

Examen Javascript DWEC

Estructura de Carpetas:

```
examen_carrito_libros/
├── assets/
│   └── datos.js
├── Libro.js
├── Carrito.js
└── main.js
└── index.html
```

Carrito de Libros



Instrucciones:

Este examen tiene como objetivo evaluar tus habilidades en JavaScript ES6 para el módulo de DWEC. Se pretende crear una **Aplicación de Carrito de compras para una tienda de Libros** en JavaScript. La aplicación permitirá a los usuarios seleccionar libros, agregarlos al carrito, aplicar descuentos, impuestos, realizar pagos y llevar un registro de las compras realizadas.

Parte 1: Archivo Libro.js (Función Constructora para Usuarios) (2p)

Crea una función constructora llamada **Libro** que permita crear objetos de libro con las siguientes propiedades:

- ISBN: El código ISBN único del libro..
- Título: El título del libro.
- Número de Páginas: El número total de páginas en el libro.
- Autor: Un objeto de tipo Autor.
- Editorial: La editorial que publicó el libro.

- Tags: Una lista o array de etiquetas relacionadas con el contenido del libro (por ejemplo, "Programación", "Ficción", "Ciencia"). Precio: El precio del libro. Valoración: La valoración del libro en forma de estrellas (entre 1 y 5 estrellas). Stock: La cantidad de copias disponibles en el inventario.

Métodos:

- modificarPrecio(nuevoPrecio): Permite modificar el precio del libro. El precio no puede ser negativo.
- modificarValoración(nuevaValoración): Permite modificar la valoración del libro. Las valoraciones deben de ser entre 1 y 5 estrellas obligatoriamente.
- modificarStock: Permite modificar el stock disponible del libro. El stock no puede ser negativo.
- addTags(nuevosTags): Permite añadir nuevos tags a la vez al libro, no pudiendo estar ya repetidos. (ejemplo: "Programación", "Js")

Parte 2: Archivo Carrito.js (Clase Carrito) (3,5p)

Crea una clase llamada **Carrito** que representará un carrito de compras. El carrito estará implementado como un array de objetos. La clase debe tener los siguientes métodos:

- agregarProducto(ISBN, cantidad): Añade un producto al carrito. Debes proporcionar el ISBN del libro que deseas agregar y la cantidad de copias que deseas comprar. *Asegúrate de verificar si el libro existe en /assets/datos.js y si hay suficiente stock disponible antes de agregarlo al carrito.*
- eliminarProducto(ISBN): Elimina un producto del carrito. Debes proporcionar el ISBN del libro que deseas eliminar, quitando dicho libro del carrito de la compra y estuviera.
- agregarDescuento(): Aplica un 10% de descuento al precio de cada libro, si en el carrito hay más de 3 libros.
- calcularTotalCarrito(): Calcula y devuelve el precio total de todos los productos en el carrito, aplicando descuentos e impuestos(iva del 10%).
- validarCompra(): Valida la compra y realiza las siguientes acciones:
 - Guarda la información de la venta en una lista o Array de objetos en el LocalStorage en la clave "ventas". La información de la venta debe incluir el ISBN del libro, el nombre del libro, la cantidad comprada y el total pagado.

Parte 3: Otras funcionalidades (3,5p)

- menorStock(): Implementa un método para encontrar los libros 3 con menos stock disponibles y devolverlos en un conjunto (SET). (1p)
- buscarLibro(Título o Editorial): Implementa método de búsqueda para encontrar libros por título o por editorial en la lista de libros disponibles. (1p) (este método ha de estar documentado) (0,5p)
- libroMasVendido: Implementa un seguimiento de las "ventas" en el LocalStorage para saber qué libro se ha vendido más y que está guardado en el localStorage.(1p)

Parte 4: Archivo main.js (Implementación de la Red Social) (1p)

En el archivo main.js, implementa las clases y funciones constructoras definidas en Libro.js y Carrito.js. Además, realiza consultas y pruebas relacionadas con la funcionalidad de un carrito de libros. Organiza

la lógica de tu venta en este archivo.

Asegúrate de que el archivo `index.html` incluye los scripts para cargar tus clases y lógica JavaScript.

Nota: Puedes realizar las consultas a la estructura de datos proporcionada en `assets/datos.js` para obtener las respuestas a las partes 1, 2 y 3 del examen si lo estimas oportuno (aunque no es necesario).

Además se deben de documentar **Todas** las cabeceras de los ficheros javascript que se entreguen.

Entrega:

- Debes presentar los archivos de acuerdo a las instrucciones. Para ello se comprimirá todo nuestra carpeta de examen a excepción de la carpeta `node_modules`. Dicho comprimido con vuestro nombre se adjuntará a la tarea de Moodle habilitada para el examen.
- Esta prueba evalúa los *Resultados de Aprendizaje 2,3,4* del módulo de DWEC, a través de la implementación correcta de la sintaxis de Javascript, usando estructuras de datos, funciones constructoras y clases.