Desarrollo de Aplicaciones para Ciencia de Datos

2º Grado en Ciencia e Ingeniería de Datos

Universidad de las Palmas de Gran Canaria, EII

PROYECTO SPOTIFY A SQLITE

Ricardo Juan Cárdenes Pérez

RICARDO JUAN CÁRDENES PÉREZ

9/11/2022

ÍNDICE

RESUMEN

Este proyecto está orientado a la toma de datos de distintos artistas de Spotify mediante una Web API RESTFul privada que nos ofrece la empresa a los desarrolladores. Estos datos que obtenemos los insertaremos en una base de datos local, montada con tecnología SQLite.

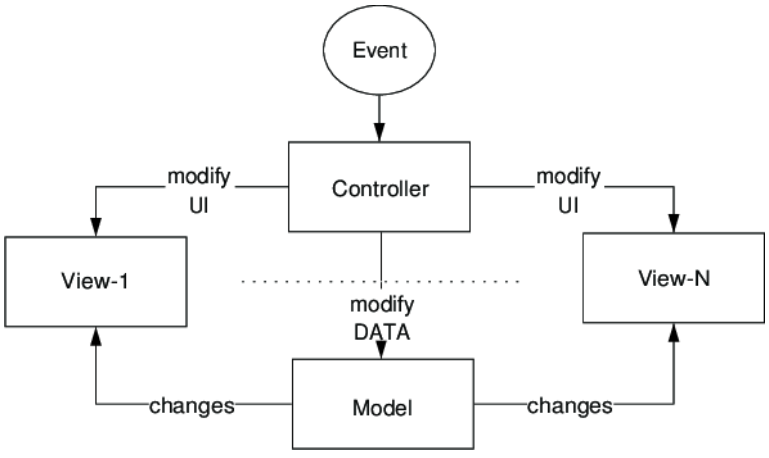
RECURSOS UTILIZADOS

Este proyecto ha sido desarrollado en IntelliJ con la herramienta de carga y gestión de artefactos Maven, lo que nos ha permitido trabajar con los json obtenidos por la Web API de Spotify y obtener sus datos de una forma cómoda y eficaz mediante la clase Gson. Además nos ha permitido instalar el driver de la base de datos SQLite para poder crear la conexión a la misma e insertar y consultar los datos recogidos de los artistas seleccionados. Como herramienta de control de versiones se ha utilizado Git, el cual se conecta con GitHub para poder almacenar el proyecto en un repositorio en la nube.

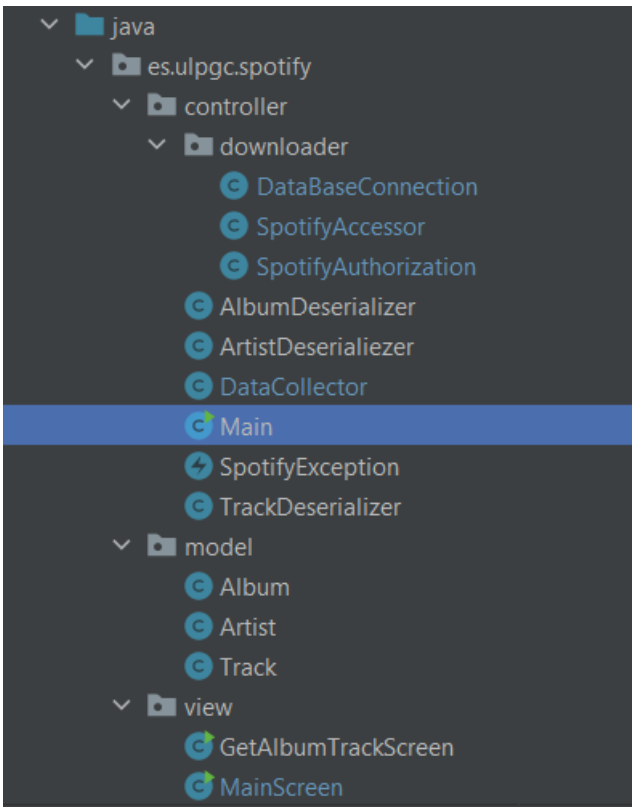
DISEÑO

El estilo arquitectónico adoptado para la implementación es el MVC. Este estilo pretende dividir la lógica de nuestro programa en tres capas o paquetes interconectados:

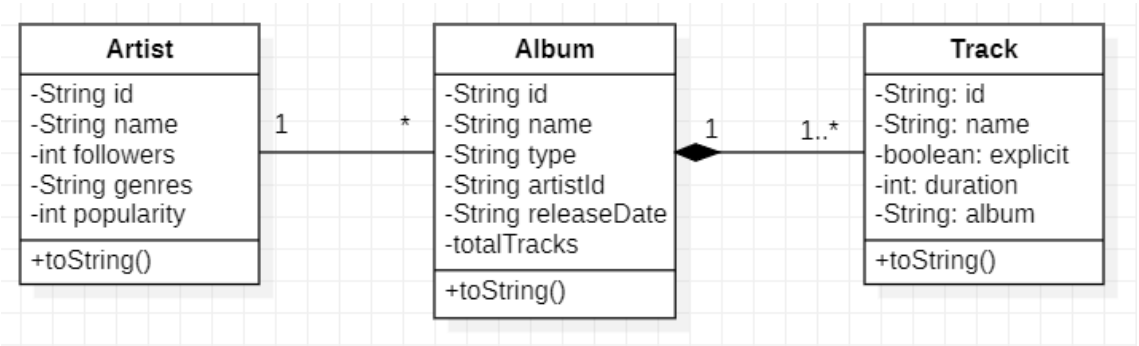
- Model: este es el componente central de nuestro arquitectura. Contiene la representación abstracta de la información con el que el sistema opera.
- Controller: dedicado a responder a eventos, ya sean internos o provocados por el usuario, e invoca peticiones al modelo cuando se solicita alguna información, como la instanciación de un álbum o artista para poder insertarlo posteriormente en la base de datos.
- View: Presenta el modelo y la lógica de negocio de nuestra aplicación en un formato fácil para que el usuario pueda interactuar con el mismo.



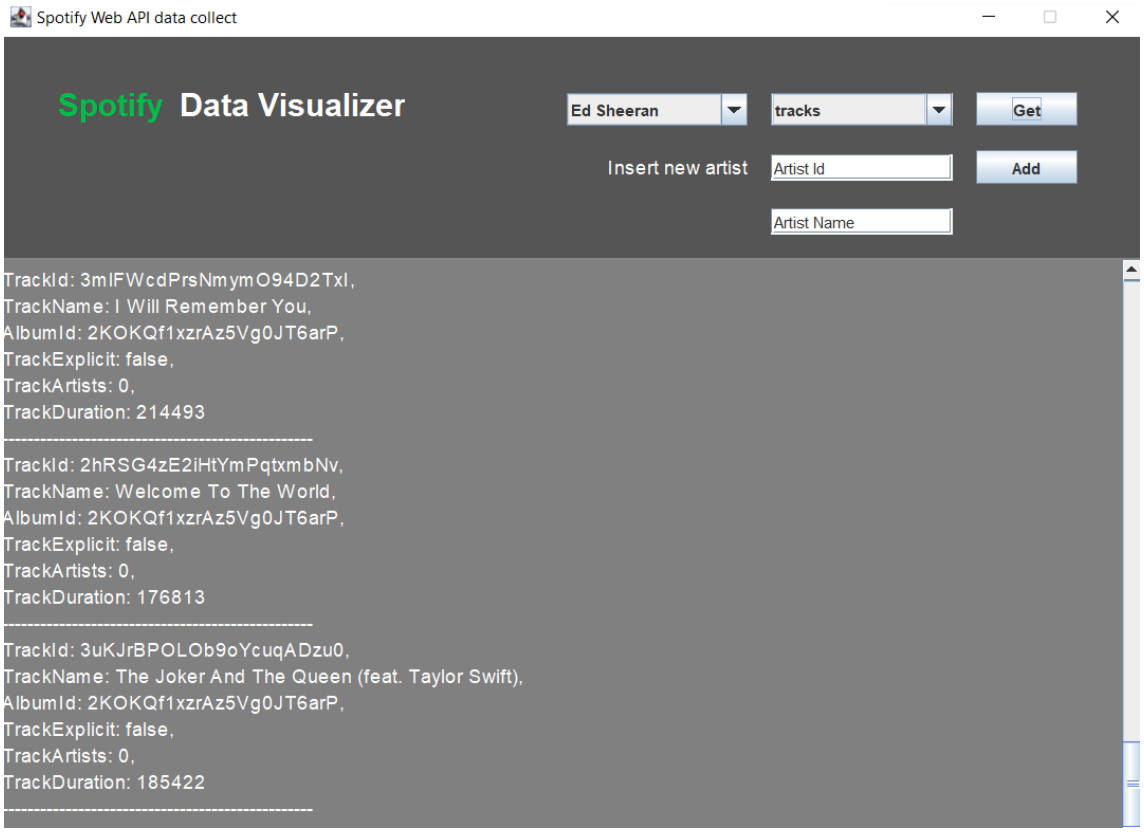
Si abrimos el proyecto en intelliJ, podremos apreciar esta arquitectura de la siguiente manera

