

SpotifyNode

INFORME SOBRE LA APLICACIÓN

ALBERTO MACIAS GUTIERREZ
RODRIGO GARCÍA-CASILLAS BARROSO

Índice

| | |
|------------------------------------|---|
| Instalación y preparación | 2 |
| Aspectos técnicos | 3 |
| Navegación | 6 |
| Vista de cada artista o grupo..... | 8 |
| Página no encontrada | 9 |

Instalación y preparación

Para poder iniciar y ejecutar la aplicación hay que instalar primero los paquetes necesarios, para este en concreto, se ha utilizado *Express*, *HandleBars* y *Path*. Al ya tener el archivo `package.json` tal y como se muestra en la Ilustración 1 con las dependencias correspondientes, bastaría con ejecutar la orden “***npm install***” dentro de la carpeta del proyecto, gracias a esta orden se instalarían todas las dependencias correspondientes que se muestran.

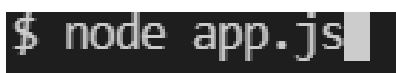


```
1 {
2   "name": "spotifynode",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "nodemon app.js",
8     "test": "echo \\\"Error: no test specified\\\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "express": "^4.17.3",
15    "hbs": "^4.2.0",
16    "path": "^0.12.7",
17    "spotify-web-api-node": "^5.0.2"
18  }
19 }
```

Ilustración 1. Archivo `package.json`

Así mismo, también se instalará el paquete que nos permite utilizar la API de Spotify en Node.

Por último, para poder hacer uso de la aplicación y que se muestre en nuestro navegador, se debe lanzar la aplicación haciendo uso del comando “***node app.js***” dentro de la carpeta del proyecto, tal y como se observa en la



```
$ node app.js
```

Ilustración 2. Comando para la ejecución de la aplicación

Aspectos técnicos

Para usar la API de Spotify fue necesario obtener un token que nos permite extraer y usar los datos que proporciona, para ello, se creó la función que se aprecia en la Ilustración 3. con el fin de obtener un token cada vez que se lanzara la aplicación, evitando así estar pendiente de obtener un token nuevo pasado el tiempo de expiración de cada token, correspondiente a 3600 segundos (1 hora).

```
23
24 //-----GENERAR TOKEN-----//
25 // Retrieve an access token.
26 spotifyApi.clientCredentialsGrant().then(
27   function (data) {
28     console.log("The access token expires in " + data.body["expires_in"]);
29     console.log("The access token is " + data.body["access_token"]);
30
31     // Save the access token so that it's used in future calls
32     spotifyApi.setAccessToken(data.body["access_token"]);
33   },
34   function (err) {
35     console.log("Something went wrong when retrieving an access token", err);
36   }
37 );
38 //-----//
39
```

Ilustración 3. Función para obtener un token con cada ejecución

Por otro lado, uno de los requisitos que se pedían era obtener una lista de artistas aleatorios de la API, esta funcionalidad se ha conseguido gracias a la función que se puede ver en la Ilustración 4. Esta función realiza una búsqueda de un carácter que se obtiene de manera aleatoria gracias al *Math.random()*, que extrae uno de los caracteres de nuestra variable “characters”, que contiene el abecedario.

```
//-----BUSQUEDA ALEATORIA-----//
// https://perryjanssen.medium.com/getting-random-tracks-using-the-spotify-api-61889b0c0c27
function getRandomSearch() {
  // Una lista de todos los caracteres que se pueden elegir
  const characters = "abcdefghijklmnopqrstuvwxyz";

  // Obtiene un carácter aleatorio de la cadena de caracteres
  const randomCharacter = characters.charAt(
    Math.floor(Math.random() * characters.length)
  );
  let randomSearch = "";

  // Coloca el carácter comodín al principio, o al principio y al final, al azar
  switch (Math.round(Math.random())) {
    case 0:
      randomSearch = randomCharacter + "%";
      break;
    case 1:
      randomSearch = "%" + randomCharacter + "%";
      break;
  }

  return randomSearch;
}
//-----//
```

