

```

using BussinessLayer;
using Entities;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Gym
{
    public partial class Clientes : Form
    {
        #region Instancias
        //Entidades
        private readonly Entities.Clientes _clientes;
        private readonly Personas _personas;

        //Clases internas
        private readonly Restricciones _restricciones;
        private readonly MetodosGenerales _metodosGenerales;

        //Capa de negocio
        private readonly BussinessClientes _bussinessClientes;
        private readonly BussinessPersonas _bussinessPersonas;

        #endregion

        #region Variables
        private int personaLogueada;

        private bool contenidoErroneo;
        private bool camposObligatoriosVacios;
        //Verificar coincidencias *****
        private bool personaRegistrada;
        private DataSet dsTablaClientes;
        private string buscar;
        private int motivoMenu = 5;
        private bool edicionCliente = false;

        #endregion

        #region Load del formulario
        public Clientes(int idPersonaLog)
        {
            InitializeComponent();
            personaLogueada = idPersonaLog;
            _clientes = new Entities.Clientes();
            _restricciones = new Restricciones();
            _metodosGenerales = new MetodosGenerales();
            _bussinessClientes = new BussinessClientes();
            _bussinessPersonas = new BussinessPersonas();
            _personas = new Personas();
        }
        private void Clientes_Load(object sender, EventArgs e)
        {
            Tipos_Documentos();
            Tipos_Sexos();

```

```

        GetClientes();
    }

#endregion

#region Métodos Encapsulados

private void GetClientes()
{
    //Primero limpio el grid. Para que no haya errores.
    //Sería como si hiciera un refresh.
    //Saco lo que haya y luego vuelvo a traer los datos
    dtgvCliente.Rows.Clear();

    //Acá traigo los datos y se los asigno a la tabla dsTablaClientes
    dsTablaClientes = _bussinessClientes.GetClientes(buscar);
    //Vacío el textbox para resetear en caso de búsqueda específica
    buscar = string.Empty;

    //Y acá traigo los datos
    //Pregunto si el dataset tiene datos, preguntando si tiene filas
    if (dsTablaClientes.Tables[0].Rows.Count > 0)
    {
        //Y por cada fila que haya en el dataset
        foreach (DataRow dr in dsTablaClientes.Tables[0].Rows)
        {
            if (dr[6].ToString() == "A")
            {
                //paso esa fila al datagrid para que se vean los datos
                dtgvCliente.Rows.Add(dr[0].ToString(), dr[1], dr[2], dr[3], dr[4],
dr[5], "Activo");
            }
            else
            {
                dtgvCliente.Rows.Add(dr[0].ToString(), dr[1], dr[2], dr[3], dr[4],
dr[5], "Inactivo");
            }
        }
    }
}

private void Tipos_Documentos()
{
    //llamo al método para traer los tipos de documentos y los asigno al combobox
    _metodosGenerales.Bring_Tipos_Documentos();
    cmbTipoDocumentoCliente.DataSource = _metodosGenerales.DtTipos_Documentos;
    //lo que quiero ver
    cmbTipoDocumentoCliente.DisplayMember = "Tipo";
    //y les asigno el id correspondiente
    cmbTipoDocumentoCliente.ValueMember = "Tipo_documento_ID";
}

private void Tipos_Sexos()
{
    //llamo al método para traer los tipos de sexos y los asigno al combobox
    _metodosGenerales.Bring_Tipos_Sexos();
    cmbSexo.DataSource = _metodosGenerales.DtTipos_Sexos;
    cmbSexo.DisplayMember = "Sexo"; //Pero solo esta columna es la que muestro
    cmbSexo.ValueMember = "Tipo_Sexo_ID";
}

private void ABMCliente()

```

```
{
    if (motivoMenu == 3)
    {
        EditarPersona();
        EditarCliente();
    }
    else
    {
        if (motivoMenu == 5)
        {
            AltaPersona();
            AltaCliente();
        }
    }
}

private void EditarPersona()
{
    //Es la edición de los datos personales
    //junto con los datos de empleado, excepto las claves
    int Persona_ID = _personas.Persona_ID;
    string Nombre = Convert.ToString(txtNombreCliente.Text);
    string Apellido = Convert.ToString(txtApellidoCliente.Text);
    int Tipo_Documento_ID = cmbTipoDocumentoCliente.SelectedIndex;
    string Nro_documento = Convert.ToString(txtNumDocumentoCliente.Text);
    int Tipo_Sexo_ID = cmbSexo.SelectedIndex;
    string Nro_Telefono = Convert.ToString(txtTelefonoCliente.Text);
    string Nro_Alternativo, Mail, Observaciones;

    //Envío todo el objeto creado a la cada de negocios para ser enviado
    //a la bdd.

