

Spotify a SQLite



Desarrollo de Aplicaciones para Ciencia de Datos

Grado en Ciencia e Ingeniería de datos

Segundo curso

Escuela de Ingeniería Informática

Autora: Cynthia Quintana Reyes

Fecha: 09/11/2022

Última versión del proyecto: 09/11/2022

Última versión de la memoria: 09/11/2022

Resumen

Las API son herramientas(librerías) que los desarrolladores de un producto ofrecen a otras compañías para que puedan utilizar sus servicios. Estas compañías no tienen acceso al código fuente, sino que se les pasa documentación de cómo utilizar estas librerías para acceder a sus productos.

En el caso de la API web de Spotify sirve para descubrir música y podcasts, administrar la biblioteca de Spotify, controlar la reproducción de audio y mucho más.

Para desarrollar este proyecto es necesario obtener datos de esta API, y para ello es imprescindible registrarnos en Spotify for Developers, y crearnos un proyecto para obtener el Client ID y Client Secret.

Los datos que se han obtenido en este proyecto han sido los álbumes y las canciones de cinco artistas solistas, y han sido almacenados en una base de datos SQLite en la que existen tres tablas (artists, albums, tracks).

En la tabla *artist* se encuentran cinco campos con las características del artista: name (nombre del artista), id (identificador del artista),uri (uri del artista), popularity (popularidad del artista) y type(género musical del artista).

En la tabla *albums* se encuentran cuatro campos con las características del álbum: name (nombre del álbum), album_type (género musical del álbum), total_tracks (número de canciones totales del álbum) y release_date (fecha de publicación del álbum).

En la tabla *track* se encuentran cuatro campos con las características de la canción: id(id de la pista), name (nombre de la pista), track_number (el número de la pista. Si un álbum tiene varios discos, el número de pista es el número del disco especificado), disc_number (número de disco donde se encuentra la canción) y duration_ms (duración de la pista en milisegundos).

Índice

Recursos utilizados.	Pág 3.
Diseño.	Pág 4.
Diseño.	Pág 5.
Conclusión.	Pág 6.
Líneas futuras	Pág 6.
Bibliografía	Pág 7.

Recursos utilizados

El lenguaje de programación que hemos utilizado para desarrollar este proyecto ha sido Java, que es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificar todo, desde dispositivos móviles, aplicaciones y software empresarial hasta aplicaciones de big data y tecnologías del servidor.

La versión de Java utilizada en este proyecto es JDK 11, que es la primera versión de Java con un JDK denominado LTS o Long Term Support. Esto significa que Oracle garantiza que te dará soporte y actualizaciones para la versión durante 3 años.

Para desarrollar el código, hemos utilizado IntelliJ, que es un entorno de desarrollo integrado para el desarrollo de software. Está desarrollado por JetBrains y está disponible en dos ediciones: edición para la comunidad y edición comercial.

Una librería Java se puede entender como un conjunto de clases que facilitan operaciones y tareas ofreciendo al programador funcionalidad ya implementada y lista para ser usada través de una API.

Las librerías que he usado en este proyecto han sido la librería Gson, que es una librería para Java que permite convertir nuestros objetos en representación Json y viceversa, y la librería JDBC, que también es una librería para Java, que permite establecer una conexión con una base de datos.

Un plugin es un programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal).

En este caso hemos instalado el plugin SQL browser para visualizar la base de datos.

Diseño

Los paquetes son el mecanismo que usa Java para facilitar la modularidad del código.

El proyecto está dividido en cuatro paquetes: `es.ulpg.api`, `es.ulpgc.controller`, `es.ulpgc.database` y `es.ulpgc.model`.

En el paquete `es.ulpgc.api` se encuentran las clases `SpotifyAuthorization` y `SpotifyAccessor`, que son las clases que nos permiten acceder a la API de Spotify, además se encuentra la clase `ApiException`, que si la conexión con la API falla, lanza una excepción.

En el paquete `es.ulpgc.model` se encuentran las clases `Album`, `Artist` y `Track`, que son las clases que representan a dichos objetos y tienen sus respectivos constructores.

En el paquete `es.ulpgc.controller` se encuentran las clases `GetAlbum`, `GetArtist` y `GetTrack`, que obtienen los datos que solicitamos desde la API y los devuelve deserializados.

En el paquete `es.ulpgc.database` se encuentra la clase `SQLiteMusicDatabase`, que crea la conexión con la base de datos, crea las tablas con sus respectivos campos y contiene los métodos para llamar a la clase donde se encuentran los registros. También se encuentra la clase `DMLtranslate`, que contiene los métodos para insertar los registros a dichas tablas.

Este proyecto da como resultado una base de datos con tres tablas y con sus respectivos registros, como observamos en las figuras 1,2 y 3.

Los principios de diseño de este proyecto son:

- 1.Principio de única responsabilidad: Este principio dice que cada módulo (clase) debería tener una sola razón para ser modificado.
2. Principio de abierto cerrado: Este principio dice que el sistema debe estar cerrado a modificaciones, y abierto a extensiones.

