

TECNICATURA
UNIVERSITARIA
EN PROGRAMACIÓN
UTN-FRC



UTN 

Facultad Regional Córdoba

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

PROGRAMACIÓN II

Unidad Temática 3: Acceso a Base de Datos

Guía de Estudio

1^{er} Año - 2^{do} Cuatrimestre

2019



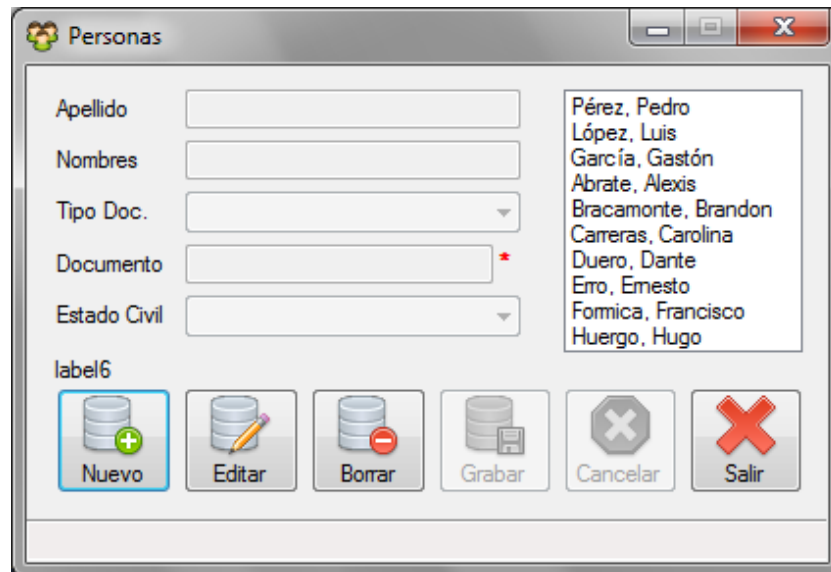
V.0.1

Índice

Ejercicio Nº 1	2
Ejercicio Nº 2	8
Ejercicio Nº 3	14
Ejercicio Nº 4	18

Ejercicio N° 1

Acceso a datos



```

/* Persona.cs *****/
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ProgrII_Ej01
{
    class Persona
    {
        //Atributos:
        string apellido;
        string nombres;
        int tipo_documento;
        double documento;
        int estado_civil;
        //Constructores:
        public Persona()
        {
            apellido = "";
            nombres = "";
            tipo_documento = 0;
            documento = 0;
            estado_civil = 0;
        }
        public Persona(string a, string n, int tipo, double d, int estado)
        {
            apellido = a;
            nombres = n;
            tipo_documento = tipo;
            documento = d;
            estado_civil = estado;
        }
        //Propiedades:
        public string pApellido
        {
            set { apellido = value; }
            get { return apellido; }
        }
        public string pNombres
        {
            set { nombres = value; }
            get { return nombres; }
        }
        public int pTipoDocumento
        {
            set { tipo_documento = value; }
            get { return tipo_documento; }
        }
    }
}

```

```

public double pDocumento
{
    set { documento = value; }
    get { return documento; }
}
public int pEstadoCivil
{
    set { estado_civil = value; }
    get { return estado_civil; }
}
//Métodos:
override public string ToString()
{
    return apellido + ", " + nombres + "(" + documento + ")";
}
}
}

/* frmPersonas.cs *****/
using System.Data.OleDb;
namespace ProgrII_Ej01
{
    public partial class frmPersonas : Form
    {
        OleDbConnection cn;
        OleDbCommand cmd;
        OleDbDataReader dr;
        DataTable dt;

        string cadenaConexion =
            @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirectory|\DBF.mdb";
        string comandoSql;

        const int tam = 100;
        Persona [] P = new Persona[tam];
        int contador;

        bool esNuevo = false;

        public frmPersonas()
        {
            InitializeComponent();
            contador = 0;
            ListarPersonas();
            HabilitarCampos(0);
            cmbTipoDocumento.Enabled = false;
            cmbEstadoCivil.Enabled = false;
        }

        private void button6_Click(object sender, EventArgs e)
        {
            DialogResult opcion = MessageBox.Show("¿Está seguro que desea salir?", "Salir",
                MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (opcion == DialogResult.Yes)
                Close();
        }

        private void frmPersonas_Load(object sender, EventArgs e)
        {
            EstablecerConexion(cadenaConexion);

            //Cargo Tipos de Documento:
            comandoSql = "SELECT * FROM tipo_documento";
            cmd.CommandText = comandoSql;
            dt = new DataTable();
            dt.Load(cmd.ExecuteReader());
            cmbTipoDocumento.DataSource = dt;
            cmbTipoDocumento.ValueMember = "id_tipo_documento";
            cmbTipoDocumento.DisplayMember = "n_tipo_documento";

            //Cargo Estado Civil:
            comandoSql = "SELECT * FROM tipo_estado_civil";
            cmd.CommandText = comandoSql;
            dt = new DataTable();
            dt.Load(cmd.ExecuteReader());
            cmbEstadoCivil.DataSource = dt;
            cmbEstadoCivil.ValueMember = "id_estado_civil";
            cmbEstadoCivil.DisplayMember = "n_estado_civil";
        }
    }
}

