

# Spotify a SQLite



Desarrollo de Aplicaciones para Ciencia de Datos.  
Segundo año del Grado en Ciencia e Ingeniería de Datos.

Proyecto realizado por Gerardo León Quintana.

Fecha de realización de la memoria 7 de noviembre de 2022.

## Resumen

El proyecto en sí consiste en la creación de una base de datos a través de la información obtenida de una serie de cantantes solistas registrados en Spotify, dicha información es recogida a través de una serie de peticiones a la API REST de Spotify, las cuales se pueden realizar utilizando un método GET. Habiendo recogido la información de dichos artistas se deberán crear tres tablas con dicha información, una para la información única del artista, otra para todos los álbumes publicados por el artista en Spotify, y por último una tabla en la que se muestre la información recopilada de las canciones del artista.

Para realizar lo mencionado previamente, se ha desarrollado un código en el lenguaje de programación JAVA, el cual nos permite realizar las acciones mencionadas de la siguiente manera.

Primero, se ha realizado la recogida de información a través del método GET de la clase SpotifyAccesor, dichas acciones se han instanciado en métodos llamados getArtist, getAlbums y getTracks, las cuales se encuentran en la clase GetInfo. Toda la información recopilada se guarda en una lista de elementos de su clase correspondiente, ya bien sea Artist, Albums o Tracks.

Tras obtener la información deseada debemos establecer una conexión con la base de datos, para así crear las tablas. Para ello se ha creado la clase DatabaseController, en la cual se encuentran los métodos necesarios para lo mencionado anteriormente, para establecer la conexión disponemos del método connect, para generar las tablas correspondientes poseemos los métodos create\_artist, create\_albums y create\_tracks, y para introducir la información recopilada en las tablas generadas disponemos de los métodos insert\_artist, insert\_albums e insert\_tracks.

Finalmente, para que la ejecución del programa sea lo más organizada posible se ha creado una clase Controller, la cual es la encargada de ejecutar los métodos predispuestos previamente. Además, Controller al ejecutarse dispone de una interfaz con la que poder interactuar con el usuario, para bien añadir el ID de un artista o bien para ejecutar el programa en sí con los usuarios añadidos a la lista.

## **Índice:**

5. Recursos utilizados, diseño y líneas futuras.

6. Conclusiones y bibliografía.

## Recursos utilizados

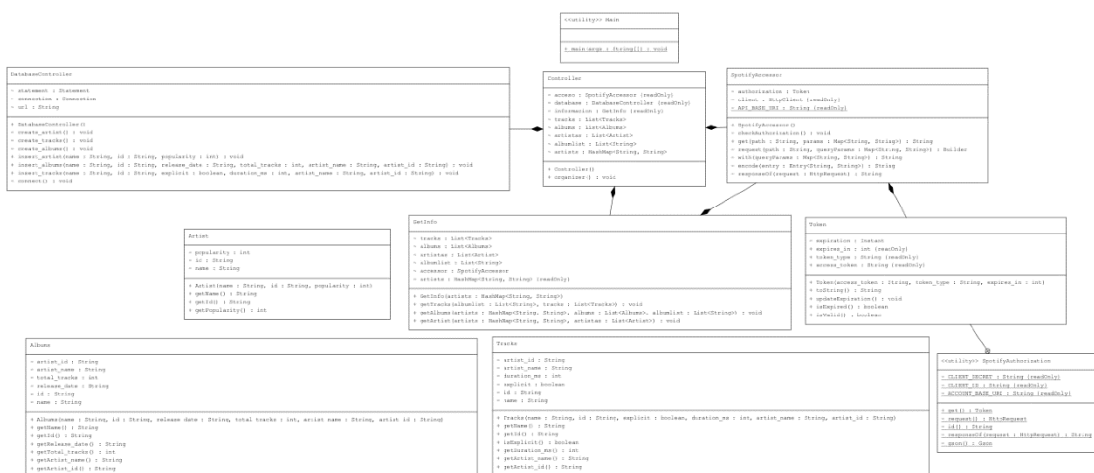
Este proyecto se ha realizado completamente en IntelliJ.

## Diseño

En cuanto al diseño del proyecto, se ha tratado ser lo más fiel al estilo arquitectónico MVC.

## Diagrama de clases

Lamento mucho la calidad de la imagen, pero UML siempre me genera la imagen con esta calidad.



## Líneas futuras

Para poder comercializar el proyecto, debería ser capaz de aceptar más de una ID, tener un tiempo de ejecución mejor, y por supuesto necesita tener una interfaz más interactiva con el usuario.

## **Conclusiones**

La realización del proyecto ha sido una experiencia que ha requerido de esfuerzo, investigación y sobre todo tiempo. Si tuviera que comenzar el proyecto desde el inicio claramente habría tomado un camino menos elaborado y más directo, dado que para llegar a la versión actual del proyecto tuve que realizar varios test para comprender el funcionamiento de la API, a la par a la hora de desarrollar los métodos comenzaba por crear el contenido de estos en el Main, para así testear y comprobar su funcionalidad y posteriormente trasladarlos a sus clases correspondientes, esta metodología era efectiva pero consumía demasiado tiempo.

## **Bibliografía**

- <https://stackoverflow.com/>
- <https://www.geeksforgeeks.org/>
- <https://iteritory.com/>
- <https://www.sqlitetutorial.net/>