

Guía de prácticas en C# 2013

Ing. Mg. Patricio Medina
Ing. Mg. Dario Robayo

Primera Edición

GUÍA DE PRÁCTICAS



EN C# 2013

Ing. Mg. Patricio Medina

*Profesor de la Universidad Técnica de Ambato
Profesor de la Pontificia Universidad Católica del Ecuador Sede Ambato
Asesor Informático*

Ing. Mg. Dario Robayo

*Profesor de la Pontificia Universidad Católica del Ecuador Sede Ambato
Asesor Informático*

Primera Edición

Editorial Independiente

Autores:

Ing. Mg. Patricio Medina

Capacitación realizada en: Cuba, Brasil y Ecuador

Contacto: ricardopmedina@uta.edu.ec, pmedina@pucesa.edu.ec, medina_patricio@yahoo.es

Ing. Mg. Dario Robayo

Capacitación realizada en: Ecuador

Contacto: drobayo@pucesa.edu.ec

ISBN- 978-9942-21-146-0

Certificado Instituto de Propiedad Intelectual: QUI-044948

Primera Edición

Ambato – Ecuador

2015

Todos los nombres propios de productos y servicios que aparecen en este texto son marcas registradas de sus respectivas compañías u organizaciones. Denotamos éstos tan solo con fines de divulgación.

Las posiciones expresadas en este libro reflejan exclusivamente la opinión de su autor y por lo tanto no representan en ningún caso la posición de la UTA y PUCESA.

Publicación de distribución gratuita. Los contenidos pueden difundirse y reproducirse con fines educativos y con la condición de reconocer los créditos correspondientes.

INTRODUCCIÓN

Los hombres que valen la pena cruzan el mar de la vida nadando.
Los demás se contentan con bañarse en la orilla
(Anónimo)

Javier Olivar ¹hace una reflexión sobre que: toda persona que pretenda construir un programa informático que de solución a determinada problemática, se enfrenta con dos grandes tareas.

La primera es el QUÉ, es decir, qué [acciones](#) debe realizar para poder resolver el problema al cual se está enfrentando y el cual se realiza con papel y lápiz, como parte del trabajo de mesa previo a toda actividad de programación; la segunda es el CÓMO, o sea, de qué instrucciones se va a valer para escribir el código que realice las acciones determinadas en el QUÉ, las cuales están determinadas por el lenguaje de programación seleccionado.

Considero que si, un programador primero se le entrena en Lógica de Programación y conoce los controles básicos de una herramienta visual podrá enfrentar cualquier lenguaje de programación al punto que, fácilmente hará relación con los diferentes controles, analizar las propiedades y comportamientos del mismo.

Si bien es cierto la programación actual se enfoca a la programación visual, de objetos, de agentes, de hilos, entre otras, no se puede dejar de lado la formación del estudiante en el manejo de estos elementos; por ello esta guía de prácticas ha sido diseñado como una herramienta de auto aprendizaje, que contiene un variado conjunto de ejercicios prácticos que ilustran el uso de dichos controles, fortaleciendo el manejo de las mismos y creando una experiencia importantísima en el estudiante

Para obtener los mejores resultados en cada práctica, recomiendo al Lector analizar detenidamente cada indicación.

Esperando recibir sus comentarios y sugerencias serán muy bienvenidas a:
medina_patricio@yahoo.es, drobayo@pucesa.edu.ec

¹ Blog <http://logicaunellezfs.blogspot.com/2011/07/actividad-de-logica.html>

PRÓLOGO

C# Net 2013, es actualmente uno de esos términos mágicos que revolucionan la programación. El éxito de este lenguaje viene de la mano de la filosofía y la forma de operación de las aplicaciones estrechamente ligadas a la plataforma Microsoft.

El hecho de que C# sea una herramienta joven en evolución hace que se requieran guías para su explotación y aprendizaje. Sin embargo esto unido al profesionalismo y entrega del docente, crean una nueva forma de involucrar de una manera más activa a los estudiantes, así como de incentivar a muchos en aras de conseguir una docencia de mayor calidad, tal como lo han logrado Patricio Medina y Darío Robayo con iniciativas como éste trabajo.

Con la realización de esta guía se perseguían una serie de objetivos tanto docentes como pragmáticos. Los objetivos docentes estaban centrados en la búsqueda de la perfección y mejora de la calidad docente, reflejada en una mayor participación de los alumnos y en una mejora de la relación profesor-estudiante. Los objetivos pragmáticos se centraban en el acercamiento del lenguaje C# al currículum de los estudiantes matriculados en la asignatura correspondiente en la Escuela de Ingeniería en Sistemas de la PUCESA.

Bienvenido este nuevo recurso académico para el docente y estudiante, a disfrutarla haciendo que el trabajo del profesor se refleje en el aprendizaje del estudiante.

Ing. Galo López Sevilla
DIRECTOR ESCUELA INGENIERÍA EN SISTEMAS
PUCESA

DEDICATORIA

"El profesor debe ser un guía para el aprendizaje, proporcionando pautas y herramientas para recorrer el camino". (Fernando Rodríguez - Operación Éxito)

Mis buenos amigos de EIS PUCESA,
gracias por tan valiosa amistad.

Patricio

A mi familia Teo, Danny y Naty por su apoyo lleno de amor. A mis amigos de EIS PUCESA, por su sincera amistad y los buenos momentos compartidos.

Darío

INDICE

ETIQUETAS.....	1
BOTONES	4
OPERACIONES BÁSICAS.....	7
MINI CALCULADORA	11
MANEJO DE FORMULARIOS POR VARIABLES	16
FORMULARIOS MDI POR HERENCIA.....	19
PALETA DE COLORES CON TRACKBAR	23
TRACKBAR Y NUMERICUPDOWN	26
SPLITCONTAINER	29
PROPIEDAD OPACIDAD	32
JUEGO DEL 7	35
CASINO	38
ENLACES	41
CARRERAS.....	43
MANEJO DE TECLAS	46
MANEJO DE TECLAS 2	49
USO DE VARIABLES	53
DECLARAR ESTRUCTURAS	55
BARRA DE HERRAMIENTAS	58
CASILLAS DE VERIFICACIÓN Y BOTONES DE ACCIÓN	62
CALENDARIO.....	66
USO DE LISTBOX	69
LISTBOX PRÁCTICA.....	73
USANDO CHECKLISTBOX	77
USO DE LISTVIEW BÁSICO	80
LISTVIEW CON ARCHIVO PLANO	83
TREEVIEW	87
FUNCIONES Y PROCEDIMIENTOS LOCALES.....	90
FUNCIONES Y PROCEDIMIENTOS GLOBALES	94
MOVIMIENTO DE FIGURA	98
MASCARAS Y ARCHIVOS DE TEXTO	101
CONTROL DE ERRORES Y AYUDA.....	104
CONEXIÓN CON BASE DE DATOS ACCESS POR CÓDIGO.....	108
CONSULTAS POR CÓDIGO	112
CONSULTAS POR COMPONENTES	118

INSERTAR REGISTROS POR CÓDIGO	127
ACTUALIZAR REGISTROS POR CÓDIGO.....	132
HA TENER PRESENTE. FORMATOS.....	138

ETIQUETAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Mensajes
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	4
Name	Mensaje1
Text	"Esteban y Ariel"
AutoSize	True
BorderStyle	None
BackColor	(A su gusto)
ForeColor	(A su gusto)
Font	(Tamaño 24, Negrita)
Name	Mensaje2
Text	"Esteban y Ariel"
AutoSize	False
BorderStyle	FixedSingle
BackColor	(A su gusto)

ForeColor	(A su gusto)
Font	(Tamaño 10, Negrita)
Size	244; 43
TextAlign	MiddleCenter
Name	Mensaje3
Text	“Esteban y Ariel”
BorderStyle	Fixed3D
BackColor	(A su gusto)
ForeColor	(A su gusto)
Font	(Tamaño 10, Negrita)
Size	244; 43
TextAlign	MiddleLeft
Image	(Seleccione una imagen)
Name	Mensaje4
Text	Fondo
BorderStyle	FixedSingle
BackColor	(A su gusto)
ForeColor	(A su gusto)
Font	(Tamaño 10, Negrita)

BUTTON

Cantidad	1
Name	Boton1
BackColor	(A su gusto)
Cursor	Hand
Font	(A su gusto)
Text	Terminar
TextAlign	MiddleCenter

COMBOBOX

Cantidad	1
Name	Lista
Ítems	Rojo Verde Azul
DropDownStyle	DropDownList

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
```

```

    {
        var Aux = MessageBox.Show("Desea Abandonar", "Aviso",
        MessageBoxButtons.YesNo);
        if (Aux == DialogResult.Yes)
        {
            Application.Exit();
        }
    }

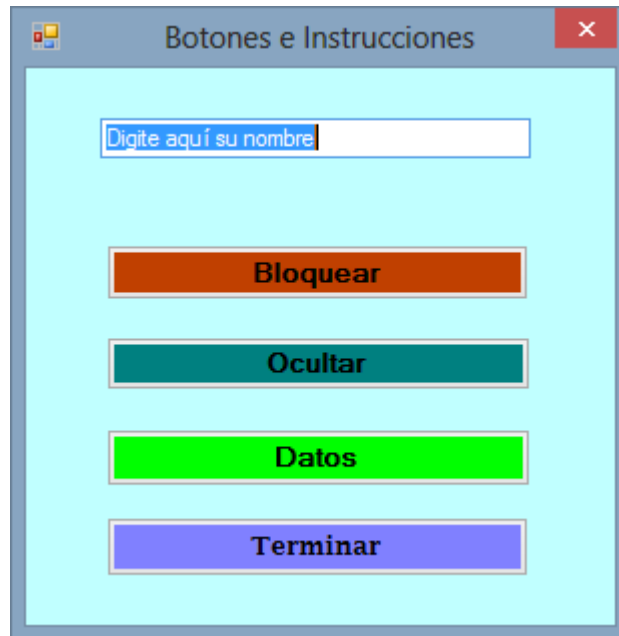
private void Lista_SelectedIndexChanged(object sender,
EventArgs e)
{
    switch (Lista.SelectedIndex)
    {
        case 0:
            this.BackColor = Color.Red;
            break;
        case 1:
            this.BackColor = Color.Green;
            break;
        case 2:
            this.BackColor = Color.Blue;
            break;
    }
}
}
}

```

5. GRABAR Y EJECUTAR

BOTONES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Botones e Instrucciones
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

TEXTBOX

Cantidad	1
Name	Tdato
Text	Digite aquí su Nombre

BUTTON

Cantidad	4
Name	Boton1
BackColor	(A su gusto)
Cursor	Hand
Font	(A su gusto)
Text	Bloquear

TextAlign	MiddleCenter
Name	Boton2
BackColor	(A su gusto)
Cursor	Hand
Font	(A su gusto)
Text	Ocultar
TextAlign	MiddleCenter
Name	Boton3
BackColor	(A su gusto)
Cursor	Hand
Font	(A su gusto)
Text	Datos
TextAlign	MiddleCenter
Name	Boton4
BackColor	(A su gusto)
Cursor	Hand
Font	(A su gusto)
Text	Terminar
TextAlign	MiddleCenter

4. CODIGO

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        if (Tdato.Enabled)
        {
            Boton1.Text = "Desbloquear";
            Tdato.Enabled = false;
        }
        else
        {
            Boton1.Text = "Bloquear";
            Tdato.Enabled = true;
        }
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        if (Tdato.Visible)
        {
            Boton2.Text = "Mostrar";
        }
    }
}

```

```
        Tdato.Visible = false;
    }
    else
    {
        Boton2.Text = "Ocultar";
        Tdato.Visible = true;
    }
}

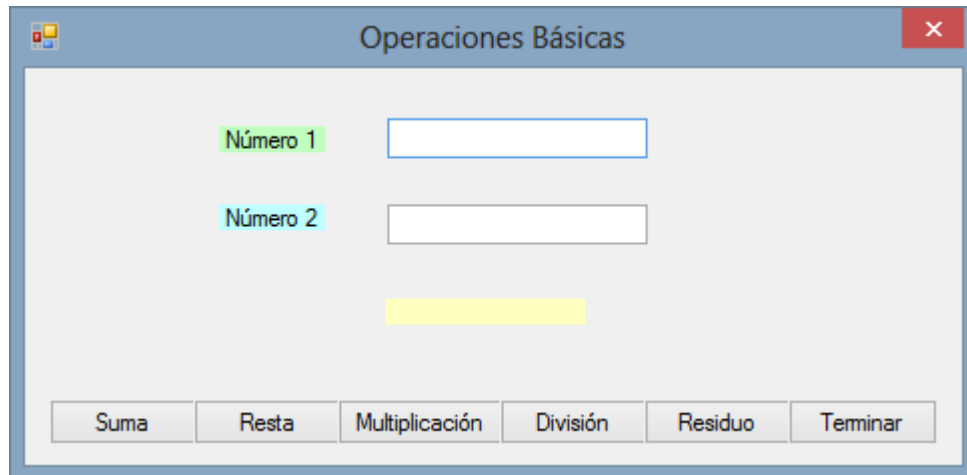
private void Boton3_Click(object sender, EventArgs e)
{
    // Forma de visualizar rápida
    MessageBox.Show("Tú nombre es: " + Tdato.Text);
}

private void Boton4_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
```

5. GRABAR Y EJECUTAR

OPERACIONES BÁSICAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



The screenshot shows a window titled "Operaciones Básicas" with a standard Windows title bar. Inside the window, there are two text boxes. The first is labeled "Número 1" and the second is labeled "Número 2". Below these text boxes is a yellow rectangular area. At the bottom of the window, there is a row of six buttons: "Suma", "Resta", "Multiplicación", "División", "Residuo", and "Terminar".