```

```

        cn.Close();

        cmbEstadoCivil.SelectedIndex = -1;
        cmbTipoDocumento.SelectedIndex = -1;
    }

    private void EstablecerConexion(string cadenaConex)
    {
        cn = new OleDbConnection(cadenaConex);
        cn.Open();
        cmd = new OleDbCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
    }

    private void lstPersonas_SelectedIndexChanged(object sender, EventArgs e)
    {
        HabilitarCampos(0);
        int i = lstPersonas.SelectedIndex;
        txtApellido.Text = P[i].pApellido;
        txtNombres.Text = P[i].pNombres;
        cmbTipoDocumento.SelectedValue = P[i].pTipoDocumento;
        txtDocumento.Text = P[i].pDocumento.ToString();
        cmbEstadoCivil.SelectedValue = P[i].pEstadoCivil;
    }

    private void ListarPersonas()
    {
        EstablecerConexion(cadenaConexion);

        comandoSql = "SELECT * FROM personas";
        cmd.CommandText = comandoSql;

        dr = cmd.ExecuteReader();
        contador = 0;
        while (dr.Read())
        {
            P[contador] = new Persona();
            P[contador].pDocumento = Convert.ToDouble(dr.GetValue(0));
            P[contador].pApellido = dr.GetString(1);
            P[contador].pNombres = dr.GetString(2);
            P[contador].pTipoDocumento = Convert.ToInt32(dr.GetValue(3));
            P[contador].pEstadoCivil = Convert.ToInt32(dr.GetValue(4));

            contador++;
        }
        dr.Close();
        cn.Close();

        lstPersonas.Items.Clear();
        for (int i = 0; i < contador; i++)
        {
            lstPersonas.Items.Add(P[i].pApellido + ", " + P[i].pNombres);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        esNuevo = true;
        HabilitarCampos(1);
    }

    private void HabilitarCampos(int nroBoton)
    {
        switch(nroBoton)
        {
            case 0: //Inicio:
                cmbTipoDocumento.SelectedIndex = -1;
                cmbEstadoCivil.SelectedIndex = -1;
                btnNuevo.Enabled = true;
                btnEditar.Enabled = true;
                btnBorrar.Enabled = true;
                btnGrabar.Enabled = false;
                btnCancelar.Enabled = false;
                btnSalir.Enabled = true;
                cmbTipoDocumento.Enabled = false;
                cmbEstadoCivil.Enabled = false;

```

```

foreach (Control c in this.Controls)
{
    if (c is TextBox)
    {
        c.Text = "";
        c.Enabled = false;
    }
}
txtApellido.Focus();
break;
case 1: //btnNuevo:
cmbTipoDocumento.SelectedIndex = -1;
cmbEstadoCivil.SelectedIndex = -1;
foreach (Control c in this.Controls)
{
    if (c is TextBox)
    {
        c.Text = "";
        c.Enabled = true;
    }
}
txtApellido.Focus();
btnGrabar.Enabled = true;
btnEditar.Enabled = false;
btnCancelar.Enabled = true;
btnBorrar.Enabled = false;
cmbTipoDocumento.Enabled = true;
cmbEstadoCivil.Enabled = true;
break;
case 2: //btnEditar:
btnNuevo.Enabled = false;
btnEditar.Enabled = false;
btnBorrar.Enabled = false;
btnGrabar.Enabled = true;
btnCancelar.Enabled = true;
btnSalir.Enabled = true;
cmbTipoDocumento.Enabled = true;
cmbEstadoCivil.Enabled = true;
foreach (Control c in this.Controls)
{
    if (c is TextBox)
    {
        c.Enabled = true;
    }
}
txtApellido.Focus();
break;
case 3: //btnBorrar:
cmbTipoDocumento.SelectedIndex = -1;
cmbEstadoCivil.SelectedIndex = -1;
cmbTipoDocumento.Enabled = false;
cmbEstadoCivil.Enabled = false;
foreach (Control c in this.Controls)
{
    if (c is TextBox)
    {
        c.Text = "";
    }
}
break;
case 4: //btnGrabar:
btnNuevo.Enabled = true;
btnEditar.Enabled = true;
btnBorrar.Enabled = true;
btnGrabar.Enabled = false;
btnCancelar.Enabled = false;
btnSalir.Enabled = true;
cmbTipoDocumento.Enabled = false;
cmbEstadoCivil.Enabled = false;
break;
case 5: //btnCancelar
cmbTipoDocumento.Enabled = false;
cmbEstadoCivil.Enabled = false;
HabilitarCampos(0);
break;
}
}

