

SISTEMAS DE GESTIÓN Y BUSINESS INTELLIGENCE

Raquel Chamorro Giganto

**Informe final
de la
aplicación**

ÍNDICE

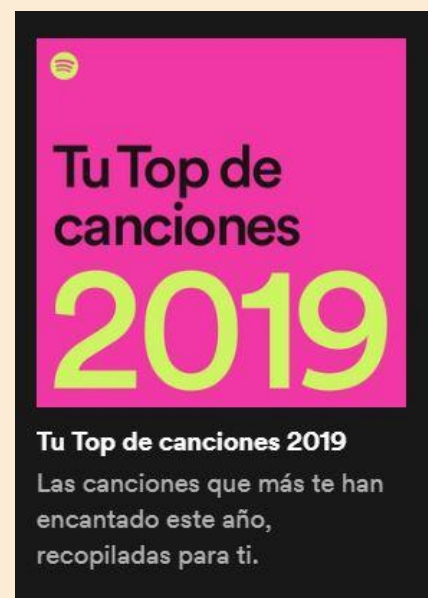
1. Introducción	2
2. Descripción del objeto de trabajo	3
3. Objetivos	3
4. Herramientas	4
5. Trabajo	9
6. Análisis crítico (DAFO)	22
7. Líneas de futuro	26
8. Lecciones aprendidas	28
9. Bibliografía	28

La aplicación “Songify”

1. Introducción

Spotify, la aplicación de streaming más usada de estos últimos años estrenó hace 4 años Spotify Wrapped, una especie de grafo de conocimiento y playlist que recopila todas tus estadísticas de escucha durante todo un año.

Este año, con una interfaz bastante atractiva, publicó una historia interactiva con datos curiosos de este año para cada usuario, incluyendo una lista con las canciones más escuchadas del año y una lista complementaria con las canciones que te “perdiste” este último año, donde se muestran canciones que nunca han sido escuchadas por ese usuario y de acuerdo a su historial de escucha y gustos propios con el objetivo de que descubra nuevas canciones que puedan ser de su agrado.



Todo ello resulta bastante atractivo al público y este año, sobre todo, ha tenido bastante expectación ya que es una costumbre para los usuarios de esta aplicación el recibir su “regalo personalizado” cada diciembre por parte de Spotify. Además de esta lista, Spotify tiene una lista que cambia cada semana especial para cada persona y listas de reproducción de diferentes géneros que escucha el usuario. Todo esto utiliza inteligencia artificial y es en lo que me he inspirado para realizar mi aplicación: Songify.

2. Descripción del objeto de trabajo

- Dificultad para encontrar un tipo de canción en concreto, ya sea por artista o por un tipo o clasificación concreta
- Dificultad para clasificar las canciones
- Falta de aplicaciones que se centren en el clasificado de canciones por parámetros musicales y técnicos
- Spotify es considerado una gran biblioteca de música dispuesto para todo el mundo y gratuitamente, aunque tenga su versión premium con funciones más concretas, el problema es que, aunque Spotify tenga su motor de búsqueda propio, no permite realizar una búsqueda precisa y más compleja, usando parámetros o características de cada canción, además de que siempre favorece a los artistas o canciones más conocidos. Con esta app se pretende que esto no ocurra y que el usuario pueda tener una gran variedad de filtros a la hora de buscar canciones, ya sea un único track concreto o sobre una recomendación basada en sus gustos o elecciones.

3. Objetivos

A simple vista, esa aplicación podría parecer similar a lo que ofrecen aplicaciones ya existentes como Spotify, Apple Music, Tidal, etc. Pero esta aplicación no es un servicio de música streaming, si no que complementa a estas apps, pues:

- Proporciona un motor de búsqueda completo que permite hacer búsquedas concretas con parámetros complejos.
- Permite escuchar una pequeña muestra demostrativa de las canciones de 30 segundos de duración.
- Ofrece recomendaciones personalizables según el género, estado de ánimo, y otros parámetros musicales.
- Clasificación y filtración de canciones por diferentes parámetros, como el género, la popularidad...

- Poder elegir el número de canciones mostradas en la búsqueda o incluso hacer un ranking de tracks según las características mostradas y que el usuario puede escoger.
- Una página de información donde se explican los parámetros usados para clasificar y hacer las recomendaciones con el objetivo de que cualquier usuario pueda entender el funcionamiento de la aplicación.
- Una interfaz accesible y fácil de usar que permita realizar todo lo mencionado anteriormente

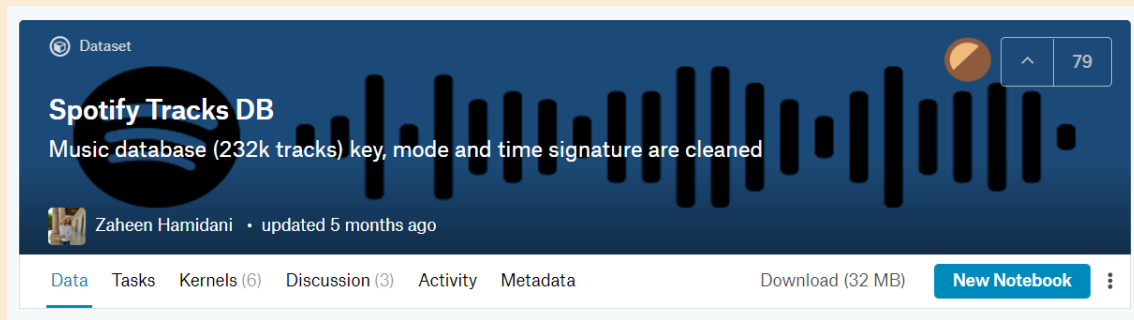
4. Herramientas

Para realizar esta aplicación he utilizado varios recursos, todos ellos gratuitos, que se pueden descargar fácilmente de internet.

- **REPOSITORIO DE DATOS**

Antes de hablar de la base de datos como tal, es necesario mencionar la fuente de información. En mi caso he utilizado un fichero de texto separado por comas (.csv) que encontré en una página llamada **Kaggle**, donde tienen una gran biblioteca de este tipo de ficheros para bases de datos. El fichero original que yo utilicé tenía más de 200.000 canciones con sus respectivos datos (artista, género, popularidad...y una serie de parámetros) obtenidos de Spotify y que ocupaba un tamaño total de 32MB (un tamaño considerable, teniendo en cuenta que es un simple fichero de texto).

El gran inconveniente de usar esta base de datos es que el fichero es de un gran tamaño y al cargarlo en la base de datos consume mucha RAM y tiempo, así que el repositorio de datos fue reducido manualmente hasta conseguir un tamaño más razonable para la base de datos (8MB). Ahora cuenta con un total de 9 géneros y alrededor de 60.000 canciones con 18 parámetros diferentes.

