/repositories/disk_repository.dart: Capa de acceso a datos.
- app/routes/routes.dart: Definición de rutas de navegación.
- app/theme/theme.dart: Configuración del tema visual.
- app/widgets/menu_lateral.dart: Widget del menú lateral.

5. Enlaces de interés

- Documentación de Discogs API
- Flutter Docs