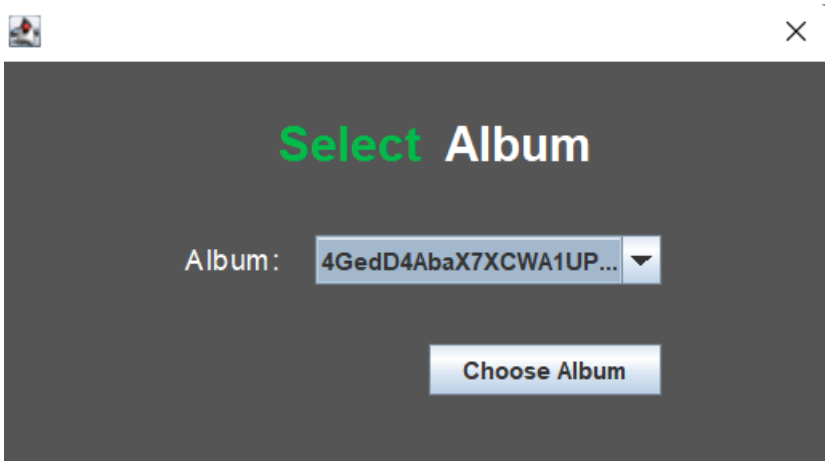


Donde el modelo podemos representarlo mediante el siguiente diagrama de clases



En el paquete View se encuentran las clases relacionadas con la interfaz gráfica de usuario. Existen dos ventanas para esta: MainScreen (la ventana principal) y GetAlbumTrackScreen. Las cuales visualizamos de la siguiente forma:





Principios de Diseño

En cuanto a los principio de diseño, quizás el que más destaca en nuestro proyecto es el de responsabilidad única. De esta manera tenemos un código divido en clases, cada una con una responsabilidad en concreto. Por ejemplo, la clase AlbumDeserializer está enfocada única y exclusivamente en la deserialización del JSON obtenido mediante la Web API. El resto de principios SOLID pueden apreciarse a lo largo del código fuente del proyecto, aunque de una forma menos destacable.

Conclusiones

Considero que este es un proyecto realmente útil, en el que se adquieren conocimientos y destrezas realmente útiles. He aprendido a obtener datos reales de una Web API, algo muy importante en la actualidad, y a guardar todos los datos en una base de datos.

Líneas Futuras

Para las próximas versiones, trataré de mejorar la interfaz gráfica y dejar el código más limpio, aparte de documentar el código con Java Doc. Trataré de utilizar Git desde el inicio del proyecto para tener un buen control de versiones en caso de querer volver a versiones anteriores. Trataré además de modificar las tablas, añadiendo clases ajenas a las tablas para obtener un modelo relacional de los datos.

Bibliografía

- https://docs.oracle.com/javase/7/docs/api/
- https://developer.spotify.com/documentation/web-api/
- https://www.sqlitetutorial.net/sqlite-java/
- https://mvnrepository.com/repos/central