



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN®

**UANL**



FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

**FIME**

# UNIVERSIDAD AUTONOMA DE NUEVO LEON

## Facultad de Ingeniería Mecánica y Eléctrica

Materia: Programación web

Hora: Jueves N1-N3

Maestro: Luis Daniel Lepe Rodríguez

Actividad: Documentación Técnica

Integrantes		
Nombre	Matricula	Carrera
Erick Azahel Aguilar Soto	1917204	ITS
Saul de León Frías	1997378	IAS

## **Índice**

**FRONTEND**

**pag 3**

**Introducción**

**Pantalla de Inicio / Home**

**Pantalla Music**

**Pantalla Discover**

**Pantalla Lists**

**Pantalla User**

**Pantalla Login**

**Pantalla My Profile**

**Pantalla Artist – Album – Song – Band**

**Pantalla Admin**

**BACKEND**

**pag 35**

## **Introducción**

Sententia es una página web orientada a el manejo de las opiniones de las personas sobre la música en general, un espacio donde el usuario podrá expresar libremente sus pensamientos sobre algún artista en específico, alguna canción que escuchó en la radio, o sobre algún álbum que haya escuchado recientemente.

Con todo esto en mente, nosotros los creadores de esta página pensamos como desarrollar de la manera correcta y mas optima este proyecto o idea que teníamos en mente.

El Frontend de esta pagina es entre comillas “sencillo” optamos por lenguajes e interfaces que fueran amigables al ojo del cliente e intuitivos, creemos que una de las claves para que una app o en este caso una página tenga éxito es que sea amigable y fácil de manejar.

Teniendo esto en mente, nuestro Frontend tiene herramientas de lenguaje como lo son: HTML – JavaScript – CSS – Boostrap5. Estas son las principales herramientas que se utilizaron, así como el uso de varias librerías a lo largo de los códigos, sobre todo en los css para poder ahorrar código y usar una librería vía internet de forma más rápida.

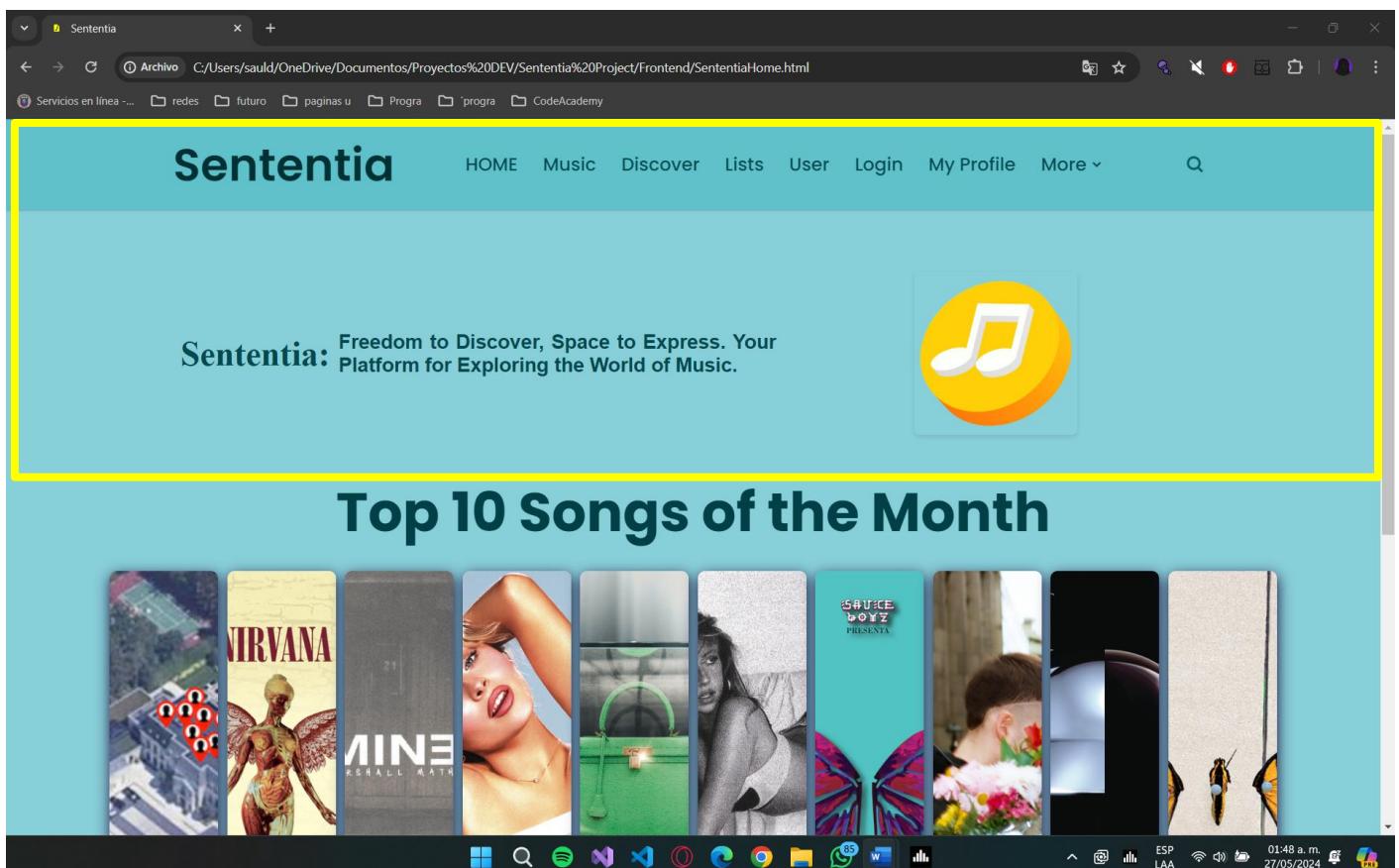
A continuación, mostrare las paginas o pantallas de nuestra pagina Sententia y daré una breve explicación técnica de lo que se uso para dicho espacio, pero nos limitaremos a puntuar los detalles mas importantes, ya que como mencione se hizo de manera en que fuera amigable al usuario e intuitiva, por lo tanto, varias páginas de Sententia se parecen o tienen la misma base.

## Pantalla Inicio / Home

La pantalla inicio o Home, es la pantalla que te recibe al iniciar en la pagina de Sententia, y surgirá la pregunta de por que no ir directo a la pestaña de Login o de Registrarse, bueno esto se debe a que, nosotros creemos que una mejor experiencia para el usuario es que no sienta instantáneamente la petición de registrarse a una pagina que aun no conoce, por lo tanto, esta pantalla es ideal para darle la bienvenida al usuario.

En ella podemos observar que se muestra la navbar en la parte del header que ahorita hablaremos de ella, después una frase con el mensaje de la pagina y seguido tenemos un tipo carrusel bastante llamativo a la vista con las canciones rankeadas como "Top Songs of the Month" que son las Top Canciones del Mes, por que lo decidimos así, pues creemos que acompañado de la forma en la que esta siendo mostrada el contenido que en este caso es en forma de un carrusel, lo acompañamos de temas de actualidad que en este caso son canciones, entonces el usuario vera contenido actual y podrá generale iteres o el que reconozca algún contenido de actualidad.

Después más abajo en la parte inferior de la página, podremos observar un footer que no se presenta en todas las pantallas ya diremos la razón, podremos observar en el footer el nombre de la pagina algunas pestañas principales accesibles desde ese punto donde no se observa el Login por el mismo motivo de que no incentivamos a sobre presionar al usuario también veremos un pequeño resumen de lo que es Sententia y de donde viene su nombre, así como los correos de los desarrolladores y links de contacto como LinkedIn y GitHub.



Sententia

HOME Music Discover Lists User Login My Profile More

# Top 10 Songs of the Month

Instant  
Crush

Daft Punk ft. Pharrell  
Williams

Sententia

HOME Music Discover Lists User Login My Profile More

All I  
Wanted

Paramore

**Sententia**

| Home | Music | Discover | Lists

Sententia © 2024

**About Sententia**

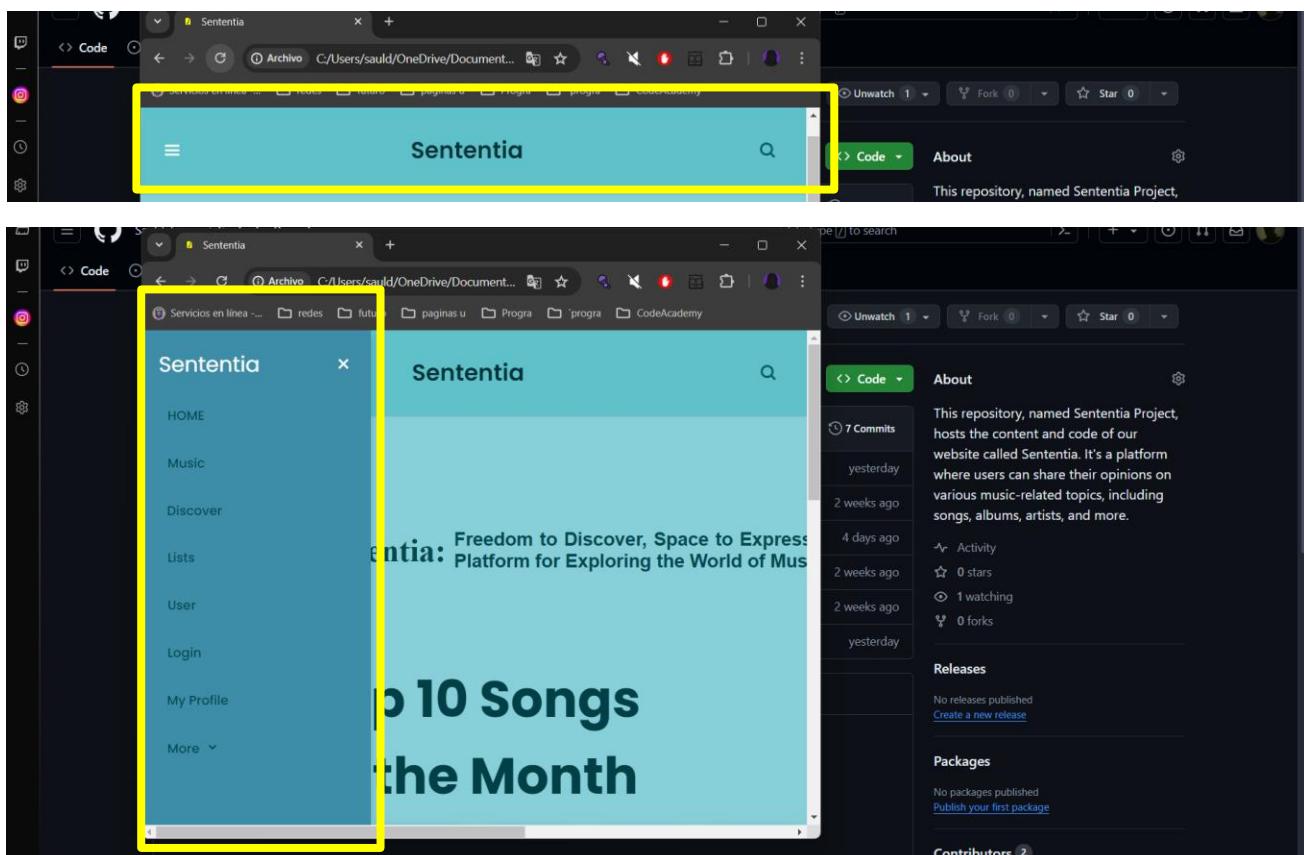
Named Sententia (Opinion in Latin). It's a platform where users can share their opinions on various music-related topics, including songs, albums, artists, and more.

Ahora del lado técnico podemos destacar que es una pantalla cargada de programas, para empezar la navbar que estará presente en todas las pestañas a lo largo de la pagina tiene la herramienta de Bootstrap5 incluida en ella, de manera que al momento en que la pagina se reduzca en cuanto pantalla, la página responderá de manera responsiva y se adaptara a el usuario.

Por otro lado, podemos destacar el uso de JavaScript en el carrusel que se muestra como contenido principal de la página, con ayuda de esta herramienta, al momento en el que el usuario coloca su mouse sobre algún elemento del carrusel este se mostrara de la forma como se ve en la imagen, mostrándose superior a los demás elementos del carrusel.

Cabe mencionar que el carrusel es fijo, tiene dinamismo, pero no se ira activando de forma automática, ocupa interés del usuario para pasar el selector por el contenido y este mismo responderá al llamado.

A continuación, un ejemplo de como reacciona la navbar al momento de reducir su tamaño y ahora como se muestra el contenido.



## Pantalla Music

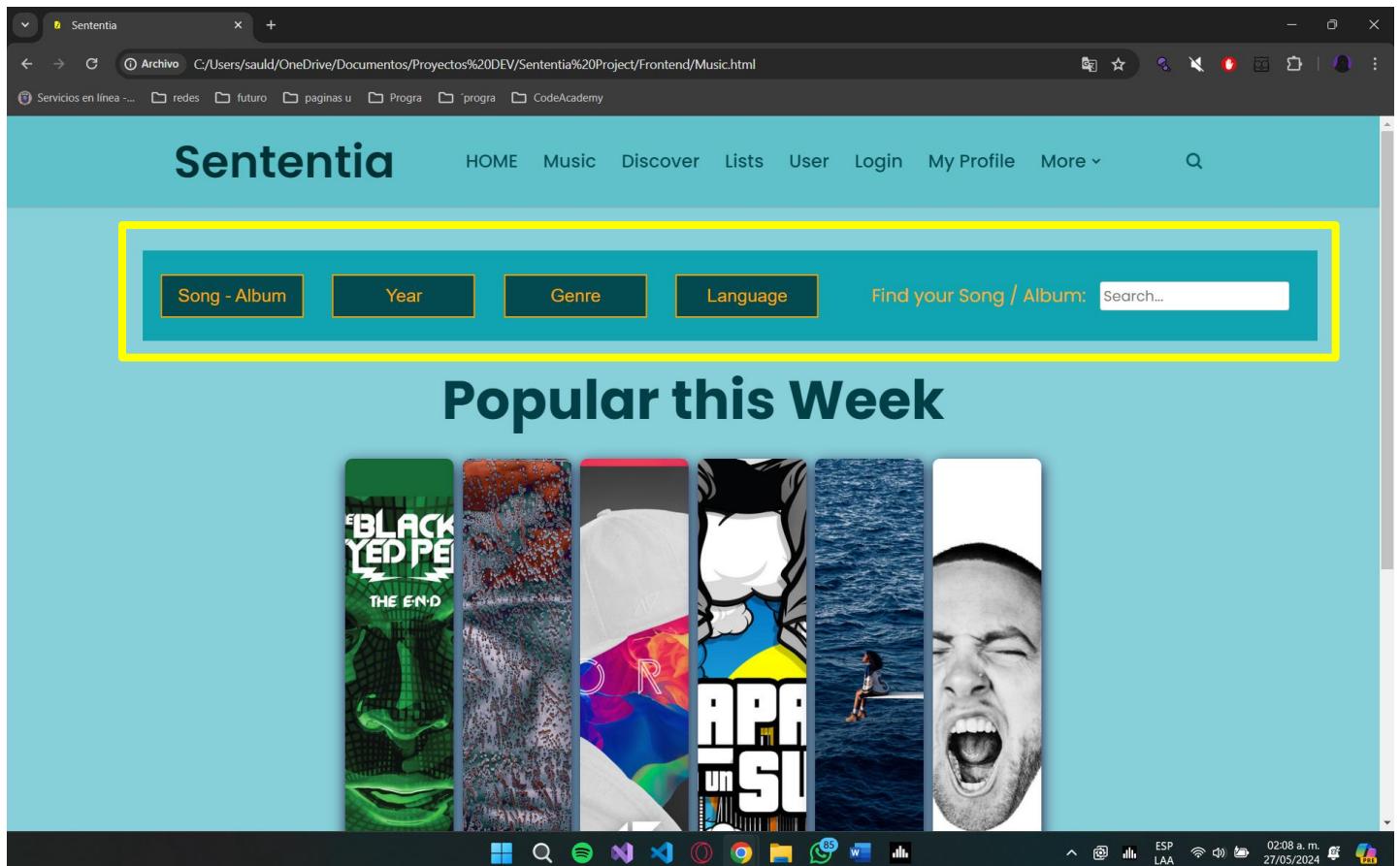
En esta pantalla que es accesible desde la Navbar o desde el footer, podemos observar la misma base que se mantendrá en todas las pantallas lo que es la navbar y todo el contenido que esta tiene y contenido que en las categorías principales se encuentra como lo es el footer.

Por lo tanto, lo que cambia en la pantalla Music es su contenido en el main, en el podremos observar primero otro tipo navbar o menú de búsqueda, en el podremos destacar botones y una search box la cual nos ayudaran para poder buscar alguna canción o álbum en específico. Los botones fueron diseñados con puro css y HTML solamente.

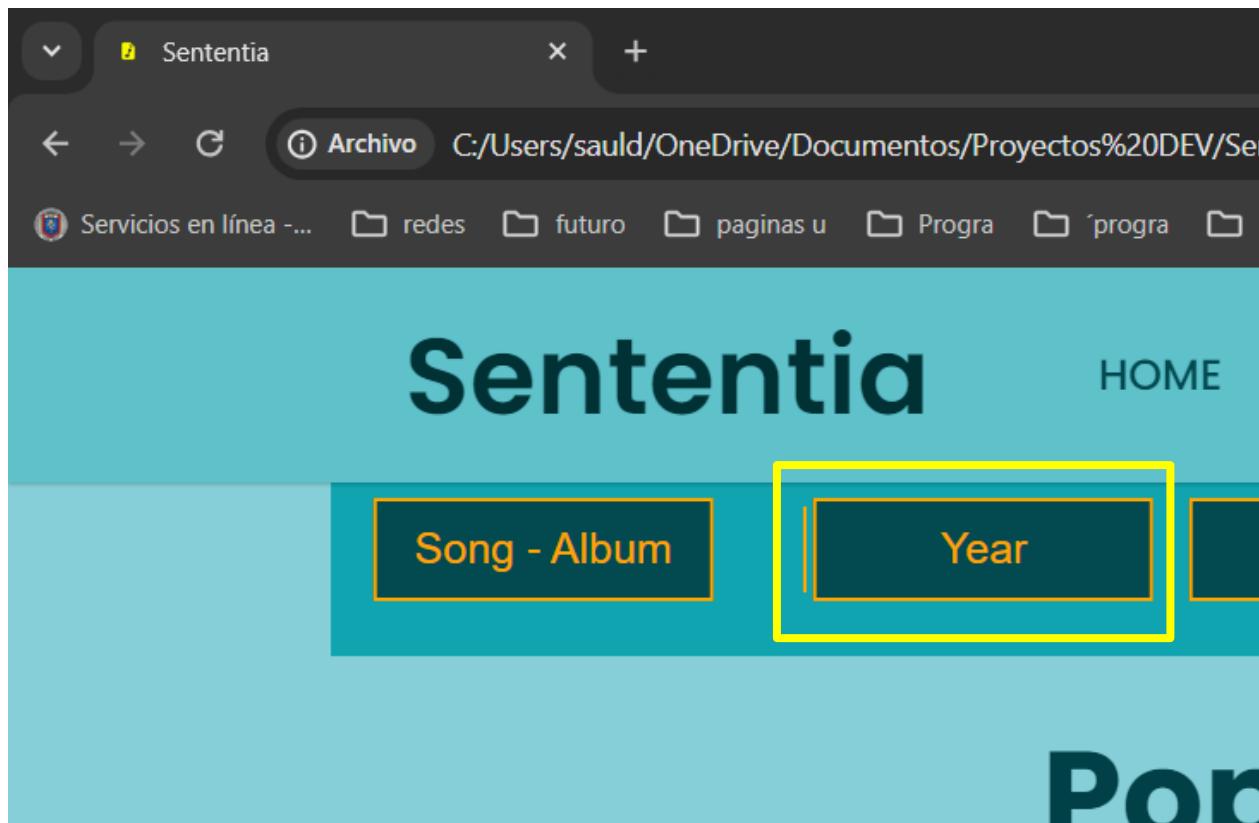
Por debajo de esto se muestra un carrusel con exactamente la misma dinámica de construcción que el de la pagina pasada de Home, solo que, en este ejemplar en esta página, mostramos menor contenido, ya que creemos que en esta pestaña un look mas minimalista y sencillo aporta un punto de vista mejor para el usuario.

En este carrusel podremos consultar cuales son los elementos musicales mas populares en la semana como lo pueden ser canciones, álbum, artistas o bandas y eso será desplegado.

Finalizamos con un footer como en la página anterior de Home.



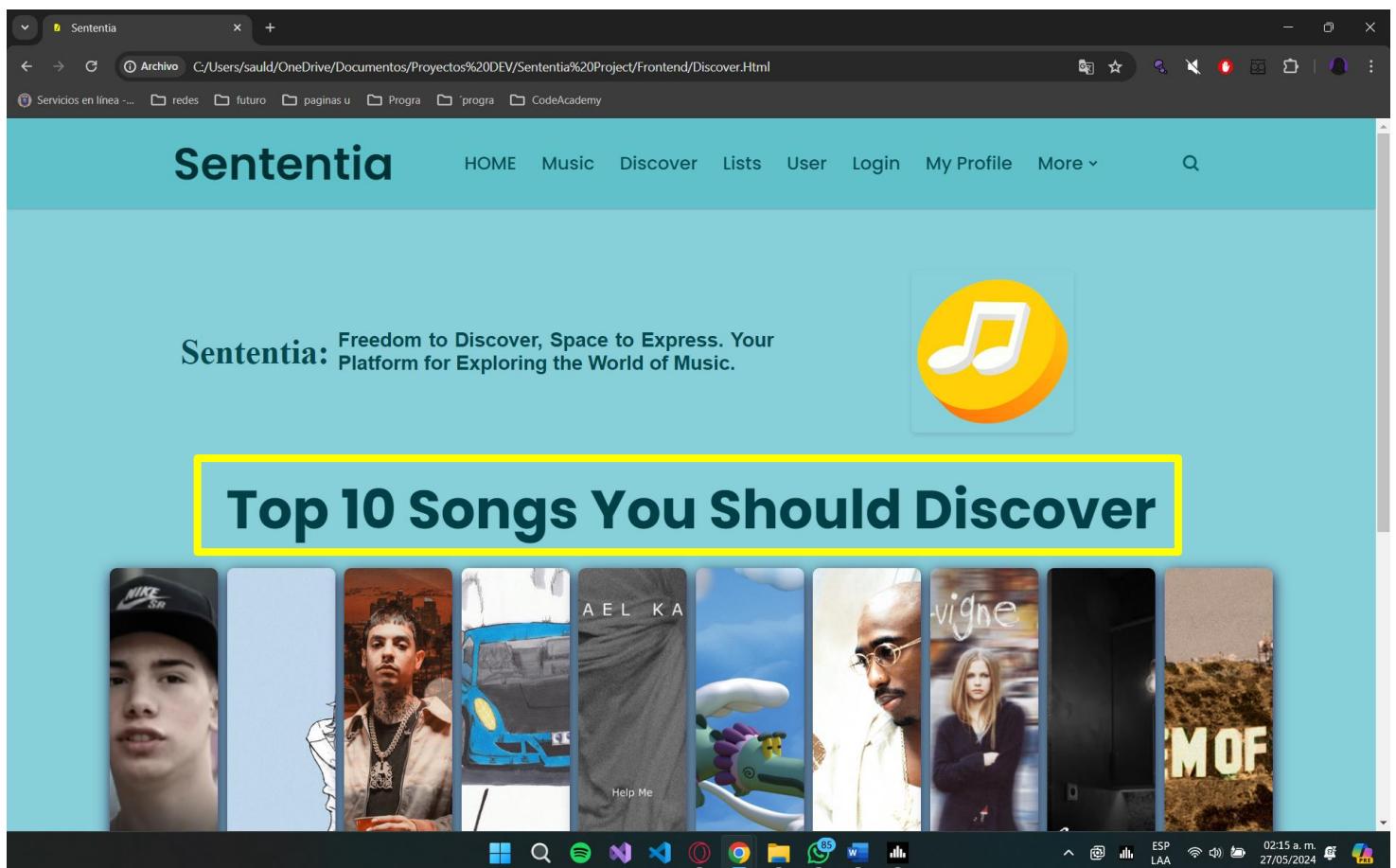
Podemos destacar que los botones tienen una ligera animación al tener el mouse encima de este mismo.



