



IS-, NOMBRE CLASE Primer Periodo 2019

Informe De Programación Orientada a Objetos
fecha: 10/03/2019

Elaborado por:

20161005918 Jose Enrique Barrientos
20161000203 Víctor Amaya
20121012266 Edwin Quintero

Catedrático: Erick Vladimir Reyes Marín

Índice

Tema	pagina
Introducción	3
Título	4
Código y funcionalidad	6
Niveles de usuarios	13
Técnicas de Investigación	14
Distribucion Trello	16
Glosario	18
Tabla de evaluación	19

Introducción.

En el siguiente informe se detalla las técnicas utilizadas para la creación de un sitio web inspirado en la aplicación SPOTIFY. En la cual se utilizaron etiquetas de html, clases de css y bootstrap.

En esta etapa del proyecto está centrada en la parte visual de la página. Se creó landing page y toda la estructura de las páginas. Cabe destacar que la parte de funcionalidad de la misma todavía no está implementada; En este informe se dará a conocer las técnicas e implementaciones utilizadas, también se adjuntaran capturas de pantalla como ilustración de lo que aquí se va a documentar.

Cabe destacar que este proyecto contara con dos fases, en la siguiente el objetivo será unir la parte visual con la funcionalidad de la página (utilizando java script y otras herramientas)

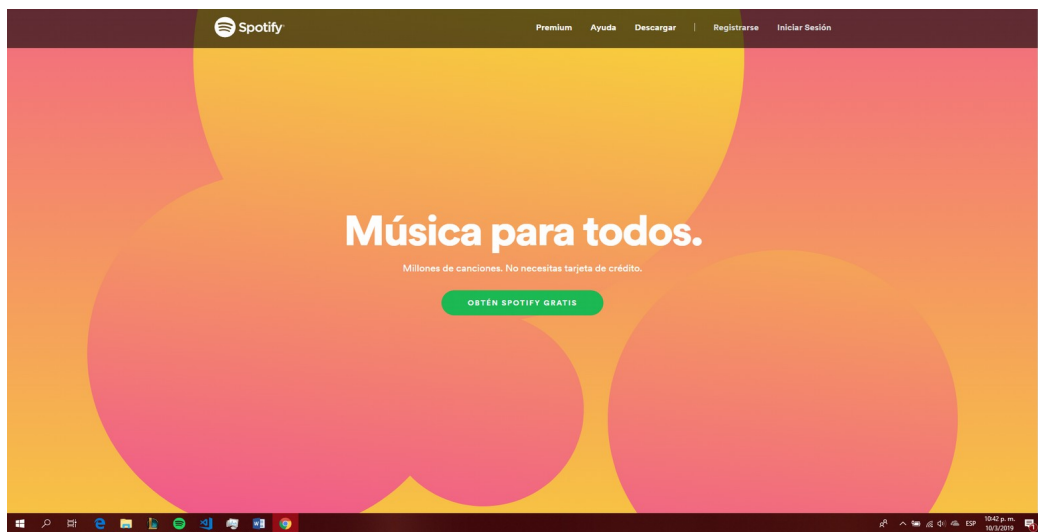
Titulo

Proyecto-Spotify:

Como indica el título, el objetivo de este proyecto es imitar la famosa página de spotify, una página orientada a la reproducción de música mediante una red de internet o previamente descargada en algún dispositivo de almacenamiento.

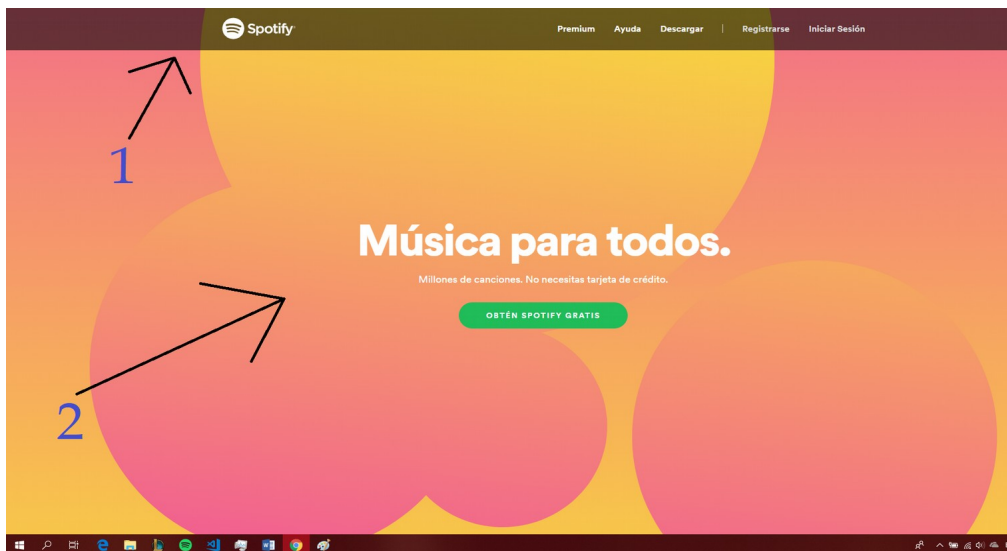
Esta página se caracteriza por presentar diseños muy coloridos y una funcionalidad muy predecible y amigable con el usuario, para así llamar la atención de nuevos posibles usuarios.

