

## **Sistema de venta de Zapatillas (Sneakers – Botines )**

- El sistema comienza con la carga del formulario donde se podrán visualizar dos grillas, predispuestas para la carga del stock de ambos productos, los cuales se traerán al programa por medio de sus botones correspondientes.
- El usuario podrá seleccionar un producto de la o las tablas ya cargadas y agregarlo a un carrito mediante el botón “ADD CARRITO” el cual habilitará la función de los dos botones siguientes. Cabe especificar que solo se podrá agregar de a UN producto al carrito.
- Se podrá visualizar la información del carrito mediante el botón “Ver Carrito” el cual se habilitará tras cargar algún producto al mismo, mostrando un nuevo formulario con la información de los items seleccionados, podrán ser n cantidad de “Sneakers” y solo UN “Botin”. el programa lo guiará..
- Para que el usuario finalice la compra tendrá que hacer click en el botón “Efectuar Pago” el cual desplegará un formulario para diligenciar la información de dicho usuario y tras ACEPTAR se generará un ticket y se serializará el usuario en su dispositivo.

### **TEMAS IMPLEMENTADOS**

- 1. EXCEPCIONES:** Maneje validaciones en el formulario de Clientes para controlar el casteo de información, con esto se arrojará un mensaje indicando si el dato es invalido, al igual que si se intenta agregar mas de un botin al carrito o por otro lado si se intenta agregar el mismo Sneaker al carrito.
- 2. TEST UNITARIOS:** Lo utilice para probar distintos métodos y para verificar que se lanzan excepciones pertinentes según lo estipulado por el sistema.
- 3. GENERICS:** Implemente listas genéricas como atributos en la clase “Cliente” las cuales sólo almacenan datos con los tipos de productos creados, ayudará a tener un registro de cada producto. y en el formulario principal para almacenar los index de la tabla Sneakers que se desean eliminar(Vender).
- 4. INTERFACE:** La interface fue utilizada para generar un método que permitiera la facturación del carrito al finalizar la compra, a su vez para serializar el cliente que realiza dicha compra. pueden visualizarse implementadas en la clase “Cliente”
- 5. ARCHIVOS/SERIALIZACION.** : Implemente archivos al generar un ticket con los datos del cliente y el total de su compra en el sistema, se puede encontrar implementada en la clase cliente y definida en la interface “IFactura”.  
La serialización fue utilizada para generar un archivo XML del cliente que realiza la compra en el sistema, esto contendrá toda su estructura de datos. se definió en la interface “Serializar” e implementó en la clase “Cliente”
- 6. BASE DE DATOS Y SQL:** Fue utilizado a la hora de traer los stocks de productos al sistema, por medio de un dataadapter y un datatable que configurado permiten la

visualización e interacción, además al momento de realizar la venta se realiza una baja de la misma según el o los productos seleccionados y agregados al carrito.

**7. THREADS:** El tema de hilos fue utilizado para sincronizar la base de datos en memoria con el motor de la misma, con ello se finaliza la compra y actualiza la información.

**8. EVENTOS :** Se hizo uso de eventos al momento de seleccionar dos productos al tiempo (de diferente DataTable) para agregar al "CARRITO", si ello ocurre se dispara el evento que informara la imposibilidad de la acción. y que será capturado al hacer click en "Add Carrito"

**9. METODOS DE EXTENSIÓN:** Fue agregado a la clase "Cliente" y servirá para hacer un mapeo de la información suministrada por la base de datos y la necesaria para instanciar un objeto del tipo cliente. Recibirá un string y devolverá un dato de tipo "EMetodoPago"