

Entrega 2 Semana 5

Fredy Andres Quevedo

Duvan Mauricio Rodríguez Penagos

Yonathan Alexis Montilla Combita

Politécnico Gran Colombiano

FrontEnd

Olarte John

Grupo B01

27 de septiembre de 2022

Contenido

Contenido

Introducción	3
Registro de usuario	4
About Us:	7
Login Usuario:	8
What's new:	11
Index:	14

Introducción

El presente documento tiene como objetivo la presentación del código que mejora visualmente la página de badstore.

Registro de usuario

El código del html para esta vista se basa en un conjunto de DIV anidado que permuten acomodar los input en un formato tipo tabla, es decir con filas y columnas.

```
<form id="formulario">
  <div class="row justify-content-center cuadro-formulario">
    <div class="row col-auto justify-content-center col-md-6 sombra">
      <div class="group col-md-6">
        <label>Full Name</label>
        <input id="nombre" type="text" required />
        <span class="highlight"></span>
        <span class="bar"></span>
      </div>

      <div class="group col-md-6">
        <label>Email Adress</label>
        <input id="correo" type="email" required />
        <span class="highlight"></span>
        <span class="bar"></span>
      </div>

      <div class="group col-md-6">
        <label>Password</label>
        <input id="clave" type="password" required />
        <span class="highlight"></span>
        <span class="bar"></span>
      </div>

      <div class="group col-md-6">
        <label>Recovery Key</label>
        <input id="codigoRecuperacion" type="text" required />
        <span class="highlight"></span>
        <span class="bar"></span>
      </div>
    </div>
  </div>
</form>
```

Resultado.

Full Name

Email Address

Password

Recovery Key

return

save

Como se puede observar se generan 3 filas y dos columnas con los datos correspondientes, para el diseño se creó su archivo independiente el cual da los diseños de input con borde ovalado, sombreado a la caja y algunos hover para dar animaciones a los botones.

```

1  body{
2      margin: 0;
3      padding: 0px;
4      font-family: 'Roboto', sans-serif;
5      height: 99vh;
6      width: 100%;
7      display: block;
8      position: relative;
9  }
10
11  input, label{
12      margin-left: auto;
13      margin-right: auto;
14      margin-bottom: 20px;
15      margin-top: 20px;
16      display: block;
17      width: 80%;
18      height: 35px;
19      align-content: center;
20      border-radius: 15px;
21      font-size: 20px;
22  }
  
```

Para dar un funcionamiento que simule el registro, se uso javascript para capturar información del formulario y rellenar un arreglo de usuarios

```

1  const formulario = document.forms["formulario"];
2
3
4  const guardar = () => {
5      const nombre = formulario.nombre.value;
6      const correo = formulario.correo.value;
7      const clave = formulario.clave.value;
8      const codigoRecuperacion = formulario.codigoRecuperacion.value;
9      usuarios.push({nombre, correo, clave, codigoRecuperacion});
10     console.log(usuarios);
11     reiniciarValores();
12     alert(`se ha creado el usuario ${nombre}`);
13 }
14
15 const reiniciarValores = () => {
16     formulario.nombre.value = '';
17     formulario.correo.value = '';
18     formulario.clave.value = '';
19     formulario.codigoRecuperacion.value = '';
20 }
21

```

La variable a la que se le hace push "usuarios", proviene de la clase independiente "usuarios.json.js", la cual almacena en memoria un listado de usuarios, estos se pueden ver reflejados en el log, por defecto solo hay uno precargado, sin embargo cuando vamos guardando más usuarios el arreglo se va cargando

```

2: Object { nombre: "duvan", correo: "dasd", codigoRecuperacion: "dsasd", _ }
length: 3
<prototype>: Array []

Array(4) [ {_, {_, {_, {_} } ]
  0: Object { nombre: "duvan", correo: "duvan@gmail.com", clave: "123", _ }
  1: Object { nombre: "duvan", correo: "", clave: "wekD6ecbPLf66CK", _ }
  2: Object { nombre: "duvan", correo: "dasd", codigoRecuperacion: "dsasd", _ }
  3: Object { nombre: "javier", correo: "ddfsqmil.com", clave: "", _ }
length: 4
<prototype>: Array []

```

El diseño de esta vista a comparación del anterior es más sencillo de manejar debido a su independencia del resto de módulos, también contiene un diseño de un formulario sencillamente lógico con sus botones de guardar y regresar claramente visibles.

