

## **Trabajo Practico Nº4**

Alumno: HERRERA, Martín.

División: 2D.

### **Funcionalidad pretendida:**

El programa está enfocado en poder administrar productos de dos tipos distintos en una base de datos creada y generar ventas. Un tipo de producto es Software y el otro es Hardware, para insertarlos a la base de datos se creó un formulario principal donde se puede ver los datos cargados en la base de datos, refrescarlos constantemente, ya que se hace en otro hilo, y a su vez, abrir otro formulario que permitirá realizar dicha inserción en la base de datos. Con respecto a las ventas, estas se testean en el proyecto Test.

A continuación, se detallan los temas según cada clase:

#### Clase 15 - Excepciones:

Hay dos excepciones, creadas en el proyecto llamado Excepciones. Una es DatosErroneosException, es utilizada en las validaciones que se aplican en las propiedades de la creación del producto para validar que los datos sean correctos.

La otra es ArchivosException, es utilizada para manejar los posibles errores al trabajar con archivos.

#### Clase 16 - Test Unitarios:

Hay dos métodos de testeo, ubicados en el proyecto llamado TestUnitario. Uno de ellos verifica que se cargue una de las listas estáticas que se carga con los productos que trae de la base de datos.

Y el otro método testea que se lance la excepción DatosErroneosException al crear un producto de Software.

## Clase 17 - Tipos Genéricos:

Este tema se aplica en la interfaz genérica llamada ISerializar y en la clase Xml ubicadas en el proyecto llamado Archivos y también, en varios métodos a lo largo del programa. Como los siguientes:

Ubicado en la clase ManejoBaseDeDatos:

```
public static bool ActualizarProductoVendido<T>(T productoParaActualizar) where T : Producto
```

Ubicados en la clase Stock:

```
public static string ListarLosProductos<T>(List<T> auxLista) where T : Producto
```

```
public static int GetUltimoIdListaProductos<T>(List<T> auxLista) where T : Producto
```

```
public static int GetIndiceProducto<T>(T producto) where T : Producto
```

## Clase 18 - Interfaces:

Este tema es utilizado en el proyecto llamado Archivos, donde fue creada la interfaz IManejoArchivos la cual contiene el metodo GuardarEnArchivo y es usada por las clases Xml y Texto.

## Clase 19 - Archivos y serialización:

El tema de archivos es utilizado en el proyecto Archivos, donde se encuentra la clase texto que maneja el guardado de archivos de texto y la clase Xml que maneja el guardado y la lectura de archivos xml.

El tema de la serialización es aplicado cuando se deserealiza el archivo Clientes.xml para cargar la lista estática de clientes a través del método DeserealizarArchivoClientes ubicado en la clase Stock y lo mismo para las ventas, pero con el método DeserealizarArchivoVentas.

Y también es aplicado cuando se serializan las ventas, a través del método SerializarVentas ubicado en la clase Venta.

## Clase 21 y 22 - SQL y Bases de Datos:

Se tiene una base de datos que cuenta con dos tablas, una para productos de Software y otra para productos de Hardware. Es utilizada para administrar los productos.

A su vez, fue creada una clase estática llamada ManejoBaseDeDatos ubicada en el proyecto entidades, la cual tiene métodos para obtener los datos ubicados en las tablas, otros métodos para insertar datos en las tablas y un último método para actualizar los productos en la base de datos.

Recuerde verificar el `ConnectionString` ya que este programa fue realizado con SQL Express.

Se adjunta dentro de la solución, un archivo de texto con el script para crear la base.

#### Clase 23 - Hilos:

En el formulario llamado `FormPrincipal`, se encuentra declarado un hilo que se utiliza para ejecutar el método `refrescarDataGridViewProductos` cuando el usuario presiona el botón que dice “Comenzar a refrescar”.

En dicho botón se lanza el hilo nuevo con el método para refrescar los `DataGridView`.

Esto permite seguir refrescando la información mostrada mientras que se ejecuta la carga de productos a la base de datos.

#### Clase 24 - Eventos:

Hay dos eventos creados, uno ubicado en el formulario llamado `FormInsertarHardware` y otro en el formulario `FormInsertarSoftware`, ubicados en el proyecto `Formularios`.

Estos dos eventos son del tipo de delegado llamado “Insert”, declarado en el formulario `FormPrincipal` y se utilizan para ejecutar la carga del producto nuevo a la base de datos y luego de esto, la ejecución del método que agrega a el archivo “`LogProductosInsertados.txt`” el producto cargado.

A continuación, se muestra cómo se declaró el delegado:

```
public delegate bool Insert(Producto auxProd);
```

Este delegado admite solamente métodos que devuelvan un `bool` y que reciban un objeto del tipo `Producto` o de los que heredan de `Producto`.

#### Clase 25 – Métodos de extensión:

Este tema es aplicado en la clase `ExtensionList` ubicada en el proyecto `Entidades`.

Se genero el siguiente método, que permite obtener la cantidad total de stock disponible de la lista de productos que le llegue:

```
public static int GetCantidadStockTotal<T>(this List<T> listaPedidos) where T : Producto
```

