

Proyecto Spotify a SQLite

*DESARROLLO DE APLICACIONES
PARA CIENCIA DE DATOS*

CIENCIA E INGENIERÍA DE DATOS

*UNIVERSIDAD DE LAS PALMAS DE
GRAN CANARIA*

LAURA LASSO GARCÍA
09/11/2022
VERSIONES: 1.0-SANPSHOT

Resumen

El proyecto se divide en varios módulos junto con 2 interfaces; MusicDatabase y MusicSource. Los módulos en sí se corresponden con las funciones que se realizan a la hora de coger los datos necesarios de la base de datos de Spotify mediante una API, guardarlos en clases POJO e insertarlos en las tablas que se crean en la base de datos.

Para descargar los datos en Json nos bastará únicamente con el id del artista correspondiente que lo pasaremos como parámetros a las funciones tracksOf(), artistsOf() y albumsOf(). Esto nos permitirá conocer todos los datos necesarios para poder inicializar los objetos correspondientes a las clases Artist, Album y Track y añadirlos a los arrays correspondientes para luego añadir cada objeto de esa lista, uno por uno, a la tabla de la base de datos mediante las funciones que se encuentran definidas en la interfaz MusicDatabase.

Éstas a su vez harán uso de las funciones definidas en el módulo DMLTranslator, en donde se formarán las sentencias SQL junto con los valores de los objetos que se han pasado como parámetros para que luego las funciones add() puedan ejecutarlas.

Primero se crearán las 3 tablas correspondientes y luego se ejecutarán las sentencias SQL en el módulo SqliteMusicDatabase para así añadir cada objeto que se pase como parámetro a las funciones add() a la base de datos, obteniendo en la base de datos las 3 tablas junto con la información de cada artista.

Índice

Recursos utilizados.....	5
Diseño.....	5
Conclusiones.....	6
Líneas futuras.....	6
Bibliografía.....	6

Recursos utilizados

El entorno de desarrollo es IntelliJ e IntelliJ Ultimate y las herramientas de documentación es Word.

Diseño

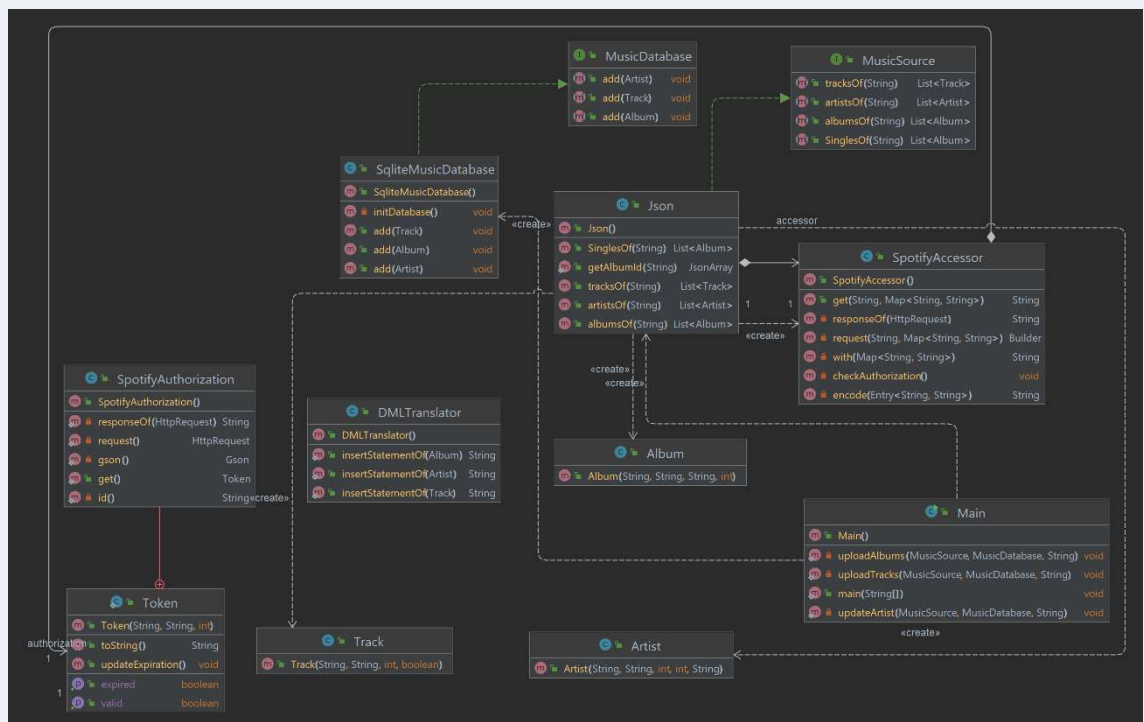
Los principios implementados son los SOLID para así mantener un código limpio, ordenado, fácil de mantener de manera que podamos facilitar la nueva entrada de funcionalidades a nuestra aplicación y corrección de errores. Tal y como podemos ver, cada módulo implementado se encarga de una sola responsabilidad;

