

Hernández Posadas Diego Ángel

Matricula: 184369

Universidad politécnica de San Luis Potosí

Programación web

Manual de programador

Carrera: ITI

**UNIVERSIDAD
POLITÉCNICA**

DE SAN LUIS POTOSÍ

Página principal HTML Avance Proyecto

```
1 <!DOCTYPE html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Pagina principal LootZone</title>
7   <link rel="stylesheet" type="text/css" href="Menu.css">
8 </head>
9 <body>
10  <header class="encabezado">
11    
12    <h1 class="titulo">LootZone</h1>
13    
14  </header>
15  <section>
```

Esta es la manera en como enlazamos con el archivo css

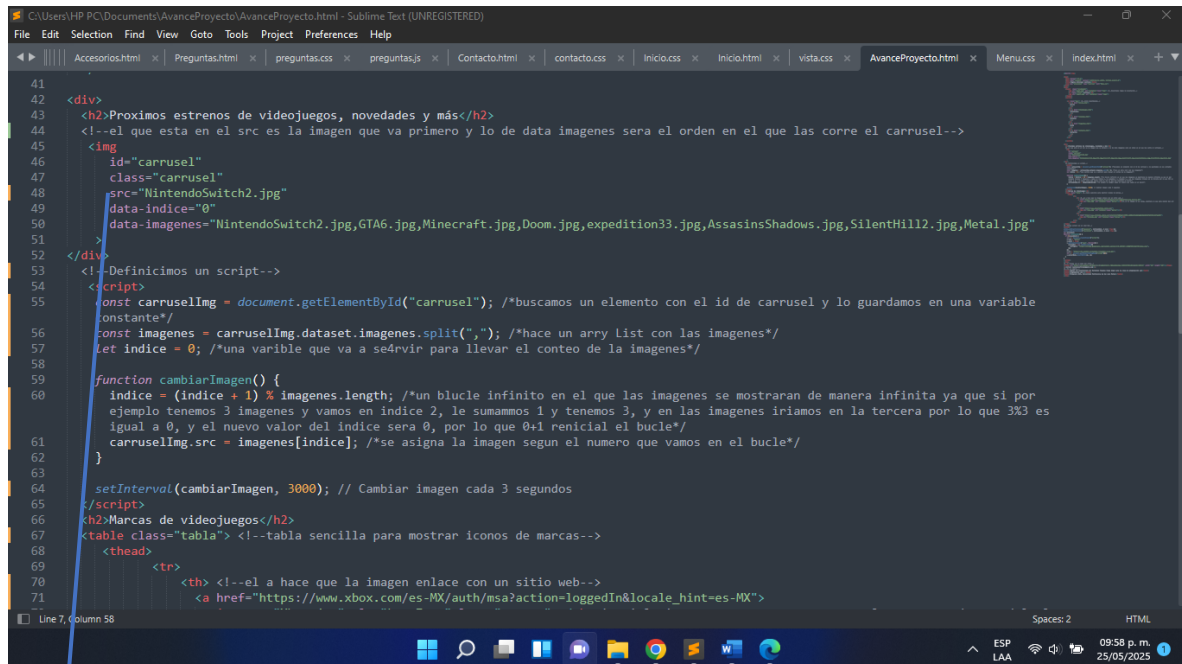
```
<link rel="stylesheet" type="text/css" href="Menu.css">
```

Creación de una lista desordenada

```
<section>
  <ul class="menu"> <!--Lista desordenada-->
    <li><a href="Inicio.html">
      Inicio
    </a>
    </li>
    <li><a href="Videojuegos.html">
      Videojuegos
    </a>
    </li>
    <li><a href="Consolas.html">
      Consolas
    </a>
    </li>
    <li><a href="Preguntas.html">
      Ayuda
    </a>
    </li>
    <li><a href="Contacto.html">
      Contactos
    </a>
    </li>
  </ul>
</section>
```

Esta lista desordenada enlaza por medio de la etiqueta <a> con los demás sitios web de la página.

