

## Ejercicio Guiado 10 - JS

En este ejercicio utilizarán Bootstrap e implementarán un diseño adecuado, para que el diseño del ejercicio sea más atractivo, vamos a utilizar **Bootstrap** para darle un estilo moderno y agradable. Esta es una biblioteca que facilita el diseño responsivo y atractivo con clases CSS predefinidas. En la práctica se muestra un código de ejemplo que puede servirles de guía para que ustedes lo realicen con Bootstrap y mejoren el diseño.

### Pasos para implementar Bootstrap:

1. **Incluir Bootstrap en el proyecto.**
2. **Agregar estilos y estructuras con clases de Bootstrap.**
3. **Optimizar el diseño para que sea responsivo** y se vea bien en diferentes tamaños de pantalla.

### Teoría: Manipulación del DOM

El **DOM** (Document Object Model) es una representación en árbol de todos los elementos HTML en una página web. Permite que JavaScript interactúe con estos elementos, modificándolos, eliminándolos o agregando otros nuevos. Algunos de los métodos comunes para manipular el DOM incluyen:

- `document.getElementById(id)`: Selecciona un elemento por su `id`.
- `document.querySelector(selector)`: Selecciona el primer elemento que coincida con el selector CSS dado.
- `element.innerHTML`: Cambia el contenido HTML dentro de un elemento.
- `element.style`: Permite modificar el estilo CSS directamente desde JavaScript.
- `element.classList.add()` / `remove()` / `toggle()`: Agrega, elimina o alterna clases en un elemento.
- `document.createElement(tag)`: Crea un nuevo elemento HTML.
- `element.appendChild()`: Agrega un nodo hijo a un elemento.
- `element.remove()`: Elimina un elemento del DOM.

### Ejercicio Práctico: Manipulación del DOM

En este ejercicio, haremos una pequeña aplicación web que permite agregar y eliminar elementos de una lista dinámica. Cada vez que un usuario escriba un elemento en un campo de texto y haga clic en el botón "Agregar", el elemento se agregará a una lista. Además, cada elemento de la lista tendrá un botón para eliminarlo.

#### HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Manipulación del DOM</title>
  <style>
    #lista {
```

## Ejercicio Guiado 10 - JS

```
        list-style-type: none;
        padding: 0;
    }
    .elemento {
        display: flex;
        justify-content: space-between;
        padding: 8px;
        background-color: #f0f0f0;
        margin-bottom: 5px;
        border-radius: 4px;
    }
</style>
</head>
<body>

    <h1>Agregar y Eliminar Elementos</h1>

    <input type="text" id="nuevoElemento" placeholder="Escribe algo">
    <button id="agregarBtn">Agregar</button>

    <ul id="lista"></ul>

    <script src="app.js"></script>

</body>
</html>
```

### JavaScript (*app.js*)

```
// Seleccionar los elementos importantes del DOM
const input = document.getElementById('nuevoElemento');
const botonAgregar = document.getElementById('agregarBtn');
const lista = document.getElementById('lista');

// Función para agregar un nuevo elemento a la lista
function agregarElemento() {
    const texto = input.value.trim(); // Obtiene el valor del input y
    elimina espacios innecesarios

    if (texto !== '') {
        // Crear un nuevo elemento 'li' y un botón de eliminar
        const li = document.createElement('li');
        li.classList.add('elemento'); // Añadimos una clase al li

        const textoNode = document.createTextNode(texto);
        li.appendChild(textoNode); // Agrega el texto al li

        // Crear el botón de eliminar
        const botonEliminar = document.createElement('button');
        botonEliminar.textContent = 'Eliminar';
        botonEliminar.addEventListener('click', function() {
            li.remove(); // Eliminar el li al hacer clic en el botón de
eliminar
        });

        // Añadir el botón al li
        li.appendChild(botonEliminar);
    }
}
```

## Ejercicio Guiado 10 - JS

```
// Agregar el li a la lista
lista.appendChild(li);

// Limpiar el campo de texto
input.value = '';
} else {
    alert('Escribe algo para agregar a la lista.');
```

```
}

// Asignar la función al botón de agregar
botonAgregar.addEventListener('click', agregarElemento);
```

### Explicación del Ejercicio:

#### 1. Seleccionar elementos del DOM:

- Utilizamos `document.getElementById` para obtener el campo de texto, el botón y la lista donde se agregarán los elementos.

#### 2. Agregar elementos a la lista:

- Cuando el usuario hace clic en el botón "Agregar", se obtiene el valor del campo de texto y se crea un nuevo elemento `li`.
- Si el valor no está vacío, se añade el texto como nodo de texto dentro del `li`, y luego se agrega este nuevo `li` al `ul` existente.

#### 3. Eliminar elementos:

- Cada `li` que se agrega a la lista también contiene un botón de "Eliminar". Cuando el usuario hace clic en ese botón, el elemento `li` se elimina del DOM con el método `remove()`.

#### 4. Limpieza del campo de texto:

- Una vez que se agrega un nuevo elemento a la lista, se limpia el campo de texto para que el usuario pueda seguir añadiendo más elementos.