

## Criterio E: Desarrollo del producto

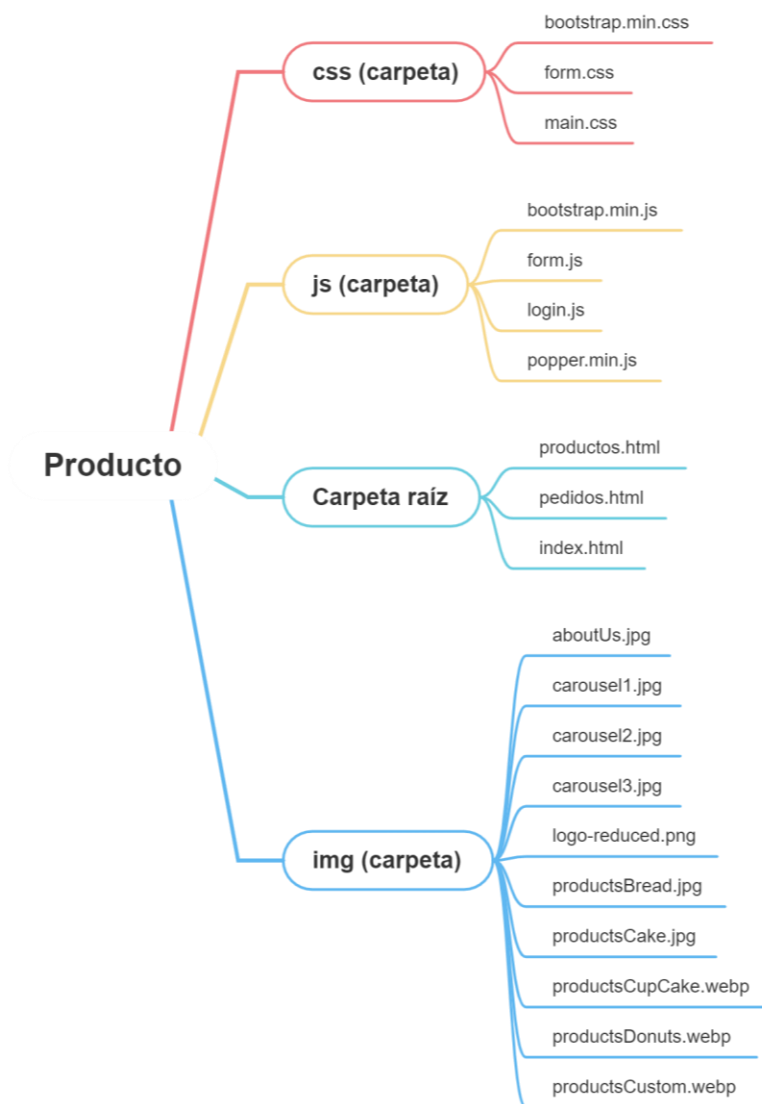
### Técnicas básicas

- Navegación (enlaces externos e internos)
- Uso de encabezados
- Uso de tablas

### Técnicas adecuadas

- Manipulación de código HTML, y JavaScript
- Hojas de estilo en cascada
- Integración de la página a una base de datos

### Estructura del producto



## Técnicas adecuadas

### 1. Manipulación de código HTML

#### a. Barra de navegación

Inicio de la etiqueta nav que contiene a los elementos del header. Contiene la clase navbar-dark para darle estilos css.

Etiqueta Anchor conteniendo el logo con data-bs-toggle="modal" para mostrar un modal de login al presionarse en ella.

Etiqueta Button con clase navbar-toggler que permite insertar un botón de navegación al reducirse el tamaño del viewport.