→ DMLTranslator: Se va a encargar de construir las sentencias SQL que se van a encargar de insertar los objetos con sus respectivos atributos para cada una de las tablas de la base de datos.

→ Json: Se va a encargar de coger el Json del path que se indica gracias a la clase implementada SpotifyAccessor. Luego cogerá cada objeto del Json que se necesita y lo convertirá en el tipo del atributo que esta definido en las clases POJO. De esta manera, inicializaremos los objetos que se van a añadir en las tablas de las bases de datos y los añadiremos al array que guardará los objetos creados de una clase. Ese array es lo que devolverán las clases que se encuentran definidas en la interfaz MusicSource. Luego en el main se recorrerán cada uno de esos array y cada objeto se irá añadiendo a las tablas correspondientes gracias a las funciones add() que se encuentran en la interfaz MusicDatabase.

→ SqliteMusicDatabase: Este módulo se encargará de crear las tablas de los artistas, de sus álbumes y sus canciones y de ejecutar las sentencias SQL que devuelven las funciones definidas en el módulo DMLTranslator para así añadir esos objetos creados a las tablas de la base de datos.

En cuanto al diagrama de clases y relaciones de dependencia, se ha realizado con IntelliJ IDEA Ultimate, gracias a una de sus funcionalidades implementadas:



Conclusiones

Este proyecto se puede enfocar de múltiples formas; no usando las clases POJO o implementar menos módulos. Sin embargo, he optado por usar más módulos para que el código sea mucho más fácil de comprender y que de esta forma esté más ordenado y sencillo.

En el caso de no haber usado las clases POJO, lo que se habría hecho es, en vez inicializar nuevos objetos con los valores que se cogen del Json, se pasarían como parámetros a una función que se encargaría de insertar esos datos a las tablas de la base de datos mediante sentencias SQL. Aquí también podemos elegir en usar un módulo que se encargue de traducir esas sentencias SQL y luego pasarlas con los valores a insertar para que se pueda ejecutar en la función, o directamente haber usado un tipo de sentencia SQL llamadas sentencias SQL preparadas pasando los valores que se quieran insertar.

Líneas futuras

En cuestión de los usos que se pueden dar a este proyecto desde el punto de vista empresarial, es poder desarrollarlo de manera que, desde una aplicación podamos buscar a un artista y que nos dé toda su información relativa a él, a sus álbumes y sus canciones.

Esto puede ser muy útil en la música, en cuanto a desarrollar publicidad y enfocar el producto a los consumidores. El tener una aplicación donde recopile toda la información relativa del artista y de los demás que se indiquen puede ayudar a poder enfocar dicho producto de una manera correcta y darlo a conocer de una manera determinada.

Bibliografía

[Principios del diseño de clases - Adictos a la Informática \(adictosalainformatica.com\)](http://adictosalainformatica.com)
[Doce principios de diseño que todo desarrollador debería conocer \(genbeta.com\)](http://genbeta.com)