## Pantalla Discover

La pantalla Discover es muy muy parecida en apariencia y técnicamente a la pantalla de home, el propósito de esta página es meramente por y para que el usuario tenga una estancia más cómoda, grata y amplia en contenido dentro de la página, esto ya que en la pantalla de Discover podremos observar contenido musical que en este caso son canciones, que sea de los actuales o no, famosas o no, galardonadas o no, que la pagina recomienda al usuario visitar o darle una oportunidad de escuchar.

Por lo dicho anteriormente el contenido será parecido a lo mencionado anteriormente hablando de lo técnico y visual.



The screenshot shows the homepage of the Sententia platform. At the top, there is a navigation bar with links for HOME, Music, Discover, Lists, User, Login, My Profile, More, and a search icon. Below the navigation bar is a grid of ten small images, each representing a different topic or user-generated content. The images include a person's face, a hand holding a cup, a shirt with text, a drawing of a person, a shirt with 'bye' written on it, a shirt with 'Help Me' and a logo, a cartoon character, a person's hands, a dark album cover for 'Complicated' by Avril Lavigne, and a landscape photo. The main body of the page has a light blue background. On the left, there is a sidebar with the 'Sententia' logo and links for Home, Music, Discover, and Lists. In the center, there is contact information: an email address (Sauldeleonn@hotmail.com) and another (Erick.aguilar.soto@gmail.com). On the right, there is a section titled 'About Sententia' with a brief description: 'Named Sententia (Opinion in Latin). It's a platform where users can share their opinions on various music-related topics, including songs, albums, artists, and more.' Below this is a row of five small dark squares. At the very bottom of the screen, there is a taskbar with icons for various Windows applications like File Explorer, Task View, and Control Panel.

# Sententia

HOME Music Discover Lists User Login My Profile More ▾

Servicios en línea ... redes futuro paginas u Progra 'progra CodeAcademy

Archivo C:/Users/saul/OneDrive/Documentos/Proyectos%20DEV/Sententia%20Project/Frontend/Discover.Html

HOME Music Discover Lists User Login My Profile More ▾

Sententia

Help Me

by

SI MIRAR

Help Me

AL TO T

Complicated

Avril Lavigne

enoñ

ABON ARMA

MOF

TDE

ITX

GABITO BALLESTEROS X L

TECHNO GERMANY

Sendentia © 2024

Sauldeleonn@hotmail.com

Erick.aguilar.soto@gmail.com

About Sententia

Named Sententia (Opinion in Latin). It's a platform where users can share their opinions on various music-related topics, including songs, albums, artists, and more.

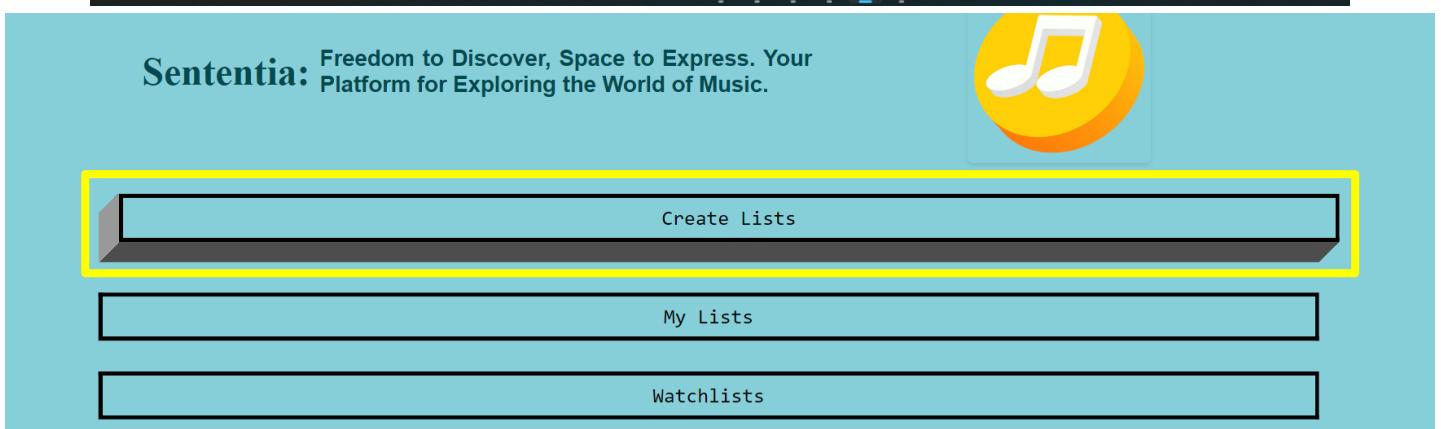
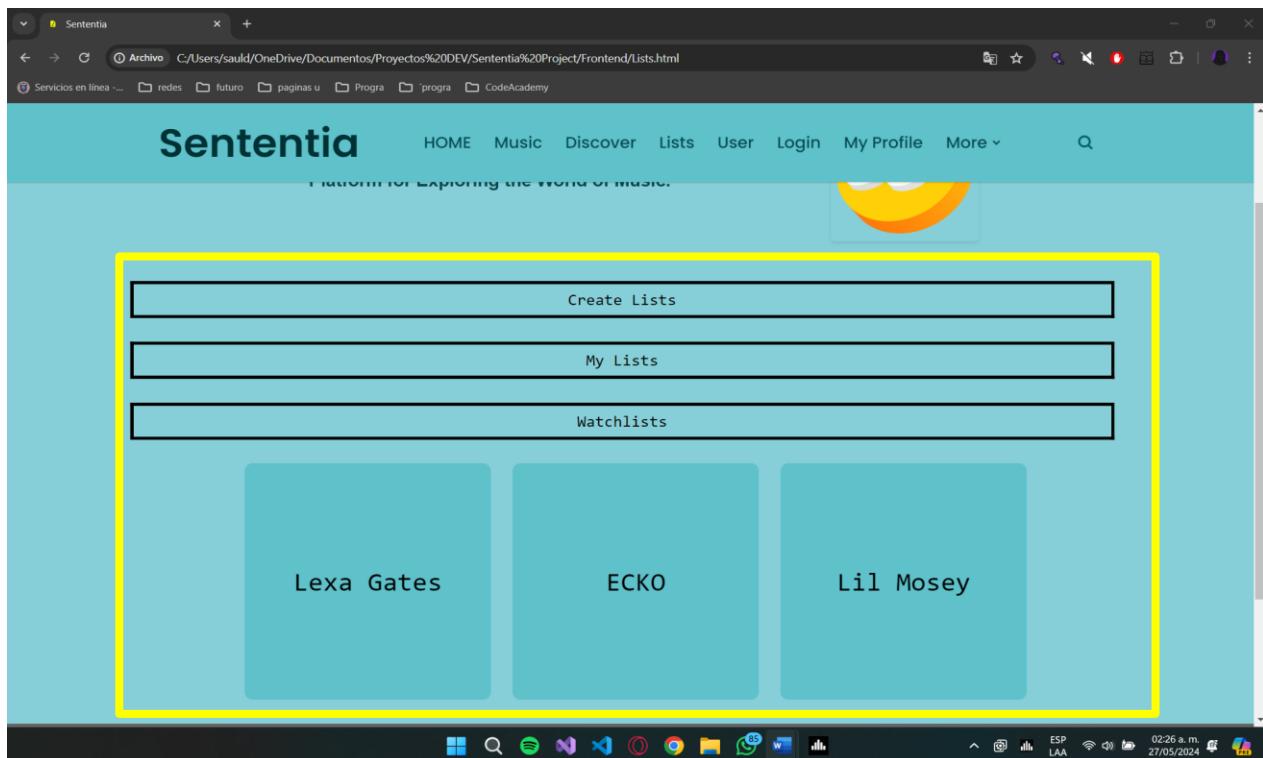
ESP LAA 02:16 a.m. 27/05/2024

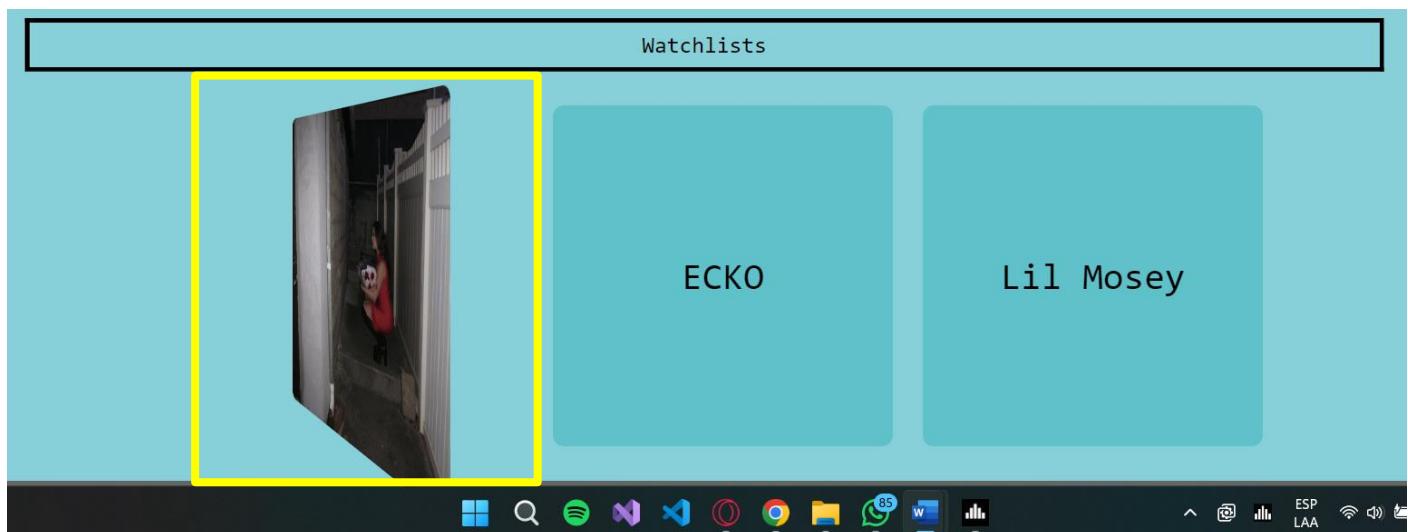
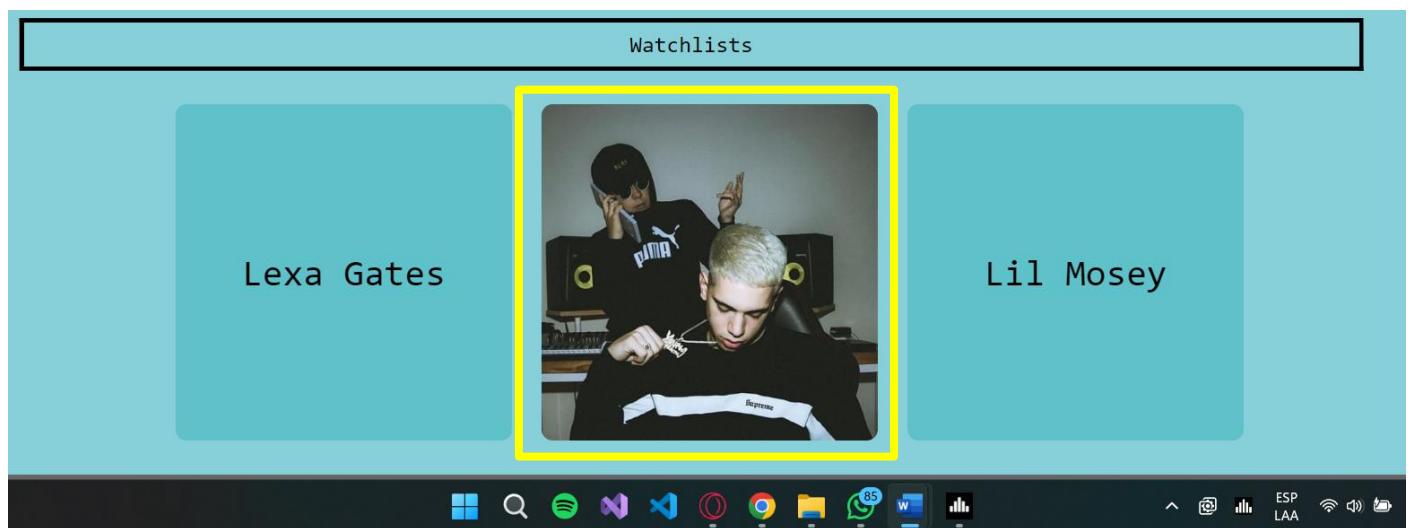
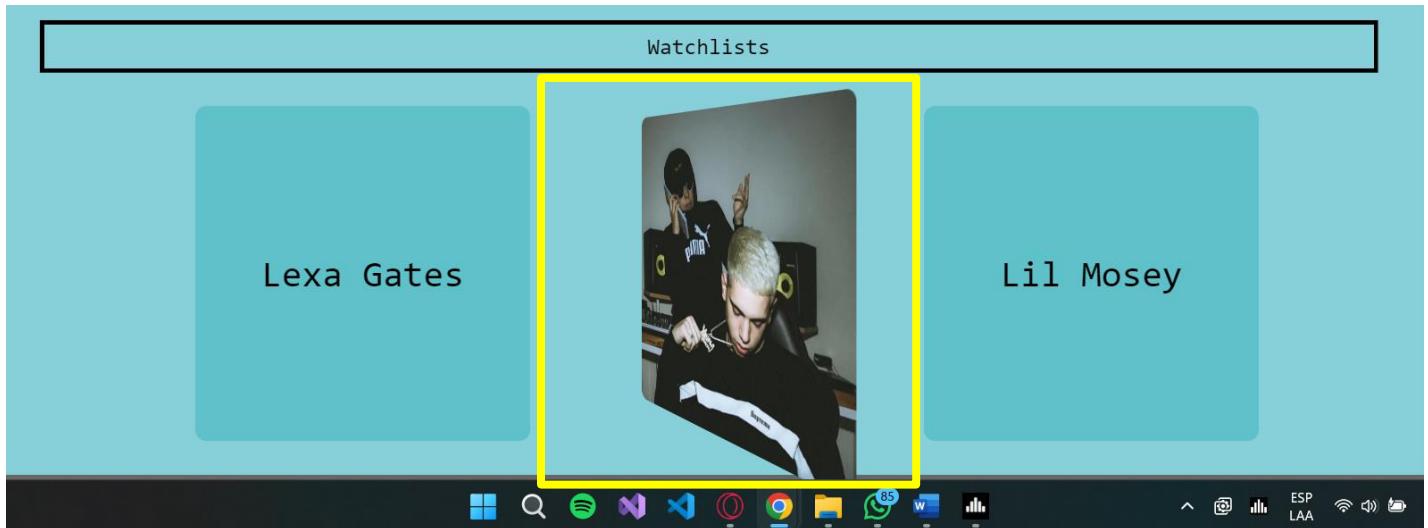
## Pantalla Lists

Como ya es mencionado anteriormente mantenemos el Header o navbar de las anteriores páginas, así como el footer, todo esto mencionando que se mantiene el contenido de ambos igualmente.

En el contenido main podemos encontrar las funcionalidades que le dan nombre a esta pantalla de Lists, en el main podremos observar 3 botones de manera horizontal que tienen animación para poder Crear una Lista, Ver tus Lista (Propias de usuario), Ver las Listas de Otros Usuarios.

Y justo debajo de lo mencionado se muestran 3 elementos en forma de cartas que muestran el nombre de un artista que Sententia te invita a que te des una oportunidad de ver su trabajo, estas cartas al pasar el mouse sobre las cartas, estas se giran individualmente una de otra mostrando una imagen correspondiente a el artista mostrado en el nombre.



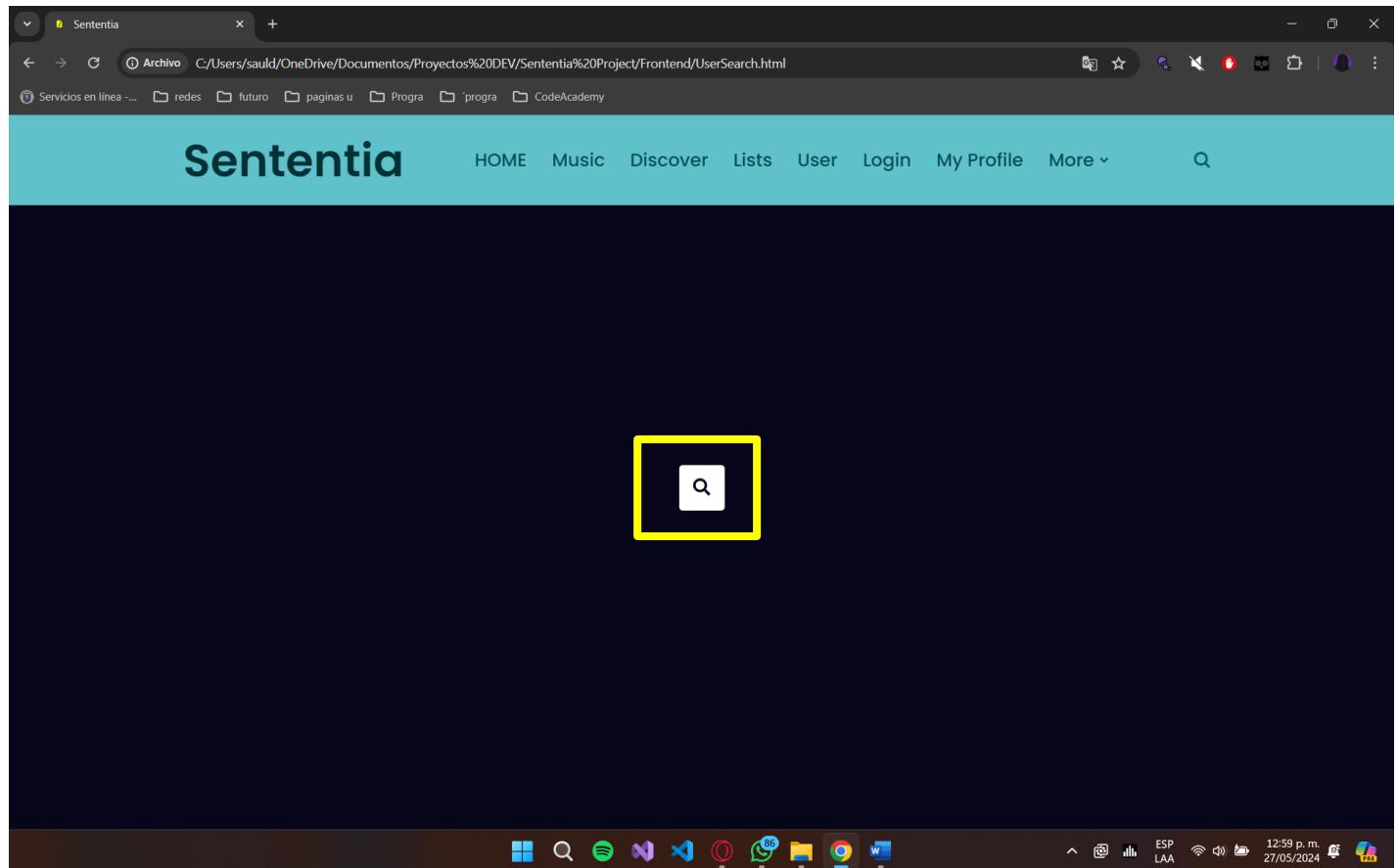


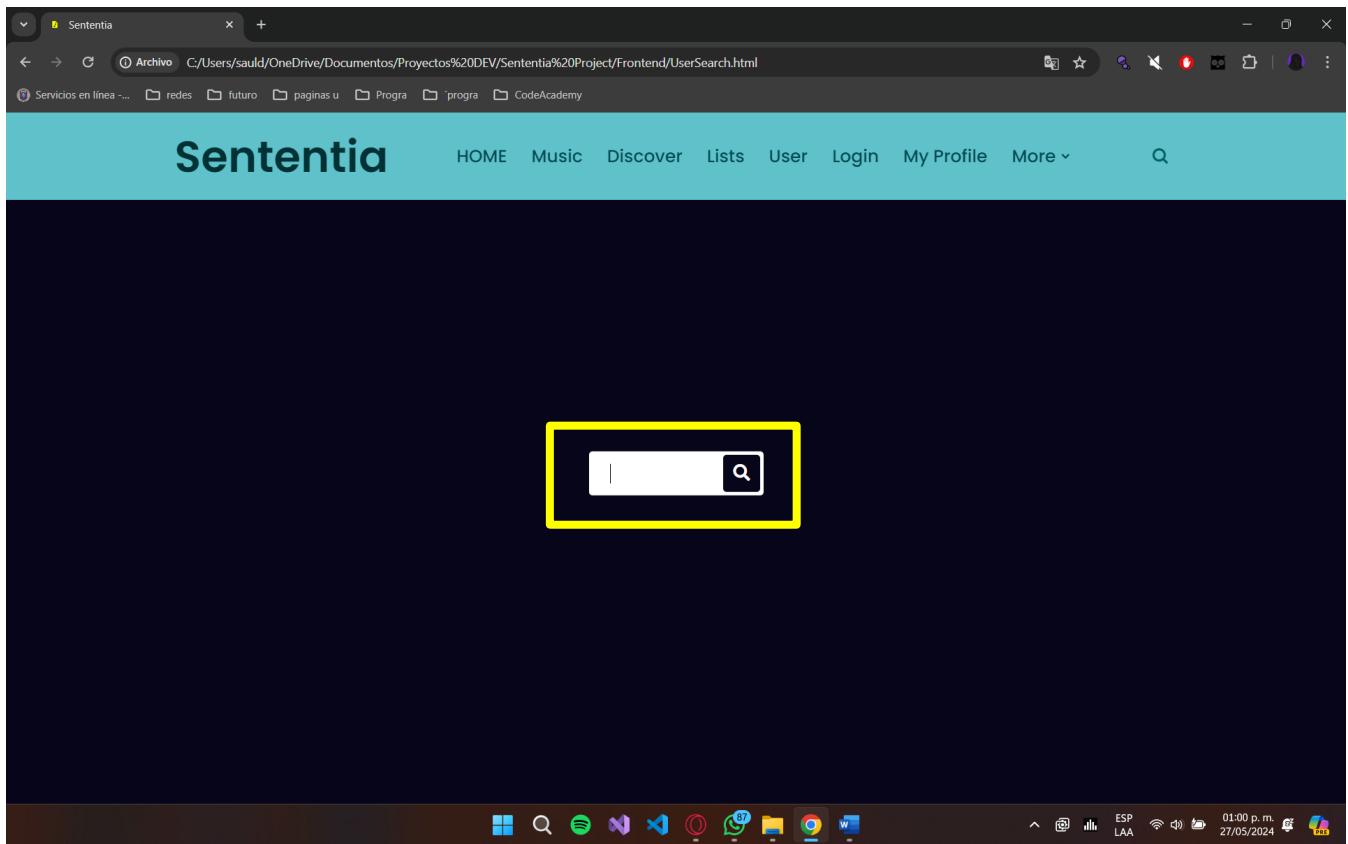
## Pantalla User

La pantalla User que se accede desde la navbar que hemos estado mencionando que se encuentra a lo largo y profundo de la pagina de Sententia, podemos acceder a esta pantalla llamada User.