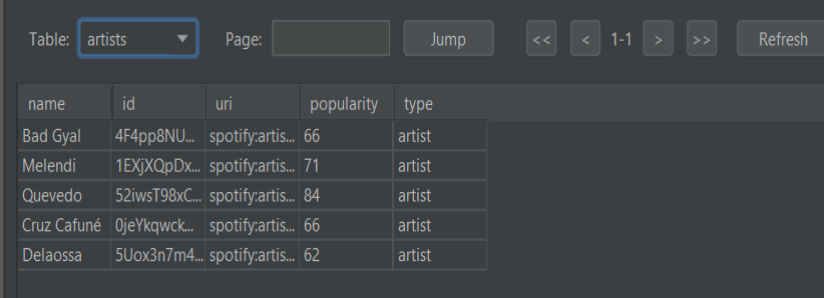


Table:	artists	Page:		Jump	<<	<	1-1	>	>>	Refresh
name	id	uri	popularity	type						
Bad Gyal	4F4pp8NU...	spotify:artis...	66	artist						
Melendi	1EXjXQpDx...	spotify:artis...	71	artist						
Quevedo	52iwsT98xC...	spotify:artis...	84	artist						
Cruz Cafuné	0jeYkqwck...	spotify:artis...	66	artist						
Delaossa	5Uox3n7m4...	spotify:artis...	62	artist						

Figura 1

Table: **albums** Page: Jump << < 1-2 > >> Refresh

name	album_type	release_date	total_tracks
Warm Up ...	album	2021-03-18	8
Worldwide ...	album	2018-02-21	9
Slow Wine ...	album	2016-11-11	7
Sin Carné	single	2022-10-06	1
Sexy	single	2022-07-14	1
Tremendo ...	single	2022-06-16	1
La Prendo	single	2022-05-26	1
Formosa (R...	single	2022-05-06	1
Flow 2000 (...)	single	2022-03-17	1
Sound Syst...	single	2021-12-09	3
Su Payita (G...	single	2021-12-02	1
Nueva York...	single	2021-11-25	1
Flow 2000	single	2021-10-14	1
Esta Noche	single	2021-07-08	1
BOBO	single	2021-06-25	1
Pussy	single	2021-03-04	1
Zorra (Remi...	single	2021-01-21	1
Blin Blin	single	2020-11-13	1

Figura 2

Table: **tracks** Page: Jump << < 1-5 > >> Refresh

name	id	track_num...	duration_ms	disc_number
Pussy	7aZ30GAOJ...	137253	1	1
44	0NipDzam...	185466	1	2
Blin Blin	4OWPAMZ...	144213	1	3
Judas	0nIfdjswV0...	145973	1	4
Iconic	3zNzqaks2...	134506	1	5
Gasto	5KBW0JaG...	128080	1	6
Aprendiend...	19U0hb6nU...	147040	1	7
Zorra - Re...	5tTPP8qbaj...	216400	1	8
Intro - Prod...	68C2QbQU...	36000	1	1
Internationa...	3PLcsZsFC1j...	194390	1	2
Tra - Prod: ...	4Wozolbec...	225628	1	3
Yo Sigo Iual...	25RN6owX...	189242	1	4
Candela - P...	7AkGPjtb6Y...	196753	1	5
Trust - Prod...	6yAMdceZ...	223917	1	6
Blink - Prod...	1XLxTEhFn...	271978	1	7
Tu Moto - P...	3opJjMxkA...	195256	1	8
Realize - Pr...	2lOkmY3t6q...	238147	1	9
D Way You ...	0VfPpXQU...	290000	1	1

Figura 3

Conclusiones

En este proyecto he aprendido a trabajar con las API, obteniendo datos de estas.

Las API son un recurso muy interesante, puesto que se utilizan para integrar nuevas aplicaciones con los sistemas de software existentes. Esto aumenta la velocidad de desarrollo, ya que no hay que escribir cada funcionalidad desde cero. Puede utilizar las API para aprovechar el código existente.

Estos datos se han procesado y se han introducido en una base de datos, que es importante saber trabajar con ellas, ya que una base de datos permite almacenar gran número de información de una forma organizada para su futura consulta, realización de búsquedas, nuevo ingreso de datos, etc. Todo esto lo permite realizar de una forma rápida y simple desde un ordenador.

El trabajo podría mejorar incluyéndole interfaces para que el usuario pueda introducir los identificadores de cada artista

Líneas futuras

Este proyecto debería evolucionar para poder ser comercializable, necesitaría una interfaz gráfica para que los clientes puedan usar esta aplicación, y trabajar en el para que incluya un número ilimitado de artistas.

Bibliografía

<https://www.applesfera.com/aplicaciones-ios-1/spotify-pone-a-disposicion-de-los-desarrolladores-las-herramientas-para-incluir-su-reproductor-en-aplicaciones-ios>

<https://www.jetbrains.com/es-es/idea/features/#:~:text=IntelliJ%20IDEA%20incluye%20una%20integraci%C3%B3n,hacer%20implementaciones%20y%20otras%20actividades>.

<https://www.campusmvp.es/recursos/post/java-11-ya-esta-aqui-te-toca-pagar-a-oracle-o-cambiarte-a-otras-opciones.aspx#:~:text=Bien%2C%20pues%20Java%2011%20es,de%20tan%20solo%206%20meses>.

<https://aws.amazon.com/es/what-is/api/#:~:text=Las%20API%20se%20utilizan%20para,para%20aprovechar%20el%20c%C3%B3digo%20existente>.