Ejemplo de página principal de spotify:



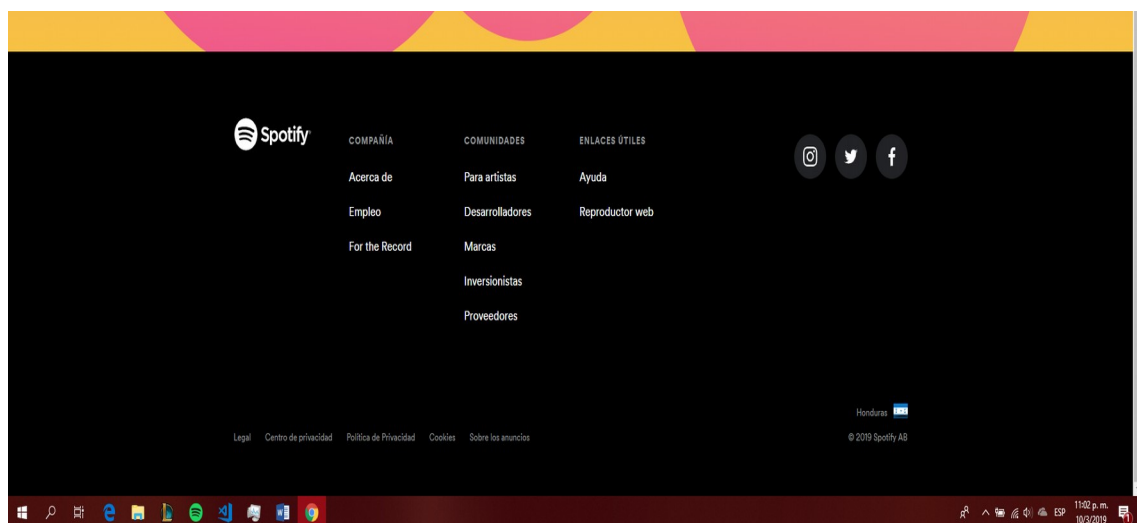
(Esta screen fue tomada de la pagina oficial de; <https://www.spotify.com/hn/>)

Para empezar, fue muy importante identificar detalladamente las partes que iba a contener la realización de este proyecto, si tomamos de ejemplo la pagina principal de su sitio web, podemos obtener que:



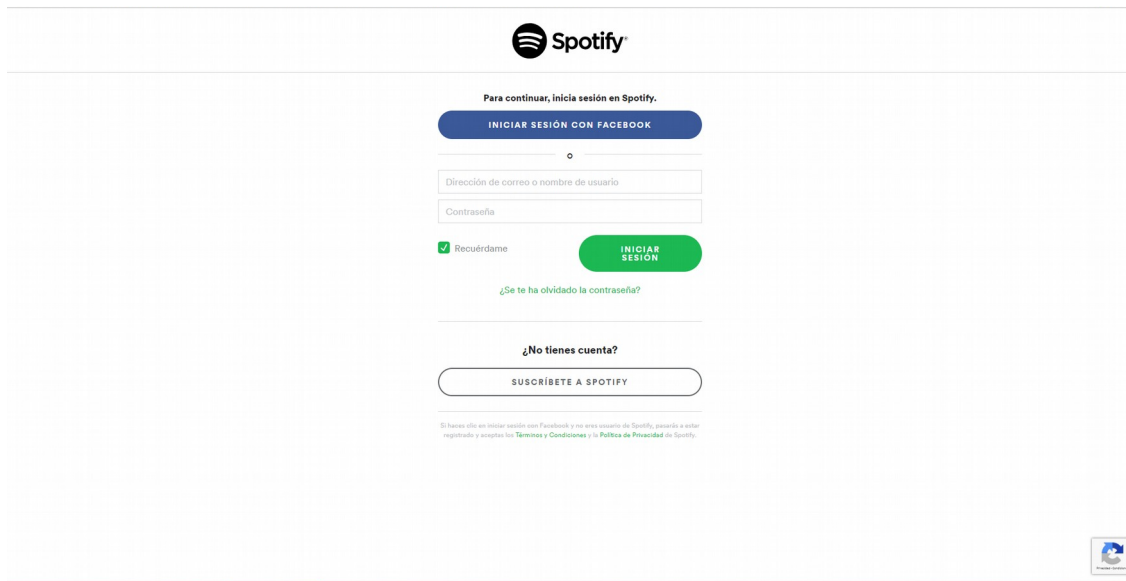
El elemento 1, es una barra, la cual sirve para brindar utilidad al usuario, se utilizó la clase nav (de la cual se detallara más adelante). Por su parte el elemento dos, es el cual más espacio abarca del tamaño de la pagina, se pueden apreciar solo dos oraciones y un botón, con el cual se aplicaron etiquetas de: H1, H5 y una para creación del boton de tipo button. Todo esto esta encapsulado dentro de un main, el cual tiene etiquetas como “div class=container”, “div class=row”.

Hay un tercer elemento, el cual es el pie de página, también conocido como “footer”, el cual incluye utilidades y información para el usuario:



Con esto podemos entender la estructura básica, la cual está dada por un header o encabezado, después su main que estará contenido en un body y por último su pie de página o footer. Cabe destacar que esta identificación planteada es muy a grandes rasgos, ya que se implementaron muchas más etiquetas para poder aplicar la visualización correcta del sitio.

