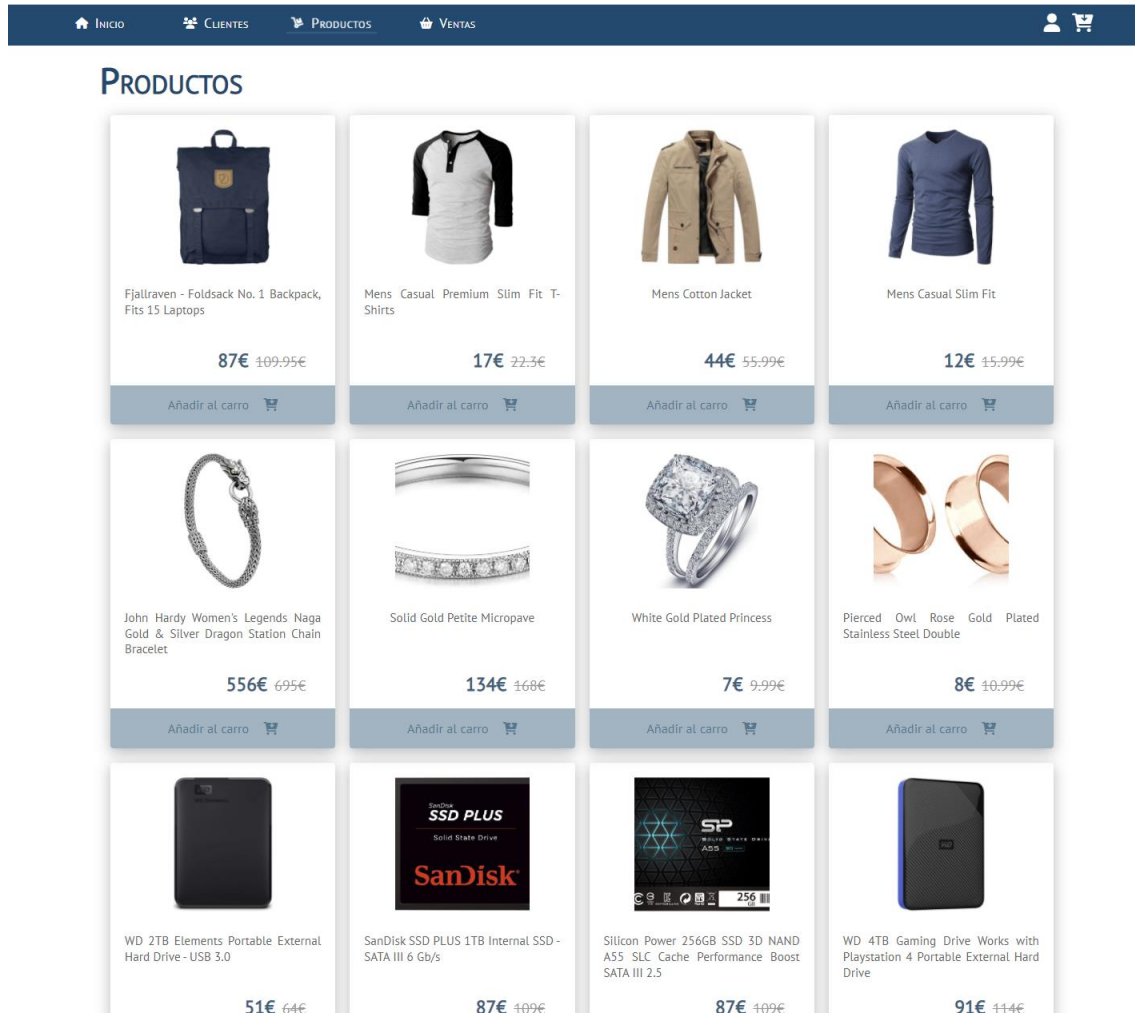


El examen consiste en la realización de una “tienda” para la compra de productos y para la consulta de las ventas. Es decir, el examen tiene dos partes que se consideran dos ejercicios separados y que se evalúan por separado.

Para la realización de las dos partes tendremos tres clases una para Cliente, otra para Productos y otra para Ventas.

EJERCICIO 1

Es una aplicación para realizar compras en nuestra tienda y aparece en nuestro index.



Tendremos 20 Productos, con los siguientes datos:

- ✓ Código (**id**)
- ✓ Nombre (**title**)
- ✓ Precio (**price**)
- ✓ Imagen (**image**)

Estos datos los obtendremos de hacer una consulta al API <https://fakestoreapi.com>, en la que pediremos 20 productos, los datos que utilizaremos de la petición son los que aparecen en negrita junto con el nombre del campo.

Vamos a necesitar una clase Productos para crear los productos y un array con los 20 productos.