3. Componentes

FORM

Name	Form1
Text	Operaciones Básicas
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	3
Name	Mensaje1
Text	Número 1
BackColor	(A su gusto)
Name	Mensaje2
Text	Número 1
BackColor	(A su gusto)
Name	Mensaje3
Text	(Vacío)
BackColor	(A su gusto)

TEXTBOX

Cantidad	2
Name	Tdato1

Text (Vacío)

Name Tdato2

Text (Vacío)

BUTTON

Cantidad 6

Name Boton1

Text Suma

Name Boton2

Text Resta

Name Boton3

Text Multiplicación

Name Boton4

Text División

Name Boton5

Text Residuo

Name Boton6

Text Terminar

4. CODIGO

Public Class Form1

```
public partial class Form1 : Form
{
    //Son las Variables para todo el Formulario
    decimal Dato1, Dato2, Resul;

    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //Suma
        decimal Num1, Num2, Aux;
        try{
            Num1 = Convert.ToDecimal(Tdato1.Text);
            Num2 = Convert.ToDecimal(Tdato2.Text);
        }
        catch {
```

```

        Num1 = 0;
        Num2 = 0;
    }
    Aux = Num1 + Num2;
    Mensaje3.Text = String.Format("{0:###0.00}", Aux);
}

private void Boton2_Click(object sender, EventArgs e)
{
    //Resta
    Ingreso();
    Result = Dato1 - Dato2;
    Mensaje3.Text = String.Format("{0:###0.00}", Result);
}

private void Boton3_Click(object sender, EventArgs e)
{
    //Multiplicacion
    Ingreso();
    Result = Dato1 * Dato2;
    Mensaje3.Text = String.Format("{0:###0.00}", Result);
}

private void Boton4_Click(object sender, EventArgs e)
{
    //División
    Ingreso();
    if (Dato2 <= 0)
        MessageBox.Show("Número 2 es Incorrecto");
    else
        Result = Dato1 / Dato2;
    Mensaje3.Text = String.Format("{0:###0.00}", Result);
}

private void Boton5_Click(object sender, EventArgs e)
{
    //Residuo de la división
    Ingreso();
    if (Dato2 <= 0)
        MessageBox.Show("Número 2 es Incorrecto");
    else
        Result = Dato1 % Dato2;
    Mensaje3.Text = String.Format("{0:###0.00}", Result);
}

private void Boton6_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Ingreso()
{
    try

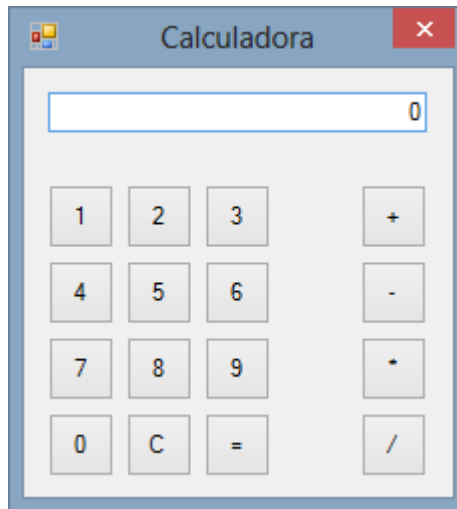
```

```
        {  
            Dato1 = Convert.ToDecimal(Tdato1.Text);  
            Dato2 = Convert.ToDecimal(Tdato2.Text);  
        }  
        catch  
        {  
            Dato1 = 0;  
            Dato2 = 0;  
        }  
    }  
}
```

5. GRABAR Y EJECUTAR

MINI CALCULADORA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Calculadora
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

TEXTBOX

Cantidad	1
Name	TxtPantalla
TextAlign	Right
Text	0

BUTTON

Cantidad	16
Name	Btn1
Text	1
TextAlign	MiddleCenter
Name	Btn2
Text	2
TextAlign	MiddleCenter
Name	Btn3

Text	3
TextAlign	MiddleCenter
Name	Btn4
Text	4
TextAlign	MiddleCenter
Name	Btn5
Text	5
TextAlign	MiddleCenter
Name	Btn6
Text	6
TextAlign	MiddleCenter
Name	Btn7
Text	7
TextAlign	MiddleCenter
Name	Btn8
Text	8
TextAlign	MiddleCenter
Name	Btn9
Text	9
TextAlign	MiddleCenter
Name	Btn10
Text	0
TextAlign	MiddleCenter
Name	BtnC
Text	C
TextAlign	MiddleCenter
Name	BtnIgual
Text	=
TextAlign	MiddleCenter
Name	BtnSuma
Text	+
TextAlign	MiddleCenter
Name	BtnResta
Text	-
TextAlign	MiddleCenter
Name	BtnMulti
Text	*
TextAlign	MiddleCenter

Name	BtnDiv
Text	/
TextAlign	MiddleCenter

4. CODIGO

```
public partial class Form1 : Form
{
    int Guarda = 0;
    String Signo;

    public Form1()
    {
        InitializeComponent();
    }

    private void calcular(int Valor)
    {
        TxtPantalla.Text =
            Convert.ToString(((Convert.ToInt16(TxtPantalla.Text) * 10)
            + Valor));
    }

    private void Operador(String operador)
    {
        Signo = operador;
        Guarda = Convert.ToInt16(TxtPantalla.Text);
        TxtPantalla.Text = "0";
    }

    private void Btn1_Click(object sender, EventArgs e)
    {
        calcular(1);
    }

    private void Btn2_Click(object sender, EventArgs e)
    {
        calcular(2);
    }

    private void Btn3_Click(object sender, EventArgs e)
    {
        calcular(3);
    }

    private void Btn4_Click(object sender, EventArgs e)
    {
        calcular(4);
    }

    private void Btn5_Click(object sender, EventArgs e)
    {
        calcular(5);
    }
}
```

```

}

private void Btn6_Click(object sender, EventArgs e)
{
    calcular(6);
}

private void Btn7_Click(object sender, EventArgs e)
{
    calcular(7);
}

private void Btn8_Click(object sender, EventArgs e)
{
    calcular(8);
}

private void Btn9_Click(object sender, EventArgs e)
{
    calcular(9);
}

private void Btn10_Click(object sender, EventArgs e)
{
    calcular(0);
}

private void BtnC_Click(object sender, EventArgs e)
{
    TxtPantalla.Text = "0";
}

private void BtnSuma_Click(object sender, EventArgs e)
{
    Operador("+");
}

private void BtnResta_Click(object sender, EventArgs e)
{
    Operador("-");
}

private void BtnMulti_Click(object sender, EventArgs e)
{
    Operador("*");
}

private void BtnDiv_Click(object sender, EventArgs e)
{
    Operador("/");
}

private void BtnIgual_Click(object sender, EventArgs e)
{

```

```

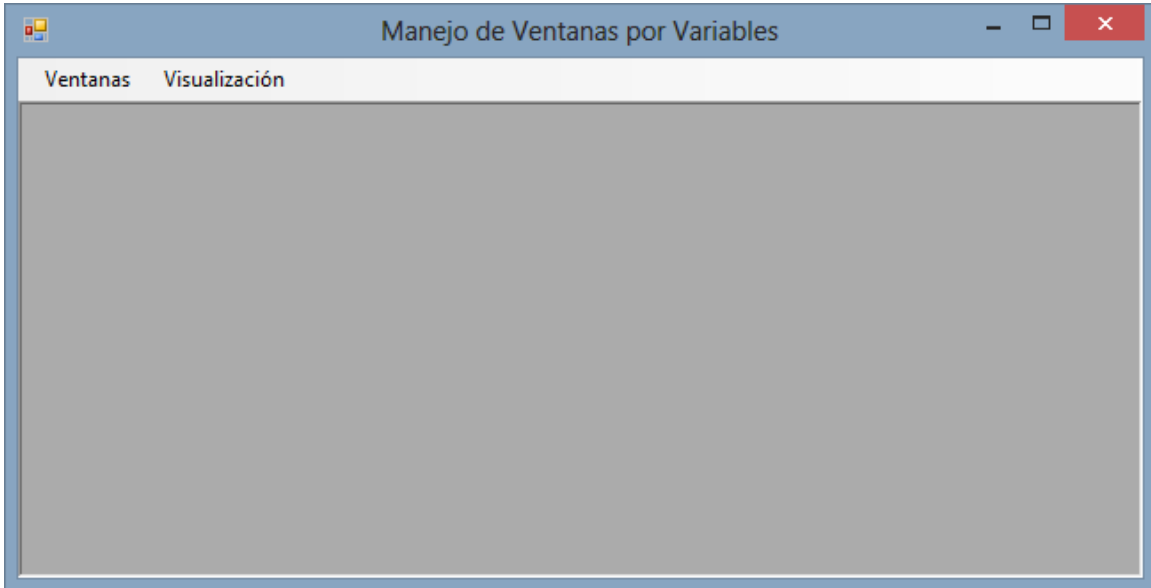
switch (Signo)
{
    case "+" : TxtPantalla.Text = Convert.ToString(Guarda
+ Convert.ToInt16(TxtPantalla.Text));
        break;
    case "-" : TxtPantalla.Text = Convert.ToString(Guarda
- Convert.ToInt16(TxtPantalla.Text));
        break;
    case "*" : TxtPantalla.Text = Convert.ToString(Guarda
* Convert.ToInt16(TxtPantalla.Text));
        break;
    case "/":
        if (Convert.ToInt16(TxtPantalla.Text) == 0)
            MessageBox.Show("No se puede dividir para 0");
        else
            TxtPantalla.Text = Convert.ToString(Guarda /
            Convert.ToInt16(TxtPantalla.Text));
        break;
}
}
}

```

5. GRABAR Y EJECUTAR

MANEJO DE FORMULARIOS POR VARIABLES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

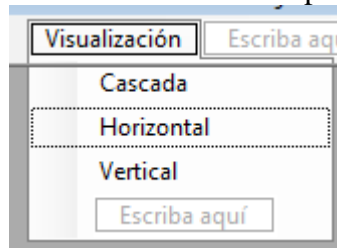
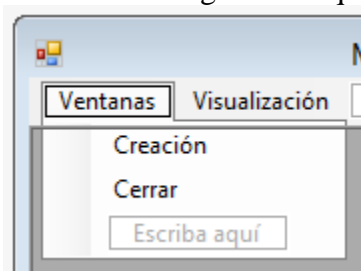
FORM

Name	Form1
Text	Manejo de Ventanas por Variables
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Maximized
IsMdiContainer	True

MENUSTRIP

Cantidad	1
Name	MiMenu

Diseñe el siguiente esquema, teniendo seleccionado el MenuStrip



4. CODIGO

Nota: Para colocar código en cada opción del menú ejemplo Creación; hacer doble clic sobre la misma.

```
public partial class Form1 : Form
{
    byte Posi;
    public Form1()
    {
        InitializeComponent();
    }

    private void creaciónToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //CREACION
        Form MiNuevoForm = new Form();
        MiNuevoForm.MdiParent = this;
        Posi += 1;
        MiNuevoForm.Text = "Ventana" + Convert.ToString(Posi);
        MiNuevoForm.Show();
    }

    private void cerrarToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //CERRAR
        this.Close();
    }

    private void cascadaToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //CASCADA
        this.LayoutMdi(MdiLayout.Cascade);
    }

    private void horizontalToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //HORIZONTAL
        this.LayoutMdi(MdiLayout.TileHorizontal);
    }

    private void verticalToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
```

