

2019

wRap



Rodrigo Alarma Álvarez

Unileon

18/12/2019



Índice

- 1.- Introducción (página 3)**
- 2.- Objetivos (páginas 3-5)**
- 3.- Herramientas (páginas 5-11)**
- 4.- Desarrollo del trabajo (páginas 11-28)**
- 5.- DAFO (páginas 29-30)**
- 6.- Líneas de futuro (páginas 31-32)**
- 7.- Lecciones aprendidas (página 33)**
- 8.- Conclusión (página 33)**
- 9.- Bibliografía (páginas 34-36)**



1.- Introducción

La música es universal, todos estamos de acuerdo con eso, y a cada persona le gusta un género u otro. Si bien es cierto, en el caso del rock es muy fácil descubrir otros grupos similares a los que te gustan, igual ocurre con la música clásica o con la música comercial actual.

Pero, ¿qué hay del rap? Este género es generalmente música de barrio, no tan comercial, y por tanto la mayor parte de los artistas que lo componen no son nada, o muy poco conocidos, y solo constan de un canal de YouTube, donde suben sus canciones. Debido a esto, la única recomendación que puede recibir un aficionado de rap, en muchos casos, es el algoritmo de YouTube, o el de Spotify.

Al darme cuenta de este problema, decidí desarrollar una herramienta que pudiera solucionarlo, así surgió wRap. wRap es una página web de recomendación de canciones de rap, y géneros similares, en español. En ella, el usuario puede obtener una recomendación completamente aleatoria o una basada en su estado de ánimo, dado que las canciones tienen asociado uno, que es en el que se suelen encontrar los usuarios cuando las escuchan.

2.- Objetivos

Una vez que me di cuenta de la dificultad que existía a la hora de encontrar música de rap afín a los gustos de cada persona, comencé a pensar cómo aprovechar al máximo una base de datos de Neo4j con el fin de realizar las mejores recomendaciones posibles. Fue entonces cuando se me ocurrió la idea de la página web, puesto que me daba la posibilidad de poder acceder tanto en ordenador como en móvil, de forma rápida y sencilla, recibir una recomendación, y buscarla en otra plataforma complementaria al instante. Esto último fue lo que más llamó mi atención, dado que solo teníamos un par de meses para realizarlo cumpliría los objetivos a corto plazo, y sentaría las bases para el objetivo a largo plazo. Pero, ¿cuáles son estos objetivos?

Objetivos a corto plazo:

- Crear una herramienta que me fuera útil a mí mismo.
- Que la herramienta fuera accesible, rápida, eficaz y minimalista.
- Que el centro o la parte más importante fuese la base de datos.



Objetivo a largo plazo:

- Evolucionar la herramienta a un centro de contenidos multimedia personalizados de rap.

Propósito

El logo de wRap son unos auriculares con una onda que se transmite desde el lado izquierdo al derecho. Representa la música sonando, es simple y con un mensaje muy visual.

El nombre de wRap viene de la conjunción entre la “w” de palabras como “world” (mundo en inglés) o “web” (World Wide Web, red informática mundial) junto a “rap”, y cumple el propósito básico de unir a gente con artistas desconocidos para ellos, y viceversa. Es decir, por un lado al usuario le presenta música nueva que se asemeja a lo que le gusta, o a lo que escucha según su estado de ánimo. Por otro lado, al músico le presenta la oportunidad de darse a conocer, independientemente de su número de seguidores en redes sociales.

- Para el músico: wRap es la página web que se hará con la excelencia del panorama del rap en español, pronto los aficionados a este género la visitarán cada día en busca de nueva música gracias a su algoritmo de recomendación. Por ello, los artistas que estén en nuestra base de datos contarán con una ventaja comercial respecto a aquellos que no lo estén.
- Para el usuario: wRap cuenta con un algoritmo de recomendación basado en el estado de ánimo en un comienzo, pero que será mejorado a personalización y gustos del usuario que realice la búsqueda. Por ello, y al ser específico de rap y géneros similares, los usuarios de Spotify, YouTube, SoundCloud, y otras plataformas preferirán wRap cuando quieran escuchar los géneros musicales de nuestra base de datos, ya que no solo incluyen artistas desconocidos, sino que encima la recomendación es mucho más específica.

Nichos de mercado:

wRap se dirige a personas desde los 14 hasta los 28 años en España, y como ya se ha mencionado anteriormente, hay dos segmentos principales a los que nos dirigimos, los músicos que hacen rap en español, y los fans de este género.



- Músicos: Las plataformas actuales no ofrecen ventajas a los artistas desconocidos, lo que dificulta su crecimiento.
- Fans: Las plataformas actuales no ofrecen buenas recomendaciones para este género.

3.- Herramientas

Actualmente wRap es una página web alojada como aplicación de escritorio gracias a Electron.js, programada con Vue.js (JavaScript), y conectada de forma local a una base de datos de grafos de Neo4j (Cypher).

Pero, ya que el “core” de este trabajo es la base de datos de grafos, es importante dejar claro qué es una base de datos de este tipo.

Según los desarrolladores, una base de datos de grafos es una base de datos diseñada para tratar las relaciones entre los datos con la misma importancia con la que se tratan los datos en sí. Está destinado a contener los datos sin restringirlos a un modelo predefinido. Por el contrario, los datos se almacenan como los sacamos primero, mostrando cómo cada entidad individual se conecta o se relaciona con otras.

Tal y como se menciona anteriormente, es igual de importante los nodos que las relaciones entre ellos, por ello es necesario explicar:

- Nodos: Los nodos son las entidades en el grafo. Pueden contener cualquier número de atributos, también llamados propiedades, y se pueden etiquetar con labels, etiquetas, que representan sus diferentes roles en su dominio. Las etiquetas de los nodos también sirven para adjuntar metadatos (como información de índice o restricciones) a ciertos nodos (What Is a Graph Database and Property Graph | Neo4j).
- Relaciones: Las relaciones proporcionan conexiones dirigidas, nombradas y semánticamente relevantes entre dos nodos. Una relación siempre tiene una dirección, un tipo, un nodo de inicio y un nodo final. Al igual que los nodos, las relaciones también pueden tener propiedades, que en la mayoría de los casos son propiedades cuantitativas, como pesos, costos, distancias, clasificaciones, intervalos de tiempo o fortalezas. Debido a la forma eficiente en que se almacenan las relaciones, dos nodos pueden compartir cualquier número o tipo de relaciones sin sacrificar el rendimiento. Aunque se almacenan en una dirección específica, las relaciones siempre se pueden navegar de manera



eficiente en cualquier dirección (What Is a Graph Database and Property Graph | Neo4j).

En el caso de WRap, la base de datos consta de 89 nodos y 88 relaciones, siendo 78 nodos de canciones, 4 nodos de músicos, divididos en 2 artistas y 2 grupos; 6 nodos de álbumes, y 1 para el género, por si en un futuro se amplía a más. Además, las relaciones son “GRUPO”, “ARTISTA”, “INTERPRETA” y “CANCION”, para conectar los grupos con el género, los artistas con el género, los álbumes con los artistas o grupos, y las canciones con los álbumes, respectivamente.

Las tecnologías usadas son:

Neo4j:



The #1 Database for Connected Data

- Es una base de datos de grafos nativos de código abierto NoSQL que proporciona un Backend transaccional compatible con ACID para sus aplicaciones.
- Se usa para crear la base de datos.

Cypher:

- Es un lenguaje de consulta declarativo similar a SQL, pero optimizado para grafos.
- Se usa para programar la base de datos.

**JavaScript:**

- Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico (JavaScript | Wikipedia).
- Se usa para programar la página web.

Vue.js:

- Es un framework progresivo para la construcción de interfaces de usuario. A diferencia de otros frameworks, Vue está diseñado desde cero para ser



gradualmente adoptable. La biblioteca principal se centra solo en la capa de vista y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar aplicaciones sofisticadas de una sola página cuando se usa en combinación con herramientas modernas y bibliotecas de soporte (Introduction — Vue.js).

- Se usa para diseñar el Frontend o interfaz gráfica con la el usuario interactúa con nuestra web.

Vuetify:



- Es una biblioteca de Vue UI con componentes hechos a mano.
- Se usa para diseñar la interfaz de usuario.

Electron.js:

- Es un framework pensado para crear aplicaciones "nativas" de escritorio usando JavaScript, HTML y CSS. Básicamente ejecuta tu código en node.js y realiza el render de la app en Chromium (Carracedo, S. (2017). Usando Vue.js + electron.js para crear aplicaciones de escritorio).
- Se usa para alojar y mostrar el resultado que la página web tendría en Internet, de una forma mucho más sencilla, ideal para la etapa de desarrollo, puesto que los cambios del Frontend se actualizan automáticamente sin tener que reiniciar el servicio.



En el futuro se espera actualizar el Backend a Python con la ayuda de Flask, de forma que continúe con las ventajas que proporciona una página web, así como el Frontend de Vue.js, pero permitiendo la inclusión de algoritmos de inteligencia artificial desarrollados en Python.

Las tecnologías que se añadirían son:

Python:



- Es un lenguaje de programación que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico, multiplataforma y código abierto, cuya filosofía hace hincapié en la legibilidad de su código (Python | Wikipedia).
- Se usaría para programar el Backend de la página web, así como los algoritmos de inteligencia artificial.

Flask:

- Es un framework escrito en Python que permite crear aplicaciones web rápidamente y con un número mínimo de líneas de código (Flask | Wikipedia).
- Se usaría para adaptar la aplicación de Python a página web.

Py2Neo:



- Es una librería y un toolkit para trabajar con Neo4j desde aplicaciones de Python y desde la línea de comandos. La biblioteca admite tanto Bolt como HTTP y proporciona una API de alto nivel, un OGM, herramientas de



administrador, una consola interactiva y muchas otras características (Small, N. (2019). The Py2neo v4 Handbook — The Py2neo v4 Handbook).

- Se usaría para conectar la aplicación hecha mediante el lenguaje de programación Python con la base de datos de Neo4j.

Respecto a los algoritmos de inteligencia artificial mencionados anteriormente, se espera que sea capaz de cumplir las cuatro etapas básicas y necesarias:

Recolectar datos:

- Es la conjunción entre los datos que proporciona el usuario dentro de la página, buscando unas canciones u otras, valorando canciones, etc.; y los que proporciona fuera de la página gracias a las cookies, a la conexión con Google o Facebook, etc.; con la que conocemos la música que busca fuera de nuestra página y sus gustos o aficiones entre otras cosas de posible utilidad (Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación?).

