**PRIMEROS PASOS**

Una vez asignado el trabajo lo primero fue aprender las tecnologías que iban a ser empleadas para interactuar con el API de Spotify.

Por un lado, consistiría en aprender el lenguaje de programación Python que es el utilizado para hacer las llamadas a la API. Al haber ya desarrollado ya con lenguajes de todo tipo en la carrera se me hizo bastante fácil aprender la sintaxis de Python.

Por otro lado, debía aprender cómo funcionaba la base de datos Node4j. La base de datos Node4j, se trata básicamente de un grafo formado por nodos y aristas que representan la información y relaciones guardadas en la misma.

Comencé por aprender Python poco a poco mediante problemas y ejercicios simples como realizar funciones, bucles y algunas estructuras de control de flujo para irme familiarizándome con el lenguaje. En esta parte me resultó un poco complejo la manera de llamar a mis programas con los argumentos por consola, pero una vez solventados estos problemas se volvió mucho más rápido y sencillo el desarrollo.

Una vez ya con el lenguaje interiorizado me dispuse a realizar mis primeras pruebas con el API de Spotify. Me resultó fácil el registro para sacar los identificadores propios de mi cuenta de desarrollador, pero un poco más complejo el tema de realizar peticiones al API una vez que me autenticase ya que no estaba muy bien indicado el tema de las variables de entorno en Windows. El poder realizar muchos de estos ajustes en PyCharm ha hecho que sea un IDE con el que me he sentido muy cómodo a lo largo del desarrollo.

Posteriormente comencé a realizar diferentes peticiones basándome en ejemplos de la API y propios míos para entender y comprender de primera mano cómo funciona dicha API. Estos ejemplos fueron tales como los álbumes publicados por un artista, la búsqueda de canciones dentro de un álbum, la búsqueda de los propios artistas mediante la URI o simplemente su nombre, la búsqueda de artistas relacionados con uno dado y la búsqueda de las propiedades de una canción. Éste último ejemplo nos acerca bastante a lo que buscamos para poder implementar nuestra aplicación sobre Spotify.

Ya entrando un poco más en materia, implementé un par de ejercicios más complejos. El primero consistía en buscar con una profundidad 3 los artistas relacionados con uno dado, para saber de esta manera un conjunto de artistas relacionados entre sí. El segundo consistía en sacar todas las canciones de todos los álbumes de un artista determinado y de cada una de estas canciones sus principales propiedades como la bailabilidad, ritmo, positivismo...

Una vez llegados a este punto, comencé a familiarizarme con la base de datos Neo4j y su lenguaje para realizar consultas y demás operaciones con la base de datos, Cypher. El aprendizaje me resultó muy sencillo ya que es similar a SQL y ya estaba acostumbrado a trabajar con bases de datos relacionales. Realicé algunos ejemplos de grafos sencillos con libros y películas relacionándolos entre sí.

Una vez aprendida las tecnologías a emplear, me pareció buena idea conectarlas en forma de ejercicio realizado anteriormente sobre las propiedades de las canciones de un artista dado. De forma que toda esta información se muestre en forma de grafo en la base de datos Neo4j.

En este paso del aprendizaje resultó problemático el tema de las relaciones entre nodos para ciertos ejemplos ya que estas relaciones son más apropiadas para asociar distintos tipos de entidades como pueden ser álbumes y canciones o canciones y artistas.

Una vez que hemos conseguido almacenar la información en la base de datos Neo4j, el próximo paso es representarla correctamente. Debido a la poca información y documentación en internet con respecto al grafo de Neo4j, me he visto obligado a utilizar la herramienta Bokeh para representar dicha información almacenada.

Debido al desconocimiento de dicha librería de Python he procedido a instalarla, que gracias al IDE Pycharm ha sido muy sencillo de realizar. Antes de ponernos con el problema real y cohesionarlo todo, he ido poco a poco implementando aplicaciones más sencillas para familiarizarme con la librería.

Teniendo en cuenta las exigencias del problema, decidí utilizar el framework de Python llamado Flask mediante el cual se pueden crear fácilmente interfaces gráficas utilizando la herramienta de D3JS en lugar de Bokeh como había planteado en un principio. Una vez encontrada una aplicación de ejemplo el siguiente paso es adaptarla a nuestro problema concreto. Cuando haya completado esta tarea, el siguiente paso será documentarse acerca de la maquetación con D3JS.

Tras unas semanas familiarizándome con Flask, gracias a un repositorio qure realizaba un trabajo similar al nuestro pero con películas en lugar de canciones y artistas, conseguí hacer una aplicación de prueba que guardaba la información de la API en Neo4j y posteriormente mostraba el grafo en la pestaña del navegador. Esta aplicación relacionaba un álbum dado con las canciones de este. El siguiente paso es profundizar más con la API y con Flask para representar relaciones más complejas.