

PROGRAMA DE ESTUDIOS

DESARROLLOS DE SISTEMAS DE INFORMACIÓN

**DESARROLLO FRONTEND
DE SISTEMAS DE WEB**

Tema

**MANEJO AVANZADO DE
EVENTOS EN JAVASCRIPT**

Manejo Avanzado de Eventos en JavaScript

Objetivo:

Dominar el uso avanzado de eventos en JavaScript, como la **delegación**, la **propagación**, la **prevención de comportamiento por defecto** y su uso eficiente para mejorar el rendimiento de aplicaciones web interactivas.

⌚ PARTE CONCEPTUAL (20 minutos)

❓ Pregunta detonadora:

¿Cómo podemos manejar múltiples eventos en una página sin afectar el rendimiento?

🔄 Repaso: Eventos básicos

javascript

```
document.getElementById("btn").addEventListener("click", () => {
  alert("¡Clic!");
});
```

❖ Delegación de eventos

Permite asignar un solo evento a un **elemento contenedor**, que maneja eventos para sus **hijos dinámicos**.

Ejemplo:

javascript

```
document.getElementById("lista").addEventListener("click", function(e) {
  if (e.target.tagName === "LI") {
    alert("Elemento clickeado: " + e.target.textContent);
```

```
}
```

```
});
```

- Ventaja: menos listeners → mejor rendimiento

Propagación de eventos

Fases de propagación:

- **Capturing phase:** va desde el <html> hasta el objetivo.
- **Target phase:** el evento alcanza el elemento.
- **Bubbling phase:** sube desde el objetivo hacia arriba.

Ejemplo:

```
javascript
```

```
element.addEventListener("click", function(e) {  
    e.stopPropagation(); // evita que el evento siga subiendo  
});
```

Prevención de comportamiento por defecto

Evita que un evento realice su acción nativa (como enviar un formulario o seguir un enlace).

```
javascript
```

```
form.addEventListener("submit", function(e) {  
    e.preventDefault();  
    // Validar manualmente antes de enviar  
});
```

 **Eliminación de eventos**

javascript

```
function saludar() {  
    alert("Hola");  
}  
  
btn.addEventListener("click", saludar);  
btn.removeEventListener("click", saludar);
```

**PARTE DEMOSTRATIVA (20 minutos)****1** **Lista dinámica con delegación**

HTML:

html

```
<ul id="lista">  
    <li>Tarea 1</li>  
    <li>Tarea 2</li>  
</ul>  

```

JS:

javascript

```
document.getElementById("agregar").addEventListener("click", () => {  
    const nueva = document.createElement("li");  
    nueva.textContent = "Tarea nueva";  
    document.getElementById("lista").appendChild(nueva);
```

```
});
```

```
document.getElementById("lista").addEventListener("click", function(e) {
  if (e.target.tagName === "LI") {
    e.target.style.textDecoration = "line-through";
  }
});
```

[2] Formulario con validación y prevención de recarga

html

```
<form id="mi-form">
  <input type="text" id="nombre" placeholder="Nombre">
  <button type="submit">Enviar</button>
</form>
<p id="error"></p>
```

javascript

```
document.getElementById("mi-form").addEventListener("submit", function(e) {
  e.preventDefault();
  const nombre = document.getElementById("nombre").value;
  if (nombre === "") {
    document.getElementById("error").textContent = "El nombre es obligatorio.";
  } else {
    document.getElementById("error").textContent = "";
    alert("Formulario enviado con: " + nombre);
  }
});
```

3 Cierre de menú al hacer clic fuera

javascript

```
const menu = document.getElementById("menu");

document.addEventListener("click", function(e) {
  if (!menu.contains(e.target)) {
    menu.style.display = "none";
  }
});
```

ACTIVIDAD PRÁCTICA INDIVIDUAL (20 minutos)

Desafío:

Crea una miniaplicación que incluya:

- **Formulario validado con eventos de input y submit.**
- **Menú desplegable que se cierre si se hace clic fuera.**
- **Lista de tareas dinámica con delegación de eventos para eliminar tareas.**

CIERRE DE SESIÓN (15 minutos)

Reflexión:

¿Por qué es importante manejar correctamente los eventos en páginas dinámicas?

Ideas clave:

- Mejor rendimiento (menos listeners)
- Código más limpio y reutilizable
- Mayor control sobre la interacción del usuario

 **Revisión:**

- ¿Funciona la delegación correctamente?
- ¿Los eventos se propagan como se espera?
- ¿Se previene el comportamiento por defecto correctamente?

 **Tarea:**

Implementar una lista de tareas dinámica en la que se puedan:

- Agregar elementos desde un input
- Eliminar tareas con delegación de eventos
- Usar event.preventDefault() en formularios

