PRÁCTICA: SERVICIO DE COMPRA ONLINE EN *MERCADAW*

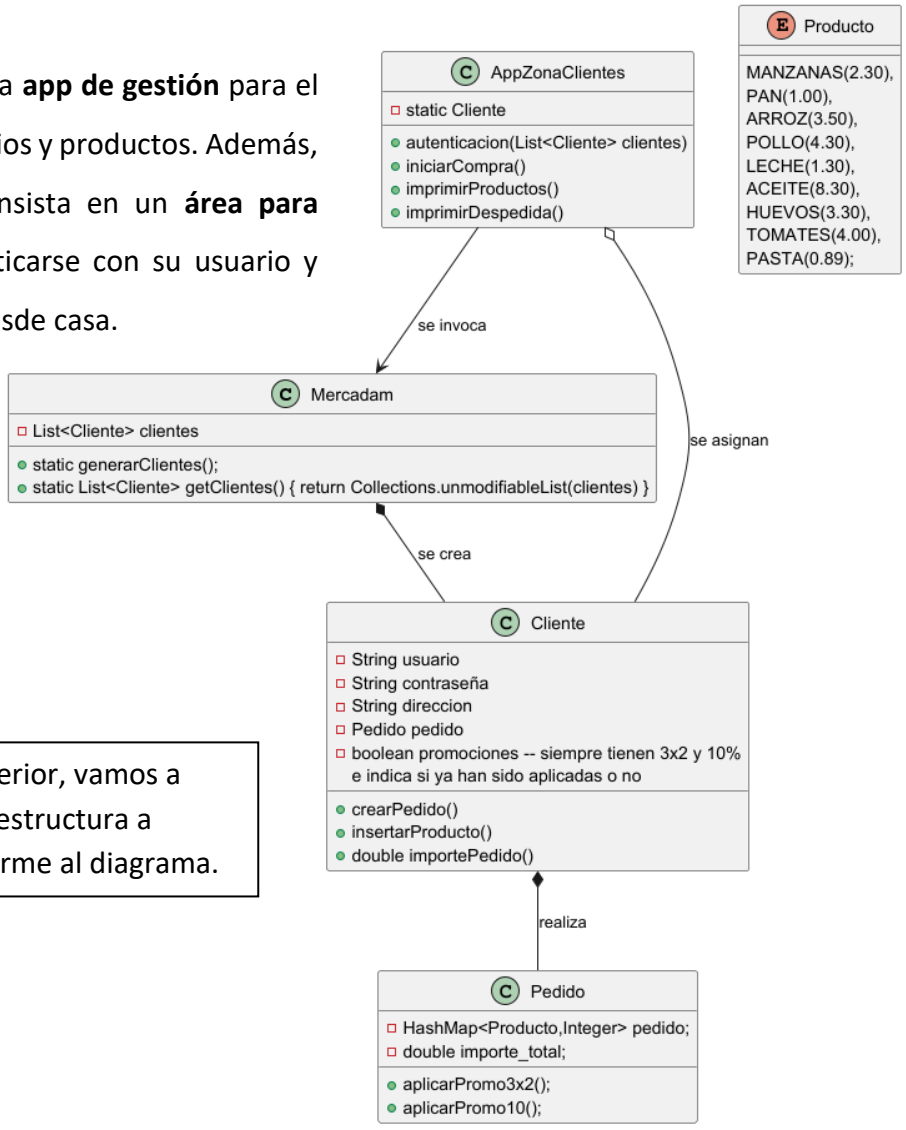
➔ INTRODUCCIÓN

Tenemos nuevo supermercado en la ciudad. Puede que te suene el nombre, porque sus dueños, cabreados con las políticas de empresa del súper original, han querido montar su propia tienda.

Para no ser menos, tienen claro que quieren implementar la famosa compra online con su correspondiente reparto a domicilio. Como no tienen ni idea de informática, deciden acoger en prácticas a 4 estudiantes del instituto IES MUTXAMEL para que los asesoren un poco y construyan una app que, aunque de momento no sea bonita, implemente toda la lógica necesaria.