Almacenar datos:

- Toda esta información debe ser almacenada de forma inteligente y automatizada, para poder ser procesada después (Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación?).

Analizar datos:

- Aquí el propósito es el tratamiento de los datos recolectados y almacenados anteriormente, de forma que obtengamos las mejores recomendaciones posibles para el usuario, siendo estas completamente personalizadas (Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación?).

Extraer una recomendación:

- Gracias al análisis anterior, ahora el usuario recibirá la mejor recomendación posible para él (Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación?).

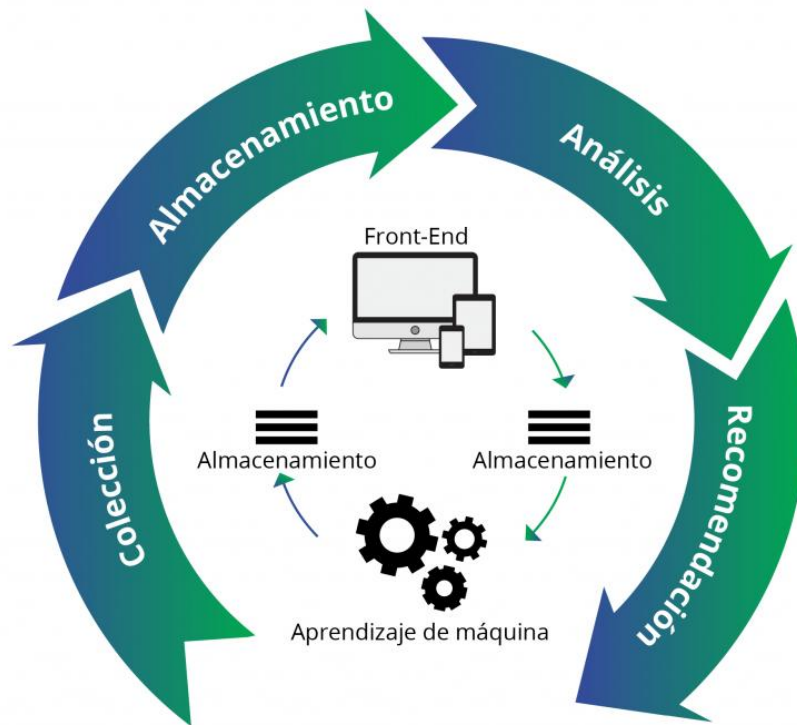


Ilustración 1 - Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación?

4.- Desarrollo del trabajo

Desarrollo del logo:

Para desarrollar el logo, así como sus diferentes versiones en la aplicación, primero elegí una imagen gratuita de la página pixabay. Después, en la página Canva, seleccioné la opción “Crea un diseño”, y en ella “Logo”. Entonces modifiqué la imagen original y la importé al programa de edición “GIMP”. En él, modifiqué colores, y quité el fondo a las imágenes que lo requerían. Todas estas imágenes se pueden encontrar en mi GitHub, en la carpeta “assets”, dentro de “src”, dentro del Frontend.



Base de datos:

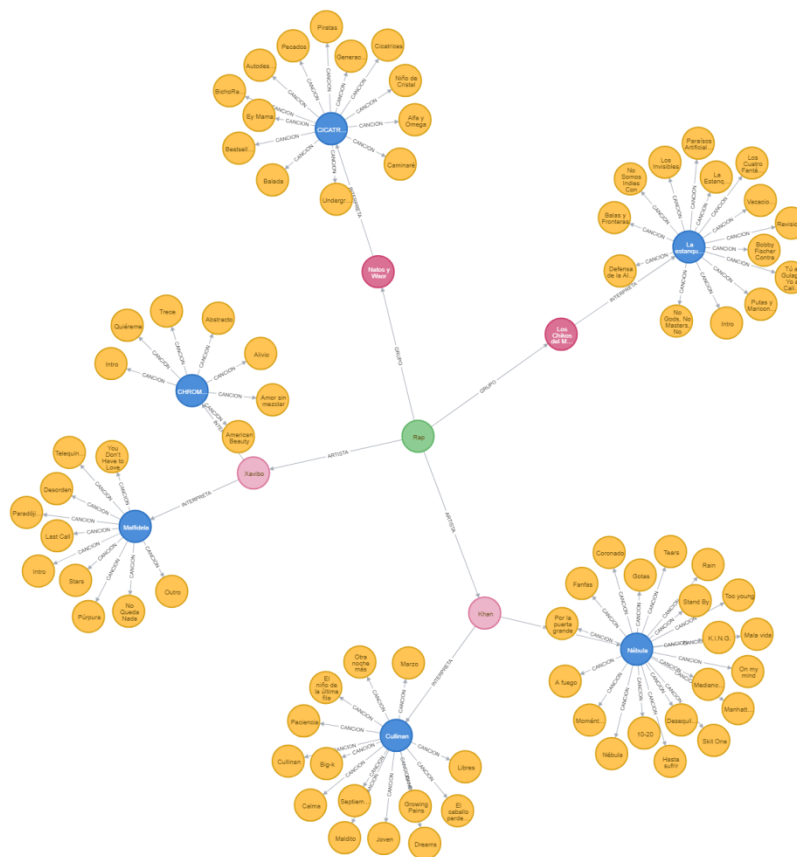


Ilustración 2 - Base de datos de wRap



Interfaz de usuario:

La interfaz de usuario general es la siguiente:

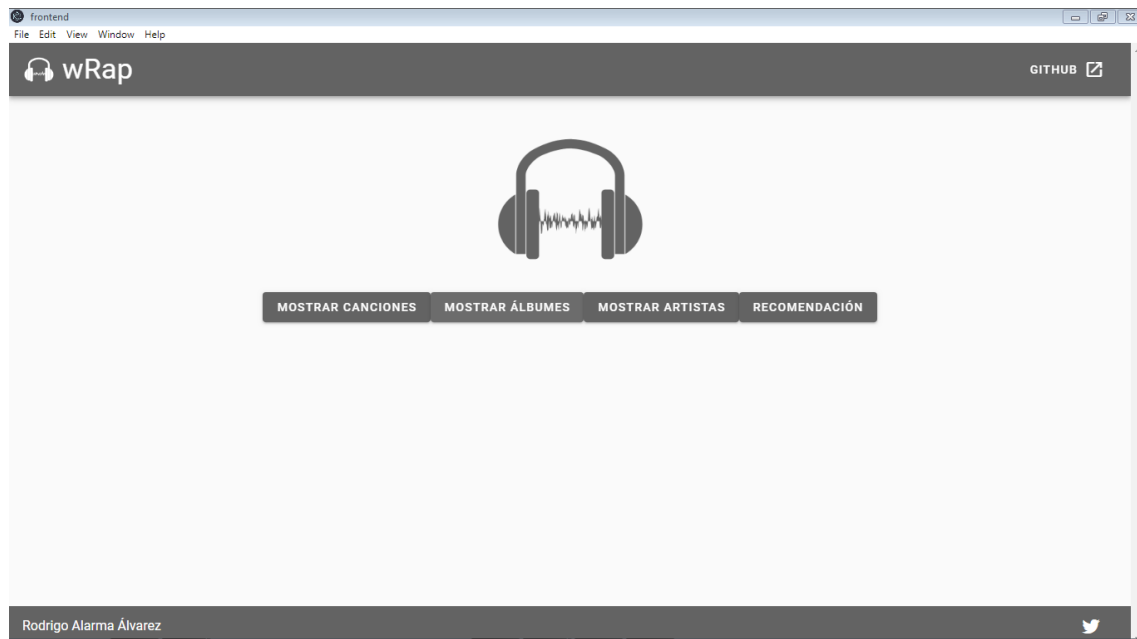
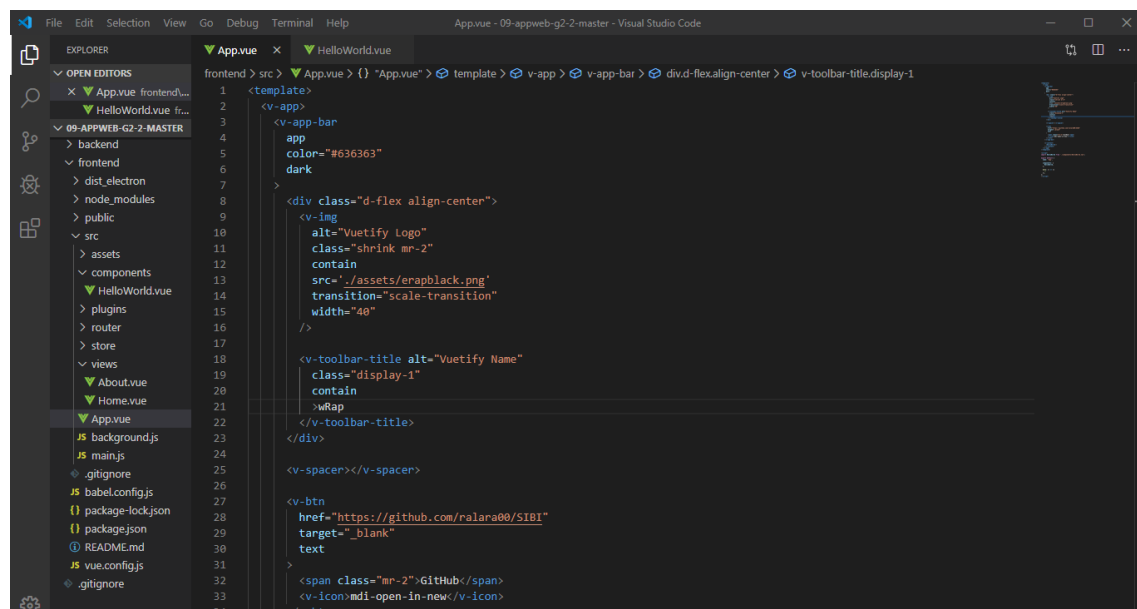


Ilustración 3 - captura de la interfaz de wRap

El código de la interfaz de usuario es muy simple, ya que sencillamente en una barra superior con un título y un logo en la parte izquierda, y un botón en la derecha:





El botón en la derecha abre una pestaña que dirige al usuario a mi GitHub personal:

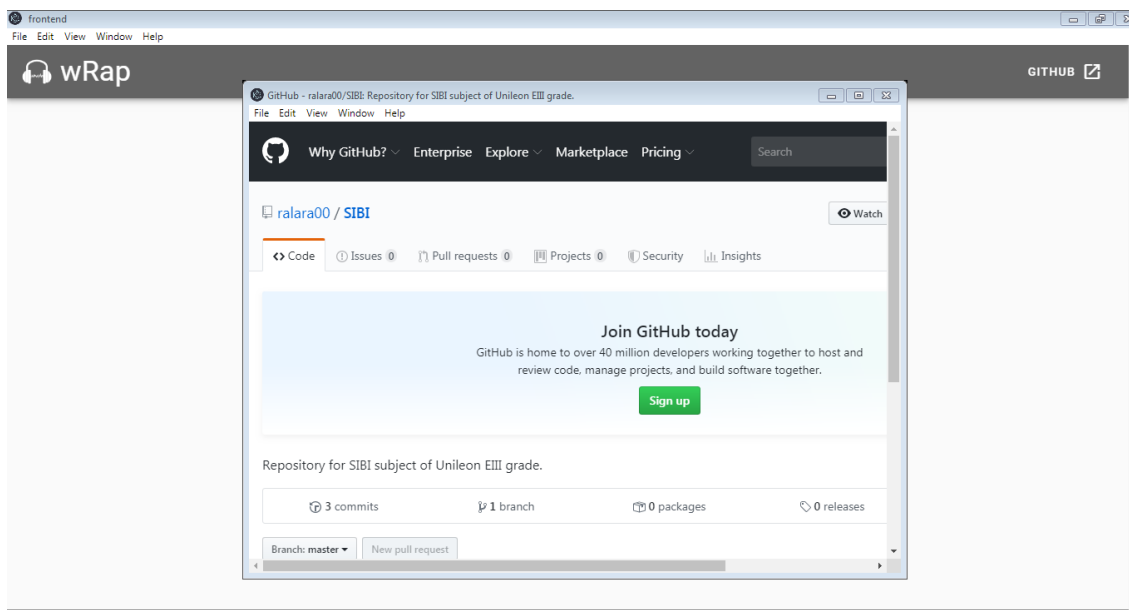


Ilustración 5 - captura de la interfaz de wRap

El logo del medio y los botones también son sencillos, corresponden a:

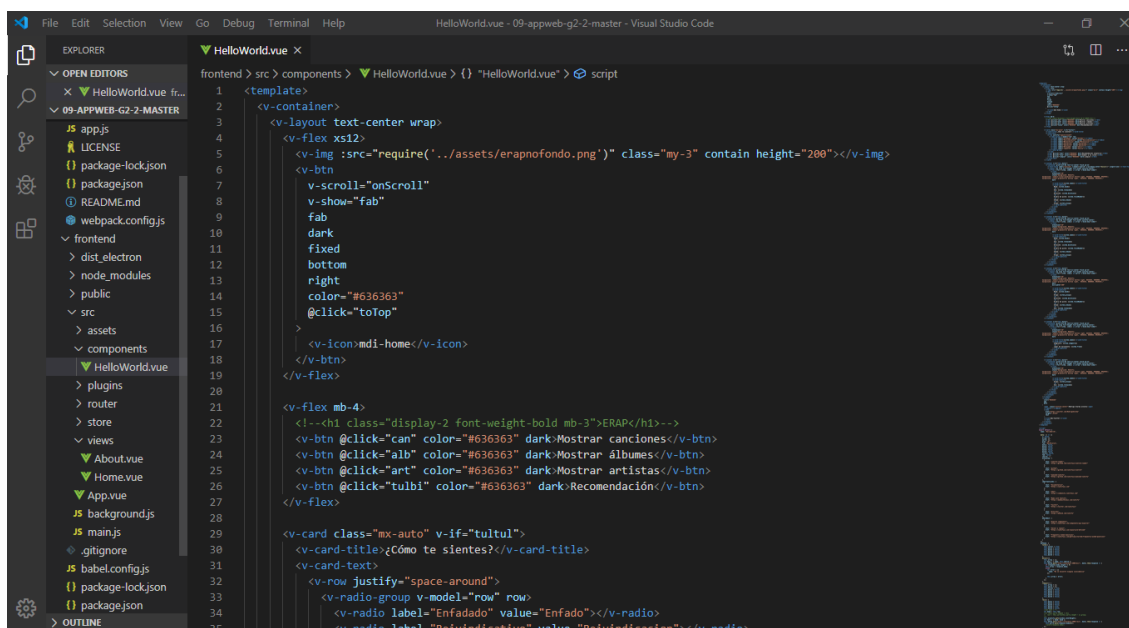


Ilustración 6 - captura del desarrollo de wRap

Como se puede observar, la paleta de colores de wRap se basa en una escala de grises, donde predomina siempre el #636363, en hexadecimal. En esa captura, también



podemos observar el botón “toTop”, el cual permite volver en cualquier momento a la parte superior de la pantalla, y es el siguiente:

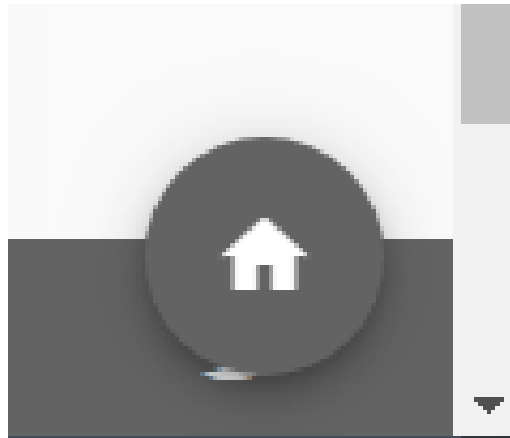


Ilustración 7 - captura de la interfaz de wRap

La función correspondiente a este botón es muy simple también, pues solo lo envía a la posición 0 de la página:

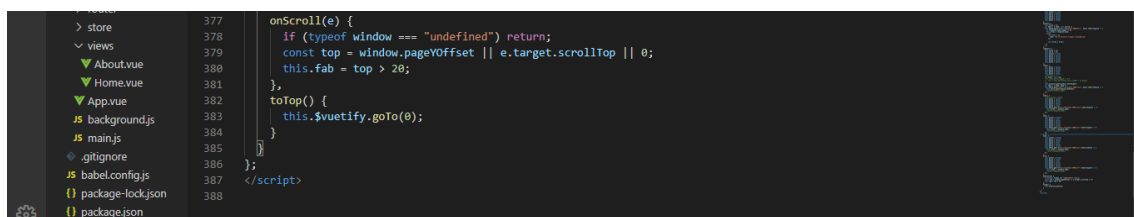


Ilustración 8 - captura del desarrollo de wRap

Además podemos observar onScroll que es la función que permite que aparezca dicho botón cuando detecta que bajamos el “scroll” en la página.

En la interfaz también podemos observar un pie de página, con el nombre del desarrollador así como su Twitter personal:



Ilustración 9 - captura de la interfaz de wRap

Esto es algo tan simple de hacer como:



Ilustración 10 - captura del desarrollo de wRap

Por último, falta por explicar los botones así como su funcionamiento.

Botón “Mostrar canciones”:

Este botón muestra todas las canciones de la base de datos, además de un buscador para que sea más fácil comprobar si existe aquella en la que el usuario esté pensando.

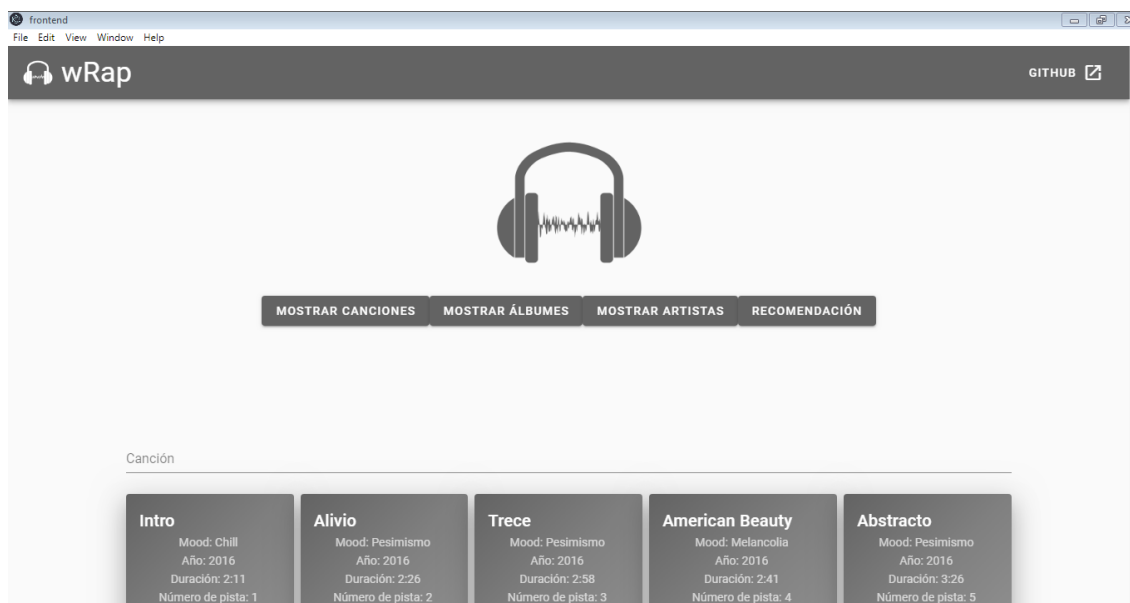


Ilustración 11 - captura de la interfaz de wRap

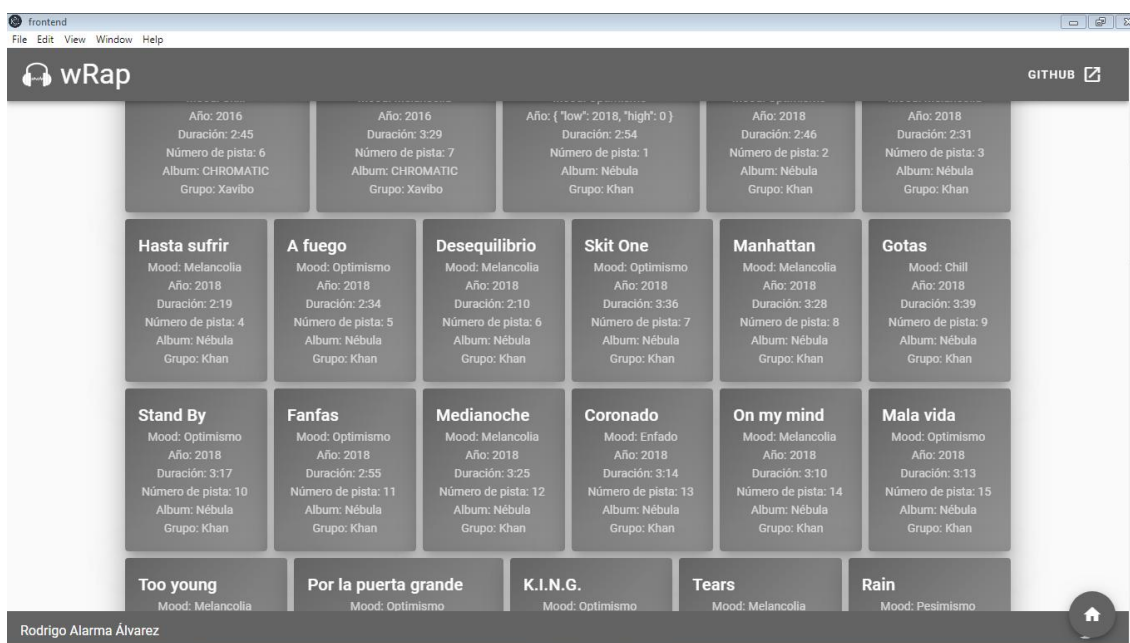


Ilustración 12 - captura de la interfaz de wRap

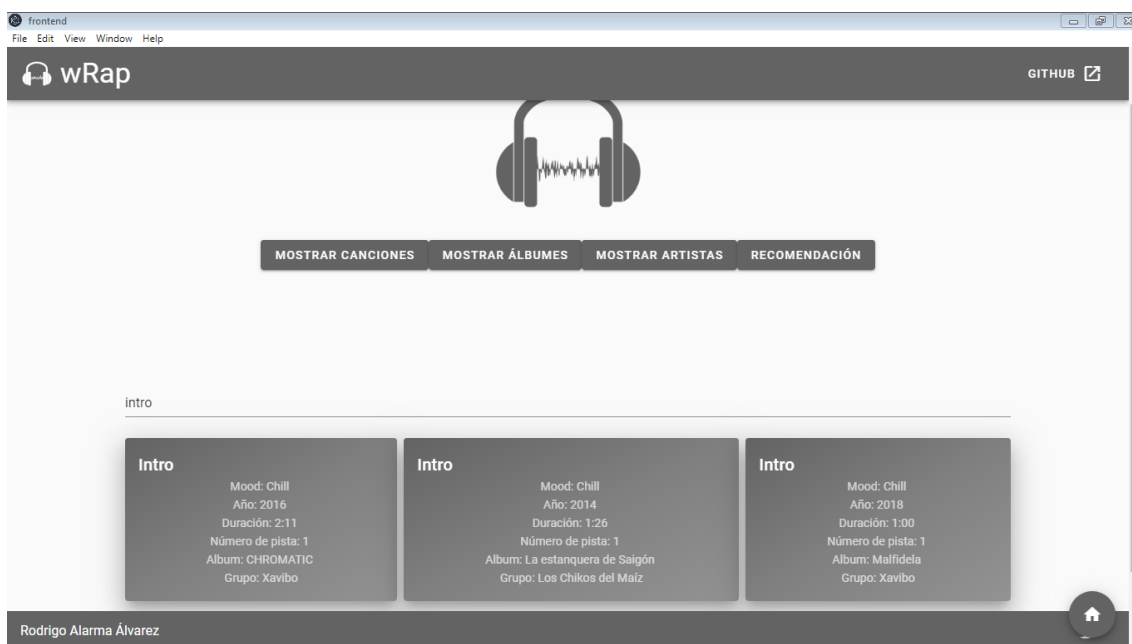


Ilustración 13 - captura de la interfaz de wRap

Para realizar esto, sencillamente hicimos enviamos al Backend la función:

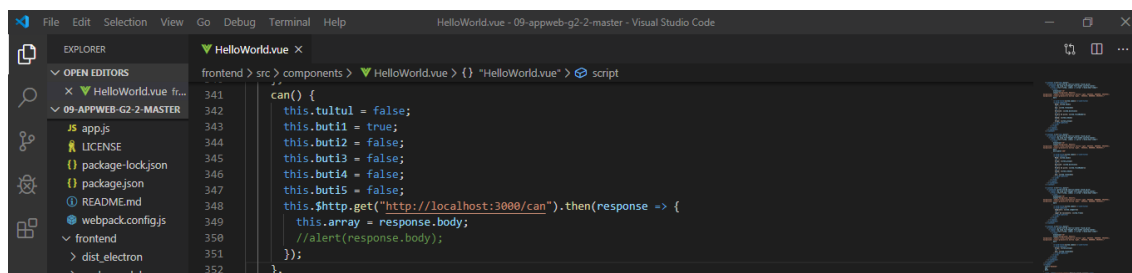


Ilustración 14 - captura del desarrollo de wRap

Y ésta realiza una consulta con todas las canciones a la base de datos:



Ilustración 15 - captura del desarrollo de wRap



En esta captura se observa como la consulta realizada es “Match (n:Song) return n”, puesto que lo que interesa es recoger todas las canciones de la base de datos.

El buscador en cambio recoge el texto introducido en el campo de la interfaz:

```
<v-content v-if="this.buti1">
  <v-container grid-list-md text-xs-center fluid pa-12>
    <v-text-field label="Canción" v-model="search" v-on:keyup.enter="buscar()" single-line></v-text-field>
    <v-layout row wrap fill-height fill-width justify-center>
      <v-for v-for="(item, index) in array" v-bind:key="index">
        <v-card
          elevation="18"
          style="background: #3A1C71;
            background: -webkit-linear-gradient(to bottom right, #636363, #808080, #939393);
            background: linear-gradient(to bottom right, #636363, #808080, #939393);"
          dark>
          <v-card-title>{{item.name}}</v-card-title>
          <v-card-subtitle>
            Mood: {{item.mood}}
          <br />
            Año: {{item.released}}
          <br />
            Duración: {{item.duration}}
        </v-card>
      </v-for>
    </v-layout>
  </v-container>
</v-content>
```

Ilustración 16 - captura del desarrollo de wRap

Se lo envía al Backend:

```
},
buscar(){
  this.array = [];
  var data = { name: this.search };
  this.$http.post("http://localhost:3000/bscr", data).then(response => {
    if (response.body.length != 0)
      this.array = response.body;
    else{
      var error = [{
        name: "No se encontró ninguna coincidencia"
      }];
      this.array = error;
    }
  });
},
```

Ilustración 17 - captura del desarrollo de wRap

Y lo compara en la base de datos, sin tener en cuenta en qué posición se encuentre del título, y si tiene mayúsculas, minúsculas, tildes, etc.:

```
app.post("/bscr", function (req, res) {
  var song1 = req.body.name;
  console.log(song1);
  var query = "MATCH (n:Song) where n.name =~ '(?i).*" + song1 + ".*' return n";
  console.log(query);
  var array = [];
  const resultPromise = session.run(query).subscribe({
    onNext: function (record) {
      array.push(record.get(0).properties);
    },
    onCompleted: function () {
      console.log(array);
      res.send(array);
      session.close();
    },
    onError: function (error) {
      console.log(error);
    }
  });
});
app.get("/real", function (req, res) {
  var query = "MATCH (n:Song) return n";
```

Ilustración 18 - captura del desarrollo de wRap



Botón “Mostrar álbumes”:

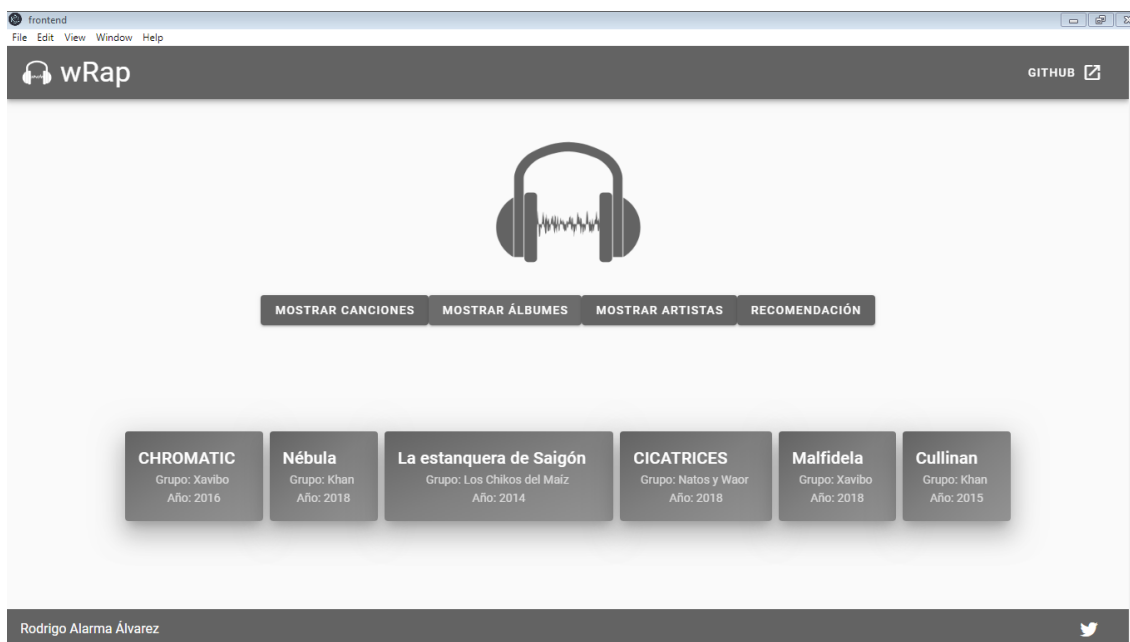


Ilustración 19 - captura de la interfaz de wRap

Este botón sencillamente muestra en la interfaz los álbumes de la base de datos:

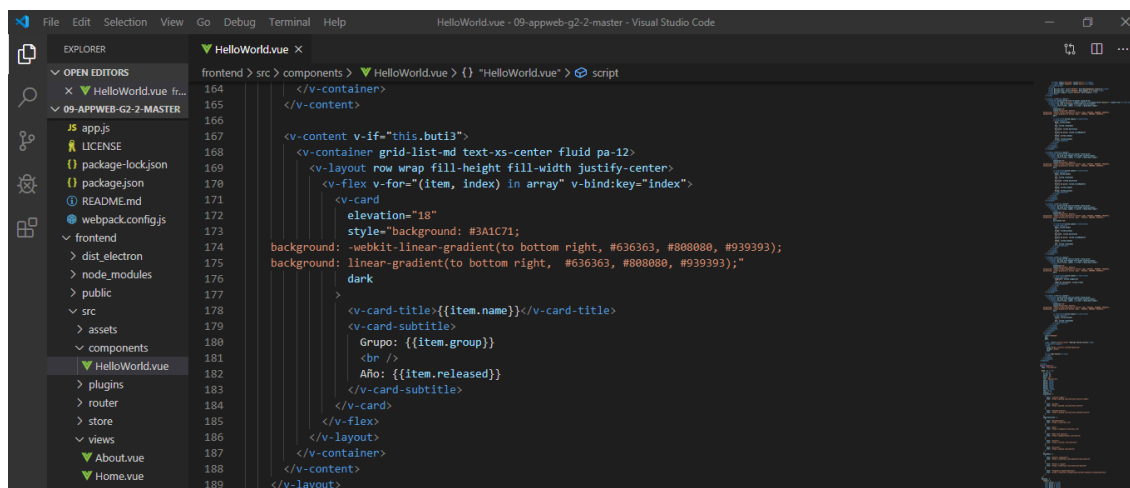


Ilustración 20 - captura del desarrollo de wRap

Para ello, realiza una consulta pidiendo los álbumes, y los devuelve en el array igual que en el caso de las canciones:



```
109 )),
110 app.get("/alb", function (req, res) {
111   var query = "MATCH (n:Album) return n";
112   var array = [];
113   const resultPromise = session.run(query).subscribe({
114     onNext: function (record) {
115       array.push(record.get(0).properties);
116     },
117   },
118   onCompleted: function () {
119     console.log(array);
120     res.send(array);
121     session.close();
122   },
123   onError: function (error) {
124     console.log(error);
125   }
126   });
127 },
128 app.get("/art", function (req, res) {
```

Ilustración 21 - captura del desarrollo de wRap

Botón “Mostrar artistas”:

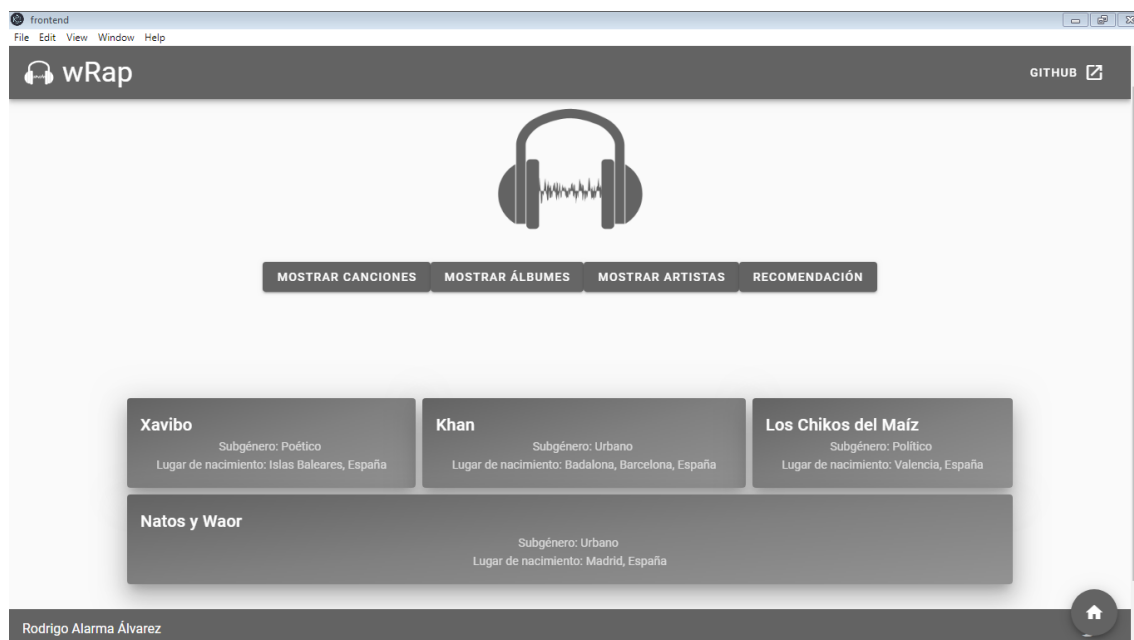


Ilustración 22 - captura de la interfaz de wRap

Este botón se encarga de mostrar en la interfaz todos los artistas y grupos de la base de datos:



```

129  },
130  app.get("/art", function (req, res) {
131    var query = "match (n where {-Genre}-[]->(n) return n";
132    var array = [];
133    const resultPromise = session.run(query).subscribe({
134      onNext: function (record) {
135        array.push(record.get(0).properties);
136      },
137      onCompleted: function () {
138        console.log(array);
139      },
140      res.send(array);
141    },
142    session.close();
143  },
144  onError: function (error) {
145    console.log(error);
146  }
147  });
148  });
149  });
150  });

```

Ilustración 24 - captura del desarrollo de wRap

Sistemas de Información de Gestión y Business Intelligence

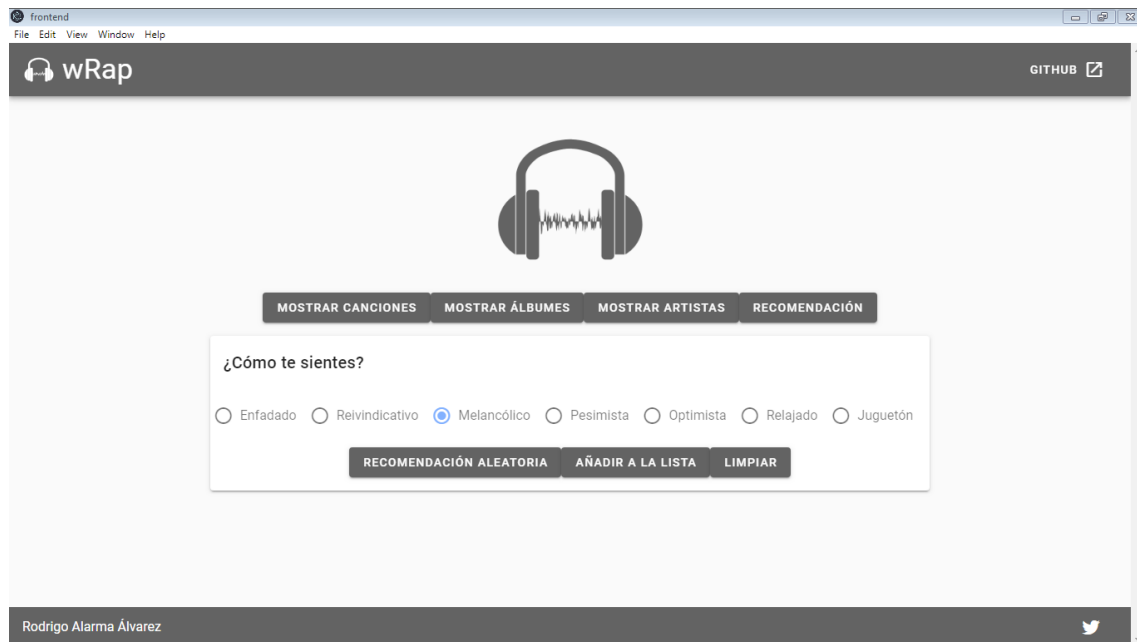


Ilustración 25 - captura de la interfaz de wRap

Este botón solo te despliega una serie de opciones que son los “checkbox” para seleccionar tu estado de ánimo, el botón de “Recomendación aleatoria”, el botón de “Añadir a la lista” y el botón de “Limpiar”.

Botón de “Recomendación aleatoria”:

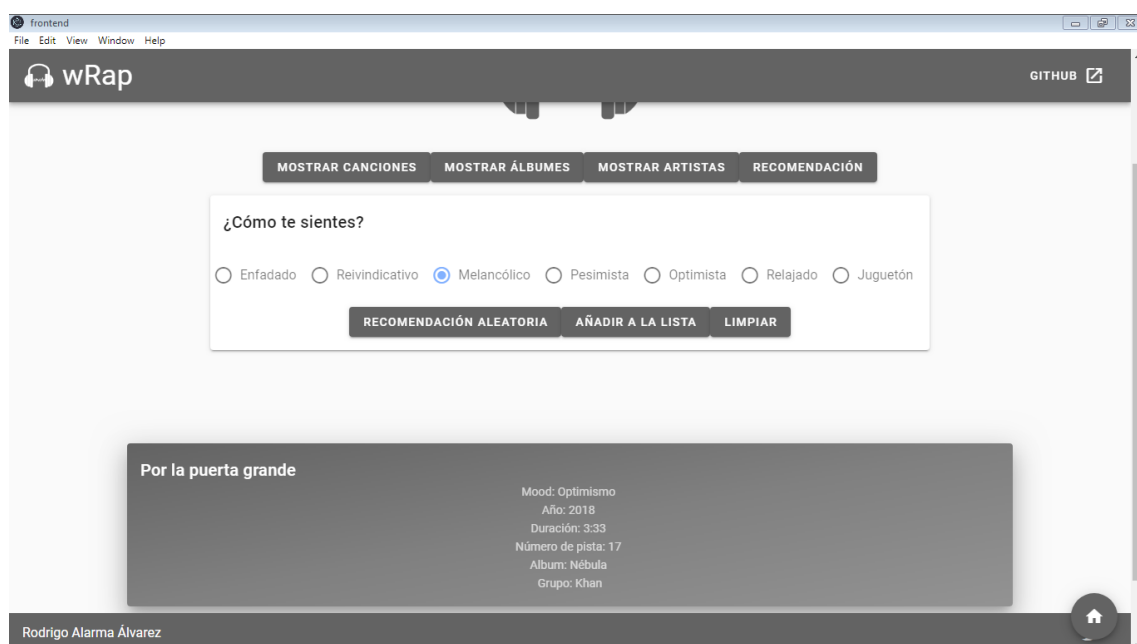


Ilustración 26 - captura de la interfaz de wRap



Este botón únicamente muestra por pantalla una canción aleatoria del array de canciones de la base de datos:

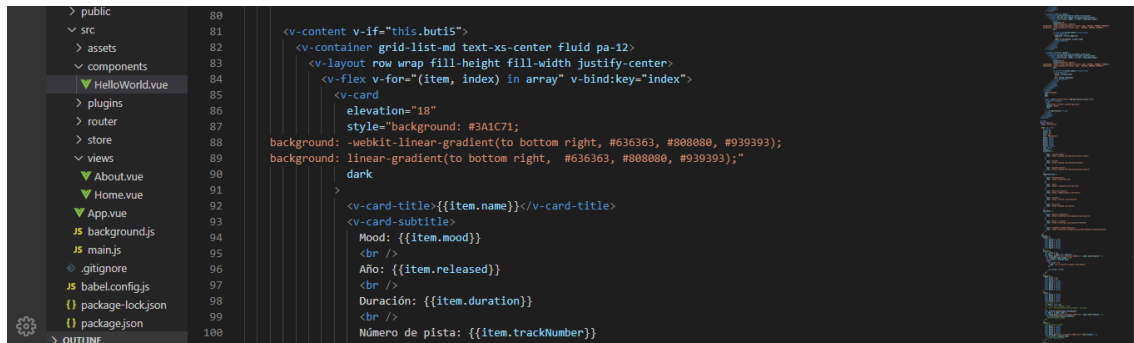


Ilustración 27 - captura del desarrollo de wRap

Aquí observamos que la consulta es la misma que para obtener las canciones, pero en este caso devolvemos una posición aleatoria del array, es decir, una canción aleatoria:



Ilustración 28 - captura del desarrollo de wRap

Botón “Añadir a la lista”:

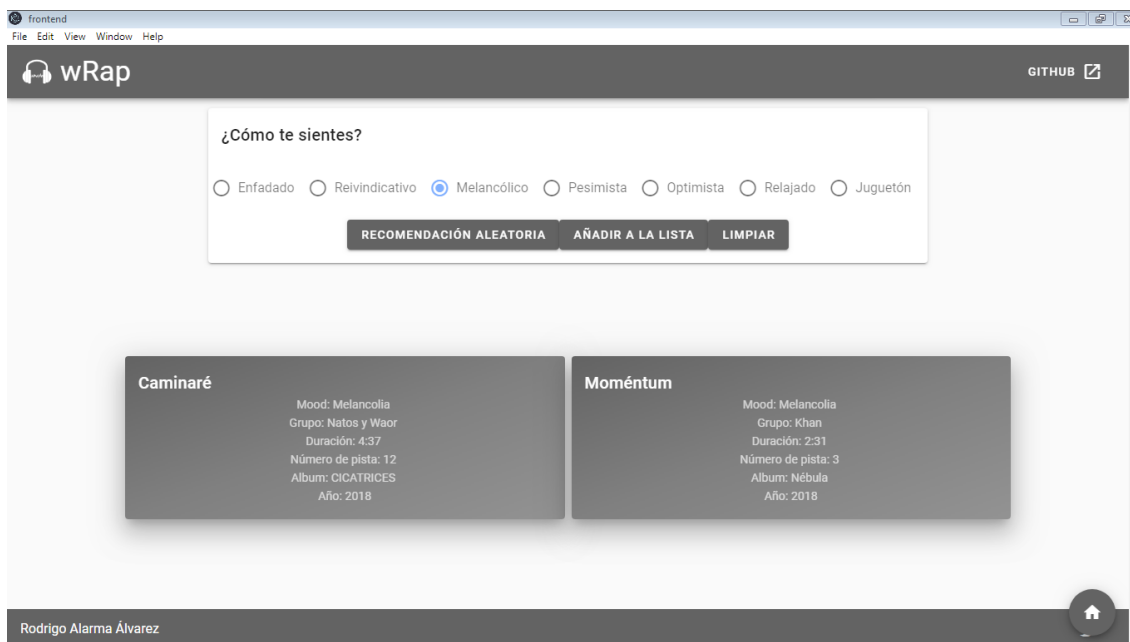


Ilustración 29 - captura de la interfaz de wRap

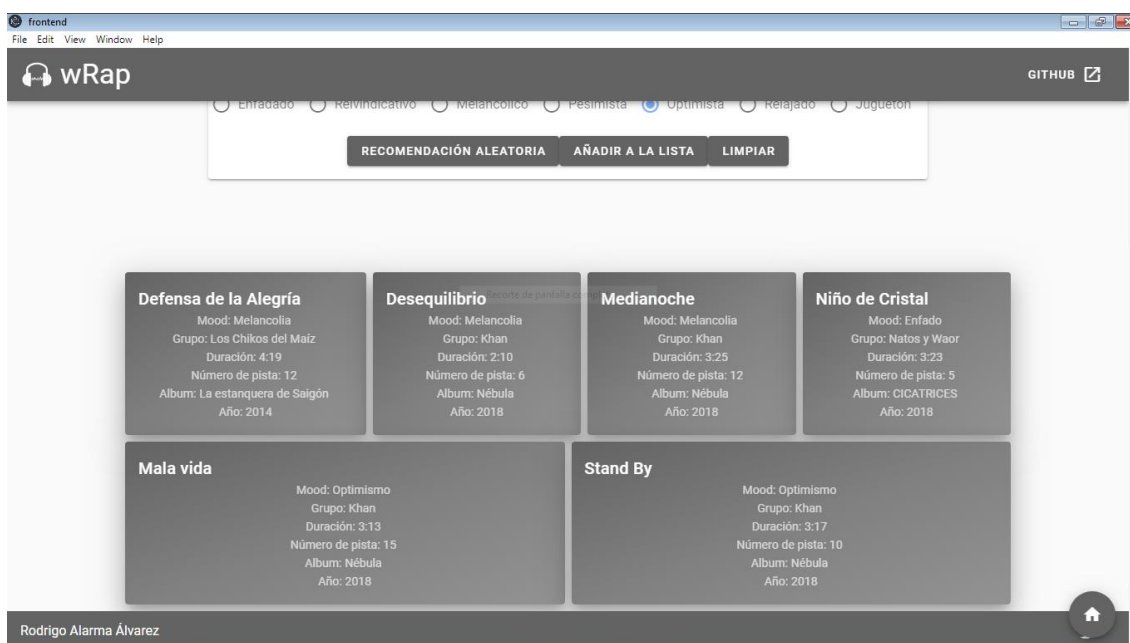


Ilustración 30 - captura de la interfaz de wRap

Este botón añade a un array, una recomendación aleatoria de una canción de un estado de ánimo concreto, y continúa mostrando una canción nueva cada vez del estado de ánimo seleccionado más las anteriores, hasta que el usuario le dé al botón de “Limpiar”.



En esta captura observamos como se llaman tanto la función de recomendación personalizada “reco” como la función de recomendación aleatoria “real”, siendo la primera un post puesto que el usuario manda información al Backend, y siendo la segunda un get puesto que solo debe recibir información de éste.

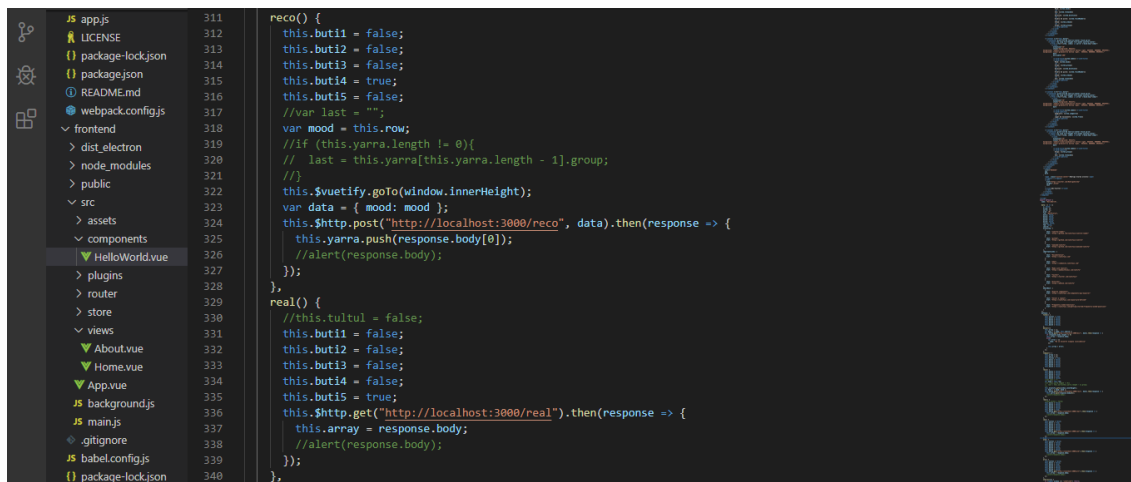


Ilustración 31 - captura del desarrollo de wRap

La forma en la cual muestra la información por pantalla es igual que en el caso de la captura de la recomendación aleatoria, pero no lo es la forma en la que la trata en el Backend:

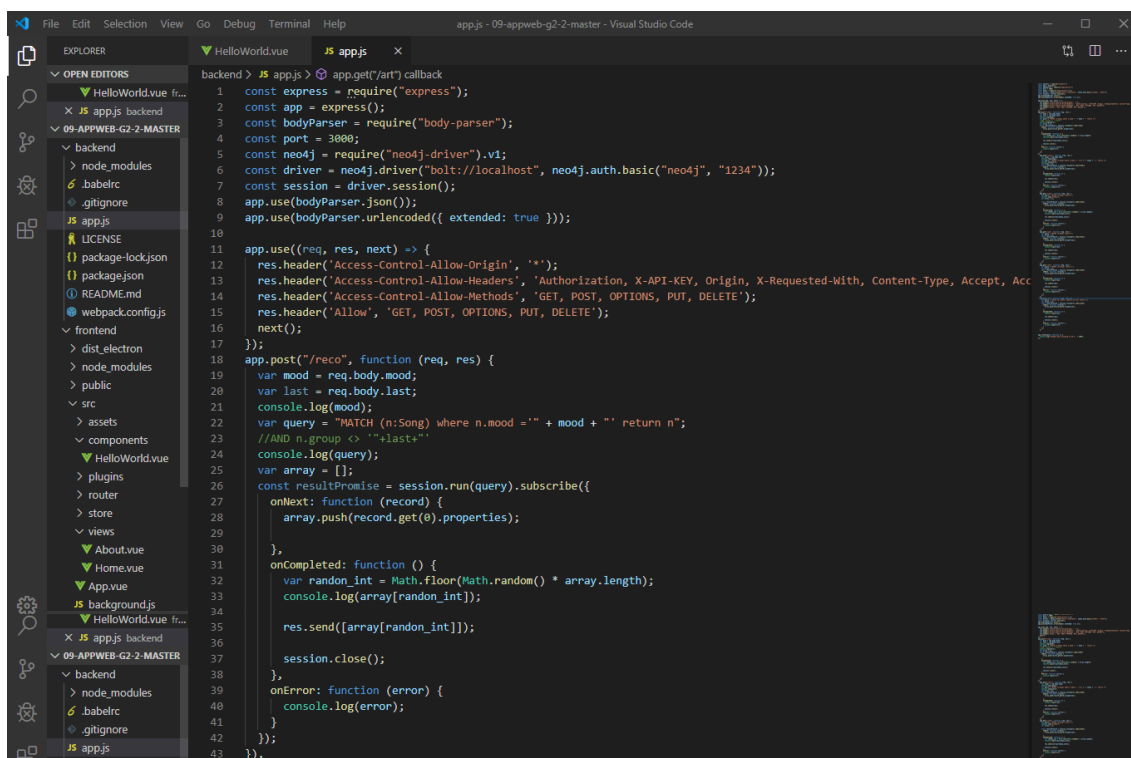


Ilustración 32 - captura del desarrollo de wRap



Aquí observamos como recoge la información seleccionada por el usuario con el mood, hace la consulta a Neo4j, que busca que devuelva las canciones cuyos estados de ánimo coincidan con el estado de ánimo seleccionado por el usuario.