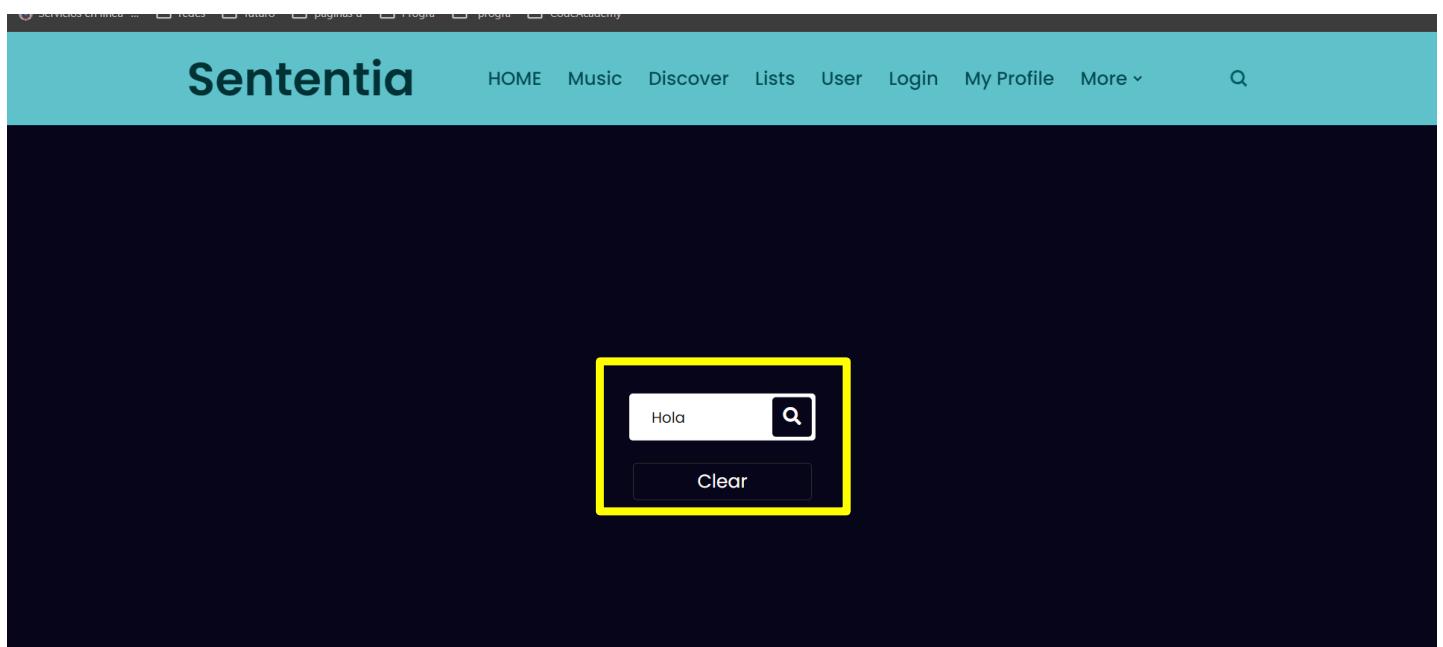
En ella podremos observar que es básicamente una página con un search box grande que se muestra de manera horizontal en el centro de la página. La función de este search box y de esta pagina en general y primordialmente es que es un buscador para poder encontrar a un usuario en específico y posteriormente poder acceder a su perfil.

Sobre el lado técnico de esta página podemos subrayar una pequeña animación que se logró tener con ayuda del JavaScript que al momento de pasar el mouse sobre la lupa de buscador o al hacer clic en ella, se despliega de manera horizontal el search box para realizar la búsqueda.





También en el momento de escribir algún carácter en el search box, inmediatamente se despliega justo debajo un botón con la palabra “clear” como se muestra en la imagen, o borrar para como su nombre lo indica vaciar el contenido que se encuentra en ese momento en la search box.



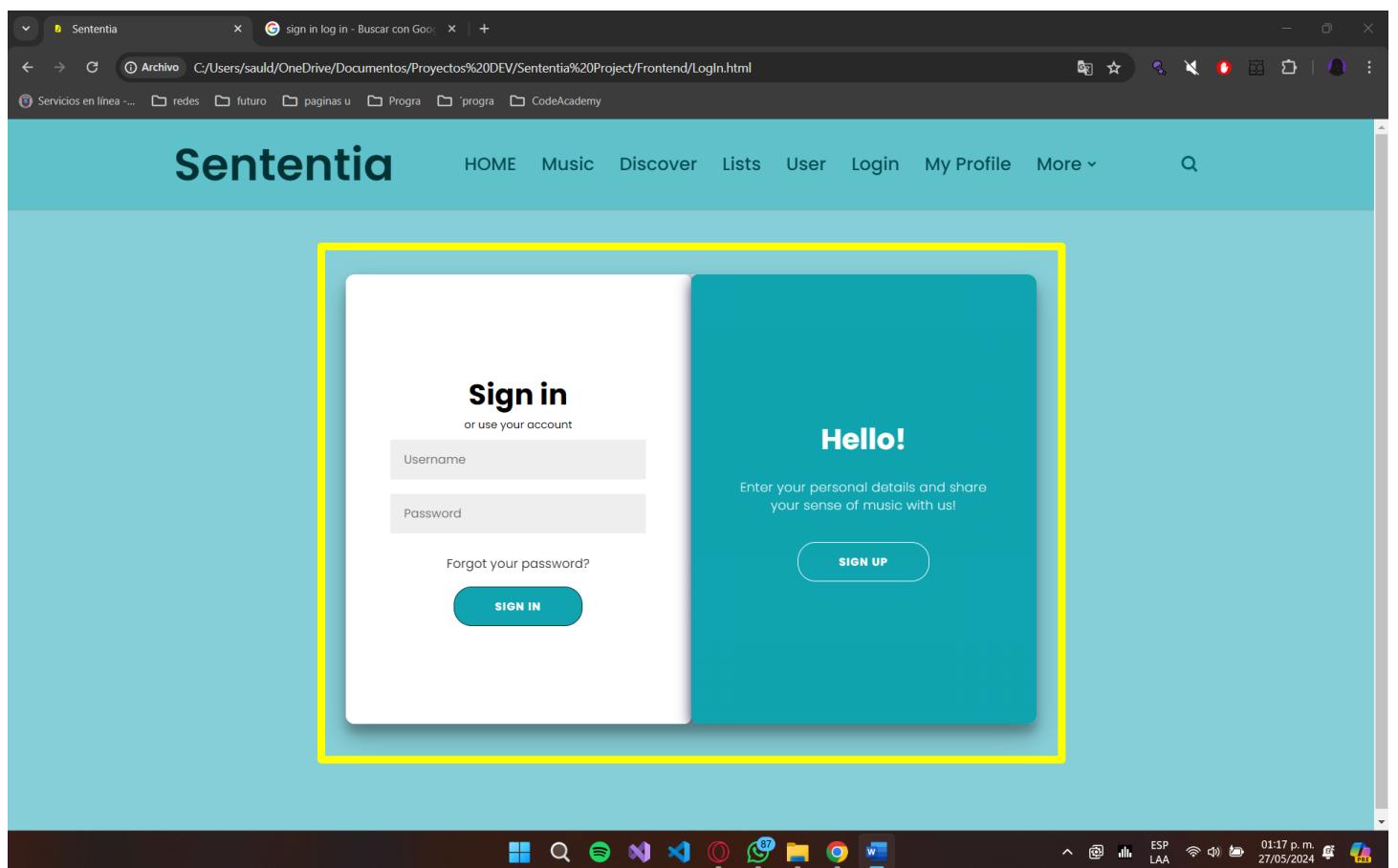
## Pantalla Log in / Sign in

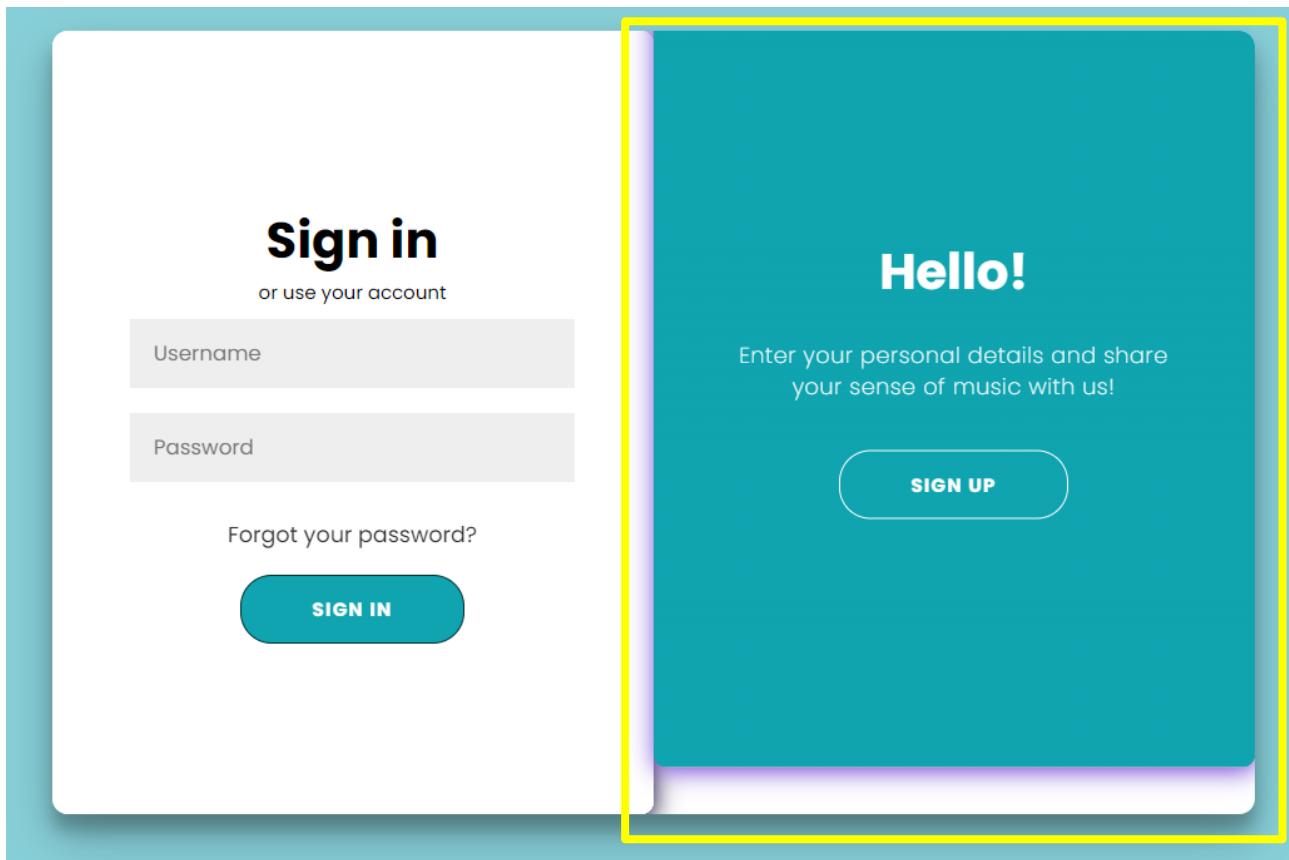
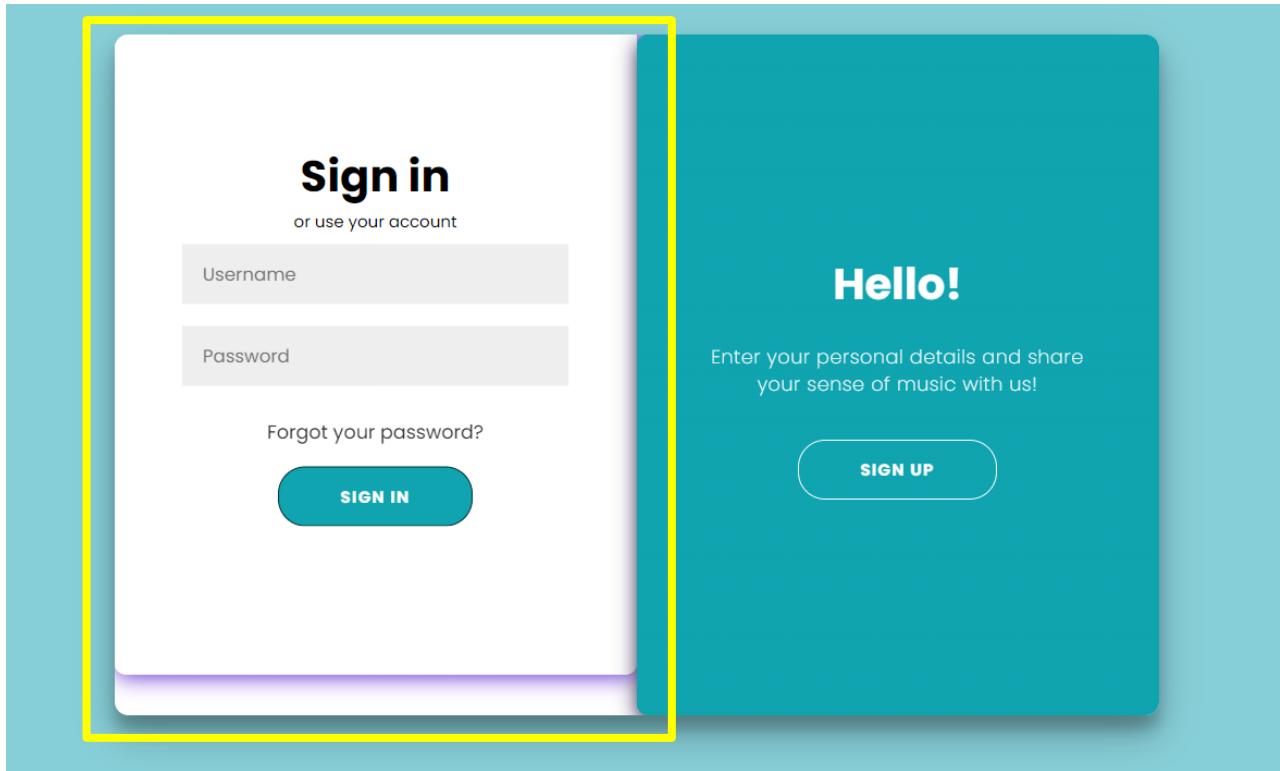
La pantalla de Login, se accede desde el navbar como todo en la página y al hacer clic en ese botón de tal nombre, nos lleva a lo que veremos en este punto que es un punto clave en la pagina ya que para poder hacer uso y generar contenido que sería la opinión de un usuario a algún elemento musical ocupa tener una cuenta creada e iniciada.

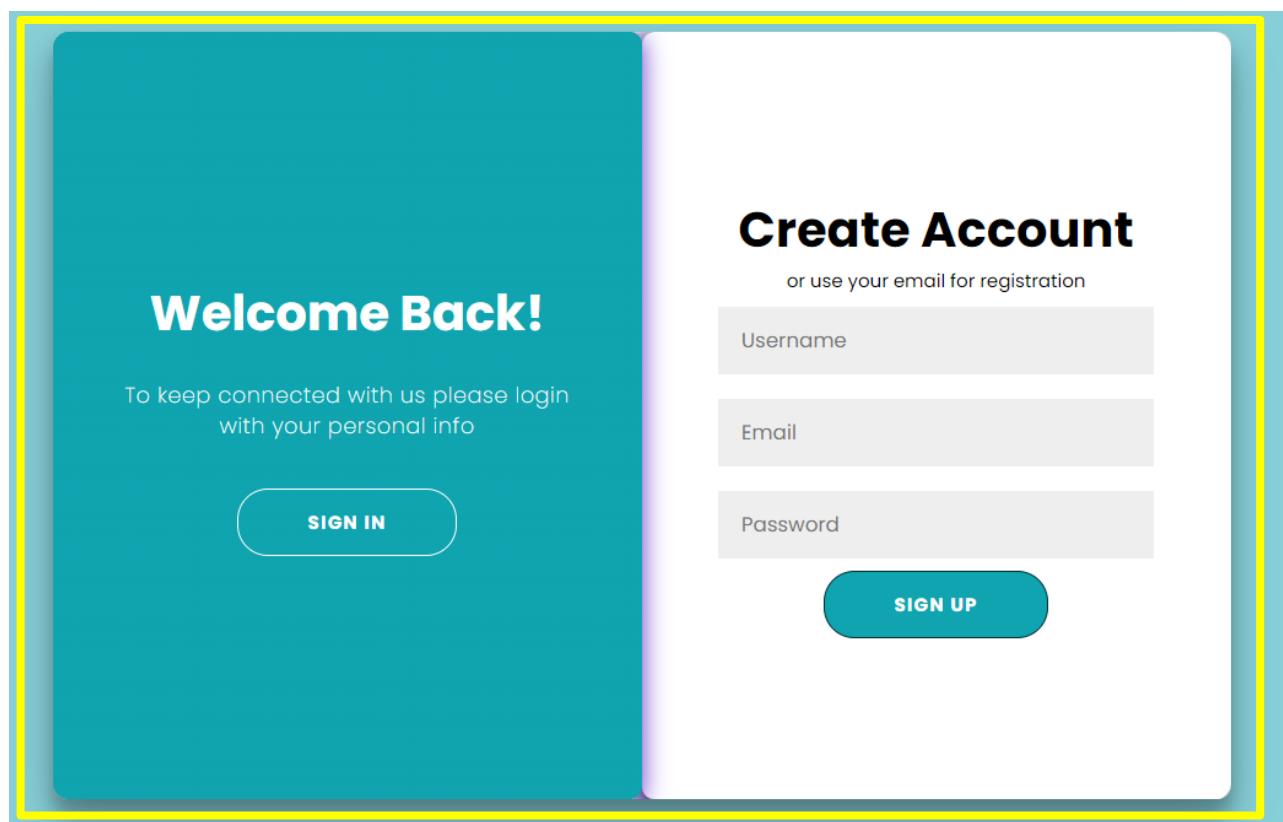
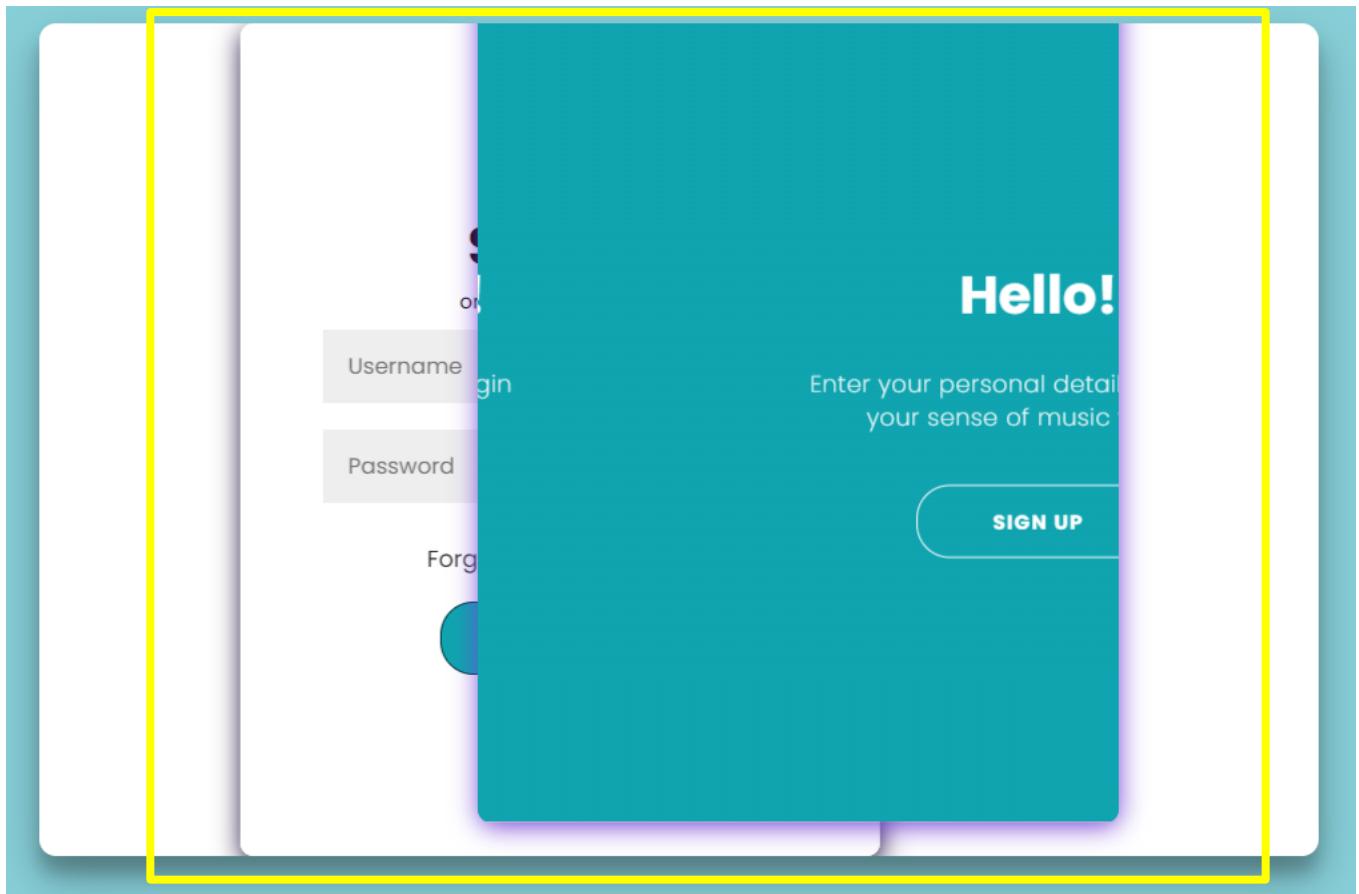
En esta pantalla podremos ver las dos opciones principales, que es el crearse una cuenta o iniciar sesión en alguna cuenta existente.