```

```
private void button4_Click(object sender, EventArgs e)
{
    //Validar
    if (string.IsNullOrEmpty(txtDocumento.Text))
    {
        MessageBox.Show("Debe introducir los campos requeridos.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        HabilitarCampos(0);
    }
    else
    {
        HabilitarCampos(4);
        grabarPersona(esNuevo);

        HabilitarCampos(0);
        MessageBox.Show("La persona se ha ingresado satisfactoriamente.", "Datos ingresados",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        esNuevo = false;
        ListarPersonas();
    }
}

private void grabarPersona(bool esNuevo)
{
    Persona persona = new Persona();
    persona.pApellido = txtApellido.Text;
    persona.pNombres = txtNombres.Text;
    persona.pTipoDocumento = Convert.ToInt32(cmbTipoDocumento.SelectedValue);
    persona.pDocumento = Convert.ToDouble(txtDocumento.Text);
    persona.pEstadoCivil = Convert.ToInt32(cmbEstadoCivil.SelectedValue);
    if (esNuevo)
    {
        comandoSql = @"INSERT INTO Personas VALUES (" +
            persona.pDocumento + "," +
            persona.pApellido + "," +
            persona.pNombres + "," +
            persona.pTipoDocumento + "," +
            persona.pEstadoCivil + ")";
    }
    else
    {
        int personaSeleccionada = Convert.ToInt32(lstPersonas.SelectedIndex);
        string dni = P[personaSeleccionada].pDocumento.ToString();

        comandoSql = "UPDATE personas SET apellido = '" + persona.pApellido +
            "', nombres = '" + persona.pNombres +
            "', tipo_documento = '" + persona.pTipoDocumento.ToString() +
            "', documento = '" + persona.pDocumento.ToString() +
            "', tipo_estado_civil = '" + persona.pEstadoCivil.ToString() +
            " WHERE documento = '" + dni;
    }
    EstablecerConexion(cadenaConexion);
    cmd.CommandText = comandoSql;
    cmd.ExecuteNonQuery();
    cn.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    HabilitarCampos(3);
    if (lstPersonas.SelectedIndex >= 0)
    {
        DialogResult opcion =
            MessageBox.Show("¿Está seguro que desea borrar la persona seleccionada?",
                "Eliminar Persona", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (opcion == DialogResult.Yes)
        {
            int item = Convert.ToInt32(lstPersonas.SelectedIndex); //SelectedValue???
            string dni = P[item].pDocumento.ToString();

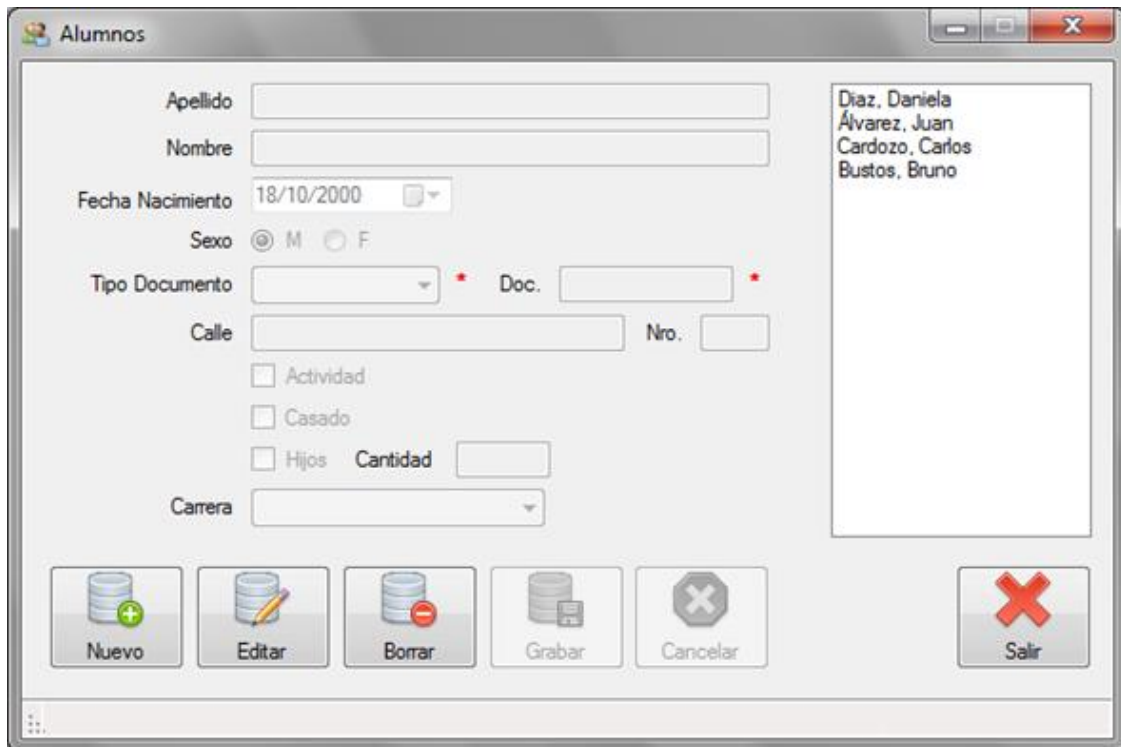
            EstablecerConexion(cadenaConexion);
            comandoSql = "DELETE FROM personas WHERE documento=" + dni;
            cmd.CommandText = comandoSql;
            cmd.ExecuteNonQuery();
            cn.Close();

            ListarPersonas();
        }
    }
}
```

```
        else if (opcion == DialogResult.No)
        {
            HabilitarCampos(0);
        }
    }
    else
    {
        MessageBox.Show("Seleccione la persona que desea eliminar.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
private void btnEditar_Click(object sender, EventArgs e)
{
    if (lstPersonas.SelectedIndex < 0)
        MessageBox.Show("Seleccione la persona que desea editar.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
        HabilitarCampos(2);
}
private void btnCancelar_Click(object sender, EventArgs e)
{
    HabilitarCampos(0);
    esNuevo = false;
}
private void button1_Click_1(object sender, EventArgs e)
{
    lstPersonas.Items.Clear();
    ListarPersonas();
}
}
```


Ejercicio N° 2

Acceso a datos – Alumnos



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ProgrII_Ej02
{
    public partial class frmAlumno : Form
    {
        OleDbCommand cmd;
        OleDbConnection cn;
        OleDbDataReader dr;
        DataTable dt;

        string cadenaConexion = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=x\VS.Net\ProgrII_Ej02\DBF_ABM_alumno.mdb";
        string comandoSql;
        string comando;
        string value;
        string display;
        const int tam = 50;
        Alumno[] alumno = new Alumno[tam];
        int contador = 0;
        bool alumnoNuevo = false;

        public frmAlumno()
        {
            InitializeComponent();
        }

        private void EstablecerConexion(string cadenaConex)
        {
            cn = new OleDbConnection(cadenaConexion);
            cn.Open();
            cmd = new OleDbCommand();
            cmd.Connection = cn;
            cmd.CommandType = CommandType.Text;
        }
    }
}
```

```
private void frmAlumno_Load(object sender, EventArgs e)
{
    EstablecerConexion(cadenaConexion);
    CargarAlumno();
    ListarAlumnos();

    comando = "Select * from tipo_documento";
    value = "id_tipo_documento";
    display = "n_tipo_documento";
    CargarCombos(cmbTipoDocumento, comando, value, display);

    comando = "Select * from carreras";
    value = "id_carrera";
    display = "n_carrera";
    CargarCombos(cmbCarrera, comando, value, display);

    LimpiarControles();
    HabilitarControles(false);
    HabilitarBotones(true);

    cn.Close();
}
#region acceso a datos
private void CargarAlumno()
{
    comandoSql = "SELECT * FROM alumnos";
    EstablecerConexion(cadenaConexion);
    cmd.CommandText = comandoSql;
    dr = cmd.ExecuteReader();
    contador = 0;
    while (dr.Read())
    {
        alumno[contador] = new Alumno();
        alumno[contador].pApellido = dr.GetString(0);
        alumno[contador].pNombres = dr.GetString(1);
        alumno[contador].pFechaNacimiento = dr.GetDateTime(2);
        alumno[contador].pSexo = dr.GetInt32(3);
        alumno[contador].pTipoDocumento = dr.GetInt32(4);
        alumno[contador].pDocumento = dr.GetInt32(5);
        alumno[contador].pCalle = dr.GetString(6);
        alumno[contador].pNumero = dr.GetInt32(7);
        alumno[contador].pActividad = dr.GetBoolean(8);
        alumno[contador].pCasado = dr.GetBoolean(9);
        alumno[contador].pHijos = dr.GetBoolean(10);
        alumno[contador].pCantidad = dr.GetInt32(11);
        alumno[contador].pCarrera = dr.GetInt32(12);
        contador++;
    }
    dr.Close();
}
public void CargarCombos(ComboBox cmb, string comando, string value, string display)
{
    cmd.CommandText = comando;
    dt = new DataTable();
    dt.Load(cmd.ExecuteReader());
    cmb.DataSource = dt;
    cmb.ValueMember = value;
    cmb.DisplayMember = display;
}
private void btnBorrar_Click(object sender, EventArgs e)
{
    DialogResult opcion = MessageBox.Show("¿Está seguro que desea eliminar el alumno  
seleccionado?", "Salir", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (opcion == DialogResult.Yes)
    {
        string dni = alumno[lstAlumnos.SelectedIndex].pDocumento.ToString();
        comandoSql = "DELETE FROM alumnos WHERE documento = " + dni;