En onNext las va añadiendo al array, y en onCompleted genera un número aleatorio con el tamaño del array, que será la canción escogida para recomendársela al usuario.

Botón “Limpiar”:

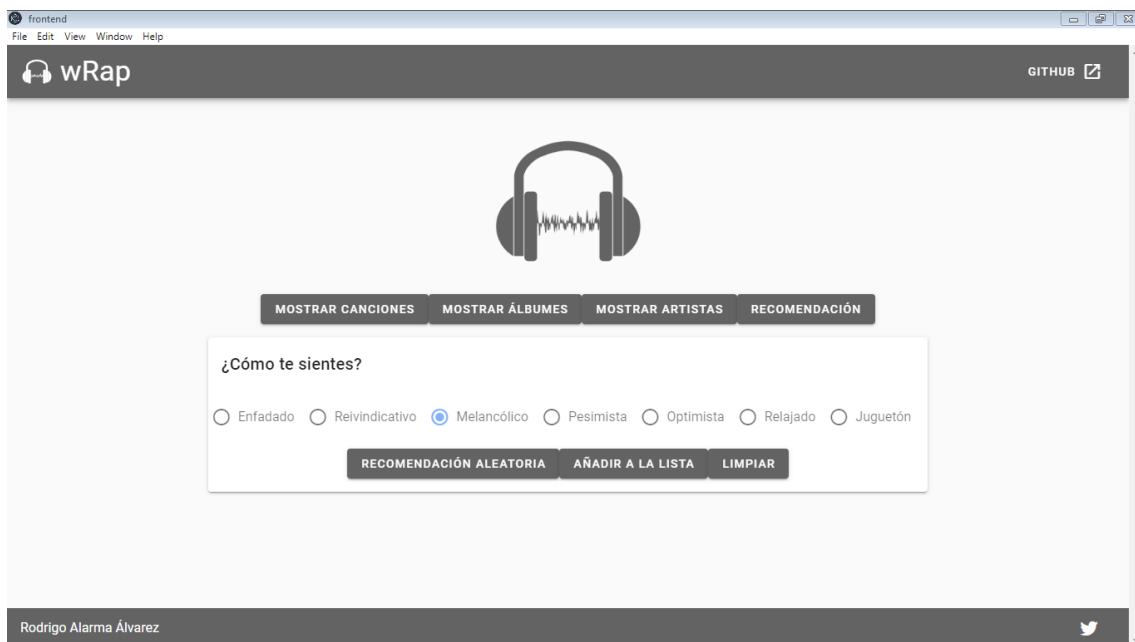


Ilustración 33 - captura de la interfaz de wRap

Este botón vacía la lista que se estaba creando.

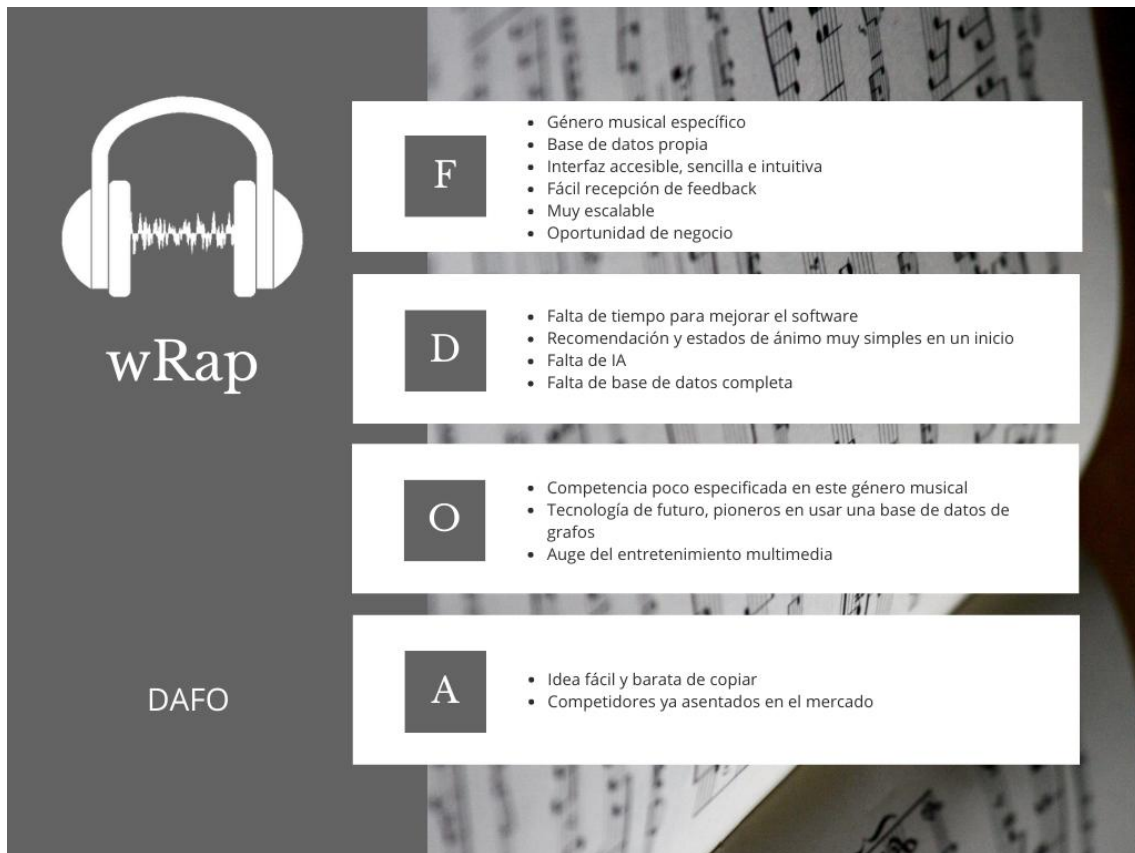


```
methods: {
  tulbi() {
    this.tultul = true;
    this.but11 = false;
    this.but12 = false;
    this.but13 = false;
    this.but14 = false;
    this.but15 = false;
  },
  buscar(){
    this.array = [];
    var data = { name: this.search };
    this.$http.post("http://localhost:3000/bscr", data).then(response => {
      if (response.body.length != 0)
        this.array = response.body;
      else{
        var error = [{
          name: "No se encontró ninguna coincidencia"
        }];
        this.array = error;
      }
    });
  },
  limpiar() {
    this.array = [];
    this.yarra = [];
    this.tultul = true;
    this.but11 = false;
    this.but12 = false;
    this.but13 = false;
    this.but14 = false;
    this.but15 = false;
  },
}
```

Ilustración 34 - captura del desarrollo de wRap



5.- DAFO



Desarrollo del DAFO:

Fortalezas:

- Género musical específico: Aplicación que se basa únicamente en el rap, logrando una especificación mucho mayor que si se disgregase en varios géneros musicales.
- Base de datos propia: Base de datos de grafos desarrollada íntegramente, con el acceso únicamente del desarrollador y alojada por él.
- Interfaz accesible, sencilla e intuitiva: Para que las personas accedan rápido a su objetivo, aunque no suelen usar internet o tengan problemas de vista o discapacidades.
- Fácil recepción de Feedback: GitHub y Twitter del desarrollador en la propia página web, para que el contacto sea sencillo.
- Muy escalable: El producto tiene la capacidad de evolucionar mucho.
- Oportunidad de negocio: No existe un producto igual.

**Oportunidades:**

- Competencia poco especificada en este género musical: La competencia no es específica de este género musical y, por tanto, sus algoritmos no tienen en cuenta las mismas cosas y contienen muchos menos artistas de rap.
- Tecnología de futuro, pioneros en usar una base de datos de grafos: Se accede a la información más rápido y de forma más sencilla cuando existen muchos nodos.
- Auge del entretenimiento multimedia: Actualmente el entretenimiento multimedia como el que ofrece Netflix o HBO está de moda, lo que nos afecta positivamente para entrar en el mercado.

Transformación de debilidades en fortalezas:

- Falta de tiempo para mejorar el software: En dos meses da tiempo a aprender una tecnología nueva como Neo4j, y poco más, por ello con más tiempo se podrá añadir contenido multimedia a la base de datos y mejorar el algoritmo a uno de inteligencia artificial.
- Recomendación y estados de ánimo muy simples en un inicio: El algoritmo realiza una recomendación muy simple, y hay pocos estados de ánimo y con una selección muy pobre de éstos, lo cual se puede mejorar con un algoritmo de inteligencia artificial, que aprenda los estados de ánimo del usuario en vez de que éste los seleccione en la interfaz.
- Falta de IA: Con más tiempo para desarrollar el proyecto, se puede añadir la inteligencia artificial en el Backend, manteniendo el Frontend actual.
- Falta de base de datos completa: Con más tiempo para desarrollar el proyecto, se puede añadir el contenido multimedia, así como más información de los artistas en la base de datos.

Transformación de amenazas en oportunidades:

- Idea fácil y barata de copiar: Una vez entremos en el mercado, quizá los competidores quieran copiarnos, pero no podrán hacerlo si ya nos hemos diferenciado y encontrado un público fiel.
- Competidores ya asentados en el mercado: Los principales competidores ya tienen un público fijo y están asentados en el mercado, pero nosotros ofrecemos algo más específico que ellos, lo que nos permite ocupar su lugar para muchos usuarios si realizamos la campaña de marketing adecuada.