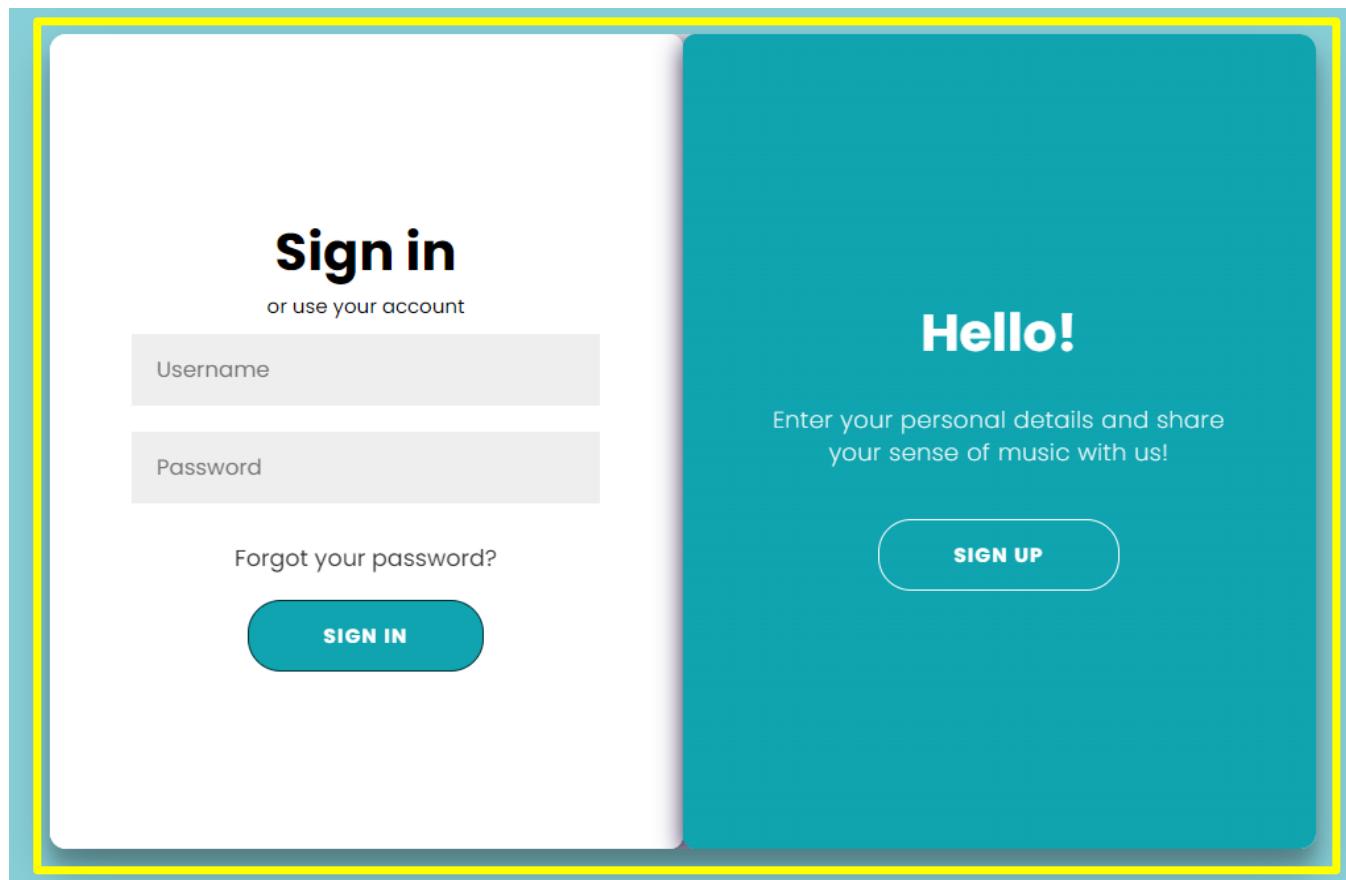
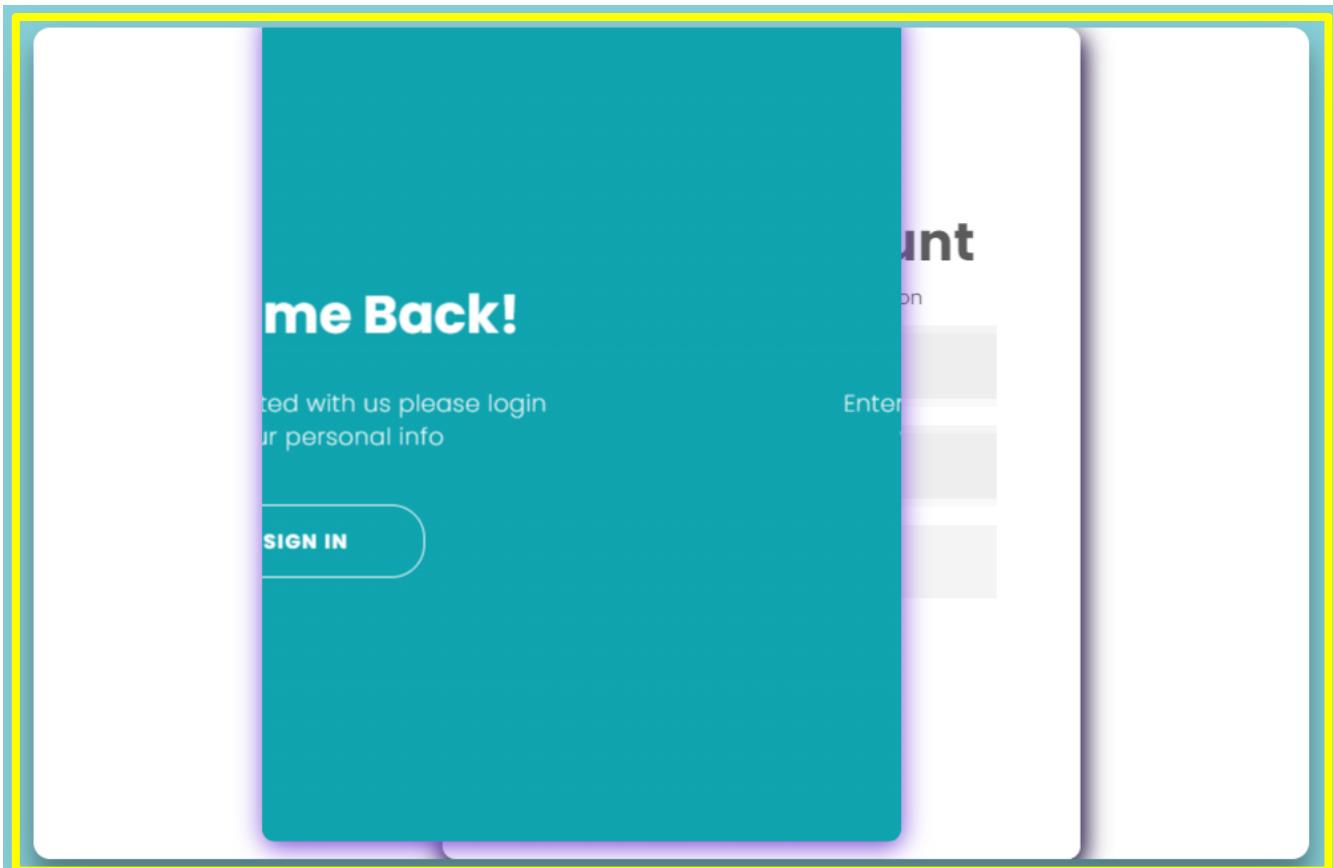
En el lado técnico podremos ver que es bastante dinámico este elemento de la página, fue creado con principalmente JavaScript, que le aporto todo el dinamismo y movimiento que lo hace llamativo al usuario y le da un aspecto mas profesional y mas trabajado.

Al pasar el mouse por encima de cualquiera de las dos cartas ya sea Sign in o Sign Up habrá un pequeño desplazamiento de la carta para hacer notar al usuario que esta por seleccionar ese elemento. También si quieres pasar de Sign in a Sign Up habrá una animación que cambiará de lugar y color a las cartas lo que le da un aspecto más tecnológico.









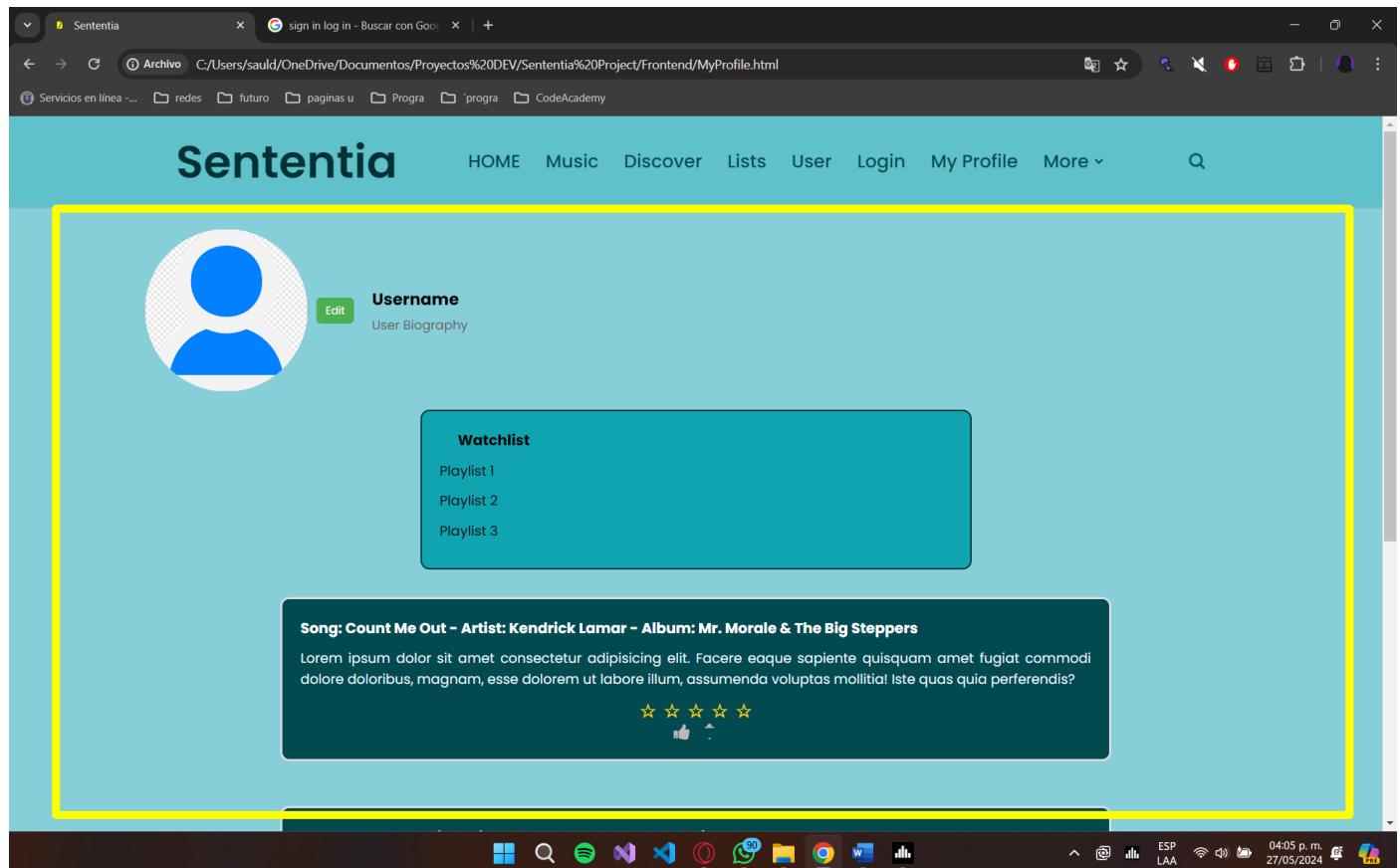
## Pantalla My Profile

En esta pantalla que también es accesible desde el navbar que se encuentra en el header de la pagina se llama My Profile que corresponde a el perfil de un usuario ya registrado en la página de Sententia.

El funcionamiento de esta pagina es bastante sencillo, en esta pantalla solo se utilizo HTML y css y muy poco JavaScript en algunos detalles minúsculos.

Empezando por lo que compone al contenido de la página podemos observar que primero veremos la imagen de perfil de un usuario y una pequeña biografía, después veremos algunas listas ya creadas por el mismo usuario en un box con el nombre de WatchList después en seguida de eso se mostraran en formato cascada las Reviews a elementos musicales que el usuario haya realizado dentro de la página de Sententia.

En cuanto al pequeño uso de JavaScript que se menciono que se usa, lleva lugar en las Reviews de los usuarios, las que se presentan varios botones interactivos dentro de el box de Review, primero tenemos las estrellas que sirven para puntuar por método tipo ranking la review de 1 a 5, después veremos el botón de Like y por último el botón de Report, este ultimo sirve para reportar por si algún usuario nota que la review infringe con el ambiente de la pagina el cual es que se respete siempre la opinión pero sin faltar a los demás.



Sententia

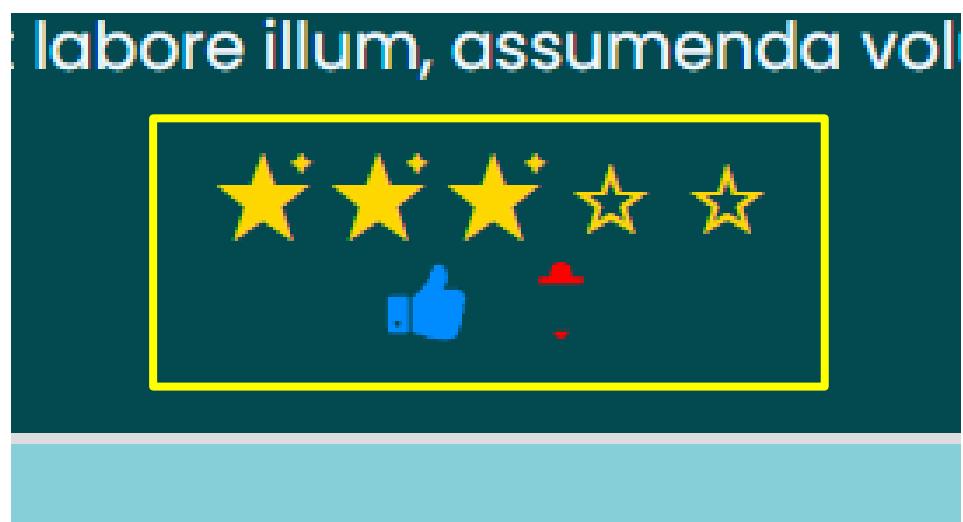
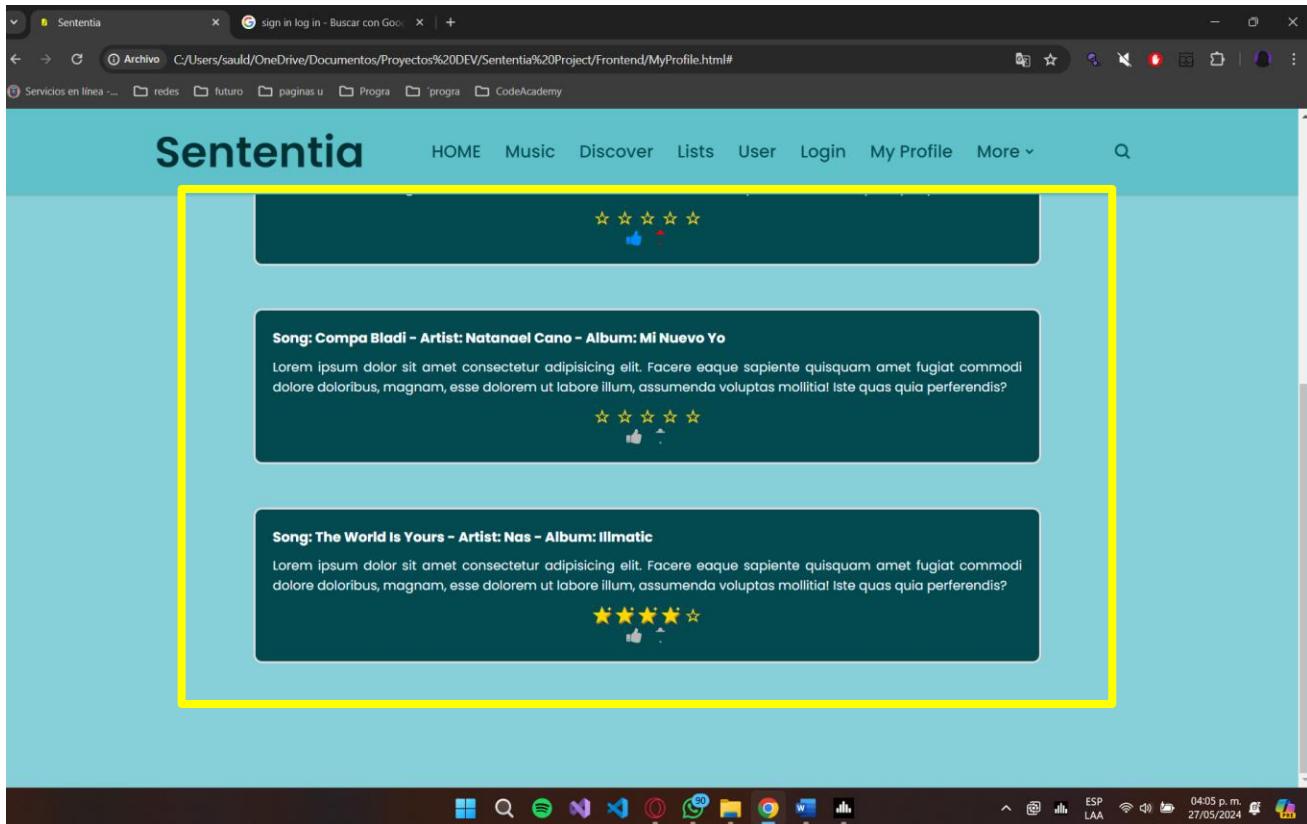
HOME Music Discover Lists User Login My Profile More ▾

Song: Compa Bladi - Artist: Natanael Cano - Album: Mi Nuevo Yo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Facere eaque sapiente quisquam amet fugiat commodi dolore doloribus, magnam, esse dolorem ut labore illum, assumenda voluptas mollitia iste quas quia perferendis?

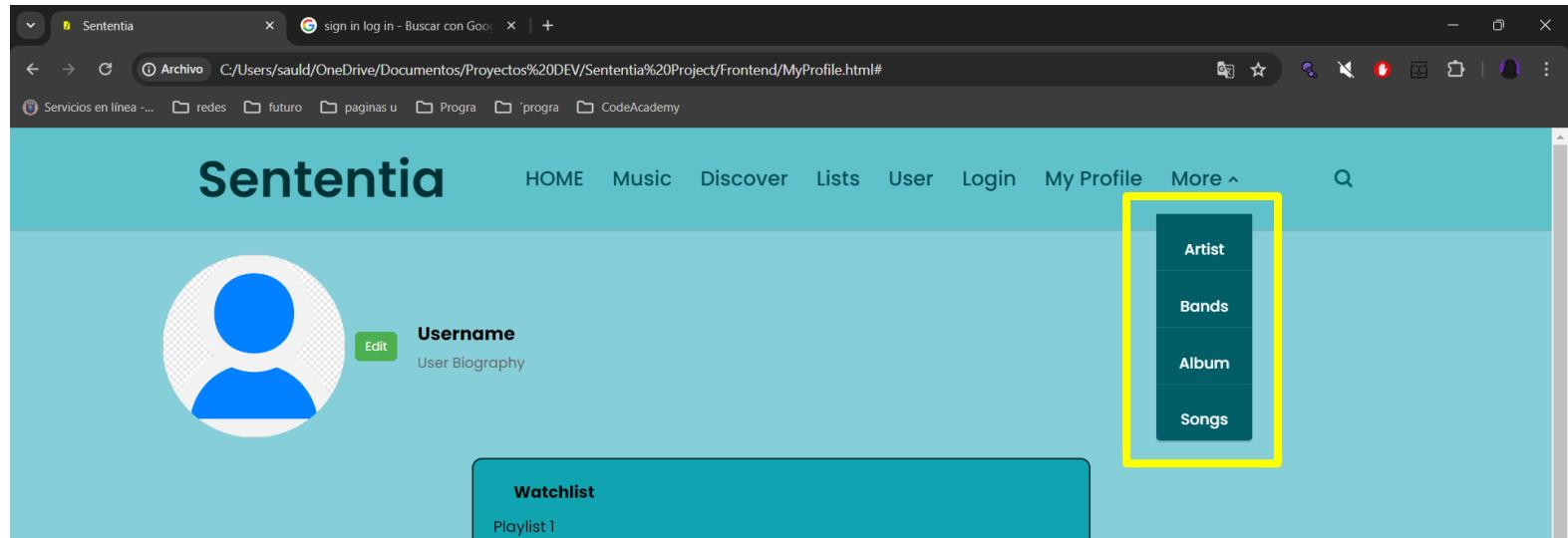
Song: The World Is Yours - Artist: Nas - Album: Illmatic

Lorem ipsum dolor sit amet consectetur adipisicing elit. Facere eaque sapiente quisquam amet fugiat commodi dolore doloribus, magnam, esse dolorem ut labore illum, assumenda voluptas mollitia iste quas quia perferendis?



## Pantalla Artists – Bands – Songs – Albums

Esta pantalla es accesible, pasando el mouse por encima de el botone de MORE en la navbar, ya que nos desplegara un mini menú de opciones desplegables hacia abajo tipo cascada y podremos hacer clic en cualquiera de las 4 opciones que se mostraran.



Ya haciendo clic en cualquiera de las 4 opciones que se muestran en pantalla como en la imagen anterior, podremos visualizar el método que podremos usar para poder hacer alguna review a alguno de los 4 tipos de elementos musicales que están disponibles en la página.

Después una vez seleccionado alguno de las 4 opciones podremos ver que el diseño de los 4 tipos tiene la misma base, esto con el fin de que el usuario no tenga que familiarizar un nuevo entorno para cada tipo de elemento musical que se ofrece en la página.

Del lado técnico, no hay mucho de lo que podamos destacar ya que es una pantalla bastante sencilla en cuanto programación con HTML y css mas que nada para poder hacer un uso fácil del contenido que se mostrara en cada uno de los 4 tipos de elementos musicales.

The screenshot shows a web browser window with the title bar "Sententia" and "sign in log in - Buscar con Google". The address bar shows the path "C:/Users/sauld/OneDrive/Documentos/Proyectos%20DEV/Sententia%20Project/Frontend/Artist.html". Below the address bar, the navigation menu includes "HOME", "Music", "Discover", "Lists", "User", "Login", "My Profile", and "More". A search icon is also present. The main content area features a teal header with the text "Rate This Artist". Below this, there is a banner section labeled "Banner of the artist + bio + details". Underneath the banner is a rating section with a thumbs up icon, five yellow stars, and the text "Add to list Add review". Three buttons labeled "Albums", "Songs", and "Reviews" are displayed below the rating section. The entire central content area is enclosed in a yellow rectangular border.

Podremos observar que vendrá el nombre del artista, un banner con una pequeña biografía del mismo, y podremos ver que habrá un box la cual nos dará la opción de agregar una review, agregar ese elemento a alguna lista propia existente o de ver algunos Albums que tenga el artista así como canciones y las propias Reviews de otros usuarios de Sententia que se mostraran en forma cascada hacia debajo de la página.

## Pantalla Control Center / Admin

La pantalla admin esta accesible desde.

La funcionalidad de la pantalla de Admin es llevar el control general de toda la página, ya que en esta pantalla podrás acceder dependiendo tu rol si es de Admin o de Staff o moderador podrás acceder a toda la información que la pagina puede generar o almacenar sobre la misma pagina o sobre los mismos usuarios que están transitando sobre la misma.

En esta pantalla como recalcamos dependiendo tu rango podrás hacer varias actividades como lo son, el dar de alta un nuevo miembro para alguno de los roles, poder ver el crecimiento exponencial de la página, poder editar, crear o borrar algún genero o como lo conocemos como tag dentro de la página, también podrás como moderador tener la tarea y responsabilidad de ver los posibles reports que pudiera haber en la página echo por algún usuario que haya notado un comportamiento inadecuado de otro usuario.

Otra de las tareas que se pueden realizar desde esta pantalla, es el llevar un control de los elementos musicales, desde el poder buscar un elemento en específico, hasta también poder editar su contenido o poder borrarlo, así como también el poder crear un elemento musical nuevo.

Otra de las posibilidades que tiene, pero solo el Admin es el poder ver la información de un usuario, desde cuando se registró, su actividad y su nombre de usuario, y así por ejemplo poder viajar mediante alguna search box que corresponda ir a su perfil propio del usuario.

Sententia

Manage

Staff

Song

Album

Artist

Band

Tag

Language

Genre

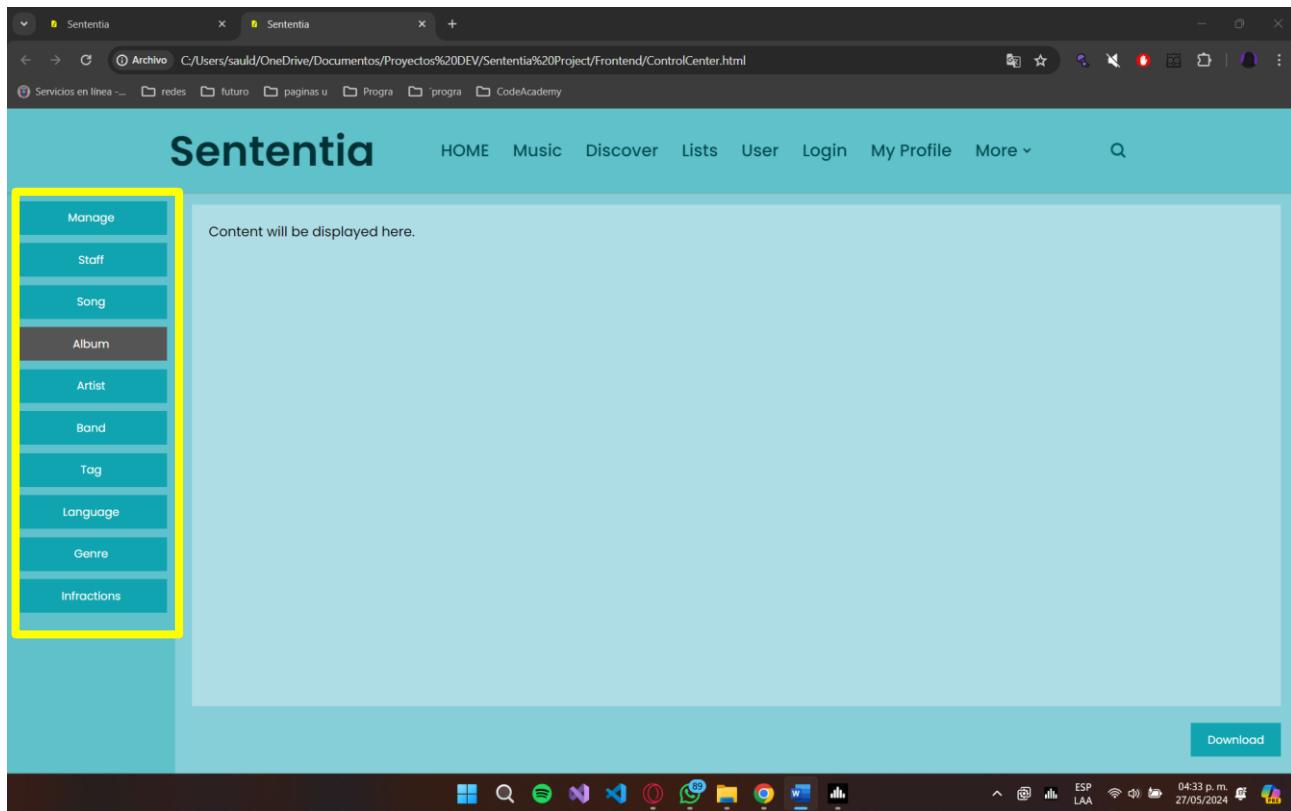
Infractions

Content will be displayed here.

