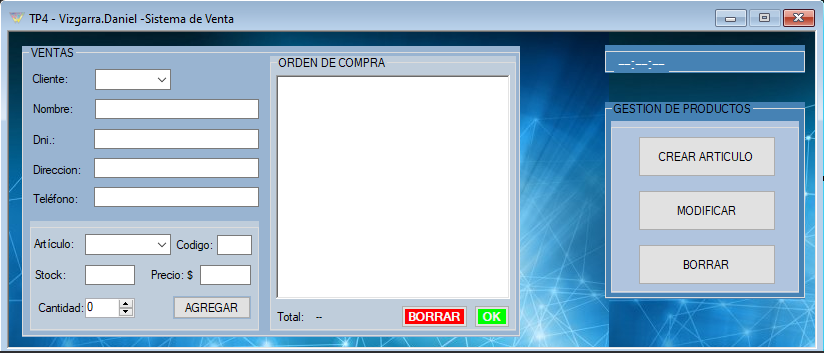
**TRABAJO PRACTICO N°4 – Vizgarra Livón Daniel S.**



**FUNCIONAMIENTO:**

El sistema inicia con un menú principal *(FrmMenuPrincipal)* desde donde se elige que hacer.

Este formulario va instanciar un objeto de tipo *Negocio*, este va a cargar los datos Clientes y Productos en sus listas desde la base de datos.

**FrmMenuPrincipal:**

* **Menú desplegable Cliente:** Permite elegir un cliente de la base de datos.
* **Menú desplegable Artículo**: Permite elegir un producto de la base de datos.

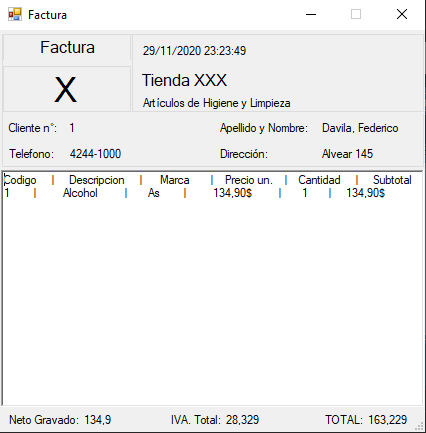
* **Botón “Agregar”:** Va a agregar una Venta (producto, cliente, cantidad) a la lista de ventas y cada artículo agregado se verá reflejado en la Orden de Compra.
* **Botón “Borrar”:** Anula la Orden de compra
* **Botón “Ok”:** Va generara una Factura con los datos de la Orden de compra. La mostrará mediante el Formulario *FrmFactura*.

**Menú “Gestión de productos”:** Modulo ABM del sistema.

* **Botón “Crear Artículo”:** Permite crear un artículo ingresando sus datos en los *Textbox* y se guardará en la Base de Datos.
* **Botón “Modificar”:** Permite modificar un artículo ya existente (seleccionado del “Menú desplegable Artículo”) y se guardará en la Base de Datos.
* **Botón “Borrar”:** Permite borrar un artículo ya existente (seleccionado del “Menú desplegable Artículo”) y se guardará en la Base de Datos.

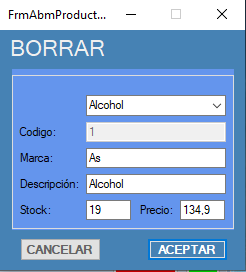
**FrmFactura:**

Recibe los datos del cliente y de la venta realizada, lista de productos, cantidades, Importe neto gravado, IVA total y Total a abonar, y lo muestra por pantalla.



**FrmAbmProductos:**

Las 3 opciones utilizan este mismo form, *FrmAbmProductos*. Para ello se emplea un enumerado que representa cada tipo de operación y mediante un *switch* modera como operar. Para “Creación” no va mostrar el *comboBox* de selección de artículo ya q no es necesario, mientras que para las otras dos opciones, sí. Así mismo, los *textBox* en la opción “Creación” aparecen vacíos.



* **Botón “Cancelar”:** Cierra el formulario y vuelve al menú principal.
* **Botón “Aceptar”:** Nuevamente se emplea el “mode” o enumerado *EAccionABM* para moderar su funcionamiento. Dependiendo del caso, realizará una operación de tipo *Insert, Update* o *Delete* en la Base de Datos, para luego actualizar las listas en el sistema.

**Temario:**

**Excepciones**: Proyecto que contiene las distintas excepciones para manejar el proyecto.

**Test Unitarios**: Proyecto *Unitest* que pruebas funcionalidades pequeñas del sistema.

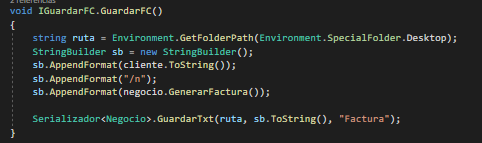
**Base de Datos**: Proyecto Persistencia de datos.

* **DatabaseConnection**: posee el método estático de orden try/catch/finally para abrir la DB, ejecutar una query y cerrar la conexión
* ***ProductosDB***: Posee 4 métodos estáticos *Insert,Update, Delete y SelectAll* para cargar los datos al sistema y realizar operaciones según su moderador
* ***ClientesDB***: Posee un método estático para cargar la lista de Clientes al sistema
* ***Moderador***: Posee un método estático para moderar según el enumerado *EAccionABM* que tipo acción realizar.

**Interfaces:** Esta declarada en la clase *IGuardarFC* del proyecto *Comun*.

Luego en el formulario *frmFactura*, se implementa la interface de forma explícita para realizar el guardado del extracto.

****

****

**Hilos:** En el formulario *frmFactura*, se utiliza un nuevo hilo para ejecutar un el método *GuardarFC*(). **Aclaración:** No finalizo de forma manual el hilo, ya que este termina al guardar el .txt de forma natural.

****

**Eventos y delegados:** Mediante el *DelegadoModerador* y su evento correspondiente se realizan las tareas de modificación ABM de la BD.



**Métodos de Extensión**: Se calcula el monto final de una venta y luego se retorna un string formateado con la siguiente sintaxis. "{0:0.00}$".

**Serialización:** Serializador (dentro del proyecto “Persistencia de datos”) posee métodos de escritura y lectura.xml para serializar un objeto y guardarlo en la ruta que se le especifique. También en formato posee métodos de escritura y lectura .txt para guardar un texto en la ruta que se le especifique. (Lo aplico en la consola)

**Generics:** Utilizado para especificar en la clase Serializador que objeto <T> se puede serializar o deserializar, siguiendo la restricción:

