**PROYECTO002>ADMINISTRACIÓN DE PRODUCTOS**

**.HTML**

<div class="body">

  <h1>Administración de articulos</h1>

  <div class="table">

    <table>

      <tr>

        <td>Codigo</td>

        <td>Descripcion</td>

        <td>Precio</td>

        <td>Borrar</td>

        <td>Seleccionar</td>

      </tr>

      <tr \*ngFor="let producto of productos; index as i">

        <td>{{ producto.codigo }}</td>

        <td>{{ producto.descripcion }}</td>

        <td>{{ producto.precio }}</td>

        <td><button (click)="**deletePrice**(i)">Borrar?</button></td>

        <td><Button (click)="**seleccionar**(producto.codigo, producto.descripcion, producto.precio)">Seleccionar</Button>

      </tr>

    </table>

  </div>

  <div class="form">

    <label for="">Codigo: <input type="number" [(ngModel)]="art.codigo" /> <br></label>

    <label>Descripción: <input type="text" [(ngModel)]="art.descripcion" /><br/></label>

    <label>Precio: <input type="number" [(ngModel)]="art.precio" /><br/></label>

    <button (click)="**agregarProducto**()">Agregar</button>

    <button (click)="**modificarProducto**(art.codigo, art.descripcion,art.precio)">Modificar</button>

  </div>

  <div class="log">

    <h2>Log</h2>

    <p class="log-text">{{log}}</p>

  </div>

</div>

**.ts**

import { Component } from '@angular/core';

@**Component**({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'proyecto002';

  log= 'Esperando cambios';

  art={

    codigo:0,

    descripcion:"",

    precio:0

  }

  productos: any[] = [{codigo:1, descripcion:'papas', precio:10.55},

  {codigo:2, descripcion:'manzanas', precio:12.1},

  {codigo:3, descripcion:'melon', precio:52.3},

  {codigo:4, descripcion:'cebollas', precio:17},

  {codigo:5, descripcion:'calabaza', precio:20}];

**agregarProducto**() {

    const existeCodigo = this.productos.**some**(producto => producto.codigo == this.art.codigo);

    /\* se utiliza para comprobar si existe al menos un producto en el arreglo cuyo código coincide con el código del producto almacenado en this.art.codigo soltando un boolean \*/

    if (existeCodigo) {

      console.**log**('El código de producto ya existe.');

      this.log="El código de producto ya existe.";

      return;

    }

    const producto = {codigo: this.art.codigo, descripcion:this.art.descripcion,precio : this.art.precio };

    this.productos.**push**(producto);

    /\* Push es una funcion que permite agregar el valor al final del arreglo\*/

    this.log="Producto agregado correctamente";

    this.art.codigo=0;

    this.art.descripcion = "";

    this.art.precio =0;

}

**deletePrice**(index: number){

    this.productos.**splice**(index,1 );

    /\* permite modificar el contenido de un arreglo a partir de

    especificar la posicion del elemento y el uno indica cuantos valores se desean eliminar, osea solo 1  \*/

    this.log="Se elimino un producto.";

  }

**seleccionar**(codigo:number, descripcion:string, price:number){

    this.art.codigo = codigo;

    this.art.descripcion = descripcion;

    this.art.precio = price;

    this.log="Se selecciono un producto.";

  }

**modificarProducto**(codigo: number, nuevaDescripcion: string, nuevoPrecio: number) {

    const index = this.productos.**findIndex**(producto => producto.codigo == codigo);

    /\* a partir de la función findIndex esto permite buscar en el arreglo algun valor que

    coincida con el codigo que tenemos para ser analizado \*/

    if (index === -1) {

      console.**log**('El código de producto no existe.');

      this.log="Este codigo de producto no existe.";

      return;

    }

    this.productos[index].descripcion = nuevaDescripcion;

    this.productos[index].precio = nuevoPrecio;

    this.log="Se Modifico un producto.";

    this.art.codigo=0;

    this.art.descripcion = "";

    this.art.precio =0;

  }

}

**.css**

table {

  border: 2px solid rgb(27, 76, 111);

  margin-bottom: 10px;

  color:rgb(0, 0, 0);

  font-weight: bold;

}

th, td {

  text-align: left;

  vertical-align: top;

  border: 2px solid rgb(27, 76, 111);

  border-spacing: 0;

