

Nota: Crear un repositorio y versionar cada ejercicio mediante un commit con el mensaje que considere mas claro. Crear un repositorio remoto y sincronizar el local con el mismo.

Datos simulados:

```
const productos = [
  { id: 1, nombre: "Remera", precio: 1500, categoria: "Ropa" },
  { id: 2, nombre: "Pantalón", precio: 2500, categoria: "Ropa" },
  { id: 3, nombre: "Zapatillas", precio: 8000, categoria: "Calzado" },
  { id: 4, nombre: "Gorra", precio: 1200, categoria: "Accesorios" },
  { id: 5, nombre: "Campera", precio: 9500, categoria: "Ropa" }
];
```

Parte 1: Acceso a elementos y recorrido de arrays

1. Acceder a propiedades individuales:

Mostrá en la consola el nombre del primer producto de la lista.

2. Recorrido con `for...of`:

Usá un bucle `for...of` para mostrar en consola el nombre y el precio de cada producto.

Formato sugerido: "Producto: Remera - Precio: \$1500"

3. Recorrido con `.forEach()`:

Repetí el ejercicio anterior, pero usando el método `.forEach()` en lugar del bucle tradicional.

Parte 2: Métodos de transformación y filtrado

4. Obtener solo los nombres con `.map()`:

Creá un nuevo array que contenga únicamente los nombres de los productos.

Mostralo en consola.

Resultado esperado: ["Remera", "Pantalón", "Zapatillas", "Gorra", "Campera"]

5. **Filtrar por categoría con `.filter()`:**

Creá un nuevo array que contenga solo los productos cuya categoría sea "Ropa".
Mostrar en consola.

6. **Filtrar por precio con `.filter()`:**

Creá un array que contenga todos los productos cuyo precio sea mayor a \$3000.

7. **Buscar un producto específico con `.find()`:**

Usá el método `.find()` para encontrar el objeto que tenga como nombre "Gorra".
Mostralo completo en consola.

Parte 3: Métodos de validación

8. **Verificar con `.some()`:**

Usá `.some()` para comprobar si **existe algún producto** con un precio mayor a \$10.000.
Mostrá el resultado (`true` o `false`).

9. **Verificar con `.every()`:**

Usá `.every()` para saber si **todos los productos** cuestan más de \$1000.
Mostrá el resultado.

10. **Verificar existencia con `.includes()`:**

Usando el array de nombres creado en el punto 4, comprobá si contiene el nombre "Campera".

Parte 4:

11. **Ordenar con `.sort()`:**

Ordená los productos por precio de menor a mayor.
Mostrá el nuevo array ordenado.

12. **Generar mensajes personalizados con `.map()`:**

Usá `.map()` para crear un array de strings como este:
"El producto Remera cuesta \$1500 y pertenece a la categoría Ropa."
Mostrá el resultado en consola.

13. **Agregar productos con `spread`:**

Crear un nuevo array de objetos con más productos.
Utilizar el operador `spread` para combinar ambos arrays de productos

en un nuevo array.

Mostrar el array resultante en la consola.