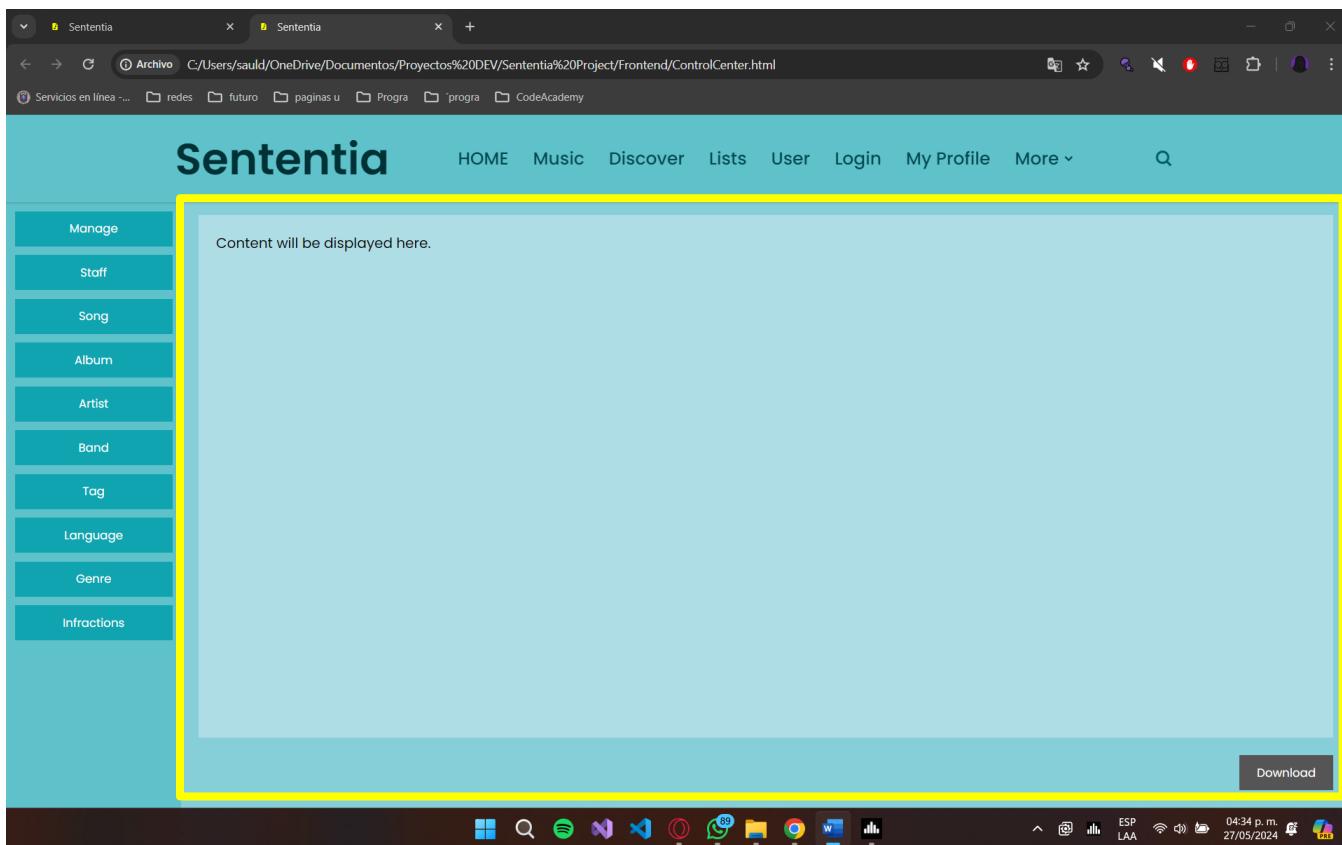
HOME Music Discover Lists User Login My Profile More ▾

Download

04:33 p. m. 27/05/2024



Como se muestra también, tiene un botón de Download para poder descargar los datos de la pagina web.



## Código del Frontend

### Sententia Home

Frontend > SententiaHome.html > HTML > body > header > nav > div.nav-bar > div.nav-links > ul.links > li > ul.js-sub-menu.sub-menu > li

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="UTF-8">
5     <title>Sententia</title>
6     <link rel="icon" href="images/file-aesthetic-music-fill.ico" type="image/x-icon">
7     <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600;700&family=Ubuntu:wght@400;700;900" rel="stylesheet">
8     <link href="https://unpkg.com/huxicon@0.8.7/css/huxicons.min.css" rel="stylesheet">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    </head>
11    <body>
12      <nav>
13        <div class="nav-bar">
14          <div class="bx menu"></div>
15          <div class="logos">
16            <a href="SententiaHome.html">Sententia</a>
17            <div class="js-logos"><a href="SententiaHome.html">Sententia</a></div>
18          </div>
19          <div class="nav-links">
20            <ul class="links">
21              <li class="logos"><a href="#">Logo</a>
22              <li class="links"><a href="#">Home</a>
23              <li class="links"><a href="#">Discover</a>
24              <li class="links"><a href="#">Artists</a>
25              <li class="links"><a href="#">Albums</a>
26              <li class="links"><a href="#">Reviews</a>
27              <li class="links"><a href="#">Rating Reviews</a>
28              <li class="links"><a href="#">recommendedArtists</a>
29              <li class="links"><a href="#">SearchResults</a>
30              <li class="links"><a href="#">SongResults</a>
31              <li class="links"><a href="#">ArtistResults</a>
32              <li class="links"><a href="#">AlbumResults</a>
33              <li class="links"><a href="#">SongResults</a>
34            </ul>
35            <ul class="more">
36              <li href="#">More</a>
37              <li href="#">Chevron-down <span>arrow</span></li>
38            </ul>
39          </div>
40          <div class="js-sub-menu sub-menu">
41            <ul class="links">
42              <li href="Artist.html">Artist</a>
43              <li href="Album.html">Album</a>
44              <li href="Song.html">Song</a>
45            </ul>
46          </div>
47        </div>
48        <div class="search-box">
49          <div class="bs-search"></div>
50          <div class="input-box">
51            <input type="text" placeholder="Search..."/>
52          </div>
53        </div>
54      </div>
55    </header>
56
57    <main>
58      <div class="page-info">
59        <div class="page-info-content">
60          <h3>What's New <span>Sententia</span></h3>
61          <p>Freedom to Discover, Space to Express. Your Platform for Exploring the world of Music.</p>
62          <div class="page-info-image">
63            <img alt="Imagen de la página" data-bbox="388 488 488 518"/>
64          </div>
65        </div>
66      </div>
67
68      <div class="tagline">
69        <h2>Top 10 Songs of the Month</h2>
70      </div>
71    </main>
72
73  </body>
74</html>
```

Lineas 39 col 58 Espacio 2 UTF-8 HTML Blacklist

04:49 p. m. 27/05/2024

Frontend > SententiaHome.html > HTML > body > header > nav > div.nav-bar > div.nav-links > ul.links > li > ul.js-sub-menu.sub-menu > li

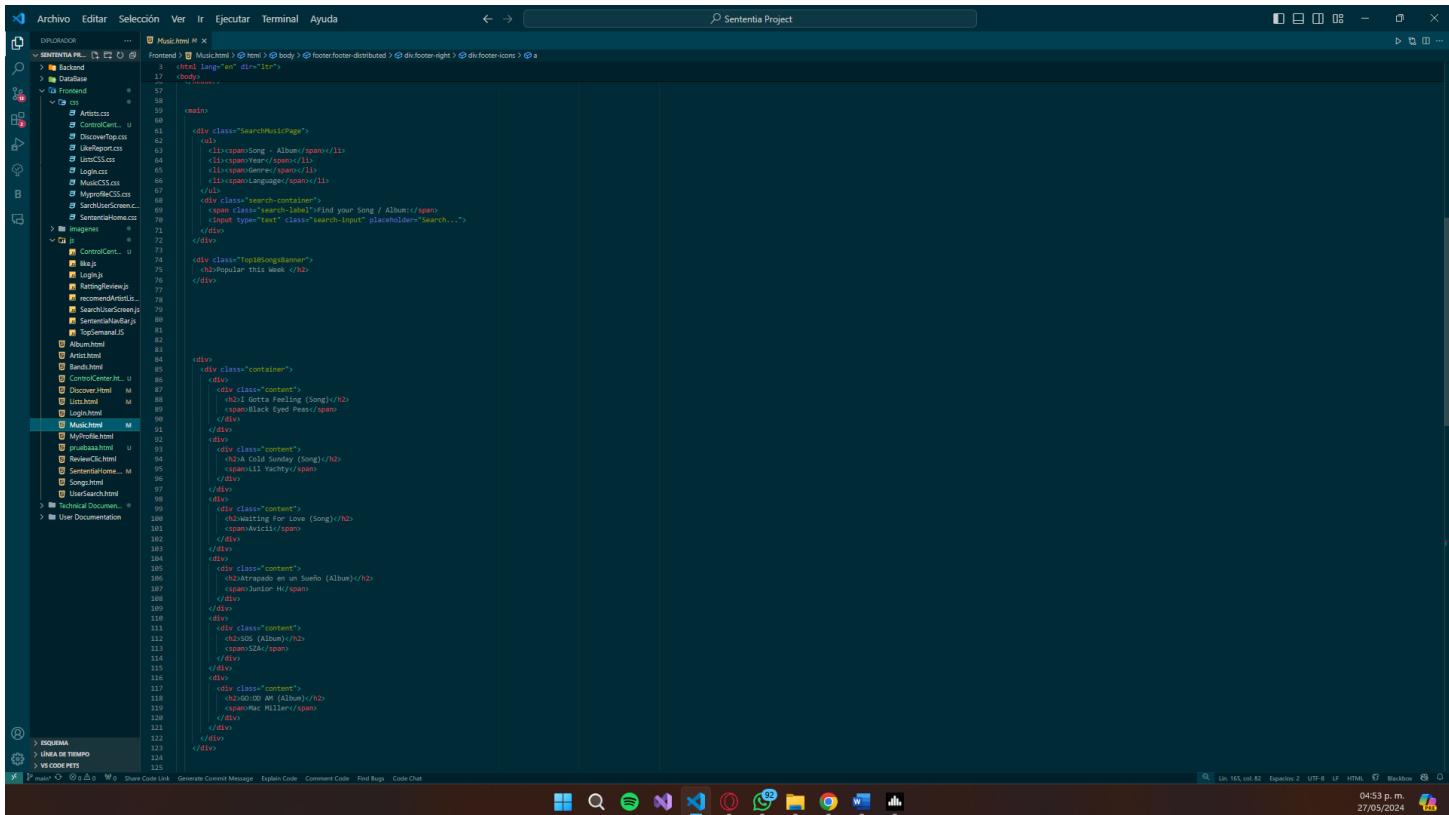
```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="UTF-8">
5     <title>Sententia</title>
6     <link rel="icon" href="images/file-aesthetic-music-fill.ico" type="image/x-icon">
7     <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600;700&family=Ubuntu:wght@400;700;900" rel="stylesheet">
8     <link href="https://unpkg.com/huxicon@0.8.7/css/huxicons.min.css" rel="stylesheet">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    </head>
11    <body>
12      <nav>
13        <div class="nav-bar">
14          <div class="content">
15            <div>Bueno o Malo? RD</div>
16            <div>Instant Crush ft. Corina Smith</div>
17          </div>
18        </div>
19        <div class="content">
20          <div>Morfo</div>
21          <div>Instant Crush</div>
22          <div>Soy un Gafit Punk Ft. Pharrell Williams</div>
23        </div>
24        <div class="content">
25          <div>Paranore</div>
26          <div>All I wanted</div>
27          <div>Paranore</div>
28        </div>
29        <div class="content">
30          <div>Instant Crush</div>
31          <div>All I wanted</div>
32          <div>Paranore</div>
33        </div>
34        <div class="content">
35          <div>Instant Crush</div>
36          <div>All I wanted</div>
37          <div>Paranore</div>
38        </div>
39        <div class="content">
40          <div>Instant Crush</div>
41          <div>All I wanted</div>
42          <div>Paranore</div>
43        </div>
44        <div class="content">
45          <div>Instant Crush</div>
46          <div>All I wanted</div>
47          <div>Paranore</div>
48        </div>
49        <div class="content">
50          <div>Instant Crush</div>
51          <div>All I wanted</div>
52          <div>Paranore</div>
53        </div>
54        <div class="content">
55          <div>Instant Crush</div>
56          <div>All I wanted</div>
57          <div>Paranore</div>
58        </div>
59        <div class="content">
60          <div>Instant Crush</div>
61          <div>All I wanted</div>
62          <div>Paranore</div>
63        </div>
64        <div class="content">
65          <div>Instant Crush</div>
66          <div>All I wanted</div>
67          <div>Paranore</div>
68        </div>
69        <div class="content">
70          <div>Instant Crush</div>
71          <div>All I wanted</div>
72          <div>Paranore</div>
73        </div>
74        <div class="content">
75          <div>Instant Crush</div>
76          <div>All I wanted</div>
77          <div>Paranore</div>
78        </div>
79        <div class="content">
80          <div>Instant Crush</div>
81          <div>All I wanted</div>
82          <div>Paranore</div>
83        </div>
84        <div class="content">
85          <div>Instant Crush</div>
86          <div>All I wanted</div>
87          <div>Paranore</div>
88        </div>
89        <div class="content">
90          <div>Instant Crush</div>
91          <div>All I wanted</div>
92          <div>Paranore</div>
93        </div>
94        <div class="content">
95          <div>Instant Crush</div>
96          <div>All I wanted</div>
97          <div>Paranore</div>
98        </div>
99        <div class="content">
100          <div>Instant Crush</div>
101          <div>All I wanted</div>
102          <div>Paranore</div>
103        </div>
104        <div class="content">
105          <div>Instant Crush</div>
106          <div>All I wanted</div>
107          <div>Paranore</div>
108        </div>
109        <div class="content">
110          <div>Instant Crush</div>
111          <div>All I wanted</div>
112          <div>Paranore</div>
113        </div>
114        <div class="content">
115          <div>Instant Crush</div>
116          <div>All I wanted</div>
117          <div>Paranore</div>
118        </div>
119        <div class="content">
120          <div>Instant Crush</div>
121          <div>All I wanted</div>
122          <div>Paranore</div>
123        </div>
124        <div class="content">
125          <div>Instant Crush</div>
126          <div>All I wanted</div>
127          <div>Paranore</div>
128        </div>
129        <div class="content">
130          <div>Instant Crush</div>
131          <div>All I wanted</div>
132          <div>Paranore</div>
133        </div>
134        <div class="content">
135          <div>Instant Crush</div>
136          <div>All I wanted</div>
137          <div>Paranore</div>
138        </div>
139        <div class="content">
140          <div>Instant Crush</div>
141          <div>All I wanted</div>
142          <div>Paranore</div>
143        </div>
144        <div class="content">
145          <div>Instant Crush</div>
146          <div>All I wanted</div>
147          <div>Paranore</div>
148        </div>
149        <div class="content">
150          <div>Instant Crush</div>
151          <div>All I wanted</div>
152          <div>Paranore</div>
153        </div>
154        <div class="content">
155          <div>Instant Crush</div>
156          <div>All I wanted</div>
157          <div>Paranore</div>
158        </div>
159        <div class="content">
160          <div>Instant Crush</div>
161          <div>All I wanted</div>
162          <div>Paranore</div>
163        </div>
164        <div class="content">
165          <div>Instant Crush</div>
166          <div>All I wanted</div>
167          <div>Paranore</div>
168        </div>
169        <div class="content">
170          <div>Instant Crush</div>
171          <div>All I wanted</div>
172          <div>Paranore</div>
173        </div>
174        <div class="content">
175          <div>Instant Crush</div>
176          <div>All I wanted</div>
177          <div>Paranore</div>
178        </div>
179      </div>
180    </div>
181  </body>
182</html>
```

Lineas 39 col 58 Espacio 2 UTF-8 HTML Blacklist

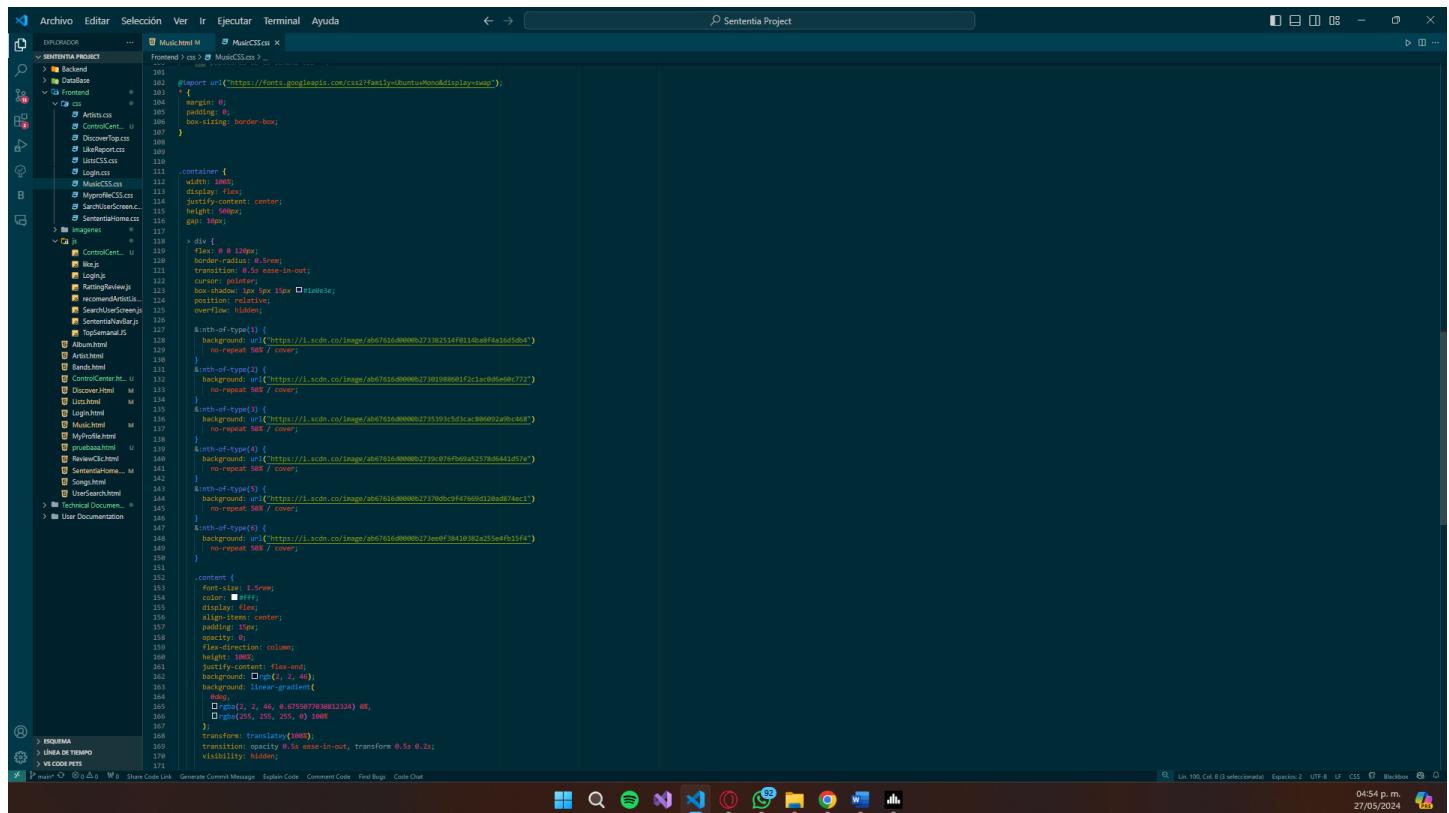
04:49 p. m. 27/05/2024



# Pantalla Music



```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Sentient Project</title>
  </head>
  <body>
    <div class="header">
      <img alt="Sentient Logo" />
      <span>Music</span>
    </div>
    <div class="nav">
      <a href="#">Home
      <a href="#">Artist
      <a href="#">Album
      <a href="#">Song
      <a href="#">Discover
      <a href="#">Profile
      <a href="#">Logout
    </div>
    <div class="content">
      <div class="search-container">
        <input type="text" placeholder="Search..." />
      </div>
      <div class="popular-banner">
        <h2>Popular this week</h2>
        <ul>
          <li>Song A</li>
          <li>Song B</li>
          <li>Song C</li>
        </ul>
      </div>
      <div class="list">
        <ul>
          <li>Song 1</li>
          <li>Song 2</li>
          <li>Song 3</li>
          <li>Song 4</li>
          <li>Song 5</li>
        </ul>
      </div>
    </div>
  </body>
</html>
```



```
@import url("https://fonts.googleapis.com/css?family=Ubuntu+Mono&display=swap");
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.container {
  width: 100%;
  display: flex;
  justify-content: center;
  height: 100px;
  gap: 10px;
}

.container > div {
  flex: 1 1 25px;
  border-radius: 8px;
  transition: 0.5s ease-in-out;
  cursor: pointer;
  border: 2px solid transparent;
  position: relative;
  overflow: hidden;
}

&::before {
  background: url("https://i.scdn.co/image/ab67616d0000002733e2514f014ba08fa316dSubk")
    no-repeat 50% / cover;
}
&::after {
  background: url("https://i.scdn.co/image/ab67616d0000002733e19886bf212a0d0dc772")
    no-repeat 50% / cover;
}
&::nth-of-type(2) {
  background: url("https://i.scdn.co/image/ab67616d0000002733e915d1ac8086072a0c6d8")
    no-repeat 50% / cover;
}
&::nth-of-type(3) {
  background: url("https://i.scdn.co/image/ab67616d0000002733e078f069a2578d644107w")
    no-repeat 50% / cover;
}
&::nth-of-type(4) {
  background: url("https://i.scdn.co/image/ab67616d0000002733e7f76f069a2578d644107w")
    no-repeat 50% / cover;
}
&::nth-of-type(5) {
  background: url("https://i.scdn.co/image/ab67616d0000002733e9f38410382a25e4fb15f4")
    no-repeat 50% / cover;
}

.content {
  font-size: 1.5rem;
  color: #fff;
  display: flex;
  align-items: center;
  justify-content: center;
  opacity: 0;
  flex-direction: column;
  height: 100px;
  justify-content: flex-end;
  background: linear-gradient(
    45deg,
    rgba(2, 46, 46, 0.95507708812324) 65deg,
    #fff 255, 255, 255, 0) 100px;
  transition: opacity 0.5s ease-in-out, transform 0.5s 0.2s;
  visibility: hidden;
}
```

## Pantalla Discover

