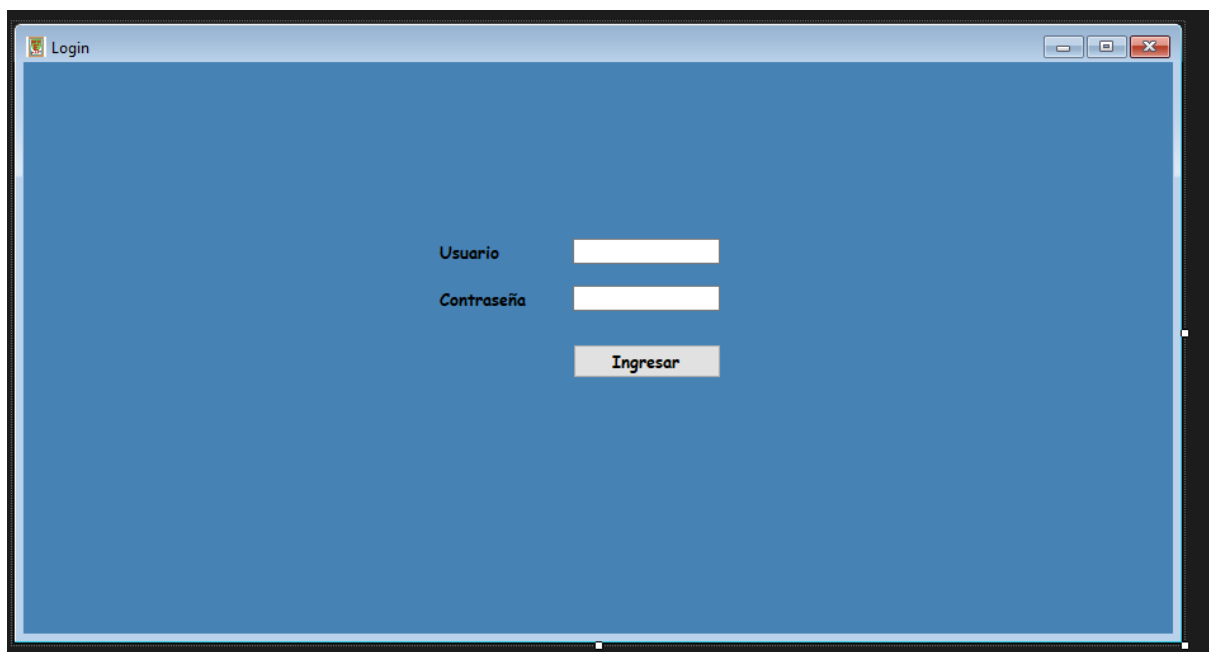


El Sistema propuesto para este trabajo, consiste en la administración de la venta de productos de un mercado que se dedica solo a la venta de celulares y teclados. El mismo sirve para administrar stock de productos a la venta, realizar las ventas, y una simulación del registro de Empleados y Clientes, donde se puede agregar, editar y eliminar clientes/empleados.

El programa está realizado en el lenguaje C#, con conexión a SQL Server.

En cuanto a la arquitectura del proyecto, el camino es el siguiente:

- GUI: Capa Formularios
- Entidades que contienen sobrecargas, propiedades
- Capa de Negocio: Que cumple la función de ser la intersección entre la capa de datos y la capa GUI, para evitar la desprolijidad de utilizar sql directo desde el formulario. Esta capa también contiene métodos específicos de la lógica del negocio.
- Base de Datos: Contiene todas las conexiones SQL del programa.
- Excepciones: Contiene las excepciones de acuerdo a la lógica del programa que fueron requeridas para un código limpio y bien fundamentado.
- Validaciones: Realiza validaciones que utilizan las excepciones de mencionada capa
- Interfaces: Contienen Guardar y Leer, que prosiguen a utilizarse para serializar
- Serialización: Realiza informes y resultados a .xml y .txt



