



---

# PROYECTO SPOTIFY A SQLITE

---

Desarrollo de Aplicaciones para Ciencia de Datos (DACD)



2º GRADO DE CIENCIA E INGENIERÍA DE DATOS (GCID)

ESCUELA DE INGENIERÍA INFORMÁTICA (EII)

Universidad de las Palmas de Gran Canaria (ULPGC)

## **Autor**

José Gabriel Reyes Rodríguez

## **Fechas de desarrollo del proyecto**

desde el 02-11-2022 hasta el 07-11-2022

## **Versiones**

1

## **Revisiones de la memoria**

3 revisiones

## Índice:

Resumen:.....	3
Recursos utilizados.....	4
Entornos de desarrollo.....	4
Herramientas de control de versiones.....	4
Herramientas de documentación .....	4
Diseño.....	4
Conclusiones .....	5
<b>Líneas futuras</b> .....	5
Bibliografía .....	5

## Resumen:

En este proyecto el objetivo consistía en descargar desde Spotify la información de un artista, sus álbumes y sus canciones para insertarlas en una base de datos en SQLite. Para ello, he utilizado clases POJO para guardar la información que iba a insertar en la base de datos pasando el resultado JSON de la petición a Spotify a un objeto de la clase deseada empleando GSON. La forma en la que está implementado es que por cada id de artista que haya en la lista, se hace una petición para acceder al artista, otro para acceder a los álbumes del artista y otro para acceder a las canciones del álbum, se convierte cada resultado JSON de la petición a objeto de cada tipo y le paso cada atributo de la clase al método para insertar datos en las tablas para cada objetos, en donde para evitar duplicados uso en el "SQL statement" un "insert" que, si encuentra un conflicto porque la id es la misma, haga un "update" de todos los datos del objeto. Además, está implementado un escáner en la consola para que en la ejecución pueda elegir ejecutar el caso base con 5 artistas, añadir más artistas al caso base (donde se acepta una lista de ids separadas por comas) o si quiere eliminar las tablas con un "drop table". También se intenta controlar el exceso de peticiones con Spotify en donde si se llega a un múltiplo de 100 peticiones, el programa para unos segundos, como el número de peticiones máximo que se puede hacer a la API de Spotify varía según la hora o el día, puede que todavía siga saltando un error 429.

Dentro del trabajo hay 13 clases que podemos destacar:

Las clases Albums, Artist, Followers y Tracks, que nos permiten pasar a objetos los resultados de las peticiones a Spotify para conseguir de ellos los datos que se requieran. Los nombres de los atributos de estas clases deben tener exactamente el mismo nombre que tengan en Spotify para poder obtenerlos correctamente.

Las clases SQLAlbumManagement, SQLArtistManagement, SQLTrackManagement y SQLiteDataBase son las clases que se encargan de la conexión con la base de datos, y el manejo de las tablas dentro de ella.

La clase RefactoredInsert sirve principalmente como soporte de la clase Controller para evitar que esta se quedase muy grande, en el Controller están los pasos a seguir del programa y en el Main hay un escáner por línea de comandos que dirige lo que tiene que hacerse según lo que quiera el usuario.

Para mantener la seguridad de la cuenta de Spotify he quitado el ID y el cliente secreto del código por si acaso. Estos estarán adjuntos junto con el enlace al repositorio.

## Recursos utilizados

### Entornos de desarrollo

IntelliJ IDEA versión educativa edición 222.4167.41

### Herramientas de control de versiones

Git

### Herramientas de documentación

Microsoft Word.

## Diseño

En el diseño del trabajo se ha intentado seguir el modelo MVC en donde la clase main solo tiene el escáner implementado ya que considero que esto lo que hace es dirigir que se va a ejecutar según las elecciones del usuario, además, como también muestra información al usuario, el escáner no encajaría del todo en el controller. En el controller, por el contrario, tenemos las llamadas a los métodos para crear las tablas, insertar los datos, eliminar las tablas en caso de querer reiniciarlas además de tener la lista con los ids de los cantantes. Además, hay un paquete con todas las clases relacionadas con el manejo de la base de datos.

Se puede ver una imagen mejor en el repositorio:



Spotify developers web:

[Web API Reference | Spotify for Developers](#)

Sqlite tutorial:

[SQLite Java: Inserting Data \(sqlitetutorial.net\)](#)

Manual para el git en Inteliij:

<https://www.jetbrains.com/help/idea/using-git-integration.html>

Idea básica para implementar el escáner:

<https://iteritory.com/read-interactive-command-line-input-with-java-scanner-class/>