Cabe destacar que esta es solo la página principal de Spotify, entonces surge la siguiente pregunta ¿se aplica esta misma distribución para los otros apartados de Spotify?, la respuesta es que relativamente no, aunque se conserva cierta idea e igualdad antes establecida, es innegable el hecho de que hay diferencias en tema si es una página de registro o el famoso reproductor web. Esto llevó a la tarea de hacer un planteamiento diferente para cada página a montar para aplicar la mayor similitud al sitio original web.

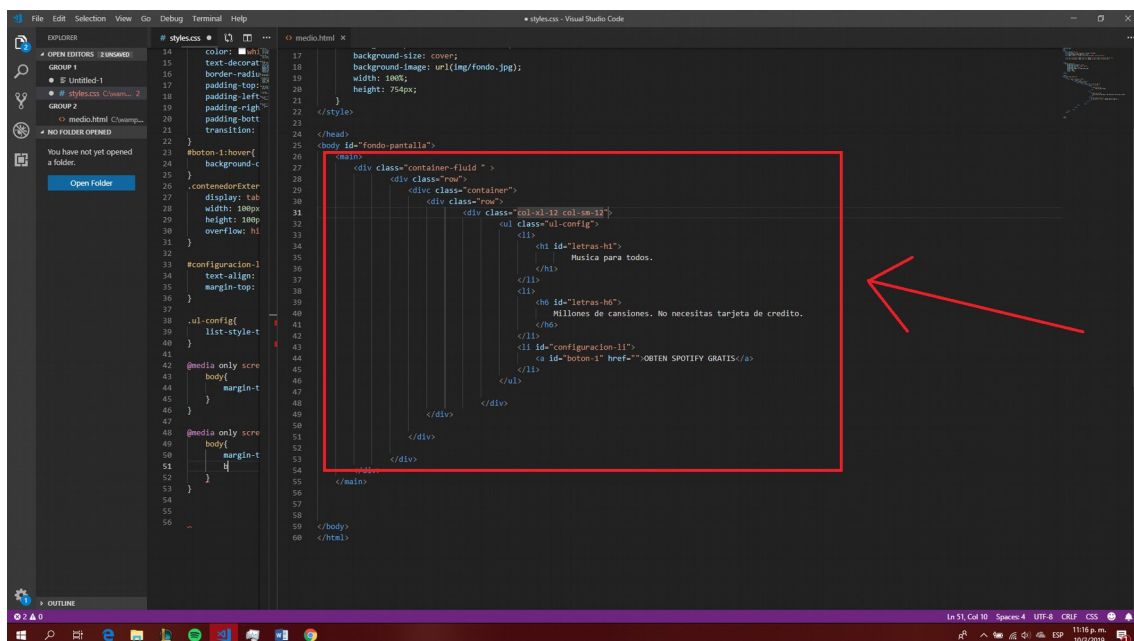


(Ejemplo: página de registro de spotify, obtenida <https://accounts.spotify.com/es>)

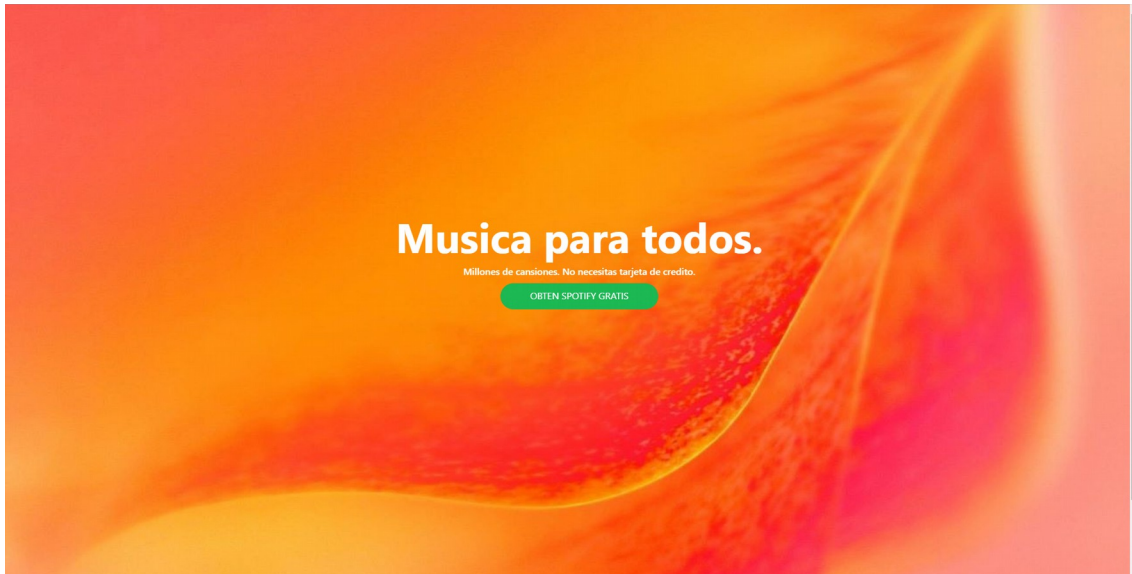
Código y funciones utilizadas.

Como ya hemos plasmado una idea general de lo que se quería imitar es hora de pasar a como se logró imitar, esto solo será posible como la visualización de código y ejemplos del mismo.

