

Ejercicio carrito

1.- Breve descripción del ejercicio

En este ejercicio se pide implementar un carrito de la compra de tal manera que pueda observarse el uso de módulos en la resolución de este.

Para poder obtener esta visión de módulos, hemos dividido esta pequeña “app” en diferentes archivos/módulos haciendo de este proyecto una versión modular que permite cambiar e importar los diferentes módulos según se vayan necesitando en el proyecto.

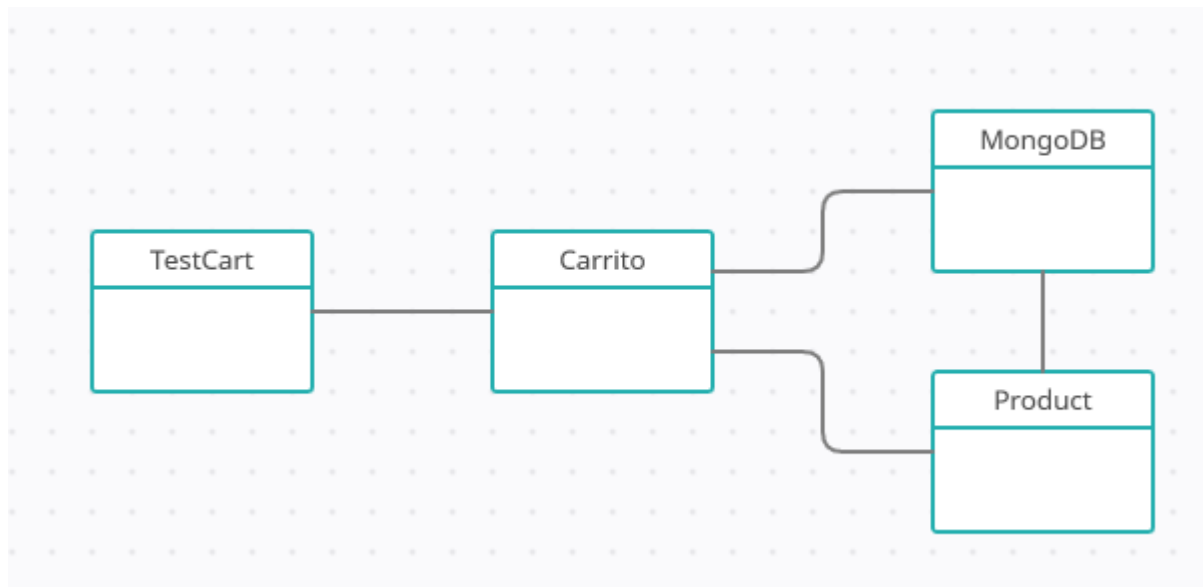


Figura 1.- Diagrama de clases de la solución presentada.

Para desarrollar esta actividad se ha optado por utilizar el lenguaje *Typescript* en vez de *Javascript*. El proceso para compilar el código a *Javascript* y ejecutarlo está detallado en otro apartado a continuación.

Además, en vez de almacenar los datos localmente, se ha optado por utilizar el servicio de base de datos en la nube de Mongo, llamado *MongoDB Atlas*, para almacenar los productos insertados por el script que gestiona las conexiones con *MongoDB*.

2.- Explicación de las clases propuestas en la solución

La solución explorada cuenta con 4 archivos *Typescript* y sus correspondientes compilaciones en *Javascript*. A continuación, revisaremos la funcionalidad de cada una de ellas:

- **MongoDB:** se incorporan todos aquellos métodos correspondientes al acceso a la base de datos (aquellos métodos relativos al ejercicio 2).
- **Producto:** definición de la clase producto. Contiene atributos, constructores y el método `toString()`.
- **Carrito:** define el carrito propiamente. Se encarga de gestionar los productos añadiendo o borrando nuevos, además de tener control sobre ellos. Además permite

mostrar al usuario los productos almacenados (aquellos métodos relativos al ejercicio 1 y ejercicio 2, es decir, adaptando los primeros para que funcionen junto a la clase `MongoDB`).

- **TestCart:** este script se encarga de ejecutar dos funciones: la primera, para probar la funcionalidad base del carrito comprobando que puede añadir y borrar productos, y la segunda, prueba esto último pero esta vez con la inclusión de la base de datos de *MongoDB* insertando en ella productos para comprobar su correcto funcionamiento y comprobando stock antes de insertar un producto en el carrito.

3.- Pasos a seguir para su compilación y ejecución

Para probar esta funcionalidad y debido a que se ha implementado en *Typescript*, será necesario tener el compilador de *Typescript* instalado, siendo muy fácil hacerlo teniendo previamente instalado *Nodejs* y *npm* mediante el comando: `npm install -g typescript`

Una vez instalado, es necesario compilar cada archivo `.ts` para ello usaremos el comando: `tsc <nombre_archivo>.ts`, de esta forma, obtendremos un archivo *JavaScript* autogenerado para cada contraparte *Typescript*. Cuando estén todos los archivos compilados, lo único que tendremos que hacer es ejecutar el archivo de tests mediante el comando: `node TestCarts.js`

Hemos incluido los archivos compilados a *JavaScript* en caso de querer saltarse el paso de compilación. Para leer el código, resultará más cómodo ver los archivos *Typescript*, ya que los *JavaScript* autogenerados por el compilador tienen un formato poco legible.

Por último, para proveer el código fuente, se han adjuntado tanto un zip con los archivos como un link al repositorio de GitHub mostrado a continuación: https://github.com/pabletefest/SAD_MUIINF