```
Frontend > Discover.html > Body > Footer/footer-distributed > Footer/footer-right > Div/footer-icons > A

SINTENTA PROJECT ... Discover.html M > Discover.html > Body > Footer/footer-distributed > Footer/footer-right > Div/footer-icons > A

Explorador ... Discover.html M > Discover.html > Body > Footer/footer-distributed > Footer/footer-right > Div/footer-icons > A

Artistas.js ControlCenter_U.js DiscoverRequest.js LikeReports.js ListaCSS.css LogIn.js Music.js MyFriends.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Imágenes ControlCent_U.js Like.js LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos HTML Artist.html Bands.html ControlCenter_HU.js Discover.html LogIn.html Music.html MyFriends.html ReviewClick.html Sentimentalome..._M Songs.html Bruleaa.html ReviewClick.html

Archivos CSS ListaCSS.css LogIn.css MusicCSS.css MyFriendsCSS.css SearchResultsCSS.js SearchResultsScreen.js SemanalesCSS.js TopSemanalesCSS.js

Archivos JS ControlCenter_U.js Like.js LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos Images like.jpg LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos Backend Database

Archivos Documentación Technical Documentation User Documentation

Barra de menú Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

visor de errores > ESQUEMA > LÍNEA DE TIEMPO > VS CODE PETS

visor de salida > muser > D0 > M0 > Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

visor de terminal > 04:58 p.m. 27/05/2024
```

```
Frontend > css > DiscoverTop.css > Page-Info-Content p

SINTENTA PROJECT ... Discover.html M > Discover.html > Body > Footer/footer-distributed > Footer/footer-right > Div/footer-icons > A

Explorador ... Discover.html M > Discover.html > Body > Footer/footer-distributed > Footer/footer-right > Div/footer-icons > A

Artistas.js ControlCenter_U.js DiscoverRequest.js LikeReports.js ListaCSS.css LogIn.js Music.js MyFriends.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Imágenes ControlCent_U.js Like.jpg LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos HTML Artist.html Bands.html ControlCenter_HU.js Discover.html LogIn.html Music.html MyFriends.html ReviewClick.html Sentimentalome..._M Songs.html Bruleaa.html ReviewClick.html

Archivos CSS ListaCSS.css LogIn.css MusicCSS.css MyFriendsCSS.css SearchResultsCSS.js SearchResultsScreen.js SemanalesCSS.js TopSemanalesCSS.js

Archivos JS ControlCenter_U.js Like.js LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos Images like.jpg LogIn.js RatingReview.js Recomendaciones.js SearchResults.js SearchResultsScreen.js Semanales.js TopSemanales.js

Archivos Backend Database

Archivos Documentación Technical Documentation User Documentation

Barra de menú Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

visor de errores > ESQUEMA > LÍNEA DE TIEMPO > VS CODE PETS

visor de salida > 04:58 p.m. 27/05/2024
```

## Pantalla Lists

The screenshot shows a Windows desktop environment with two instances of a code editor open. Both instances have a dark-themed interface.

**Top Window (Code Editor):**

- Title Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Toolbar:** Back, Forward, Refresh, Save, Find, Replace, Select All, Copy, Paste, Cut, Delete, Undo, Redo, Find in Files, Open, Save, Save As, Save All, Close, Minimize, Maximize, Restore, Close.
- Code Area:** Shows the content of the 'List.html' file. The code includes HTML structure like <div>, <ul>, <li>, and <img>. It also includes CSS classes such as 'list-item', 'list-item\_\_img', 'list-item\_\_text', 'list-item\_\_name', 'list-item\_\_desc', 'list-item\_\_link', and 'list-item\_\_icon'. JavaScript functions like 'recommendArtist()' and 'recommendArtistList()' are present.
- Sidebar:** Shows the project structure under 'SENTENTA PROJECT'. It includes 'Frontend' (css, js, images), 'Backend' (Database, Model, Controller, Service, Repository, Model), 'User Documentation' (Index.html, About.html, Contact.html, Privacy.html, Terms.html), and 'Technical Documentation' (Architecture.html, Design.html, Code.html).
- Bottom Bar:** Includes icons for Share Code Link, Generate Commit Message, Explain Code, Comment Code, Find Bugs, and Code Chat.
- Status Bar:** Lin 169 col 90 Espacio: 2 UTF-8 LF HTML Blackbox. 05:03 p.m. 27/05/2024

**Bottom Window (Code Editor):**

- Title Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Toolbar:** Back, Forward, Refresh, Save, Find, Replace, Select All, Copy, Paste, Cut, Delete, Undo, Redo, Find in Files, Open, Save, Save As, Save All, Close, Minimize, Maximize, Restore, Close.
- Code Area:** Shows the content of the 'recommendArtist.js' file. It contains a single function 'recommendArtistList()' which uses jQuery to handle touch events and update a list.
- Sidebar:** Shows the project structure under 'SENTENTA PROJECT'. It includes 'Frontend' (css, js, images), 'Backend' (Database, Model, Controller, Service, Repository, Model), 'User Documentation' (Index.html, About.html, Contact.html, Privacy.html, Terms.html), and 'Technical Documentation' (Architecture.html, Design.html, Code.html).
- Bottom Bar:** Includes icons for Share Code Link, Generate Commit Message, Explain Code, Comment Code, Find Bugs, and Code Chat.
- Status Bar:** Lin 10 col 6 Espacio: 2 UTF-8 LF JavaScript Blackbox. 05:03 p.m. 27/05/2024

## Pantalla User

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="UserSearch.html" rel="stylesheet" type="text/css" />
</head>
<body>
    <header>
        <div class="nav-bar">
            <div class="bx-smenu"></div>
            <div class="logo"><a href="SententiaHome.html">Sententia</a></div>
            <div class="side-bar-logo">
                <span class="log-name">Sententia</span>
                <div class="bx-sbox"></div>
            </div>
            <ul class="limbs">
                <li><a href="ArtistScreen.html">Home</a></li>
                <li><a href="Music.html">Music</a></li>
                <li><a href="Discover.html">Discover</a></li>
                <li><a href="Band.html">Band</a></li>
                <li><a href="Song.html">Song</a></li>
                <li><a href="Login.html">Login</a></li>
                <li><a href="Profile.html">My profile</a></li>
            </ul>
            <div class="more">
                <a href="#">More</a>
                <ul class="sub-menu sub-More">
                    <li><a href="Artist.html">Artist</a></li>
                    <li><a href="Band.html">Band</a></li>
                    <li><a href="Song.html">Song</a></li>
                </ul>
            </div>
        </div>
    </header>
    <div class="search-box">
        <div class="be-search"></div>
        <div class="input-box">
            <input type="text" placeholder="Search..." />
        </div>
    </div>
    <div class="SearchUserBar">
        <form action="">
            <input type="text" search required="required" />
            <i class="fa fa-search"></i>
            <button onclick="ClearInput()">Clear</button>
        </form>
    </div>
    <div class="main">
        <br><br><br><br>
        <br><br><br><br>
        <br><br><br><br>
    </div>
    <div class="Footer">
        <script src="js/SententiaAvatar.js"></script>
        <script src="js/SearchUserScreen.js"></script>
    </div>
</body>

```

```
.SearchUserBar {
    position: relative;
    width: 100px;
    margin: auto;
}
.SearchUserBar input {
    border: none;
    border-bottom: 1px solid #ccc;
    padding: 5px;
    width: 100px;
    height: 30px;
    outline: none;
    font-size: 14px;
}
.SearchUserBar i {
    position: absolute;
    right: -10px;
    top: 50%;
    color: #ccc;
    font-size: 16px;
    transform: rotate(-45deg);
}
.SearchUserBar button {
    border: 1px solid #ccc;
    border-radius: 4px;
    background-color: #fff;
    width: 30px;
    height: 30px;
    position: absolute;
    left: 0;
    top: 0;
    outline: none;
    cursor: pointer;
}
.SearchUserBar button::before {
    content: "Search";
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%);
    color: #ccc;
    font-size: 12px;
}
.SearchUserBar button::after {
    content: "";
    position: absolute;
    left: 0;
    top: 0;
    width: 100px;
    height: 100px;
    background: linear-gradient(45deg, transparent 45%, #ccc 45%, #ccc 55%, transparent 55%);
    border-radius: 50% 50% 0 0;
    transform: rotate(-45deg);
}
.SearchUserBar .SearchUserBar-- {
    align-items: center;
}
```

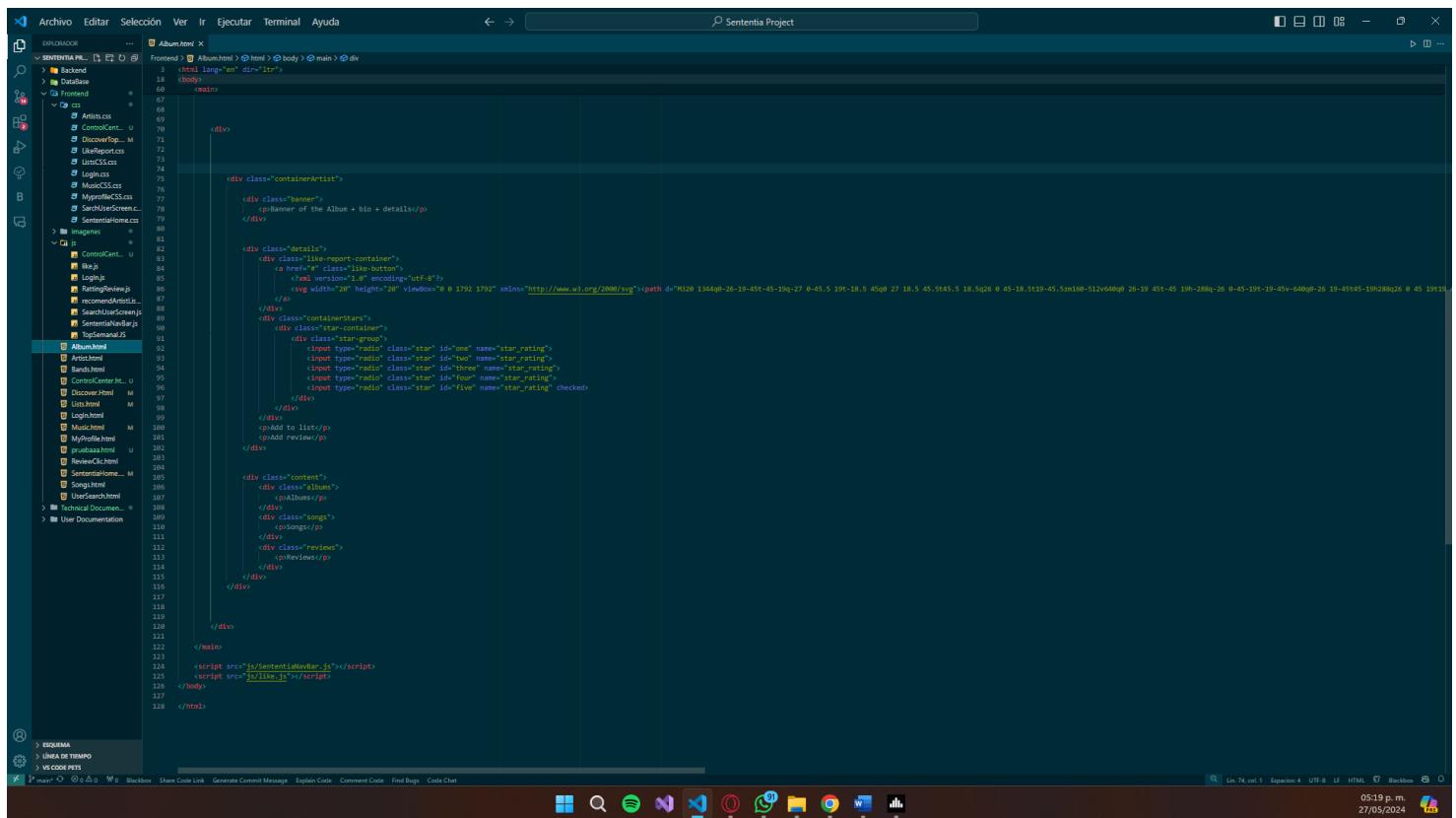
## Pantalla Login

The screenshot shows the Blackbox IDE interface with the following details:

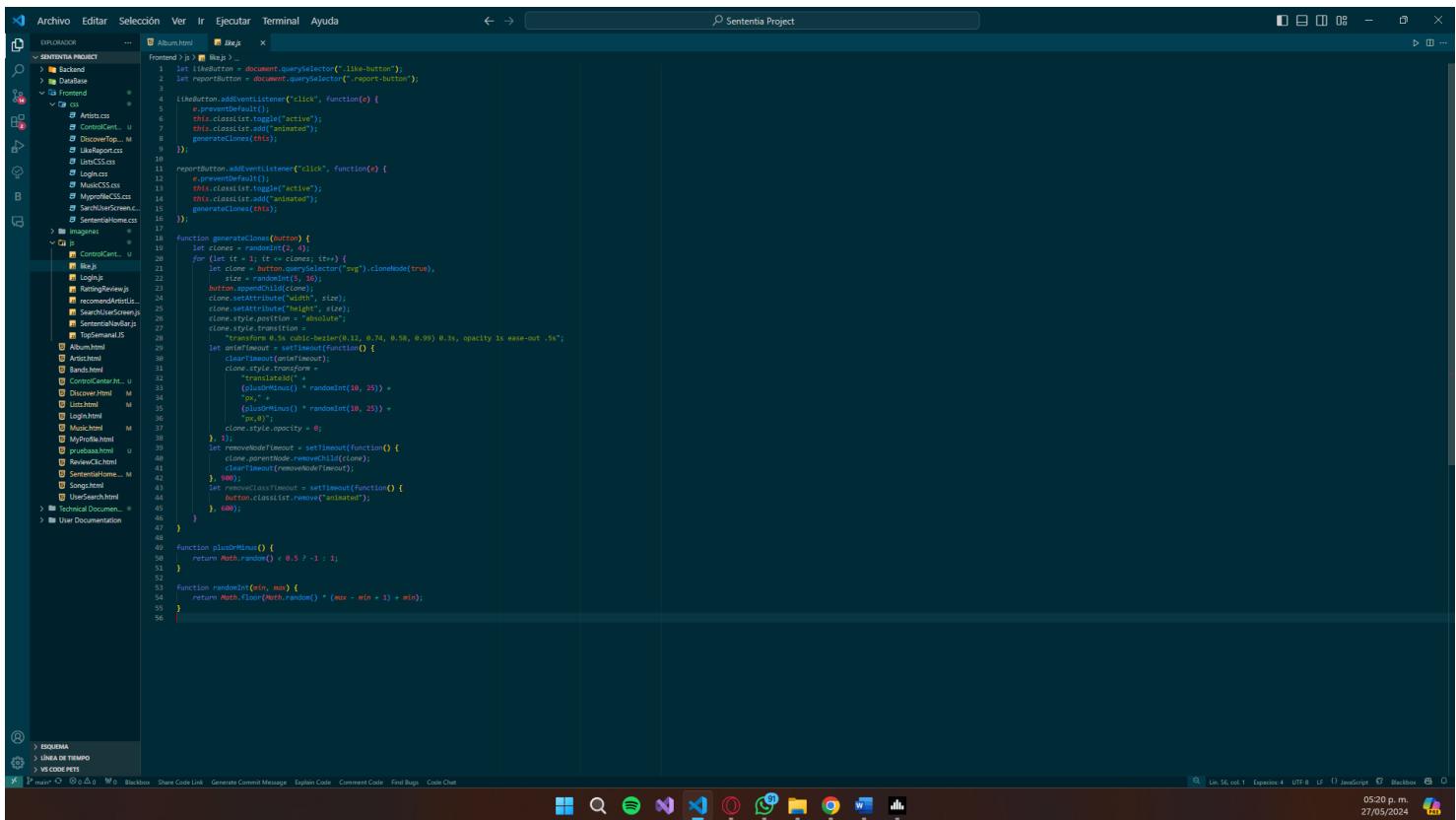
- Toolbar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Search Bar:** Sentient Project
- Left Sidebar (Explorador):**
  - SENTIENT PROJECT
  - Backend
  - Database
  - Frontend
    - Artists
    - ControlCent\_U
    - DiscoverTop\_M
    - LikeReport
    - UltiCSS.css
    - Login.css
    - MusicCSS.css
    - MyProfileCSS.css
    - ScheduleScreen\_c
    - SentientHome.css
  - Imagenes
  - js
    - ControlCent\_U
    - Index
    - Login.js
    - RatingReview.js
    - recommendation.js
    - SearchUserScreen.js
    - SentientNavBar.js
    - TopGamedJS
    - Album.html
    - Artist.html
    - Background
    - ControlCenter.html\_U
    - Discover.html\_M
    - Usc.html\_M
    - Login.html
    - Music.html\_M
    - MyProfile.html
    - Profile.html
    - ReviewClic.html
    - SentientHome\_M
    - Song.html
    - UserSearch.html
  - Technical Document.
  - User Documentation
- Code Editor:** The main area displays the content of the `Login.js` file. The code includes HTML structure for a login form, CSS classes for styling, and JavaScript for user interaction. It also includes imports for `sentientNavbar.js` and `login.js`.
- Bottom Bar:** Includes icons for Blackbox, Share Code Link, Generate Commit Message, Explain Code, Comment Code, Find Bugs, Code Chat, and various system status indicators.
- Bottom Right:** Status bar showing line 104, column 22, expansion 2, UFT-8, LF-HTML, and build time 07/05/2024.

## Pantalla My Profile

## Pantalla Artists – Band – Song – Album



```
<html lang="en" dir="ltr">
  <head>
    ...
  </head>
  <body>
    <main>
      <div>
        <div class="containerArtist">
          <div class="banner">
            <p>Banner of the Album + bio + details</p>
          </div>
          <div class="details">
            <div class="like-report-container">
              <a href="#" class="like-button">
                <img alt="Like button icon" data-bbox="178 138 208 158" />
                <small>Like</small>
              </a>
              <div style="display: flex; justify-content: space-between; width: 100px; margin-top: 5px;">
                <small>0</small>
                <small>1792 1792</small>
                <small>http://www.wl.org/2000/svg</small>
                <small><path d="M16 1344q-26-19-45-45-19q-27 0-45.5 19t-18.5 45q27 26 18.5 45.5 18.5q26 0 45-19.5t33-45.5m108-512v640q0 26 19-45t45-19q288z/></path>
              </div>
            </div>
            <div class="containerRating">
              <div class="star-rating">
                <input type="radio" class="star" id="one" name="star_rating">
                <input type="radio" class="star" id="two" name="star_rating">
                <input type="radio" class="star" id="three" name="star_rating">
                <input type="radio" class="star" id="four" name="star_rating">
                <input type="radio" class="star" id="five" name="star_rating" checked="" />
              </div>
            </div>
            <div class="content">
              <div class="albums">
                <p>Albuns:</p>
                <ul>
                  ...
                </ul>
              </div>
              <div class="songs">
                <p>Songs:</p>
                <ul>
                  ...
                </ul>
              </div>
              <div class="reviews">
                <p>Reviews:</p>
                <ul>
                  ...
                </ul>
              </div>
            </div>
          </div>
        </div>
      </main>
      <script src="js/SentientialBar.js"></script>
      <script src="js/like.js"></script>
    </body>
  </html>
```



```
Frontend>js B.js
Frontend>j B.js
1 let likeButton = document.querySelector('.like-button');
2 let reportButton = document.querySelector('.report-button');
3
4 likeButton.addEventListener('click', function(e) {
5   e.preventDefault();
6   this.classList.toggle('active');
7   this.classList.toggle('disabled');
8   generateClone(this);
9 });
10
11 reportButton.addEventListener('click', function(e) {
12   e.preventDefault();
13   this.classList.toggle('active');
14   this.classList.toggle('disabled');
15   generateClone(this);
16 });
17
18 function generateClone(button) {
19   let clones = randomInt(2, 4);
20   for (let i = 1; i < clones; i++) {
21     let clone = button.cloneNode(true);
22     size = randomInt(5, 16);
23     button.appendChild(clone);
24     clone.style.height = size + 'px';
25     clone.style.width = size + 'px';
26     clone.style.position = 'absolute';
27     clone.style.transition = 'all 0.5s ease-in-out';
28     clone.style.outline = 'none';
29     let animation = settimeout(function() {
30       clearTimeout(animation);
31       clone.style.transform =
32         'translate3d(0, 0, 0)';
33       clone.style.opacity =
34         (randomInt() * randomInt(10, 25)) +
35         'px';
36       clone.style.height =
37         (randomInt() * randomInt(10, 25)) +
38         'px';
39     }, 3000);
40     let removeClone = settimeout(function() {
41       clone.parentNode.removeChild(clone);
42       clearTimeout(removeClone);
43     }, 9000);
44     let removeClass = settimeout(function() {
45       if (button.classList.remove('animated')) {
46         button.classList.add('animating');
47       }
48     }, 6000);
49   }
50 }
51
52 function randomInt(min, max) {
53   return Math.floor(Math.random() * (max - min + 1)) + min;
54 }
55 }
```

Archieve Editar Selección Ver Ir Ejecutar Terminal Ayuda

SENTENTIA PROJECT

Frontend.css > Artista.css > containerArtist.content > div

```
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #E7CFD6;
4 }
5
6 .containerArtist {
7   width: 800px;
8   margin: 20px auto;
9   padding: 10px;
10  flex-direction: column;
11  gap: 20px;
12 }
13
14 .containerArtist .header {
15   background-color: #E6F2FF;
16   padding: 20px;
17   text-align: center;
18   justify-content: center;
19   border-radius: 10px 10px 0 0; /* Redondear las esquinas superiores */
20 }
21 .containerArtist .details {
22   background-color: #FFF;
23   padding: 20px;
24   display: flex;
25   flex-direction: row;
26   justify-content: center;
27   align-items: center;
28   gap: 10px;
29 }
30 .width: 80px;
31 margin-right: 20px;
32 border-radius: 10px; /* Redondear las esquinas */
33
34 .containerArtist .content > div {
35   background-color: #FDD;
36   padding: 20px;
37   border-radius: 10px; /* Redondear las esquinas */
38 }
39
40 .containerArtist .details .like-report-container {
41   align-self: center;
42 }
43 .containerArtist .details .containerStars {
44   align-self: center;
45 }
46
47 .containerArtist .details > {
48   align-self: center;
49 }
50
51 .containerArtist .details {
52   gap: 10px;
53 }
54
55 .containerArtist .content {
56   display: flex;
57   flex-direction: column;
58   gap: 20px;
59 }
60
61 .containerArtist .content > div {
62   background-color: #FFF;
63   padding: 20px;
64   border-radius: 10px; /* Redondear las esquinas */
65 }
```

Lín. 66, col. 2 Espacio: 4 LF CSS Blackbox

ESQUEMA  
LÍNEA DE TIEMPO  
VIS CODEPETS

main O Backbox Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

05:20 p.m.  
27/05/2024

## Pantallas Backend

- Se uso un servicio API REST creado con .NET y controladores
- Se trabajo con un arquitectura en capas
  - presentación
  - Negocio
  - Acceso datos
  - Modelos
- Se utilizo SQL Server como BD
- Se utilizo Entity Framework como ORM

## Motivos

- Se utilizo EF debido a la necesidad de agilidad en el desarrollo y para tener vigencia en el desarrollo, alusar tecnologias de actualidad (en este ORMs)
- Se utilizo una arquitectura en de 3 capaz para darle una

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
    <InvariantGlobalization>false</InvariantGlobalization>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="AutoMapper" Version="13.0.1" />
    <PackageReference Include="AutoMapper.Extensions.Microsoft.DependencyInjection" Version="12.8.0" />
    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="8.0.5" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="8.0.5" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="8.0.5" />
    <PrivateAssets>all</PrivateAssets>
    <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
  </ItemGroup>
  <ItemGroup>
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
  </ItemGroup>
  <ItemGroup>
    <ProjectReference Include="..\Business\Business.csproj" />
  </ItemGroup>
</Project>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "BACKEND". The selected item is "API\_Sententia.csproj.user". Other files visible include "API\_Sententia.csproj", "API\_Sententia.http", "API\_Sententia.sln", "appsettings.Development.json", "Program.cs", and "pruebaIGNORAR.py".
- Code Editor (Center):** Displays the XML content of the "API\_Sententia.csproj.user" file.
- Bottom Status Bar:** Shows "main" as the active tab, file statistics (0△0 ⏴ 0), and various status icons like Share Code Link, Generate Commit Message, Explain Code, Comment Code, Find Bugs, and Code Chat.
- Bottom Right:** Includes a timestamp "11:04 p. m. 27/05/2024" and a "Code Chat" icon.

