

# EJERCICIOS UNIDAD 4

## EJERCICIO 1

Vamos a gestionar una lista de países haciendo uso de Arrays.

Cuando el usuario cargue la página, se cargarán las opciones:

- Mostrar el número de países.
- Mostrar listado de países (y le preguntará si quiere mostrarlos en el orden que se encuentran en el array, del revés u ordenados alfabéticamente).
- Añadir un país (y le preguntará si quiere añadir al principio o al final).
- Borrar un país (y le preguntará si quiere borrar al principio o al final).
- Consultar un país (y le preguntará si quiere consultar por posición o por nombre).

Para ello necesitarás crear un archivo arrays.js que exporte las siguientes funciones que recibirán un array por parametro:

- Mostrar el número de elementos del array.
- Mostrar todos los elementos del array.
- Muestra los elementos del array en sentido inverso.
- Muestra los elementos del array ordenados alfabéticamente.
- Añadir un elemento solicitado al usuario al principio del array.
- Añadir un elemento solicitado al usuario al final del array.
- Borrar un elemento al principio del array (y decir cuál se ha borrado).
- Borrar un elemento al final del array (y decir cuál se ha borrado).
- Muestra el elemento que se encuentra en una posición que el usuario indica.
- Muestra la posición en la que se encuentra un elemento que le indica el usuario.

## EJERCICIO 2

Haz un programa que pregunte una letra de la A a la Z. Tras ello te dirá cuantos DNIs de 3 cifras (del 001 al 999) tienen esa letra y los meterá en un array, tras ello te mostrará el listado de todos los DNIs que tienen esa letra.

## EJERCICIO 3

Crea una clase Cliente y define un array de clientes con los siguientes datos:

Nombre;Localidad;Cuota Laura;Santander;50 Álvaro;Castro;50 Igor;Castro;60 Ivan;Santander;40 Mónica;Zamora;30 Javi;Bilbao;30 David;Bilbao;50
--

A partir del mismo, el usuario podrá elegir del menú:

1. Todos los clientes: se mostrará una tabla con los valores que están en la variable anterior.
2. Usuarios de una localidad: y a partir de la localidad introducida por el usuario se mostrarán en una tabla los nombres y cuotas de las personas que viven en esa provincia.
3. Usuarios que tengan una cuota mayor que un valor: y se mostrarán en una tabla los nombres de usuario, provincias y cuotas de aquellos que tienen una cuota superior al valor introducido por el usuario (valora cuál es el mejor modo de hacerlo).

## EJERCICIO 4

Necesitamos almacenar en un programa todos los discos de música que tenemos en casa.

Crea una clase "Disco" que almacene la siguiente información:

- Nombre del disco.
- Grupo de música o cantante.
- Año de publicación.
- Tipo de música (podrá ser "rock", "pop", "punk" o "indie");
- Localización: almacenará un número de estantería.
- Prestado: almacenará un valor booleano. Por defecto será false.
- Carátula: nombre del archivo de imagen de la carátula (p.ej: imagen.png)

Además tendrá los siguientes métodos:

- Un constructor (la localización será 0 por defecto y prestado estará a false, carátula por defecto imagen.png).
- Un método que permitirá cambiar el número de estantería en la localización.
- Un método que permitirá cambiar la propiedad Prestado.
- Un método que muestre toda la información de un disco.

Guarda todo el código en un archivo llamado disco.js y reutiliza en tu página el archivo de arrays que hicimos en la práctica 1.

Crea un array vacío para almacenar los discos.

Cuando el usuario cargue la página, se mostrarán las opciones:

- Mostrar número de discos.
- Mostrar tabla de discos con todos los datos de cada disco(y le preguntará si quiere mostrarlos en el orden que se encuentran en el array, del revés u ordenados alfabéticamente).
- Mostrar un intervalo de discos por año de publicación(y le pedirá que introduzca el intervalo en formato inicio-fin)
- Añadir un disco (y le preguntará si quiere añadir al principio o al final).
- Borrar un disco (y le preguntará si quiere borrar al principio o al final).
- Consultar un disco (y le preguntará si quiere consultar por posición o por nombre).

## EJERCICIO 5

Estamos desarrollando un juego y necesitamos crear un objeto jugador, que tenga una propiedad fuerza que inicialmente vale 1, tendremos un método incrementarFuerza que aumentará la fuerza del jugador en 1 y un método consultar fuerza que mostrará un mensaje: "Tu fuerza es x" donde x es la fuerza actual del jugador.

Mostrar 2 botones que sirvan para incrementar la fuerza y para mostrar la fuerza.

## EJERCICIO 6

Crea un programa que reciba como entrada un array de cadenas con la convocatoria de jugadores para un partido:

Por ejemplo:

```
const convocatorias =  
["Marcos,Luis,Pepe,Manuel,Lolo","Marcos,Luis,Pepe,David,Antonio,Pedro"]
```

El array puede contener 2 o más cadenas, con diferente número de nombres.

El programa debe de obtener la lista de jugadores que han sido convocados a todos los partidos.

En el caso del ejemplo saldría: Marcos, Luis y Pepe