}

input {

  margin-bottom: 10px;

}

h1{

  text-transform: uppercase;

  font-weight: bold;

}

button{

  background-color: black;

  color:white;

  border-radius: 20px;

}

.form{

  margin-top: 30px;

}

.log-text{

  width: 400px;

  background-color: black;

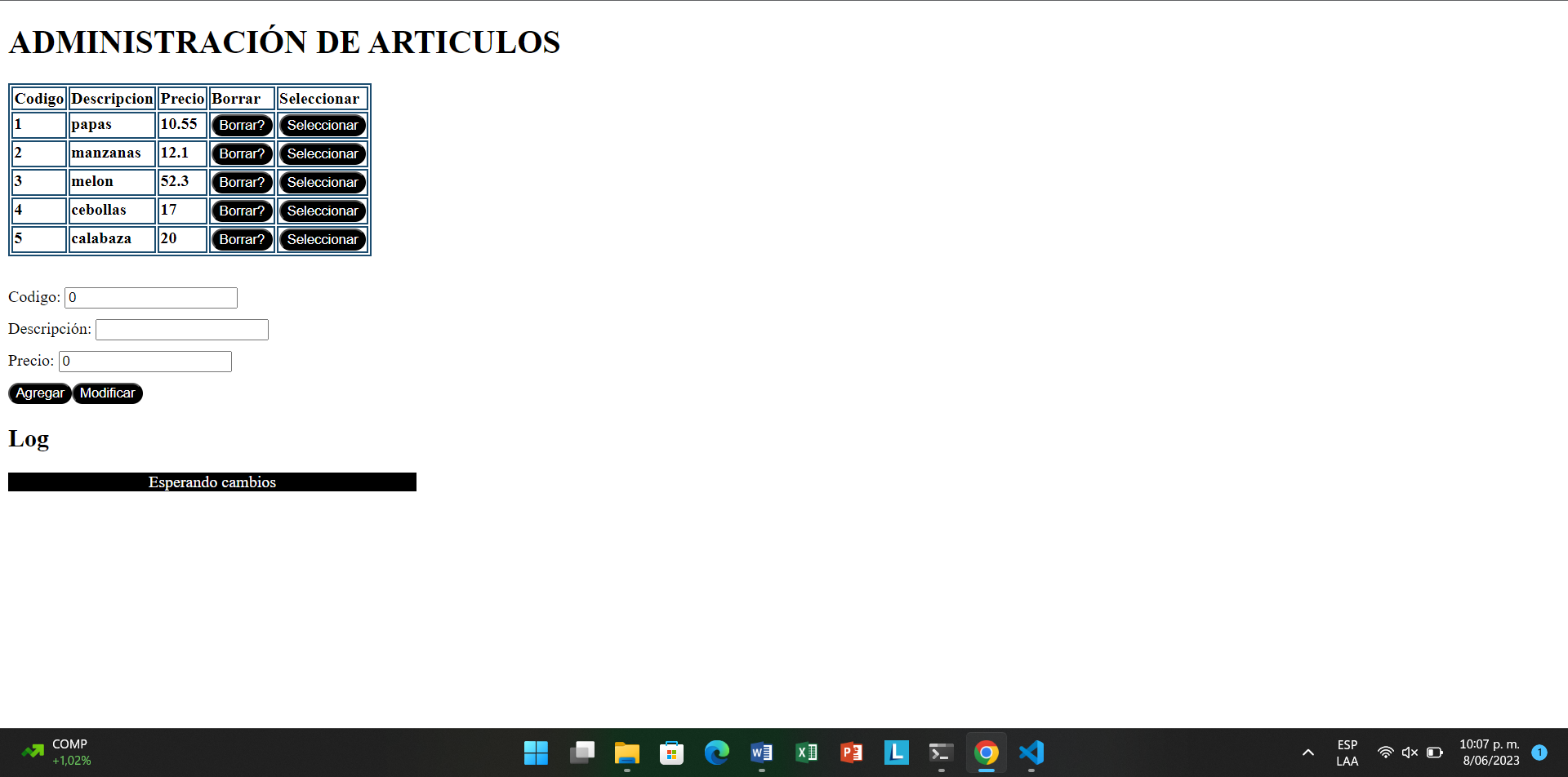
  color: white;

  align-items: center;

  text-align: center;

}

**Vista**

****