

T2Actividad8: Barajar un array

Ejercicio: Barajar un array introducido por el usuario

Enunciado

El objetivo del ejercicio es:

1. Solicitar al usuario que introduzca números mediante `prompt` y almacenarlos en un array.
2. Detener la solicitud cuando el usuario introduzca el número `0`.
3. Implementar una función `shuffle(array)` que:
 - Reorganice los elementos del array de forma aleatoria.
 - Garantice que todos los reordenamientos posibles tienen la misma probabilidad.
4. Mostrar el array original y el array barajado mediante `alert`.

Análisis del Problema

1. **Entrada de datos:**
 - Solicitar números al usuario mediante un bucle.
 - Detener el bucle cuando se ingrese `0`.
 - Validar que las entradas sean números.
2. **Barajar el array:**
 - Usar un algoritmo como el de **Fisher-Yates** para reordenar aleatoriamente los elementos del array.
 - Intercambiar elementos dentro del array con índices seleccionados aleatoriamente.
3. **Mostrar resultados:**
 - Mostrar el array original.
 - Mostrar el array barajado.

Solución Propuesta

Se implementó una solución en JavaScript que utiliza:

- Una función `shuffle` para reordenar aleatoriamente un array.
- Una función principal para interactuar con el usuario y ejecutar el flujo del programa.
- Interacción mediante `prompt` y `alert`.

Código Fuente

JavaScript

```
javascript
Copiar código
// Función para reordenar aleatoriamente un array
function shuffle(array) {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [array[i], array[j]] = [array[j], array[i]]; // Intercambia los elementos
  }
  return array;
}
```

```
// Función principal para ejecutar el programa
function ejecutarBarajarArray() {
    let numeros = [];
    let entrada;

    // Solicitar números hasta que el usuario introduzca 0
    while (true) {
        entrada = prompt("Introduce un número (0 para terminar):");

        if (entrada === null || isNaN(entrada) || entrada.trim() === "") {
            alert("Error: Introduce un número válido.");
            continue;
        }

        const numero = parseInt(entrada, 10);
        if (numero === 0) break;

        numeros.push(numero);
    }

    if (numeros.length === 0) {
        alert("No se ingresaron números.");
        return;
    }

    // Mostrar el array original
    alert(`Array original: [${numeros.join(", ")}]`);

    // Barajar el array
    const arrayBarajado = shuffle([...numeros]);

    // Mostrar el array barajado
    alert(`Array barajado: [${arrayBarajado.join(", ")}]`);
}

// Ejecutar la función
ejecutarBarajarArray();
```

Pruebas Realizadas

1. Entrada de números válidos:

- Entrada: 5, 10, 15, 20, 0 .
- Salida esperada:
 - Array original: [5, 10, 15, 20]
 - Array barajado: [15, 5, 20, 10] (u otra combinación aleatoria).

2. Entrada con valores no válidos:

- Entrada: abc, , 10, 0 .
- Salida esperada:
 - Error: Introduce un número válido.

3. Sin entrada válida:

- Entrada: 0 .
- Salida esperada: No se ingresaron números.

4. Repetir barajado:

- Ejecutar varias veces con el mismo array (1, 2, 3, 4, 0).
 - Salida esperada: Diferentes combinaciones aleatorias en cada ejecución.
-

Conclusión

El programa permite:

- Solicitar números y almacenarlos en un array.
- Detener la entrada al ingresar 0 .
- Mostrar el array original y barajarlo correctamente con la misma probabilidad para todos los reordenamientos posibles.
- Manejar errores de entrada no válida.