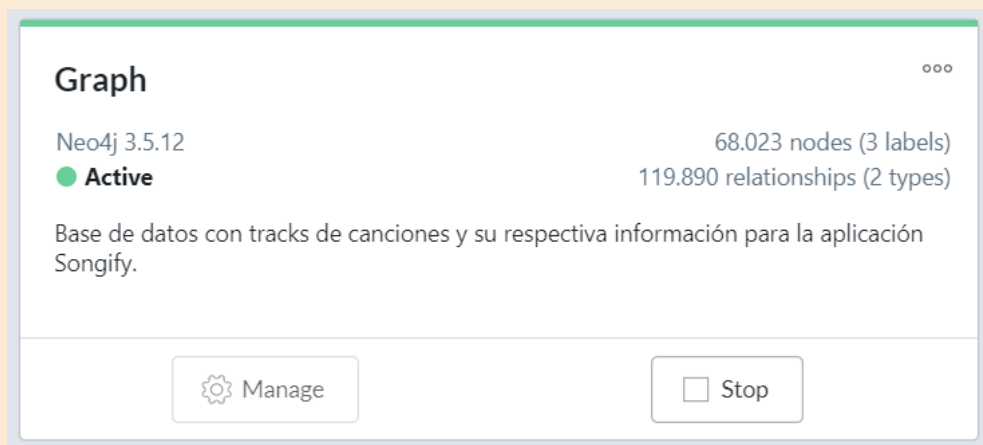


- **BASE DE DATOS**

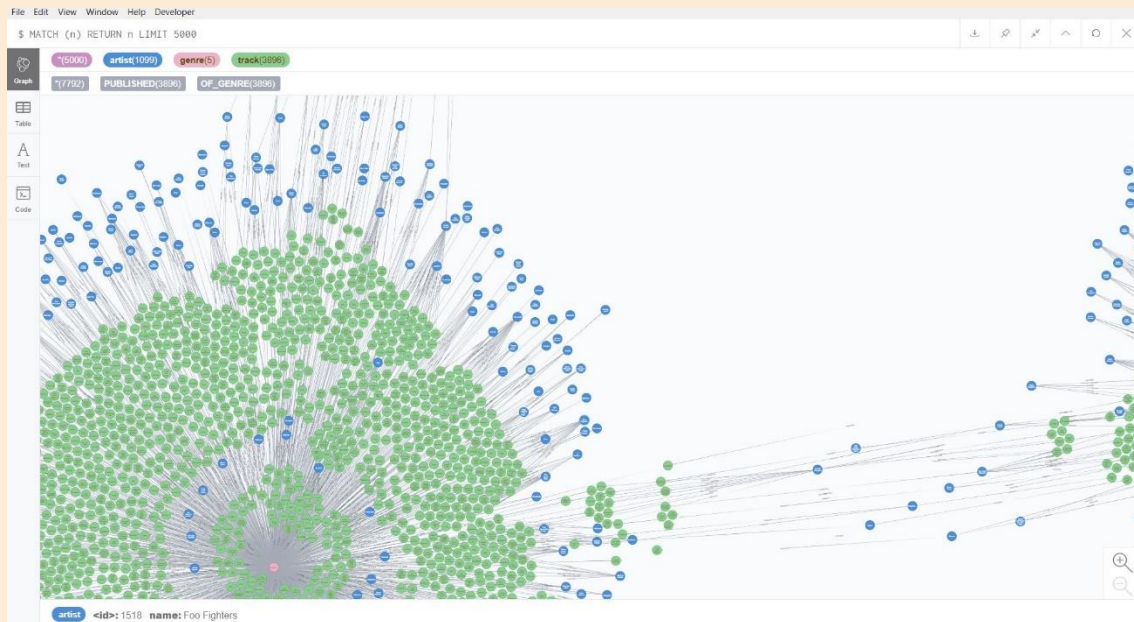
Para crear la base de datos he utilizado Neo4j, como se especificó previamente en la asignatura. **Neo4j** es una base de datos de grafos que permite realizar consultas mucho



más fácil y rápido que con las bases de datos relacionales comunes. En este caso, la base de datos que he creado cuenta con 68.023 nodos y 119.890 relaciones. Con esta base de datos de más de 68.000 nodos, la cual ya tiene una dimensión bastante considerable, se pueden realizar búsquedas en menos de un segundo buscando entre todos los nodos.



Aquí se puede observar una muestra del grafo de la base de datos de Neo4j. Cada color es un tipo de nodo distinto. El rosa, es el del género, del cual salen los tracks, que son los nodos verdes. Cada track tiene su artista que son los azules. Estos también pueden tener varios tracks y de diferente género. El grafo es tan inmenso que en el ordenador es imposible de cargar ya que requiere de mucha memoria.



Esta base de datos usa el lenguaje **Cypher** para las consultas. Este lenguaje es muy sencillo y fácil de comprender, lo que facilita realizar las “queries”. Además, es muy asequible encontrar información en foros y en su propia documentación para aprender a usar este lenguaje.

En mi caso ha sido de gran ayuda. Con una sola query en esta base de datos conseguí importar e introducir todos los datos del repositorio .csv:

```
LOAD CSV WITH HEADERS FROM 'file:///SpotifyTracks.csv' AS line
FIELDTERMINATOR ';' WITH line

CREATE (t:track {name: line.track_name, genre: line.genre,
popularity: toFloat(line.popularity), mode: line.mode, tempo:
toFloat(line.tempo), id: line.track_id, acousticness:
toFloat(line.acousticness), danceability:
toFloat(line.danceability), duration: toInteger(line.duration),
energy: toFloat(line.energy), instrumentalness:
toFloat(line.instrumentalness), key: line.key, liveness:
toFloat(line.liveness), loudness: toFloat(line.loudness), mode:
line.mode, speechiness: toFloat(line.speechiness), tempo:
toFloat(line.tempo), time_signature: line.time_signature, valence:
toFloat(line.valence)})

FOREACH (n IN (CASE WHEN line.artist_name IS NULL THEN [] ELSE [1]
END)| MERGE (a:artist {name: line.artist_name}))

CREATE (a)-[:PUBLISHED]->(t)

MERGE (g:genre {name: line.genre})

CREATE (t)-[:OF_GENRE]->(g)
```

- **FRAMEWORK (Backend y Frontend)**

Una vez que se tiene la base de datos lista, esta se tiene que introducir dentro del programa en sí, que será la aplicación web. En mi caso he usado el framework de código abierto que ofrece **Vue**, el cual tiene una gran cantidad de herramientas que permiten crear aplicaciones web de una manera sencilla, intuitiva y con un diseño actual. Vue es muy fácil de instalar por consola y lleva menos de 10 minutos crear un proyecto. Con Neo4j fue muy fácil ya que en internet se encuentra cómo introducir librerías para usar Neo4j y maneras de llamar a la base de datos, crear consultas, vincular el repositorio...



Además de usar Vue en este apartado también usé combinado **Node.js**, para la parte de JavaScript, el Backend. Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor.

Para la parte del Frontend utilicé **Vuetify**, que es un framework de Vue que permite acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes "listos para usar" usando Material Design. En mi caso he elegido estas herramientas por el hecho de que ya he trabajado con ellas y me han facilitado mucho la creación de aplicaciones web, además de que sus componentes ofrecen un diseño minimalista y cuidado para la interfaz que yo valoro bastante.



Para mostrar la aplicación en ejecución a tiempo real utilice **electron.js**, otro framework para aplicaciones web muy sencillo de usar y que se puede incorporar a Vue.

- **SPOTIFY**

Spotify ha sido usado tanto como una fuente indirecta de datos (los tracks del repositorio junto con toda su información fueron obtenidos de él), como para poder mostrar un pequeño fragmento de audio, el cual es un plugin que permite visualizar la portada del track y un pequeño reproductor. Además, cierta información que aparece en la pestaña “Track Features” ha sido obtenida de la propia documentación de Spotify.



A continuación, se muestran los enlaces a las herramientas explicadas anteriormente:

REPOSITORIO:

- Kaggle (repositorio usado):
<https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db>

BASE DE DATOS:

- Neo4j: <https://neo4j.com/>

FRAMEWORKS

- Vue: <https://vuejs.org/>
- Vuetify: <https://vuetifyjs.com/>
- Node.js: <https://nodejs.org/es/>
- Electron.js: <https://electronjs.org/>

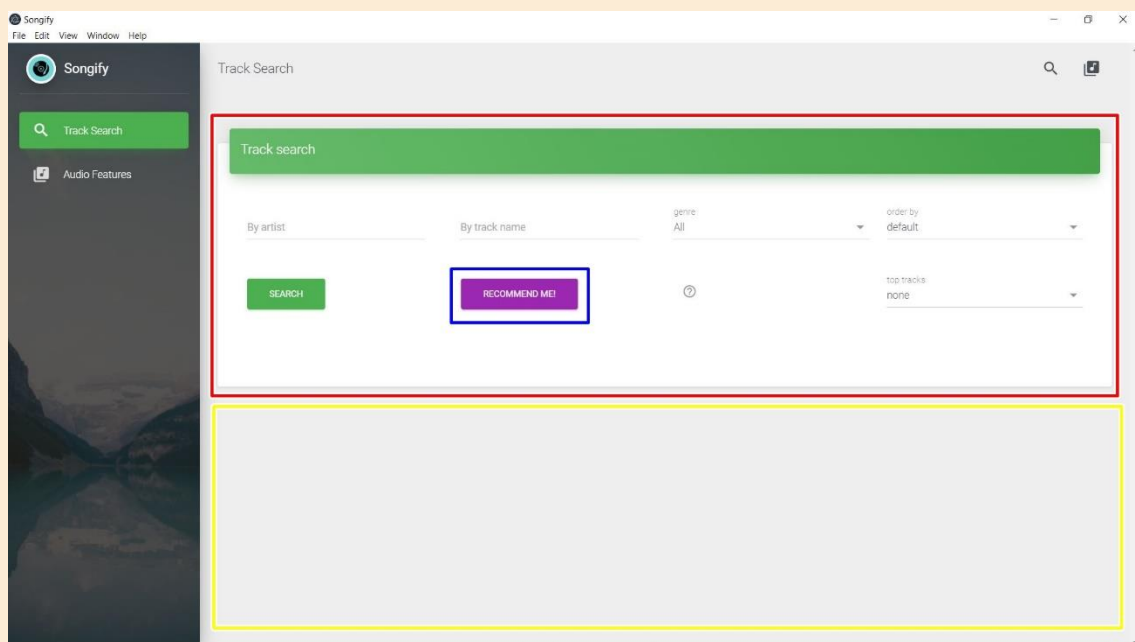
SPOTIFY

- Spotify (API): <https://developer.spotify.com/documentation/web-api/>

5. Trabajo

La aplicación **Songify** se basa en que, a partir de un directorio de datos, en este caso, de “tracks” obtenidos de Spotify, se consiga hacer una base de datos de grafos que sea utilizada para realizar consultas sobre ellas y aplicar algoritmos para obtener resultados de recomendación. Esta base de datos está compuesta de “tracks”, en total unas 60.000 canciones que tienen unos valores (género, artista, duración, popularidad...) asociados. La aplicación en sí se divide en dos páginas, la página principal, **“Track Search”** que es la de búsqueda y recomendación de tracks, y la pagina que muestra información sobre las características de cada track, y cómo se usan para realizar los algoritmos, **“Audio Features”**.

