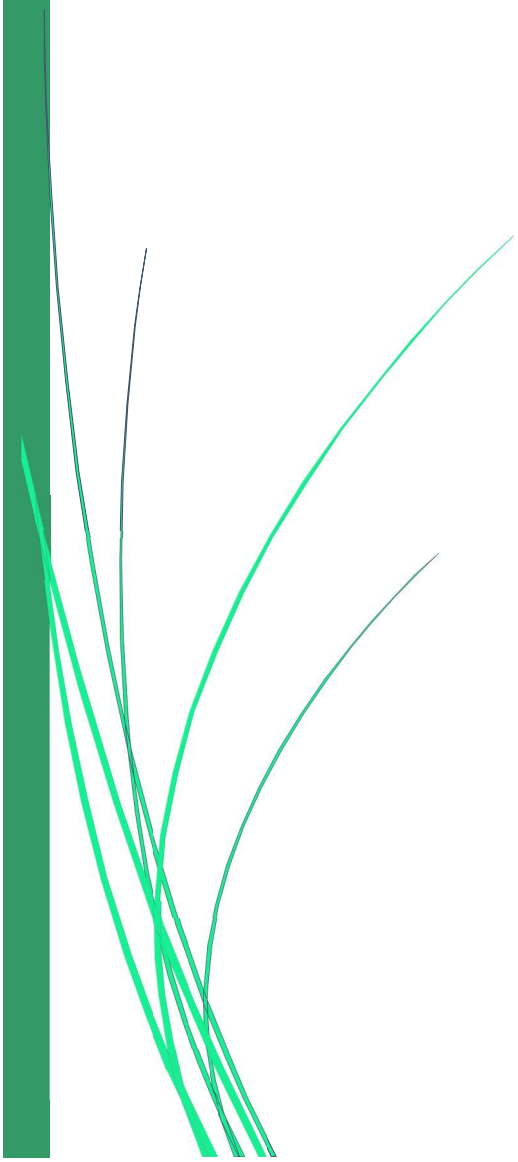
Escuela de Ingeniería Informática

Proyecto Spotify a SQLite

Desarrollo de Aplicaciones para la Ciencia de Datos



Susana Suárez Mendoza
GRADO EN CIENCIA E INGENIERÍA DE DATOS
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Susana Suárez Mendoza

09-11-2022

ÍNDICE

RESUMEN

Este proyecto consiste en extraer de la API pública de Spotify la información sobre los artistas, álbumes y canciones e introducirla en una base de datos compuesta por tres tablas: artistas, álbumes y canciones.

El mismo tiene distintas partes:

1. Package “Model”: este paquete contiene las clases POJO “Artist”, “Album” y “Track” y la clase “MapManager” que se encarga de la gestión del mapa que contiene el nombre y el id de cada uno de los artistas.
2. Package “View”: este paquete contiene el “Main” ejecutable donde se muestra la información de los artistas.
3. Package “Controller”: este paquete está compuesto por la clase “Controller” que ejecuta la clase “GetAPI” donde se solicita la información correspondiente a los artistas, álbumes y canciones. Además, contiene los siguientes paquetes:
    a. Package “dataBase” que contiene las clases que crean, insertan, seleccionan y gestionan la base de datos.
    b. Package “accessor” que contiene las clases SpotifyAccessor y SpotifyAuthorization que permiten acceder y autorizar el registro para trabajar con la API de Spotify.

RECURSOS UTILIZADOS

- **Entorno de desarrollo y recursos.**

Este proyecto se ha desarrollado en el entorno IntelliJ IDEA Edu utilizando Maven.

Los artefactos importados de Maven Central en el pom.xml son las dependencias de "gson" para poder trabajar con los datos de la API de Spotify los cuales vienen en formato json y "sqlite-jdbc" para insertar esos datos en la base de datos SQL.

- **Herramienta de control de versiones distribuida.**

Se ha utilizado Git para registrar los cambios de la aplicación en torno a un futuro. Además, se ha subido a un repositorio de github:

susanasrez/Spotify (github.com)

DISEÑO

- **Estilo arquitectónico.**

Para desarrollar este proyecto se ha utilizado el estilo Model View Controller (MVC) que trata de separar la lógica de negocio de su presentación e interacción con el usuario.

- Model: representa la estructura lógica de los datos en la aplicación de Spotify.
- Controller: el controlador es el encargado de gestionar las solicitudes a la API y contacta con el Model para cualquier dato que pueda necesitar el View. Además, es el encargado de gestionar la base de datos.
- View: en este caso, la vista presenta la información al usuario a través de la consola.

- **Principios de diseño utilizados.**

- Principio de responsabilidad única (Single Responsibility): cada clase solo tiene una responsabilidad.
- Abierto para extensión, cerrado para modificación (Open for Extension, Closed for Modification): al separar el modelo del controlador se permite la extensión del código en el futuro.
- Sustitución de Liskov (Liskov Substitution). En mi gestor de base de datos hay una herencia cuyo padre es "DBManager" y cada subclase es capaz de cumplir cada característica de su clase padre y podría ser tratada como su clase padre.
- Inversión de dependencia (Dependency Inversión). Desacoplamiento de módulos de software. En lugar de Módulos de alto nivel Dependiendo de los módulos de bajo nivel, ambos dependen de abstracciones.

CONCLUSIONES

Considero que he aprendido a trabajar con parte de la API pública de Spotify, así como a trabajar con bases de datos en el idioma SQL con su interfaz JDBC. Además, la parte que más me ha costado ha sido la deserialización de un JsonObject así como el INSERT INTO de SQL.

Si empezara de nuevo este proyecto, empezaría a modularizar desde un principio el código ya que una vez desarrollado en una sola clase, me ha costado dividirlo.

LÍNEAS FUTURAS

En líneas futuras, me gustaría desarrollar una interfaz gráfica para poder visualizar de manera más limpia los datos, además de temporizar los get de la API para no sobrecararla.

BIBLIOGRAFÍA

Web API Reference | Spotify for Developers

Maven Central Repository Search

SQLite Tutorial - An Easy Way to Master SQLite Fast