Sesión 6: Integrando JavaScript y Bootstrap para Interactividad en Páginas Web

Duración: 3 horas

Objetivo de la clase:

Al finalizar esta sesión, los estudiantes serán capaces de manipular el DOM con **JavaScript**, crear interacciones dinámicas con eventos, y gestionar archivos de **HTML**, **CSS**, y **JavaScript** separados, integrando **Bootstrap** para estilizar sus páginas de manera eficiente.

---

1. Revisión de la Sesión Anterior (15 minutos)

- Repaso de la creación de layouts con **Flexbox** y **Grid Layout** utilizando **Bootstrap**.
- Verificación de las tareas anteriores, viendo la implementación de las técnicas de diseño adaptativo en los proyectos de los estudiantes.

¿Alguna duda antes de comenzar con los nuevos temas de JavaScript y Bootstrap?

---

2. Introducción a JavaScript, el DOM y Bootstrap (45 minutos)

JavaScript es el lenguaje de programación que permite la manipulación dinámica del contenido de una página web, proporcionando interactividad mediante la manipulación del **DOM (Document Object Model)**.

2.1. ¿Qué es el DOM?

El DOM es la representación estructurada de los elementos HTML en forma de árbol jerárquico. Con JavaScript podemos:

- Acceder a los elementos del DOM utilizando selectores como getElementById, querySelector y querySelectorAll.
- Modificar el contenido, estilos y atributos de los elementos.

2.2. Archivos separados para una estructura más limpia

En esta sesión, vamos a trabajar con **HTML**, **CSS**, y **JavaScript** en archivos separados para mantener una estructura más organizada y escalable.

2.3. Insertar y enlazar JavaScript y CSS

En el archivo **HTML**, enlazamos tanto **Bootstrap** como nuestros archivos personalizados de **CSS** y **JavaScript**.

- **Enlazar CSS (Bootstrap y personalizado)**:

```
<link href="styles.css" rel="stylesheet">
```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootst
rap.min.css" rel="stylesheet">
```

- **Enlazar JavaScript personalizado**:

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

3. Eventos en JavaScript con Interacción Dinámica (45 minutos)

Eventos son las acciones que ocurren en una página web cuando el usuario interactúa con ella (clics, escritura, desplazamiento). JavaScript puede capturar estos eventos y ejecutar funciones en respuesta.

3.1. Crear una lista dinámica con eventos

Vamos a crear una lista interactiva que permita agregar elementos, eliminarlos y alternar su visibilidad, todo utilizando **JavaScript**.

3.2. Asignar eventos a elementos

- **addEventListener**: Asigna un evento a un elemento seleccionado en el DOM.

```
const boton = document.querySelector('#miBoton');
```

```
boton.addEventListener('click', function() {
  alert('Botón clickeado');
});
```

- **Eventos comunes en formularios**: click, submit, focus, blur.

4. Ejercicio Práctico: Crear una Lista Dinámica con JavaScript, CSS y Bootstrap (45 minutos)

En este ejercicio, los estudiantes crearán una página web que incluya:

1. Un campo de entrada para agregar nuevos elementos a una lista.
2. Un botón para alternar la visibilidad de la lista.
3. Un botón "Eliminar" en cada elemento de la lista que permite eliminar ese elemento.
4. Uso de **Bootstrap** para el diseño de la página.

Instrucciones del ejercicio práctico:

4.1. Archivo HTML (lista-dinamica.html)

El archivo HTML contiene la estructura básica del formulario, la lista y los botones.

```
<!DOCTYPE html>
<html lang="es">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista Dinámica con JavaScript</title>
```

```
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootst
rap.min.css" rel="stylesheet">
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h1 class="mb-4">Lista Dinámica con JavaScript</h1>
    <div class="mb-3">
      <input type="text" id="itemInput" class="form-control"
placeholder="Ingresa un nuevo elemento">
      <button id="addButton" class="btn btn-primary mt-2">Agregar
a la lista</button>
      <button id="toggleButton" class="btn btn-secondary mt-
2">Ocultar Lista</button>
    </div>
    <ul id="miLista" class="list-group">
      <!-- Los elementos de la lista se agregarán aquí
dinámicamente -->
    </ul>
  </div>
  <script src="script.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

4.2. Archivo JavaScript (script.js)

El archivo **JavaScript** controla la lógica de agregar, eliminar y alternar la visibilidad de los elementos de la lista.

```
// Función para agregar un nuevo elemento a la lista
function agregarElemento() {
  const input = document.querySelector('#itemInput');
  const lista = document.querySelector('#miLista');

  if (input.value.trim() !== '') {
    const nuevoItem = document.createElement('li');
    nuevoItem.className = 'list-group-item d-flex justify-content-
between align-items-center';
    nuevoItem.innerText = input.value;

    // Botón para eliminar el elemento
    const botonEliminar = document.createElement('button');
    botonEliminar.className = 'btn btn-danger btn-sm';
    botonEliminar.innerText = 'Eliminar';
    botonEliminar.onclick = function() {
      lista.removeChild(nuevoItem);
    };

    nuevoItem.appendChild(botonEliminar);
    lista.appendChild(nuevoItem);
    input.value = '';
```

```
  } else {
    alert('Por favor, ingresa un texto para el nuevo elemento.');
  }
}

// Función para alternar la visibilidad de la lista
function alternarLista() {
  const lista = document.querySelector('#miLista');
  const boton = document.querySelector('#toggleButton');

  if (lista.style.display === 'none') {
    lista.style.display = 'block';
    boton.innerText = 'Ocultar Lista';
  } else {
    lista.style.display = 'none';
    boton.innerText = 'Mostrar Lista';
  }
}

// Asignar eventos al cargar el documento
document.addEventListener('DOMContentLoaded', function() {
  document.querySelector('#addButton').addEventListener('click',
agregarElemento);

document.querySelector('#toggleButton').addEventListener('click',
alternarLista);
});
```

4.3. Archivo CSS (styles.css)

El archivo **CSS** incluye estilos personalizados adicionales para mejorar la presentación, además de los estilos de **Bootstrap**.

```
body {
  background-color: #f8f9fa; /* Color de fondo claro para la página */
}

h1 {
  color: #343a40; /* Color oscuro para el título */
  text-align: center;
}

#itemInput {
  border: 2px solid #6c757d; /* Borde personalizado para el campo de entrada */
}

#toggleButton {
  margin-left: 5px; /* Espaciado entre botones */
}

.list-group-item {
  background-color: #ffffff; /* Fondo blanco para los elementos de la lista */
```

```
  border: 1px solid #dee2e6; /* Borde suave */
}


.list-group-item button {
  margin-left: 10px; /* Espaciado entre el texto y el botón de eliminar */
}
```

5. Cierre y Preguntas (15 minutos)

Hoy hemos aprendido a integrar **JavaScript** y **Bootstrap** para agregar interactividad a nuestras páginas web. Hemos trabajado con archivos separados de HTML, CSS y JavaScript, siguiendo las mejores prácticas para mantener el código organizado y eficiente.