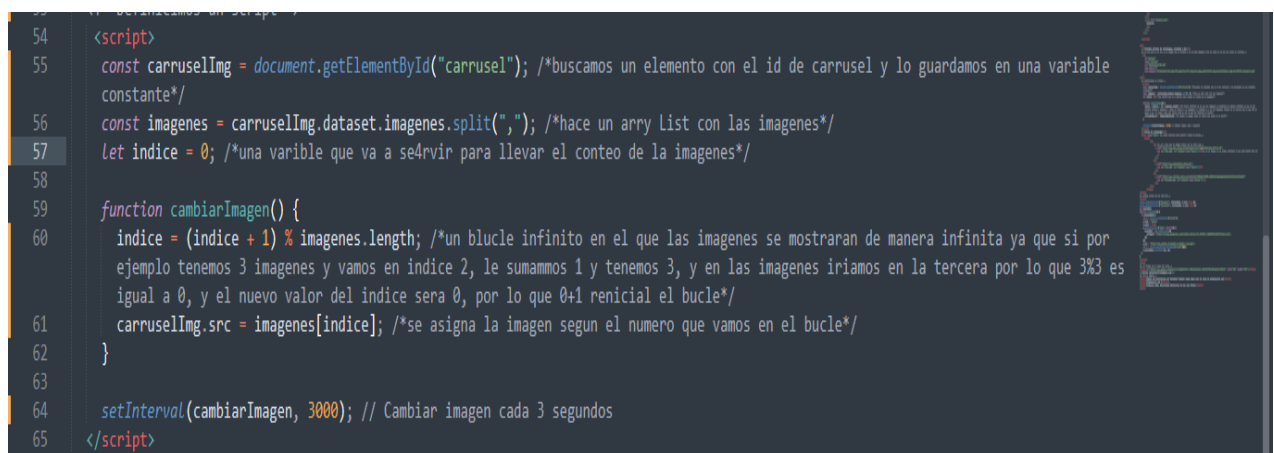
Sección de carrusel de imágenes



```
41
42 <div>
43   <h2>Proximos estrenos de videojuegos, novedades y más</h2>
44   <!--el que esta en el src es la imagen que va primero y lo de data-imagenes sera el orden en el que las corre el carrusel-->
45   
52 </div>
53 <!--Definimos un script-->
54 <script>
55   const carruselImg = document.getElementById("carrusel"); /*buscamos un elemento con el id de carrusel y lo guardamos en una variable
56   constante*/
57   const imagenes = carruselImg.dataset.imagenes.split(","); /*hace un array List con las imagenes*/
58   let indice = 0; /*una variable que va a servir para llevar el conteo de la imagenes*/
59   function cambiarImagen() {
60     indice = (indice + 1) % imagenes.length; /*un bucle infinito en el que las imagenes se mostraran de manera infinita ya que si por
61     ejemplo tenemos 3 imagenes y vamos en indice 2, le sumamos 1 y tenemos 3, y en las imagenes iriamos en la tercera por lo que 3%3 es
62     igual a 0, y el nuevo valor del indice sera 0, por lo que 0+1 renicial el bucle*/
63     carruselImg.src = imagenes[indice]; /*se asigna la imagen segun el numero que vamos en el bucle*/
64   }
65   setInterval(cambiarImagen, 3000); // Cambiar imagen cada 3 segundos
66 </script>
67 <h2>Marcas de videojuegos</h2>
68 <table class="tabla"> <!--tabla sencilla para mostrar iconos de marcas-->
69 <thead>
70   <tr>
71     <th> <!--el a hace que la imagen enlace con un sitio web-->
72     <a href="https://www.xbox.com/es-MX/auth/msa?action=loggedIn&locale_hint=es-MX">
```

- src="NintendoSwitch2.jpg"
Función: es la primera imagen que empieza en el carrusel.
- data-imagenes="NintendoSwitch2.jpg,GTA6.jpg,Minecraft.jpg,Doom.jpg,expedition33.jpg,AssasinsShadows.jpg,SilentHill2.jpg,Metal.jpg"
Función: La parte de data imágenes incluye a todas las imágenes del carrusel