Si iniciamos con nuestro primer ejemplo, nuestro main, barra de tareas, pie de paginas:



Si nos centramos en el cuadro marcado (color rojo) podemos ver que es una etiqueta main conteniendo las demás etiquetas, esta funcionalidad se aplicó para poder lograr un ajuste automático de imagen, logrando esto:



Como se puede apreciar en el ejemplo, es solo su main, sin su barra de tareas ni pie de página, esto se hizo solo para demostrar la utilidad del código e irlo explicando por partes.

Función destacable: @media

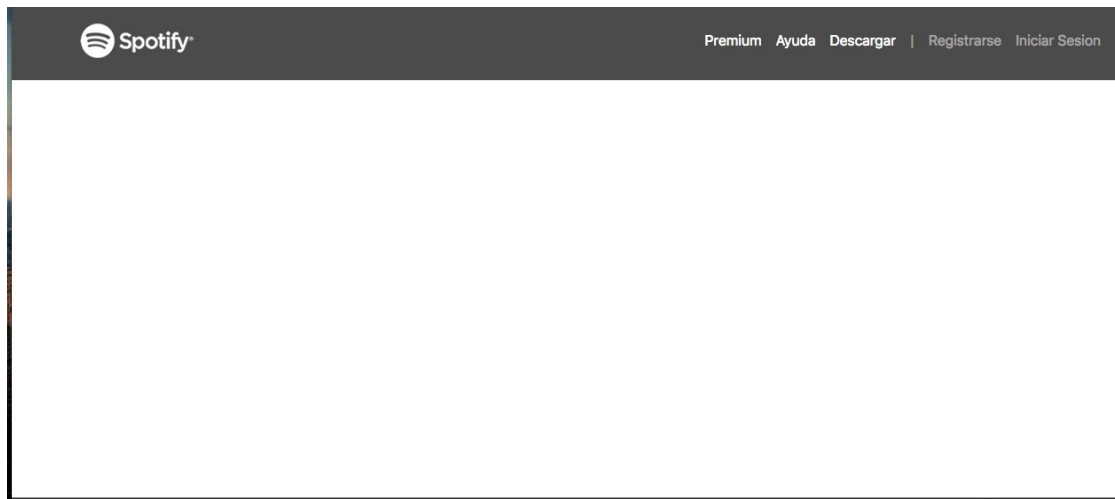
Esta clase o función, se utilizó muy seguido para brindar ajustes automáticos de pantalla (relacionada al tamaño de pantalla), ejemplo:

```
40 }
41
42 @media only screen and (max-width:600px){
43     body{
44         margin-top:200px;
45     }
46 }
47
48 @media only screen and (min-width:600px){
49     body{
50         margin-top:350px;
51         b
52     }
53 }
54
```

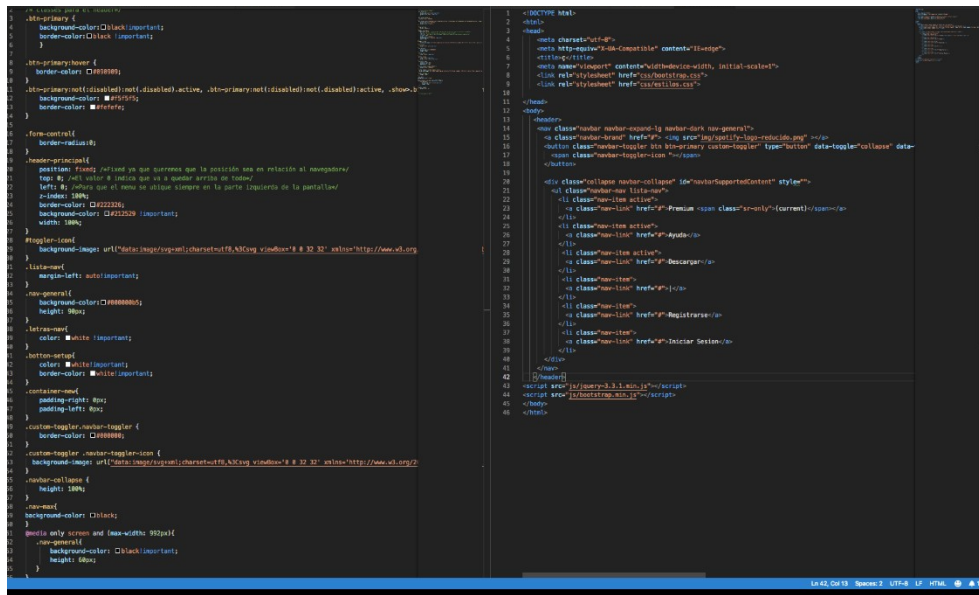
Este fragmento de código (aplicado en el documento “styles.css” respectivamente) regula el tamaño de separación que tendrá una etiqueta respecto a la variación de tamaño de la pagina.

Se utilizó en conjunto, clases como container, container-fluid, y row para poder lograr el efecto deseado.

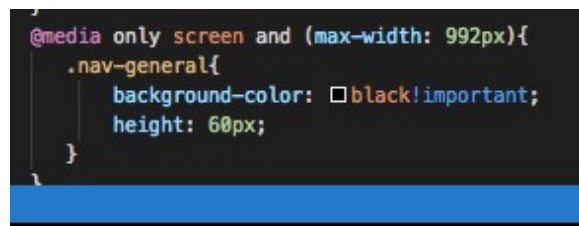
Ahora pasaremos a otra categoría, la barra de tareas o heder, esta barra tiene características que brindan al usuario de muchas funcionalidades (aunque solo nos centramos en que estén disponibles visualmente y no mecánicamente en esta fase del proyecto).