La página principal se puede dividir en tres partes claras: la parte de búsqueda normal, la parte de recomendación y la parte donde se muestran los tracks de las búsquedas como se puede ver en esta captura. Cada parte está señalada por los recuadros rojo, azul y amarillo.

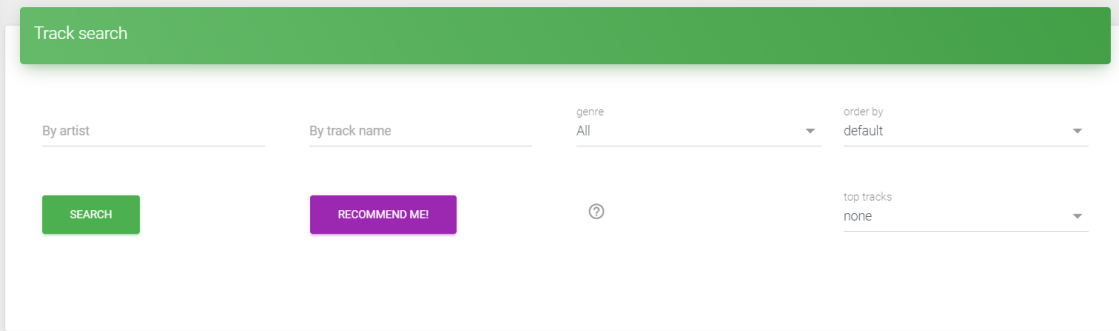


Rojo: cuadro de búsqueda

Azul: cuadro de recomendación

Amarillo: cuadro de resultados de tracks.

El **cuadro de la búsqueda** contiene 5 opciones diferentes de búsqueda para filtrar los resultados:



The screenshot shows a search interface with a green header bar labeled "Track search". Below the header, there are four main sections: "By artist" with a text input field, "By track name" with a text input field, "genre" with a dropdown menu currently set to "All", and "order by" with a dropdown menu currently set to "default". Below these sections are two buttons: a green "SEARCH" button and a purple "RECOMMEND ME!" button. To the right of the "RECOMMEND ME!" button is a small question mark icon. At the bottom right, there is a "top tracks" dropdown menu currently set to "none".

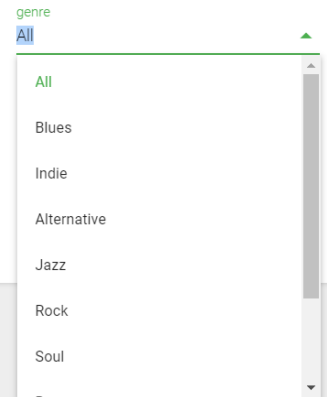
- Por nombre de la canción (by track name). Identifica canciones sin distinción de mayúsculas y de caracteres “exactos”.

By track name

- Por artista / banda (by artist). Identifica artistas sin distinción de mayúsculas y de caracteres “exactos”.

By artist

- Por género (genre). En total 9 géneros a elegir más una opción para mostrar resultados de todos los géneros.



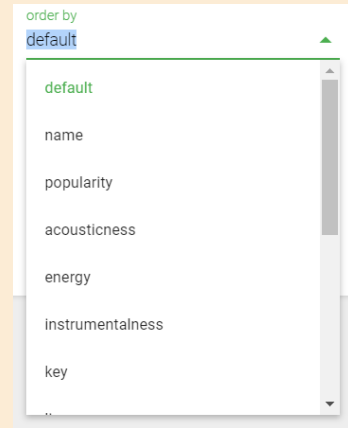
The screenshot shows a dropdown menu for the "genre" filter. The menu is open, displaying a list of genres: "All", "Blues", "Indie", "Alternative", "Jazz", "Rock", and "Soul". The "All" option is currently selected and highlighted in blue.

Los géneros son:

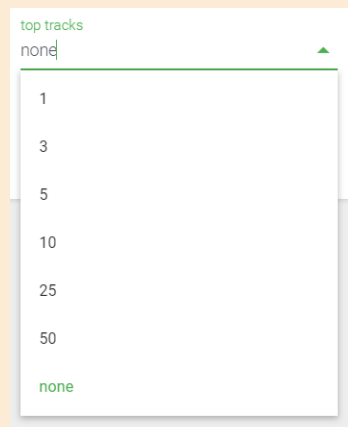
Blues, Indie, Alternative, Jazz, Rock, Soul, Reggae, Ska y Folk.

- Por un orden aplicable al resultado de la búsqueda según ciertos parámetros (order by). Estos 15 parámetros son:

- Default: orden por defecto de los tracks
- Name: orden alfabético del título del track [A - Z]
- Popularity: orden según su popularidad [0 - 100]
- Acousticness: según lo acústico que sea un track [0 - 1.0]
- Energy: según la fuerza que tenga un track [0 - 1.0]
- Instrumentalness: según lo instrumental que sea un track [0 - 1.0]
- Key: la clave en la que está el track, por ejemplo, C# (Do sostenido)
- Liveness: según si el track sea de un directo o no. [0 - 1.0]
- Loudness: según la intensidad que tenga el track (volumen) [-60 - 0 dB]
- Mode: si está en tonalidad mayor o menor [major - minor]
- Speechiness: si el track contiene voz hablada o narrada [0 - 1.0]
- Tempo: Beats por Minuto (bpm) que tiene el track [0 - 240]
- Time signature: el compás del track, por ejemplo, 4/4.
- Valence: mide la positividad del track [0 - 1.0]
- Danceability: mide cómo de bailable es un track [0 - 1.0]



- Por un límite de resultados para refinar la consulta. (top tracks). Utilizando un límite la aplicación se ahorra tiempo de computación en mostrar todas las canciones y además ordenarlas, y los resultados se muestran más rápido.

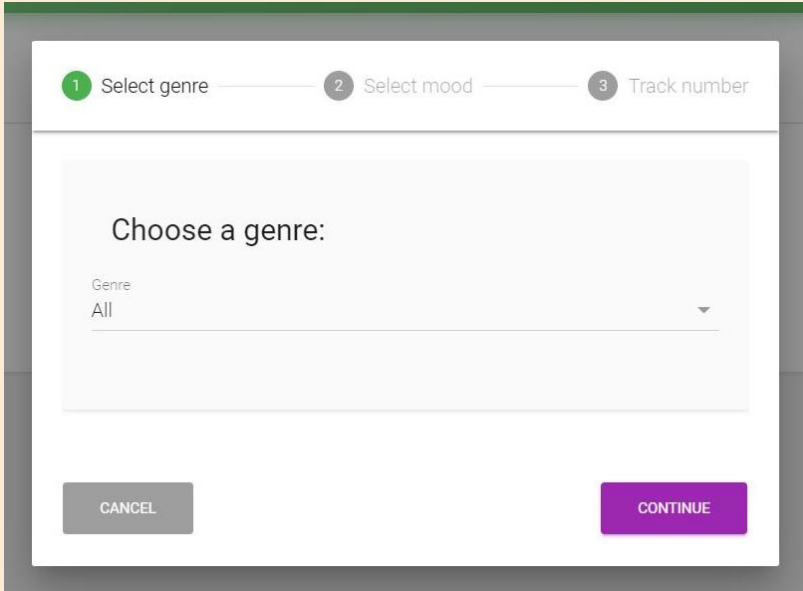


- Por último, tiene el botón de búsqueda (search) que activa la búsqueda. Una cosa a resaltar es que para realizar la búsqueda no es necesario pulsar el botón "search", sino que también se puede hacer con la tecla Enter.

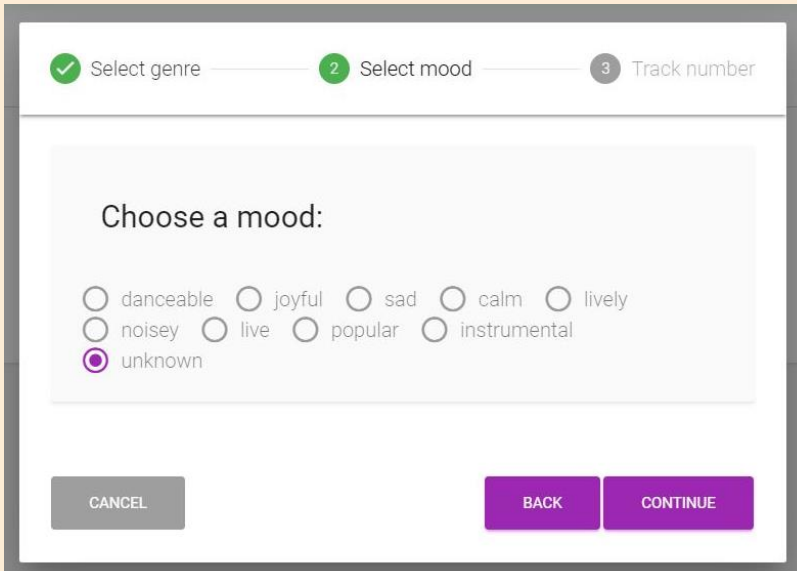
Cuadro de recomendación. Cuando se pulsa el botón “recommend me!”, se accede a un pop up que nos permite elegir en tres pasos el tipo de recomendación que queremos:

A purple rectangular button with the text "RECOMMEND ME!" in white, uppercase letters.

