



# TI: Ejercicios JS

Javier Ribal del Río

2025-12-13

## Table of contents

<b>1 Ejercicio 1</b>	<b>1</b>
<b>2 Ejercicio 2</b>	<b>2</b>
2.1 Salida esperada (ejemplo): . . . . .	2
<b>3 Ejercicio 3</b>	<b>2</b>
<b>4 Ejercicio 4</b>	<b>2</b>
<b>5 Ejercicio 5</b>	<b>3</b>
<b>6 Ejercicio 6</b>	<b>3</b>
<b>7 Ejercicio 7</b>	<b>3</b>
<b>8 Ejercicio 8</b>	<b>3</b>

El objetivo de estos ejercicios es repasar aquellas funcionalidades de JavaScript que difieren del resto de lenguajes de programación empírica, por lo que los siguientes ejercicios de **DEBEN DE RESOLVER SIN UTILIZAR NINGUNA CONDICIONAL NI NINGÚN BUCLE**.

Principalmente trabajarás con:

- La función `filter`
- La función `map`
- Las funciones `arrow`

Debe de realizar la tarea en un archivo js, para comprobar el resultado debe de ejecutar desde la terminal con `node nombreArchivo.js`.

## 1 Ejercicio 1

Defina un array llamado `arrOriginal` con el siguiente contenido numérico. Este array será el origen de datos para los ejercicios posteriores, salvo que se indique lo contrario.



3, 4, 2, 4, 4, 5, 12, 24, 3, 23, 232, 2, -3, -9, 2, -23, 200, -23, 12, 43, 4, -32, -32, 0, 23, 23

---

## 2 Ejercicio 2

Partiendo del array `arrOriginal`, imprima por pantalla la siguiente frase:

"En el subsistema de software somos x miembros"

donde `x` debe tomar, sucesivamente, **cada uno de los valores de `arrOriginal`**.

Imprima por pantalla el resultado.

### 2.1 Salida esperada (ejemplo):

En el subsistema de software somos 3 miembros  
En el subsistema de software somos 4 miembros  
En el subsistema de software somos 2 miembros

---

## 3 Ejercicio 3

Partiendo de `arrOriginal`, cree un nuevo array llamado `arrPositivos` que contenga **únicamente números positivos**.

Utilice `arrPositivos` para imprimir por pantalla las frases correspondientes.

---

## 4 Ejercicio 4

Partiendo de `arrPositivos`, cree un nuevo array llamado `arrValidos` que cumpla además el siguiente criterio:

- El número máximo de personas es **10**

Imprima por pantalla las frases correspondientes a `arrValidos`.

---



## 5 Ejercicio 5

Partiendo nuevamente de `arrOriginal`, cree un nuevo array llamado `arrFiltrado` que contenga únicamente aquellos elementos que cumplan **todas** las siguientes condiciones:

- Sean positivos
  - Sean pares
  - Sean menores que 100
- 

## 6 Ejercicio 6

Partiendo de `arrFiltrado`:

1. Cree un nuevo array llamado `arrCubos` elevando **al cubo** cada uno de sus elementos
  2. A partir de `arrCubos`, cree un nuevo array llamado `arrCubosFiltrados` que contenga únicamente aquellos valores cuyo resultado **termine en 0**.
- 

## 7 Ejercicio 7

Partiendo de `arrCubosFiltrados`, cree un nuevo array llamado `arrFinal` añadiendo el valor 21.

---

## 8 Ejercicio 8

Partiendo de `arrFinal`, cree un **nuevo array independiente** llamado `arrCopia`, de forma que cualquier modificación posterior **no afecte a arrFinal**.

---