Varela Vargas Leandro Gastón 2°A

Sistema de Venta

- ## Funcionalidad pretendida:

  Este proyecto es un sistema de ventas de un negocio donde se comercia con productos de Comida y Limpieza. El mismo permitirá crear lista de Productos y Personas (que podrán ser ‘Cliente’ o ‘Vendedor’).
  El proyecto cuenta con un formulario con una Barra de Herramientas superior y un menú que permitirá crear una factura, agregando Cliente, Vendedor y productos a la misma, que podrá Serializar como Xml y al mismo tiempo “imprimir” en txt con el botón confirmar.
  En la Barra de Herramientas habrá botones que permitirán abrir ventanas que, mediante el uso de DataTables, darán la posibilidad de agregar y/o alterar datos de la Base de Datos.

- ## Temas:

Clase 15 (Exceptions):

```
    /// <summary>
    /// Clase 15. Exceptions.
    /// Esta exception se lanzará en caso de que los datos de la base de Datos de Personas
no puedan ser obtenidos.
    /// </summary>
    public class ExceptionErrorActualizacionPersonas : Exception

    /// <summary>
    /// Clase 15. Exceptions.
    /// Esta exception se lanzará en caso de que los datos de la base de Datos de los
Productos no puedan ser obtenidos.
    /// </summary>
    public class ExceptionErrorActualizacionProductos : Exception
```

Clase 16 (Test Unitarios): Proyecto TestUnitarios.

Clase 17 (Tipos Genéricos):

```
    /// <summary>
    /// Clase 21 y 22. SQL
    /// Clase 17. Generics.
    /// Clase 18. Interfaces.
    /// En esta clase se almacenará una lista de Productos.
    /// </summary>
    /// <typeparam name="T"></typeparam>
    [Serializable]
    public class Productos<T> : IEnBD where T : Producto

    /// <summary>
    /// Clase 17. Generics.
    /// Agrega un producto.
    /// </summary>
    /// <param name="ps"></param>
    /// <param name="p"></param>
    /// <returns>'True' si la operación es exitosa. 'False' caso contrario.</returns>
    public static bool operator +(Productos<T> ps, T p)
```

Varela Vargas Leandro Gastón 2°A

```
/// <summary>
/// Clase 17. Generics
/// Elimina un producto.
/// </summary>
/// <param name="ps"></param>
/// <param name="p"></param>
/// <returns>'True' si la operación es exitosa. 'False' caso contrario.</returns>
public static bool operator -(Productos<T>ps, T p)
```

Clase 18 (Interfaces):

```
/// <summary>
/// Clase 18 Interfaces.
/// </summary>
interface IEnBD

/// <summary>
/// Clase 21 y 22. SQL
/// Clase 17. Generics.
/// Clase 18. Interfaces.
/// En esta clase se almacenará una lista de Productos.
/// </summary>
/// <typeparam name="T"></typeparam>
[Serializable]
public class Productos<T> : IEnBD where T : Producto

/// <summary>
/// Clase 21 y 22. SQL Y BD.
/// Clase 18. Interfaces.
/// Obtiene la lista de productos en Stock de la tabla.
/// </summary>
/// <param name="NombreTabla"></param>
public void actualizarMedianteBD(string NombreTabla)

/// <summary>
/// Clase 21 y 22. SQL.
/// Clase 18 Interfaces.
/// Esta clase contendrá una lista de personas y podrá obtener datos de la BD.
/// </summary>
public class Personas : IEnBD

/// <summary>
/// Clase 21 y 22. SQL Y BD.
/// Clase 18. Interfaces.
/// Obtiene todas las personas que se encuentren en la Tabla especificada.
/// </summary>
/// <param name="NombreTabla"></param>
public void actualizarMedianteBD(string NombreTabla)
```

Clase 19 (Archivos y Serialización):

```
/// <summary>
/// Clase 23. Hilos.
/// Clase 19. Archivos y Serialización.
/// Esta clase contendrá toda la información para la generación de una factura.
/// Permitirá Serialización y creación de archivos de texto.
/// </summary>
[Serializable]
public class Factura
```

Varela Vargas Leandro Gastón 2°A

```
 /// <summary>
 /// Clase 17. Archivos y Serialización.
/// Guardará la factura en la carpeta "Documentos" del usuario, cuyo nombre será "Factura"
seguido de la fecha de Emisión de la factura
/// </summary>
private void privadoGuardarComoTxt()

/// <summary>
/// Clase 17. Archivos y Serialización.
/// Guardará la factura en la carpeta "Documentos" del usuario, cuyo nombre será "Factura"
seguido de la fecha de Emisión de la factura en formato Xml
/// </summary>
private void privadoGuardarComoXml()
```