Ilustración 4. Función que permite obtener artistas aleatoriamente

Mencionar que en la vista en la que se muestra cada artista seleccionado es necesario la llamada de varias funciones tales como *getArtist()*, para obtener el propio artista, *getArtistTopTracks()*, para obtener sus canciones más populares y *getArtistAlbums()*, para obtener los últimos álbumes publicados por el artista. Todas ellas son llamadas cuando un artista es seleccionado por el usuario para que sea mostrado gracias a la función que se muestra en la Ilustración 5.

```
app.get("/artista/:artistaId", (req, res, next) => {
  let artistaSeleccionado = req.params.artistaId;
  console.log(req.params.artistaId);

  spotifyApi.getArtist(artistaSeleccionado)
    .then(function(data) {
      //console.log('Artist information', data.body);
      let artista = data.body;

      spotifyApi.getArtistTopTracks(artistaSeleccionado, 'ES')
        .then(function(data) {
          console.log(data.body);
          let topCanciones = data.body.tracks;

          spotifyApi.getArtistAlbums(artistaSeleccionado,{ limit: 6, offset: 6 })
            .then(function(data) {
              //console.log('Artist albums', data.body);
              let albums = data.body.items;
              res.render('artistaInfo', { artista, topCanciones, albums });
            }, function(err) {
              console.error(err);
            });

          }, function(err) {
            console.log('Something went wrong!', err);
          });

        }, function(err) {
          console.error(err);
        });
    });
});
```

Ilustración 5. Función para la obtención de los datos de un artista específico

Por último, se procede a explicar el funcionamiento de la función que controla si el usuario ha realizado un inicio de sesión correcto. Inicialmente se controla con la variable *“inicioSesionIncorrecto”* inicializada a *false*. Cuando el usuario introduce los datos correspondientes en el formulario de la vista *Login* y pulse en el botón “Iniciar Sesión”, nuestro código recibirá esa petición *POST* y comprobará los datos introducidos por el usuario, de ser correctos, el usuario habrá iniciado sesión correctamente. Para este correcto funcionamiento se ha implementado el código que se puede observar en la Ilustración 6.

```
114
115 app.get("/iniciarSesion", (req, res, next) => {
116   inicioSesionIncorrecto = false
117   res.render("iniciarSesion",{inicioSesionIncorrecto});
118 });
119
120 app.post("/iniciarSesion", (req, res, next) => {
121   let fEmail = req.body.falseEmail;
122   let fPass = req.body.falsePass;
123
124   if (fEmail === "pepe@pepe.com" && fPass === "1234") {
125     console.log("Ha iniciado sesion correctamente");
126     res.render("index");
127   } else {
128     console.log("ERROR: NO ha iniciado sesion correctamente");
129     inicioSesionIncorrecto = true;
130     res.render("iniciarSesion",{inicioSesionIncorrecto});
131   }
132 });
```

Ilustración 6. Control del inicio de sesión

Navegación

Una vez lanzada o iniciada la aplicación, se puede acceder a ella a través del navegador si se escribe “<http://localhost:3000/>” en el buscador. Aparece entonces la vista principal *Home* tal y como se muestra en la Ilustración 7. Esta vista tiene una imagen con el logo de Spotify y un cuadro con una breve descripción de la aplicación.

