

## **Trabajo Practico n 3**

### **Franco Buonfrate**

#### **Descripción**

Mi programa está enfocado en facilitar la venta de artículos en un local de pintura. Esta este dirigido directamente para el vendedor el cual manejara el 100% del programa, teniendo “del otro lado del mostrador” al cliente.

#### **Funcionamiento**

La aplicación consta de 6 forms los cuales se generarán y cerrarán a medida que avance el programa.

En primera instancia se mostrará el form de clientes, el cual posee un atributo estático List<Cliente<IProductos>> en el que se irán almacenando todos los clientes. Esta lista será cargada por medio del constructor estático con los datos almacenados en el archivo “DatosClientes.xml” ubicado en la siguiente ruta:

Buonfrate.Franco.2A.TP3\Pintureria\bin\Debug\net5.0-windows\Datos\

Una vez cargados los datos a la lista se cargarán al DataGridView mediante el método DibujarTabla en el constructor por defecto de la clase.

En este form se puede:

- Dar de alta un nuevo cliente, completando todos los campos y presionando el botón “crear”. Internamente este botón crea una nueva instancia de cliente<IProductos> con los parámetros previamente obtenidos de los campos, lo agrega a la lista y finalmente actualiza el archivo .xml.

- Modificar un usuario. Al clickear sobre cualquiera de los campos del DataGridView del usuario a modificar se mostrarán los datos del mismo (gracias al método dataGridView1\_CellClick). Luego de modificar el campo deseado se oprime el botón “Modificar” e internamente este

buscara el usuario modificado, mediante su Id única (gracias a la sobrecarga del método Equals) y se sobrecarga con los nuevos datos.

-Eliminar Usuario. Se selecciona el usuario y se presiona el botón “Eliminar”, el programa buscará nuevamente el usuario por su id y no elimina sus datos simplemente cambia su propiedad de clienteActivo a “false” y de esta forma no se cargará más al Datagrid.

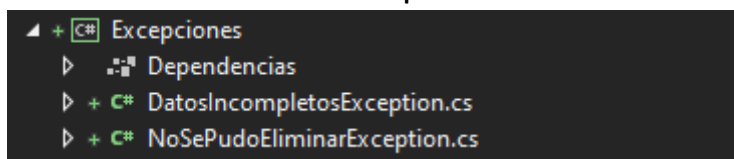
Para pasar al siguiente form se deberá seleccionar un cliente y presionar el botón "Productos", este generara un form Productos pasandole el cliente seleccionado donde se cargarán los productos agregados en su campo listaProductos.

El form Productos consta de 3 botones, uno para cada tipo de producto. Cada botón abrirá un form distinto mediante el cual se podrán agregar los distintos productos a la compra. Finalmente se pasa al form de Factura, en el que se podrá generar un ticket que se guardará en el escritorio con los datos de la facturación en una nueva carpeta llamada “Facturas”.

## Temario

### *-Excepciones*

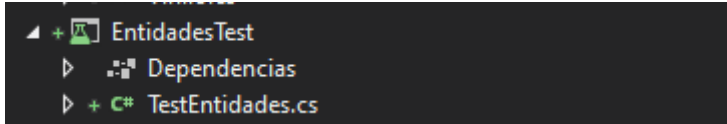
Biblioteca de clase: Excepciones



Se utilizan en el form Clientes.

### *-Pruebas Unitarias*

## Proyecto de pruebas: EntidadesTest



Se prueba la funcionalidad de Entidades

### -Tipos Genéricos

Biblioteca de clase: Entidades

Clase: Cliente<T>

```
45 referencias
public class Cliente<T> : Persona where T : IProductos
{
    [XmlIgnore]
    private List<T> listaDeCompras;
```

Se utiliza a lo largo del código

### -Interfaces

Biblioteca de Clases: Entidades

Interfaz: IProductos



Se utiliza a lo largo del código

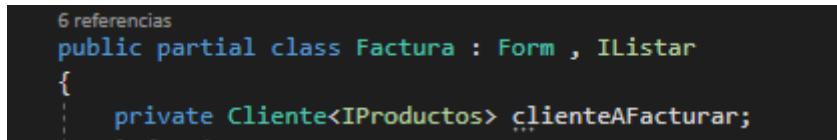
```
45 referencias
public class Cliente<T> : Persona where T : IProductos
{
    [XmlIgnore]
    private List<T> listaDeCompras;
```

Form Program: Pinturería

## Interfaz: IListar

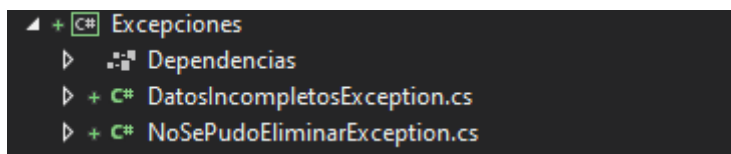


Se utiliza en los forms Cllientes y Factura



## -Archivos

### Biblioteca de clases: ManejoDeDatos



Se utiliza para crear archivos de texto en el form Factura y Para serializar los datos de los clientes en el form Clientes