- Paso 1: elegir el género de la recomendación. Se puede elegir un género o que la recomendación sea de cualquiera.

A screenshot of a mobile app's recommendation selection screen. At the top, there are three steps: "1 Select genre" (highlighted with a green circle), "2 Select mood", and "3 Track number". The main area is titled "Choose a genre:" and contains a dropdown menu labeled "Genre" with "All" selected. At the bottom, there are two buttons: "CANCEL" (grey) and "CONTINUE" (purple).

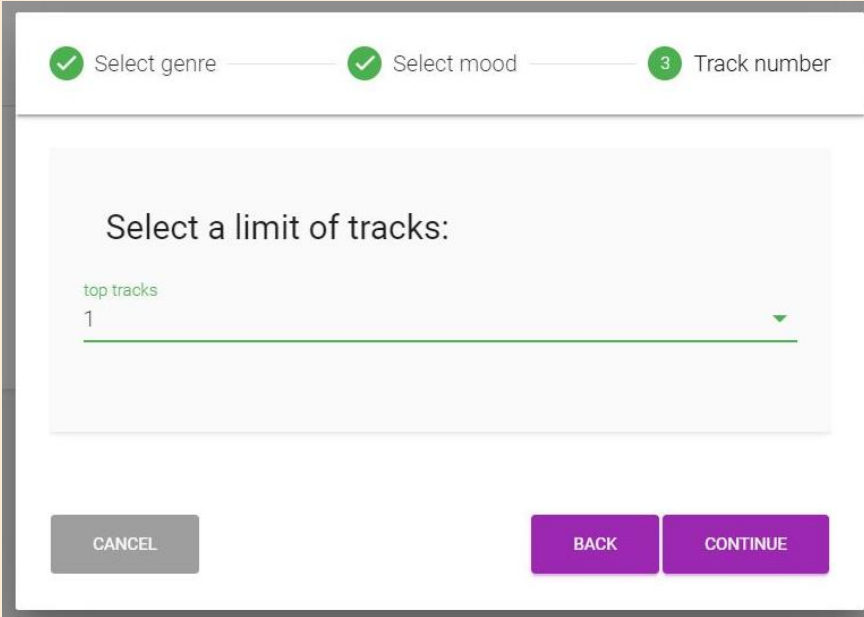
- Paso 2: Seleccionar “mood”. En este caso hay 10 tipos de recomendación:

A screenshot of the second step in the recommendation selection process. The top bar shows "1 Select genre" with a green checkmark, "2 Select mood" (highlighted with a green circle), and "3 Track number". The main area is titled "Choose a mood:" and contains ten radio button options: "danceable", "joyful", "sad", "calm", "lively", "noisy", "live", "popular", "instrumental", and "unknown" (which is selected with a purple dot). At the bottom, there are three buttons: "CANCEL" (grey), "BACK" (purple), and "CONTINUE" (purple).

- Danceable: canciones que sean bailables, animadas.
- Joyful: canciones alegres.
- Sad: canciones tristes, melancólicas...
- Calm: canciones tranquilas.
- Lively: canciones animadas y alegres que desprenden vitalidad.
- Noisy: canciones “escandalosas”, con mucho volumen y energía.
- Live: canciones tocadas en directo.
- Popular: canciones conocidas
- Instrumental: canciones sin voz, instrumentales.
- Unknown: canciones poco conocidas.

Este tipo de recomendaciones se hace utilizando como parámetros las características propias de cada canción mencionadas anteriormente, como puede ser “danceability”, que claramente es un parámetro que mide lo bailable que es una canción en un intervalo de números, en este caso entre 0 y 1. En este apartado solo se puede marcar un único mood de los que se muestran.

- Paso 3: elegir el límite de canciones a mostrar.

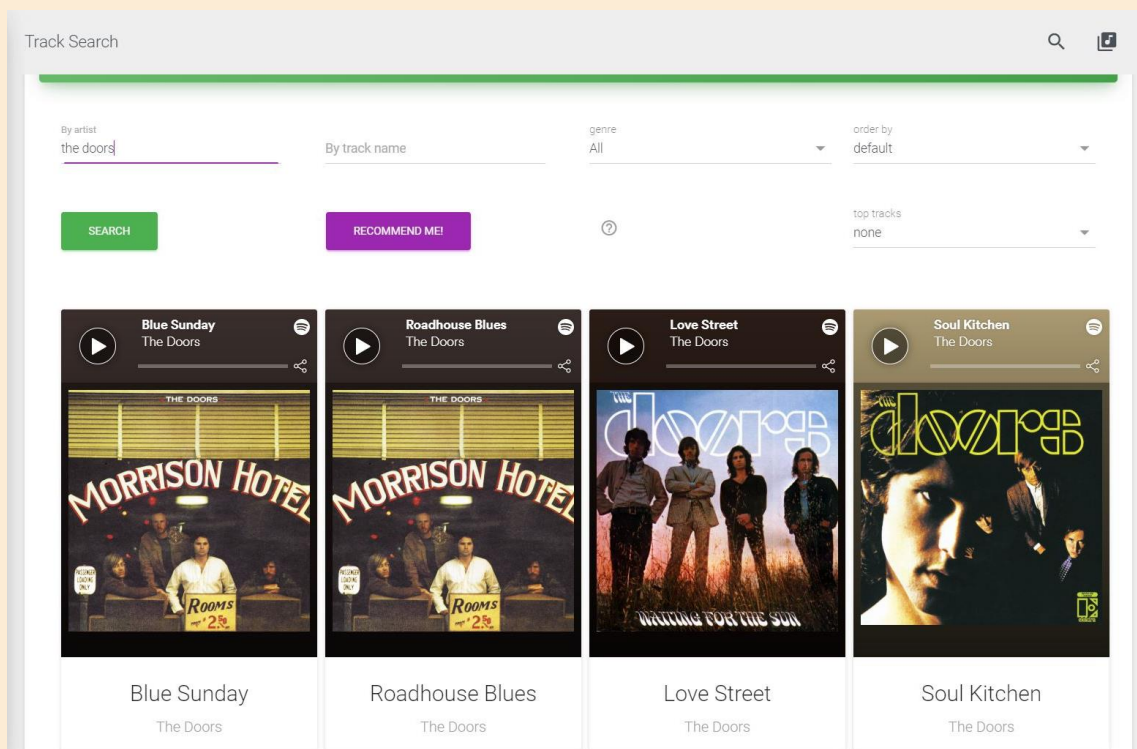


The screenshot shows a web form with three steps at the top: 'Select genre' (checked), 'Select mood' (checked), and 'Track number' (3). The main content area is titled 'Select a limit of tracks:' and contains a dropdown menu labeled 'top tracks' with the value '1' selected. At the bottom, there are three buttons: 'CANCEL' (grey), 'BACK' (purple), and 'CONTINUE' (purple).

Este límite permite agudizar la recomendación mostrando la selección que el usuario quiera, tratando de filtrar canciones que podrían ser menos precisas en cuanto a recomendación.

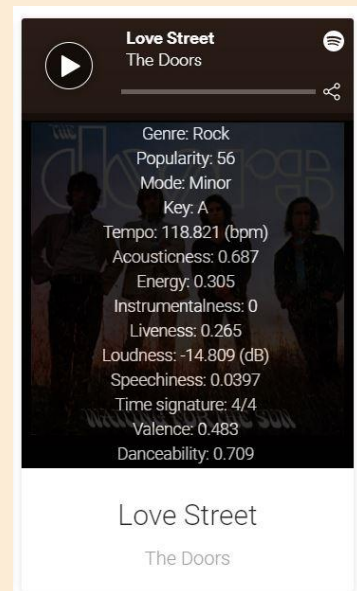
Una vez seleccionados los 3 apartados, la recomendación ya estaría hecha y aparecería los tracks concretos para esa recomendación.

Cuadro de resultado de tracks.