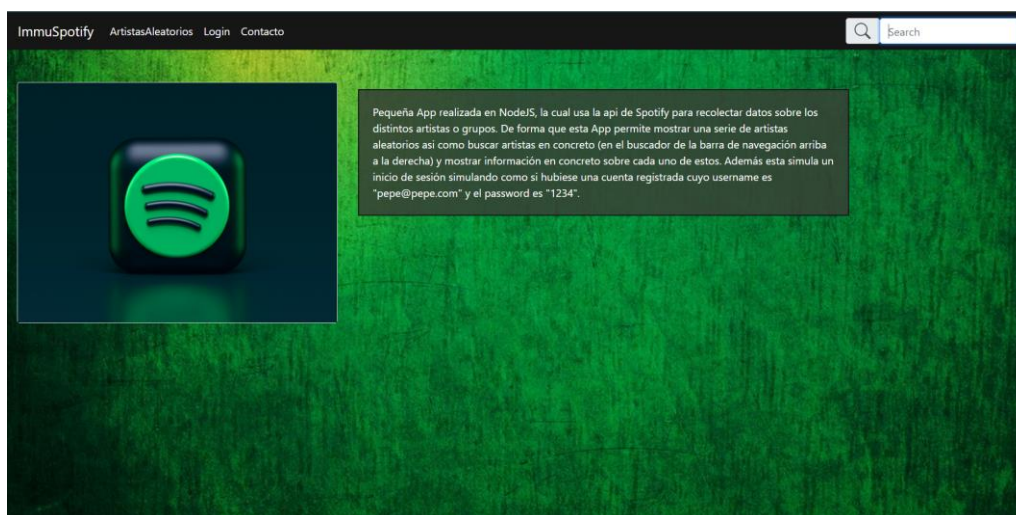


Ilustración 7. Vista Home

En la parte superior de nuestra página *Home* se observa una barra de navegación que permite desplazar al usuario de una vista a otra. Si se pulsa sobre *ImmuSpotify* la aplicación redirige al usuario a *Home*, de pulsar sobre *Artistas Aleatorios*, redirige a otra vista donde se muestran cincuenta artistas aleatorios que se obtienen de la API de Spotify, tal y como se muestra en la Ilustración 8.

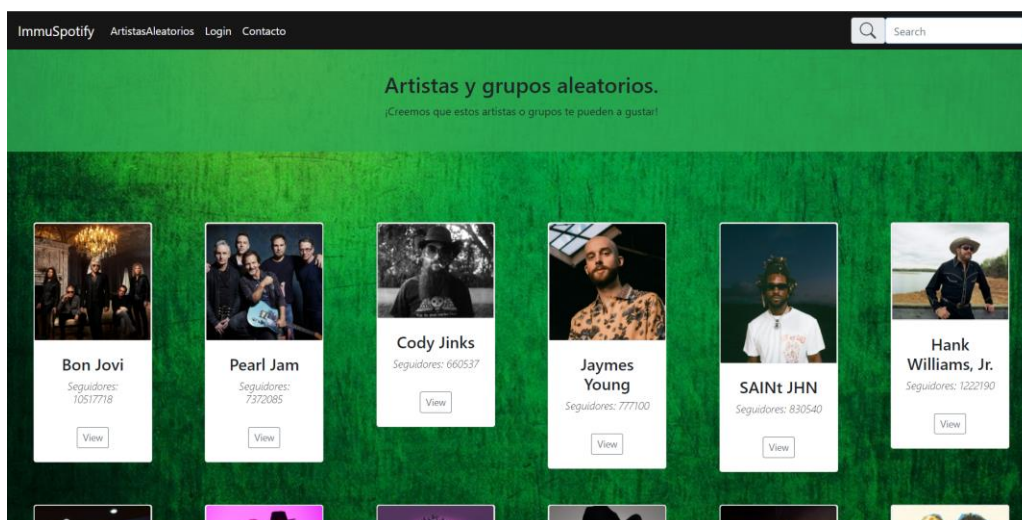


Ilustración 8. Vista de Artistas Aleatorios

Si el usuario pulsa sobre *Login* se le mostrará un pequeño formulario para que pueda iniciar sesión, tal y como se muestra en la Ilustración 9.

