

# Práctica 2: Dinamismo con JavaScript y el DOM

Trabajo en parejas

Curso 2025–2026

## Descripción

En esta práctica se amplía el trabajo realizado en la práctica 1, pero con un nuevo objetivo: **introducir dinamismo con JavaScript y el DOM**. No se evaluará nada de la entrega anterior (responsive, accesibilidad, SEO), sino únicamente los aspectos relacionados con la **programación en JS** y la **manipulación del DOM y eventos**.

## Qué hay que entregar (punto a punto)

### 1. Gestión de datos con JavaScript:

- Definir y usar variables (`let`, `const`) de forma clara.
- Crear al menos una función con parámetros y valor de retorno.
- Mostrar el uso de arrays y objetos para almacenar información (ej. lista de productos).

### 2. Manipulación del DOM (mínimo 3 evidencias):

- Crear un nodo dinámicamente con `document.createElement` y añadirlo con `append()`.
- Modificar contenido existente con `textContent`.
- Insertar o eliminar elementos en una lista o tabla.
- Cambiar atributos (ej. `src`, `alt`, `value`) de un elemento.

### 3. Eventos:

- Registrar eventos con `addEventListener`.
- Incluir al menos 2 tipos de eventos distintos (ej. `click` y `change`).
- Mostrar uso de `event.target` para identificar el elemento que dispara el evento.
- Usar `classList.toggle()` para mostrar/ocultar elementos.

#### 4. Buenas prácticas modernas:

- Evitar manejadores en línea (`onclick="..."`) o el uso de `document.write`.
- Código estructurado, comentado y legible.
- Variables y funciones con nombres claros.

#### 5. README (máx. 2 páginas):

- Explicar las evidencias de DOM/JS implementadas.
- Describir brevemente qué aprendiste sobre eventos y manipulación del DOM.

## Requisitos técnicos

- Uso de `let` y `const` para declarar variables.
- Al menos una función definida con sintaxis moderna (función flecha o clásica).
- Arrays y/u objetos para gestionar datos.
- DOM manipulado exclusivamente con `querySelector`, `addEventListener`, `createElement`, `append`, `classList`.
- Prohibido: `document.write`, manejadores en línea y métodos obsoletos.

## Evaluación (cómo se evaluará)

Se valorará exclusivamente lo entregado en esta práctica:

- Variables, funciones y estructuras de datos (20 %).
- Manipulación del DOM (30 %).
- Eventos y su correcta gestión (25 %).

- Buenas prácticas de programación (15 %).
- Claridad en README y evidencias (10 %).

## Rúbrica

Criterio	Excelente (10)	Adecuado (7–9)	Insuficiente (0–6)
Variables y funciones	Uso correcto de <code>let/const</code> , funciones bien diseñadas, arrays/objetos con sentido.	Variables y funciones presentes pero con errores menores.	Variables globales mal gestionadas, funciones inexistentes o incorrectas.
DOM	Crea, modifica y elimina nodos dinámicamente de forma correcta.	Muestra cambios básicos pero limitados o repetitivos.	No manipula el DOM o lo hace con métodos obsoletos.
Eventos	Uso variado de eventos, con <code>event.target</code> y delegación o toggle.	Eventos presentes pero poco variados o con errores puntuales.	Eventos mal gestionados o solo con <code>onclick</code> .
Buenas prácticas	Código claro, comentado, sin inline handlers ni <code>document.write</code> .	Código funcional pero con inconsistencias o poca claridad.	Código desordenado, sin comentarios o con malas prácticas.
README	Explica con claridad lo hecho y lo aprendido.	Explicación incompleta o superficial.	No entrega README o no explica evidencias.