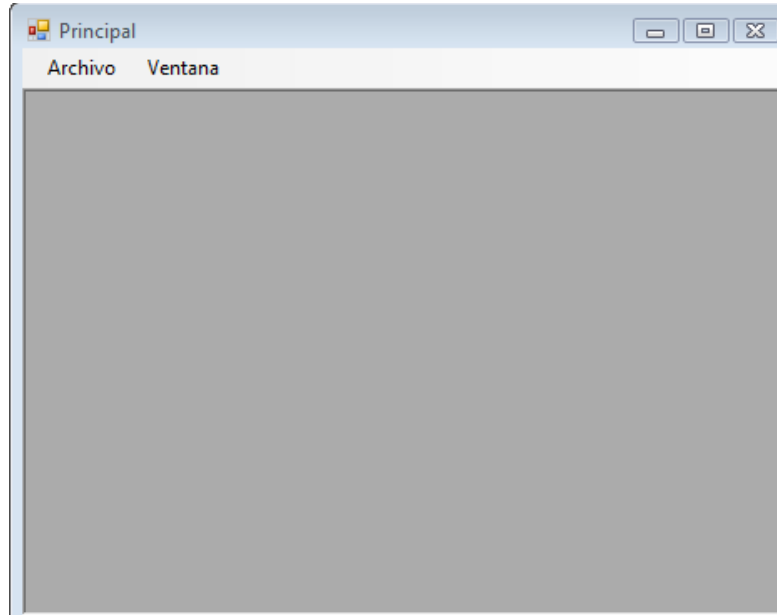
```
        //VERTICAL
        this.LayoutMdi(MdiLayout.TileVertical);
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        // INICIALIZACION DE VARIABLES CUANDO SE CARGA EL PROYECTO
        Posi = 0;
    }
}
```

5. GRABAR Y EJECUTAR

FORMULARIOS MDI POR HERENCIA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario Principal



3. Componentes

FORM

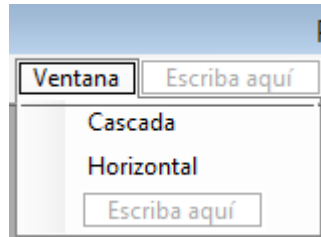
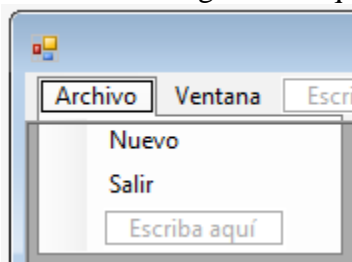
Name	FormPrincipal
Text	Principal
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Maximized
IsMdiContainer	True

MENUSTRIP

Cantidad 1

Name MiMenu

Diseñe el siguiente esquema, teniendo seleccionado el MainMenu



4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void nuevoToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //OPCION NUEVO
        FormSecundario Miform = new FormSecundario();

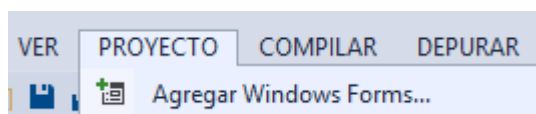
        //FORMSECUNDARIO ES EL NOMBRE DEL OTRO FORMULARIO
        Miform.MdiParent = this;
        Miform.Show();
    }

    private void salirToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //OPCION SALIR
        this.Close();
    }

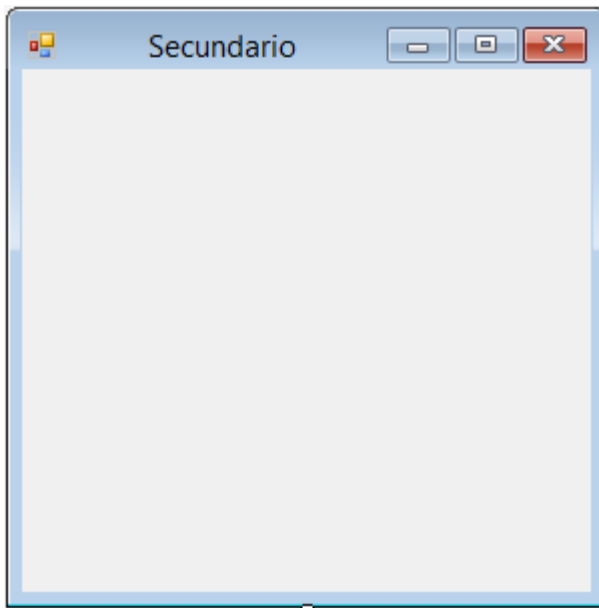
    private void cascadaToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //OPCION CASCADA
        this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);
    }

    private void horizontalToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        //OPCION HORIZONTAL
        this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizonta
        l);
    }
}
```

5. Agregar un Formulario en blanco (Windows Form) en:



6. Apariencia del Formulario Secundario



7. Componentes

FORM

Name	FormSecundario
Text	Secundario
MaximizeBox	True
MinimizeBox	True
StartPosition	WindowsDefaultLocation
WindowState	Normal

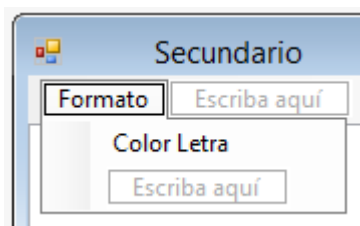
MENUSTRIP

Cantidad	1
Name	OtroMenu

TEXTBOX

Cantidad	1
Name	Tdato
Multiline	True
Dock	Fill

Diseñe el siguiente esquema, teniendo seleccionado el MainMenu



NOTA. Poner de nombre a la opción como LetraMenuItem

8. CODIGO

EVENTOS FORMULARIO SECUNDARIO

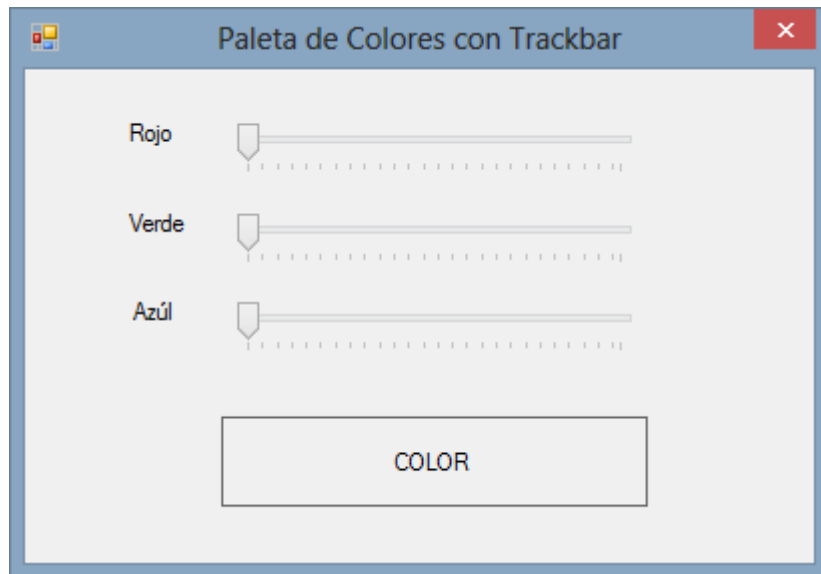
```
public partial class FormSecundario : Form
{
    public FormSecundario()
    {
        InitializeComponent();
    }

    private void LetraMenuItem_Click(object sender, EventArgs e)
    {
        if (LetraMenuItem.Checked)
        {
            LetraMenuItem.Checked = false;
            Tdato.ForeColor = System.Drawing.Color.Black;
        }
        else
        {
            LetraMenuItem.Checked = true;
            Tdato.ForeColor = System.Drawing.Color.Blue;
        }
    }
}
```

9. GRABAR Y EJECUTAR

PALETA DE COLORES CON TRACKBAR

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Paleta de Colores con TrackBar
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	4
Name	Mensaje1
Text	Rojo
BackColor	(A su gusto)
Name	Mensaje2
Text	Verde
BackColor	(A su gusto)
Name	Mensaje3
Text	Azul
BackColor	(A su gusto)
Name	Mensaje4
AutoSize	False

BorderStyle	FixedSingle
Text	Color:
TextAlign	MiddleCenter
BackColor	(A su gusto)

TRACKBAR

Cantidad 3

Name	trackBar1
LargeChange	5
SmallChange	1
Maximum	255
Minimum	0
Orientation	Horizontal
TickFrequency	10
TickStyle	Opción 3

Name	trackBar2
LargeChange	5
SmallChange	1
Maximum	255
Minimum	0
Orientation	Horizontal
TickFrequency	10
TickStyle	Opción 3

Name	trackBar3
LargeChange	5
SmallChange	1
Maximum	255
Minimum	0
Orientation	Horizontal
TickFrequency	10
TickStyle	Opción 3

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void MiPaleta()
    {
        //Tomamos los valores de cada TrackBar
        int CR, CV, CA;
        CR = trackBar1.Value;
        CV = trackBar2.Value;
    }
}
```

```

        CA = trackBar3.Value;
        Mensaje4.BackColor = Color.FromArgb(CR, CV, CA);
        Mensaje4.Text = "Color: Rojo " + CR + " Verde " + CV + "
Azul " + CA;
    }

    private void trackBar1_Scroll(object sender, EventArgs e)
    {
        //Rojo
        MiPaleta();
    }

    private void trackBar2_Scroll(object sender, EventArgs e)
    {
        //Verde
        MiPaleta();
    }

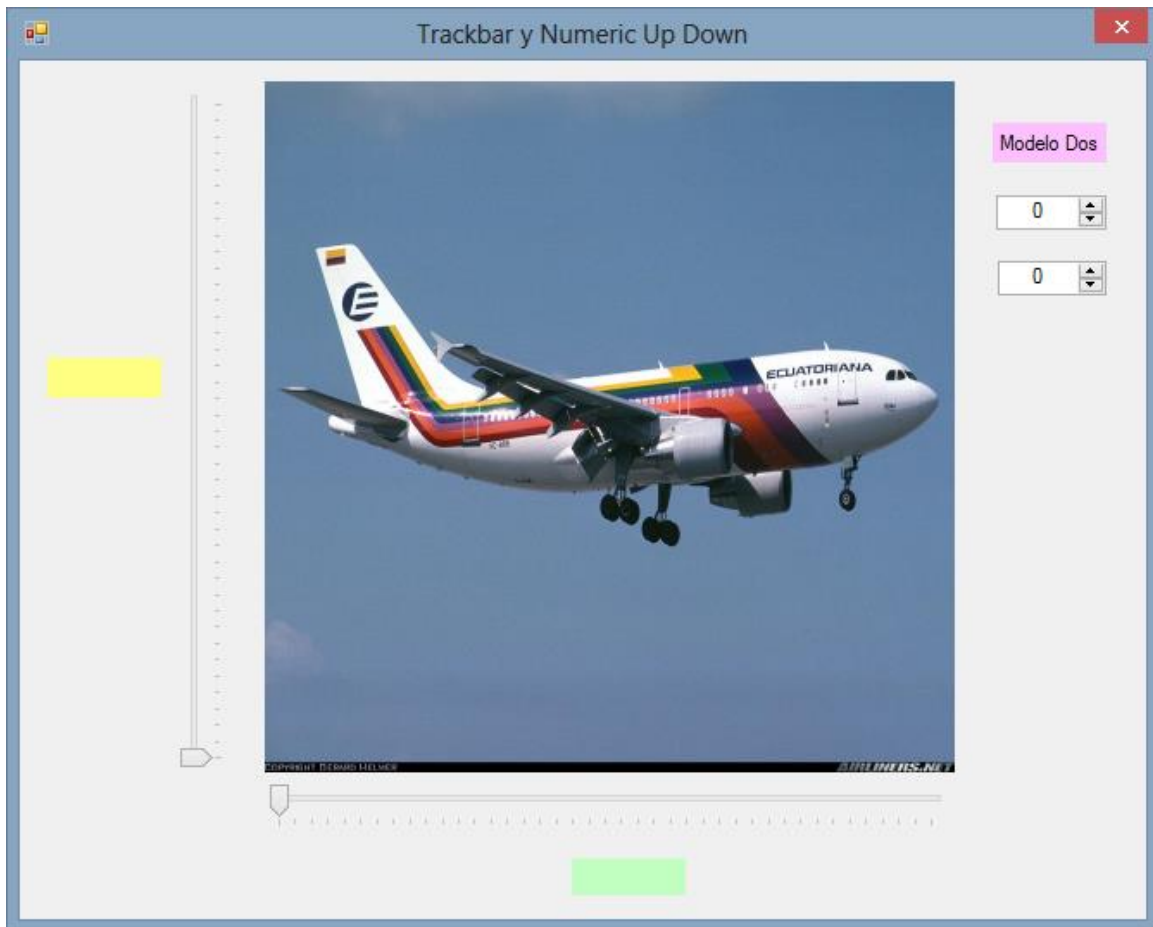
    private void trackBar3_Scroll(object sender, EventArgs e)
    {
        //Azul
        MiPaleta();
    }
}

