

EJERCICIOS DE CLASE — CLASE 2

DOM AVANZADO: CONTROL, ESTADO Y ARQUITECTURA

EJERCICIO 1 — SEPARAR DOM Y ESTADO

ENUNCIADO

Dado este HTML:

```
<div id="counter">0</div>
<button id="inc">+</button>
<button id="dec">-</button>
```

TAREAS:

1. Crea un objeto `state` con la propiedad `value`.
2. Los botones **modifican el estado**, no el DOM directamente.
3. Crea una función `render()` que actualice el DOM desde el estado.
4. Cada click debe:
 - cambiar el estado
 - llamar a `render()`
 -

EJERCICIO 2 — DELEGACIÓN DE EVENTOS (MENÚ)

OBJETIVO

Un solo listener para muchos elementos.

HTML

```
<ul id="menu">
  <li data-section="partida">Partida</li>
  <li data-section="ranking">Ranking</li>
  <li data-section="config">Configuración</li>
</ul>
```

```
<p id="output"></p>
```

TAREAS

1. Añade **un único** `addEventListener` al ``.
2. Detecta qué `` fue pulsado con `event.target`.
3. Muestra en `#output`:
4. Sección seleccionada: `ranking`

EJERCICIO 3 — FLUJO EVENTO → LÓGICA → ESTADO → RENDER

OBJETIVO

Interiorizar el **flujo profesional** de una app.

HTML

```
<button id="addMove">Añadir jugada</button>
<p id="info"></p>
```

TAREAS

1. Usa un estado:
2. `const gameState = {`
3. `moves: 0`
4. `};`
5. Al pulsar el botón:
 - o aumenta `moves`
 - o renderiza el texto:
 - o `Jugadas realizadas: X`
6. El botón **no toca el DOM directamente**, solo el estado.

EJERCICIO 4 — MINI TABLERO CON ESTADO CENTRALIZADO

OBJETIVO

Simular ajedrez sin caos.

HTML

```
<div id="board"></div>
```

TAREAS

1. Crea un estado:

```
const gameState = {
  selected: null
};
```
2. Genera 8 casillas (`div.square`) dinámicamente.
3. Usa **delegación de eventos** sobre `#board`.
4. Al hacer click en una casilla:
 - o guarda su índice en `state.selected`
 - o vuelve a renderizar
5. La casilla seleccionada debe resaltarse con una clase CSS.

EJERCICIO 5 — POR QUÉ EL DOM IMPERATIVO SE VUELVE UN PROBLEMA

OBJETIVO

Preparar mentalmente React.

ENUNCIADO

Partiendo del ejercicio 4:

1. Añade un contador de clicks por casilla.
2. Guarda los clicks en un objeto del estado:
3. `clicks: {}`
4. En cada render:
 - se debe mostrar el número de clicks en cada casilla
5. Reflexiona:
 - ¿cuántas cosas dependen ahora del estado?
 - ¿qué pasa si modificamos el DOM “a mano”?