Funcionamiento de cambio de imágenes en el carrusel



```
54 <script>
55   const carruselImg = document.getElementById("carrusel"); /*buscamos un elemento con el id de carrusel y lo guardamos en una variable
56   constante*/
57   const imagenes = carruselImg.dataset.imagenes.split(","); /*hace un array List con las imagenes*/
58   let indice = 0; /*una variable que va a servir para llevar el conteo de la imagenes*/
59   function cambiarImagen() {
60     indice = (indice + 1) % imagenes.length; /*un bucle infinito en el que las imagenes se mostraran de manera infinita ya que si por
61     ejemplo tenemos 3 imagenes y vamos en indice 2, le sumamos 1 y tenemos 3, y en las imagenes iriamos en la tercera por lo que 3%3 es
62     igual a 0, y el nuevo valor del indice sera 0, por lo que 0+1 renicial el bucle*/
63     carruselImg.src = imagenes[indice]; /*se asigna la imagen segun el numero que vamos en el bucle*/
64   }
65   setInterval(cambiarImagen, 3000); // Cambiar imagen cada 3 segundos
66 </script>
```

```
const carruselImg = document.getElementById("carrusel");
```

Función: buscamos un elemento con el id de carrusel y lo guardamos en una variable constante.

```
const imagenes = carruselImg.dataset.imagenes.split(",");
```

Función: Hacemos un arraylist con las imagenes.

```
let indice = 0;
```

Función: una variable que va a servir para llevar el conteo de las imágenes.

```
indice = (indice + 1) % imagenes.length;
```

Función: un bucle infinito en el que las imágenes se mostraran de manera infinita ya que, si por ejemplo tenemos 3 imágenes y vamos en índice 2, le sumamos 1 y tenemos 3, y en las imágenes iríamos en la tercera por lo que $3\%3$ es igual a 0, y el nuevo valor del índice será 0, por lo que $0+1$ reinicia el bucle.

```
carruselImg.src = imagenes[indice];
```

Función: se asigna la imagen según el número que vamos en el bucle.

```
setInterval(cambiarImagen, 3000);
```

Función: Cambiar imagen cada 3 segundos.

Explicación CSS general

```
1
2 body {
3   background-color: #1E1E1E; /*color negro de fondo*/
4   margin: 0;
5   padding: 0;
6   font-family: Arial, sans-serif; /*estilo de fuente, el sans-serif es sin adornos de letra*/
7 }
8
```

background-color: #1E1E1E;

Función: color negro de fondo.

margin: 0;

padding: 0;

Función: margin es el tamaño externo y padding es el tamaño interno.

font-family: Arial, sans-serif;

Función: estilo de fuente, el sans-serif es sin adornos de letra.

```
/* menu horizontal */
.menu {
  list-style-type: none; /*quitar viñetas o estilo preddefinido*/
  width: 60%; /*tamaño preddefinido para la barra de listas*/
  max-width: 800px; /*maximo de tamaño dependiendo de la ventana del navegador*/
  padding: 0;
  background-color: #0D0D0D;
  display: flex; /*activa el modo flexible*/
  align-items: center; /*alinear elementos en el centro*/
  margin: 20px auto; /*20px margen de arriba y de abajo y el auto es el margen de la izquierda y derecha*/
  flex-wrap: wrap; /* Para que se ajuste en pantallas pequeñas */
  border-radius: 8px; /*borde redondo*/
}

.menu li {
  margin: 5px;
  flex: 1 1 auto; /*Este elemento puede crecer para llenar espacio disponible, puede reducirse si no cabe,
  y su tamaño base es el contenido o el valor de width*/
  text-align: center;
}

.menu li a {
  background-color: #0D0D0D;
  color: #00FFFF;
  display: block; /*los elementos de la lista se comportan como bloques, van a ocupar el maximo de ancho que le digas el contenedor*/
  padding: 16px; /*espacio interno*/
  text-decoration: none; /*eliminar decoraciones de texto*/
  text-align: center;
}
```

class="menu" es el indentificador en el html para que en el css podamos aplicarle los estilos

list-style-type: none;

Función: quitar viñetas o estilo preddefinido.

width: 60%;

Función: tamaño preddefinido para la barra de listas.

max-width: 800px;

Función: máximo de tamaño dependiendo de la ventana del navegador.

display: flex;

Función: activa el modo flexible.

align-items: center;

Función: alinear elementos en el centro.

margin: 20px auto;

Función: 20px margen de arriba y de abajo y el auto es el margen de la izquierda y derecha.

flex-wrap: wrap;

Función: Para que se ajuste en pantallas pequeñas.

border-radius: 8px;

Función: borde redondo.

menu li

Los siguientes son los estilos del menú en estado normal.

flex: 1 1 auto;

Función: Este elemento puede crecer para llenar espacio disponible, puede reducirse si no cabe,

y su tamaño base es el contenido o el valor de width.

.menu li a

estilos del menú con los enlaces puestos en modo normal.

color: #00FFF7;

Función: color de fuente.

display: block;

Función: los elementos de la lista se comportan como bloques, van a ocupar el máximo de ancho que le digas el contenedor.

text-decoration: none;

Función: eliminar decoraciones de texto.

.menu li a:hover

Función: hover hace que se puedan cambiar los estilos cuando se pase el mouse.

Parte responsiva del archivo CSS

```
/* Responsive */
@media (max-width: 768px) { /*solo aplica en pantallas de 768 pixeles o menos*/
  .menu {
    width: 100%;
    flex-direction: column; /*cambia la dirección a vertical*/
  }

  .menu li {
    flex: none;
    width: 100%;
  }

  .logo,
  .logo1 {
    position: static;
    display: block;
    margin: 0 auto;
    width: 120px;
    height: 120px;
  }

  .encabezado {
    padding: 20px;
    flex-direction: column;
  }

  .marcas {
    width: 90%;
    height: auto;
  }

  .carrusel {
    width: 100%;
    height: auto;
  }
}
```

@media (max-width: 768px)

Función: solo aplica en pantallas de 768 pixeles o menos.

Como ya expliqué muchas de las líneas de código es solo como se van a mover los elementos al verse afectados por una ventana menor como la de un celular.

margin-top: 30px;

Función: margen de la parte superior.

margin-bottom: 15px;

Función: margen de la parte inferior.

margin-left: auto;

Función: según el espacio que sobre el navegador lo utilizara.

cursor: pointer;

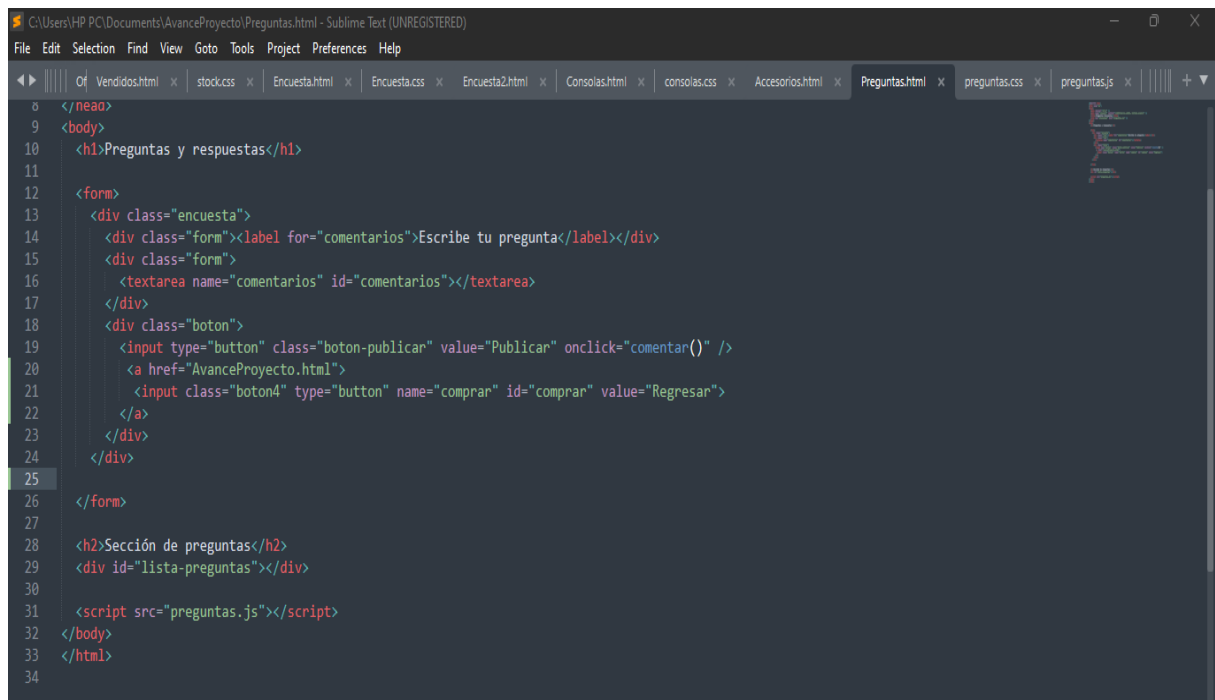
Función: hace que se vea el icono del mouse cuando se pasa por encima del elemento.