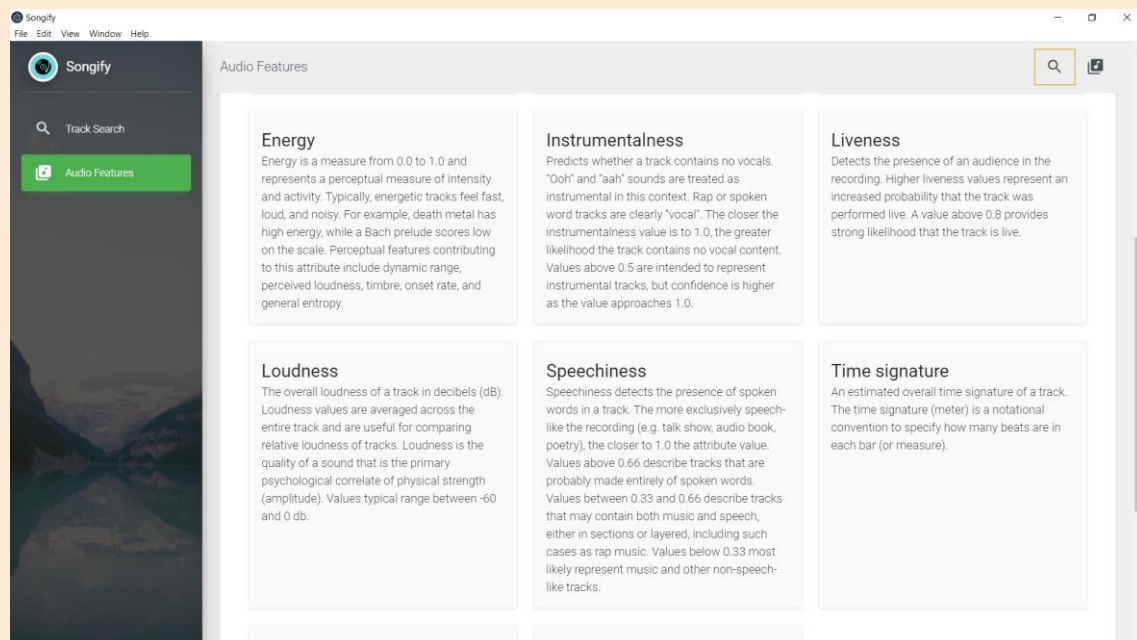
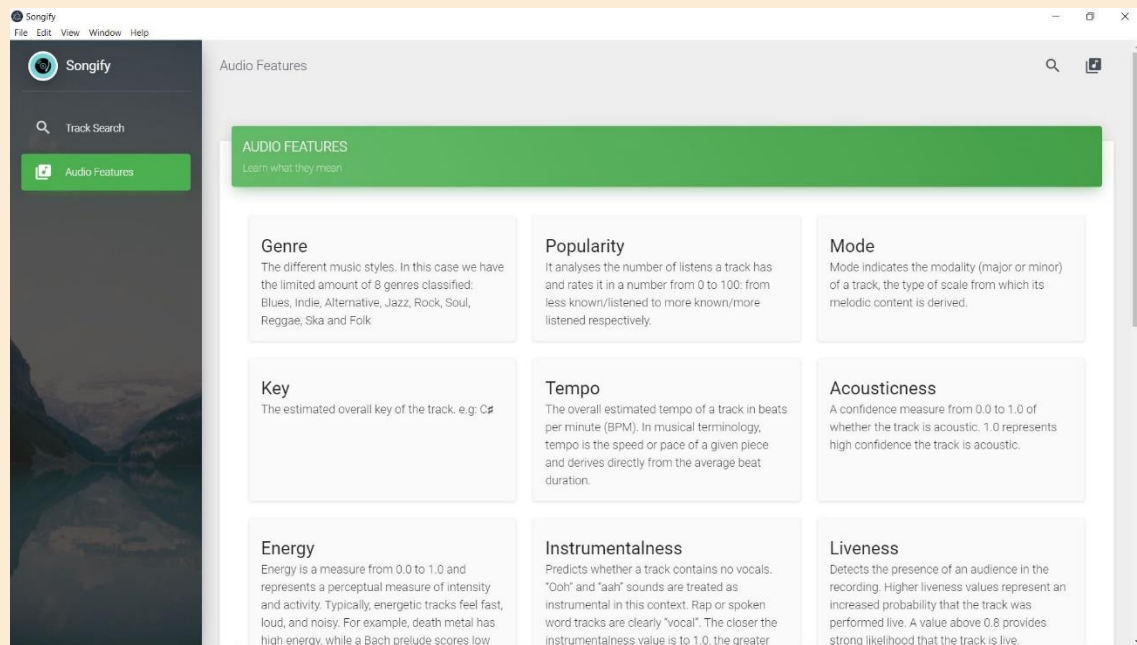
En este apartado debajo del cuadro de búsqueda aparecerán los tracks obtenidos de la búsqueda o recomendación en forma de "card".

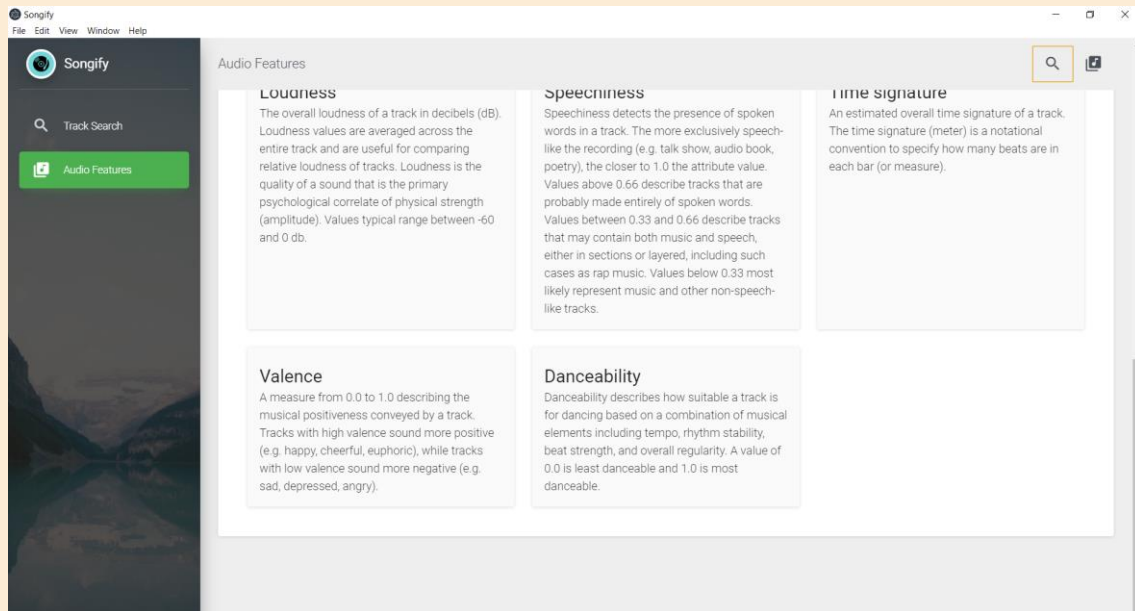
Una card en este contexto significa un elemento por separado para cada track que muestra el contenido multimedia: el plugin de Spotify con sus 30 segundos de muestra de cada canción, su respectiva portada, e incluso un botón que redirige nuestra aplicación a la aplicación de Spotify o al navegador en su defecto donde se reproduciría la canción al completo si el usuario dispusiera de cuenta de esta plataforma (Spotify). La card también muestra pasando el cursor sobre la portada otra información de ese track, como lo son todos los parámetros que hemos mencionado anteriormente y que se muestran en la siguiente imagen.



La otra página con la que cuenta la aplicación es con la “Audio Features”, la cual contiene un panel de información relevante sobre los parámetros que tiene cada track y que son usados para hacer las búsquedas y recomendaciones. En cada parámetro se da una breve definición técnica de su concepto y qué es lo que mide exactamente en el track (mayormente es un número), así como el rango que abarca y sus variaciones.

Un ejemplo de esto sería el parámetro que mide lo acústica que es una canción. Su definición nos dice que cuanto más alto es el número, mayor es la certeza de que ese track sea efectivamente acústico. A continuación se pueden observar estas características:

