TRABAJO PRÁCTICO FINAL

Este programa consiste en poder realizar la carga de productos correspondientes a un kiosko y registrar las ventas de los mismos al público en general.
Se puede dar de alta, baja y modificar el precio y cantidad en stock de todos los productos, así como también consultar cuántos productos le corresponden a cada venta realizada.

Temas

Excepciones

Se crearon excepciones personalizadas para los distintos casos de error posibles, mayormente en la carga de datos de un nuevo producto, como así también en la carga de datos a través de archivos o la obtención de productos a través de su DataGrid correspondiente

Pruebas Unitarias

Se testea la creación exitosa de un nuevo producto, la creación fallida de una nueva venta y la escritura exitosa de un archivo de tipo json.

Tipos Genéricos

La clase Archivo, que se encarga de leer y escribir distintos formatos de archivos, es genérica, pudiendo trabajar con datos de tipo Producto o Venta según corresponda. También en la clase Lógica dentro del proyecto de formularios existe un método genérico que se encarga de actualizar el DataGrid que corresponda con una lista de datos de tipo genérico.

Interfaces

Se encuentra definida la interfaz IInformacion que implementan tanto la clase Producto como Venta, la cual obliga a implementar el método DarInfo que se encarga de devolver toda la información necesaria de la instancia del objeto que la esté invocando.

Archivos y Serialización

Se utiliza la lectura y escritura de archivos de texto para llevar un registro fehaciente del ID de los productos y las ventas a medida que van siendo creados, sin repetirse.
El programa cuenta con la opción de guardar los cambios realizados, persistiendo todas las modificaciones realizadas al stock de productos general y a las ventas realizadas. En caso de no contar con los archivos correspondientes de carga de información, realiza una carga forzada de algunos productos a modo de prueba, sin ventas realizadas. Para variar el tipo de archivo utilizado, se guardan los productos en archivos de formato json y las ventas en formato XML.

Parte TP4

Introducción a SQL, Conexión a Base de Datos

Se creó una clase “BaseDatos” en la biblioteca de Entidades donde se maneja la conexión y se encuentran las distintas funciones correspondientes al ABM de Productos y Ventas.

Expresiones lambda

En la clase Producto, se reemplaza método "ExisteProductoEnCarrito" por expresión lambda en método "AgregarProductoACarrito", dejando su funcionalidad sin cambios. También se utilizan en el método Load del FormPrincipal, y en el método “HabilitarControles” también del FormPrincipal.

Delegados

Se crean 2 delegados en el archivo Delegados, en la biblioteca de Entidades. Ambos retornan void y reciben un string, pero se crean por separado ya que uno(StockAcabado) se utiliza para crear un evento. El otro (EstadoConexionBBDD) se utiliza como tipo de parámetro en el método EvaluarConexionBBDD, el cual recibe 2 métodos que coinciden con la firma y según las condiciones se ejecutará uno u otro de los métodos pasados por parámetro.

Hilos

Se crea un hilo paralelo en el método Load del FormPrincipal. El mismo simula una carga de 3 segundos para buscar datos a la BBDD y habilita o no los controles, según corresponda. Si los controles son habilitados, un label a modo de informe mostrará que la conexión fue establecida de forma exitosa y se volverá invisible luego de 2 segundos.

Eventos

Se crea un evento de instancia “onStockAcabado” en Producto, el cual es del mismo tipo que un delegado creado exclusivamente (a fin de demostrar la utilización de este evento) para el mismo. Para invocar este evento se crea un método “RestarStock”, el cual es utilizado para todos los métodos que realizan esa operación. Si el stock de algún producto llega a 0, se dispara el evento que, en el FormPrincipal, suscriben al evento de la instancia el método “AvisarQueNoHayStock”, el cual le notifica al usuario que ese producto en particular ya no cuenta con más stock.

Métodos de extensión

Se crea una clase “ExtString” la cual contiene el método de extensión de String “FormatearDescripcionDeProducto”, que al dar de alta un producto es llamada y utilizada para que los productos tengan una descripción prolija.