

```

/**
 * APLICACIÓN DE LISTA DE TAREAS
 * =====
 * Esta aplicación permite agregar y eliminar tareas dinámicamente
 * usando JavaScript DOM manipulation
 */

// Variable global para contar el número total de tareas
let taskCounter = 0;

// Obtener referencia al botón "Agregar Tarea" desde el DOM
let btn = document.getElementById('btn-add-task');

/**
 * EVENT LISTENER PRINCIPAL
 * Maneja el evento click del botón "Agregar Tarea"
 */
btn.addEventListener("click", function () {

    /* ===== CREACIÓN DE ELEMENTOS DOM ===== */

    // Crear elemento <li> que contendrá toda la tarea
    let li = document.createElement("li");

    // Asignar clases de Bootstrap para estilos (flexbox, spacing, etc.)
    li.className = "list-group-item d-flex justify-content-between align-items-start";

    // Crear elemento <span> para mostrar el texto de la tarea
    let span = document.createElement("span");
    span.className = "mt-1"; // Margen superior usando Bootstrap

    // Obtener referencia al contador de tareas en el DOM

```

```
let countTask = document.getElementById("countTask");

// Crear botón de eliminar para cada tarea
let deleteButton = document.createElement("button");
deleteButton.textContent = "Delete";
deleteButton.className = "btn btn-danger"; // Estilo rojo de Bootstrap

/* ===== REFERENCIAS A ELEMENTOS EXISTENTES ===== */

// Campo de entrada donde el usuario escribe la tarea
let taskInput = document.getElementById("taskInput");

// Contenedor donde se mostrarán todas las tareas
let taskList = document.getElementById("taskList");

/* ===== VALIDACIÓN DE ENTRADA ===== */

// Verificar si el campo de entrada está vacío
if (taskInput.value == "") {
    alert("Please, write a task!"); // Mostrar alerta de error
    taskInput.focus();           // Enfocar el campo de entrada
    return;                      // Salir de la función sin crear tarea
}

/* ===== CONSTRUCCIÓN Y RENDERIZADO DE LA TAREA ===== */

// Asignar el texto ingresado por el usuario al span
span.textContent = taskInput.value;

// Agregar el span (texto de la tarea) dentro del elemento li
li.appendChild(span);
```

```

// Incrementar el contador global de tareas
taskCounter++;

// Agregar el botón de eliminar al elemento li
li.appendChild(deleteButton);

// Actualizar el contador visual usando template literals
countTask.textContent = `Cantidad de Tareas : ${taskCounter} `;

// Agregar la nueva tarea (li completo) a la lista en el DOM
taskList.appendChild(li);

/* ===== LIMPIEZA DE INTERFAZ ===== */

// Limpiar el campo de entrada para la próxima tarea
taskInput.value = "";

// Mantener el foco en el campo de entrada para facilitar uso continuo
taskInput.focus();

/* ===== EVENT LISTENER PARA ELIMINACIÓN ===== */

/**
 * Maneja la eliminación individual de cada tarea
 * Se crea un listener único para cada botón "Delete"
 */
deleteButton.addEventListener('click', function () {

    // Mostrar confirmación antes de eliminar
    if (confirm("¿Estás seguro de que quieres eliminar esta tarea?")) {

```

```

// Decrementar el contador global
taskCounter--;

// Actualizar el contador visual (nota: texto en inglés vs español inconsistente)
countTask.textContent = "Number of tasks : " + taskCounter;

// Remover la tarea específica del DOM
taskList.removeChild(li);
}

// Si el usuario cancela, no se ejecuta nada (tarea permanece)
});

});

/**
 * ESTRUCTURA HTML ESPERADA:
 * =====
 * - #btn-add-task: Botón para agregar tareas
 * - #taskInput: Campo de entrada de texto
 * - #taskList: Contenedor de la lista de tareas
 * - #countTask: Elemento para mostrar el contador
 *
 * DEPENDENCIAS:
 * =====
 * - Bootstrap CSS (para las clases de estilo)
 *
 * FUNCIONALIDADES:
 * =====
 * ✓ Agregar tareas dinámicamente
 * ✓ Validación de entrada vacía
 * ✓ Contador automático de tareas

```

* ✓ Eliminación individual con confirmación

* ✓ Interfaz responsiva con Bootstrap

* ✓ Limpieza automática del formulario

* /