



# Certified Tech Developer

The Ultimate Degree



## Programación Imperativa

### Ejercicios de repaso Bucles

#### Objetivo

Vamos a repasar las distintas sintaxis y posibilidades de hacer bucles en JS.



#### Micro desafíos - Paso I

Realizar estas acciones simples y de uso muy común para trabajar con bucles y arrays.

Recorrer y mostrar:

```
const puntajes = [12, 34, 56, 34, 5, 8, 76, 4, 1, 90];

for (let index = 0; index < puntajes.length; index++) {
  console.log(puntajes[index]);
}
```

Recorrer, Cambiar y Mostrar:

```
const preciosSinIva = [1200, 340, 560, 30400, 500, 80, 1000, 90000];
const iva = 1.21;
```

```
for (let i = 0; i < preciosSinIva.length; i++) {  
  console.log(preciosSinIva[i] * iva);  
}
```

## Recorrer, Cambiar, Guardar y Mostrar:

Ya es hora de usar funciones y separar dominios, guardar es una cosa y mostrar es otra, no debe hacerse todo junto. Una posibilidad podría ser:

```
const preciosSinIva = [1200, 340, 560, 30400, 500, 7060, 400, 1000];  
const iva = 1.21;  
let preciosConIva = [];  
  
function guardarPreciosConIva(preciosSinIva) {  
  for (let i = 0; i < preciosSinIva.length; i++) {  
    preciosConIva.push(preciosSinIva[i] * iva);  
  }  
}  
  
function mostrarPreciosConIva(array) {  
  for (let i = 0; i < array.length; i++) {  
    console.log(array[i]);  
  }  
}  
  
guardarPreciosConIva(preciosSinIva);  
mostrarPreciosConIva(preciosConIva);
```

Sin embargo, mejor es que la función no esté modificando una variable externa, daría lugar a efectos secundarios sorpresa (BUG) si esa variable también fuera tocada en algún otro momento.

Recuerden que la función debe ser lo más pura posible, recibe entrada, devuelve salida.

```
const preciosSinIva = [1200, 340, 560, 30400, 500, 7060, 400, 1000];  
const iva = 1.21;  
  
function guardarPreciosConIva(preciosSinIva) {  
  let preciosConIva = [];  
  for (let i = 0; i < preciosSinIva.length; i++) {  
    preciosConIva.push(preciosSinIva[i] * iva);  
  }  
  return preciosConIva;  
}  
  
function mostrarPreciosConIva(array) {  
  for (let i = 0; i < array.length; i++) {
```

```

        console.log(array[i]);
    }
}

const preciosConIva = guardarPreciosConIva(preciosSinIva);
mostrarPreciosConIva(preciosConIva);

```

## Recorrer, Seleccionar, Guardar y Mostrar:

```

const preciosSinIva = [1200, 340, 560, 30400, 500, 80, 7060, 400, 1000,
90000];
const iva = 1.21;
const límiteParaDescuento = 4000;
const descuento = 0.75;

function guardarPreciosConIva(preciosSinIva) {
    let preciosConIva = [];
    for (let i = 0; i < preciosSinIva.length; i++) {
        preciosConIva.push(preciosSinIva[i] * iva);
    }
    return preciosConIva;
}

function aplicarDescuento(array) { //en la definición de la función
puedo llamar como quiera a mis parámetros, de hecho si puedo poner
nombres muy genéricos como en este caso, puedo pasar cualquier array y
se puede reutilizar este fragmento para otros propósitos.
    let res = [];
    for (let i = 0; i < array.length; i++) {
        const element = array[i];
        if (element > límiteParaDescuento) {
            res.push(element * descuento);
        }
    }
    return res;
}

function mostrarPrecios(array) {
    for (let i = 0; i < array.length; i++) {
        console.log(array[i]);
    }
}

```

```
}  
  
const preciosConIva = guardarPreciosConIva(preciosSinIva);  
const preciosConDescuento = aplicarDescuento(preciosConIva);  
mostrarPrecios(preciosConIva);  
console.log("-----");  
mostrarPrecios(preciosConDescuento);
```