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="Current" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <ActiveDebugProfile>https://ActiveDebugProfile</ActiveDebugProfile>
    <Controller_SelectedScaffolderID>MvcControllerEmptyScaffolder</Controller_SelectedScaffolderID>
    <Controller_SelectedScaffolderCategoryPath>root/Common/MVC/Controller</Controller_SelectedScaffolderCategoryPath>
  </PropertyGroup>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "BACKEND". The selected item is "appsettings.Development.json". Other files visible include "API\_Sententia.csproj", "API\_Sententia.http", "API\_Sententia.sln", "Program.cs", and "pruebaIGNORAR.py".
- Code Editor (Center):** Displays the JSON content of the "appsettings.Development.json" file.
- Bottom Status Bar:** Shows "main" as the active tab, file statistics (0△0 ⏴ 0), and various status icons like Share Code Link, Generate Commit Message, Explain Code, Comment Code, Find Bugs, and Code Chat.
- Bottom Right:** Includes a timestamp "11:04 p. m. 27/05/2024" and a "Code Chat" icon.

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  }
}
```

A tooltip message is displayed in the bottom right corner: "Desea instalar la 'C# Dev Kit' extensión de Microsoft, recomendada para API\_Sententia.sln?". It includes "Instalar" and "Mostrar recomendaciones" buttons.

Backend

```
API_Sententia > { } appsettings.json > ...
```

```
1 {
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionString": {
    "DefaultConnection": "Data Source=LAPTOP-026F53PT\\SQLEXPRESS;Initial Catalog=MusicalReviews;Integrated Security=True;Trust Server Certificate=True"
  }
}
```

Program.cs

Business  
DataAccess  
Models  
pruebalGNORAR.py

ESQUEMA  
LÍNEA DE TIEMPO  
VS CODE PETS

main\* 0 0 0 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lín. 1, col. 1 Espacios: 2 UTF-8 CRLF JSON Code Chat

11:04 p. m.  
27/05/2024

Backend

```
Program.cs > Extensión: C# Dev Kit
```

```
API_Sententia > Program.cs
```

```
1 namespace API_Sententia
2 {
3     public class Program
4     {
5         public static void Main(string[] args)
6         {
7             var builder = WebApplication.CreateBuilder(args);
8
9             // Add services to the container.
10
11             builder.Services.AddControllers();
12             // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
13             builder.Services.AddEndpointsApiExplorer();
14             builder.Services.AddSwaggerGen();
15             //builder.Services.AddAutoMapper(typeof(Program).Assembly);
16
17             var app = builder.Build();
18
19             // Configure the HTTP request pipeline.
20             if (app.Environment.IsDevelopment())
21             {
22                 app.UseSwagger();
23                 app.UseSwaggerUI();
24             }
25
26             app.UseHttpsRedirection();
27
28             app.UseAuthorization();
29
30             app.MapControllers();
31
32             app.Run();
33
34         }
35     }
36 }
37 }
```

Program.cs

Business  
DataAccess  
Models  
pruebalGNORAR.py

ESQUEMA  
LÍNEA DE TIEMPO  
VS CODE PETS

main\* 0 0 0 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lín. 1, col. 1 Espacios: 4 UTF-8 CRLF C# Code Chat

11:04 p. m.  
27/05/2024

Backend

```
GenreController.cs
```

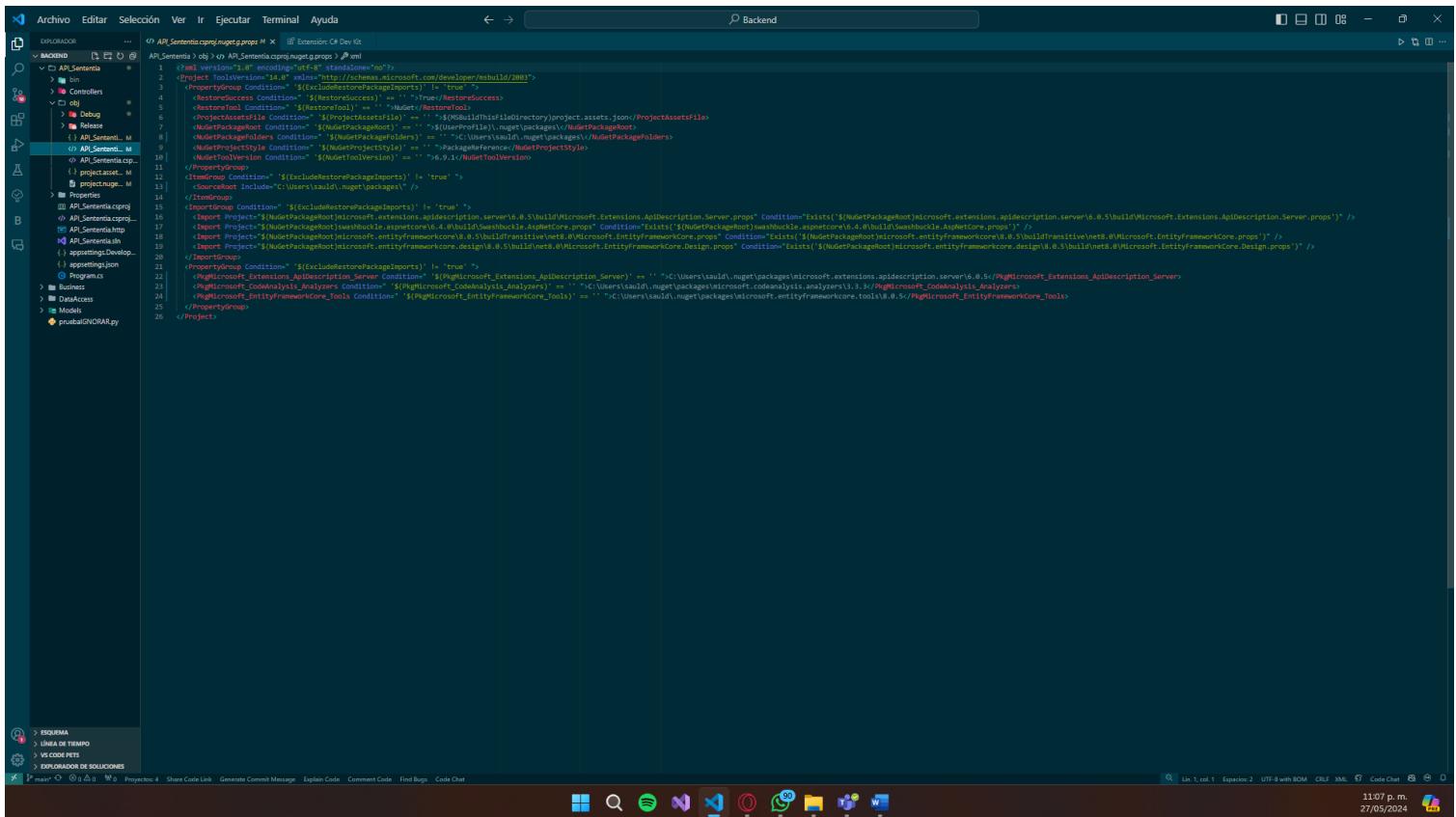
```
1 using Business;
2 using Microsoft.AspNetCore.Mvc;
3 using Models.Genre;
4 
5 namespace API_Sentencia.Controllers
6 {
7     [Route("api/[controller]")]
8     [ApiController]
9     public class GenreController : Controller
10    {
11        private readonly GenreService _genreService;
12 
13        public GenreController(IConfiguration config)
14        {
15            _genreService = new GenreService(config);
16        }
17 
18        [HttpPost]
19        public async Task<ActionResult<GenrePost_Response>> PostGenre([FromBody] GenrePost_Request genre)
20        {
21            try
22            {
23                var genreCreated = await _genreService.PostAsync(genre);
24 
25                if (genreCreated == null)
26                {
27                    return NotFound();
28                }
29                return Ok(genreCreated);
30            }
31            catch (Exception e)
32            {
33 #if DEBUG
34                return StatusCode(500, new { message = e.Message });
35            }
36            else
37                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
38            #endif
39        }
40 
41        [HttpGet]
42        public async Task<ActionResult<List<GenreGet_Response>>> GetGenresAsync()
43        {
44            try
45            {
46                var genres = await _genreService.GetGenresAsync();
47 
48                if (genres == null)
49                {
50                    return NotFound();
51                }
52                return Ok(genres);
53            }
54            catch (Exception e)
55            {
56 #if DEBUG
57                return StatusCode(500, new { message = e.Message });
58            }
59            else
60                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
61            #endif
62        }
63 
64        //get by id
65        [HttpGet("{id}")]
66        public async Task<ActionResult<GenreGet_Response>> GetGenreById(int id,[FromQuery] GenreGetById_Request request)
67        {
68            try
69            {
70                var genre = await _genreService.GetByIdAsync(id);
71 
72                if (genre == null)
73                {
74                    return NotFound();
75                }
76                return Ok(genre);
77            }
78            catch (Exception e)
79            {
80 #if DEBUG
81                return StatusCode(500, new { message = e.Message });
82            }
83            else
84                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
85            #endif
86        }
87    }
88 }
```

Backend

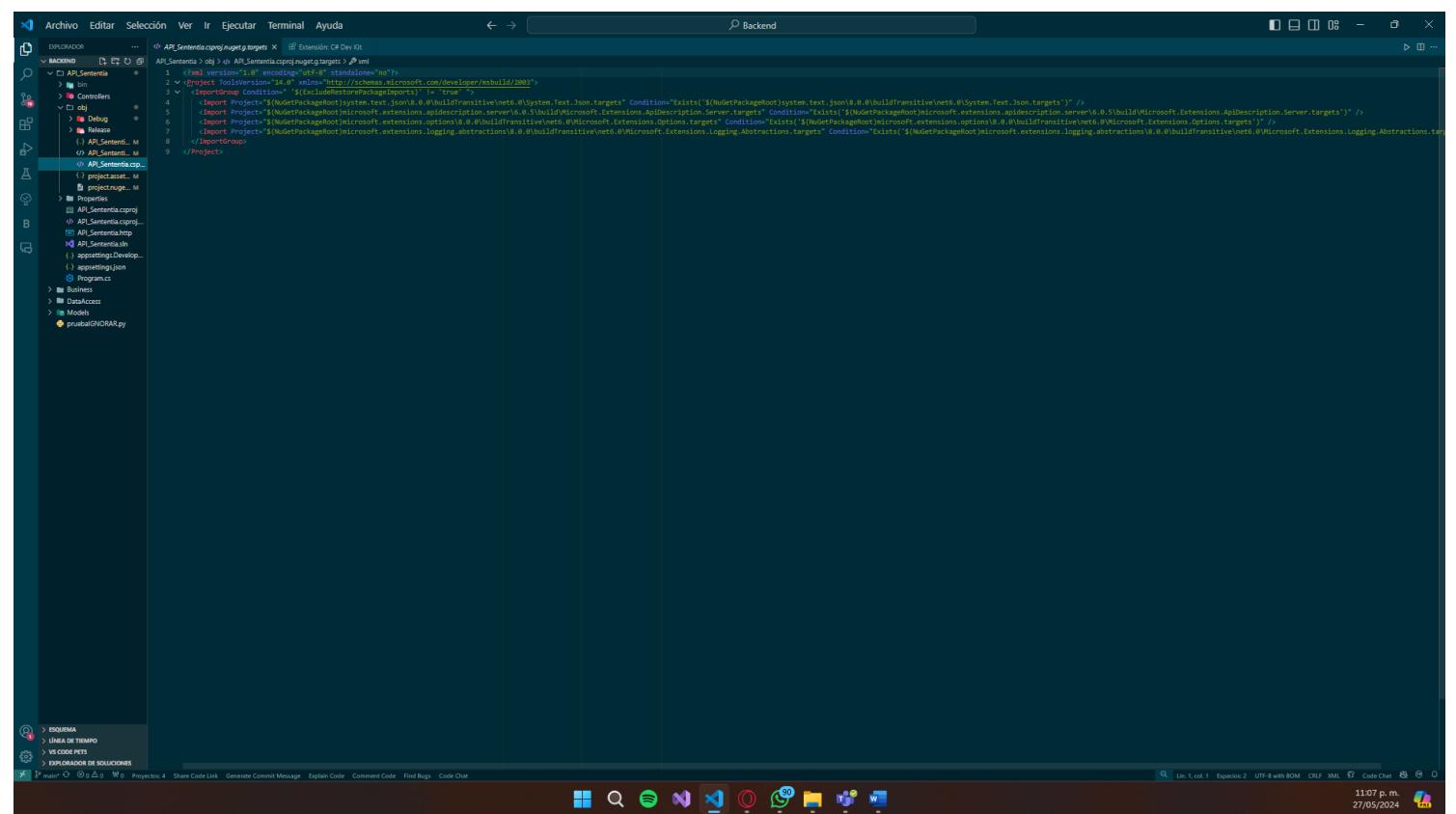
```
LanguageController.cs
```

```
1 using Business;
2 using Microsoft.AspNetCore.Mvc;
3 using Models.Language;
4 
5 namespace API_Sentencia.Controllers
6 {
7     [Route("api/[controller]")]
8     [ApiController]
9     public class LanguageController : Controller
10    {
11        private readonly LanguageService _languageService;
12 
13        public LanguageController(IConfiguration config)
14        {
15            _languageService = new LanguageService(config);
16        }
17 
18        [HttpPost]
19        public async Task<ActionResult<LanguagePost_Response>> PostLanguage([FromBody] LanguagePost_Request request)
20        {
21            try
22            {
23                var languageCreated = await _languageService.PostLanguage(request);
24 
25                if (languageCreated == null)
26                {
27                    return NotFound();
28                }
29                return Ok(languageCreated);
30            }
31            catch (Exception e)
32            {
33 #if DEBUG
34                return StatusCode(500, new { message = e.Message });
35            }
36            else
37                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
38            #endif
39        }
40 
41        [HttpGet]
42        public async Task<ActionResult<LanguageGetAll_Response>> GetAllLanguages()
43        {
44            try
45            {
46                var response = await _languageService.GetLanguages();
47 
48                if (response == null)
49                {
50                    return NotFound();
51                }
52                return Ok(response);
53            }
54            catch (Exception e)
55            {
56 #if DEBUG
57                return StatusCode(500, new { message = e.Message });
58            }
59            else
60                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
61            #endif
62        }
63 
64        [HttpGet("{id}")]
65        public async Task<ActionResult<Language GetById_Response>> GetLanguageById(int id)
66        {
67            try
68            {
69                var language = await _languageService.GetByIdAsync(id);
70 
71                if (language == null)
72                {
73                    return NotFound();
74                }
75                return Ok(language);
76            }
77            catch (Exception e)
78            {
79 #if DEBUG
80                return StatusCode(500, new { message = e.Message });
81            }
82            else
83                return StatusCode(500, new { message = "An error occurred while processing the request, report to the mail in the footer" });
84            #endif
85        }
86    }
87 }
```





```
1 <?xml version="1.0" encoding="utf-8" standalone="no"?>
2 <Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <PropertyGroup Condition=" '$(RestoreTool)' == '' && '$(RestoreTool)' != '$(MSBuildThisFile)'>
4     <RestoreTool Condition=" '$(RestoreTool)' == '' >MSBuild</RestoreTool>
5     <ProjectAssetsFile Condition=" '$(ProjectAssetsFile)' == '' >$(MSBuildThisFile).project.assets.json</ProjectAssetsFile>
6     <MSBuildProjectStyle Condition=" '$(MSBuildProjectStyle)' == '' >$(UserProjectStyle)</MSBuildProjectStyle>
7     <MSBuildPackageFolders Condition=" '$(MSBuildPackageFolders)' == '' >$(UserPackageFolders)</MSBuildPackageFolders>
8     <MSBuildToolVersion Condition=" '$(MSBuildToolVersion)' == '' >$(UserToolVersion)</MSBuildToolVersion>
9     <ItemGroup Condition=" '$(ExcludeRestorePackageImports)' != 'true' >
10       <SourceCode Include="C:\Users\sauld\nguy\packages\>
11       <ImportGroup Condition=" '$(ExcludeRestorePackageImports)' != 'true' >
12         <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.Extensions.ApiDescription.Server.props" Condition="Exists('$(NuGetPackageRoot)\\microsoft.extensions.apidescription.server\\6.0.5\\build\\Microsoft.Extensions.ApiDescription.Server.props')"/>
13         <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.EntityFrameworkCore.Design\\EntityFrameworkCore\\6.0.5\\build\\Microsoft.EntityFrameworkCore.EntityFrameworkCore.props" Condition="Exists('$(NuGetPackageRoot)\\microsoft.entityframeworkcore\\6.0.5\\build\\Microsoft.EntityFrameworkCore.EntityFrameworkCore.props')"/>
14         <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.EntityFrameworkCore\\EntityFrameworkCore\\6.0.5\\build\\Microsoft.EntityFrameworkCore.EntityFrameworkCore.Design.props" Condition="Exists('$(NuGetPackageRoot)\\microsoft.entityframeworkcore.design\\6.0.5\\build\\Microsoft.EntityFrameworkCore.Design.props')"/>
15       </ImportGroup>
16     <PropertyGroup Condition=" '$(ExcludeRestorePackageImports)' != 'true' >
17       <ImportGroup Condition=" '$(PkgMicrosoft_Authentication_ApiDescription_Server)' != '' >$(UserToolPath)\\microsoft.extensions.apidescription.server\\6.0.5\\PkgMicrosoft_extensions_ApiDescription_Server>
18       <ImportGroup Condition=" '$(PkgMicrosoft_CodeAnalysis_Analyzers)' != '' >$(UserToolPath)\\microsoft.codeanalysis.analyzers\\3.3.3\\PkgMicrosoft_CodeAnalysis_Analyzers>
19       <ImportGroup Condition=" '$(PkgMicrosoft_EntityFrameworkCore_Tools)' != '' >$(UserToolPath)\\microsoft.entityframeworkcore.tools\\8.0.5\\PkgMicrosoft_EntityFrameworkCore_Tools>
20     </PropertyGroup>
21   </PropertyGroup>
22   <ImportGroup Condition=" '$(ExcludeRestorePackageImports)' != 'true' >
23     <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.Extensions.Logging.Abstractions\\System.Text.Json\\8.0.0\\build\\Microsoft.Extensions.Logging.Abstractions.targets" Condition="Exists('$(NuGetPackageRoot)\\system.text.json\\8.0.0\\build\\Microsoft.Extensions.Logging.Abstractions.targets')"/>
24   </ImportGroup>
25 </Project>
```



```
1 <?xml version="1.0" encoding="utf-8" standalone="no"?>
2 <Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.Extensions.ApiDescription.Server.targets" Condition="Exists('$(NuGetPackageRoot)\\microsoft.extensions.apidescription.server\\6.0.5\\build\\Microsoft.Extensions.ApiDescription.Server.targets')"/>
4   <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.Extensions.Options\\System.Text.Json\\8.0.0\\build\\Microsoft.Extensions.Options.targets" Condition="Exists('$(NuGetPackageRoot)\\system.text.json\\8.0.0\\build\\Microsoft.Extensions.Options.targets')"/>
5   <Import Project="$(NuGetPackageRoot)\msbuild\Microsoft.Extensions.Logging.Abstractions\\System.Text.Json\\8.0.0\\build\\Microsoft.Extensions.Logging.Abstractions.targets" Condition="Exists('$(NuGetPackageRoot)\\system.text.json\\8.0.0\\build\\Microsoft.Extensions.Logging.Abstractions.targets')"/>
6 </Project>
```

Backend