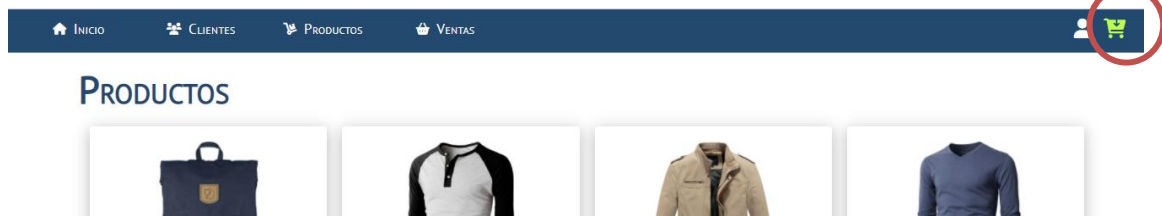
Cuando inicie nuestra tienda se cargarán los productos en las tarjetas con las informaciones obtenidas.

El usuario podrá realizar la compra del producto que desee, para ello primero tendrá que añadirlo al carro el producto (**ver video**).

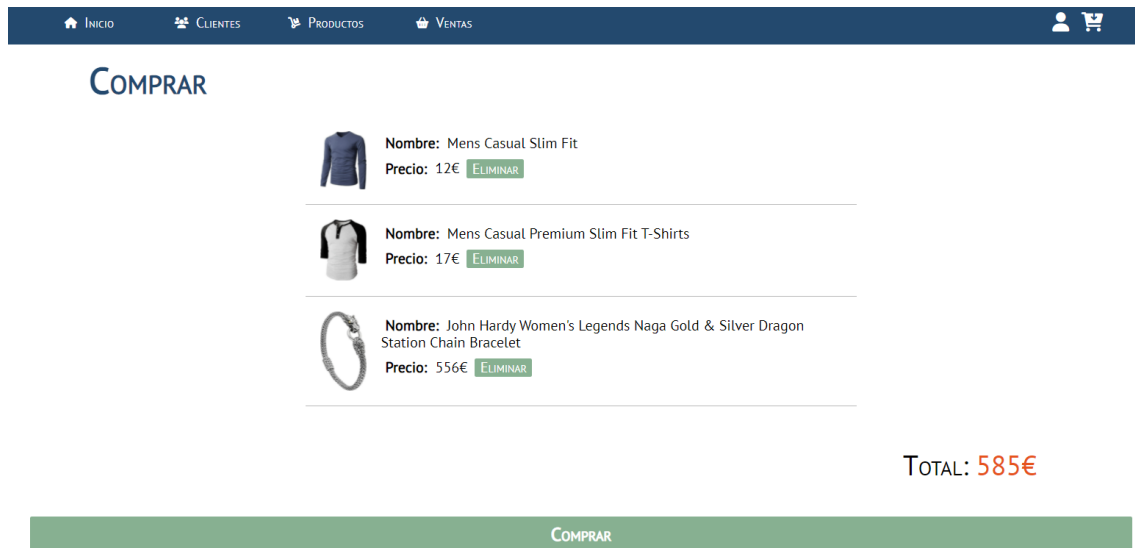
Los productos añadidos al carro serán almacenados en el localStorage, de tal forma que estarán siempre disponibles para poder acceder a ellos.

Las informaciones que se añaden a localStorage son el nombre del producto, el precio y la imagen que se obtendrán de la tarjeta selecciona, recuerda como se accede al elemento padre, a los hermanos.....

Los productos se irán añadiendo al carro hasta que el usuario decida realizar la compra, pulsado el icono del carro en la parte superior derecha.



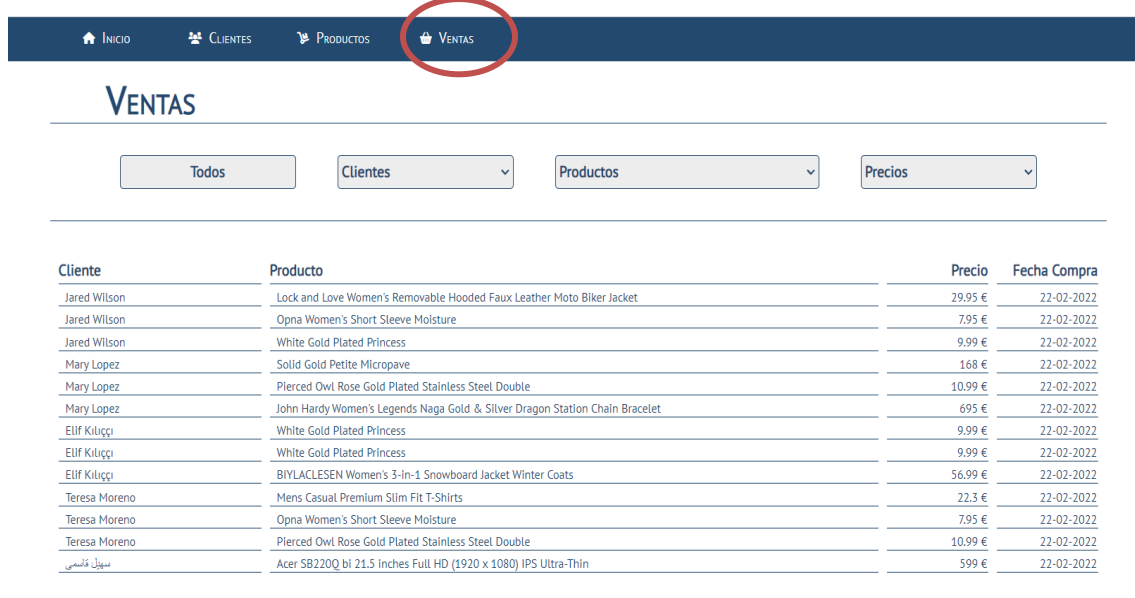
Una vez que accedemos al carro, nos aparecerán los productos que hemos añadido anteriormente, con el total de la compra.