    //Acá verifico si el usuario ingresó datos nuevos o no a los campos
    //que reciben nulos.
    //Si no agregaron datos, los mandamos como nulos
    //caso contrario, se envía lo que haya en su correspondiente Textbox
    if (txtAlternativoCliente.Text != "Alternativo")
    {
        Nro_Alternativo = Convert.ToString(txtAlternativoCliente.Text);
    }
    else
    {
        Nro_Alternativo = string.Empty;
    }
    if (txtMailCliente.Text != "Mail")
    {
        Mail = Convert.ToString(txtMailCliente.Text);
    }
    else
    {
        Mail = string.Empty;
    }
    if (txtObservacionesCliente.Text != "Observaciones y/o consideraciones")
    {
        Observaciones = Convert.ToString(txtObservacionesCliente.Text);
    }
    else
    {
        Observaciones = string.Empty;
    }
    _metodosGenerales.EditarPersona(Persona_ID, Nombre,
```

```
        Apellido,
        Tipo_Documento_ID,
        Nro_documento,
        Tipo_Sexo_ID,
        Nro_Telefono,
        Nro_Alternativo,
        Mail,
        Observaciones);
    }

    private void EditarCliente()
    {
        if (cmbEstado.SelectedIndex == 0)
        {
            _clientes.Estado = "A";
        }
        else
        {
            _clientes.Estado = "I";
        }
        _bussinessClientes.EditarCliente(_clientes);
    }

    private void AltaPersona()
    {
        //En la clase "MetodosGenerales" tengo un método para enviar los datos de
        //una nueva persona a la entidad y luego a la base de datos a través
        //de la capa de negocio
        string Nombre = Convert.ToString(txtNombreCliente.Text);
        string Apellido = Convert.ToString(txtApellidoCliente.Text); ;
        int Tipo_Documento_ID = cmbTipoDocumentoCliente.SelectedIndex;
        string Nro_Documento = Convert.ToString(txtNumDocumentoCliente.Text);
        int Tipo_Sexo_ID = cmbSexo.SelectedIndex;
        string Nro_Telefono = Convert.ToString(txtTelefonoCliente.Text);
        string Nro_Alternativo;
        string Mail;
        string Observaciones;
        DateTime FechaAlta = DateTime.Now;

        if (txtAlternativoCliente.Text != "Alternativo")
        {
            Nro_Alternativo = Convert.ToString(txtAlternativoCliente.Text);
        }
        else
        {
            Nro_Alternativo = string.Empty;
        }

        if (txtMailCliente.Text != "Mail")
        {
            Mail = Convert.ToString(txtMailCliente.Text);
        }
        else
        {
            Mail = string.Empty;
        }

        if (txtObservacionesCliente.Text != "Observaciones y/o consideraciones")
        {
            Observaciones = Convert.ToString(txtObservacionesCliente.Text);
        }
    }
}
```

```
else
{
    Observaciones = string.Empty;
}

//Así que asigno cada campo a una variable y mando esas variables como argumentos
//para el método en la clase.
//Eso lo hago para poder utilizar el mismo método desde otro formulario
_metodosGenerales.AltaPersona(Nombre,
                              Apellido,
                              Tipo_Documento_ID,
                              Nro_Documento,
                              Tipo_Sexo_ID,
                              Nro_Telefono,
                              Nro_Alternativo,
                              Mail,
                              Observaciones,
                              FechaAlta);
}
private void AltaCliente()
{
    //El ID de la última persona que se registró (método anterior)
    //Tenía otra sentencia, para traer el ID que se generó
    //En esta primera línea, traigo el valor de esta variable y o asigno.
