

## DOCUMENTACIÓN TÉCNICA: CAZA DEL TESORO

**1. Arquitectura de la Aplicación** La aplicación funciona como una página única (SPA). Esto significa que no recargamos la web cada vez que cambiamos de pantalla (del inicio al juego), sino que usamos JavaScript para borrar y crear elementos en la pantalla según lo que necesite el usuario. Todo el control del juego (nombres, posiciones, puntos) se guarda en un objeto central llamado estado, lo que ayuda a tener los datos ordenados en un solo sitio.

## 2. Implementación JavaScript

- **Manipulación del DOM (Crear elementos):** Para dibujar la pantalla y el tablero, he evitado escribir bloques grandes de texto HTML. En su lugar, uso funciones de JavaScript como `document.createElement` y `appendChild`.
  - **¿Por qué?** Porque es una forma más ordenada y segura de programar. Evita errores si, por ejemplo, el nombre del usuario contiene caracteres extraños que podrían romper la página si se insertaran directamente como texto HTML.
- **Lógica de Movimiento y Algoritmo:** Para saber dónde puede moverse el pirata, uso la función `marcarPosiblesMovimientos`.
  - **Cómo funciona:** Tengo una lista con las 4 direcciones (arriba, abajo, izquierda, derecha). El código recorre esa lista y va sumando casillas según el número que salió en el dado.
  - **Comprobación de bordes:** Lo más importante aquí es que, antes de iluminar una casilla, el código comprueba si las coordenadas están dentro del tablero (entre 0 y 9). Si el cálculo da una casilla fuera del mapa (por ejemplo, la fila -1), el código la ignora para que el juego no falle.
- **Manejo de Eventos (Clics):** Uso `addEventListener` para detectar cuándo el usuario hace clic. Un detalle importante es que, al crear el tablero, le asigno un evento de clic a cada casilla automáticamente. Así, cuando el usuario pulsa una celda, el juego sabe exactamente sus coordenadas y verifica si es un movimiento válido antes de mover al pirata.
- **Guardado de Datos (LocalStorage):** Para cumplir con el requisito de que el récord no se borre al cerrar el navegador, utilizo `localStorage`.
  - **Cómo se guardan:** Como `localStorage` solo guarda texto, uso `JSON.stringify` para convertir mi lista de partidas en texto antes de guardarla, y `JSON.parse` para volver a convertirla en datos útiles cuando cargo el juego.
  - **Lógica de Récord:** Al ganar, el juego mira el número de tiradas. Si es menor que el récord que había guardado (o si no había ninguno), lo actualiza.

**3. Validaciones** Para asegurar que el nombre del usuario es correcto, uso una pequeña comprobación antes de empezar. El código revisa dos cosas:

1. Que el nombre tenga al menos 4 letras.
2. Que no contenga números (usando una expresión regular sencilla `/\d/`). Si algo falla, se muestra un mensaje de error en rojo al usuario.

#### **4. Desafíos y Soluciones**

- **Sincronizar el Tablero:** Un reto fue hacer que lo que se ve en pantalla coincida siempre con los datos internos del juego. Lo solucioné haciendo que, cada vez que el pirata se mueve, la función `generarTablero()` borre la tabla vieja y dibuje una nueva con las posiciones actualizadas.
- **Mensaje de Victoria:** Al ganar, el aviso de "¡Has ganado!" salía tan rápido que no se veía al pirata llegar al cofre. Para arreglarlo, usé la función `setTimeout`. Esto hace que el aviso espere un instante (100 milisegundos), dando tiempo al navegador a dibujar al pirata sobre el cofre antes de mostrar la alerta.