

Tarea 013 - Javascript – Operaciones sobre Inventario de Productos

Los resultados de todas las tareas incorporarán, además del código fuente, los comentarios precisos y necesarios para su fácil comprensión. No escatimes esfuerzos en comentar el código, es una buena práctica para aprender, además de ser muy útil para modificaciones o reutilizaciones futuras.

Sería muy buena práctica añadir el título y el enunciado del ejercicio (como comentarios) al principio del código fuente.

Añade documentos en formato Word con capturas de la salida por pantalla (al ejecutar la página) si consideras que queda más clara la resolución del ejercicio.

Deberás entregar dos archivos con el resultado:

1. **Un documento en Word sin comprimir:** que contenga las imágenes que demuestren el resultado.
2. **Los archivos de código fuente:** en un único archivo, ya sea de extensión: **html**, **js**, o **css**. Si precisas entregar varios archivos, comprímelos en un único **zip**. (No admito rar).

El nombre del archivo entregado comenzará por tu nombre seguido por TareaXXX. Ejemplo: **federicoTarea014.zip**

Enunciado:

Tenemos un inventario de productos en una tienda, cada producto tiene un **nombre**, **precio**, y **cantidad** en stock. Se pide realizar varias operaciones sobre el inventario utilizando **programación funcional**.

Parte 1ª Dado el siguiente array de productos:

```
const inventario = [  
  { nombre: "Laptop", precio: 1500, cantidad: 4 },  
  { nombre: "Teléfono", precio: 500, cantidad: 10 },  
  { nombre: "Teclado", precio: 30, cantidad: 20 },  
  { nombre: "Monitor", precio: 200, cantidad: 5 }  
];
```

Escribe las siguientes funciones:

1. **obtenerProductosCaros:** Retorna un array con los nombres de los productos que tienen un precio mayor a 500€.
 - Usando **filter** para obtener los productos con precio mayor a 500 y **map** para obtener solo los nombres.
2. **calcularValorTotalInventario:** Calcula el valor total de todos los productos en inventario, considerando su precio y cantidad. El valor total de un producto es precio * cantidad.
 - Utilizando **map** y **reduce** para sumar los valores de los productos.
 - Utilizando únicamente **reduce**
3. **incrementarPrecios:** Retorna un nuevo array de productos donde todos los precios se han incrementado en un 10%, sin modificar el array original.
 - Usando **map** y **spread** para modificar el precio de cada producto sin mutar el array original.
 - Utilizando **ArrayFrom()**

Parte 2:

Escribe una función que combine estas operaciones:

4. **ObtenerProductosCarosConPreciosIncrementados:** Retorna los nombres de los productos que cuestan más de 500€ **después** de haber incrementado los precios en un 10%.