Clase 21 y 22 (SQL, Bases de Datos, DataTable y DataAdapter):

```
/// <summary>
/// Clase 24. Eventos.
/// Clase 21 y 22. SQL, BD, DataTable y DataAdapter.
/// </summary>
public partial class FormSQL : Form

/// <summary>
/// Clase 21 y 22. SQL y BD.
/// Inicializa la conexión con la base de datos
/// </summary>
/// <param name="sender"></param>
/// <param name="args"></param>
private void inicializarConnection(object sender, EventArgs args)

/// <summary>
/// Clase 21 y 22. DataTable.
/// Inicializa el DataTable con los nombres de las columnas que corresponden para el Stock
/// </summary>
/// <param name="sender"></param>
/// <param name="args"></param>
private void inicializarDataTableStock(object sender, EventArgs args)

/// <summary>
/// Clase 21 y 22. SQL, Bases de Datos y DataAdapter.
/// Inicializa el DataAdapter con los comandos necesarios para la tabla Productos
/// </summary>
/// <param name="sender"></param>
/// <param name="args"></param>
private void inicializarDataAdapterStock(object sender, EventArgs args)

/// <summary>
/// Clase 21 y 22. DataTable.
/// Inicializa el DataTable con los nombres de las columnas que corresponden para Clientes o
Vendedores
/// </summary>
/// <param name="sender"></param>
/// <param name="args"></param>
private void inicializarDataTablePersonas(object sender, EventArgs args)

/// <summary>
/// Clase 21 y 22. SQL, Bases de Datos y DataAdapter.
/// Inicializa el DataAdapter con los comandos necesarios para la tabla Clientes o
Vendedores según corresponda
/// </summary>
```

Varela Vargas Leandro Gastón 2°A

```
/// <param name="sender"></param>
/// <param name="args"></param>
private void inicializarDataAdapterPersonas(object sender, EventArgs args)

/// <summary>
/// Clase 21 y 22. SQL
/// Clase 17. Generics.
/// Clase 18. Interfaces.
/// En esta clase se almacenará una lista de Productos.
/// </summary>
/// <typeparam name="T"></typeparam>
[Serializable]
public class Productos<T> : IEnBD where T : Producto

/// <summary>
/// Clase 21 y 22. SQL Y BD.
/// Clase 18. Interfaces.
/// Obtiene la lista de productos en Stock de la tabla.
/// </summary>
/// <param name="NombreTabla"></param>
public void actualizarMedianteBD(string NombreTabla)

/// <summary>
/// Clase 21 y 22. SQL.
/// Clase 18 Interfaces.
/// Esta clase contendrá una lista de personas y podrá obtener datos de la BD.
/// </summary>
public class Personas : IEnBD

/// <summary>
/// Clase 21 y 22. SQL Y BD.
/// Clase 18. Interfaces.
/// Obtiene todas las personas que se encuentren en la Tabla especificada.
/// </summary>
/// <param name="NombreTabla"></param>
public void actualizarMedianteBD(string NombreTabla)
```

Clase 23 (Hilos):

```
/// <summary>
/// Clase 23. Hilos.
/// Clase 19. Archivos y Serialización.
/// Esta clase contendrá toda la información para la generación de una factura.
/// Permitirá Serialización y creación de archivos de texto.
/// </summary>
[Serializable]
public class Factura

/// <summary>
/// Clase 23. Hilos.
/// Invoca un hilo que guardará la factura como un Txt.
/// </summary>
public void GuardarComoTxt()

/// <summary>
/// Clase 17. Archivos y Serialización.
/// Guardará la factura en la carpeta "Documentos" del usuario, cuyo nombre será "Factura"
seguido de la fecha de Emisión de la factura en formato Xml
/// </summary>
private void privadoGuardarComoXml()
```

Varela Vargas Leandro Gastón 2°A

Clase 24 (Eventos):

En el proyecto se utilizan todo tipo de eventos (de DataTable, de Forms, etc.), pero aquí solo muestro un evento creado por mí y las funciones asignadas a él.

```
/// <summary>
/// Clase 24. Eventos.
/// </summary>
public partial class FrmSistemaDeVenta : Form

/// <summary>
/// Clase 24. Eventos.
/// Firma con la que deberán cumplir todos los métodos que signifiquen algún tipo de
actualización
/// de los controles del formulario.
/// </summary>
private delegate void Actualizaciones();

/// <summary>
/// Clase 24. Eventos.
/// Evento de actualización de controles del formulario.
/// </summary>
private event Actualizaciones actualizaciones;

/// <summary>
/// Clase 24. Eventos.
/// Permite que se invoquen actualizaciones desde otra parte.
/// </summary>
public void invocarActualizaciones()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará las variables de los productos con la Base de Datos.
/// </summary>
private void actualizarProductos()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará la variable "clientes" con la Base de Datos.
/// </summary>
private void actualizarClientes()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará la variables "vendedores" con la Base de Datos.
/// </summary>
private void actualizarVendedores()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará los ComboBox del Form con los respectivos datos correspondientes.
/// </summary>
private void actualizarCombos ()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará el "textBoxFecha" con la Fecha y Hora actuales.
/// </summary>
private void actualizarFecha ()
```

Varela Vargas Leandro Gastón 2°A

```
/// <summary>
/// Clase 24. Eventos.
/// Actualizará el "textBoxTotal" a 0.
/// </summary>
private void actualizarPrecio()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará la lista de productos seleccionados.
/// </summary>
private void actualizarProductoSeleccionados()

/// <summary>
/// Clase 24. Eventos.
/// Actualizará la tabla a su valor vacío por defecto.
/// </summary>
private void actualizarTabla()
```

Clase 25 (Métodos de extensión):

```
/// <summary>
/// clase 25. Métodos de Extensión.
/// Esta clase agrega un método a la clase Factura.
/// </summary>
public static class FacturaExtension

/// <summary>
/// Clase 25. Métodos de Extensión.
/// Realiza un string customizado a partir de un objeto Factura.
/// </summary>
/// <param name="f">objeto Factura</param>
/// <returns>string customizado</returns>
public static string CustomPrintFactura (this Factura f)
```