    _clientes.Persona_ID = _metodosGenerales.persona_ID;

    //Luego relleno los campos faltantes de la entidad Cliente.
    if (cmbEstado.SelectedIndex == 0 || cmbEstado.SelectedItem == null)
    {
        _clientes.Estado = "A";
    }
    else
    {
        _clientes.Estado = "I";
    }
    _bussinessClientes.AltaCliente(_clientes);
}

private void ValidarCamposVacios()
{
    //Estos campos que valido, son los obligatorios.
    if (string.IsNullOrEmpty(txtNombreCliente.Text) ||
        string.IsNullOrEmpty(txtApellidoCliente.Text) ||
        string.IsNullOrEmpty(txtNumDocumentoCliente.Text) ||
        string.IsNullOrEmpty(txtTelefonoCliente.Text))
    {
        camposObligatoriosVacios = true;
    }
    else
    {
        camposObligatoriosVacios = false;
    }
}

private void ValidarContenido()
{
    //acá valido el contenido de los campos obligatorios
    //xq pueden estar llenos pero con el texto que los identifica como "Nombre"
    //Y esos strings son los que no quiero que se guarden en la base de datos
    foreach (Control txt in this.gbClientes.Controls)
```

```

{
    if (txt is TextBox)
    {
        if (txt.Text == "Nombre" ||
            txt.Text == "Apellido" ||
            txt.Text == "Documento" ||
            txt.Text == "Teléfono")
        {
            MessageBox.Show("Existen campos con datos erróneos. Por favor, rellene  

            todos los campos con datos válidos", "Campos obligatorios sin rellenar.",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            contenidoErroneo = true;
            break;
        }
        else
        {
            contenidoErroneo = false;
        }
    }
}

private void ResetControls()
{
    //Reseteamos todos los controles del form.
    txtNombreCliente.Text = "Nombre";
    txtApellidoCliente.Text = "Apellido";
    txtNumDocumentoCliente.Text = "Documento";
    txtTelefonoCliente.Text = "Teléfono";
    txtAlternativoCliente.Text = "Alternativo";
    txtMailCliente.Text = "Mail";
    txtObservacionesCliente.Text = "Observaciones y/o consideraciones";
    cmbTipoDocumentoCliente.SelectedItem = 0;
    cmbSexo.SelectedItem = 0;
    foreach (Control txt in this.gbClientes.Controls)
    {
        if (txt is TextBox box)
        {
            txt.ForeColor = Color.DimGray;
        }
    }
}

private void VerificarCoincidencias()
{
    if (motivoMenu != 3)
    {
        int id = _clientes.Persona_ID;
        string documento = txtNumDocumentoCliente.Text.ToString();
        _bussinessPersonas.BuscarCoincidencias(id, documento, _personas);
        if (_personas.Persona_ID == id && _personas.Nro_documento == documento)
        {
            personaRegistrada = true;
        }
        else
        {
            personaRegistrada = false;
        }
    }
}

```

#endregion

```
#region Eventos de Foco en Textbox
```

```
private void txtNombreCliente_Enter(object sender, System.EventArgs e)
{
    //Primero preguntamos si el texto dice Nombre
    //Al principio siempre lo va a decir, porque así
    //inicializa el form. Por lo que lo vamos a vaciar
    //y cambiar el color de la letra.