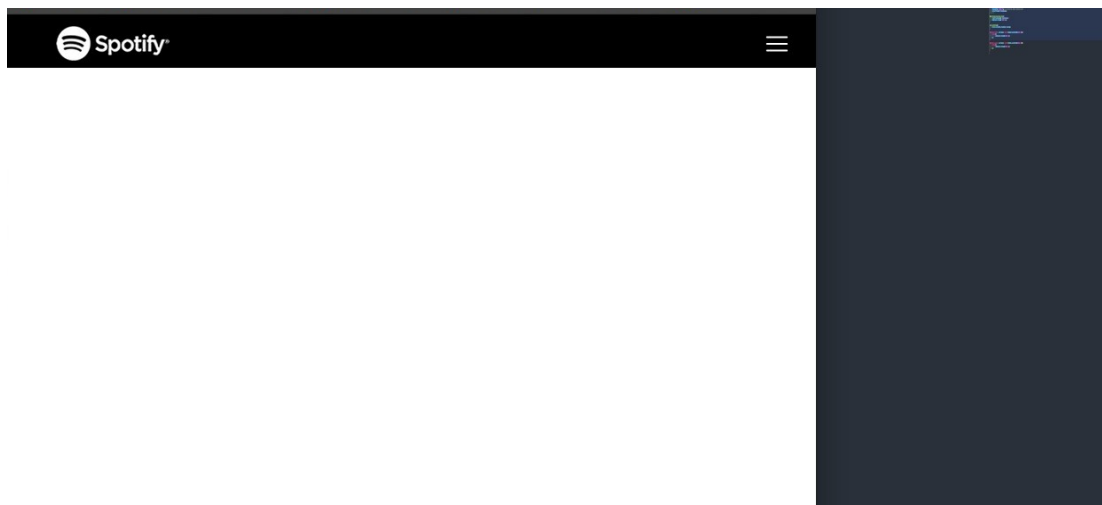
Este es solo la página siendo cargada con la estructura de código del header. Para entender mejor esto se procederá a mostrar su código, etiquetas, clases etc.



Con este segmento de código se logra generar la barra antes mostrada, algunas etiquetas son ya las antes conocidas, aunque hay algunas interesantes que merecen ser comentadas:



Es la misma función de @media, pero ¿a que se aplico exactamente? , en la función anterior vimos que fue aplicada a un cambio de distancia en “top”, aquí esto no se distancio mucho, aunque se aplico otro atributo “background-color”, para mostrar su utilidad es necesario la ilustración:



Lo que podemos notar es que, si la página se reduce la barra de tareas, también tiene que ajustarse y no solo eso, cambiar su tonalidad de color, para este fin es la estructura de código antes mostrada.

Cabe aclarar que este cambio sucede después de los 992px en pantalla.

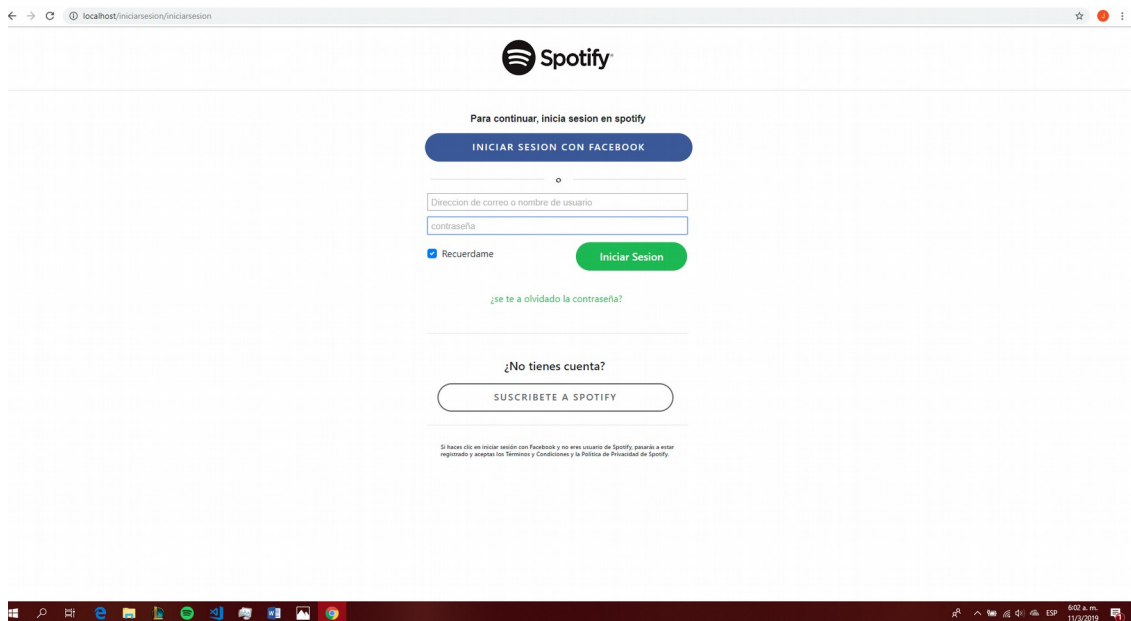
Función destacable: Position: fixed;

Esta función ha sido comentada en el código mismo, su explicación está contenida en ella.

