

Ingeniería Inversa

Analizar el código fuente del repositorio. Resolver las siguientes preguntas:

- Cuál es la estructura general del proyecto
- Cómo es el flujo de la información. Qué sucede desde que se hace click en search hasta que se muestran los datos en la UI.
- Cuál es la fuente de los datos. Qué información se obtiene de los servicios, cuáles se usan en el proyecto y cuáles se ignoran.
- Dónde y por qué se setea [*].
- Imágenes por defecto: condiciones, fuente de la imagen.
- Analizar los edge cases: song no encontrada, sin conexión a internet.
- Qué librerías externas se utilizan.

SOLID

Realizar los siguientes cambios en el paquete "home" cumpliendo con los principios SOLID:

La clase "Song" tiene una propiedad "year" que se deriva de "releaseDate" y se muestra en la descripción de la canción resultado.

Se desea realizar el siguiente cambio:

- Los resultados de Spotify traen una propiedad llamada "release_date_precision". Agregarla al modelo y guardarla en la base de datos. Eliminar la propiedad year.
- Modificar la descripción de la canción, mostrando la fecha de lanzamiento de la siguiente manera:
 - Si la precisión es "day" mostrar: "Release date: 22/11/1968"
 - Si la precisión es "month" mostrar: "Release date: March, 1969"
 - Si la precisión es "year" mostrar: "Release date: 1937 (not a leap year)"

Ejemplos de búsqueda:

Search: "Little Wing"

"release_date" : "1967-12-01",

"release_date_precision" : "day"

Search: "My Way Sinatra"

"release_date" : "1969-03",

"release_date_precision" : "month"

Search "Robert Johnson"

"release_date" : "1937",

"release_date_precision" : "year"

Refactor Inicial (Clean Code)

Realizar los cambios necesarios en el paquete `otherdetails` para que sea un código limpio.

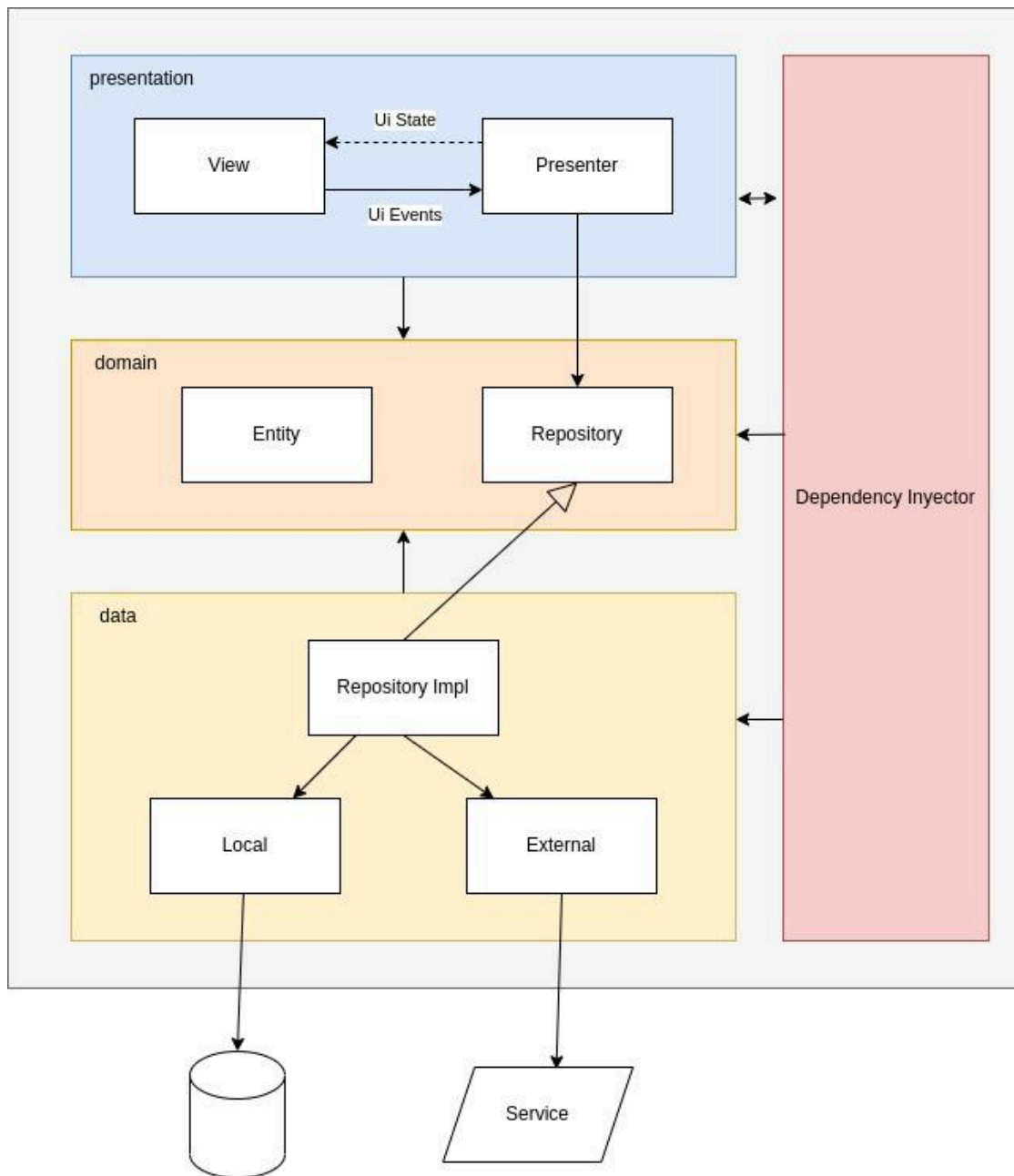
En esta etapa no hace falta crear subclases ni nuevas estructuras de paquetes.

Clean MVP

Modificar la estructura del paquete `moredetails` para que corresponda a una arquitectura

Clean MVP, siempre respetando los principios SOLID y clean code.

Realizar el diagrama de clases resultante (sin atributos ni métodos).



Unit Testing

- Implementar los tests unitarios del paquete *moredetails* siguiendo la misma estrategia que *home* (*Presenter*, *Repository* y *helpers de presentation*).

Abstracción del Servicio Externo + Integración

Parte 1:

- Mover el paquete de lógica de comunicación con el servicio externo al módulo `libs`.

Parte 2:

Implementar los siguiente refactors:

Generalizar el More Details: Reemplazar la clase de dominio `Article` por una clase `"Card"` con las siguientes propiedades:

- `Description`
- `Info Url`
- `Source` (por ahora es única)
- `Source Logo Url`

Agregar en la UI un label que indique la fuente de la información (LastFM, Wikipedia, New York Times).

Parte 3:

Integrar al proyecto los servicios New York Times y Wikipedia. La interfaz de los mismos se encuentra en el módulo `libs`

El objetivo es que el usuario, al seleccionar "more details", vea todos los resultados y sus fuentes.

Implementar el patrón broker en la integración de los 3 servicios externos.

Modificar la UI para visualizar los 3 cards.