



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



Departamento de Computación  
Programación Integrativa de Componentes

Nombre: Carlos Pogo  
NRC: 16496

## Tarea 1.1: Estructura lógica de elementos web

Tema:

La aplicación que se realizó es una Lista de Tareas (To-Do List). Esta aplicación permite a los usuarios agregar, ver y eliminar tareas. Es una aplicación web sencilla que demuestra el manejo básico del DOM (Document Object Model) usando JavaScript, junto con HTML y CSS para la estructura y el diseño. Además, usaremos Bootstrap para facilitar el diseño responsivo.

Estructura:

Tarea 1.1

- css
  - styles.css
- js
  - app.js
- index.html

**Código:**

index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista de Tareas</title>
  <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
```

```

    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <div class="container">
        <div class="row mt-5">
            <div class="col-md-6 offset-md-3">
                <h1 class="text-center">Lista de Tareas</h1>
                <div class="input-group mb-3">
                    <input type="text" id="taskInput" class="form-control"
placeholder="Agregar una nueva tarea">
                    <div class="input-group-append">
                        <button class="btn btn-primary" id="addTaskBtn"
type="button">Agregar</button>
                    </div>
                </div>
                <ul class="list-group" id="taskList"></ul>
            </div>
        </div>
    </div>
    <script src="js/app.js"></script>
</body>
</html>

```

styles.css

```

body {
    background-color: #f8f9fa;
    font-family: Arial, sans-serif;
}

h1 {
    color: #343a40;
}

#taskList {
    margin-top: 20px;
}

.list-group-item {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.list-group-item .btn {

```

```

background-color: #dc3545;
border: none;
}

/*# sourceMappingURL=styles.css.map */

```

app.js

```

document.addEventListener('DOMContentLoaded', () => {
  const taskInput = document.getElementById('taskInput');
  const addTaskBtn = document.getElementById('addTaskBtn');
  const taskList = document.getElementById('taskList');

  addTaskBtn.addEventListener('click', () => {
    const taskText = taskInput.value.trim();
    if (taskText !== '') {
      addTask(taskText);
      taskInput.value = '';
    }
  });

  taskList.addEventListener('click', (event) => {
    if (event.target.classList.contains('delete-task')) {
      event.target.parentElement.remove();
    }
  });

  function addTask(taskText) {
    const li = document.createElement('li');
    li.className = 'list-group-item';
    li.innerHTML = `
      ${taskText}
      <button class="btn btn-danger btn-sm delete-task">Eliminar</button>
    `;
    taskList.appendChild(li);
  }
});

```

**Explicación:**

**index.html:**

1. Define la estructura básica de la aplicación usando HTML.
2. Incluye Bootstrap para estilos y una apariencia moderna.
3. Un campo de entrada (input) para agregar tareas y un botón (button) para agregar la tarea a la lista.
4. Una lista (ul) para mostrar las tareas.

#### **css/styles.css:**

1. Define los estilos CSS para la aplicación.
2. Configura el fondo y la fuente del cuerpo de la página.
3. Estilos para el título (h1) y la lista de tareas (#taskList).
4. Define cómo se deben ver las tareas individuales (.list-group-item) y el botón de eliminación.

#### **js/app.js:**

1. Espera a que el DOM se cargue completamente (DOMContentLoaded).
2. Obtiene referencias a los elementos de entrada, botón y lista.
3. Agrega una nueva tarea a la lista cuando se hace clic en el botón.
4. Permite eliminar una tarea de la lista cuando se hace clic en el botón de eliminación correspondiente.

#### **DOM:**

La manipulación del DOM (Document Object Model) es una parte fundamental de la programación web, especialmente cuando se usa JavaScript para crear aplicaciones interactivas. El DOM es una representación estructurada del documento HTML en la memoria del navegador, lo que permite a los lenguajes de programación (como JavaScript) interactuar y manipular el contenido, estructura y estilo del documento de manera dinámica.

#### **Conceptos Clave del DOM**

1. Nodo del Documento: Representa el documento completo (HTML).
2. Elementos del Documento: Representan las etiquetas HTML, como <div>, <ul>, <li>, etc.
3. Atributos del Documento: Representan los atributos de los elementos HTML, como id, class, etc.
4. Texto del Documento: Representa el contenido textual dentro de los elementos HTML.

## Explicación

### 1. Esperar a que el DOM esté Cargado

Usamos `DOMContentLoaded` para asegurarnos de que todo el HTML está cargado antes de intentar manipularlo.

### 2. Obtener Referencias

Utilizamos `document.getElementById` para obtener referencias directas a los elementos del DOM que vamos a manipular (campo de entrada, botón de agregar, y lista de tareas).

### 3. Agregar Tareas

a. *Evento click*: Cuando se hace clic en el botón, se ejecuta una función que:

- Obtiene el texto de la tarea del campo de entrada.
- Llama a `addTask` si el texto no está vacío.

b. *Función `addTask`*:

- Crea un nuevo elemento `li`.
- Establece su clase y contenido HTML.
- Lo agrega al final de `taskList`.

### 4. Eliminar Tareas

a. *Evento click*: Cuando se hace clic en cualquier parte de `taskList`:

- Se comprueba si el elemento clicado (`event.target`) es el botón de eliminación.
- Si es así, se elimina el `li` correspondiente (`event.target.parentElement`).

Link video:

[https://uespe-my.sharepoint.com/:v:/g/personal/cepogo\\_espe\\_edu\\_ec/EdlRryA8He9JpftVx4JuKTsBe-MHvU6bPkSGl-OkcDcUYg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoiJPbmVEcmI2ZUZvc kJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6In ZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=SEM1nV](https://uespe-my.sharepoint.com/:v:/g/personal/cepogo_espe_edu_ec/EdlRryA8He9JpftVx4JuKTsBe-MHvU6bPkSGl-OkcDcUYg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoiJPbmVEcmI2ZUZvc kJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6In ZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=SEM1nV)