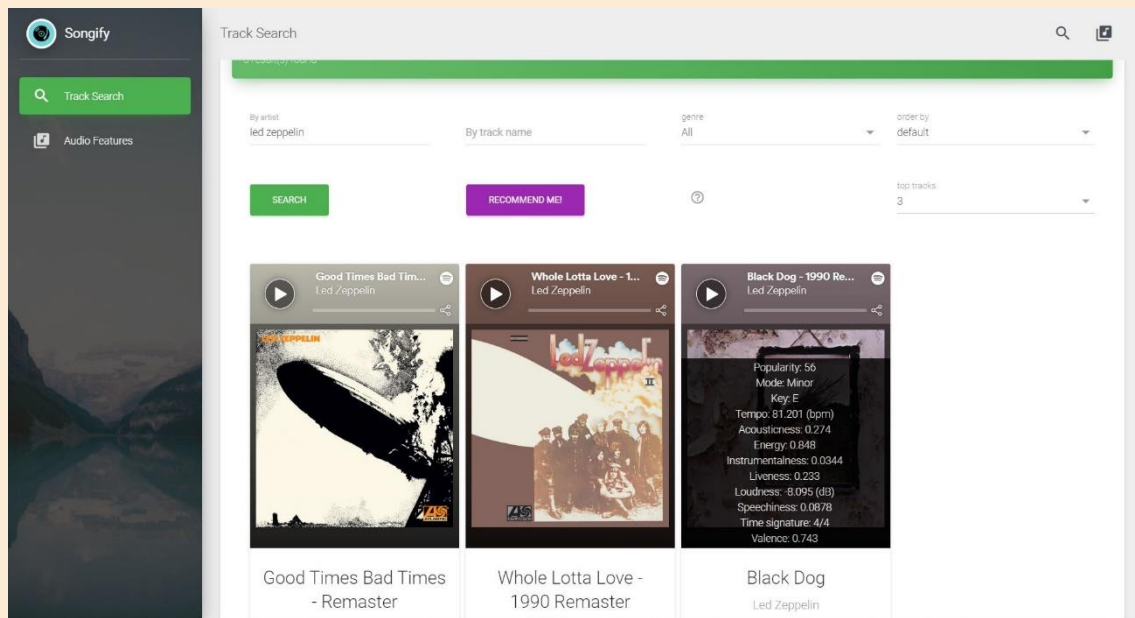




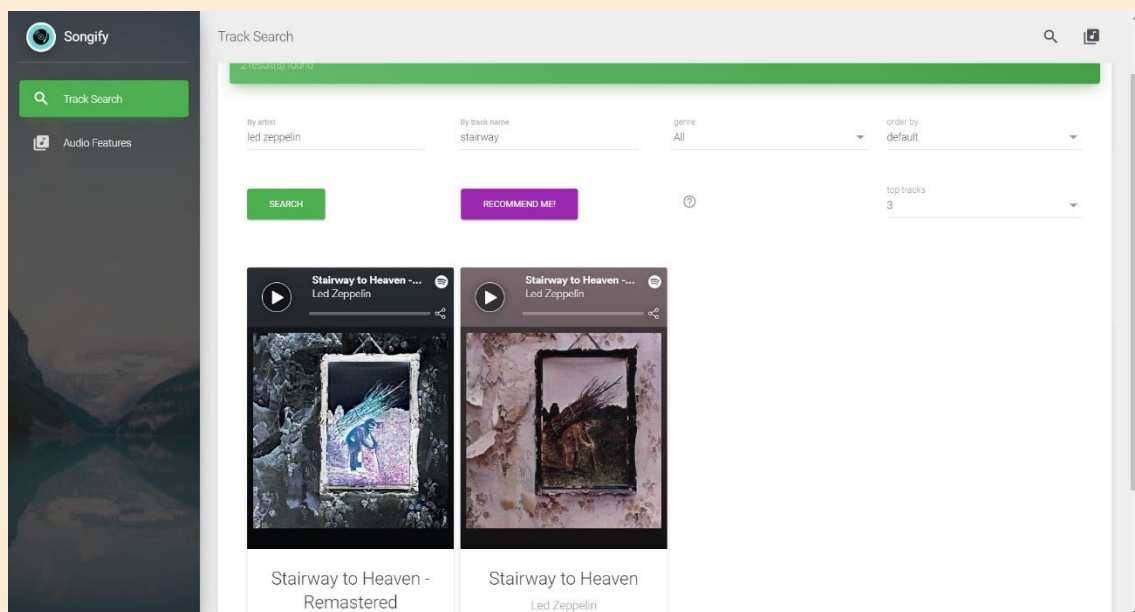
CASOS DE USOS

En este apartado se verán posibles ejecuciones de la búsqueda y de la recomendación de esta aplicación.

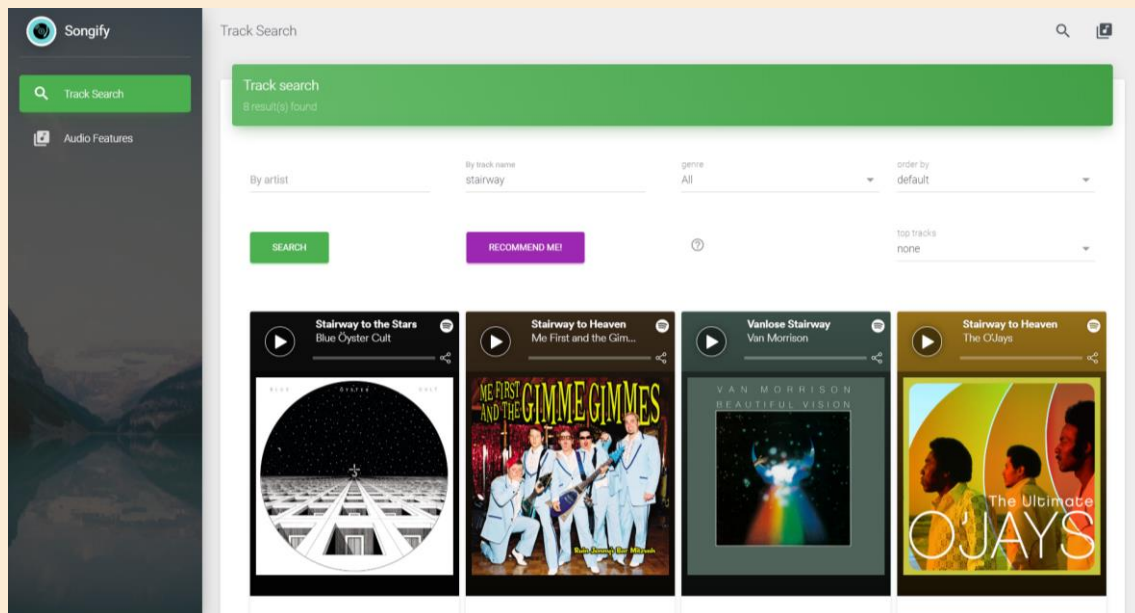
El caso más sencillo de búsqueda es introducir el artista en la caja de búsqueda y dar al botón “search”. En este caso he escogido buscar a Led Zeppelin ya que sé con seguridad que se encuentran en la base de datos. También he puesto un límite de 3 canciones para que no muestre la totalidad de su discografía (en este caso tienen más de 70 canciones en la app). Como se puede apreciar, al ejecutar la búsqueda, aparecen 3 canciones concretas de este grupo, las cuales se pueden reproducir y observar los datos (como se puede apreciar en la última canción). Esta búsqueda ha sido instantánea y no ha tardado un tiempo excesivo en cargar.



Otro caso de búsqueda simple sería buscar un track por su nombre. Además de buscarlo por su artista, como ya se vio arriba, si incluimos el nombre (entero o parcial) del track, nos aparecerá en la gran mayoría de los casos una canción concreta. En este caso la archiconocida Stairway to Heaven.



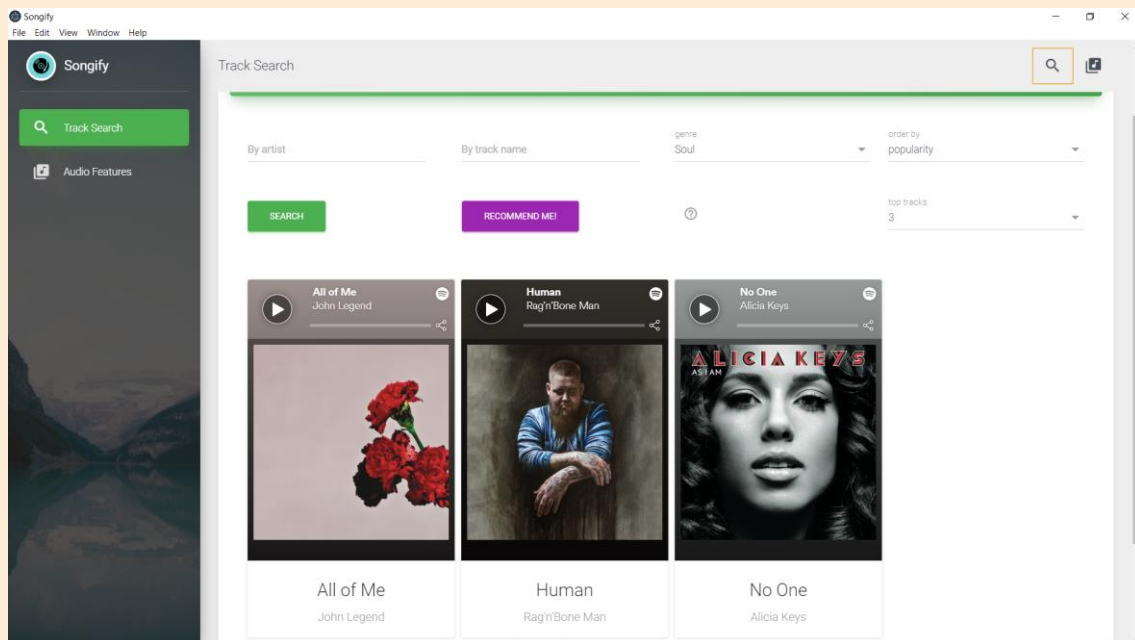
NOTA: si no hubiéramos puesto el autor podría ocurrir que nos salieran más canciones de otros artistas que contuvieran la palabra “stairway”