About Us:

Se realiza un diseño que sea más amigable con el usuario y de fácil visualización para que tengan acceso a la información más relevante de la página.

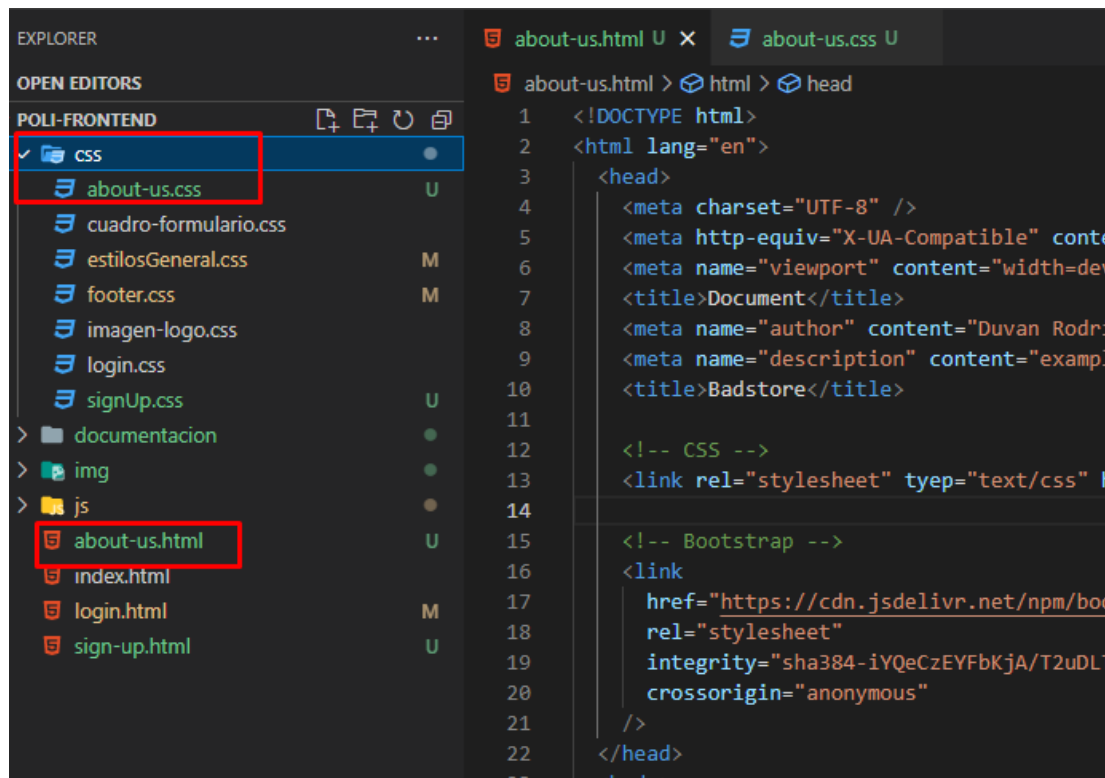
About Us

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.



Para este desarrollo se usaron las cards de bootstrap más css personalizado, cada carta quiere decir la información a la que tendrá acceso.

El código usado se dejó de manera independiente, es decir que tiene su propio css, para este caso que solo era diseño visual no se utilizó javascript.



Login Usuario:

En el caso del Login se permite la visualización en los distintos tipos de pantalla

 **BadStore**

Sing in

or



Sing in

Sing in

or

Sing up

Se programo para que se ajuste a los distintos tipos de pantalla manteniendo el diseño de los mockups realizados con anterioridad:

```

<body>
  <div class="flex-container">
    <header class="flex-item text-logo">
      
      <a href="index.html">
        <h1>BadStore</h1>
      </a>
    </header>
    <main class="flex-item">
      <div class="login">
        <h1>Sing in</h1>
        <form>
          <input type="text" placeholder="insert your email">
          <input type="password" placeholder="insert your password">
          <input type="submit" value="Sing in">
        </form>
        <h2>or</h2>
        <button onclick="registrarse()">Sing up</button>
      </div>
    </main>
    <script src="./js/redirigir.js"></script>
  </body>