El programa tiene un sistema de login que se encarga de validar que el nombre de usuario no pueda tener espacios y la contraseña debe ser solamente numérica. Para dicho fin, se realizaron excepciones para su control, que se encuentran en el proyecto Validaciones.

```

public partial class FrmLogin : Form
{
    Dictionary<string, string> diccionario = new Dictionary<string, string>();
    1 referencia
    public FrmLogin()
    {
        InitializeComponent();
    }
    1 referencia
    private void btnLogin_Click(object sender, EventArgs e)
    {
        string usuario = txtUsuario.Text;
        string contraseña = txtContraseña.Text;

        string validacionUsuario = Validar.ValidarUsuario(usuario);
        string validacionContraseña = Validar.ValidarNumeroDeEntrada(contraseña);

        if (validacionUsuario == "Vacio")
        {
            MessageBox.Show("El usuario esta vacio. Reingrese usuario");
        }

        if (validacionUsuario == "Espacios")
        {
            MessageBox.Show("El usuario contiene espacios. Reingrese usuario");
        }

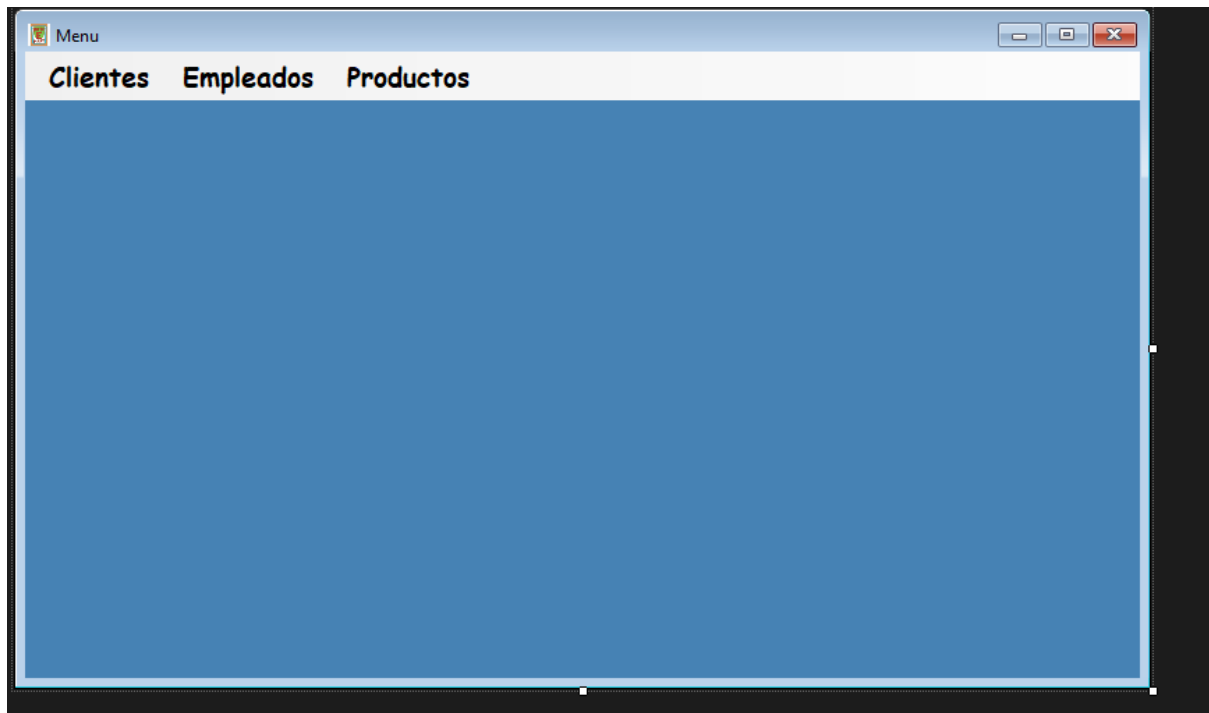
        if (validacionContraseña == "Vacía")
        {
            MessageBox.Show("La contraseña esta vacía. Reingrese contraseña");
        }

        if (validacionContraseña == "Numero erroneo")
        {
            MessageBox.Show("La contraseña debe tener solo numeros");
        }

        if (validacionUsuario == "Correcto" && validacionContraseña == "Correcta")
        {
            diccionario.Add(usuario, contraseña);
            MessageBox.Show("Bienvenido/a !!!");
        }
    }
}