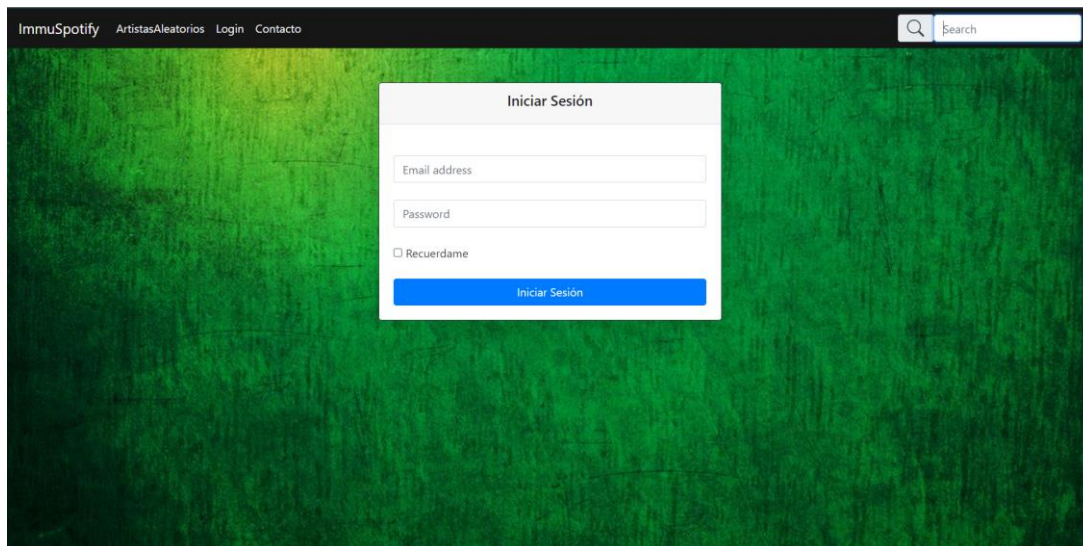
The screenshot shows the 'Iniciar Sesión' (Login) form on the ImmuSpotify website. The form is centered on a green textured background. It includes a search bar at the top right with the text 'Search'. The login form itself has a title 'Iniciar Sesión' and two input fields: 'Email address' and 'Password'. Below these fields is a checkbox labeled 'Recuerdame' (Remember me) and a blue button labeled 'Iniciar Sesión'.

Ilustración 9. Vista de Login

Si se introduce correctamente el usuario y la contraseña, ya establecidos como *“pepe@pepe.com”* y *“1234”*, se produce un inicio de sesión correcto y se accederá a la vista *Home*. Por el contrario, si el usuario o la contraseña son incorrectos, aparecerá un mensaje en rojo indicándolo.

Si el usuario navega hacia *Contacto*, será redirigido a la vista que aparece en la Ilustración 10, en ella se puede observar un mapa de donde está situado Immune Technology Institute y un correo donde se puede contactar con cualquier incidente o sugerencia.