        EstablecerConexion(cadenaConexion);
        cmd.CommandText = comandoSql;
        cmd.ExecuteNonQuery();

        CargarAlumno();
        ListarAlumnos();
        cn.Close();
    }
}
```

```

else //Cancelo:
{
    LimpiarControles();
    HabilitarBotones(true);
}
}

private void btnGrabar_Click(object sender, EventArgs e)
{
    //Verificacion:
    if (cmbTipoDocumento.SelectedValue != null && !string.IsNullOrEmpty(txtDocumento.Text)
        && dtpFechaNacimiento.Value <= DateTime.Now)
    {
        //grabar
        Alumno alumnoGrabar = new Alumno();
        alumnoGrabar = new Alumno();
        alumnoGrabar.pApellido = txtApellido.Text;
        alumnoGrabar.pNombres = txtNombre.Text;
        alumnoGrabar.pFechaNacimiento = dtpFechaNacimiento.Value;

        if (rbMasculino.Checked)
            alumnoGrabar.pSexo = 1;
        else
            alumnoGrabar.pSexo = 2;

        alumnoGrabar.pTipoDocumento = Convert.ToInt32(cmbTipoDocumento.SelectedValue);
        alumnoGrabar.pDocumento = int.Parse(txtDocumento.Text);
        alumnoGrabar.pCalle = txtCalle.Text;
        alumnoGrabar.pNumero = int.Parse(txtNumero.Text);
        alumnoGrabar.pActividad = checkActividad.Checked;
        alumnoGrabar.pCasado = checkCasado.Checked;
        alumnoGrabar.pHijos = checkHijos.Checked;

        if (checkHijos.Checked)
            alumnoGrabar.pCantidad = Convert.ToInt32(txtCantidad.Text);
        else
            alumnoGrabar.pCantidad = 0;
        alumnoGrabar.pCarrera = Convert.ToInt32(cmbCarrera.SelectedValue);

        if (alumnoNuevo)
        {
            string verificacionDni = @"SELECT COUNT(*)
                FROM alumnos
                WHERE tipo_documento = " + alumnoGrabar.pTipoDocumento +
                "ANDdocumento = " + alumnoGrabar.pDocumento;
            EstablecerConexion(cadenaConexion);
            cmd.CommandText = verificacionDni;
            if (Convert.ToInt32(cmd.ExecuteScalar()) > 0)
                MessageBox.Show("Ya existe el alumno", "Error",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            else
            {
                //Insert
                comandoSql = @"INSERT INTO alumnos VALUES ('" +
                    alumnoGrabar.pApellido + "', '" +
                    alumnoGrabar.pNombres + "', #" +
                    alumnoGrabar.pFechaNacimiento + "#, " +
                    alumnoGrabar.pSexo + ", " +
                    alumnoGrabar.pTipoDocumento + ", " +
                    alumnoGrabar.pDocumento + ", '" +
                    alumnoGrabar.pCalle + "', " +
                    alumnoGrabar.pNumero + ", " +
                    alumnoGrabar.pActividad + ", " +
                    alumnoGrabar.pCasado + ", " +
                    alumnoGrabar.pHijos + ", " +
                    alumnoGrabar.pCantidad + ", " +
                    alumnoGrabar.pCarrera + "')";
                EstablecerConexion(cadenaConexion);
                cmd.CommandText = comandoSql;
                cmd.ExecuteNonQuery();
                cn.Close();
                CargarAlumno();
                ListarAlumnos();
            }
        }
    }
}

```

```

else
{
    //Update
    int alumnoSeleccionado = Convert.ToInt32(lstAlumnos.SelectedIndex);
    string dni = alumno[alumnoSeleccionado].pDocumento.ToString();

    comandoSql = "UPDATE alumnos " +
        "SET apellido = '" + alumnoGrabar.pApellido + "', " +
        "nombres = '" + alumnoGrabar.pNombres + "', " +
        "fecha_nacimiento = #" + alumnoGrabar.pFechaNacimiento + "#, " +
        "sexo = '" + alumnoGrabar.pSexo + "', " +
        "tipo_documento = '" + alumnoGrabar.pTipoDocumento + "', " +
        "documento = '" + alumnoGrabar.pDocumento.ToString() + "', " +
        "calle = '" + alumnoGrabar.pCalle + "', " +
        "numero = '" + alumnoGrabar.pNumero + "', " +
        "actividad = '" + alumnoGrabar.pActividad + "', " +
        "casado = '" + alumnoGrabar.pCasado + "', " +
        "hijos = '" + alumnoGrabar.pHijos + "', " +
        "cantidad = '" + alumnoGrabar.pCantidad + "', " +
        "carrera = '" + alumnoGrabar.pCarrera + "' " +
        "WHERE tipo_documento = '" + alumno[alumnoSeleccionado].pTipoDocumento +
        " AND documento = '" + alumno[alumnoSeleccionado].pDocumento.ToString();

    EstablecerConexion(cadenaConexion);
    cmd.CommandText = comandoSql;
    cmd.ExecuteNonQuery();
    cn.Close();
    CargarAlumno();
    ListarAlumnos();
}
HabilitarBotones(true);
HabilitarControles(false);
LimpiarControles();
alumnoNuevo = false;
}
else
{
    MessageBox.Show("Faltan Datos o la fecha especificada es incorrecta.",
        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    HabilitarControles(true);
    HabilitarBotones(false);
}
}
#endregion

private void btnNuevo_Click(object sender, EventArgs e)
{
    LimpiarControles();
    HabilitarControles(true);
    HabilitarBotones(false);
    alumnoNuevo = true;
}

private void btnEditar_Click(object sender, EventArgs e)
{
    HabilitarControles(true);
    HabilitarBotones(false);

    alumnoNuevo = false;
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    HabilitarBotones(true);
    HabilitarControles(false);
    LimpiarControles();
    alumnoNuevo = false;
}

private void btnSalir_Click(object sender, EventArgs e)
{
    DialogResult opcion = MessageBox.Show("¿Está seguro que desea salir?", "Salir",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
    if (opcion == DialogResult.Yes)
        Close();
}

```

```

/*****/
#region manejo de la interfaz
void LimpiarControles()
{
    txtApellido.Text = null;
    txtNombre.Text = null;
    txtCalle.Text = null;
    txtNumero.Text = null;
    dtpFechaNacimiento.Value = DateTime.Now;
    rbMasculino.Checked = true;
    rbFemenino.Checked = false;
    cmbTipoDocumento.SelectedIndex = -1;
    txtDocumento.Text = null;
    checkActividad.Checked = false;
    checkCasado.Checked = false;
    checkHijos.Checked = false;
    txtCantidad.Text = null;
    cmbCarrera.SelectedIndex = -1;
}
void HabilitarControles(bool opcion)
{
    txtApellido.Enabled = opcion;
    txtNombre.Enabled = opcion;
    txtCalle.Enabled = opcion;
    txtNumero.Enabled = opcion;
    dtpFechaNacimiento.Enabled = opcion;
    rbMasculino.Enabled = opcion;
    rbFemenino.Enabled = opcion;
    cmbTipoDocumento.Enabled = opcion;
    txtDocumento.Enabled = opcion;
    checkActividad.Enabled = opcion;
    checkCasado.Enabled = opcion;
    checkHijos.Enabled = opcion;
    txtCantidad.Enabled = false;

    if (checkHijos.Checked)
        txtCantidad.Enabled = true;
    else
        txtCantidad.Enabled = false;

    cmbCarrera.Enabled = opcion;
    txtApellido.Focus();
    lstAlumnos.Enabled = !opcion;
}

public void HabilitarBotones(bool opcion)
{
    btnNuevo.Enabled = opcion;
    btnEditar.Enabled = opcion;
    btnBorrar.Enabled = opcion;
    btnGrabar.Enabled = !opcion;
    btnCancelar.Enabled = !opcion;
}

private void checkHijos_CheckedChanged(object sender, EventArgs e)
{
    if (checkHijos.Checked)
        txtCantidad.Enabled = true;
    else
        txtCantidad.Enabled = false;
}

private void textBox_Enter(object sender, EventArgs e)
{
    (sender as TextBox).SelectAll();
}

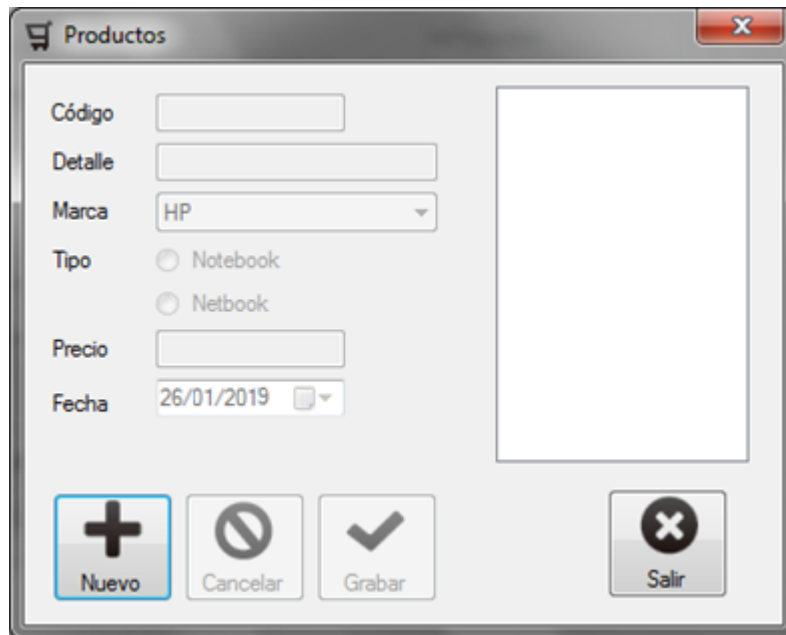
private void ListarAlumnos()
{
    lstAlumnos.Items.Clear();
    for (int i = 0; i < contador; i++)
    {
        lstAlumnos.Items.Add(alumno[i].pApellido + ", " + alumno[i].pNombres);
    }
}

```

```
private void lstAlumnos_SelectedIndexChanged(object sender, EventArgs e)
{
    int i = lstAlumnos.SelectedIndex;
    txtApellido.Text = alumno[i].pApellido;
    txtNombre.Text = alumno[i].pNombres;
    dtpFechaNacimiento.Value = alumno[i].pFechaNacimiento;
    if (alumno[i].pSexo == 1)
        rbMasculino.Checked = true;
    else
        rbFemenino.Checked = true;
    cmbTipoDocumento.SelectedValue = alumno[i].pTipoDocumento;
    txtDocumento.Text = alumno[i].pDocumento.ToString();
    txtCalle.Text = alumno[i].pCalle;
    txtNumero.Text = alumno[i].pNumero.ToString();
    checkActividad.Checked = alumno[i].pActividad;
    checkCasado.Checked = alumno[i].pCasado;
    checkHijos.Checked = alumno[i].pHijos;
    txtCantidad.Text = alumno[i].pCantidad.ToString();
    cmbCarrera.SelectedValue = alumno[i].pCarrera;
}
#endregion
}
```

Ejercicio N° 3

Acceso a datos – Productos



```

/* Producto.cs *****/
namespace ProgrII_Ej03
{
    class Producto
    {
        public int codigo { get; set; }
        public int marca { get; set; }
        public string detalle { get; set; }
        public int tipo { get; set; }
        public double precio { get; set; }
        public DateTime fecha { get; set; }

        public override string ToString()
        {
            return codigo.ToString() + " - " + detalle;
        }
    }
}

/* Form1.cs *****/
namespace ProgrII_Ej03
{
    public partial class frmProductos : Form
    {
        SqlConnection cn;
        SqlCommand cmd;
        SqlDataReader dr;
        DataTable dt;

        string cadenaConexion = @"Data Source=.\SQLEXPRESS; Initial Catalog = progII_dbf;
            Integrated Security = True";
        string comandoSql;

        const int tam = 50;
        Producto[] prod = new Producto[tam];
        int contador = 0;

        public frmProductos()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            habilitarBotones(true);
            HabilitarControles(false);
            cargarLista();
        }
    }
}

```