Si el usuario realiza la comprar los productos serán eliminados del localStorage y **no hay que realizar ninguna operación más.**

EJERCICIO 2

Consiste en la “gestión” de las ventas.



Tendremos 20 Productos, con los siguientes datos:

- ✓ Código (**id**)
- ✓ Nombre (**title**)
- ✓ Precio (**price**)
- ✓ Imagen (**image**)

Estos datos los obtendremos de hacer una consulta al API <https://fakestoreapi.com>, en la que pediremos 20 productos, los datos que utilizaremos de la petición son los que aparecen en negrita junto con el nombre del campo.

Vamos a necesitar una clase Productos para crear los productos y un array con los 20 productos.

Tendremos 20 Clientes, con los siguientes datos:

- ✓ Código (login.uuid)
- ✓ Nombre (name.first)
- ✓ Apellidos (name.last)
- ✓ Email (email)

Estos datos los obtendremos de hacer una petición al API <https://randomuser.me/documentation#howto>, en la que pediremos datos de 20 usuarios. los datos que utilizaremos de la petición son los que aparecen en negrita junto con el nombre del campo.

Vamos a necesitar una clase Clientes para crear los clientes y un array con los 20 clientes.

Tendremos 60 Ventas, con los siguientes datos:

- ✓ Cliente
- ✓ Producto
- ✓ Fecha

Cómo no tenemos una API con las ventas, vamos a generarlas de forma automática. Para ellos vamos a generar tres ventas por cada uno de los clientes, es decir, todos los clientes van a realizar la “compra” de tres productos. Para seleccionar los productos que ha comprado un cliente, seleccionaremos tres productos de forma aleatoria del array de productos

Cada compra tendrá el objeto con los datos de cliente que ha hecho la compra, el objeto del producto que se ha comprado, con sus datos y la fecha de la compra.

Cuando accedemos a ventas se mostrará un listado con todas las ventas, se mostrarán las 60 ventas que hemos generado

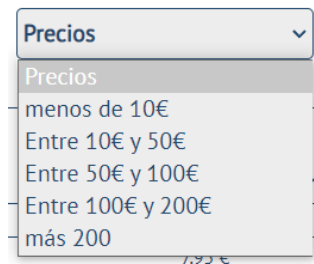
Las acciones que podremos hacer en ventas son consultas sobre todas las ventas. **Siempre que se haga una consulta se realizará sobre todas las ventas.**

Cuando accedemos a ventas se mostrará un listado con todas las ventas, se mostrarán las 60 ventas que hemos generado

Se podrán realizar las siguientes consultas:

- ✓ Todas, se realiza sobre el botón de todas y se mostrarán todas las ventas, igual que hacemos cuando accedemos a ventas.
- ✓ Ventas de un Cliente, se realiza sobre un select donde tendremos que cargar todos los nombres de los clientes. Las consultas se realizan por el código del cliente. Y nos mostrará las ventas del cliente seleccionado.
- ✓ Ventas de un producto, se realiza sobre un select donde tendremos que cargar todos los nombres de los productos. Las consultas se realizan por el código del producto. Y nos mostrará las ventas del producto seleccionado.

- ✓ Precios de los productos, se realiza sobre un select donde tenemos los siguientes rangos de precio, nos aparecerán las ventas de los productos que se encuentren en esos rangos



A screenshot of a web application's dropdown menu. The dropdown is titled 'Precios' and is currently open, showing a list of price ranges. The options are: 'menos de 10€', 'Entre 10€ y 50€', 'Entre 50€ y 100€', 'Entre 100€ y 200€', and 'más 200'. The dropdown is styled with a light blue header and a light gray body.

Precios
menos de 10€
Entre 10€ y 50€
Entre 50€ y 100€
Entre 100€ y 200€
más 200