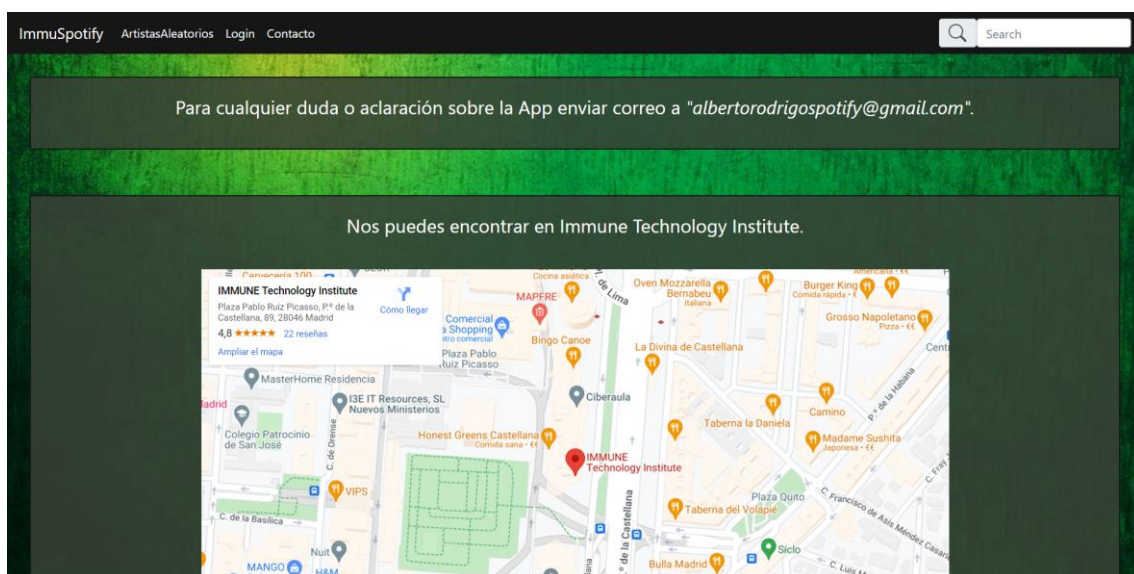


Ilustración 10. Vista de Contacto

Para terminar con la barra superior, a la derecha se observa un buscador, en caso de que el usuario quiera buscar un artista o grupo en concreto deberá introducir el nombre o una palabra relacionada, hacer clic sobre el botón de la lupa o pulsar “enter” y se mostrarán los resultados de su búsqueda. Para poner un ejemplo, se realiza una búsqueda de la palabra “bad” y los resultados son los que se muestran en la Ilustración 11.

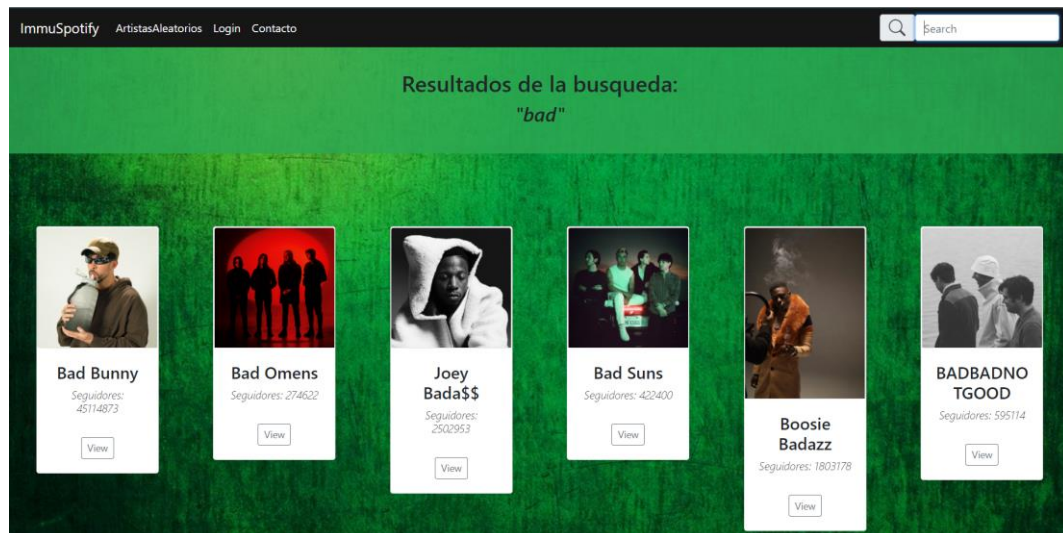


Ilustración 11. Búsqueda de la palabra bad

Vista de cada artista o grupo

En el caso de que un usuario quiera ver un artista o grupo en concreto, debe pulsar sobre el botón “View” que aparece en la carta de cada uno tanto en la vista de artistas aleatorios como en la vista de la búsqueda. Continuando con el ejemplo, se ha hecho clic para ver al artista Bad Bunny, se puede observar en la Ilustración 12.

