

3 EVAL PROGRAMACIÓN

Tenemos una base de datos para gestionar las compras en una plataforma online.
Hay dos tipos de clientes, los normales y los que tienen tarjeta, que además acumulan puntos.
De todos se guarda su código de usuario, su password y su teléfono.
Se almacenan los productos en una tabla, cada producto tiene su identificador de producto, su nombre, su precio y las unidades en stock.
En la tabla carrocompra almacenamos los productos que el cliente ha seleccionado pero aún no ha comprado. Cada línea de esta tabla tiene el código de usuario, el número que ocupa el artículo en el carrito, el código del producto, las unidades y el precio unidad.
Haz una correcta jerarquía de clases usando herencia **(0,5 ptos)**

Repetid el siguiente proceso mientras el usuario no decida cerrar el programa.

Pedid el código y la password al usuario que se quiere conectar, extraed su información de la tabla de clientes de la base de datos. **(0,5 ptos)** Se permiten tres intentos fallidos de acceso, al tercero se lanza una excepción propia ClienteBloqueado y se da de alta una incidencia en un archivo de texto en el que se guarda la fecha y hora de este último intento fallido, así como el código del usuario y durante el minuto siguiente no se permite que se conecte ningún usuario. **(1 pto)**

Si todo ha ido bien accedemos a la base de datos y creamos un vector con los productos que tenga ese usuario en el carrito. **(0,5 pto)** Usad una clase Carrito en la que además del vector anterior se guarde el importe total, y el número total de unidades. Cread un interface Almacenable, que tenga un método vaciar, y otro mostrar, haced que la clase Carrito implemente dicho interface **(0,75 pto)**

Haced un menú para realizar las siguientes opciones:

1. **Añadir artículo al carrito.** Se anota el código del producto (se valida que tenga de dos a tres letras y de dos a tres dígitos) y el número de unidades a comprar y se accede a la BBDD, si no existe o no hay suficientes unidades se lanza la excepción correspondiente y se vuelve al menú. **(1 pto)** En otro caso se añade el producto al vector-carrito, teniendo en cuenta que si ya estaba en el vector-carrito sólo se incrementa el número de unidades. **(0,5 pto)** Volvemos al menú.
2. **Borrar del carrito un número de unidades de un determinado producto dado el código de éste.** **(0,5)** Volvemos al menú.
3. **Comprar los productos del carrito-vector:** Se calcula el importe de la compra usando polimorfismo: Si se trata de un usuario normal se le cobran además 7 euros por el envío si éste es inferior a 100 euros, y si es un cliente con carné, el envío es gratis, y por cada 20 euros de compra se generan 3 ptos. Vaciar el carrito **(1,5 ptos)** Actualizar la información necesaria en las tres tablas de la BBDD **(1,5 ptos)** Volvemos al menú.
4. **Logout.** Se actualiza en la BBDD la tabla carrocompra con la información del vector-carrito. **(0,75 pto)** Volvemos al inicio del programa, a pedir de nuevo usuario y password.
5. **Cerrar el programa.**
 - a. Mostrad de qué productos se han comprado más unidades usando un array de frecuencias. **(0,5 ptos)**
 - b. Dada una fecha, mostrar todos los accesos fallidos anteriores a esa fecha accediendo al fichero de conexiones fallidas. **(0,5 ptos)**
 - c. Terminamos.