6.- Líneas de futuro

Como hemos mencionado anteriormente, quizá la mayor ventaja de wRap es su escalabilidad. Esto quiere decir que el propio producto puede llegar a ser mucho más grande de lo que es actualmente, creando un ecosistema del cual será el centro. Para llegar a esto, se han de seguir ciertos pasos, que son:

- Lo primero y más importante es ampliar el tamaño de la base de datos, para englobar a todos los artistas importantes del rap y de los géneros más cercanos a él, así como contrastar y aumentar la información que existe sobre ellos.
- Se debe incluir una previsualización de cada una de las canciones de la base de datos, ya sea a través de un vídeo de YouTube, de un plug-in de Spotify o de SoundCloud, o de otra forma.
- Con la base de datos totalmente completa ya, es necesario que los usuarios tengan la capacidad de crearse un perfil de usuario en la página web, en el cual guardar sus preferencias, y recibir información de sus artistas favoritos, así como información comercial. También se podrá conectar con Facebook o Google para que dicho perfil sea sincronizado y más completo.
- Gracias a los perfiles de usuario, y a las cookies, es importante ser capaz de recopilar la información de cada uno de estos usuarios en nuestra página, para poder ver qué artistas importantes gustan más, qué artistas desconocidos gustan más, qué le gusta a cada usuario, y por tanto hacer recomendaciones personalizadas más efectivas.
- Del mismo modo que en el punto anterior, también es importante recolectar la información que las cookies nos puedan dar de los usuarios fuera de la página, mejorando por tanto las recomendaciones personalizadas que se realicen.
- Una vez tanto la información del usuario como la de la base de datos es amplia, se debe empezar a recomendar mediante un algoritmo de inteligencia artificial, pues con la cantidad de datos que se manejarán en este punto es la mejor opción para maximizar la personalización.
- Cuando solo haya que mejorar cada vez más el algoritmo, lo posterior será vender sectores publicitarios en la página web, para diversas empresas que crean que su público objetivo usa nuestro servicio.
- Para el usuario, se debe vender un servicio tanto gratuito con publicidad, como una versión Premium, sin publicidad y con más funciones.
- También se deberá vender al artista, protagonismo en las recomendaciones, así como publicidad del estilo de la que vende Facebook e Instagram, es decir, que bajo unos parámetros de gustos, localización, sexo, edad, etc., el artista aparezca más frecuentemente en las recomendaciones básicas, así como en las personalizadas.

**Comercialización:**

Como se especificó anteriormente, wRap es gratuita si aceptas la publicidad, o con un coste de 1,99 euros al mes para evitar esta publicidad.

En el futuro, cuando se pueda consumir multimedia desde wRap, la tarifa aumentará a 4,95 euros la tarifa normal, pero será gratuita con publicidad, como antes.

Para el artista, siempre será gratuito su almacenaje en la base de datos, pero el posicionamiento en las recomendaciones, así como la publicidad mediante segmentación demográfica tendrá un coste aún por determinar en función de estudios de mercado.

En todos los casos se ha de cumplir con la ley de protección de datos en caso de que el cliente acceda a compartir sus datos con wRap.

Se espera que la evolución total del producto del estado actual al esperado ocurra en aproximadamente entre 1 y 3 años.

Marketing:

Para los objetivos del proyecto se deben seguir los siguientes tipos de marketing:

Marketing en redes sociales y de influencia.

Nuestro público objetivo es joven, por lo que las redes sociales es el lugar más importante al que debemos acudir, especialmente Instagram. Además, sus ídolos del mundo del rap y de fuera deben recomendar wRap, o al menos mostrar que la usan y que les gusta, eso nos dará una gran ventaja competitiva.

Inbound Marketing.

El objetivo principal y más importante que tenemos es fidelizar a nuestros usuarios, hacer que crean en nuestro producto tanto o más que nosotros. Es decir, conseguir que todo aquel que use wRap al menos ocasionalmente, no solo esté contento, sino que encima haga de portavoz de ésta, vendiéndola a sus amigos y compañeros.



7.- Lecciones aprendidas

A lo largo de la asignatura, así como de la realización de este proyecto, he aprendido una serie de lecciones, así como mejorado una serie de competencias:

He aprendido a crear y manejar bases de datos de grafos de Neo4j.

He aprendido a usar el lenguaje Cypher.

He mejorado mis capacidades de programación de páginas web.

He mejorado mi capacidad de realizar interfaces web de usuario.

He mejorado mi capacidad de pedir ayuda a mis compañeros.

He mejorado mi capacidad de ayudar a mis compañeros.

He mejorado mi capacidad de compartir conocimiento, así como de adquirirlo de diversas fuentes de internet y filtrarlo.

8.- Conclusión

He logrado realizar aquello que me propuse una vez vislumbré el camino que quería seguir en esta asignatura. Espero con ansia la llegada del verano, para poder continuar con este proyecto por mi cuenta, rellenar la base de datos, probar la reproducción multimedia, y aprender a incorporar alguno de los algoritmos de inteligencia artificial que valoré al comenzar con el proyecto.

Gracias a este proyecto, he podido observar de primera mano otra forma de valorar los datos, las bases de datos de conocimiento. Es un campo que nunca conocí o toqué, y que ahora me parece mucho más interesante incluso de lo que pensé en un inicio, haciendo que valore mucho más por tanto todo lo que he aprendido en esta asignatura.



9.- Bibliografía y referencias

Bibliografía:

What Is a Graph Database and Property Graph | Neo4j. Retrieved 18 December 2019, from <https://neo4j.com/developer/graph-database/>

JavaScript. Retrieved 18 December 2019, from <https://es.wikipedia.org/wiki/JavaScript>

Introduction — Vue.js. Retrieved 18 December 2019, from <https://vuejs.org/v2/guide/>

Vue Material Design Component Framework — Vuetify.js. Retrieved 18 December 2019, from <https://vuetifyjs.com/es-MX/>

Carracedo, S. (2017). Usando Vue.js + electron.js para crear aplicaciones de escritorio. Retrieved 18 December 2019, from <https://www.opsou.com/es/blog/usando-vuejs-electronjs-para-crear-aplicaciones-de-escritorio>

Python. Retrieved 18 December 2019, from <https://es.wikipedia.org/wiki/Python>

Flask. Retrieved 18 December 2019, from <https://es.wikipedia.org/wiki/Flask>

Small, N. (2019). The Py2neo v4 Handbook — The Py2neo v4 Handbook. Retrieved 18 December 2019, from <https://py2neo.org/v4/>

Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación? - SmartHint. Retrieved 18 December 2019, from <https://www.smarthint.co/es/como-funciona-la-inteligencia-artificial-en-los-sistemas-de-recomendacion/>

Ilustración 1 - Schiavini, R. (2019). ¿Cómo funciona la inteligencia artificial en los sistemas de recomendación? - SmartHint. Retrieved 18 December 2019, from <https://www.smarthint.co/es/como-funciona-la-inteligencia-artificial-en-los-sistemas-de-recomendacion/>

Ilustración 2 – Base de datos de wRap

Ilustración 3 - captura de la interfaz de wRap

Ilustración 4 - captura del desarrollo de wRap

Ilustración 5 - captura de la interfaz de wRap

Ilustración 6 - captura del desarrollo de wRap

Ilustración 7 - captura de la interfaz de wRap



Ilustración 8 - captura del desarrollo de wRap

Ilustración 9 - captura de la interfaz de wRap

Ilustración 10 - captura del desarrollo de wRap

Ilustración 11 - captura de la interfaz de wRap

Ilustración 12 - captura de la interfaz de wRap

Ilustración 13 - captura de la interfaz de wRap

Ilustración 14 - captura del desarrollo de wRap

Ilustración 15 - captura del desarrollo de wRap

Ilustración 16 - captura del desarrollo de wRap

Ilustración 17 - captura del desarrollo de wRap

Ilustración 18 - captura del desarrollo de wRap

Ilustración 19 - captura de la interfaz de wRap

Ilustración 20 - captura del desarrollo de wRap

Ilustración 21 - captura del desarrollo de wRap

Ilustración 22 - captura de la interfaz de wRap

Ilustración 23 - captura del desarrollo de wRap

Ilustración 24 - captura del desarrollo de wRap

Ilustración 25 - captura de la interfaz de wRap

Ilustración 26 - captura de la interfaz de wRap

Ilustración 27 - captura del desarrollo de wRap

Ilustración 28 - captura del desarrollo de wRap

Ilustración 29 - captura de la interfaz de wRap

Ilustración 30 - captura de la interfaz de wRap

Ilustración 31 - captura del desarrollo de wRap

Ilustración 32 - captura del desarrollo de wRap

Ilustración 33 - captura de la interfaz de wRap



Ilustración 34 - captura del desarrollo de wRap

Referencias

GitHub personal: Alarma Álvarez, R. (2019). ralara00/SIBI. Retrieved 18 December 2019, from <https://github.com/ralara00/SIBI>

Pixabay: Más de 1 millón de imágenes gratis para descargar - Pixabay. Retrieved 18 December 2019, from <https://pixabay.com/es/>

Canva: Retrieved 18 December 2019, from <https://www.canva.com/>

GIMP: GIMP Descargas, tutoriales y foros. Alternativa a Photoshop gratis y libre. Retrieved 18 December 2019, from <http://www.gimp.org.es/>

Neo4j: Neo4j Graph Platform – The Leader in Graph Databases. Retrieved 18 December 2019, from <https://neo4j.com/>

JS: Free JavaScript training, resources and examples for the community. Retrieved 18 December 2019, from <https://www.javascript.com/>

Python: Welcome to Python.org. Retrieved 18 December 2019, from <https://www.python.org/>

Flask: Flask. Retrieved 18 December 2019, from <https://www.palletsprojects.com/p/flask/>

Py2Neo: Python Language - Neo4j y Cypher usando Py2Neo | python Tutorial. Retrieved 18 December 2019, from <https://riptutorial.com/es/python/topic/5841/neo4j-y-cypher-usando-py2neo>

Lecturas de Neo4j:

Brath, R., & Jonker, D. (2015). Graph analysis and visualiztion. Indianapolis: Wiley.

Pan, J., Vetere, G., Gomez- Perez, J., & Wu, H. Exploiting Linked Data and Knowledge Graphs in Large Organisations.

Needham, M., & Hodler, A. Graph algorithms.

Robinson, I., Webber, J., & Eifrem, E. (2015). Graph Databases. O'Reilly Media.

Erciyas, K. (2018). Guide to graph algorithms. Cham: Springer.