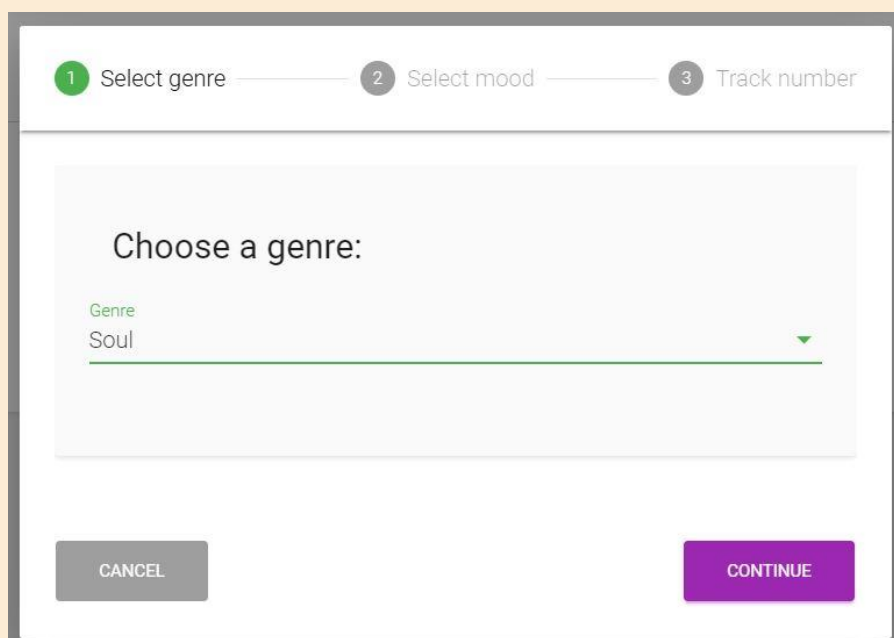
Otro posible caso de uso sería el de utilizar solamente los despleables (genero, orden y límite). Cabe recordar que si no ponemos límite es fácil que el programa deje de funcionar debido a que tendría que ordenar 60.000 tracks y mostrarlos todos.

Pongamos que queremos ver las 3 canciones más populares del género Soul.

Lo que tendríamos que hacer es simplemente marcar el género en el desplegable, elegir el orden por popularidad y poner el límite de 3. Sin necesidad de poner artista ni título del track nos aparecerán instantáneamente las canciones como se puede ver:



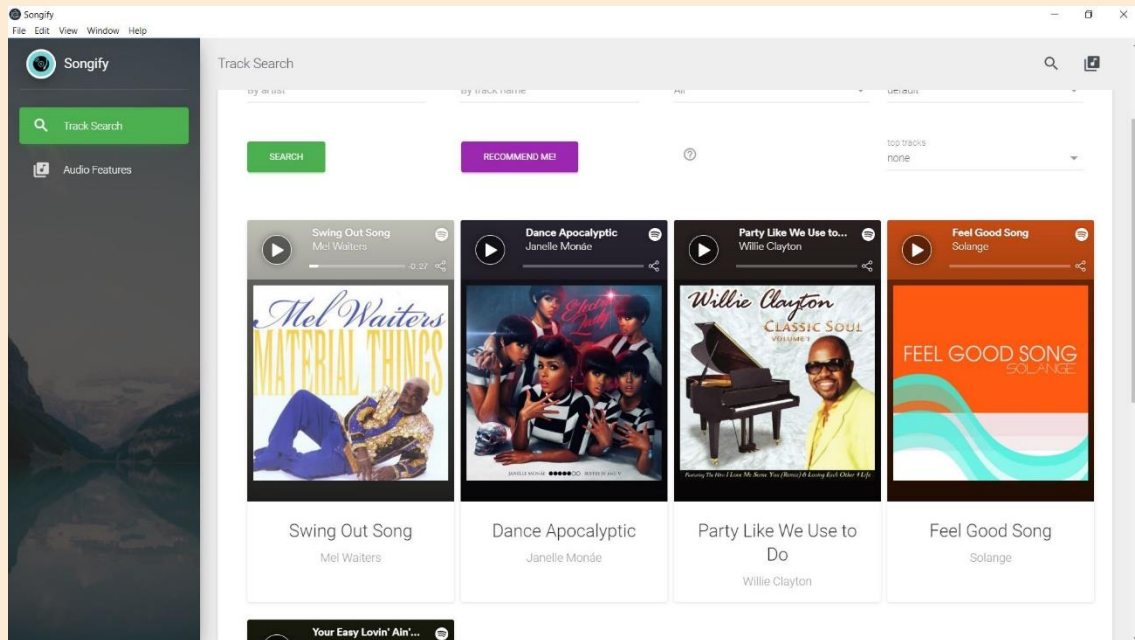
Los casos de uso del apartado de recomendación son más sencillos, puesto que no es necesario que el usuario escriba ni el nombre del artista ni el título del track. Solamente es necesario que seleccione tres campos. En este ejemplo he seleccionado dentro del género Soul, “mood” lively, es decir, canciones que tengan vitalidad, que sean animadas y un límite de 5 canciones. Estas canciones están ordenadas de las “lively” a menos



The screenshot shows a three-step process bar at the top: 'Select genre' (completed with a green checkmark), '2 Select mood' (current step, highlighted with a green circle), and '3 Track number' (pending, grey circle). Below the bar, the title 'Choose a mood:' is followed by a grid of radio button options: 'danceable', 'joyful', 'sad', 'calm', 'lively' (selected with a purple dot), 'noisy', 'live', 'popular', 'instrumental', and 'unknown'. At the bottom are three buttons: 'CANCEL' (grey), 'BACK' (purple), and 'CONTINUE' (purple).

The screenshot shows the same three-step process bar, but now 'Select mood' is completed with a green checkmark, and '3 Track number' is the current step, highlighted with a green circle. Below the bar, the title 'Select a limit of tracks:' is followed by a dropdown menu labeled 'top tracks' with the value '5' selected. At the bottom are three buttons: 'CANCEL' (grey), 'BACK' (purple), and 'CONTINUE' (purple).

A continuación, se muestra el resultado obtenido, que efectivamente se tratan de canciones animadas, alegres y positivas.



6. Análisis crítico (DAFO)

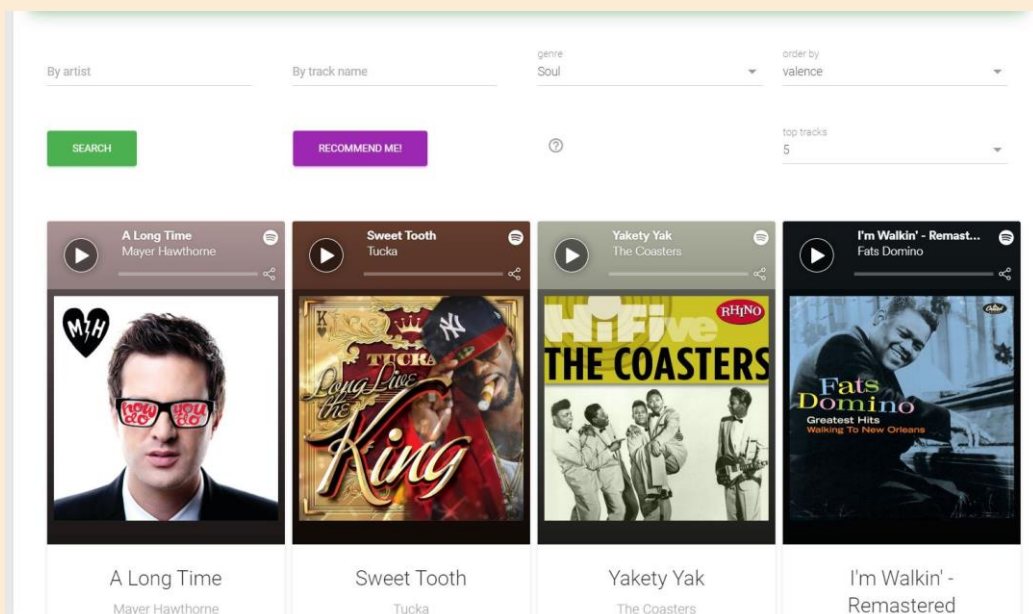
A lo largo de este trabajo me he encontrado con bastantes dificultades, tanto a la hora de realizar el código del programa como a la hora de pensar qué era exactamente lo que quería hacer. Lo primero que hice al plantearme este proyecto fue a partir del repositorio que encontré, pues me resultó atractiva ya que estoy muy interesada en el campo de la música y su análisis, lo cual también me ayudó a que me interesara este proyecto y ponerle esperanzas.

Al principio me pareció que este proyecto sería demasiado ambicioso y que no conseguiría terminarlo en el periodo de tiempo que teníamos. A pesar de que haya tenido que prescindir de algunas funcionalidades como introducir inteligencia artificial como tal en la parte de búsqueda y recomendación o de que sea de un mayor tamaño la base de datos (de 250.000 tracks pasó a 60.000 tracks).

A continuación, intentaré hacer un análisis DAFO de esta aplicación y mi experiencia con ella:

- **Debilidades:** las principales debilidades como tal se podrían resumir en que yo misma no tengo mucha experiencia con bases de datos, y menos con bases de datos de grafos relacionadas con inteligencia artificial. La falta de inspiración que tenía a principios de curso me hizo tener que pensar bastante sobre qué quería hacer la aplicación una vez que tenía el repositorio.