```

5. GRABAR Y EJECUTAR

TRACKBAR Y NUMERICUPDOWN

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	TrackBar y NumericUpDown
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	3
Name	Mensaje1
Text	(Vacío)
BackColor	(A su gusto)

Name Mensaje2
Text (Vacío)
Backcolor (A su gusto)

Name Mensaje3
Text Modelo Dos
Backcolor (A su gusto)

TRACKBAR

Cantidad 2

Name TrackBarH
LargeChange 10
SmallChange 5
Maximum 400
Minimum 0
Orientation Horizontal
TickFrequency 10
TickStyle TopLeft

Name TrackBarV
LargeChange 10
SmallChange 5
Maximum 400
Minimum 0
Orientation Vertical
TickFrequency 10
TickStyle TopLeft

PICTUREBOX

Cantidad 1

Name Foto
Image (Seleccione un archivo de imagen)
SizeMode 2
Size 400,400

NUMERICUPDOWN

Cantidad 2

Name UpDown1
Increment 2
Maximum 400
Minimum 0
TextAlign Center

Name UpDown2
Increment 2

Maximum	400
Minimum	0
TextAlign	Center

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void TrackBarH_Scroll(object sender, EventArgs e)
    {
        //Horizontal
        Foto.Width = TrackBarH.Value;
        MensajeH.Text = Convert.ToString(TrackBarH.Value);
    }

    private void TrackBarV_Scroll(object sender, EventArgs e)
    {
        //Vertical
        Foto.Height = TrackBarV.Value;
        MensajeV.Text = Convert.ToString(TrackBarV.Value);
    }

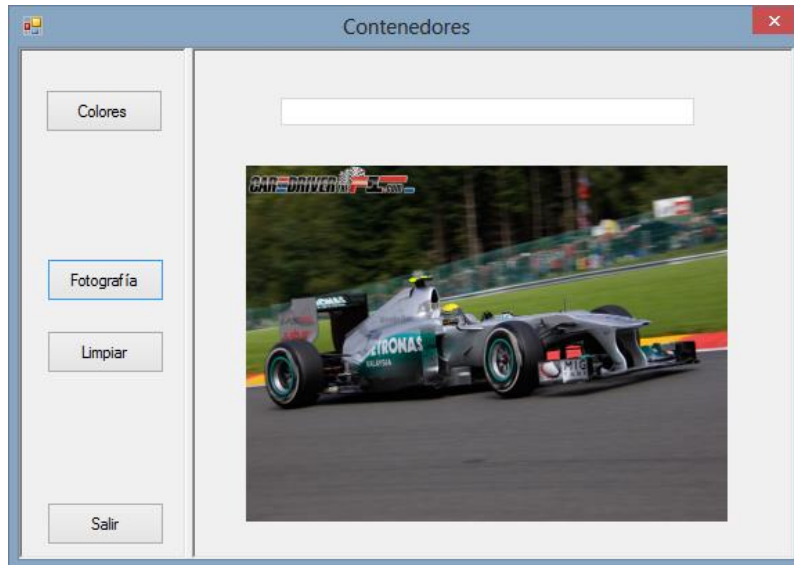
    private void UpDown1_ValueChanged(object sender, EventArgs e)
    {
        //Horizontal
        Foto.Width = Convert.ToInt16(UpDown1.Value);
    }

    private void UpDown2_ValueChanged(object sender, EventArgs e)
    {
        //Vertical
        Foto.Height = Convert.ToInt16(UpDown2.Value);
    }
}
```

5. GRABAR Y EJECUTAR

SPLITCONTAINER

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Contenedores
BackColor	(A su gusto)
Icon	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal

TEXTBOX

Cantidad	1
Name	Tdato
Text	(Vacío)
Enabled	False

BUTTON

Cantidad	3
Name	Boton1
Text	Colores
Name	Boton2
Text	Fotografía
Name	Boton3
Text	Limpiar

Name Boton4
Text Salir

PICTUREBOX

Cantidad 1

Name PictureBox1
Image (Seleccione un archivo de imagen)
SizeMode 2

SPLITCONTAINER

Cantidad 1

Name SplitContainer1
BordeStyle 3

COLORDIALOG

Cantidad 1

Name colorDialog1

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //Colores
        if (colorDialog1.ShowDialog() == DialogResult.OK)
        {
            Tdato.BackColor = colorDialog1.Color;
            SplitContainer1.Panel1.BackColor = colorDialog1.Color;
        }
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        // fotografía
        PictureBox1.Visible = true;
    }

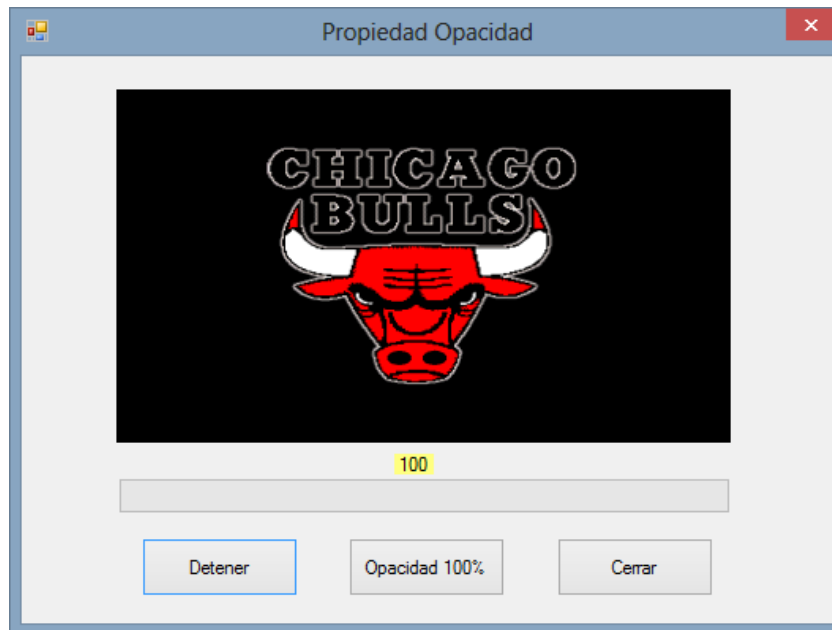
    private void Boton3_Click(object sender, EventArgs e)
    {
        Tdato.BackColor = Color.White;
        PictureBox1.Visible = false;
    }
}
```

```
private void Boton4_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
```

5. GRABAR Y EJECUTAR

PROPIEDAD OPACIDAD

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Propiedad Opacidad
BackColor	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad 3

Name Boton1
Text Detener

Name Boton2
Text Opacidad 100%

Name Boton3
Text Cerrar

TIMER

Cantidad 1

Name Reloj

PICTUREBOX

Cantidad 1

Name Foto1

Image (Seleccione un archivo de imagen)

PROGRESSBAR

Cantidad 1

Name ProgressBar1

LABEL

Cantidad 1

Name Mensaje

BackColor (A su gusto)

4. CODIGO

```
public partial class Form1 : Form
{
    double Aumenta = 0.0;

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        Mensaje.Text = Convert.ToString(this.Opacity * 100);
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        Mensaje.Text = "0";
        ProgressBar1.Value = 0;
        this.Opacity = 0;
        Aumenta = 0;
        Reloj.Enabled = true;
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        Reloj.Enabled = false;
    }

    private void Reloj_Tick(object sender, EventArgs e)
    {
        if (Aumenta > 0.5)
        {
```

```
        Relej.Stop();
        Relej.Enabled = false;
    }
    else
    {
        this.Opacity = 0.5 + Aumenta;
        Aumenta = Aumenta + 0.01;
        Mensaje.Text = Convert.ToString((this.Opacity) * 100);
        ProgressBar1.Value = Convert.ToInt16(Mensaje.Text);
    }
}

private void Boton3_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
```

5. GRABAR Y EJECUTAR

JUEGO DEL 7

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Juego del 7
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	3
Name	Mensaje1
Text	(Vacío)
BackColor	(A su gusto)
BoderStyle	2
Font	(Tamaño 24, Negrita)
TextAlign	MiddleCenter
Name	Mensaje2
Text	(Vacío)
BackColor	(A su gusto)
BoderStyle	2

Font (Tamaño 24, Negrita)
TextAlign MiddleCenter

Name Mensaje3
Text (Vacío)
BackColor (A su gusto)
BoderStyle 2
Font (Tamaño 24, Negrita)
TextAlign MiddleCenter

BUTTON

Cantidad 3

Name Boton1
Text Jugar
BackColor (A su gusto)
Font (Tamaño 24, Negrita)

Name Boton2
Text Terminar
BackColor (A su gusto)
Font (Tamaño 24, Negrita)

Name Boton3
Text Cambiar Foto
BackColor (A su gusto)
Font (Tamaño 8, Negrita)

PICTUREBOX

Cantidad 1

Name Foto1
Image (Seleccione un archivo de imagen)
SizeMode 2
Visible False

4. CODIGO

Incluir la siguiente línea de código en la parte superior de la ventana de edición, en la sección de los “using” : `using System.Media;`

```
public partial class Form1 : Form
{
    Random r = new Random();

    public Form1()
    {
```

```

        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        /*Este juego es similar a las maquinas traga
        monedas si sale un 7 en cualquier
        posicion GANAS */

        Foto1.Visible = false;
        Mensaje1.Text =
        Convert.ToString(Convert.ToInt16(r.Next(0,9)));
        Mensaje2.Text =
        Convert.ToString(Convert.ToInt16(r.Next(0,9)));
        Mensaje3.Text =
        Convert.ToString(Convert.ToInt16(r.Next(0,9)));
        if ((Mensaje1.Text == "7") || (Mensaje2.Text == "7") ||
        (Mensaje3.Text == "7"))
        {
            SystemSounds.Beep.Play(); //Reproduce un sonido
            Foto1.Visible = true;
        }
    }

    private void Boton3_Click(object sender, EventArgs e)
    {
        Foto1.Image =
        System.Drawing.Image.FromFile("C:\\Fotos\\Peques.jpg");
        //Lo que esta entre comillas es la ruta donde se encuentra
        otra Imágen
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}

```

5. GRABAR Y EJECUTAR

CASINO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Casino 2
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

PICTUREBOX

Cantidad 3

Name	Foto1
Image	(Seleccione un archivo de imagen)
BorderStyle	3
SizeMode	2
Visible	True
Size	115, 120

Name	Foto2
Image	(Seleccione un archivo de imagen)
BorderStyle	3
SizeMode	2
Visible	True

Size 115, 120
Name Foto3
Image (Seleccione un archivo de imagen)
BorderStyle 3
SizeMode 2
Visible True
Size 115, 120

BUTTON

Cantidad 2

Name Boton1
Text Jugar
BackColor (A su gusto)
Font (Tamaño 24, Negrita)

Name Boton2
Text Terminar
BackColor (A su gusto)
Font (Tamaño 24, Negrita)

4. CODIGO

```
public partial class Form1 : Form
{
    String[] Vector = new String[3];

    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        byte N1, N2, N3;
        Random rnd = new Random();

        do
            N1 = Convert.ToByte(rnd.Next(1,3));
        while ((N1 < 1) || (N1 > 3));

        do
            N2 = Convert.ToByte(rnd.Next(1,3));
        while ((N2 < 1) || (N2 > 3));

        do
            N3 = Convert.ToByte(rnd.Next(1,3));
        while ((N3 < 1) || (N3 > 3));
    }
}
```

```

Foto1.Image = System.Drawing.Image.FromFile(Vector[N1]);
Foto2.Image = System.Drawing.Image.FromFile(Vector[N2]);
Foto3.Image = System.Drawing.Image.FromFile(Vector[N3]);

if ((N1 == N2) && (N1 == N3))
    MessageBox.Show("FELICITACIONES GANASTE");
else
    MessageBox.Show("LO SIENTO INTENTA NUEVAMENTE");
}

private void Form1_Load(object sender, EventArgs e)
{
    //Coloque entre las comillas la ruta correspondiente a cada
    imagen
    Vector[1] = "C:\\Fotos\\Casino\\Foto1.Ico";
    Vector[2] = "C:\\Fotos\\Casino\\Foto2.Ico";
    Vector[3] = "C:\\Fotos\\Casino\\Foto3.Ico";
}

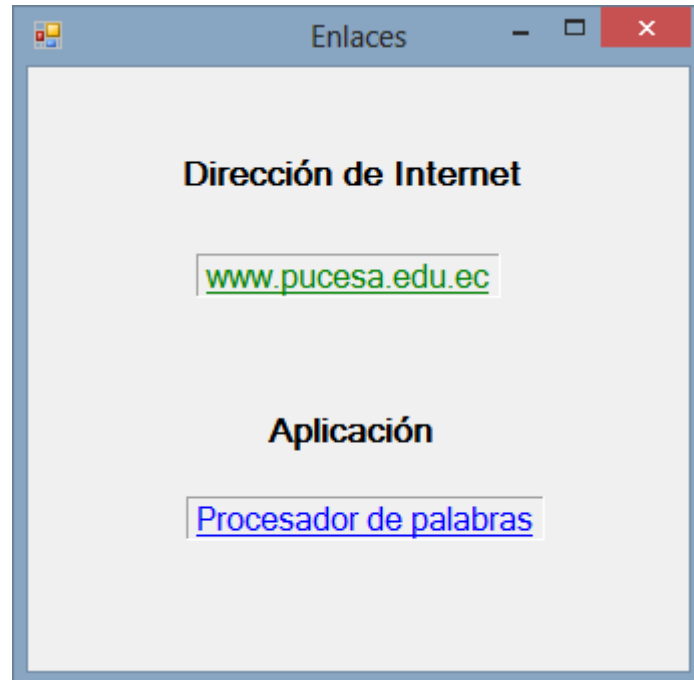
private void Boton2_Click(object sender, EventArgs e)
{
    this.Close();
}
}