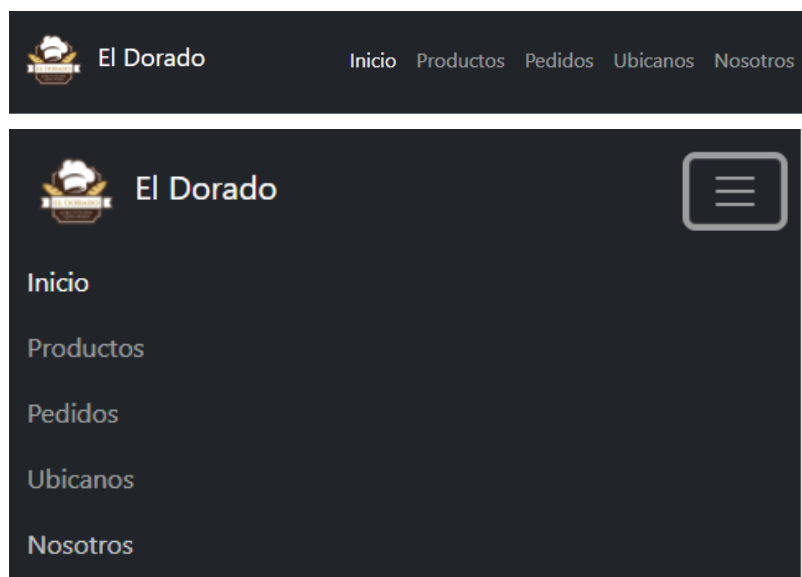
Div contenedor de una lista desordenada que se colapsa de acuerdo a la resolución de pantalla. Asimismo, está presente una lista desordenada con etiquetas li que declaran los elementos en la lista permitiendo un orden semántico en HTML

```

- Navbar -->
<nav class="navbar navbar-expand-sm navbar-dark bg-dark" id="navbar">
  <div class="container-fluid">
    <!-- Logo -->
    <a class="navbar-brand" href="#" data-bs-toggle="modal" data-bs-target="#login-modal">
      
      El Dorado
    </a>
    <!-- Toggler -->
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navTog">
      #navTog
    </button>
    <!-- Nav items -->
    <div class="collapse navbar-collapse" id="navTog">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="">Inicio</a>
        </li>
        <li class="nav-item"><a class="nav-link" href="productos.html">Productos</a></li>
        <li class="nav-item"><a class="nav-link" href="pedidos.html">Pedidos</a></li>
        <li class="nav-item"><a class="nav-link" href="#ubicanos">Ubicanos</a></li>
        <li class="nav-item"><a class="nav-link" href="#nosotros">Nosotros</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Esta sección permitirá que los usuarios puedan navegar con mayor facilidad entre las diferentes páginas. Permitiendo que la experiencia de usuario sea buena y que la velocidad de búsqueda sea menor.

#### a. Resultado



## 2. Hojas de estilo en cascada

### a. Carrusel de imágenes

El texto del carrusel no debe ocupar un espacio en el DOM, estará una capa adelante en el z-index para que se muestre encima de las imágenes. Finalmente tendrá un tamaño de 50px y color blanco.

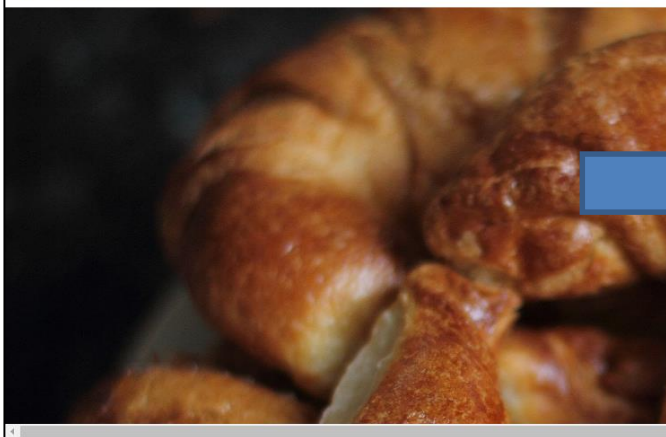
Las imágenes del carrusel estarán posicionadas sin cambiar la relación de aspecto así como tener un contraste bajo y un brillo bajo para que la letra se muestre correctamente.

```
16  /* Carousel */
17
18  .carousel-texto {
19    position: absolute;
20    margin: 17% auto;
21    width: 100%;
22    z-index: 2;
23    text-align: center;
24    color: white;
25    font-size: 50px;
26  }
27
28  .carousel-item {
29    height: 600px;
30  }
31
32  .carousel-img {
33    position: absolute;
34    min-height: 0px;
35    filter: contrast(90%);
36    filter: brightness(0.7);
37  }
```

Los estilos CSS permiten que se construya una página llamativa para el usuario para evitar que por cuestiones de diseño este abandone la página.

### a. Resultado

Panadería y pastelería  
El dorado



### 3. Integración de base de datos

#### a. Código Javascript

Conectarse a la base de datos Firestore y seleccionar el formulario y el Spinner de envío por su ID.

Escuchar el evento "submit" y lanzar una función que guarde registros en la base de datos.

```
1 const db = firebase.firestore(); // Database
2 const taskform = document.getElementById('task-form'); // Form
3 const spinner = document.getElementById('sending-spinner').style; // Loading spinner style
4
5 // Wait for 'submit'
6
7 taskform.addEventListener('submit', async(e) => {
8
9   e.preventDefault();
10
11   spinner.display = 'inline-block'; // show spinner
12   await saveReg(); // Save reg
13 });
```

Seleccionar los datos a registrar

En la colección registros se crea un documento con los datos del formulario

```
19 function saveReg() {
20
21   const name = taskform['nombre'].value;
22   const mail = taskform['Email'].value;
23   const phone = taskform['telefono'].value;
24   const type = taskform['tipo'].value;
25   const date = taskform['fecha'].value;
26   const amount = taskform['cantidad'].value;
27   const description = taskform['task-description'].value;
28
29   db.collection('registros').doc().set({ // Save data in 'registros'
30     name,
31     mail,
32     phone,
33     type,
34     date,
35     amount,
36     description,
37   })
38   .then(() => { // Wait and create a thanks message
39     greetings();
40   });
41
42 }
```

La base de datos es una parte importante de la página debido a que con ella se podrán guardar los registros de los pedidos que los usuarios realicen mediante la web. Asimismo, permite que el señor Edus tenga un control de todas las listas en un panel que se encarga de agrupar todos los registros en tiempo real.

#### a. Resultado

🏠 > registros > QBLFX81dAfqY...		
📁 el-dorado-bda85	📁 registros	📁 QBLFX81dAfqYzoZLccGj
+ Start collection	+ Add document	+ Start collection
registros >	QBLFX81dAfqYzoZLccGj >	+ Add field
		amount: "6" date: "2021-07-08" description: "Pedido prueba" mail: "correo@ejemplo.com" name: "Angel" phone: "123456789" type: "Postres"

**Número de palabras:** 403

#### **Bibliografía:**

- Otto, M. (2021). *Introduction*. Getbootstrap.com. Consultado el 22 de julio del 2021. Recuperado de <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
- *Cloud Firestore | Firebase*. (2021). Firebase. Consultado el 22 de julio del 202. Recuperado de <https://firebase.google.com/docs/firestore>