```
//Cargo marcas:
cn = new SqlConnection(cadenaConexion);
cn.Open();
cmd = new SqlCommand();
cmd.Connection = cn;
cmd.CommandType = CommandType.Text;
comandoSql = "SELECT * FROM marca";
cmd.CommandText = comandoSql;
dt = new DataTable();
dt.Load(cmd.ExecuteReader());
cmbMarca.DataSource = dt;
cmbMarca.ValueMember = "id_tipo_marca";
cmbMarca.DisplayMember = "n_tipo_marca";
cn.Close();
}

private void btnNuevo_Click(object sender, EventArgs e)
{
    habilitarBotones(false);
    LimpiarControles();
    HabilitarControles(true);
    txtCodigo.Focus();
}

private void HabilitarControles(bool opcion)
{
    txtCodigo.Enabled = opcion;
    txtDetalle.Enabled = opcion;
    txtPrecio.Enabled = opcion;
    rbNetbook.Enabled = opcion;
    rbNotebook.Enabled = opcion;
    dtFecha.Enabled = opcion;
    cmbMarca.Enabled = opcion;
    lstProductos.Enabled = !opcion;
}

private void LimpiarControles()
{
    txtCodigo.Clear();
    txtDetalle.Clear();
    txtPrecio.Clear();
    rbNotebook.Checked = true;
    rbNetbook.Checked = false;
    dtFecha.Value = DateTime.Now;
    cmbMarca.SelectedIndex = 0;
}

private bool Validar()
{
    if (string.IsNullOrEmpty(txtCodigo.Text))
    {
        MessageBox.Show("Faltan Datos", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        txtCodigo.Focus();
        return false;
    }
    if (string.IsNullOrEmpty(txtPrecio.Text))
    {
        MessageBox.Show("Faltan Datos", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        txtPrecio.Focus();
        return false;
    }
    if (string.IsNullOrEmpty(txtDetalle.Text))
    {
        MessageBox.Show("Faltan Datos", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        txtDetalle.Focus();
        return false;
    }
    return true;
}

private void habilitarBotones(bool opcion)
{
    btnNuevo.Enabled = opcion;
    btnGrabar.Enabled = !opcion;
    btnCancelar.Enabled = !opcion;
}
```



```
private void cargarLista()
{
    contador = 0;
    Conectar();
    cmd.CommandText="SELECT * FROM producto";
    dr = cmd.ExecuteReader();
    while(dr.Read())
    {
        prod[contador] = new Producto();
        prod[contador].codigo = dr.GetInt32(0);
        prod[contador].detalle = dr.GetString(1);
        prod[contador].tipo = dr.GetInt32(2);
        prod[contador].marca = dr.GetInt32(3);
        prod[contador].precio =(double) (dr.GetDecimal(4));
        prod[contador].fecha = dr.GetDateTime(5);
        contador++;
    }
    Desconectar();
    lstProductos.Items.Clear();
    for (int i = 0; i < contador; i++)
    {
        lstProductos.Items.Add(prod[i].ToString());
    }
}
private void btnGrabar_Click(object sender, EventArgs e)
{
    if (Validar())
    {
        Producto nuevoProd = new Producto();
        nuevoProd.codigo = Convert.ToInt32(txtCodigo.Text);
        nuevoProd.detalle = txtDetalle.Text;
        nuevoProd.marca = Convert.ToInt32(cmbMarca.SelectedValue);
        if (rbNotebook.Checked)
            nuevoProd.tipo = 1;
        else if (rbNetbook.Checked)
            nuevoProd.tipo = 2;
        else
            nuevoProd.tipo = 0;
        nuevoProd.precio = Convert.ToDouble(txtPrecio.Text);
        nuevoProd.fecha = dtFecha.Value;

        comandoSql = "INSERT INTO producto " +
            "(codigo, detalle, tipo, marca, precio, fecha) VALUES (" +
            nuevoProd.codigo.ToString() + ", '" +
            nuevoProd.detalle + "', " +
            nuevoProd.marca + ", " +
            nuevoProd.tipo + ", " +
            nuevoProd.precio + ", '" +
            nuevoProd.fecha.ToShortDateString() + "')";

        Conectar();
        cmd.CommandText = comandoSql;
        cmd.ExecuteNonQuery();
        Desconectar();
        cargarLista();
        habilitarBotones(true);
        HabilitarControles(false);
        LimpiarControles();
    }
}

private void Desconectar()
{
    cn.Close();
}

private void Conectar()
{
    cn = new SqlConnection(cadenaConexion);
    cn.Open();
    cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandType = CommandType.Text;
}
```

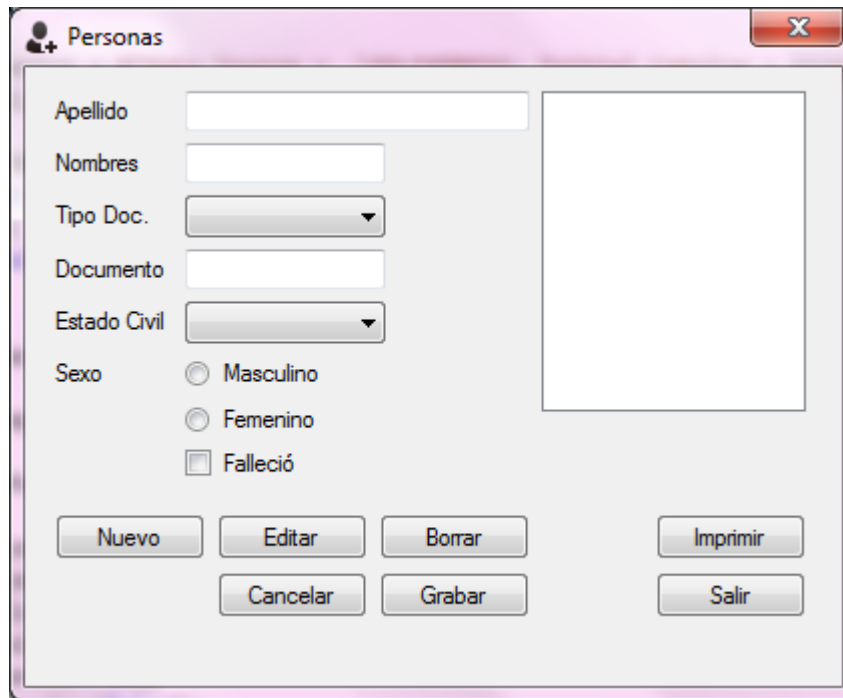
```
private void btnCancelar_Click(object sender, EventArgs e)
{
    habilitarBotones(true);
    LimpiarControles();
    HabilitarControles(false);
}