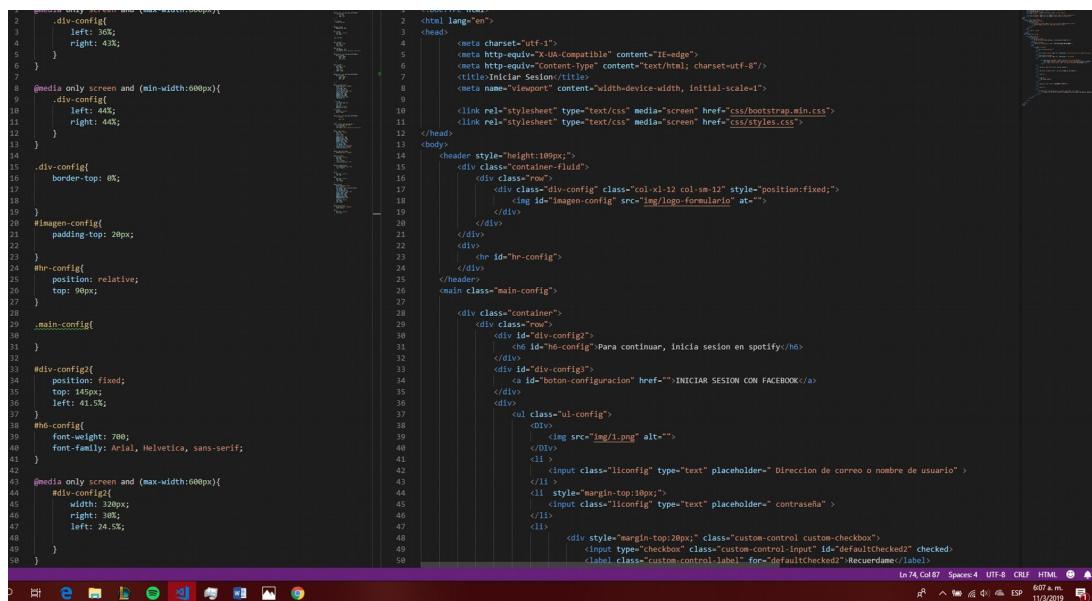
```
.header-principal{  
  position: fixed; /*Fixed ya que queremos que la posición sea en relación al navegador*/  
  top: 0; /*El valor 0 indica que va a quedar arriba de todo*/  
  left: 0; /*Para que el menu se ubique siempre en la parte izquierda de la pantalla*/  
  z-index: 100%;  
  border-color: #222326;  
  background-color: #212529 !important;  
  width: 100%;  
}
```

Diseño de formulario (registro e iniciar sesión)

No puede faltar una de las partes más importantes para una página, es el uso de formularios para pedir información al usuario. Este tipo de formularios son muy usados aunque pueden variar sus estructuras de uno con otros. Para ellos mostraremos el diseño para la página de Spotify:



En este formulario podemos encontrar algunas funcionalidades como ser botones o entradas para que el usuario ingrese sus datos, (las verificaciones se realizan con java, pero no se implementaran en este avance).



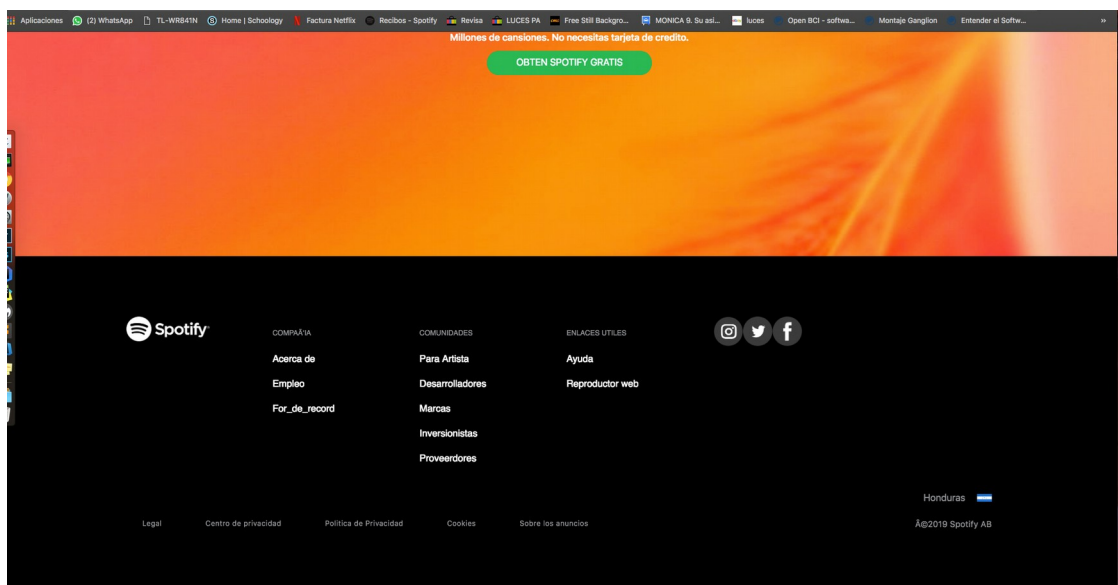
En la imagen anterior tenemos el código usado para el formulario de inicio de sesión, cabe destacar que elementos como el “media” fueron usados reiteradas veces para ajustar tamaños en pantalla. Etiquetas como “ul” y “li” se aplicaron para la creación de una lista desordenada:

```

<div id="div-config3">
  <a id="boton-configuracion" href="">INICIAR SESION CON FACEBOOK</a>
</div>
<div>
  <ul class="ul-config">
    <li>
      <div>
        
      </div>
    </li>
    <li>
      <input class="liconfig" type="text" placeholder=" Direccion de correo o nombre de usuario" >
    </li>
    <li style="margin-top:10px;">
      <input class="liconfig" type="text" placeholder=" contraseña" >
    </li>
    <li>
      <div style="margin-top:20px;" class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="defaultChecked2" checked>
        <label class="custom-control-label" for="defaultChecked2">Recuerdame</label>
      </div>
    </li>
    <li>
      <a id="boton-registrate" href="">Iniciar Sesion</a>
    </li>
    <li>
      <a id="boton-olvidado" href="">¿se te a olvidado la contraseña?</a>
    </li>
    <li>
      <hr id="hr2">
    </li>
    <li>
      <h5 id="h5">
        ¿No tienes cuenta?
      </h5>
    </li>
    <li>
      <a id="boton-final" href="">SUSCRIBETE A SPOTIFY</a>
    </li>
    <li>
      <hr id="hr3">
    </li>
    <li>
      <h6 style="font-size:10px;position: relative; top: 135px; left: 22px;">
        Si haces clic en iniciar sesión con Facebook y no eres usuario de Spotify, pasarás a estar <br>
      </h6>
    </li>
  </ul>
</div>
</div>