```

5. GRABAR Y EJECUTAR

ENLACES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Enlaces
BackColor	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	2
----------	---

Name	Mensaje1
Text	Dirección de Internet
BackColor	(A su gusto)
Font	(Tamaño 12, Negrita)

Name	Mensaje2
Text	Aplicación
BackColor	(A su gusto)
Font	(Tamaño 12, Negrita)

LINKLABEL

Cantidad 2

Name	Enlace1
ActiveLinkColor	(Escoger Rojo)
BorderStyle	Fixed3D
Font	12
LinkColor	(Escoger Verde)
Text	www.pucesa.edu.ec
VisitedLinkColor	(Escoger Negro)

Name	Enlace2
ActiveLinkColor	(Por defecto)
BorderStyle	Fixed3D
Font	12
LinkColor	(Por defecto)
Text	Procesador de Palabras
VisitedLinkColor	(Por defecto)

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

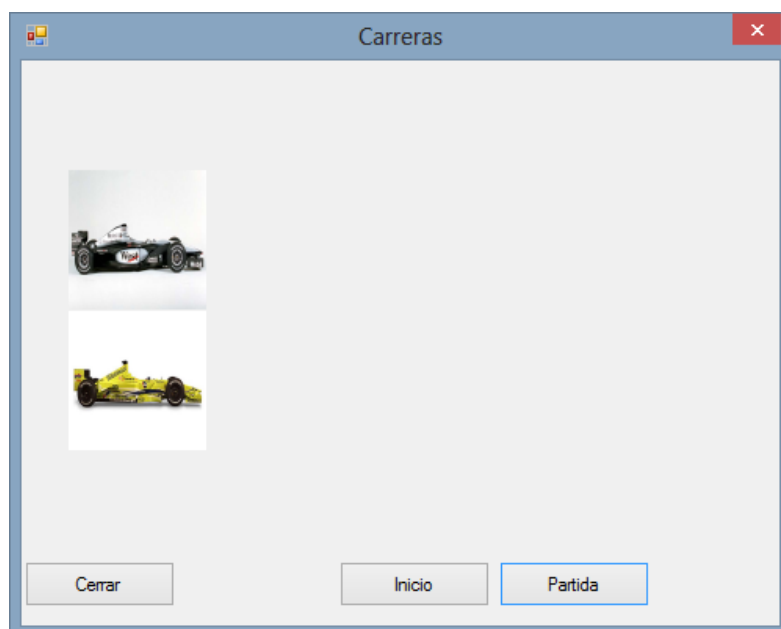
    private void Enlace1_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        Enlace1.LinkVisited = true;
        System.Diagnostics.Process.Start("http://www.pucesa.edu.ec");
        //Tener presente el cambio de colores en este link
    }

    private void linkLabel1_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        Enlace2.LinkVisited = true;
        System.Diagnostics.Process.Start("winword.exe");
    }
}
```

5. GRABAR Y EJECUTAR

CARRERAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Carreras
BackColor	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal
Size	400,500

BUTTON

Cantidad	3
Name	Boton1
Text	Inicio
Name	Boton2
Text	Partida
Name	Boton3
Text	Cerrar

TIMER

Cantidad	1
----------	---

Name Reloj

PICTUREBOX

Cantidad 2

Name Auto1
Image (Seleccione un archivo de imagen)
SizeMode 2

Name Auto2
Image (Seleccione un archivo de imagen)
SizeMode 2

4. CODIGO

```
public partial class Form1 : Form
{
    int Mov;
    int Posi1 = 30, Posi2 = 30;
    Random rnd = new Random();

    public Form1()
    {
        InitializeComponent();
    }

    public void Movimiento()
    {
        Mov = rnd.Next(0,10);
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        Reloj.Enabled = true;
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        Reloj.Enabled = false;
        Auto1.Location = new Point(30, 70);
        Auto2.Location = new Point(30, 160);
        Posi1 = 30;
        Posi2 = 30;
    }

    private void Reloj_Tick(object sender, EventArgs e)
    {
        for (int X = 1; X < 3; X++)
        {
            if ((Posi1 <= 450) && (Posi2 <= 450))
```

```

    {
        Movimiento();
        if ((Mov > 0) && (Mov < 10))
            if (X == 1)
            {
                Posi2 = Posi2 + Convert.ToInt16(Mov);
                Auto2.Location = new Point(Posi2, 160);
            }
            else
            {
                Posi1 = Posi1 + Convert.ToInt16(Mov);
                Auto1.Location = new Point(Posi1, 70);
            }
        }
    else
    {
        Reloj.Stop();
        MessageBox.Show("Ganó el auto: " + X);
        Posi1 = 30;
        Posi2 = 30;
    }
}

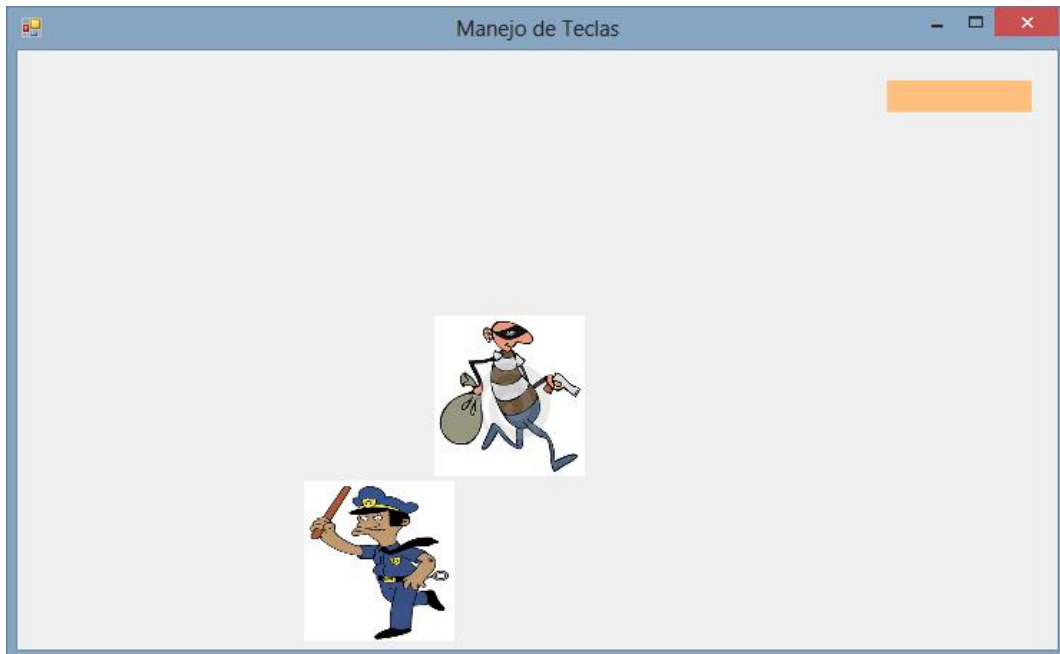
private void Boton3_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}

```

5. GRABAR Y EJECUTAR

MANEJO DE TECLAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Manejo de Teclas
BackColor	(A su gusto)
WindowState	Maximized

LABEL

Cantidad	1
Name	Mensaje1
BackColor	(A su gusto)

PICTUREBOX

Cantidad	2
Name	Foto1
Image	(Seleccione un archivo de imagen) Ejemplo POLICÍA
SizeMode	2
Name	Foto2
Image	(Seleccione un archivo de imagen) Ejemplo LADRÓN
SizeMode	2

4. CODIGO

```
public partial class Form1 : Form
{
    int Fi = 200;
    int Co = 300;
    int Puntos = 0;
    Random rnd = new Random();
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        this.Width = 700;
        this.Height = 600;
    }

    private void ubica()
    {
        int F, C, Ancho, Largo;
        F = Foto2.Left;
        C = Foto2.Top;
        Ancho = Foto2.Width;
        Largo = Foto2.Height;
        if ((Foto1.Left) >= F && (Foto1.Left <= F + Ancho) &&
(Foto1.Top) >= C && (Foto1.Top <= C + Largo))
        {
            Puntos += 1;
            Mensaje1.Text = Convert.ToString(Puntos);
            Foto2.Location = new Point(GENERA(450), GENERA(700));
        }
    }

    private int GENERA (int Aux)
    {
        int N2;
        do
            N2 = (rnd.Next()*1000);
        while ((N2 < 1) || (N2 > Aux));
        return (N2);
    }

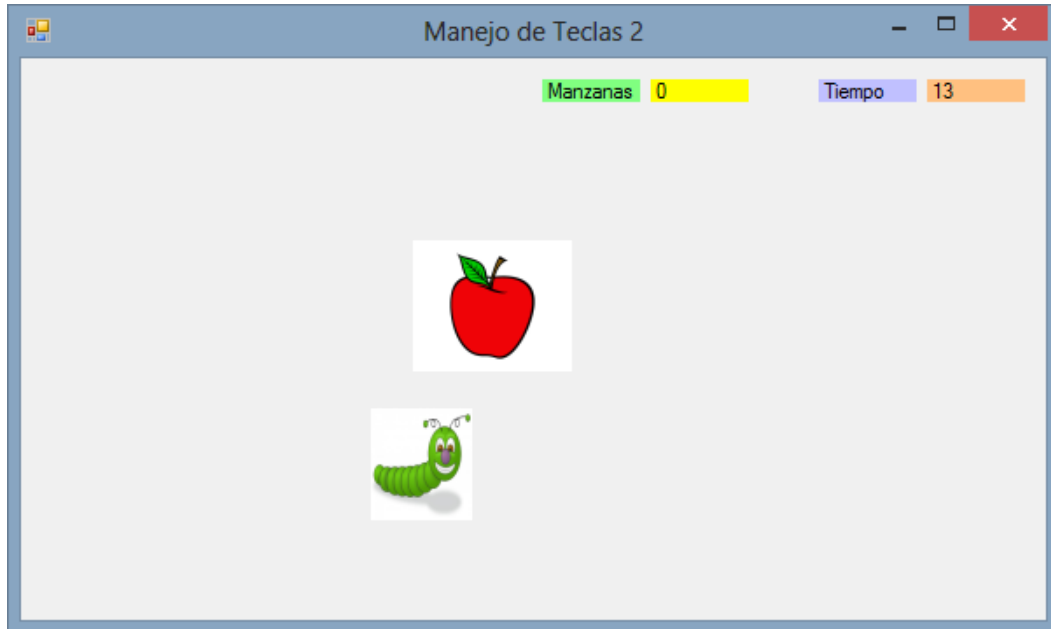
    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        switch (Convert.ToInt32(e.KeyCode))
        {
            case 37 : Fi = Fi - 3; break; // IZQUIERDA;
            case 39 : Fi = Fi + 3; break; // DERECHA;
            case 38 : Co = Co - 3; break; // ARRIBA;
            case 40: Co = Co + 3; break; // ABAJO;
        }
    }
}
```

```
        Foto1.Location = new Point(Fi, Co);  
        ubica();  
    }  
}
```

5. GRABAR Y EJECUTAR

MANEJO DE TECLAS 2

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Manejo de Teclas
BackColor	(A su gusto)
WindowState	Maximized

LABEL

Cantidad	4
Name	Mensaje1
BackColor	(A su gusto)
Text	Manzanas
Name	Mensaje2
BackColor	(A su gusto)
Text	Vacio
Name	Mensaje3
BackColor	(A su gusto)
Text	Tiempo
Name	Mensaje4
BackColor	(A su gusto)
Text	Vacío

PICTUREBOX

Cantidad 2

Name ImagenAlimento
Image (Seleccione un archivo de imagen) Ejemplo MANZANA
SizeMode 2

Name ImagenSnake
Image (Seleccione un archivo de imagen) Ejemplo GUSANO
SizeMode 2

TIMER

Cantidad 1

Name Timer1
Interval 1000
Enabled True

4. CODIGO