➔ PROBLEMA A RESOLVER

El supermercado necesita una primera **app de gestión** para el personal, desde donde se crean usuarios y productos. Además, necesitan una segunda app que consista en un **área para clientes**, donde estos puedan autenticarse con su usuario y contraseña para realizar un pedido desde casa.



Teniendo en cuenta todo lo anterior, vamos a desarrollar un software cuya estructura a implementar se ha diseñado conforme al diagrama.

Para ir probando esta estructura, el programa principal ***AppZonaClientes*** debe iniciar creando una instancia de *Mercadaw* y generando a clientes aleatorios de prueba (***generarClientes()***). El constructor de

la clase *Mercadaw* no recibe nada y genera ***usuario*** y la ***contraseña*** de tamaño 8 con caracteres *random*. Usa esto:

String caracteres = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

Pedido se iniciará a *null*, y ***promociones*** a *false*. La ***dirección*** siempre será "*Calle falsa, 123*".

- ***autenticacion(List<Cliente> clientes)***. Imprimirá y pedirá por pantalla lo siguiente:

```
*** COMPRA ONLINE DE MERCADAM ***

Usuario: patri
Constraseña: compra_patri
```

Cotejará los datos introducidos por el cliente contra la lista de clientes que recibe desde la clase *Mercadaw* para validar las credenciales. **Imprime la lista que recibes para no inventarte las pruebas.**

- En caso de que el *usuario* o *contraseña* no coincidan, se informará de que las credenciales no son correctas y el programa volverá a pedirlas. Tras **3 intentos**, el programa debe cerrarse mostrando un **error de autenticación**.

```
*** COMPRA ONLINE DE MERCADAM ***

Usuario: patri
Constraseña: compra_patri

Algo no coincide o no existe! Vuelve a intentarlo...

Usuario: patri
Constraseña: compra_patri2

Algo no coincide o no existe! Vuelve a intentarlo...

Usuario: patri
Constraseña: compra_patri3

ERROR DE AUTENTICACIÓN.
```

- En caso de que el usuario exista en la lista, se lo asignaremos a la variable de tipo *Cliente* (estático) que contiene la clase *AppZonaClientes* y llamaremos a *iniciarCompra()*.
- ***iniciarCompra()*** creará un nuevo pedido para el *Cliente*, inicializando su atributo *Pedido = new HashMap<Producto,Integer>*.

Además, llamará al método ***imprimirProductos()*** que mostrará el contenido del ***enum:***

```
BIENVENID@ patri!
Añade productos a tu lista de la compra...
    MANZANAS precio (2.30€),
    PERAS precio (1.30€),
    PAN precio (1.00€),
    ARROZ precio (3.50€),
    POLLO precio (4.30€),
    LECHE precio (1.30€),
    ACEITE precio (8.30€),
    HUEVOS precio (3.30€),
    TOMATES precio (4.00€),
    PASTA precio (0.89€);
========================================
    Elige un producto:
```

- Recogeremos desde el programa principal la opción escogida por el usuario y llamaremos a ***insertarProducto(String producto)***. Este método añadirá el producto escogido al *Pedido*. En caso de que ya exista un producto del mismo tipo, incrementaremos la cantidad (+1ud).

  CUIDADO: en caso de que el producto no exista, se debe mostrar un ERROR e imprimir de nuevo la lista de productos disponibles:

```
    TOMATES precio (4.00€),
    PASTA precio (0.89€);
========================================
    Elige un producto: MACARRONES
========================================
El producto no existe! Elige otro.
Añade productos a tu carrito de la compra...
    MANZANAS precio (2.30€),
    PERAS precio (1.30€),
    PAN precio (1.00€),
    ARROZ precio (3.50€)
```

- Si todo va bien y el producto es correcto, se informa al cliente y se muestra un resumen del importe acumulado en el carrito ***(importePedido()***):

```
    TOMATES precio (4.00€),

    PASTA precio (0.89€);

==========================================

    Elige un producto: PASTA

==========================================


Has añadido PASTA con un precio de 0.89€. Importe total del carrito: 0.89€. ¿Quieres añadir más productos a tu carrito de la
compra? [S/N]:
```

