

## Tarea 021– Manejo DOM – Lista Tareas (I)

Los resultados de todas las tareas incorporarán, además del código fuente, los comentarios precisos y necesarios para su fácil comprensión. No escatimes esfuerzos en comentar el código, es una buena práctica para aprender, además de ser muy útil para modificaciones o reutilizaciones futuras.

Sería muy buena práctica añadir el título y el enunciado del ejercicio (como comentarios) al principio del código fuente.

Añade documentos en formato Word con capturas de la salida por pantalla (al ejecutar la página) si consideras que queda más clara la resolución del ejercicio.

Deberás entregar dos versiones del resultado:

1. **Un único documento en Word sin comprimir:** que contenga todas las líneas de código, y por orden, primero el código **html**, luego el código **css** y por último el código **js**. Este archivo servirá para que el profesor haga anotaciones y/o correcciones a los alumnos.
2. **Los archivos de código fuente:** en un único archivo, ya sea de extensión: **html**, **js**, o **css**. Si precisas entregar varios archivos, comprímelos en un único **zip**. (No admito rar).

El nombre del archivo entregado comenzará por tu número, tu nombre y seguido por TareaXXX. Ejemplo: **00FedericoTarea009.zip**

---

### Enunciado

---

Crea una aplicación web básica que permita a los usuarios gestionar una lista de tareas. La aplicación debe incluir los siguientes elementos y funcionalidades:

#### Requerimientos

##### 1. Interfaz de Usuario:

- Un campo de texto donde el usuario pueda escribir el nombre de una nueva tarea.
- Un botón "Agregar" que, al hacer clic, añada la tarea ingresada en el campo de texto a una lista de tareas en la página.
- Una lista donde se mostrarán todas las tareas que el usuario ha añadido.
- Un botón "Eliminar" junto a cada tarea que permita eliminar esa tarea específica de la lista.

##### 2. Funcionalidad:

- Cuando el usuario escribe el nombre de una tarea y presiona el botón "Agregar", el texto ingresado debe agregarse a la lista de tareas.
- El campo de entrada debe limpiarse automáticamente después de agregar una tarea.
- Si el campo de texto está vacío al intentar agregar una tarea, debe mostrarse un mensaje de alerta con `window.alert()` indicando que el usuario debe ingresar una tarea.
- Cada tarea en la lista debe tener un botón "Eliminar" que permita al usuario eliminar la tarea específica.

##### 3. Consideraciones Adicionales:

- No se permite recargar la página al agregar o eliminar tareas.

---

### Ayuda y Sugerencias:

---

Para resolver este ejercicio utiliza los apuntes de clase: DOM y Eventos. Por supuesto que chatGPT te lo hace en un santiamén, pero si no te esfuerzas y le dedicas mucho tiempo, no aprenderás.

Puedes partir de un código HTML como el siguiente y completar el resto de funcionalidades desde JavaScript:

```

<body>
  <h1>Lista de Tareas</h1>

  <!-- Campo de texto y botón para agregar tareas -->
  <input type="text" id="nuevaTarea" placeholder="Escribe una nueva tarea" />
  <button id="agregarTarea">Agregar</button>

  <!-- Contenedor de la lista de tareas -->
  <ul id="listaDeTareas" style="border: 1px solid;"> </ul>

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

</body>

```

- **Selección de elementos del DOM:** Selecciona los elementos input, button y ul del DOM utilizando getElementById.
- **Función agregarTarea:**
  - Se obtiene el texto ingresado por el usuario y se valida que no esté vacío.
  - Si el texto es válido, se crea un nuevo elemento <li> y se le asigna el texto de la tarea en alguno de sus atributos (textContent, innerHTML...)
  - Se crea un botón "Eliminar" que permite eliminar la tarea. Este botón se añade como hijo del <li>.
  - Añade un evento a este botón para que llame a la función eliminarTarea cada vez que se hace clic en el botón "Eliminar".
  - Finalmente, el nuevo elemento <li> se añade al <ul>, y el campo de entrada se limpia.
- **Eventos:**
  - click en el botón "Agregar": Llama a la función **agregarTarea** cada vez que se hace clic en el botón "Agregar".
  - click en cada botón "Eliminar": Llama a la función eliminarTarea. Dentro de la función podemos utilizar **this** para obtener cada uno de los botones. Observa el siguiente código:

```

function eliminarTarea(){
  console.log(this);
  const tarea= this.parentElement;
  console.log(tarea);
  listaDeTareas.removeChild(tarea);
}

```