En cuanto al código la búsqueda en si no es siempre demasiado precisa, por el problema que existe con el género de las canciones (no siempre coinciden con el auténtico género de la canción) y porque se necesitarían más parámetros, además de mejores algoritmos, para conseguir una búsqueda o recomendación más precisa. Principalmente poder usar una red neuronal o algún tipo de inteligencia artificial más compleja. Como se puede observar en la captura y escuchando la canción en sí, la segunda canción no estaría dentro del género Soul, si no que pertenecería a Hip-Hop, lo cual da lugar a resultados erróneos.



La recomendación tiene pocos parámetros lo que hace que no sea del todo correcta ya que además de que es demasiado sencilla, los parámetros no permiten reconocer o clasificar las canciones según sus

letras, su subgénero y otros datos importantes que se necesitarían obtener de fuentes externas, además de tener que usar un algoritmo más complejo o con inteligencia artificial.

Un último fallo importante es que algunas consultas de búsqueda no funcionan del todo bien, como lo son las que solo son de ordenar elementos, ya que estas tienen que ordenar 60.000 canciones por un orden concreto, además de mostrarlas todas con su respectivo plugin de Spotify, el cual usa bastante memoria y es poco óptimo.

- **Amenazas:** La principal amenaza que me he encontrado han sido fallos en el propio repositorio sobre el género de las canciones. El género no es tan fiable como el resto de parámetros porque no está directamente extraído de Spotify si no de otra fuente externa desconocida que no aparece en la página del repositorio. Además de este fallo, intuía que también existen más errores dentro del repositorio y sus campos, como que algunas canciones estaban duplicadas, pero con distinto, género. Esto, da un problema a la hora de mostrar las canciones por su género, ya que en algunas este es erróneo. A pesar de esto, cambiando la base de datos por una que tuviera los datos correctos, no habría ningún problema.

En este repositorio he tenido que borrar y editar manualmente muchas filas de datos que eran redundantes ya que se repetía la canción, pero en distintos géneros, y otros géneros que he eliminado para reducir la base de datos... Todo ello ha llevado una cantidad de tiempo notable a la hora del desarrollo de la aplicación, cosa que me retrasó bastante.

- **Fortaleza:** a pesar de todos los inconvenientes que he mencionado anteriormente, esta aplicación también tiene puntos fuertes. La base de datos, la cual tiene 60.000 tracks actualmente, es muy completa dentro de cada género y ofrece muchos posibles tracks de recomendación

haciéndola mucho más diversa, e incluso cabe la posibilidad de que se pudiera aumentar en un futuro, pues la aplicación está hecha para adaptarse a diferentes repositorios siempre que estos conserven la misma estructura.

Otro de los puntos fuertes de esta aplicación es la introducción del plugin de Spotify, el cual aporta bastante información sobre la canción; 30 segundos de esta más su correspondiente portada, todo esto hace que la aplicación gane consistencia y tenga una buena presentación. La interfaz, en la cual puse mucho empeño también considero que aporta bastante a la aplicación, ya que su diseño minimalista y sencillo es fácil de entender por cualquier público. Además, se facilita una página donde se explican todos los parámetros para educar a los usuarios que no conozcan estas características y entiendan a la perfección el propósito de la aplicación.

La búsqueda, a pesar de que tenga ciertos errores funciona eficientemente, además posee bastantes métodos de búsqueda y recomendación lo que hace que esta aplicación tenga muchas combinaciones posibles de resultados.

- **Oportunidades:** algo aprovechable de este trabajo ha sido todo lo que he aprendido de esta oportunidad que hemos tenido para realizar una aplicación de nuestro gusto. Esto ha conseguido que me interese más en este campo, sobre todo en lo que esté relacionado con la música. Spotify es una plataforma que me ha gustado desde siempre y me resulta increíble todo lo que se puede realizar con esta herramienta. Nuestra compañera Andrea además ha sido un gran apoyo durante el proceso de creación de la aplicación. Ella me ha ayudado mucho y también me ha animado a realizar ciertas cosas en la aplicación que yo veía imposibles de realizar.

7. Líneas de futuro

En líneas de futuro se tratará de explicar tanto posibles mejoras o actualizaciones que podría tener esta aplicación como los propios usos que se le podría dar en un futuro.

Después de haber terminado con toda la aplicación, surgen ideas que se podrían realizar en este campo. Concretamente en el campo de recomendaciones de música con inteligencia artificial y redes neuronales. Además de para disfrute personal, esta aplicación podría usarse con otros fines que faciliten el trabajo en alguna profesión, como puede ser:

- Para el estudio de su propia música, en campos como la rama de la musicología, la ingeniería del sonido, estudios estadísticos relacionados con estos campos. Esta aplicación podría servir para predecir ciertos patrones en la música, estudiar su composición melódica, analizar ciertas cualidades de sonido...
- Podría emplearse en el campo de musicoterapia, para encontrar las canciones adecuadas de acuerdo a ciertas condiciones que las hagan aptas para uso terapéutico.
- Para trabajos que requieran reproducir medios audiovisuales, como puede ser en una cadena de radio un locutor que necesite encontrar unas canciones determinadas para crear cierta lista de reproducción de un género o estilo...al igual que le pasaría a un DJ profesional que necesite por ejemplo canciones con el mismo tempo para crear una sensación de continuidad. O incluso en películas para crear cierto ambiente o en programas de televisión para encontrar la canción adecuada. Esto lo podría realizar ahora mismo la aplicación.

Siempre y cuando esta aplicación se consiga desarrollar con inteligencia artificial todos los propósitos de uso anteriores podrían hacerse realidad. Es por esto que propongo unas mejoras que serían necesarias para darle futuro a la aplicación:

- Añadir una recomendación según un historial del usuario. Para ello sería necesario almacenar en una base de datos todas las búsquedas y escuchas de este usuario para luego conseguir obtener un resultado, también necesitando una especie de Login con credenciales para que se puedan almacenar correctamente el historial y las preferencias de cada usuario registrado en esta plataforma. Esto requeriría de inteligencia artificial para predecir a partir de una cantidad de escuchas el gusto del usuario.
- Al igual que con la recomendación, la búsqueda también podría ser mejorable, haciéndola más compleja añadiendo más parámetros o incluso usando la inteligencia artificial para que además de que recomiende tracks por ejemplo por su letra, sentimientos que expresa, etc., también pueda recomendar artistas que tengan cierta relación entre ellos.
- Añadir otro repositorio más completo y correcto que se vaya actualizando, pues el que existe actualmente tiene sus fallos como se comentó anteriormente. Además, sería una buena idea añadir más géneros y subgéneros para que las recomendaciones y búsquedas sean mucho mas precisas en cuanto a similitud. Otra posibilidad sería pedir permisos de developer a Spotify para poder trabajar con su biblioteca de canciones para extraer más datos y tener siempre una base de datos actualizada con las canciones más recientes, ya que el actual repositorio solo tiene canciones de hasta principios del 2019.

En conclusión, queda claro que el principal objetivo para que esta aplicación pudiera tener algún futuro y más usabilidad sería el poder añadir inteligencia artificial, ya que está a la orden del día y actualmente existe mucha competencia de mercado en el mundo de las recomendaciones musicales, así que es necesario invertir mucho en este campo para conseguir algo distinguible de lo que ya existe.

8. Lecciones aprendidas

Con esta asignatura y en concreto con la elaboración de la aplicación saco bastantes conclusiones y lecciones, además de experiencia. Durante este cuatrimestre he aprendido a trabajar aún mejor con las herramientas explicadas en otros apartados anteriores. Además, hemos tenido que aprender a usar Neo4j, una herramienta que nos ayudará bastante en el futuro y que es casi imprescindible a la hora de trabajar con redes neuronales, grafos de conocimiento...en general inteligencia artificial con bases de datos modernas y actuales.

También he aprendido a trabajar con repositorios en csv, a mejorar mi habilidad con herramientas para elaboración de aplicaciones web (Vue, Node.js Vuetify, Electron.js...) y en general a ganar experiencia programando y aprender de errores. También trabajar con el plugin de Spotify me ha dado una idea diferente a lo que yo tenía sobre esta herramienta, ya que pensaba que sería bastante difícil trabajar con ella. También he aprendido a administrar el tiempo de elaboración de una aplicación y posteriormente realizar este informe que me servirá de predecesor y ejemplo para el Trabajo de Fin de Grado.

9. Bibliografía

REPOSITORIO:

- Kaggle (repositorio usado):
<https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db>

BASE DE DATOS:

- Neo4j: <https://neo4j.com/>

FRAMEWORKS

- Vue: <https://vuejs.org/>
- Vuetify: <https://vuetifyjs.com/>

- Node.js: <https://nodejs.org/es/>
- Electron.js: <https://electronjs.org/>

SPOTIFY

- Spotify (API): <https://developer.spotify.com/documentation/web-api/>

INSPIRACIÓN:

- “Spotify y la Inteligencia Artificial en 2018”, BigData Magazine: <https://bigdatamagazine.es/spotify-la-inteligencia-artificial>
- “Por qué la inteligencia artificial no puede crear la canción del verano”, La Vanguardia, Cristina Sáez: <https://www.lavanguardia.com/ciencia/20180717/45948321612/francois-pachet-musica-inteligencia-artificial.html>