    //En caso de que no diga literalmente Nombre
    //Se va a dejar como está, ya que puede ser
    //un texto que el usuario escribió

    if (txtNombreCliente.Text == "Nombre")
    {
        txtNombreCliente.Text = string.Empty;
        txtNombreCliente.ForeColor = Color.Black;
    }
}

private void txtNombreCliente_Leave(object sender, System.EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtNombreCliente.Text))
    {
        txtNombreCliente.Text = "Nombre";
        txtNombreCliente.ForeColor = Color.DimGray;
    }
}

/*Así igual para los siguientes eventos dentro de esta región*/
private void txtApellidoCliente_Enter(object sender, System.EventArgs e)
{
    if (txtApellidoCliente.Text == "Apellido")
    {
        txtApellidoCliente.Text = string.Empty;
        txtApellidoCliente.ForeColor = Color.Black;
    }
}

private void txtApellidoCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtApellidoCliente.Text))
    {
        txtApellidoCliente.Text = "Apellido";
        txtApellidoCliente.ForeColor = Color.DimGray;
    }
}

private void txtNumDocumentoCliente_Enter(object sender, System.EventArgs e)
{
    if (txtNumDocumentoCliente.Text == "N° documento")
    {
        txtNumDocumentoCliente.Text = string.Empty;
        txtNumDocumentoCliente.ForeColor = Color.Black;
    }
}

private void txtNumDocumentoCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumDocumentoCliente.Text))
    {
```

```
txtNumDocumentoCliente.Text = "N° documento";
txtNumDocumentoCliente.ForeColor = Color.DimGray;
}
}
private void txtTelefonoCliente_Enter(object sender, System.EventArgs e)
{
    if (txtTelefonoCliente.Text == "Teléfono")
    {
        txtTelefonoCliente.Text = string.Empty;
        txtTelefonoCliente.ForeColor = Color.Black;
    }
}
private void txtTelefonoCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtTelefonoCliente.Text))
    {
        txtTelefonoCliente.Text = "Teléfono";
        txtTelefonoCliente.ForeColor = Color.DimGray;
    }
}

private void txtAlternativoCliente_Enter(object sender, System.EventArgs e)
{
    if (txtAlternativoCliente.Text == "Alternativo")
    {
        txtAlternativoCliente.Text = string.Empty;
        txtAlternativoCliente.ForeColor = Color.Black;
    }
}

private void txtAlternativoCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtAlternativoCliente.Text))
    {
        txtAlternativoCliente.Text = "Alternativo";
        txtAlternativoCliente.ForeColor = Color.DimGray;
    }
}

private void txtMailCliente_Enter(object sender, System.EventArgs e)
{
    if (txtMailCliente.Text == "Mail")
    {
        txtMailCliente.Text = string.Empty;
        txtMailCliente.ForeColor = Color.Black;
    }
}

private void txtMailCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtMailCliente.Text))
    {
        txtMailCliente.Text = "Mail";
        txtMailCliente.ForeColor = Color.DimGray;
    }
}

private void txtObservacionesCliente_Enter(object sender, System.EventArgs e)
{
    if (txtObservacionesCliente.Text == "Observaciones y/o consideraciones")
    {

```



```
        txtObservacionesCliente.Text = string.Empty;
        txtObservacionesCliente.ForeColor = Color.Black;
    }
}

private void txtObservacionesCliente_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtObservacionesCliente.Text))
    {
        txtObservacionesCliente.Text = "Observaciones y/o consideraciones";
        txtObservacionesCliente.ForeColor = Color.DimGray;
    }
}

private void txtBuscarCliente_Enter(object sender, EventArgs e)
{
    if (txtBuscarCliente.Text == "Buscar")
    {
        txtBuscarCliente.Text = string.Empty;
        txtBuscarCliente.ForeColor = Color.Black;
    }
}

private void txtBuscarCliente_Leave(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtBuscarCliente.Text))
    {
        txtBuscarCliente.Text = "Buscar";
        txtBuscarCliente.ForeColor = Color.DimGray;
    }
}

#endregion

#region Eventos KeyPress en Textbox

private void txtNumDocumentoCliente_KeyPress(object sender, KeyPressEventArgs e)
{
    string strTexto = txtNumDocumentoCliente.Text;
    _restricciones.SoloNumeros(e, strTexto);
}

private void txtTelefonoCliente_KeyPress(object sender, KeyPressEventArgs e)
{
    string strTexto = txtTelefonoCliente.Text;
    _restricciones.SoloNumeros(e, strTexto);
}

private void txtAlternativoCliente_KeyPress(object sender, KeyPressEventArgs e)
{
    string strTexto = txtAlternativoCliente.Text;
    _restricciones.SoloNumeros(e, strTexto);
}

private void txtBuscarCliente_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Enter))
    {
        buscar = txtBuscarCliente.Text;
        GetClientes();
    }
}
```

```
#endregion
```

```
#region Alta Clientes
```

```
private void btnAltaCliente_Click(object sender, EventArgs e)
{
    //Verificamos si i el contenido es distinto al texto predeterminado
    //Nombre, Apellido, etc. Tiene que ser distinto a eso,
    //para que sea válido
    ValidarContenido();
    if (!contenidoErroneo)
    {
        //Valido si hay campos obligatorios vacíos
        ValidarCamposVacios();
        if (!camposObligatoriosVacios)
        {
            VerificarCoincidencias();
            if (!personaRegistrada)
            {
                ABMCliente();
                ResetControls();
                GetClientes();
                motivoMenu = 5;
                edicionCliente = false;
            }
            else
            {
                //Vamos a mostrar un mensaje, en caso de que esta persona ya esté
registrada.
