



Examen presencial DWECE

Unidad 3

Notas:

- Descarga el documento adjunto *Actividades.zip* y descomprímelo en tu equipo.
- Trabaja sobre los archivos de la carpeta obtenida tras descomprimir.
- Al finalizar el examen, comprime la carpeta y súbela a la tarea del examen.
- **TODO SE IMPRIMIRÁ POR CONSOLA: *console.log***
- **Siempre que sea posible, utiliza constantes.**
- **Comenta el código de modo que sea más fácil de mantener.**
- **Establece también los comentarios de las funciones que se piden y de las que tú crees de modo que se documente la interfaz que ofrecen (descripción general de la función, parámetros de entrada y valor devuelto).**

Actividad 1. Clases *Producto* y *ArticuloFactura*

Utiliza el material de la carpeta **Actividades_1y2** para implementar las dos clases que se piden.

La página web principal de la aplicación es ***index.html***. La actividad será implementada en el archivo ***script.js*** (no hay que modificar nada más).

- Implementa una clase ***Producto*** que tiene los siguientes elementos:

Atributos privados:

- **id** – número entero que identifica al producto. Por ejemplo: 8.
- **descripcion** – cadena de texto que representa la descripción del producto. Por ejemplo: "Almendras bolsa 200gr".
- **precio** – número flotante que representa el precio unitario del producto. Por ejemplo: 3,12.

Métodos:

- Método **constructor** – construye un objeto **Producto** a partir de los valores de los atributos descritos.
- Métodos **get** y **set** de los atributos.
- **toString** – método que **devuelve** una cadena legible del producto incluyendo el id, descripción, precio unitario. El formato será: "(id) descripción – precio". Para el ejemplo propuesto: "(8) Almendras bolsa 200gr – 3,12€".



- Implementa la clase **ArticuloFactura** que hereda de **Producto** y que tiene los siguientes elementos:

Atributos privados:

- **cantidad** – número entero que indica la cantidad adquirida de producto. Por ejemplo: 3.

Métodos:

- Método **constructor** – construye un objeto **ArticuloFactura** a partir de los valores de los atributos descritos.
- Métodos **get** y **set** del atributo.
- **getTotal** – devuelve un número flotante que será el precio del artículo en la factura y que tendrá en cuenta el **precio** y la **cantidad** de productos adquiridos. Por ejemplo, para 3 artículos con precio 3,12 € devolverá 9,36.
- **toString** – reutilizando el código de la clase padre, devolverá una cadena de caracteres que muestre la siguiente formato: “(id) descripción – precio x cantidad”. Para el ejemplo propuesto: “(8) Almendras bolsa 200gr – 3,12€ x 3”

Actividad 2. Uso de la clase *ArticuloFactura*

Utiliza el material de la carpeta **Actividades_1y2** para implementar esta actividad. Es una actividad continuación de la primera.

Crea una función llamada **crearFactura** (que se ejecutará al cargar la página) en la que se cree un array con tres artículos de factura obtenidos en diferentes cantidades. Imprime una factura que contenga una línea por cada producto y que al final se imprima el total de la factura. Por ejemplo:

FACTURA:

(8) Almendras bolsa 200gr – 3,12€ x 3 – 9,36€

(12) Harina bolsa 1Kg – 2,30€ x 1 – 2,3€

(4) Piña conserva lata 500gr – 2,10€ x 4 – 8,4€

TOTAL – 20,06€

Pistas:

- Para realizar esta actividad puedes usar los métodos de **Array** tanto para imprimir cada línea, como para obtener el total de la factura (método **reduce** de **Array**). Si no, recorre el array con un bucle.
- Recuerda que **Number** tiene un método que redondea un número flotante al número de decimales que queramos.



Actividad 3. Funciones y arrays

Utiliza el material de la carpeta **Actividad_3** para implementar todos los apartados de esta actividad.

Notas:

- En el archivo *script.js* se encuentra el array **factura** que se utilizará para implementar esta actividad.
- Siempre que puedas, utiliza funciones del objeto **Array**.

Apartado 1. Crea una función llamada **obtenerCarosOrdenadosPorCantidad** que reciba el array **factura**. Debe devolver un nuevo array con los productos con precio superior a 2€, ordenados por la cantidad de veces que se ha pedido.

Crea código que imprima por pantalla el resultado devuelto por la función anterior, debiendo imprimir todos los atributos. Un ejemplo de impresión para un producto sería:

id: 3, descripcion: Café molido 250gr, precio: 3.99, cantidad: 2

Apartado 2. Crea una función llamada **obtenerNMasCaros** a la que se le pasa el array **factura**, el número de elementos que se quieren devolver más caros y devolverá un array nuevo con los N elementos más caros que haya en **factura**. Por ejemplo, si invocamos **obtenerNMasBaratos(factura, 4)**, devolverá los 4 productos más caros que haya. La función debe comprobar si el número que se pasa es correcto y, en caso de no serlo, imprimir por consola un error que lo indique (cero es un número correcto y devolverá un array vacío).

Crea código que imprima por pantalla el resultado devuelto por la función anterior, debiendo imprimir todos los atributos.