Además, se pregunta al usuario si se quieren añadir más productos o finalizar. En caso de responder "*S*", se repite el proceso.

```
Has añadido PASTA con un precio de 0.89€. Importe total del carrito: 0.89€. ¿Quieres añadir más productos a tu carrito de la
compra? [S/N]:

S

==========================================

Añade productos a tu carrito de la compra...

    MANZANAS precio (2.30€),

    PERAS precio (1.30€),

    PAN precio (1.00€),

    ARROZ precio (3.50€),
```

```
    LECHE precio (1.30€),

    ACEITE precio (8.30€),

    HUEVOS precio (3.30€),

    TOMATES precio (4.00€),

    PASTA precio (0.89€);

==========================================

    Elige un producto: TOMATES

==========================================


Has añadido TOMATES con un precio de 4.00€. Importe total del carrito: 4.89€. ¿Quieres añadir más productos a tu carrito de la
compra? [S/N]:
```

En caso de no querer añadir más productos, se debe actualizar el atributo ***importe_total*** del *Pedido* y **mostrar un resumen** de la compra realizada:

```
Has añadido TOMATES con un precio de 4.00€. Importe total del carrito: 4.89€. ¿Quieres añadir más productos a tu carrito de la
compra? [S/N]:

N

==========================================

RESUMEN DE TU CARRITO DE LA COMPRA:

Productos:

1 PASTA  0.89
1 TOMATES 4.00
3 PAN 1.00

IMPORTE TOTAL: 7.89€
```

En este punto, mostraremos las siguientes opciones al cliente:

[1]. Aplicar promo.

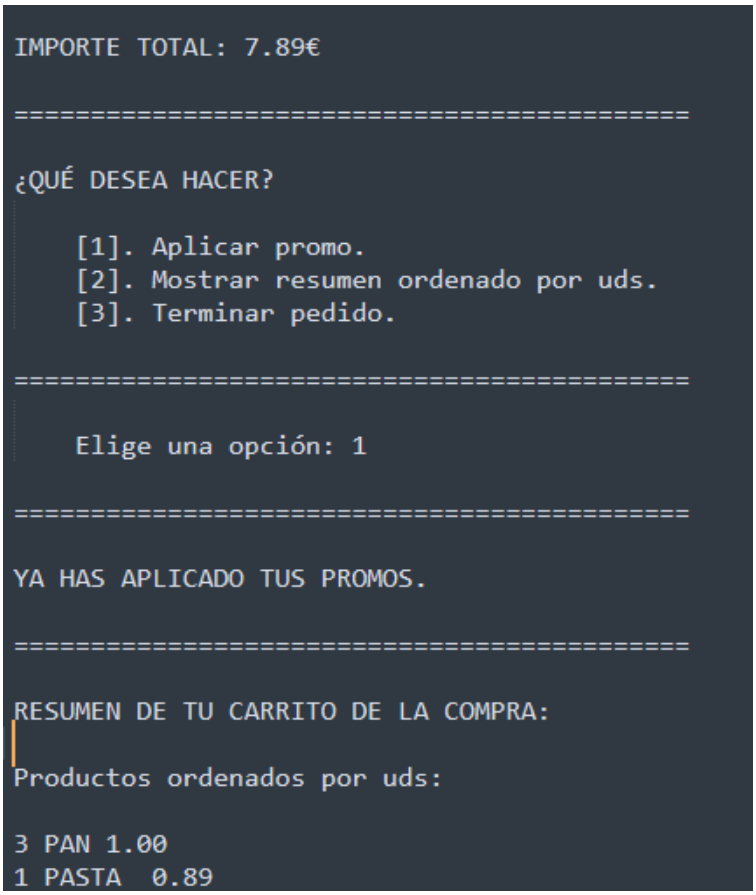
[2]. Mostrar resumen ordenado por uds.

[X]. Terminar pedido.