```
public partial class Form1 : Form
{
    public int Respuesta = 0;
    public int A = 20;
    Random Rnd = new Random();

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        switch (e.KeyCode)
        {
            case Keys.Up:
                this.ImagenSnake.Top -= 10;
                break;
            case Keys.Down:
                this.ImagenSnake.Top += 10;
                break;
            case Keys.Left:
                this.ImagenSnake.Left -= 10;
                break;
            case Keys.Right:
                this.ImagenSnake.Left += 10;
                break;
        }
    }
}
```

```

if ((ImagenSnake.Top > ImagenAlimento.Top) &&
    (ImagenSnake.Top < (ImagenAlimento.Top +
ImagenAlimento.Height)))
{
    if ((ImagenSnake.Left > ImagenAlimento.Left &&
        ImagenSnake.Left < (ImagenAlimento.Left +
ImagenAlimento.Width)))
    {
        double Alto, Ancho;
        do
            Alto = (Rnd.Next() * 1000);
        while ((Alto < 1) || (Alto > 300));

        do
            Ancho = (Rnd.Next() * 1000);
        while ((Ancho < 1) || (Ancho > 300));

        ImagenAlimento.Top = Convert.ToInt16(Alto);
        ImagenAlimento.Left = Convert.ToInt16(Ancho);
        Respuesta = Respuesta + 1;
    }
}

Mensaje2.Text = Convert.ToString(Respuesta);
if (A <= 0)
{
    ImagenSnake.Visible = false;
    MessageBox.Show("HAS COMIDO " + Respuesta + "
MANZANAS EN 20 Segs", "RESULTADO!",
    MessageBoxButtons.OK);
    timer1.Enabled = false;
    timer1.Stop();
}
}

private void Form1_Load(object sender, EventArgs e)
{
    ImagenSnake.Top = 200;
    ImagenSnake.Left = 200;
    double Alto, Ancho;
    do
        Alto = (Rnd.Next() * 1000);
    while ((Alto < 1) || (Alto > 300));
    do
        Ancho = (Rnd.Next() * 1000);
    while ((Ancho < 1) || (Ancho > 300));
    ImagenAlimento.Top = Convert.ToInt16(Alto);
    ImagenAlimento.Left = Convert.ToInt16(Ancho);
}

private void timer1_Tick(object sender, EventArgs e)
{
    if ((A <= 20) && (A != 0))
        A = A - 1;
}

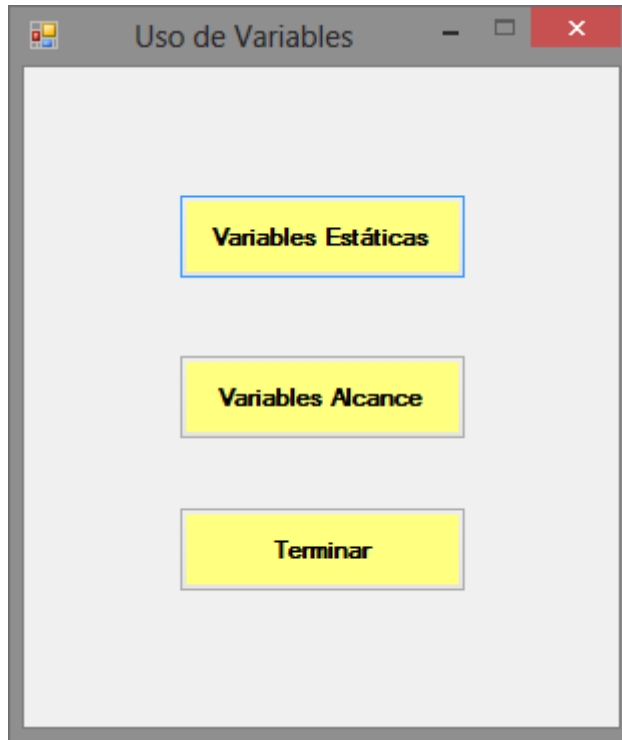
```

```
        Mensaje4.Text = Convert.ToString(A);  
    }  
}
```

5. GRABAR Y EJECUTAR

USO DE VARIABLES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Uso de Variables
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	3
Name	Boton1
Text	Variables Estáticas
BackColor	(A su gusto)
Font	(Tamaño 8, Negrita)
Name	Boton2
Text	Variables Alcance
BackColor	(A su gusto)

Font	(Tamaño 8, Negrita)
Name	Boton3
Text	Terminar
BackColor	(A su gusto)
Font	(Tamaño 8, Negrita)

4. CODIGO

Public Class Form1

```
public partial class Form1 : Form
{
    static int Pulsa = 0;

    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        Pulsa += 1;
        MessageBox.Show("Número de veces que dio clic: " + Pulsa);
        //Fijarse en el tipo de dato Static que no pierde su valor
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        String MiVar; //Variable a nivel de procedimiento
        int Pulsa1 = 0; //Variable a nivel de procedimiento
        MiVar = "Estoy practicando alcance de Variables. La
variable local mantiene su valor: ";
        Pulsa1 += 1;
        MessageBox.Show(MiVar + Pulsa1);
    }

    private void Boton3_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```

5. GRABAR Y EJECUTAR

DECLARAR ESTRUCTURAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



The image shows a screenshot of a Windows application window titled "Declarar Estructuras". The window contains a form with a header "Fábrica de Autos" in a light orange box. Below the header is a yellow button labeled "Mostrar". Underneath the button are four input fields, each with a label to its left: "Marca", "Modelo", "Precio", and "Fecha". The labels are in a light green box, and the input fields are white with a thin border.

3. Componentes

FORM

Name	Form1
Text	Declarar Estructuras
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	1
Name	Boton1
Text	Mostrar
BackColor	(A su gusto)
Font	(Tamaño 12, Negrita)

LABEL

Cantidad	5
----------	---

Name Mensaje1
Text Fábrica de Autos
BackColor (A su gusto)
Font (Tamaño 24, Negrita)

Name Mensaje2
Text Marca
BackColor (A su gusto)
Font (A su gusto)

Name Mensaje3
Text Modelo
BackColor (A su gusto)
Font (A su gusto)

Name Mensaje4
Text Precio
BackColor (A su gusto)
Font (A su gusto)

Name Mensaje5
Text Fecha
BackColor (A su gusto)
Font (A su gusto)

TEXTBOX

Cantidad 4

Name Tdato1
Text (Vacío)

Name Tdato2
Text (Vacío)

Name Tdato3
Text (Vacío)

Name Tdato4
Text (Vacío)

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    //Declaración de la estructura
    public struct Carro
```

```

{
    //Se declaran public para poder acceder a los campos de la
    estructura
    public String marca;
    public String modelo;
    public int precio;
    public DateTime fechafabrica;
}

private void Boton1_Click(object sender, EventArgs e)
{
    Carro Mivarcarro;
    // Se asigna a miembros de estructura
    Mivarcarro.marca = "Nissan";
    Mivarcarro.modelo = "Cabina simple";
    Mivarcarro.precio = 8500;
    Mivarcarro.fechafabrica = new DateTime (2001, 11, 4, 15, 0,
    0);

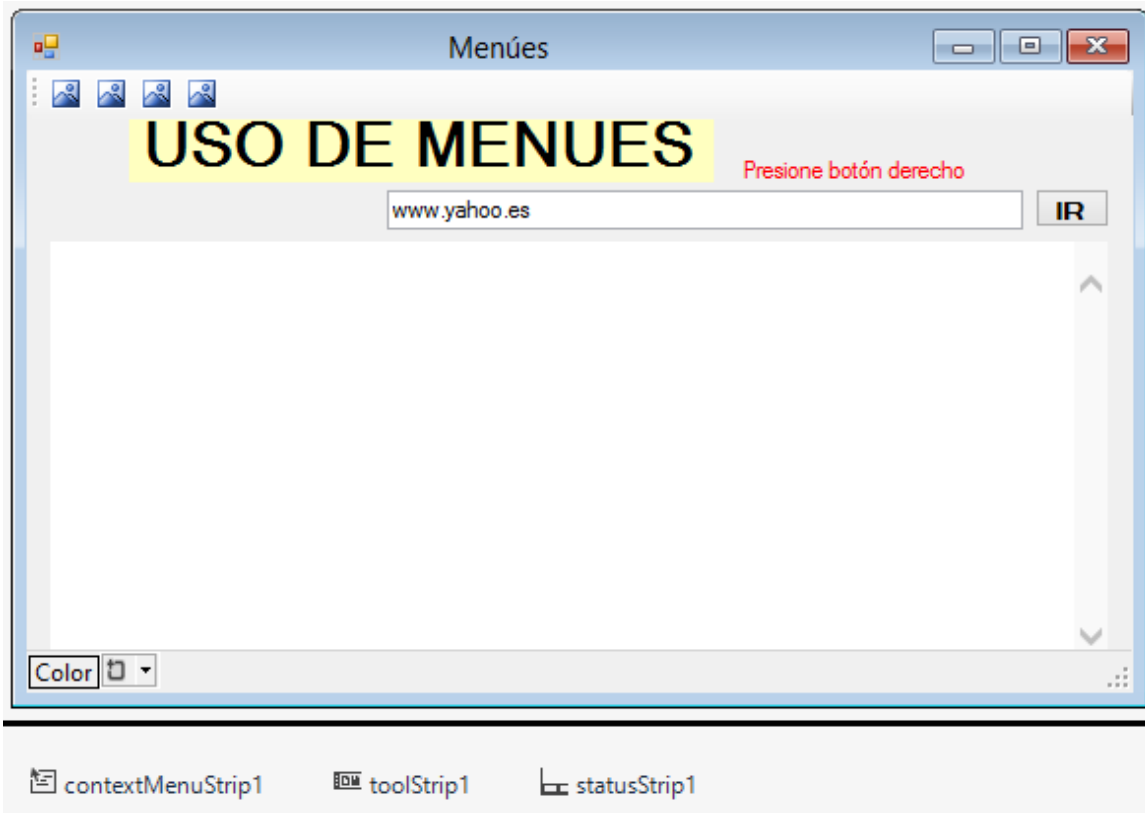
    // Se asigna a Textbox desde miembros de estructura
    Tdato1.Text = Mivarcarro.marca;
    Tdato2.Text = Mivarcarro.modelo;
    Tdato3.Text = Convert.ToString(Mivarcarro.precio);
    Tdato4.Text = Convert.ToString(Mivarcarro.fechafabrica);
}

```

5. GRABAR Y EJECUTAR

BARRA DE HERRAMIENTAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Menús
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	1
Name	Boton1
Text	IR
BackColor	(A su gusto)
Font	(Tamaño 10, Negrita)

LABEL

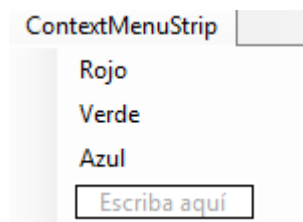
Cantidad	2
Name	Mensaje1
Text	USO DE MENÚES
Backcolor	(A su gusto)
Font	(Tamaño 24, Negrita)
Name	Mensaje2
Text	Presione Botón derecho
Forecolor	(A su gusto)

TEXTBOX

Cantidad	1
Name	Tdato1
Text	www.yahoo.es

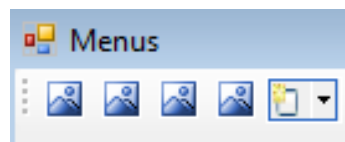
CONTEXTMENUSTRIP

Cantidad	1
Name	ContextMenuStrip1
Contenido	(Diseñar Tres opciones con elemento MenúItem)



TOOLSTRIP

Cantidad	1
Name	ToolStrip1
Contenido	(Diseñar cuatro opciones con elemento Botón)



(Propiedad Text de cada botón es Rojo, Verde, Azul, Abrir Navegador)

WEBBROWSER

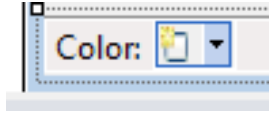
Cantidad	1
Name	WebBrowser1

STATUSSTRIP

Cantidad 1

Name StatusStrip1

Contenido (Diseñar una opción con elemento StatusLabel)



(Propiedad Text escriba **Color:**)

4. CODIGO

//Debe incluir la línea using System.Diagnostics; al inicio del código.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void toolStripButton1_Click(object sender, EventArgs e)
    {
        //Rojo
        Mensaje1.BackColor = Color.Red;
        toolStripStatusLabel1.Text = "Color: ROJO";
    }

    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        //Verde
        Mensaje1.BackColor = Color.Green;
        toolStripStatusLabel1.Text = "Color: VERDE";
    }

    private void toolStripButton3_Click(object sender, EventArgs e)
    {
        //Azul
        Mensaje1.BackColor = Color.Blue;
        toolStripStatusLabel1.Text = "Color: AZUL";
    }

    private void rojoToolStripMenuItem_Click(object sender,
        EventArgs e)
    {
        // Rojo
        Mensaje1.BackColor = Color.Red;
    }

    private void verdeToolStripMenuItem_Click(object sender,
        EventArgs e)
    {