```

Pagina ya termina con el footer, barra de tareas y main:

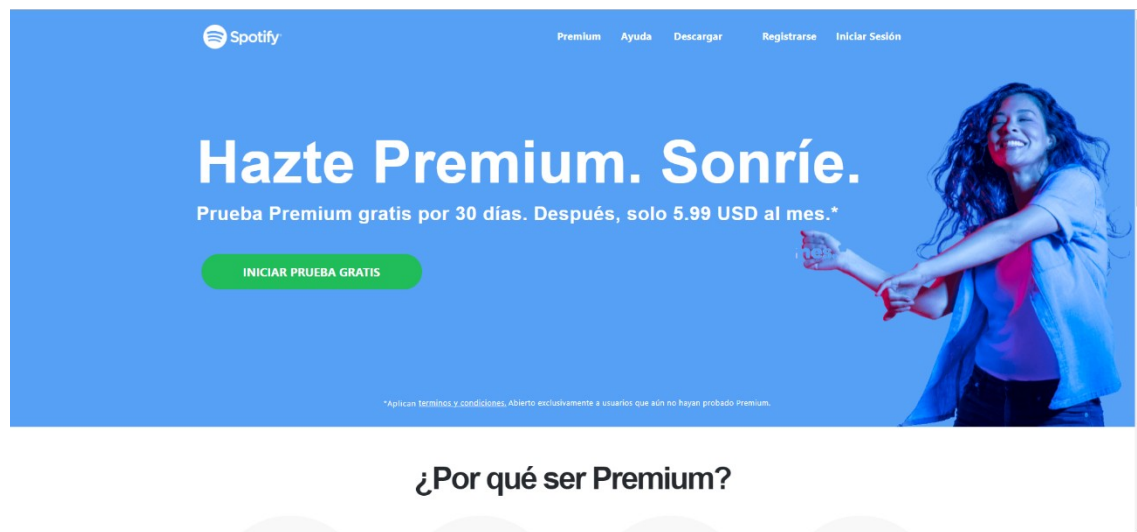


Niveles de usuario

En nuestro proyecto, tenemos 3 niveles principales de usuario con diferentes permisos de administrador, como es una aplicación de paga variada en ese ambito.

-usuarios que usan los servicios gratis: Estos usuarios no tienen acceso a muchas funcionalidades de la página, solo cuenta con algunas opciones básicas.

-usuarios de paga o premium: Estos usuarios son aquellos que utilizan la aplicación de paga, tienen acceso a todos los servicios, pero no pueden configurar el server o editar el contenido.



-administrador: Este usuario está orientado más al mantenimiento de la plataforma, es un administrador que puede editar el contenido y utilizar la información de usuario.

Técnicas de Investigación

Darle estilos a un botón tipo check-box:

Normalmente o mediante css esto no es posible, ya que no lo permite, pero mediante una investigación, hay maneras para encubrir o enmascarar con ciertos estilos este tipo de botones:

```
<div style="margin-top:20px;" class="custom-control custom-checkbox">
  <input type="checkbox" class="custom-control-input" id="defaultChecked2" checked>
  <label class="custom-control-label" for="defaultChecked2">Recuerdame</label>
</div>
```

Con este fragmento de código logramos dar una apariencia personalizada a este tipo de botones, incluyendo una etiqueta de tipo “label”.