- En caso de querer terminar, imprimiremos un mensaje de despedida dando las gracias e indicando la dirección del cliente.

```
==========================================

RESUMEN DE TU CARRITO DE LA COMPRA:

Productos:

1 PASTA  0.89
1 TOMATES 4.00
3 PAN 1.00

IMPORTE TOTAL: 7.89€

==========================================

¿QUÉ DESEA HACER?

    [1]. Aplicar promo.
    [2]. Mostrar resumen ordenado por uds.
    [3]. Terminar pedido.

==========================================

    Elige una opción: 3

==========================================

GRACIAS POR SU PEDIDO. Se lo mandaremos a la dirección Calle Falsa, 123.
```

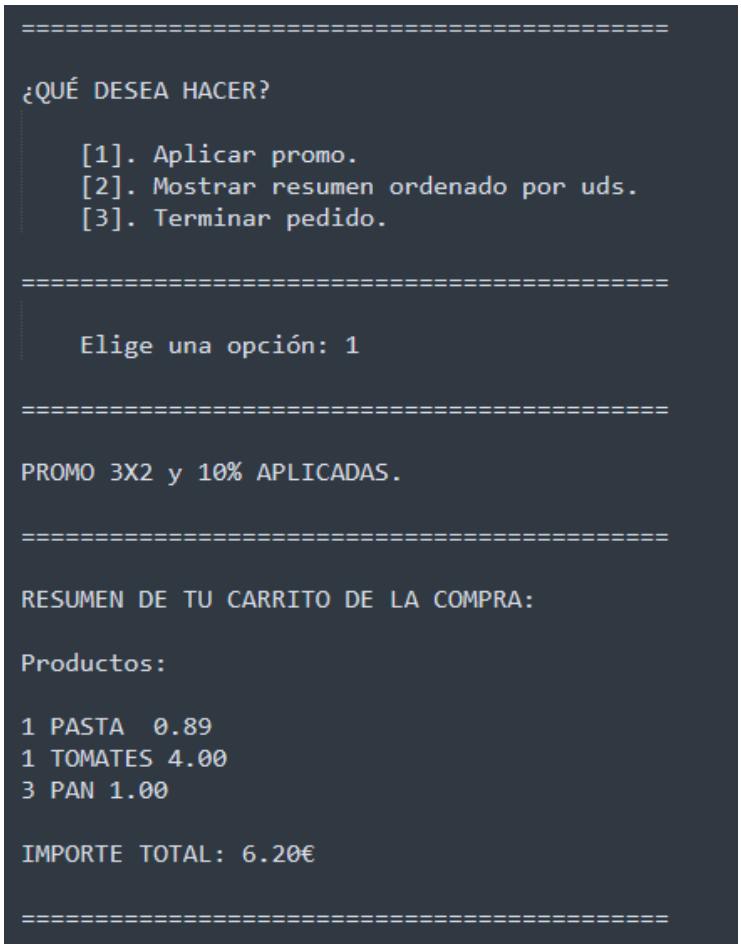
- Si decidimos ***aplicar promo***, deberemos comprobar que no se hayan aplicado ya al mismo cliente (promociones = *false*). Si ya las hemos aplicado, no haremos nada y mostraremos un mensaje para informar al cliente de que ya ha aplicado sus promos.



