

## Ejercicio 1: Sistema de pestañas (tabs)

Crea un pequeño sistema de pestañas (tabs) similar al de una página web moderna.

### Requisitos obligatorios:

1. Debe haber varios botones que representen las pestañas (por ejemplo: "Tab 1", "Tab 2", "Tab 3").
2. Cada botón debe estar asociado a un panel de contenido distinto.
3. Cuando el usuario haga clic en un botón:
  - o Ese botón debe marcarse como "activo" (por ejemplo, con una clase).
  - o Los demás botones deben perder la clase "activo".
  - o Se debe mostrar únicamente el contenido asociado a ese botón.
  - o Los paneles no seleccionados deben ocultarse.
4. La relación entre botón y panel debe hacerse mediante un atributo data-\* en los botones (por ejemplo, data-target="t1").
5. No debes usar display: none en el HTML; el comportamiento debe controlarse con JavaScript añadiendo o quitando clases.

## Ejercicio 2: Reordenar elementos mediante Drag & drop

Implementa un sistema sencillo para **arrastrar y soltar elementos dentro de una lista**, de forma que el usuario pueda cambiar su orden.

### Requisitos obligatorios:

1. Debe haber una lista con varios elementos (por ejemplo, A, B, C...).
2. Cada elemento debe poder arrastrarse usando el ratón.
3. Debe ser posible soltar un elemento encima de otro para **intercambiar posiciones** en la lista.
4. Debes usar los eventos del API de Drag & Drop de HTML5, como mínimo:
  - o **dragstart**
  - o **dragover**
  - o **drop**

5. El navegador, por defecto, no permite soltar elementos, así que **debes habilitarlo** mediante **preventDefault()** en el evento adecuado.
6. No es necesario un comportamiento complejo: basta con permitir reordenar arrastrando un elemento y soltándolo sobre otro.

## Ejercicio 3: Contador controlado con el teclado

Crea un contador en pantalla cuyo valor se pueda modificar usando únicamente el teclado.

### Requisitos obligatorios:

1. Debe mostrarse un número en pantalla que represente el valor del contador.
2. El usuario podrá modificar el contador mediante estas teclas:
  - **Flecha arriba** → **incrementar** el contador en 1.
  - **Flecha abajo** → **disminuir** el contador en 1.
  - **Tecla R** → **reiniciar** el contador a 0.
3. Cada vez que el valor cambie, debe actualizarse automáticamente en pantalla.
4. Debes usar eventos del teclado (por ejemplo, keydown) para detectar las pulsaciones.
5. El contador debe funcionar, aunque el usuario no haga clic en ninguna parte (evento global).

## Ejercicio 4: Temporizador con barra de progreso

Crea un temporizador visual que muestre una barra que se va llenando progresivamente durante un tiempo determinado.

### Requisitos obligatorios:

1. Debe existir un elemento que represente la barra de progreso.
2. La barra debe comenzar vacía (0% de anchura) y llenarse gradualmente hasta llegar al 100%.
3. La duración total debe ser de **10 segundos**.
4. Debes usar setInterval o un mecanismo equivalente para actualizar la barra cada cierto tiempo.

5. Cuando la barra llegue al 100%:
  - El temporizador debe detenerse.
  - La barra debe cambiar de color (por ejemplo, a verde) para indicar que ha terminado.
6. Toda la animación de la barra debe controlarse desde JavaScript.

## Ejercicio 5: Generador de cajas

Crea un programa que permita generar y eliminar elementos en la página mediante eventos de teclado y ratón.

### Requisitos obligatorios:

1. La página debe tener un contenedor (por ejemplo, un div) donde aparecerán las cajas generadas.
2. El usuario podrá crear una caja nueva pulsando la tecla **C**.
3. Cada caja generada debe:
  - Ser un elemento div con una clase común (por ejemplo, caja).
  - Mostrar un número identificador según el orden en el que se crean (Caja 1, Caja 2, Caja 3...).
4. Las cajas se deben añadir dinámicamente al DOM, no estar creadas desde el HTML.
5. Cada caja debe reaccionar al ratón de la siguiente manera:
  - **mouseenter** → la caja cambia temporalmente de color.
  - **mouseleave** → vuelve a su color original.
  - **click** → la caja se elimina del DOM.
6. No debe permitirse crear más de **10 cajas**. Si el usuario intenta crear más, debe mostrarse un mensaje de aviso en pantalla.
7. Todo el comportamiento debe controlarse desde JavaScript.

## Ejercicio 6: Generador de etiquetas

Crea un sistema donde el usuario puede escribir texto para añadir etiquetas dinámicamente a la página y seleccionarlas usando el teclado.

### **Requisitos obligatorios:**

1. Debe haber un campo de texto donde el usuario pueda escribir.
2. Cuando el usuario escriba una palabra y pulse **Enter**, se debe crear una nueva etiqueta dentro de un contenedor.
3. Cada etiqueta debe ser un **div** con una clase común, y mostrar:
  - El texto escrito
  - Un número de etiqueta (Etiqueta 1, Etiqueta 2, Etiqueta 3...)
4. No se pueden crear etiquetas vacías.
5. No se pueden crear más de **15 etiquetas**.  
En caso de superar el límite, se mostrará un aviso.
6. El usuario podrá navegar entre las etiquetas usando el teclado:
  - **ArrowLeft** → seleccionar la etiqueta anterior
  - **ArrowRight** → seleccionar la etiqueta siguiente
7. La etiqueta seleccionada debe:
  - Tener una clase visual específica (por ejemplo, seleccionada)
  - Hacer que todas las demás pierdan la clase seleccionada
8. Con la tecla **Supr**, la etiqueta seleccionada debe eliminarse del DOM.
9. Si la etiqueta seleccionada se elimina:
  - La selección debe pasar a la siguiente etiqueta disponible
  - Si no hay siguiente, a la anterior