



# Métodos de Manipulación

Centro de Servicios y Gestión Empresarial  
SENA Regional Antioquia



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)

# Contextualización

# Métodos de manipulación



En JavaScript, existen métodos avanzados para trabajar con Arrays y Objects, permitiendo filtrar, transformar y reducir datos de manera eficiente.

The JavaScript logo is displayed on a solid yellow square background. It consists of the letters "J" and "S" in a large, bold, black, sans-serif font, with the "S" being slightly larger and more prominent than the "J".

# Métodos de manipulación



## map()

- Crea un nuevo array aplicando una función a cada elemento.

## filter()

- Filtra elementos que cumplen una condición.

## reduce()

- Reduce los valores de un array a un único resultado.

## find()

- Encuentra el primer elemento que cumple una condición.

## some()

- Verifica si **algún** elemento cumple una condición.

## every()

- Verifica si **todos** los elementos cumplen una condición.

## sort()

- Ordena los elementos del array.

# Métodos de manipulación

# map()

- Transformar Datos en un Nuevo Array.
- Convierte cada elemento en un nuevo array sin modificar el original.

## Ejemplo:

```
// Definición del array
let costos = [3000, 45000, 6000, 75000, 9000];
// Uso de la función map
let impuestos = costos.map(valor => valor * 0.19);
// Impresión del resultado
console.log(impuestos);
// [570, 8550, 1140, 14250, 1710]
```

# filter()

- Filtrar Datos en un Array
- Crea un nuevo array con elementos que cumplen una condición.

## Ejemplo:

```
// Definición del array
let valores = [5, 10, 15, 20];
// Uso de la función filter
let numerospares = valores.filter(num => num % 2 === 0);
// Impresión del resultado
console.log(numerospares);
// [10, 20]
```



# filter()



## Ejemplo:

```
// Definición del objeto
let productos = [
  { nombre: "Laptop", stock: 5 },
  { nombre: "Mouse", stock: 0 },
  { nombre: "Teclado", stock: 8 }
];
// Uso de la función filter
let disponibles = productos.filter(prod => prod.stock > 0);
// Impresión del resultado
console.log(disponibles);
// [{ nombre: "Laptop", stock: 5 },
// { nombre: "Teclado", stock: 8 }]
```



# reduce()

- Reducir un Array a un Solo Valor
- Acumula valores y los reduce a un único resultado.

## Ejemplo:

```
// Definición del array
let numeros = [10, 20, 30, 40];
// Uso de la función reduce
let sumaTotal = numeros.reduce(
  (acumulador, num) => acumulador + num, 0);
// Impresión del resultado
console.log(sumaTotal);
// 100
```

# reduce()

## Ejemplo:

```
// Definición del objeto
let carrito = [
  { producto: "Celular", precio: 300 },
  { producto: "Auriculares", precio: 50 },
  { producto: "Funda", precio: 20 }
];
// Uso de la función reduce
let totalCompra = carrito.reduce(
  (total, item) => total + item.precio, 0);
// Impresión del resultado
console.log(totalCompra);
// 370
```

# find()

- Encontrar un Elemento en un Array
- Devuelve el primer elemento que cumple una condición.

## Ejemplo:

```
// Definición del array
let numeros = [8, 2, 12, 15];
// Uso de la función find
let encontrado = numeros.find(num => num > 10);
// Impresión del resultado
console.log(encontrado);
// 12
```

# find()



## Ejemplo:

```
// Definición del objeto
let productos = [
  { nombre: "Teclado", stock: 5 },
  { nombre: "Mouse", stock: 0 }
];
// Uso de la función find
let teclado = productos.find(prod => prod.nombre === "Teclado");
// Impresión del resultado
console.log(teclado);
// { nombre: "Teclado", stock: 5 }
```

# some() y every()


- Verificar Condiciones en un Array

## Opciones:

Método	Verifica si...	Ejemplo
<b>some()</b>	<b>Al menos un elemento cumple la condición.</b>	<code>array.some(x =&gt; x &gt; 10)</code>
<b>every()</b>	<b>Todos los elementos cumplen la condición.</b>	<code>array.every(x =&gt; x &gt; 10)</code>

# some() y every()

```
// Definición del array
let edades = [15, 18, 22, 25];
// Uso de la función some
let hayMayoresDeEdad = edades.some(edad => edad >=
18);
// Impresión del resultado
console.log(hayMayoresDeEdad);
// true (porque hay 18, 22 y 25)
```

A large, thick orange arrow originates from the left side of the slide and points towards the right, specifically towards the code block for the 'every()' function. It has a curved, ribbon-like appearance.

```
// Definición del array
let edades = [15, 18, 22, 25];
// Uso de la función every
let todosMayoresDeEdad = edades.every(edad => edad >=
18);
// Impresión del resultado
console.log(todosMayoresDeEdad);
// false (porque hay 15)
```

# sort()

- Ordenar un Array
- Ordena elementos de menor a mayor o viceversa.

## Ejemplo:

```
// Definición del array
let numeros = [5, 2, 9, 1];
// Uso de la función sort 1-N
numeros.sort((a, b) => a - b);
console.log(numeros); // [1, 2, 5, 9]
// Uso de la función sort N-1
numeros.sort((a, b) => b - a);
console.log(numeros); // [9, 5, 2, 1]
```



# sort()

- Ordenar un Array
- Ordena elementos de menor a mayor o viceversa.

## Ejemplo:

```
// Definición del array
let nombres = ["Carlos", "Ana", "Pedro"];
// Ordenar alfabéticamente
nombres.sort();
console.log(nombres); // ["Ana", "Carlos", "Pedro"]
// Ordenar alfabéticamente inverso
nombres.sort().reverse();
console.log(nombres); // ["Pedro", "Carlos", "Ana"]
```

# Resumen de Métodos para Manipular Arrays



Método	Descripción	Ejemplo
<code>map()</code>	Transforma cada elemento en un nuevo array.	<code>array.map(x =&gt; x * 2)</code>
<code>filter()</code>	Filtra los elementos que cumplen una condición.	<code>array.filter(x =&gt; x &gt; 10)</code>
<code>reduce()</code>	Reduce los valores del array a un único resultado.	<code>array.reduce((acc, x) =&gt; acc + x, 0)</code>
<code>find()</code>	Encuentra el primer elemento que cumple una condición.	<code>array.find(x =&gt; x &gt; 10)</code>
<code>some()</code>	Verifica si <b>al menos uno</b> cumple la condición.	<code>array.some(x =&gt; x &gt; 10)</code>
<code>every()</code>	Verifica si <b>todos</b> cumplen la condición.	<code>array.every(x =&gt; x &gt; 10)</code>
<code>sort()</code>	Ordena los elementos del array.	<code>array.sort((a, b) =&gt; a - b)</code>



# GRACIAS

Presentó: Alvaro Pérez Niño  
Instructor Técnico

Correo: [aperezn@sena.edu.co](mailto:aperezn@sena.edu.co)

<http://centrodeserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)