```

```

{
    //Verde
    Mensaje1.BackColor = Color.Green;
}

private void azulToolStripMenuItem_Click(object sender,
EventArgs e)
{
    //Azul
    Mensaje1.BackColor = Color.Blue;
}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    //Abrir Navegador
    Process.Start("www.yahoo.es");
}

private void Boton1_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate(Tdato1.Text);
    webBrowser1.Visible = true;
    this.Text = Tdato1.Text;
}

private void Form1_Load(object sender, EventArgs e)
{
    //Espacio de la Página Web Oculta
    webBrowser1.Visible = false;
}

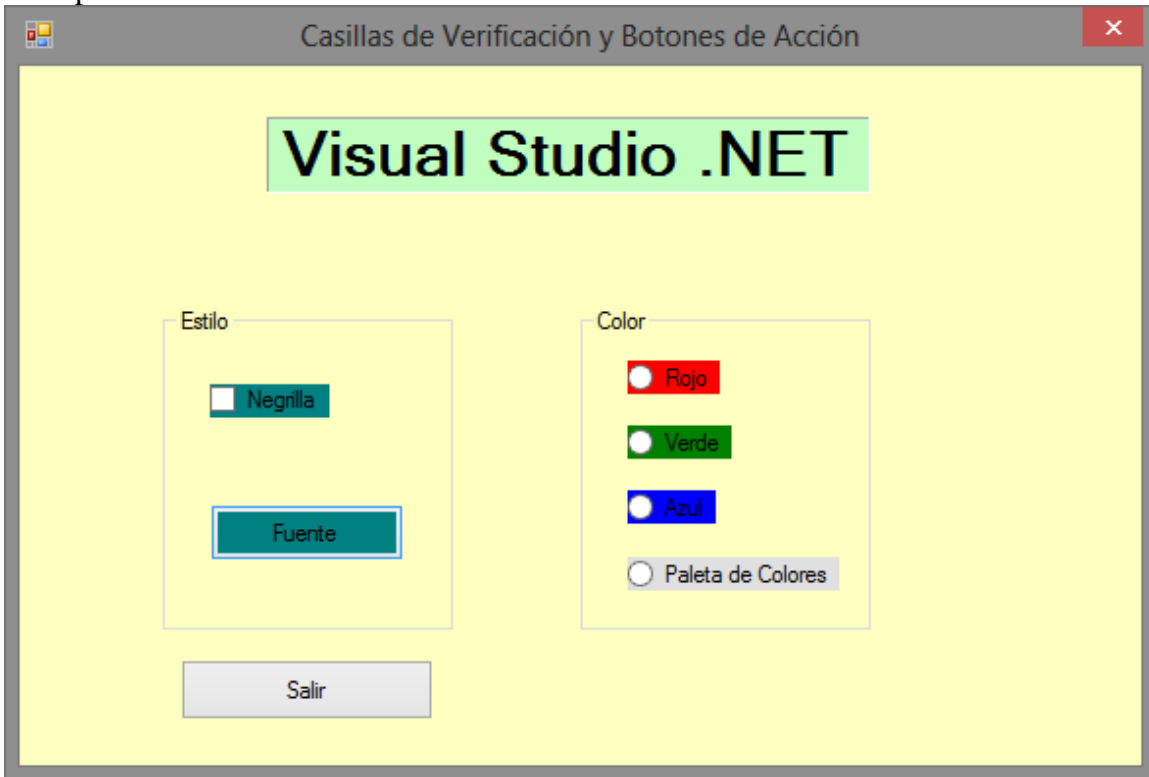
//COMPLETE EL TOOLSTRIP CON BOTONES PARA EL NAVEGADOR
//RETROCEDER, AVANZAR, DETENER, REFRESCAR
}

```

5. GRABAR Y EJECUTAR

CASILLAS DE VERIFICACIÓN Y BOTONES DE ACCIÓN

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Casillas de Verificación y Botones de Acción
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	1
Name	Mensaje1
BackColor	(A su gusto)
BorderStyle	3
Text	Visual Studio .NET
Font	(Negrita, Tamaño 24)
TextAlign	MiddleCenter

GROUPBOX

Cantidad	2
----------	---

Name Contenedor1
BackColor (A su gusto)
Text Estilo

Name Contenedor2
BackColor (A su gusto)
Text Color

BUTTON

Cantidad 2

Name Boton1
BackColor (A su gusto)
Text Salir

Name Boton2
BackColor (A su gusto)
Text Fuente

CHECKBOX

Cantidad 1

Name Casilla1
BackColor (A su gusto)
Text Negrilla

RADIOBUTTON

Cantidad 4

Name Opcion1
BackColor (A su gusto)
Text Rojo

Name Opcion2
BackColor (A su gusto)
Text Verde

Name Opcion3
BackColor (A su gusto)
Text Azul

Name Opcion4
BackColor (A su gusto)
Text Paleta de Colores

COLORDIALOG

Cantidad 1

Name DialogoColor

FONTDIALOG

Cantidad 1

Name DialogoFuente

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Opcion1_CheckedChanged(object sender, EventArgs e)
    {
        //ROJO
        Mensaje1.BackColor = Color.Red;
    }

    private void Opcion2_CheckedChanged(object sender, EventArgs e)
    {
        //VERDE
        Mensaje1.BackColor = Color.Green;
    }

    private void Opcion3_CheckedChanged(object sender, EventArgs e)
    {
        //AZUL
        Mensaje1.BackColor = Color.Blue;
    }

    private void Opcion4_CheckedChanged(object sender, EventArgs e)
    {
        //PALETA DE COLORES
        if (DialogoColor.ShowDialog() ==
            System.Windows.Forms.DialogResult.OK)
            Mensaje1.BackColor = DialogoColor.Color;
    }

    private void Casilla1_CheckedChanged(object sender, EventArgs
e)
    {
        //NEGRILLA
        if (Casilla1.Checked)
            Mensaje1.Font = new Font("Microsoft Sans Serif", 24,
                FontStyle.Bold);
        else
            Mensaje1.Font = new Font("Microsoft Sans Serif", 24,
                FontStyle.Regular);
    }

    private void Boton2_Click(object sender, EventArgs e)
```

```

    {
        //DIALOGO FUENTE
        if (DialogoFuente.ShowDialog() ==
            System.Windows.Forms.DialogResult.OK)
            Mensaje1.Font = DialogoFuente.Font;

        //Otra forma sin poner el objeto a probar
        //Dim FD As New FontDialog
        //FD.ShowDialog()
        //Mensaje1.Font = FD.Font
    }

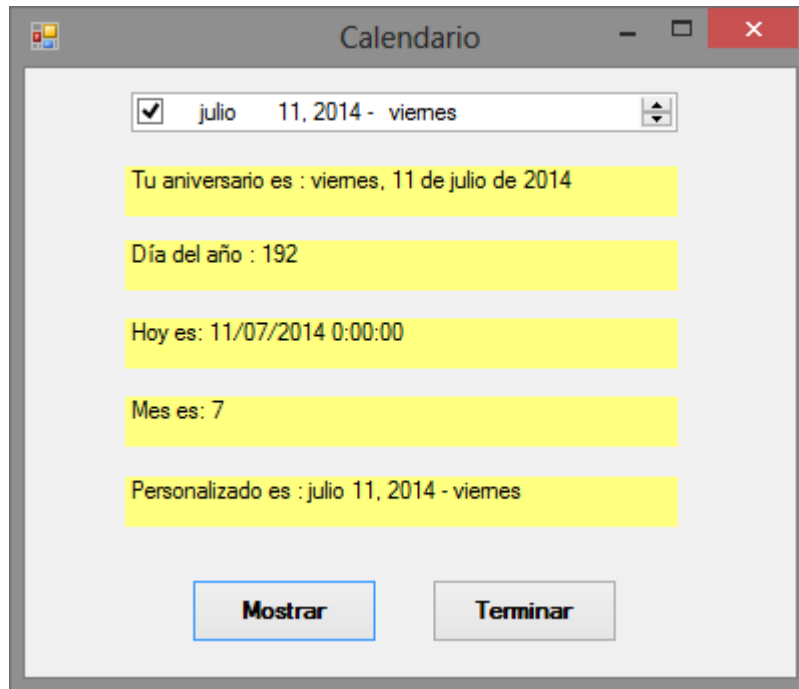
    private void Boton1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

5. GRABAR Y EJECUTAR

CALENDARIO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Calendario
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

DATETIMEPICKER

Cantidad	1
Name	Calendario
Format	Long (En que formato muestra la información)
Value	(Corresponde a la fecha que desea que aparezca)

LABEL

Cantidad	5
Name	Mensaje1

Text (Vacío)
BackColor (A su gusto)

Name Mensaje2
Text (Vacío)
BackColor (A su gusto)

Name Mensaje3
Text (Vacío)
BackColor (A su gusto)

Name Mensaje4
Text (Vacío)
BackColor (A su gusto)

Name Mensaje5
Text (Vacío)
BackColor (A su gusto)

BUTTON

Cantidad 2

Name Boton1
Text Mostrar
BackColor (A su gusto)
Font (Tamaño 8, Negrita)

Name Boton2
Text Terminar
BackColor (A su gusto)
Font (Tamaño 8, Negrita)

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        Mensaje1.Text = "Tu aniversario es : " + Calendario.Text;

        //Día del año a partir del 1 de Enero
        Mensaje2.Text = "Día del año : " +
        Calendario.Value.DayOfYear;

        //Impresión en diferentes formatos la fecha actual
        Mensaje3.Text = "Hoy es: " + Calendario.Value.Date;
    }
}
```

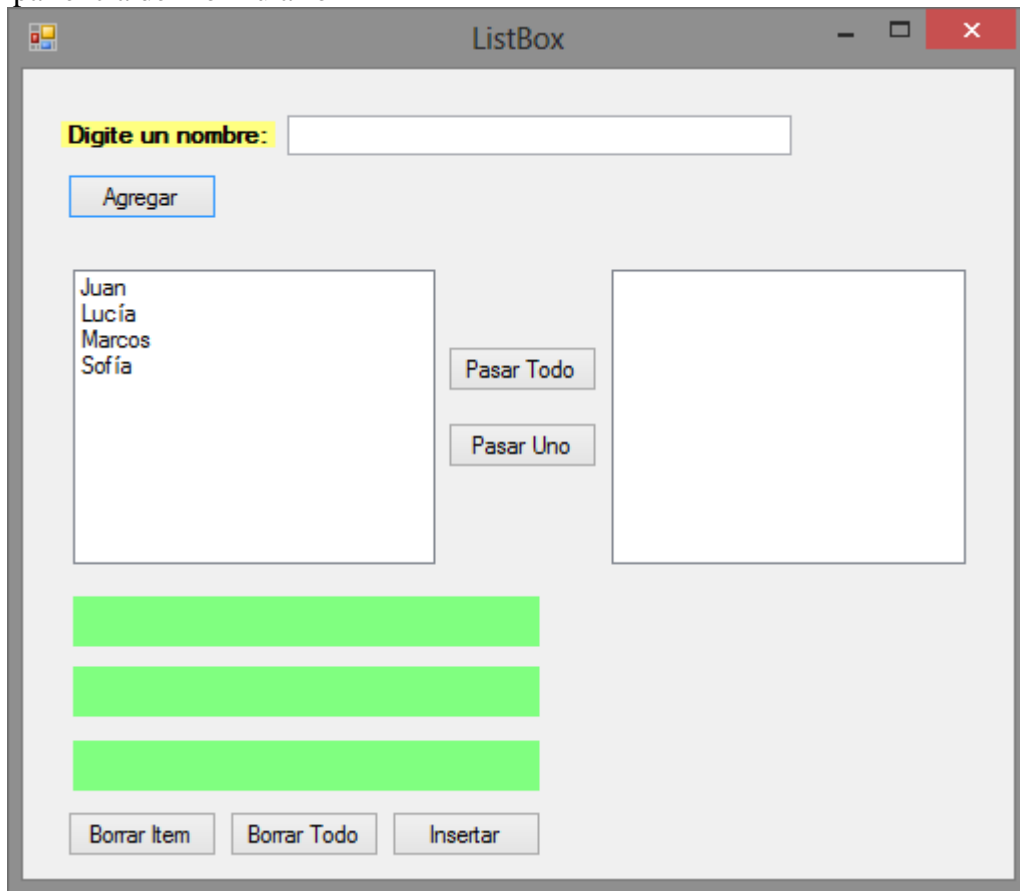
```
        Mensaje4.Text = "Mes es: " + Calendario.Value.Date.Month;
        Calendario.CustomFormat = "MMMM dd, yyyy - dddd";
        Calendario.Format = DateTimePickerFormat.Custom;
        Mensaje5.Text = "Personalizado es : " + Calendario.Text;
        Calendario.ShowCheckBox = true;
        Calendario.ShowUpDown = true;
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