                MessageBox.Show("Esta persona ya está registrada en el sistema. Solo
se pueden hacer modificaciones", "Registro encontrada", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
        }
        else
        {
            MessageBox.Show("Campos obligatorios sin completar. Por favor, complete
todos los campos requeridos", "Campos vacios", MessageBoxButtons.OK);
        }
    }
}
#endregion
```

```
private void btnEditarDatosCliente_Click(object sender, EventArgs e)
{
    motivoMenu = 3;

    if (edicionCliente)
    {
        DialogResult result = MessageBox.Show("Tiene una edición en proceso. " +
"En caso de querer cancelar la edición actual presione Aceptar", "Edición
en proceso",
        MessageBoxButtons.OKCancel);
        if (result == DialogResult.OK)
        {
            CargarDatos();
        }
    }
    else
    {
        CargarDatos();
    }
}
```

```

    }
}

private void CargarDatos()
{
    //Asigno el ID de la fila, que está oculto
    //y lo paso como argumento para buscar los datos de ese id en la bdd
    _personas.Persona_ID = Convert.ToInt32(dtgvCliente.CurrentRow.Cells[1].Value);
    _bussinessPersonas.GetPersonaUnica(_personas);

    _clientes.Cliente_ID = Convert.ToInt32(dtgvCliente.CurrentRow.Cells[0].Value);
    _bussinessClientes.GetCliente(_clientes);

    txtNombreCliente.Text = _personas.Nombre;
    txtApellidoCliente.Text = _personas.Apellido;
    cmbTipoDocumentoCliente.SelectedIndex = _personas.Tipo_Documento_ID;
    txtNumDocumentoCliente.Text = _personas.Nro_documento;
    cmbSexo.SelectedIndex = _personas.Tipo_Sexo_ID;
    txtTelefonoCliente.Text = _personas.Nro_Telefono;
    if (string.IsNullOrEmpty(_personas.Nro_Alternativo))
    {
        txtAlternativoCliente.Text = "Alternativo";
    }
    else
    {
        txtAlternativoCliente.Text = _personas.Nro_Alternativo;
    }
    if (string.IsNullOrEmpty(_personas.Mail))
    {
        txtMailCliente.Text = "Mail";
    }
    else
    {
        txtMailCliente.Text = _personas.Mail;
    }
    if (string.IsNullOrEmpty(_personas.Observaciones))
    {
        txtObservacionesCliente.Text = "Observaciones y/o consideraciones";
    }
    else
    {
        txtObservacionesCliente.Text = _personas.Observaciones;
    }
    if (_clientes.Estado == "A")
    {
        cmbEstado.SelectedIndex = 0;
    }
    else
    {
        cmbEstado.SelectedIndex = 1;
    }

    foreach (Control txt in gbClientes.Controls)
    {
        if (txt is TextBox box)
        {
            TextBox t;
            t = box;
            if (t.ForeColor == Color.DimGray &&
                t.Text != "Alternativo" &&
                t.Text != "Mail" &&

```

```
        t.Text != "Observaciones y/o consideraciones")
    {
        t.ForeColor = Color.Black;
    }
}

edicionCliente = true;
}

private void btnEliminarCliente_Click(object sender, EventArgs e)
{
    _clientes.Cliente_ID = Convert.ToInt32(dtgvCliente.CurrentRow.Cells[0].Value);
    _clientes.Estado = "I";
    _bussinessClientes.BajaCliente(_clientes);
    GetClientes();
    motivoMenu = 5;
    edicionCliente = false;
}
}
```