



## Ejercicio de POO en JavaScript

[Evaluación del ejercicio](#)

En este ejercicio se os pide crear una página web para una tienda que vende diferentes productos. Aquí un ejemplo de la archiconocida “Tienda de Adrián”:

La tienda de Adrián		
Productos Agrícolas	Productos Deportivos	Muebles
<b>Zanahoria</b> Precio: 0.5€/kg Origen: Granja del tío Pepe	<b>Gafas de natación</b> Precio: 15€/kg Marca: Spido	<b>Mesilla de Noche</b> Precio: 50€/kg Fabricante: Muebles La Abuelita
<b>Manzana</b> Precio: 2€/kg Origen: Granja del tío Pepe	<b>Calzado de marcha</b> Precio: 45€/kg Marca: Ribuk	<b>Armario empotrado</b> Precio: 1500€/kg Fabricante: Muebles La Abuelita
<b>Pera</b> Precio: 2.5€/kg Origen: Granja del tío Pepe	<b>Gorra para el sol</b> Precio: 10€/kg Marca: Naik	<b>Escritorio estudiamucho</b> Precio: 500€/kg Fabricante: Muebles La Abuelita
<b>Cebolla</b> Precio: 0.75€/kg Origen: Granja del tío Pepe	<b>Calzado para correr</b> Precio: 100€/kg Marca: Addydas	<b>Estantería ayquesecae</b> Precio: 300€/kg Fabricante: Ebanisto el Manitas
<b>Lechuga</b> Precio: 1€/kg Origen: Isabelen Baserria		<b>Sofá Piesparaqueosquero</b> Precio: 800€/kg Fabricante: Ebanisto el Manitas
<b>Remolacha</b> Precio: 1.2€/kg Origen: Isabelen Baserria		
<b>Pepino</b> Precio: 2.99€/kg Origen: Joxe Mariren Baserria		
<b>Ajo</b> Precio: 0.89€/kg Origen: Joxe Mariren Baserria		
<b>Calabacín</b> Precio: 1.99€/kg Origen: El Bach-erri		
<b>Tomate</b> Precio: 5€/kg		

A nivel de diseño, la tienda cuenta con:

- Un contenedor que se estira por completo, ocupando toda la altura de la página, aun haciendo *zoom out/scroll*.
- Una cabecera con la etiqueta <header>.
- Tres columnas con título que únicamente se estiran lo necesario, en función de la cantidad de productos que muestren.

Una opción para este diseño es usar [Flex](#).

Dicha tienda tiene la siguiente lógica de negocio detrás (todos los siguientes ficheros .js deben ir dentro de una carpeta js/):

- Una **clase Producto (producto.js)** con los atributos nombre y precio. Debe incluir los métodos *get* y *set* para ambos atributos, así como un método constructor.
- **Tres clases que extiendan Producto (en un .js separado cada una, importando Producto)** y que incluyan:
  - Un atributo nombre.
  - Un atributo precio.
  - Un atributo extra como origen, fabricante, etc.
  - Los métodos *get* y *set* para los 3 atributos previos.
  - Un método estático para devolver el nombre del tipo de producto (p.e. para libros un string “Libros”).
- Un fichero **tienda.js** que importe los 3 tipos de productos creados y exporte 2 elementos: (i) una lista con los nombres de cada tipo de producto en la tienda y (ii) una lista que contenga sublistas de productos, cada sublista con productos de una clase. La lista (i) debe crearse usando el método estático de cada clase para obtener los nombres de cada producto. Además, debe estar ordenada siguiendo el mismo orden que la lista (ii), es decir: si en la lista (i) el primer elemento son “Juguetes”, en la lista (ii) la primera sublista debe contener productos de la clase juguete.

También habrá un fichero **main.js** que debe ser importado por nuestro index.html.

Únicamente este fichero es importado por index.html. main.js incluye lo siguiente:

- Importa las dos listas de tienda.js.
- Contiene una única función que recibe la instancia de un producto cualquiera (de los 3 creados) e imprime su información en formato HTML. A la hora de imprimir la información, ésta debe de estar contenida en un <div> que separe claramente un ítem de otro (p.e. que tenga un borde y unos márgenes). El título debe incluirse en un <div> con un estilo de título (a gusto de cada persona) y el resto de datos deben tener, en negrita (mediante <span>), un nombre de campo. Por ejemplo: “Precio: 20€”. Si hubiesen unidades o cualquier otro dato necesario para entender la información, debe incluirse. La función devolverá un template string con este contenido.
- Debemos meter todo el código dentro de

```
window.onload = () => {  
    // [AQUI VUESTRO CODIGO]  
}
```

- Debe rellenar cada columna del diseño con un tipo de producto. Se le asignará a cada columna un título importado de tienda.js (no se puede poner a mano) y se mostrará la lista de productos disponible debajo del título. La información del producto se imprimirá mediante la función previamente comentada, la misma para todos los productos.

Es necesario usar la sintaxis de clases de ECMAScript 6+.

Probad a incluir muchos productos de uno de los tipos para comprobar que el diseño se extiende hasta abajo del todo y permite *scroll*.

## Evaluación

La puntuación será de 0 en caso de que el proyecto no esté dividido en carpetas correctamente (css/, js/). El fichero HTML debe ir en la raíz.

Tarea	Puntuación (sobre 10)
Diseño	3
Lógica de negocio: tienda.js, producto.js y 3 productos	3
main.js	4