Unidad 3

```
/*
Si se sabe cuántos elementos tiene un array se le puede indicar directame
nte que guarde un nuevo
 elemento al final
 */

let elementos = [1, "Ale", 7]
elementos[3] = 29

/*
 Si se desconoce el número de elementos se puede utilizar push, que añad
e directamente un
 elemento al final (y además devuelve el número de elementos del array tra
s la inserción)
*/

let elementos = [1, "Ale", 7]
elementos.push(29)

//Se puede añadir un elemento al principio con unshift

let elementos = [1, "Ale", 7]
elementos.unshift(29)

/*
NOTA:

tanto unshift como push permiten añadir en una misma instrucción varios v
alores
*/
```

```
//Eliminar elementos de un array

//shift - elimina el primer elemento

let elementos = [1, "Ale", 7]
elementos.shift()

//pop - elimina el último elemento


let elementos = [1, "Ale", 7]
elementos.pop()



//Modificar la propiedad de length

//Se eliminan todos los elementos que queden fuera de la nueva longitud

let elementos = [1, "Ale", 7]
elementos.length = 1

//Eliminar la cantidad de elementos indicada desde una posición determina
da con splice

let elementos = [1, "Ale", 7]
elementos.splice = (1, 2)

//Concat permite extender un array añadiéndole al final el contenido de otr
o array

let elementos = [1, "Ale", 7]
let elementos2 = [2, 3, 4, 5]
let extendidos = elementos.concat(elementos2)
```

```
//Los arrays se pueden copiar completos o traer una parte de ellos usando slice

let elementos = [1, "Ale", 7, 22, 45, 78, "Hola"]
let elementos3 = elementos.slice() //Copia el array "elementos"
let parcial = elementos.slice(2, 4) //No se incluye el valor de la casilla núme ro 4 en este caso

//indexOf() Sirve para buscar la posición de un elemento dentro de un arra y o un string


let frutas = ["manzana", "pera", "plátano"];

console.log(frutas.indexOf("pera")); //1
console.log(frutas.indexOf("kiwi")); //-1

//lastIndexOf() Realiza las mismas tareas que indexOf pero trabajando desd e el extremo derecho del array

frutas = ["manzana", "pera", "plátano"];

console.log(frutas.lastIndexOf("pera")); //1
console.log(frutas.lastIndexOf("plátano")); //2

//includes() Sirve para comprobar si un elemento existe dentro de un array o un string

frutas = ["manzana", "pera", "plátano"];

console.log(frutas.includes("pera")); // true
console.log(frutas.includes("kiwi")); // false

// join() El método join une todos los elementos del array en una única cade na

frutas = ["manzana", "pera", "plátano"];
let resultado = frutas.join("-") // manzana-pera-plátano
```

```
console.log(resultado)

//sort() Permite realizar cualquier ordenación que necesitemos

let numeros = [10, 6, 4, 3, 15, 1];
numeros.sort(); //1,3,4,5,6,10,15
console.log(numeros)

//reverse() invierte el orden de un array

numeros = [10, 6, 4, 3, 15, 1];
numeros.sort(); //15,10,6,4,3,1
console.log(numeros)

//conjuntos new Set() no permiten elementos repetidos

let conjunto = new Set([1, 2, 3, 4, 5, 5, 5, 5]);
console.log(conjunto); // {1,2,3,4,5}

//Recorrido de un conjunto

for (let valor of conjunto) {
   console.log(valor);
}

//Para añadir un elemento a un conjunto se utiliza add

conjunto = new Set([1, 2, 3, 4, 5]);
conjunto.add(6);

//Para eliminar un elemento se utiliza delete

conjunto = new Set([1, 2, 3, 4, 5]);
conjunto.delete(3);

//Para eliminar todos los elementos se utiliza clear

conjunto = new Set([1, 2, 3, 4, 5]);
```

```
conjunto.clear();

//Tamaño de un conjunto

conjunto = new Set([1, 2, 3, 4, 5]);
console.log(conjunto.size); //5

//Con el métdodo has se puede comprobar si un elemento está en el conjunto

conjunto = new Set([1, 2, 3, 4, 5]);
console.log(conjunto.has(3)); //true
console.log(conjunto.has(6)); //false

//Se puede convertir un conjunto en un array con propagación

conjunto = new Set([1, 2, 3, 4, 5]);
let array = [...conjunto];
console.log(array); //[1,2,3,4,5]

//Mapas new Map() son estructuras de datos que permiten almacenar pares clave-valor

let mapa = new Map([
    ["clave1", "valor1"],
    ["clave2", "valor2"],
    ["clave3", "valor3"]
]);

console.log(mapa); //Map(3) {"clave1" => "valor1", "clave2" => "valor2", "clave3" => "valor3"}

//Recorrido de un mapa

const telefonos = new Map([
    ["Pepe", "123456789"],
    ["Ana", "987654321"],
    ["Luis", "456789123"]
```

```
]);

for (persona of telefonos){
    console.log(persona);
}

for (let [clave, valor] of telefonos) {
    console.log(`${clave} : ${valor}`);
}

//Si solo estamos interesados en las claves o en los valores

for (let telefono of telefonos.keys()) {
    console.log(telefonos);
}
for (let persona of telefonos.values()) {
    console.log(persona);
}

//Añadir un nuevo par clave-valor con set

telefonos.set("Marta", "321654987");
telefonos.set("Pepe", "111222333"); //Si la clave ya existe se actualiza el val
or
console.log(telefonos);

//Eliminar un par clave-valor con delete

telefonos.delete("Ana");
console.log(telefonos);

//Busqueda de elementos en un mapa con has

if (telefonos.has("Luis")) {
    console.log("¡Luis está en la agenda!"); //true
} else {
    console.log("Luis no está en la agenda"); //false
}
```

```
//Lectura de valores de un mapa con get

console.log(telefonos.get("Pepe")); //111222333

//Con spread se puede convertir un mapa en un array

let convertirArray = new Map([
    ["clave1", "valor1"],
    ["clave2", "valor2"],
    ["clave3", "valor3"]
]);

let arrayConvertido = [...convertirArray];
console.log(array); //[["clave1", "valor1"], ["clave2", "valor2"], ["clave3", "valor3"]]
```