private void lstProductos_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstProductos.SelectedIndex < 0)
    {
        MessageBox.Show("Seleccione un producto", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    else
    {
        txtCodigo.Text = prod[lstProductos.SelectedIndex].codigo.ToString();
        txtPrecio.Text = prod[lstProductos.SelectedIndex].precio.ToString();
        txtDetalle.Text = prod[lstProductos.SelectedIndex].detalle;
        if (prod[lstProductos.SelectedIndex].tipo == 1)
            rbNotebook.Checked = true;
        else if (prod[lstProductos.SelectedIndex].tipo == 2)
            rbNetbook.Checked = true;
        cmbMarca.SelectedValue= prod[lstProductos.SelectedIndex].marca;
        dtFecha.Value = prod[lstProductos.SelectedIndex].fecha;
    }
}

private void btnSalir_Click(object sender, EventArgs e)
{
    Close();
}
}
```

Ejercicio N° 4

Acceso a datos - Personas



```

/* Persona.cs *****/
namespace ProgrII_Ej04
{
    class Persona
    {
        public string nombres { get; set; }
        public string apellido { get; set; }
        public int tipoDocumento { get; set; }
        public int documento { get; set; }
        public int estadoCivil { get; set; }
        public int sexo { get; set; }
        public int fallecio { get; set; }

        public override string ToString()
        {
            return apellido + ", " + nombres + " (" + documento.ToString() + ")";
        }
    }
}

/* Form1.cs *****/
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace ProgrII_Ej14
{
    public partial class frmPersonas : Form
    {
        #region variables globales
        SqlConnection cn;
        SqlCommand cmd;
        SqlDataReader dr;
        DataTable dt;
        string cadenaConexion = @"Data Source = .\SQLEXPRESS;
            Initial Catalog = progII_dbf_personas; Integrated Security = True";
        string comandoSql;

        const int tam = 50;
    }
}

```

```

Persona[] P = new Persona[tam];
int contador = 0;
bool esNuevo = false;
#endregion

public frmPersonas()
{
    InitializeComponent();
}

private void frmPersonas_Load(object sender, EventArgs e)
{
    CargarCombo(cmbTipoDocumento, "tipo_documento", 0, 1);
    CargarCombo(cmbEstadoCivil, "estado_civil", 0, 1);
    CargarLista(lstPersonas, "personas");
    LimpiarControles();
    HabilitarBotones(true);
    HabilitarControles(false);
}
#region controles
private void LimpiarControles()
{
    txtApellido.Clear();
    txtNombres.Clear();
    txtDocumento.Clear();
    cmbEstadoCivil.SelectedIndex = 1;
    cmbTipoDocumento.SelectedIndex = 1;
    rbMasculino.Checked = true;
    rbFemenino.Checked = false;
    ckFallecio.Checked = false;
}
private void HabilitarBotones(bool opcion)
{
    btnNuevo.Enabled = opcion;
    btnEditar.Enabled = opcion;
    btnBorrar.Enabled = opcion;
    btnCancelar.Enabled = !opcion;
    btnGrabar.Enabled = !opcion;
    btnImprimir.Enabled = opcion;
}
private void HabilitarControles(bool opcion)
{
    txtApellido.Enabled = opcion;
    txtNombres.Enabled = opcion;
    txtDocumento.Enabled = opcion;
    cmbTipoDocumento.Enabled = opcion;
    cmbEstadoCivil.Enabled = opcion;
    rbFemenino.Enabled = opcion;
    rbMasculino.Enabled = opcion;
    ckFallecio.Enabled = opcion;
    lstPersonas.Enabled = !opcion;
}
#endregion

#region acceso a datos
private void Conectar()
{
    cn = new SqlConnection(cadenaConexion);
    cn.Open();
    cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandType = CommandType.Text;
}
private void Desconectar()
{
    cn.Close();
}
private void CargarCombo(ComboBox cmb, string tabla, int valueMbr, int displayMbr)
{
    Conectar();
    cmd.CommandText = "SELECT * FROM " + tabla;
    dt = new DataTable();
    dt.Load(cmd.ExecuteReader());
    cmb.DataSource = dt;
    cmb.ValueMember = dt.Columns[valueMbr].ColumnName;
    cmb.DisplayMember = dt.Columns[displayMbr].ColumnName;
    Desconectar();
}

```

```
private void CargarLista(ListBox lst, string tabla)
{
    contador = 0;
    Conectar();
    cmd.CommandText = "SELECT * FROM " + tabla;
    dr = cmd.ExecuteReader();

    while (dr.Read())
    {
        P[contador] = new Persona();
        P[contador].nombres = dr.GetString(dr.GetOrdinal("nombres"));
        P[contador].apellido = dr.GetString(dr.GetOrdinal("apellido"));
        P[contador].tipoDocumento = dr.GetInt32(dr.GetOrdinal("tipo_documento"));
        P[contador].documento = dr.GetInt32(dr.GetOrdinal("Documento"));
        P[contador].estadoCivil = dr.GetInt32(dr.GetOrdinal("estado_civil"));
        P[contador].sexo = dr.GetInt32(dr.GetOrdinal("sexo"));
        P[contador].fallecio = dr.GetInt32(dr.GetOrdinal("fallecio"));
        contador++;
    }
    Desconectar();
    lst.Items.Clear();
    for (int i = 0; i < contador; i++)
    {
        lst.Items.Add(P[i].ToString());
    }
}
#endregion

private void btnImprimir_Click(object sender, EventArgs e)
{
    frmPersonasReporte reportePersonas = new frmPersonasReporte();
    reportePersonas.ShowDialog();
}

private void btnSalir_Click(object sender, EventArgs e)
{
    DialogResult opcion = MessageBox.Show("¿Está seguro que desea salir?",
        "Salir", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (opcion == DialogResult.Yes)
        Close();
}

private void btnNuevo_Click(object sender, EventArgs e)
{
    esNuevo = true;
    LimpiarControles();
    HabilitarControles(true);
    HabilitarBotones(false);
    txtApellido.Focus();
}

private void btnEditar_Click(object sender, EventArgs e)
{
    if (lstPersonas.SelectedIndex < 0)
    {
        MessageBox.Show("Seleccione la persona que desea editar.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        esNuevo = false;
        HabilitarControles(true);
        HabilitarBotones(false);
        txtApellido.Focus();
    }
}

private bool Validar()
{
    if (string.IsNullOrEmpty(txtApellido.Text))
    {
        MessageBox.Show("Faltan datos.", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        txtApellido.Focus();
        return false;
    }
    try
    {
        Int32.Parse(txtDocumento.Text);
        return true;
    }
}
```