Explicación HTML Videojuegos

```
File Edit Selection Find View Goto Tools Project Preferences Help
Videojuegos.html x Stock.html x Preventa.html x Nuevo.html x Ofertas.html x Vendidos.html x stock.css x Encuesta.html x Encuesta.css x Encuesta2.html x
36 </section>
37 <table>
38 <thead>
39 <tr>
40 <th>
41 <a href="Vista.html"><!-- si el usuario pulsa sobre la imagen lo va a enlazar a ese sitio web-->
42 
43 </a>
44 <p class="parrafo">Red Dead Redemption II<br>
45 XONE<br>
46 Precio $500</p>
47 </th>
48 <th>
49 <a href="Vista2.html">
50 
51 </a>
52 <p class="parrafo">Lies of P<br>
53 PS5<br>
54 Precio $1000</p>
55 </th>
56 <th>
57 <a href="Vista3.html">
58 
59 </a>
60 <p class="parrafo">Grand Thet Auto V Premium Edition<br>
61 PS4<br>
62 Precio $600</p>
63 </th>
64 </tr>
65 </thead>
66 <tbody>
67 <tr>
68 <td>
69 <a href="Vista4.html">
70 
```

Los códigos de HTML de Videojuegos, Stock, Preventa, Nuevo, Consolas, Vendidos, ofertas y en los de vistas, todos esos códigos utilicen la misma lógica de las tablas para ordenar los elementos.

Código HTML de preguntas



```
8 </nead>
9 <body>
10 <h1>Preguntas y respuestas</h1>
11
12 <form>
13   <div class="encuesta">
14     <div class="form"><label for="comentarios">Escribe tu pregunta</label></div>
15     <div class="form">
16       <textarea name="comentarios" id="comentarios"></textarea>
17     </div>
18     <div class="boton">
19       <input type="button" class="boton-publicar" value="Publicar" onclick="comentar()" />
20       <a href="AvanceProyecto.html">
21         <input class="boton4" type="button" name="comprar" id="comprar" value="Regresar">
22       </a>
23     </div>
24   </div>
25 </form>
26
27 <h2>Sección de preguntas</h2>
28 <div id="lista-preguntas"></div>
29
30 <script src="preguntas.js"></script>
31 </body>
32 </html>
33
34
```

Aquí utilice otra lógica diferente al de las tablas aquí utilice los div para organizar los elementos.

`<form>`

Función: es la etiqueta de formulario.

`<label>`

Función: es lo que aparece en la parte de arriba, el texto de arriba.

`<input type="button" class="boton-publicar" value="Publicar" onclick="comentar()" />`

Función: es un botón en el que si se le da click en publicar, nos llevara por medio de la función onclick comentar a lo que diga el archivo JS.

`<div id="lista-preguntas"></div>`

Función: aquí se pondrá lo que realice el archivo js.

Archivo JS de preguntas

```
1 function comentar() {  
2   const comentario = document.getElementById('comentarios').value.trim();  
3   const contenedor = document.getElementById('lista-preguntas');  
4  
5   if (comentario === '') { /*si no se escribe ninguna pregunta lanzara este mensaje*/  
6     alert('Por favor, escribe una pregunta antes de publicarla.');7     return;  
8   }  
9 }
```

function comentar()

Función: cuando el usuario le de click hará todo lo que esta entre los paréntesis.

const comentario =
document.getElementById('comentarios').value.trim();

Función: guarda variables constantes segun lo que esrbio el usuario en el textArea, el .trim hace que se eliminen los espacios en blanco al principio y al final.

if (comentario === '')

Función: si no se escribe ninguna pregunta lanzara este mensaje de alert.