



# SPOTIFY API

- **CURSO:** Segundo.
- **TITULACIÓN:** Grado en Ciencia e Ingeniería de Datos.
- **ESCUELA:** Escuela de Ingeniería Informática.
- **UNIVERSIDAD:** Universidad de Las Palmas de Gran Canaria.

Desarrollo de  
Aplicaciones  
para Ciencia  
de Datos.

# RESUMEN

En este proyecto hemos realizado un programa de Java mediante el cual se puede obtener información sobre artistas, álbumes y canciones de la API de Spotify. Las API permiten que una aplicación extraiga archivos o datos preexistentes dentro de un software y los use en otro programa o en uno de sus otros niveles. Luego esta información la almacenamos en tablas en una base de datos SQLite. SQL es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

Para esto he utilizado 7 clases. Dos de ellas son del código base que hemos recibido en el campus que sirven para conectarse a Spotify y autorizar tus pedidos mediante un Client ID y un Client Secret. En primer lugar, tengo las clases Album, Artist y Track en las que se encuentran los "get" para los álbumes, artistas y canciones respectivamente. Así obtenemos la información que se encuentra en la API de Spotify y la almacenamos en mapas. Luego tengo la clase DataBaseManager que se encarga de crear las tres tablas SQL para álbumes, artistas y canciones. Además, inserta la información obtenida en la API a las tablas. Finalmente, en la clase Main, tengo un mapa con los artistas que vamos a utilizar y sus respectivos IDs y se encarga de realizar las llamadas a las distintas clases para que todo funcione.



# ÍNDICE

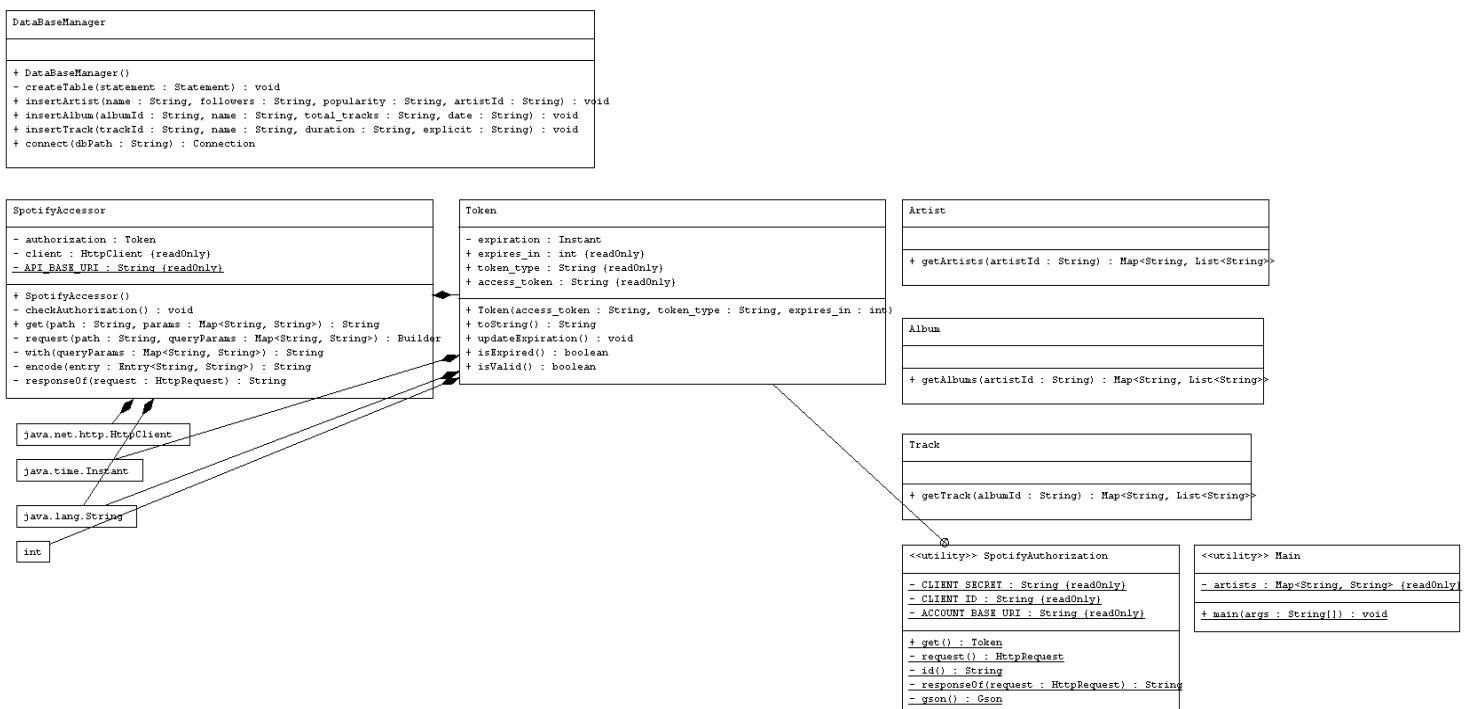
- Recursos utilizados
- Diseño.
- Conclusiones.
- Líneas futuras.
- Bibliografía

# Recursos Utilizados

Para este proyecto he utilizado “IntelliJ IDEA Community Edition 2022.2.2” como entorno de desarrollo. Para realizar esta memoria, he utilizado el procesador de texto “Microsoft Word”. Para controlar las versiones usé “Git”, que es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. También el “GitHub” donde he creado un repositorio donde se encuentra este proyecto.

# DISEÑO

Respecto al diseño, he seguido el principio SOLID de responsabilidad única, es decir, las clases tienen una sola responsabilidad. Además del principio de inversión de dependencias, que se basa en que los módulos de alto nivel no dependan de módulos de bajo nivel.



# **Conclusiones**

En definitiva, este trabajo me ha parecido muy interesante ya que trata de una aplicación que muchos de nosotros usamos a diario. Sin embargo, se me ha dificultado la realización de este ya que mis conocimientos sobre Java no son muy avanzados. Por otro lado, el enfoque que le he dado ha sido un tanto distinto ya que lo he realizado mediante el uso de mapas, lo que en un primer momento se me ocurrió y finalmente salió bien, pero quizás hubiera sido más fácil mediante el uso de clases POJO y la creación de objetos. Además de esto, tendré en cuenta para el futuro el uso de diagramas de clase para una mejor estructuración de las ideas.

# **Líneas futuras**

En un futuro, se podría convertir en una suscripción de pago extra en Spotify mediante la cuál puedas acceder a la información de todos los artistas y usando un filtro de los géneros musicales que selecciones se realicen “playlists” automáticamente.

# **Bibliografía**

<https://www.sqlitetutorial.net>

<https://stackoverflow.com>

<https://developer.spotify.com>

<https://learn.microsoft.com>

Alejandro Bolaños García  
9 de noviembre de 2022