

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL AVELLANEDA
T.P. N°4 DE LABORATORIO DE PROGRAMACIÓN II

Autor: Lisbaldy Ojeda DNI 95.878.772 2-D

Funcionalidad Pretendida:

Sistema de gestión de ventas para una tienda de artículos para Pilotos Aeronáuticos que soporte carga de nuevos productos, ventas, muestre el stock y los precios de todos los productos, soporte operaciones con base de datos, calcule descuentos según características del cliente y se refresque la información mostrada inmediatamente después de cada acción o cambio realizado.

Implementado:

1. Carrito de compras.
2. Alta de nuevo producto (Se carga en base de datos).
3. Venta de producto (Se serializa la venta simulando un ticket de venta).
4. Visualización de productos disponibles en el comercio incluido su stock y precio unitario.
5. Visualización de productos cargados al carrito de compras.
6. Validaciones para no comprar productos sin stock o con menor cantidad a la deseada.
7. Validaciones de string, int, doublé, emails con expresión regular, entre otras.

El sistema consta de 2 dataGridView donde se muestran los pedidos disponibles y los pedidos cargados al carrito de compras.

Se posee 3 formularios en total, a parte del principal, se utiliza un formulario para la carga de un nuevo producto y se utiliza otro formulario para la carga del cliente que efectúa la compra.

Inicialmente se leen los productos desde una base de datos SQL, se cargan en memoria y se muestran en un dataGridView.

La solución se encuentra dividida en 4 proyectos:

1. **Window Form:** Posee el nombre de la solución. Aquí se implementa la interfaz gráfica, es el proyecto principal del programa.
2. **Librería de clases:** Llamado Entidades, se implementa toda la lógica de negocio del programa.
3. **Unit Test:** Llamado Tests_Unitarios, se implementan los tests.
4. **Consola:** Llamado Test, se muestran algunas funcionalidades básicas del programa.

Temas y su implementación:

1. **Excepciones:** Se utilizan bloques try catch en la clase BaseDatos para capturar cualquier excepción relacionada a operaciones con la base de datos. Se implementó una excepción propia con mensaje personalizado llamada FailSqlOpException. En el método TraerProductos() de BaseDatos se hizo uso del concepto inner Exception.
2. **Tests Unitarios:** Se creó un proyecto único para este propósito el cual contiene 3 métodos divididos en 2 clases, una para test de funcionalidades con la clase Cliente y otra para tests de funcionalidad al calcular costos finales de cada venta (Se brinda descuento del 13% si el cliente se apellida "Ojeda").
3. **Tipos Genéricos:** Se utilizan en la clase Archivos para implementar la serialización de Objetos. Se utilizó la restricción ": class" para que el argumento de tipo sea un tipo de referencia.
4. **Interfaces:** Se implementó la interfaz iPersona para ser implementada por la clase cliente.
5. **Serialización:** Se serializan las ventas en un archivo XML que en su nombre posee la fecha de dicha venta.
6. **Base de Datos SQL:** Se creó la clase estática BaseDatos para manejar todas las operaciones con la base de datos (agregar productos, traer un producto, traer todos los productos, eliminar un producto y modificar el stock de un producto).
7. **Hilos:** Se implementó hilos para el manejo del formulario de carga de nuevos productos, se pueden generar cuantos formularios se quiera y cada uno será controlado por un hilo propio permitiendo el uso del formulario principal mientras se mantiene abierto uno o más formularios de carga de nuevos productos. Al cerrar el formulario principal se anulan todos los hilos secundarios (Se hizo uso de una colección List<> para el manejo de los hilos secundarios).
8. **Eventos y Delegados:** Se implementó el evento evento segundosCumplidos que implemente el delegado ManejadorTiempo en el formulario principal HomeForm para una pequeña funcionalidad que dispare un MessageBox con botones al cumplirse cierto tiempo, consultando si se desea cerrar el programa o no, posee un timer para el conteo de tiempo. (Actualmente se implementó para 20 segundos solo como muestra).
9. **Métodos de Extensión:** Se hizo uso de un método de extensión para extender la funcionalidad de las instancias de string, dicho método permite convertir el primer carácter de la cadena seleccionada a mayúscula.

NOTA: El proyecto de Consola no posee muchas funcionalidades ya que es un código hardcodeado (y por falta de tiempo no se pudo elaborar mejor), no lo recomiendo para mostrar operaciones con la base de datos, se recomienda probar el proyecto de Windows Forms.

NOTA: Al hacer doble clic sobre un fila se podrá cambiar el valor de dicha celda, esto es debido a que para serializar se necesitan las propiedades set y get de los campos de dichos objetos.

Instrucciones para Windows form:

1. Se puede agregar productos al carrito de compras haciendo doble clic sobre la fila del producto deseado o haciendo un único clic y luego seleccionando la cantidad en el box inferior, posteriormente se puede presionar el botón para agregar el producto.
2. Haciendo doble clic sobre un producto del carrito de compras se elimina de dicho carrito, también se puede limpiar todo el carrito con el botón superior correspondiente.
3. Para agregar un producto se utiliza el botón en el menú ToolStrip.