```

catch
{
    MessageBox.Show("Coloque sólo números.", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    txtApellido.Focus();
    return false;
}
if (string.IsNullOrEmpty(txtNombres.Text))
{
    MessageBox.Show("Faltan datos.", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    txtNombres.Focus();
    return false;
}
if (string.IsNullOrEmpty(txtDocumento.Text))
{
    MessageBox.Show("Faltan datos.", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    txtDocumento.Focus();
    return false;
}
return true;
}
private void btnCancelar_Click(object sender, EventArgs e)
{
    LimpiarControles();
    HabilitarControles(false);
    HabilitarBotones(true);
}

private void btnGrabar_Click(object sender, EventArgs e)
{
    if (Validar())
    {
        Persona nuevaPers = new Persona();
        nuevaPers.nombres = txtNombres.Text;
        nuevaPers.apellido = txtApellido.Text;
        nuevaPers.tipoDocumento = Convert.ToInt32(cmbTipoDocumento.SelectedValue);
        nuevaPers.documento = Convert.ToInt32(txtDocumento.Text);
        nuevaPers.estadoCivil = Convert.ToInt32(cmbEstadoCivil.SelectedValue);

        if (rbMasculino.Checked)
            nuevaPers.sexo = 1;
        else
            nuevaPers.sexo = 2;
        if (ckFallecio.Checked)
            nuevaPers.fallecio = 1;
        else
            nuevaPers.fallecio = 0;

        string buscarDuplicado = "SELECT tipo_documento FROM personas WHERE " +
            "documento = " + nuevaPers.documento.ToString() +
            "AND tipo_documento = " + nuevaPers.tipoDocumento.ToString();
        Conectar();
        cmd.CommandText = buscarDuplicado;
        bool existe;
        if (Convert.ToInt32(cmd.ExecuteScalar()) > 0)
        {
            MessageBox.Show("Ya existe la persona que desea cargar", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            existe = true;
        }
        else
            existe = false;
        if (esNuevo && !existe)
        {
            comandoSql = "INSERT INTO personas (nombres, apellido, tipo_documento, " +
                "documento, estado_civil, sexo, fallecio) " +
                "VALUES ('" +
                nuevaPers.nombres + "', '" +
                nuevaPers.apellido + "', " +
                nuevaPers.tipoDocumento.ToString() + ", " +
                nuevaPers.documento.ToString() + ", " +
                nuevaPers.estadoCivil.ToString() + ", " +
                nuevaPers.sexo + ", " +
                nuevaPers.fallecio.ToString() + ")";
        }
    }
}

```

```

    }
    else if (!esNuevo)
    {
        comandoSql = "UPDATE personas SET " +
            "apellido = '" + nuevaPers.apellido + "', " +
            "nombres = '" + nuevaPers.nombres + "', " +
            "tipo_documento = " + nuevaPers.tipoDocumento.ToString() + ", " +
            "documento = " + nuevaPers.documento.ToString() + ", " +
            "estado_civil = " + nuevaPers.estadoCivil.ToString() + ", " +
            "sexo = " + nuevaPers.sexo.ToString() + ", " +
            "fallecio = " + nuevaPers.fallecio.ToString() +
            "WHERE tipo_documento = " + P[1stPersonas.SelectedIndex].tipoDocumento.ToString() +
            "AND documento = " + P[1stPersonas.SelectedIndex].documento;
    }
    else return;
    Conectar();
    cmd.CommandText = comandoSql;
    cmd.ExecuteNonQuery();
    Desconectar();
    HabilitarControles(false);
    HabilitarBotones(true);
    LimpiarControles();
    CargarLista(1stPersonas, "Personas");
}
}
private void 1stPersonas_SelectedIndexChanged(object sender, EventArgs e)
{
    int posic = 1stPersonas.SelectedIndex;
    txtApellido.Text = P[posic].apellido;
    txtNombres.Text = P[posic].nombres;
    txtDocumento.Text = P[posic].documento.ToString();
    cmbTipoDocumento.SelectedValue = P[posic].tipoDocumento;
    cmbEstadoCivil.SelectedValue = P[posic].estadoCivil;
    if (P[posic].sexo == 1)
        rbMasculino.Checked = true;
    else
        rbFemenino.Checked = true;
    if (P[posic].fallecio == 0)
        ckFallecio.Checked = false;
    else
        ckFallecio.Checked = true;
}
private void btnBorrar_Click(object sender, EventArgs e)
{
    if (1stPersonas.SelectedIndex < 0)
    {
        MessageBox.Show("Seleccione la persona que desea eliminar.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        DialogResult opcion = MessageBox.Show("¿Está seguro que desea eliminar la persona
            seleccionada?", "Eliminar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (opcion == DialogResult.Yes)
        {
            comandoSql = "DELETE FROM personas WHERE " +
                "tipo_documento = " + P[1stPersonas.SelectedIndex].tipoDocumento.ToString() +
                "AND documento = " + P[1stPersonas.SelectedIndex].documento.ToString();
            Conectar();
            cmd.CommandText = comandoSql;
            cmd.ExecuteNonQuery();
            Desconectar();
            CargarLista(1stPersonas, "personas");
        }
    }
}
}
}
}
}

```



Atribución-NoComercial-SinDerivadas

Se permite descargar esta obra y compartirla, siempre y cuando se de crédito a la Universidad Tecnológica Nacional como autor de la misma. No puede modificarse y/o alterarse su contenido, ni comercializarse.