Overflow: Esta funcionalidad es muy útil cuando un texto excede el tamaño dado, sirve para controlar si queremos barras de desplazamiento o que comportamiento tomara el tamaño excedente de nuestra página. Hay varias variantes,

- **visible**- Defecto. El desbordamiento no se recorta. El contenido se representa fuera de la caja del elemento.
- **hidden** - El desbordamiento se recorta, y el resto del contenido será invisible.
- **scroll** - El desbordamiento se recorta y se agrega una barra de desplazamiento para ver el resto del contenido.
- **auto**- Similar a **scroll**, pero agrega barras de desplazamiento solo cuando es necesario


```

}


```

Como podemos observar, en esta clase de “ul-navegador” se aplico este atributo, para controlar el tamaño excedente de la pagina.

Funcionalidad Display

a propiedad CSS display especifica el tipo de cuadro de representación utilizado para un elemento. En HTML, los valores de propiedad predeterminados display se toman de los comportamientos de los cambios en las especificaciones HTML o de la hoja de estilo predeterminada del navegador / usuario. El valor predeterminado en XML es inline. Además de los Diferentes Tipos de caja de Visualizacion, el valor de none permite Desactivar la Visualizacion DE UN Elemento; cuando no se utiliza none, todos los elementos descendentes también quedan desactivados. El documento se procesa como si el elemento no existiera en el árbol de documentos.

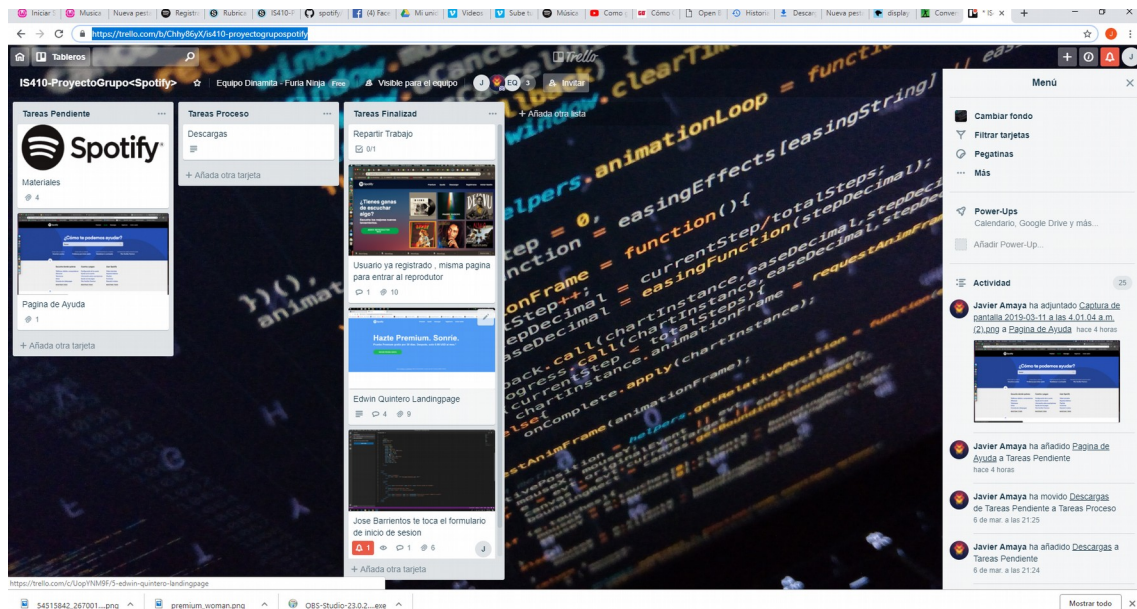
```

}


```


Link Para acceder a distribucion de tareas Trello

<https://trello.com/b/Chhy86yX/is410-proyectogrupospotify>



Glosario

- Plantilla: Normalmente asociada a bootstrap, nos brinda código ya generado para a si solo editarlo y personalizarlo, es muy utilizada para ejemplos.
- Clase: Método utilizado en css para brindar atributos o caracterizas a uno o más elementos (etiquetas) de html.
- Id: es un identificador que se puede dar a una etiqueta de html, cumple las mismas funciones que una clase, a diferencia que solo se puede dar a un solo elemento y no varios.
- Etiquetas: elementos que se crean en html.
- Header: etiqueta asociada para crear encabezados de páginas, aquí se suelen montar elementos como la barra nab o barras con utilidades o botones.
- Position: atributo de css, se puede aplicar en línea, clase o id. Sirve para controlar la posición de un elemento. Relativa, absoluta, fixed, etc.
- Main: Etiqueta aplicada a html, utilizada para montar la parte principal de una página.
- Hover: esta funcionalidad se aplica en id o clases, sirve para indicar una transición y a si realizar cambios de atributos.
- Footer: Esta etiqueta de html, sirve para indicar un pie de página, con ella se pueden centrar barras de ayudas para el usuario o servicios y políticas.
- ul: Esta etiqueta sirve para montar listas desordenadas.

Integrante	Nota
------------	------

Jose Barrientos	200%
Edwin quintero	200%
Víctor Amaya	200%

Tabla de evaluación

Sabemos que no está perfecto ingeniero, pero le pusimos corazón, los tres nos esforzamos para un mejor futuro en la clase de poo, para ser solidarios con nuestros demás compañeros de clase, denos un 100% y no el 200%.

Pdt: Estaba buena la serie, pero mejor mírese one piece(Jose Barrientos).

Pdt2:disculpe si sale medio hueca la tabla, pero es que tiro error cuando la pasamos a pdf :c