```

Para el trabajo, se realizó un menú a través de un menustrip, el cual redirige a la opción que elija el usuario. Por defecto todo se encuentra cargado desde la base de datos, por lo tanto si el usuario ingresa a un formulario y realiza cambios en la información, esta se observa modificada en los demás formularios, siendo de esta manera sincrónica todo el sistema.



Para realizar la Modificación del formulario de Clientes, se utilizó una columna con checkbox, los cuales permiten al usuario seleccionar el cliente que desean modificar, a través de darle un click a la grilla inferior y los valores seleccionados se cargan en los textbox superiores para proseguir a ser editados. Para que las modificaciones sean válidas y se vean reflejadas en la base de datos, es necesario que el usuario presione el botón modificar.

A screenshot of the "Clientes" form window. The form includes five input fields: "Nombre", "Apellido", "Edad", "DNI", and "ID Cliente". To the right of these fields are three buttons: "Agregar Cliente", "Eliminar Cliente", and "Modificar Cliente". In the top right corner, there is an "Atras" button. Below the form, there is a section titled "Lista de Clientes" which contains a table with a "Seleccionar" column. The table has one row with an asterisk (*) in the first column and a checkbox in the "Seleccionar" column. The "Modificar Cliente" button is highlighted with a dashed border, indicating it is the active or selected option.

Si el usuario prosigue a seleccionar varios registros, se restringe la opción de modificar de a bloques, por lo cual se deshabilita el botón modificar.

En cuanto al botón de Eliminar, se puede eliminar de a bloques o de a uno.

El formulario clientes, cumple la misma función que el formulario Empleados, con la misma lógica del negocio customizada.

The screenshot shows a window titled 'Clientes'. It contains a form with five input fields: 'Nombre', 'Apellido', 'Edad', 'DNI', and 'ID Cliente'. To the right of these fields are three buttons: 'Agregar Cliente' (next to Nombre and Apellido), 'Eliminar Cliente' (next to Edad and DNI), and 'Modificar Cliente' (next to ID Cliente). In the top right corner of the window is an 'Atras' button. Below the form, the text 'Lista de Clientes' is displayed in bold. Underneath this text is a table with two columns: 'Seleccionar' and an empty column. The table contains one row with an asterisk (*) in the first column and a checkbox in the second column.

En cuanto al formulario de Productos, que es la base y el enfoque que se le dio al proyecto, tiene combos que traen los datos para el vendedor y cliente que ÚNICAMENTE se utilizaran para la serialización que se prosigue a explicar.

The screenshot shows a window titled 'Productos'. It features a form with several input fields and buttons. At the top left, there are two dropdown menus labeled 'Cliente' and 'Vendedor', which are circled in red. To their right are buttons for 'Agregar', 'Eliminar', and 'Modificar'. Below these are checkboxes for 'Tilde si es celular'. Further down are input fields for 'Nombre', 'Precio', 'Marca', 'Cantidad', 'Pantalla', 'Microprocesador', 'ID Producto', and 'Tipo'. At the bottom left, there are buttons for 'Ver Celulares' and 'Ver Teclados', and labels for 'Cantidad Total Stock' and 'Total Stock'. In the bottom right, there is a 'CARGAR' button. A table with two columns, 'Seleccionar' and an empty column, is located in the bottom center, containing one row with an asterisk (*) and a checkbox. An 'Atras' button is in the top right corner.