```
1 {
  "version": 3,
  "targets": {
    "net8.0": {
      "dependencies": {
        "Microsoft.Extensions.Options": "6.0.0"
      },
      "compile": {
        "lib\net8.0\AutoMapper.dll": {
          "related": ".xml"
        }
      },
      "runtime": {
        "lib\net8.0\AutoMapper.dll": {
          "related": ".xml"
        }
      }
    },
    "AutoMapper.Extensions.Microsoft.DependencyInjection": {
      "type": "package",
      "dependencies": {
        "AutoMapper": "6.0.0",
        "Microsoft.Extensions.Options": "6.0.0"
      },
      "compile": {
        "lib\netstandard2.1\AutoMapper.Extensions.Microsoft.DependencyInjection.dll": {}
      },
      "runtime": {
        "lib\netstandard2.1\AutoMapper.Extensions.Microsoft.DependencyInjection.dll": {}
      }
    },
    "Azure.Core": {
      "type": "package",
      "dependencies": {
        "Microsoft.Acti.AsyncInterfaces": "1.1.1",
        "System.Diagnostics.DiagnosticSource": "5.0.1",
        "System.Memory": "4.0.1",
        "System.Numerics.Vectors": "4.5.0",
        "System.Text.Encoding": "4.2.2",
        "System.Text.Json": "4.7.2",
        "System.Threading.Tasks.Extensions": "4.5.4"
      },
      "compile": {
        "lib\net6.0\Azure.Core.dll": {
          "related": ".xml"
        }
      },
      "runtime": {
        "lib\net6.0\Azure.Core.dll": {
          "related": ".xml"
        }
      }
    },
    "Azure.Identity": {
      "type": "package",
      "dependencies": {
        "Azure.Identity.Client": "4.56.0",
        "Microsoft.Identity.Client.Extensions.Http": "4.56.0",
        "System.Memory": "4.5.4",
        "System.Security.Cryptography.ProtectedData": "4.7.0",
        "System.Text.Json": "4.7.2",
        "System.Threading.Tasks.Extensions": "4.5.4"
      },
      "compile": {
        "lib\netstandard2.0\Azure.Identity.dll": {
          "related": ".xml"
        }
      }
    }
  }
}
```

ESQUERDA LÍNEA DE TIEMPO VS CODE PETS EXPLORADOR DE SOLUCIONES

Proyectos 4 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lin. 1, col. 1 Espacio 2 UTF-8 CR LF JSON Code Chat

11:07 p.m. 27/05/2024

Backend

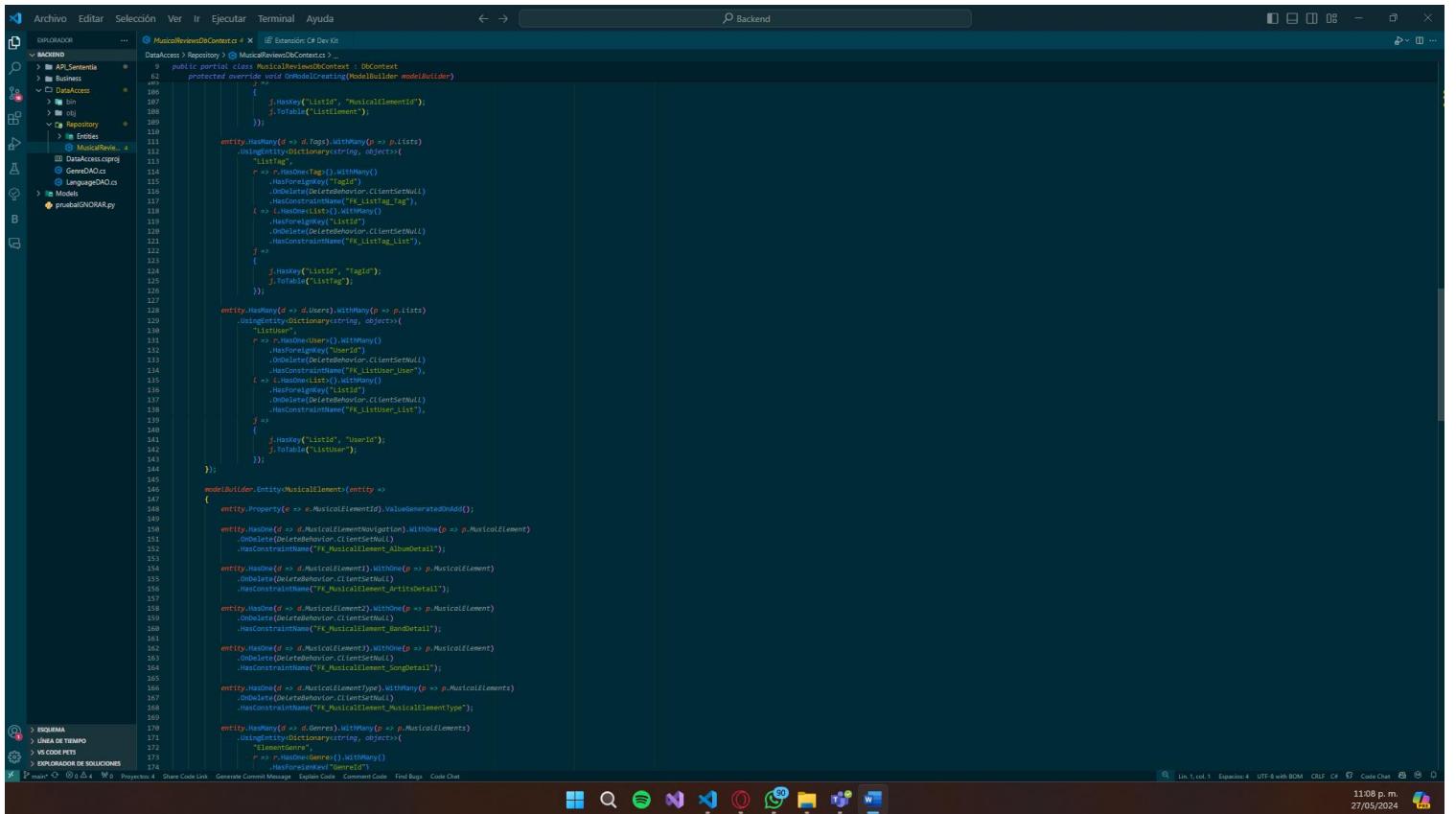
```
1 using System;
2 using System.Collections.Generic;
3 using Database.Repository.Entities;
4 using Microsoft.Extensions.Configuration;
5 using Microsoft.Extensions.DependencyInjection;
6 
7 namespace Database.Repository;
8 
9 public partial class MusicalReviewsDbContext : DbContext
10 {
11     private readonly IConfiguration _config;
12     private readonly string _connectionString;
13 
14     public MusicalReviewsDbContext(IConfiguration config)
15     {
16         _config = config;
17         _connectionString = config.GetConnectionString("DefaultConnection");
18     }
19 
20     public MusicalReviewsDbContext(DbContextOptions<MusicalReviewsDbContext> options)
21     {
22         base(options);
23     }
24 
25     public virtual DbSet<AlbumDetail> AlbumDetails { get; set; }
26 
27     public virtual DbSet<ArtistDetail> ArtistDetails { get; set; }
28 
29     public virtual DbSet<BandDetail> BandDetails { get; set; }
30 
31     public virtual DbSet<Genre> Genres { get; set; }
32 
33     public virtual DbSet<Language> Languages { get; set; }
34 
35     public virtual DbSet<Link> Links { get; set; }
36 
37     public virtual DbSet<List> Lists { get; set; }
38 
39     public virtual DbSet<MusicalElement> MusicalElements { get; set; }
40 
41     public virtual DbSet<MusicalElementType> MusicalElementTypes { get; set; }
42 
43     public virtual DbSet<Platform> Platforms { get; set; }
44 
45     public virtual DbSet<Report> Reports { get; set; }
46 
47     public virtual DbSet<ReportReason> ReportReasons { get; set; }
48 
49     public virtual DbSet<Review> Reviews { get; set; }
50 
51     public virtual DbSet<Role> Roles { get; set; }
52 
53     public virtual DbSet<SongDetail> SongDetails { get; set; }
54 }
```

ESQUERDA LÍNEA DE TIEMPO VS CODE PETS EXPLORADOR DE SOLUCIONES

Proyectos 4 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lin. 1, col. 1 Espacio 4 UTF-8 with BOM CR LF JSON Code Chat

11:08 p.m. 27/05/2024



```
public partial class MusicalReviewsDBContext : DbContext
{
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<MusicalReview>()
            .HasKey(e => e.Id);
        modelBuilder.Entity<MusicalElement>()
            .HasKey(e => e.Id);
        modelBuilder.Entity<User>()
            .HasKey(u => u.Id);
        modelBuilder.Entity<Review>()
            .HasKey(r => r.Id);

        modelBuilder.Entity<MusicalReview>()
            .HasMany(d => d.Tags).WithMany(p => p.Lists)
            .UsingEntity<Dictionary<string, object>>(
                e => e.Name,
                e => e.HasOne(e => e.Tag),
                e => e.HasOne<List>(e => e.List)
            );
        modelBuilder.Entity<List>()
            .HasMany(e => e.Tags);
        modelBuilder.Entity<Tag>()
            .HasMany(e => e.Lists);

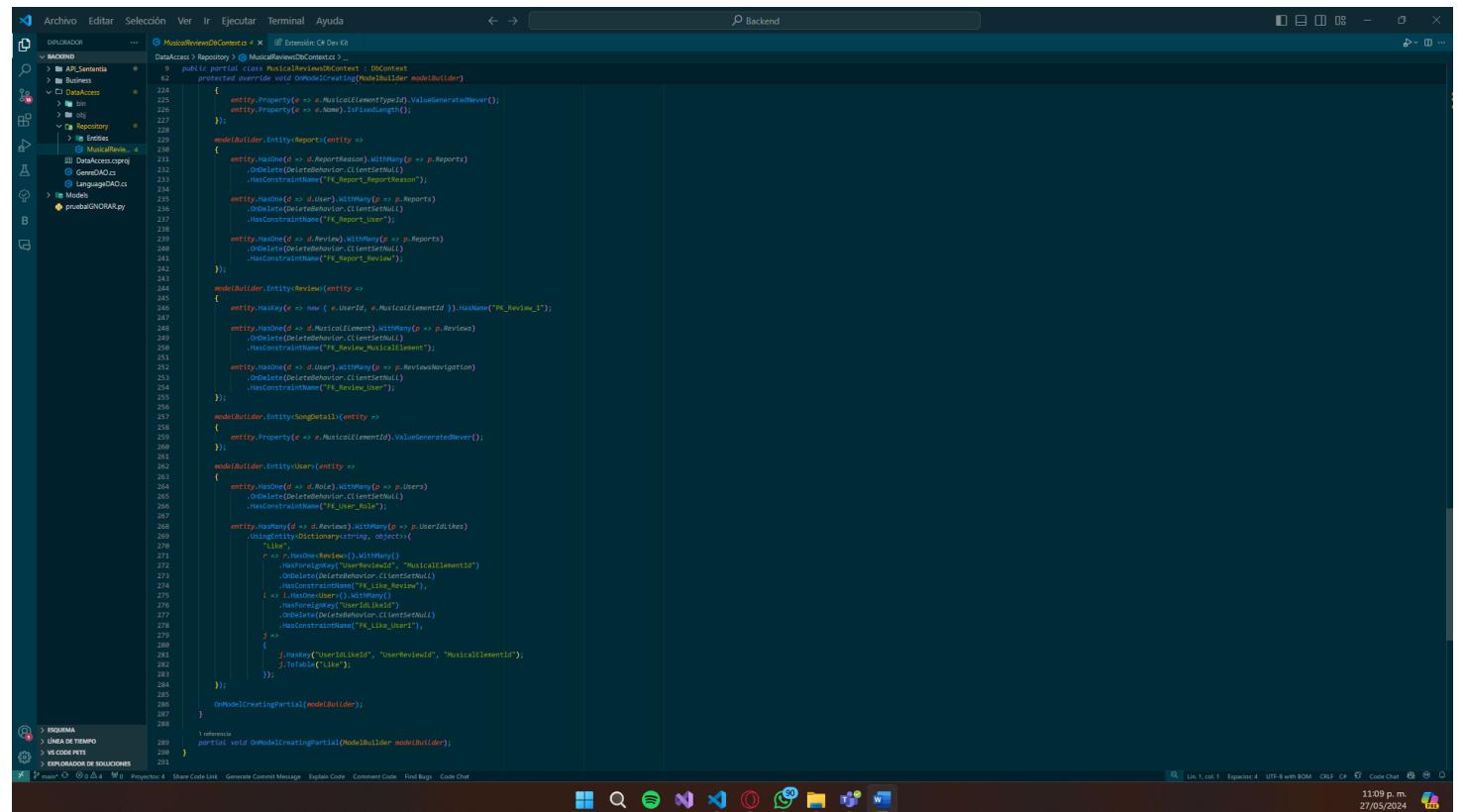
        modelBuilder.Entity<User>()
            .HasMany(d => d.Reviews);
        modelBuilder.Entity<Review>()
            .HasOne(d => d.User);

        modelBuilder.Entity<MusicalElement>()
            .HasMany(d => d.Users).WithMany(p => p.Lists)
            .UsingEntity<Dictionary<string, object>>(
                e => e.Name,
                e => e.HasOne<User>(e => e.UserId),
                e => e.HasOne<List>(e => e.List)
            );
        modelBuilder.Entity<List>()
            .HasMany(e => e.Elements);
        modelBuilder.Entity<User>()
            .HasMany(d => d.Lists);

        modelBuilder.Entity<MusicalElement>()
        {
            entity.Property(e => e.MusicalElementId).ValueGeneratedOnAdd();
            entity.HasKey(e => e.MusicalElementId);
            entity.Property(e => e.Name).HasMaxLength(50);
            entity.Property(e => e.Description).HasMaxLength(500);
            entity.Property(e => e.Type).IsFixedLength(true);

            entity.HasOne(e => e.MusicalElement).WithOne(p => p.MusicalElement)
                .HasForeignKey(e => e.MusicalElementId);
            entity.HasOne(e => e.AlbumDetail).WithOne(p => p.AlbumDetail)
                .HasForeignKey(e => e.AlbumDetailId);
            entity.HasOne(e => e.BandDetail).WithOne(p => p.BandDetail)
                .HasForeignKey(e => e.BandDetailId);
            entity.HasOne(e => e.SongDetail).WithOne(p => p.SongDetail)
                .HasForeignKey(e => e.SongDetailId);

            entity.HasOne(d => d.MusicalElementType).WithMany(p => p.MusicalElements)
                .HasForeignKey(d => d.MusicalElementType);
            entity.HasOne(d => d.Genre).WithMany(p => p.MusicalElements)
                .HasForeignKey(d => d.Genre);
            entity.HasOne(d => d.Reviews).WithMany(p => p.Reviews)
                .HasForeignKey(d => d.ReviewId);
            entity.HasOne(d => d.User).WithMany(p => p.Reviews)
                .HasForeignKey(d => d.UserId);
        };
    }
}
```



```
public partial class MusicalReviewsDBContext : DbContext
{
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<MusicalReview>()
            .HasKey(e => e.Id);
        modelBuilder.Entity<MusicalElement>()
            .HasKey(e => e.Id);
        modelBuilder.Entity<User>()
            .HasKey(u => u.Id);
        modelBuilder.Entity<Review>()
            .HasKey(r => r.Id);

        modelBuilder.Entity<MusicalReview>()
            .HasMany(d => d.Tags).WithMany(p => p.Lists)
            .UsingEntity<Dictionary<string, object>>(
                e => e.Name,
                e => e.HasOne(e => e.Tag),
                e => e.HasOne<List>(e => e.List)
            );
        modelBuilder.Entity<List>()
            .HasMany(e => e.Tags);
        modelBuilder.Entity<Tag>()
            .HasMany(e => e.Lists);

        modelBuilder.Entity<User>()
            .HasMany(d => d.Reviews);
        modelBuilder.Entity<Review>()
            .HasOne(d => d.User);

        modelBuilder.Entity<MusicalElement>()
            .HasMany(d => d.Users).WithMany(p => p.Lists)
            .UsingEntity<Dictionary<string, object>>(
                e => e.Name,
                e => e.HasOne<User>(e => e.UserId),
                e => e.HasOne<List>(e => e.List)
            );
        modelBuilder.Entity<List>()
            .HasMany(e => e.Elements);
        modelBuilder.Entity<User>()
            .HasMany(d => d.Lists);

        modelBuilder.Entity<MusicalElement>()
        {
            entity.Property(e => e.MusicalElementId).ValueGeneratedOnAdd();
            entity.HasKey(e => e.MusicalElementId);
            entity.Property(e => e.Name).HasMaxLength(50);
            entity.Property(e => e.Description).HasMaxLength(500);
            entity.Property(e => e.Type).IsFixedLength(true);

            entity.HasOne(e => e.MusicalElement).WithOne(p => p.MusicalElement)
                .HasForeignKey(e => e.MusicalElementId);
            entity.HasOne(e => e.AlbumDetail).WithOne(p => p.AlbumDetail)
                .HasForeignKey(e => e.AlbumDetailId);
            entity.HasOne(e => e.BandDetail).WithOne(p => p.BandDetail)
                .HasForeignKey(e => e.BandDetailId);
            entity.HasOne(e => e.SongDetail).WithOne(p => p.SongDetail)
                .HasForeignKey(e => e.SongDetailId);

            entity.HasOne(d => d.MusicalElementType).WithMany(p => p.MusicalElements)
                .HasForeignKey(d => d.MusicalElementType);
            entity.HasOne(d => d.Genre).WithMany(p => p.MusicalElements)
                .HasForeignKey(d => d.Genre);
            entity.HasOne(d => d.Reviews).WithMany(p => p.Reviews)
                .HasForeignKey(d => d.ReviewId);
            entity.HasOne(d => d.User).WithMany(p => p.Reviews)
                .HasForeignKey(d => d.UserId);
        };
    }

    modelBuilder.Entity<Report>()
    {
        entity.HasKey(e => new { e.UserId, e.MusicalElementId });
        entity.ToTable("Tb_Report");
        entity.Property(e => e.UserId).IsFixedLength(true);
        entity.Property(e => e.MusicalElementId).IsFixedLength(true);

        entity.HasOne(d => d.ReportReason).WithMany(p => p.Reports)
            .HasForeignKey(d => d.ReportReasonId);
        entity.HasOne(d => d.User).WithMany(p => p.Reports)
            .HasForeignKey(d => d.UserId);
        entity.HasOne(d => d.Review).WithMany(p => p.Reports)
            .HasForeignKey(d => d.ReviewId);
    };

    modelBuilder.Entity<Review>()
    {
        entity.Property(e => e.ReviewId).ValueGeneratedNever();
        entity.HasKey(e => e.ReviewId);
        entity.HasOne(d => d.User).WithMany(p => p.Reviews)
            .HasForeignKey(d => d.UserId);
        entity.HasOne(d => d.MusicalElement).WithMany(p => p.Reviews)
            .HasForeignKey(d => d.MusicalElementId);
        entity.HasOne(d => d.ReviewType).WithMany(p => p.Reviews)
            .HasForeignKey(d => d.ReviewTypeId);
        entity.HasOne(d => d.ReviewStatus).WithMany(p => p.Reviews)
            .HasForeignKey(d => d.ReviewStatusId);
    };

    modelBuilder.Entity<SongDetail>()
    {
        entity.Property(e => e.SongDetailId).ValueGeneratedNever();
        entity.HasKey(e => e.SongDetailId);
        entity.HasOne(d => d.MusicalElement).WithMany(p => p.SongDetails)
            .HasForeignKey(d => d.MusicalElementId);
    };

    modelBuilder.Entity<User>()
    {
        entity.HasOne(d => d.Role).WithMany(p => p.Users)
            .HasForeignKey(d => d.RoleId);
        entity.HasOne(d => d.UserSetting).WithOne(p => p.UserSetting)
            .HasForeignKey(d => d.UserId);
        entity.HasMany(d => d.Reviews).WithMany(p => p.UserReviews)
            .UsingEntity<Dictionary<string, object>>(
                e => e.UserId,
                e => e.ReviewId,
                e => e.MusicalElementId
            );
        entity.HasOne(d => d.UserSetting).WithOne(p => p.UserSetting)
            .HasForeignKey(d => d.UserId);
        entity.HasMany(d => d.Likes).WithMany(p => p.UserLikes)
            .UsingEntity<Dictionary<string, object>>(
                e => e.UserId,
                e => e.MusicalElementId
            );
        entity.HasOne(d => d.UserSetting).WithOne(p => p.UserSetting)
            .HasForeignKey(d => d.UserId);
        entity.HasMany(d => d.Reviews).WithMany(p => p.UserReviews)
            .UsingEntity<Dictionary<string, object>>(
                e => e.UserId,
                e => e.ReviewId,
                e => e.MusicalElementId
            );
        entity.HasOne(d => d.UserSetting).WithOne(p => p.UserSetting)
            .HasForeignKey(d => d.UserId);
        entity.HasMany(d => d.Likes).WithMany(p => p.UserLikes)
            .UsingEntity<Dictionary<string, object>>(
                e => e.UserId,
                e => e.MusicalElementId
            );
        entity.HasOne(d => d.UserSetting).WithOne(p => p.UserSetting)
            .HasForeignKey(d => d.UserId);
    };
}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}
```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

Backend

```
GeneDAO.cs X C# Dev Kit
```

```
namespace DataAccess
{
    public class GeneDAO
    {
        //reference
        public class Genre
        {
            //reference
            private readonly MusicalReviewsDbContext _context;
            public GeneDAO(IConfiguration config)
            {
                _context = new MusicalReviewsDbContext(config);
            }
            //post
            public async Task CreateGenre(Genre genre)
            {
                _context.Genres.Add(genre);
                await _context.SaveChangesAsync();
                return genre;
            }
            //get
            public Task<IEnumerable<Genre>> GetGenre()
            {
                return await _context.Genres.ToListAsync();
            }
            //get by id
            public Task<Genre> GetGenreById(int id)
            {
                return await _context.Genres.FindAsync(id);
            }
            //put
            public Task UpdateGenre(Genre genre)
            {
                _context.Entry(genre).State = EntityState.Modified;
                await _context.SaveChangesAsync();
                return;
            }
            //delete
            public Task DeleteGenre(int id)
            {
                var genre = await _context.Genres.FindAsync(id);
                _context.Genres.Remove(genre);
                await _context.SaveChangesAsync();
                return;
            }
        }
    }
}
```

ESQUEMA  
LÍNEA DE TIEMPO  
VS CODE PETS  
EXPLORADOR DE SOLUCIONES

File main O 0 0 2 Projects 4 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lin. 1, col. 1 Espacio 4 UTF-8 with BOM CRLF CR ⚡ Code Chat

11:09 p.m.  
27/05/2024

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

Backend

```
LanguageDAO.cs X C# Dev Kit
```

```
namespace DataAccess
{
    public class LanguageDAO
    {
        //reference
        public class Language
        {
            //reference
            private readonly MusicalReviewsDbContext _context;
            public LanguageDAO(IConfiguration config)
            {
                _context = new MusicalReviewsDbContext(config);
            }
            //post
            public Task<Language> CreateLanguage(Language language)
            {
                _context.Languages.Add(language);
                await _context.SaveChangesAsync();
                return language;
            }
            //get
            public Task<IEnumerable<Language>> GetLanguage()
            {
                return await _context.Languages.ToListAsync();
            }
            //get by id
            public Task<Language> GetLanguageById(int id)
            {
                return await _context.Languages.FindAsync(id);
            }
            //put
            public Task UpdateLanguage(Language language)
            {
                _context.Entry(language).State = EntityState.Modified;
                await _context.SaveChangesAsync();
                return;
            }
            //delete
            public Task DeleteLanguage(int id)
            {
                var language = await _context.Languages.FindAsync(id);
                _context.Languages.Remove(language);
                await _context.SaveChangesAsync();
                return language;
            }
        }
    }
}
```

ESQUEMA  
LÍNEA DE TIEMPO  
VS CODE PETS  
EXPLORADOR DE SOLUCIONES

File main O 0 0 2 Projects 4 Share Code Link Generate Commit Message Explain Code Comment Code Find Bugs Code Chat

Lin. 1, col. 1 Espacio 4 UTF-8 with BOM CRLF CR ⚡ Code Chat

11:10 p.m.  
27/05/2024

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Models.Genre
9 {
10     [System.ComponentModel.DataAnnotations]
11     public class GenreModel
12     {
13         public int GenrelId { get; set; }
14
15         [StringLength(50)]
16         [Required]
17         public string Name { get; set; } = null!;
18
19         [StringLength(50)]
20         //value cannot be null
21         [Required]
22         public string Description { get; set; } = null!;
23
24         #region Post
25
26         [System.ComponentModel.DataAnnotations]
27         public class GenrePost_Request
28         {
29             [StringLength(50)]
30             //value cannot be null
31             [Required]
32             public string Description { get; set; } = null!;
33
34             [StringLength(50)]
35             //value cannot be null
36             [Required]
37             public string Name { get; set; } = null!;
38
39             #endregion
40
41             public class GenrePost_Response
42             {
43                 //the post id
44                 public int GenrelId { get; set; }
45
46             #endregion
47
48             #region get by id
49
50             public class Genre GetById_Request
51             {
52                 //the id of the genre
53                 [Required]
54                 public int GenrelId { get; set; }
55
56             #endregion
57
58             public class Genre GetById_Response
59             {
60                 public int GenrelId { get; set; }
61
62                 public string Description { get; set; } = null!;
63
64                 public string Name { get; set; } = null!;
65
66             #endregion
67
68             #region get all
69
70             public class Genre GetAll_Request
71             {
72                 public int GenrelId { get; set; }
73
74                 public string Description { get; set; } = null!;
75
76                 public string Name { get; set; } = null!;
77
78             #endregion
79
80             #region put pending
81
82             public class Genre Delete_Request
83             {
84                 //the id of the genre
85                 [Required]
86                 public int GenrelId { get; set; }
87
88             #endregion
89
90             public class Genre Delete_Response
91             {
92                 //the id of the genre
93                 [Required]
94             #endregion
95         }
96     }
97 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Models.Genre
9 {
10     [System.ComponentModel.DataAnnotations]
11     public class GenreModel
12     {
13         public int GenrelId { get; set; }
14
15         [StringLength(50)]
16         [Required]
17         public string Name { get; set; } = null!;
18
19         [StringLength(50)]
20         //value cannot be null
21         [Required]
22         public string Description { get; set; } = null!;
23
24         #region Get
25
26         public class GenreGetById_Request
27         {
28             //the id of the genre
29             [Required]
30             public int GenrelId { get; set; }
31
32             #endregion
33
34             public class GenreGetById_Response
35             {
36                 public int GenrelId { get; set; }
37
38                 public string Description { get; set; } = null!;
39
40                 public string Name { get; set; } = null!;
41
42             #endregion
43
44             #region Get All
45
46             public class Genre GetAll_Request
47             {
48                 public int GenrelId { get; set; }
49
50                 public string Description { get; set; } = null!;
51
52                 public string Name { get; set; } = null!;
53
54             #endregion
55
56             #region Put Pending
57
58             public class Genre Delete_Request
59             {
60                 //the id of the genre
61                 [Required]
62                 public int GenrelId { get; set; }
63
64             #endregion
65
66             public class Genre Delete_Response
67             {
68                 //the id of the genre
69                 [Required]
70                 public int GenrelId { get; set; }
71
72             #endregion
73
74             #region Delete
75
76             public class GenreDelete_Request
77             {
78                 //the id of the genre
79                 [Required]
80                 public int GenrelId { get; set; }
81
82             #endregion
83
84             public class GenreDelete_Response
85             {
86                 //the id of the genre
87                 [Required]
88             #endregion
89         }
90     }
91 }
```