```

Se manejo html5 para su contenido y mediante css se realizaron los ajustes de diseño y compatibilidad con las demás pantallas mediante media queries y ordenando todo mediante carpetas en la cual las img y css están separadas para poder ser reutilizadas.

```

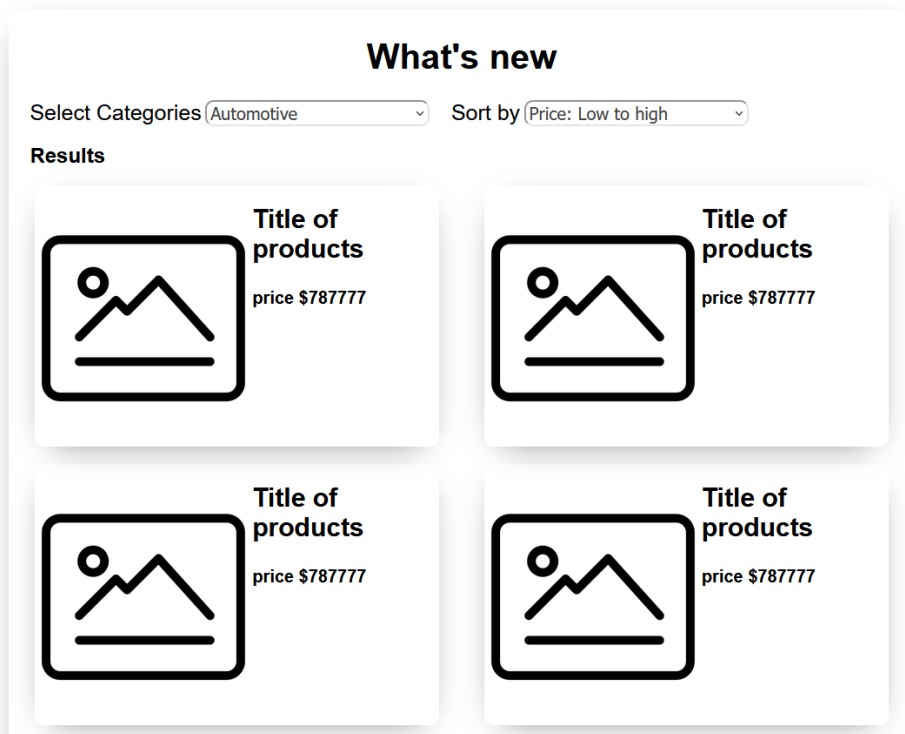
@media only screen and (max-width: 1000px) {
  .flex-item:nth-child(2) {
    width: 50%;
  }
}

@media only screen and (max-width: 700px) {
  .flex-item:nth-child(2) {
    width: 80%;
    box-shadow: none;
  }
}

```

What's new:

Para la codificación de los What's new se consideró el uso de grid en css para permitir a futuro rellenar cada producto desde DB y que automáticamente se acomode de igual forma se maneja un diseño responsive para las diferentes pantallas:



Como podemos observar en el diseño para dispositivos con una resolución superior a 800px

Se mostrará este diseño en el cual se contempló que mediante css grid que cada uno de los productos este encapsulado en un div en el cual se aplican las configuraciones del grid

```
<div class="parent">
  <div>
    
    <h3>Title of products</h3>
    <h4>price $787777</h4>
  </div>
  <div>
    
    <h3>Title of products</h3>
    <h4>price $787777</h4>
  </div>
</div>
```

El cual permite que cualquier nuevo producto cumpla con este estándar y sea responsive sin importar la pantalla

What's new

Select Categories

Automotive

Sort by

Price: Low to high

Results

Title of products
price \$787777



Title of products
price \$787777

Como podemos ver para adaptar a cada pantalla es solo modificar el grid de la clase parent en la cual se contemplan los 3 items que este tiene, en caso de necesitar una nueva adaptación gracias a esta implementación el cambio es relativamente sencillo.

```
.parent img {
  grid-area: 1 / 1 / 3 / 2;
  width: 250px;
  height: 250px;
  margin: auto;
}

.parent h3 {
  grid-area: 1 / 2 / 2 / 3;
  font-size: 30px;
  height: 40px;
  margin-top: 10%;
  margin-right: 5%;
}

.parent h4 {
  grid-area: 2 / 2 / 3 / 3;
  font-size: 20px;
  margin-top: -15%;
  margin-right: 5%;
}
```

Index:

Se diseñó el siguiente INDEX de una manera sencilla, donde los botones de la parte superior envían a un usuario a las diferentes opciones que se van a manejar en la página, donde los logos se pueden configurar de la misma manera como hipervínculos que presenten otros ingresos a información requerida.