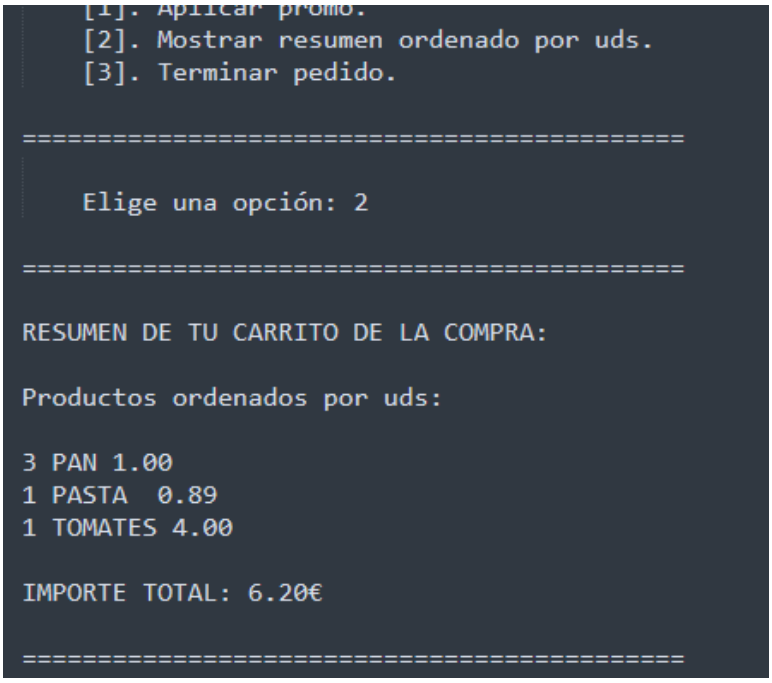
En caso de que el cliente todavía no haya usado sus promociones, aplicaremos todas las que tenemos disponibles: **3x2 en productos y 10% de descuento**. Por lo tanto:

- Deberemos recorrernos nuestro pedido en busca de aquellos productos de los cuales existan 3 uds (o múltiplos de 3 uds) para recalcular el *importe_total* (sólo cobraremos 2).
- Al *importe_total* obtenido después de aplicar la promo anterior, deberemos aplicarle un 10% de descuento más.
- Modificaremos el atributo *promociones* del cliente a ***true***.

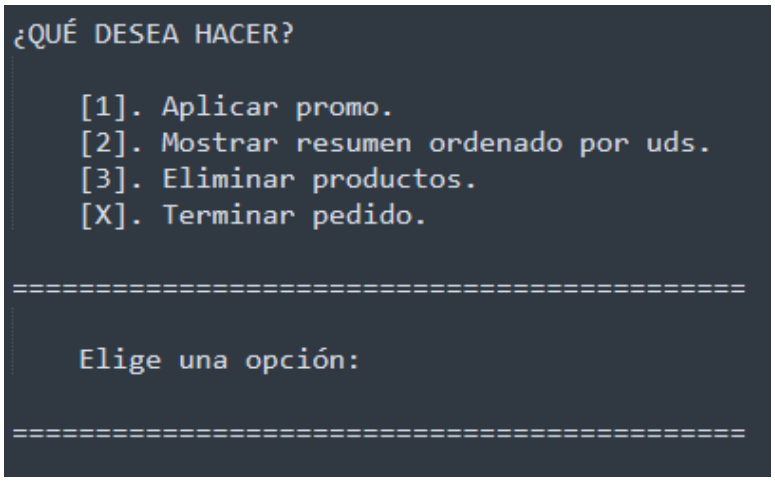
Finalmente, imprimiremos de nuevo el resumen del pedido con las promos aplicadas y el importe total actualizado:



- Si decidimos usar la opción 2, deberemos mostrar el resumen del pedido ordenado por uds descendentemente. Por ejemplo:



BONUS. Implementa una funcionalidad añadida para que el programa permita eliminar productos del carrito de la compra. En caso de que algún tipo de producto quede con 0 uds, debe eliminarse completamente de la lista (mapa).



Realiza un programa en *Java* que implemente la lógica de la aplicación dada, usando *POO* y estructuras dinámicas de datos.

➔ REALIZACIÓN DE LA PRÁCTICA

Sigue los siguientes pasos para realizar la práctica. **¡Ve guardando tu trabajo de vez en cuando para evitar que se borre el avance si se cierra el editor de textos u ocurre cualquier problema en tu equipo!**

1. **Programa en Java la aplicación requerida**
2. **Sube un vídeo ejecutando tu aplicación, replicando pruebas y explicando todos los comportamientos que hayas implementado.**

ENTREGA

REALIZA UN INFORME EN PDF CON LA INFO GENERADA Y LOS PASOS SEGUIDOS PARA REALIZAR ESTA PRÁCTICA. EXPLICA TU CÓDIGO. SÚBELO TODO A LA TAREA DE AULES DISPONIBLE.

ADEMÁS, PEGA LA URL DE TU PROYECTO EN GITHUB.