Dentro de la interfaz que utiliza la serialización, utilizo generics para poder serializar cualquier elemento que me sea brindado.

```
SerializacionTXT.cs  FrmVentas.cs [Diseño]*  Validar.cs  IArchivo.cs
C# Interfaces
1 namespace Interfaces
2 {
3     2 referencias
4     public interface IArchivo<T>
5     {
6         25 referencias
7         bool Guardar(string path, string archivo, T datos);
8         2 referencias
9         bool Leer(string archivo, out T datos);
10    }
11 }
```

Seguindo la lógica del programa, el usuario puede elegir dos productos, si tilda en el checkbox de celular, se le habilita para completar dos campos específicos para el producto mencionado.

Productos

Cliente [dropdown] [dropdown]

Vendedor [dropdown] [dropdown]

☐ Tilde si es celular

Nombre [input] Marca [input]

Precio [input] Cantidad [input]

Pantalla [input]

Microprocesador [input]

Agregar

Ver Celulares

Seleccionar
<input type="checkbox"/>

Cuando el usuario presiona agregar, se prosigue a actualizar la grilla con los productos que carga.

Productos

Cliente

Vendedor

☐ Tilde si es celular

Nombre Marca ID Producto

Precio Cantidad Tipo

Pantalla

Microprocesador

Ver Celulares

Ver Teclados

Cantidad Total Stock

Total Stock

Agregar Eliminar Modificar

CARGAR

Seleccionar
* <input type="checkbox"/>

En cuanto a la grilla, tenemos dos botones, el botón ver celulares y ver productos, ya que al utilizar dos productos diferentes con diferentes campos, se optó por la solución de cargar el datasource de acuerdo al click del usuario.

Para agregar un producto, al richtextbox y proseguir a comprar luego, se debe seleccionar el tipo a elegir, por ejemplo celulares y luego presionar el botón cargar para que se cargue a mencionado elemento. Lo mismo con teclado, no está contemplado que se puedan seleccionar de los dos datasource productos en simultaneo, es requisito que el usuario presione el botón cargar luego de elegir cada serie de productos.

Ver Celulares

Ver Teclados

0

Total Stock

Seleccionar	Tipo	Nombre	Precio
<input checked="" type="checkbox"/>	uytu	S10	80000
<input checked="" type="checkbox"/>	5435	S10	80000
<input type="checkbox"/>	66	66	66
<input type="checkbox"/>	77	77	77
<input type="checkbox"/>	jldadjslaj		59
<input type="checkbox"/>	eterter	ytutyuu	5
<input type="checkbox"/>	Mecanico	Teclado MAster	677
<input type="checkbox"/>	mecanico	Teclado 1	500
<input type="checkbox"/>	hgigh	ytutyuu	5

S10 80000 1 1 Samsung uytu
S10 80000 1 345 Samsung 5435

CARGAR

Una vez elegida las compras y cargadas al richtextbox. Se prosigue a presionar el boton COMPRAR que muestra un mensaje en pantalla de la venta realizada y crea un archivo .txt en la ruta C:\Users\...\Documents\Iacobellis.Lucas.2D.TP04\Ventas por cada venta realizada.

ndedor: GUadalupe

e si es celular

Nombre:

Marca:

ID Producto:

Prezio:

Cantidad:

Tipo:

Agregar **Eliminar** **Modificar**

S10 80000 1 1 Samsung uytu
S10 80000 1 345 Samsung 5435

Celulares

Teclados

Stock

Seleccionar	Tipo	Nombre
<input checked="" type="checkbox"/>	uytu	S10
<input checked="" type="checkbox"/>	5435	S10
<input type="checkbox"/>	66	66
<input type="checkbox"/>	77	77
<input type="checkbox"/>	jldadjslaj	
<input type="checkbox"/>	eterter	ytutyuu
<input type="checkbox"/>	Mecanico	Teclado MAster
<input type="checkbox"/>	mecanico	Teclado 1
<input type="checkbox"/>	hgigh	ytutyuu

COMPRAR

Producto: S10
Precio: \$80000
Cantidad: 1
Producto: S10
Precio: \$80000
Cantidad: 1
Cantidad total: 2
Precio total: \$160000

Aceptar