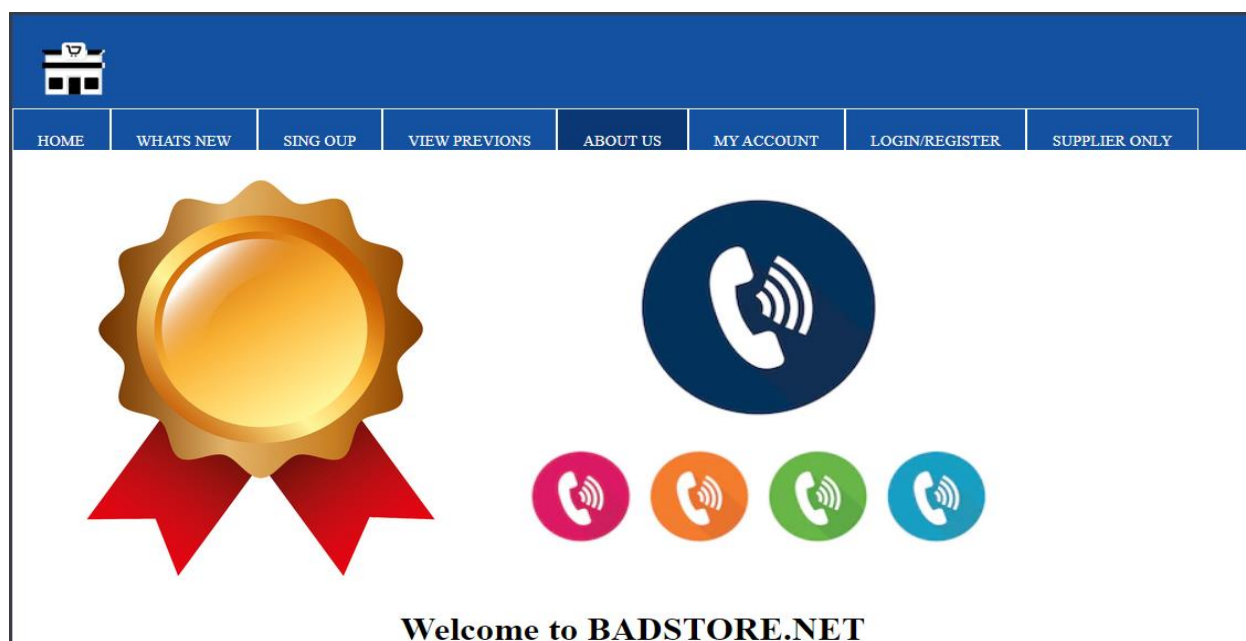
Para el diseño del índice se realizó mediante la herramienta visual studio, donde en el código se usaron las etiquetas head, title, link para poder enlazar con los estilos.

Una etiqueta body div para el container el logo la imagen, nav class y ul class para la presentación de las opciones que funcionan como vínculos para el desplazamiento a las diferentes opciones que se tienen para los usuarios finales.

