

EJERCICIOS DE PRÁCTICA —

REACT + ASINCRONÍA (FETCH / ASYNC / AWAIT)

EJERCICIO 1 — CARGA BÁSICA DESDE API EXTERNA

ENUNCIADO

Crea un componente React (sin JSX) que:

1. Muestre un botón **“Cargar aperturas”**
2. Al pulsarlo, haga una petición `fetch` a:
 - o `https://explorer.lichess.ovh/masters`
3. Muestre en una lista:
 - o El nombre SAN del movimiento
4. Limita el resultado a **5 movimientos**
5. Si ocurre un error, muestra un mensaje en rojo

No se permite manipulación directa del DOM.

EJERCICIO 2 — INDICADOR DE CARGA (LOADING)

OBJETIVO

Comprender **estado adicional** y asincronía real.

ENUNCIADO

Partiendo del ejercicio anterior:

1. Añade un estado `loading`
2. Cuando se inicie el `fetch`, muestra:
 3. Cargando datos...
4. Cuando termine:
 - o Oculta el mensaje
 - o Muestra los resultados
5. Si hay error, **loading desaparece igualmente**

Pista: el estado puede ser `true` / `false`.

EJERCICIO 3 — FILTRO DINÁMICO DE RESULTADOS

OBJETIVO

Separar **datos cargados** de **datos mostrados**.

ENUNCIADO

Amplía el ejercicio anterior:

1. Añade un `<input>` de texto
2. El usuario escribe una letra (ej: e)
3. La lista solo muestra movimientos que:
 - Empiecen por esa letra
4. El filtrado debe hacerse:
 - **sin volver a llamar a fetch**

No se permiten peticiones adicionales.

EJERCICIO 4 — HISTORIAL DE ACCIONES (AJEDREZ)

OBJETIVO

Relacionar asincronía con **registro de acciones**.

ENUNCIADO

Crea un componente que:

1. Tenga un botón:
2. "Simular movimiento"
3. Al pulsarlo:
 - Simula una espera de 1 segundo (Promise + `setTimeout`)
 - Añade un movimiento ficticio al estado:
 - "Peón e2 → e4"
4. Muestra un historial de movimientos en pantalla
5. Cada movimiento debe aparecer **en orden**

No se permite usar `setTimeout` directamente sin Promise.

EJERCICIO 5 — MINI BUSCADOR AJAX (PRE-SPRINT 3)

ENUNCIADO

Crea un buscador en React (sin JSX) que:

1. Muestre un input:
2. "Buscar movimiento"
3. Al escribir:
 - Haga fetch a la API de Lichess
4. Muestre solo los movimientos que:
 - Contengan el texto buscado
5. Si el input está vacío:
 - No se muestra nada
6. Gestiona errores correctamente

El fetch debe hacerse solo cuando haya texto.