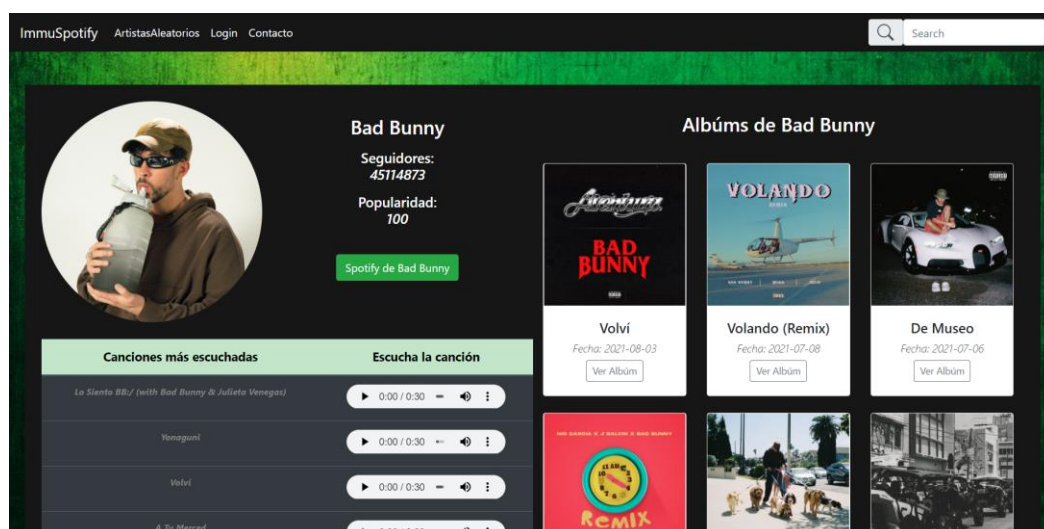


Ilustración 12. Vista de un artista en concreto

En esta vista, se observa en la parte izquierda datos del artista seleccionado, como su foto correspondiente, su popularidad y los seguidores del artista. Si el usuario lo desea, dispone de un botón verde que le redirige a la página de Spotify del artista en concreto. En la parte inferior se muestra una tabla de las canciones más escuchadas o populares del artista, en la parte derecha de esta tabla se ha proporcionado un pequeño reproductor para que el usuario pueda escuchar cada canción si así lo desea.

En la parte de la derecha se muestran los últimos álbumes que ha subido el artista a Spotify, así como la fecha de salida y un botón en el que será redirigido a la página de Spotify donde se sitúa la canción en concreto.

Página no encontrada

Si el usuario intenta acceder a una dirección que no existe o no está en nuestra aplicación, se le mostrará la vista que aparece en la Ilustración 13, permitiendo al usuario poder volver a la vista *Home*.

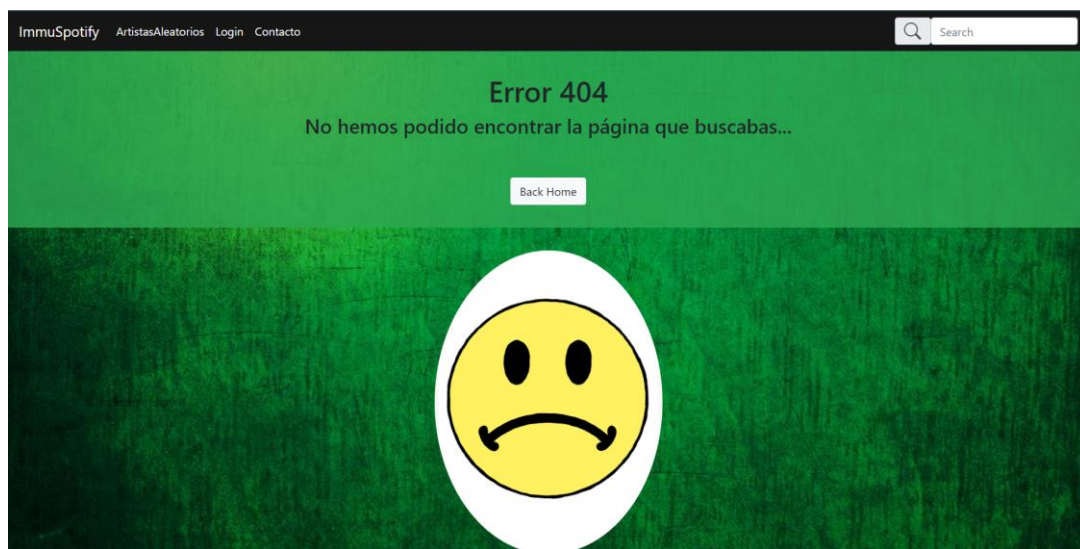


Ilustración 13. Vista de página no encontrada