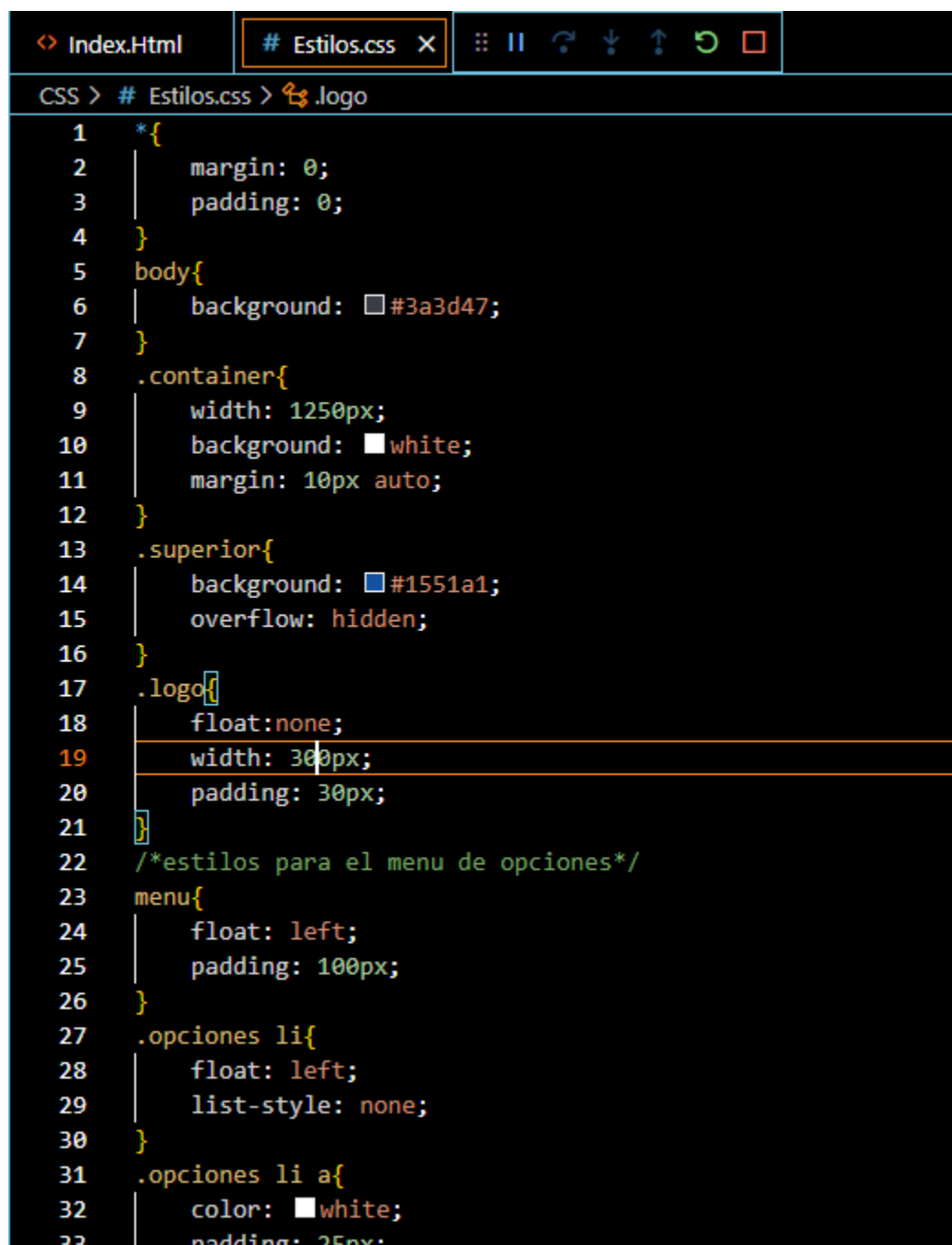
```

<> Index.Html x # Estilos.css
<> Index.Html > html > body > div.container > section.cuerpo > h1.titulo > center > h1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF 8">
5      <title>BadStore.Net</title>
6      <link rel="stylesheet" href="CSS/Estilos.css">
7  </head>
8  <body>
9      <div class="container">
10         <header class="superior">
11             <div class="logo">
12                 
13             </div>
14             <nav class="menu">
15                 <ul class="opciones">
16                     <li><a href="">HOME</a></li>
17                     <li><a href="">WHATS NEW</a></li>
18                     <li><a href="">SING OUP</a></li>
19                     <li><a href="">VIEW PREVIONS</a></li>
20                     <li><a href="">ABOUT US</a></li>
21                     <li><a href="">MY ACCOUNT</a></li>
22                     <li><a href="">LOGIN/REGISTER</a></li>
23                     <li><a href="">SUPPLIER ONLY</a></li>
24                 </ul>
25             </nav>
26         </header>
27         <section class="slider">
28             
29             
30         </section>
31         <section class="cuerpo">
32             <h1 class="titulo">

```



En cuanto a los estilos se aplicaron para el body, los container, la parte superior, lo que tiene que ver con los logos se le aplicaron etiquetas background, width, margin, float, padding para darle los ajustes y el diseño un poco más amable a los que naveguen por la pagina.



```

1  *{
2  |   margin: 0;
3  |   padding: 0;
4  | }
5  body{
6  |   background: #3a3d47;
7  | }
8  .container{
9  |   width: 1250px;
10 |   background: white;
11 |   margin: 10px auto;
12 | }
13 .superior{
14 |   background: #1551a1;
15 |   overflow: hidden;
16 | }
17 .logo{
18 |   float: none;
19 |   width: 300px;
20 |   padding: 30px;
21 | }
22 /*estilos para el menu de opciones*/
23 menu{
24 |   float: left;
25 |   padding: 100px;
26 | }
27 .opciones li{
28 |   float: left;
29 |   list-style: none;
30 | }
31 .opciones li a{
32 |   color: white;
33 |   padding: 25px;

```

Conclusión de acuerdo a la primera entrega se realizó un Index un poco mejorado y donde se va a explorar más herramientas que permitan completar el diseño de la página donde se requiere integrar con las demás opciones como por ejemplo el login del usuario.