5. GRABAR Y EJECUTAR

USO DE LISTBOX

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	ListBox
BackColor	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	6
Name	Boton1
Text	Agregar
BackColor	(A su gusto)
Name	Boton2
Text	Borrar Item
BackColor	(A su gusto)

Name Boton3
Text Borrar Todo
BackColor (A su gusto)

Name Boton4
Text Insertar
BackColor (A su gusto)

Name Boton5
Text Pasar Todo
BackColor (A su gusto)

Name Boton6
Text Pasar Uno
BackColor (A su gusto)

LABEL

Cantidad 4

Name Mensaje1
Text Digite un Nombre
BackColor (A su gusto)

Name Mensaje2
Text (Vacío)
BackColor (A su gusto)

Name Mensaje3
Text (Vacío)
BackColor (A su gusto)

Name Mensaje4
Text (Vacío)
BackColor (A su gusto)

TEXTBOX

Cantidad 1

Name Tdato
Text (Vacío)

LISTBOX

Cantidad 2

Name Lista1
BackColor (A su gusto)

Name Lista2
Sorted True

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        Lista1.Items.Add("Juan");
        Lista1.Items.Add("Lucía");
        Lista1.Items.Add("Marcos");
        Lista1.Items.Add("Sofía");
    }

    private void Lista1_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        Mensaje2.Text = "Posición : " + Lista1.SelectedIndex;
        Mensaje3.Text = "Nombre : " + Lista1.SelectedItem;
        Mensaje4.Text = "Elementos: " + Lista1.Items.Count;
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        Lista1.Items.Add(Tdato.Text);
        Tdato.Text = "";
        Tdato.Focus();
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        try
        {
            Lista1.Items.RemoveAt(Lista1.SelectedIndex);
        }
        catch (Exception ex)
        {
            MessageBox.Show("Primero seleccione un Nombre");
        }
    }

    private void Boton3_Click(object sender, EventArgs e)
    {
        Lista1.Items.Clear();
    }

    private void Boton4_Click(object sender, EventArgs e)
    {
        Lista1.Items.Insert(2, "Darío");
    }
}
```

```

private void Boton5_Click(object sender, EventArgs e)
{
    byte Posi;
    for (Posi = 0; Posi <= Lista1.Items.Count - 1; Posi++)
    {
        Lista1.SetSelected(Posi, true);
        Lista2.Items.Add(Lista1.SelectedItem);
    }
}

private void Boton6_Click(object sender, EventArgs e)
{
    try
    {
        Lista2.Items.Add(Lista1.SelectedItem);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Primero seleccione un Nombre");
    }
}
}

```

5. GRABAR Y EJECUTAR

LISTBOX PRÁCTICA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	ListBox Práctica
BackColor	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	1
Name	Mensaje1
Text	Número
BackColor	(A su gusto)

TEXTBOX

Cantidad	1
Name	Tdato
Text	8

LISTBOX

Cantidad 1
Name Lista1
BackColor (A su gusto)

GROUPBOX

Cantidad 1
Name Grupo
BackColor (A su gusto)
Text Procesos

RADIOBUTTON (Ojo poner directamente dentro de GroupBox)

Cantidad 4
Name Opcion1
BackColor (A su gusto)
Checked True
Text Tabla de Multiplicar
Name Opcion2
BackColor (A su gusto)
Text Serie Fibonacci
Name Opcion3
BackColor (A su gusto)
Text Números Aleatorios
Name Opcion4
BackColor (A su gusto)
Text Factores Primos

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Opcion1_CheckedChanged(object sender, EventArgs e)
    {
        //TABLA DE MULTIPLICAR
        int NUM, I, RES;
        NUM = Convert.ToInt16(Tdato.Text);
        Lista1.Items.Clear();
        for(I=1; I<=12; I++)
        {
```

```

        RES = NUM * I;
        Lista1.Items.Add(NUM + " * " + I + " = " + RES);
    }
}

private void Opcion2_CheckedChanged(object sender, EventArgs e)
{
    //SERIE DE FIBONACCI
    int A, B, AUX, X, NUM;
    NUM = Convert.ToInt16(Tdato.Text);
    A = 1;
    B = 1;
    Lista1.Items.Clear();
    Lista1.Items.Add(A);
    Lista1.Items.Add(B);
    for (X = 3; X <= NUM; X++)
    {
        AUX = A + B;
        Lista1.Items.Add(AUX);
        A = B;
        B = AUX;
    }
}

private void Opcion3_CheckedChanged(object sender, EventArgs e)
{
    //NUMEROS ALEATORIOS
    int NUM, X, AUX;
    Random Rnd = new Random();
    NUM = Convert.ToInt16(Tdato.Text);
    Lista1.Items.Clear();
    for(X=1;X<=NUM;X++)
    {
        AUX = Rnd.Next() * 1000;
        Lista1.Items.Add(AUX);
    }
}

private void Opcion4_CheckedChanged(object sender, EventArgs e)
{
    //FACTORES PRIMOS DE UN NÚMERO
    int NUM, X, AUX, RESI;
    NUM = Convert.ToInt16(Tdato.Text);
    X = 2;
    Lista1.Items.Clear();
    while (NUM != 1)
    {
        RESI = NUM % X;
        AUX = NUM / X;
        if (RESI == 0)
        {
            Lista1.Items.Add(X);
            NUM = AUX;
        }
    }
}

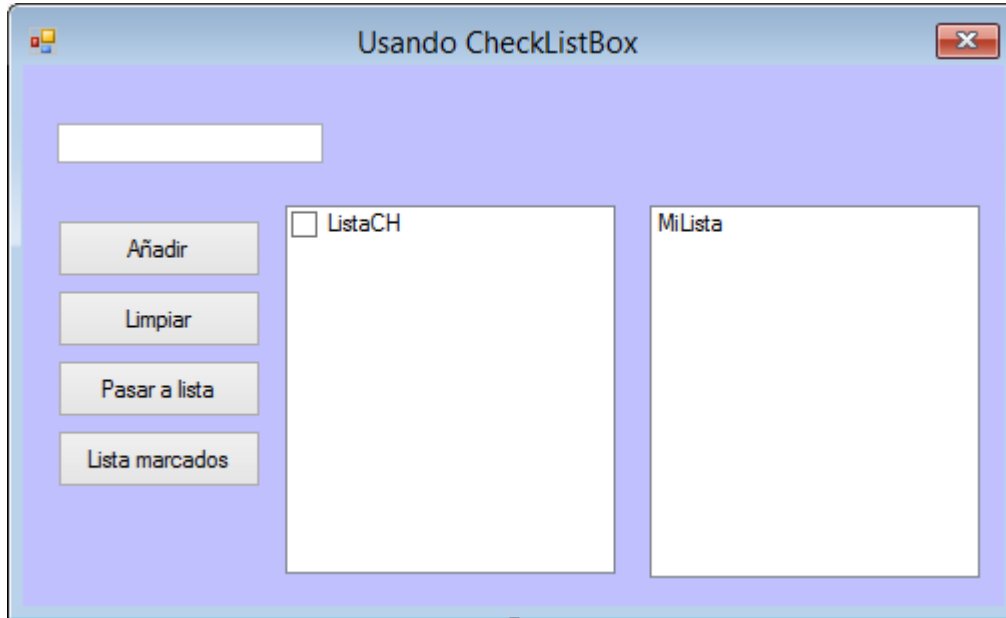
```

```
        else
            X += 1;
    }
}
}
```

5. GRABAR Y EJECUTAR

USANDO CHECKLISTBOX

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Usando CheckListBox
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

TEXTBOX

Cantidad	1
Name	Tdato
Text	(Vacío)

CHECKEDLISTBOX

Cantidad	1
Name	ListaCh
Sorted	True

LISTBOX

Cantidad	1
----------	---

Name MiLista
BackColor (A su gusto)

BUTTON

Cantidad 4

Name Boton1
BackColor (A su gusto)
Text Añadir

Name Boton2
BackColor (A su gusto)
Text Limpiar

Name Boton3
BackColor (A su gusto)
Text Pasar a la Lista

Name Boton4
BackColor (A su gusto)
Text Listar Marcados

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        //AÑADIMOS NOMBRES POR DEFECTO
        ListaCh.Items.Add("Toby");
        ListaCh.SetItemChecked(0, true);
        ListaCh.Items.Add("Bobby");
        ListaCh.SetItemChecked(1, true);
        Tdato.Focus();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        if (Tdato.Text != "")
        {
            //VERIFICA SI YA EXISTE EL TEXTO PARA NO REPETIR
            if (ListaCh.CheckedItems.Contains(Tdato.Text) == false)
                ListaCh.Items.Add(Tdato.Text, CheckState.Checked);
            else
                MessageBox.Show("Ya Existe Marcado");
        }
    }
}
```

```

        Tdato.Text = "";
    }
    Tdato.Focus();
}

private void Boton2_Click(object sender, EventArgs e)
{
    //LIMPIAR MARCADOS
    int n;
    for (n = 0; n <= ListaCh.Items.Count - 1; n++)
        ListaCh.SetItemChecked(n, false);
}

private void Boton3_Click(object sender, EventArgs e)
{
    //PASA ITEMS MARCADOS DEL CHECKEDLISTBOX AL LISTBOX
    MiLista.Items.Clear();
    int i;
    for (i = 0; i <= ListaCh.CheckedItems.Count - 1; i++)
        MiLista.Items.Add(ListaCh.CheckedItems[i].ToString());
}

private void Boton4_Click(object sender, EventArgs e)
{
    //LISTAR MARCADOS
    int I;
    String Valor = "";
    String Dato;
    for (I = 0; I <= this.ListaCh.CheckedItems.Count - 1; I++)
    {
        Dato = ListaCh.CheckedItems[I].ToString();
        Valor = Valor + Convert.ToChar(13) + Dato;
    }
    MessageBox.Show(Valor);
}

private void ListaCh_ItemCheck(object sender,
ItemCheckEventArgs e)
{
    //validar el evento de cuando un item es marcado o no
    if (e.NewValue == CheckState.Unchecked)
        if (ListaCh.CheckedItems.Count == 1)
            MessageBox.Show("Todavía existen ítems activos");
}
}

```

5. GRABAR Y EJECUTAR

USO DE LISTVIEW BÁSICO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

Uso de ListView

Nombre

Teléfono

Ciudad

Estudia

Nombre	Teléfono	Ciudad	Estudia	
--------	----------	--------	---------	--

3. Componentes

FORM

Name	Form1
Text	Uso de ListView
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	4
Name	Mensaje1
Text	Nombre
BackColor	(A su gusto)
Name	Mensaje2
Text	Teléfono
BackColor	(A su gusto)
Name	Mensaje3

Text Ciudad
BackColor (A su gusto)

Name Mensaje4
Text Estudia
BackColor (A su gusto)

TEXTBOX

Cantidad 2

Name Tdato1
Text (Vacío)

Name Tdato2
Text (Vacío)

COMBOBOX

Cantidad 1

Name ComboBox1
Items Ambato
Baños
Pelileo
Quero
Latacunga
Sorted True

CHECKBOX

Cantidad 1

Name CheckBox1
Text (Vacío)
Checked True
RigthToLeft Yes

BUTTON

Cantidad 2

Name Boton1
Text Añadir
BackColor (A su gusto)

Name Boton2
Text Salir
BackColor (A su gusto)

LISTVIEW

Cantidad 1

Name ListView1

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        //SE DEFINE QUE COLUMNAS TENDRÁ LA MATRIZ
        this.ListView1.Columns.Add("Nombre");
        this.ListView1.Columns.Add("Teléfono");
        this.ListView1.Columns.Add("Ciudad");
        this.ListView1.Columns.Add("Estudia");
        this.ListView1.View = View.Details;
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //PASAR LA INFORMACIÓN AL LISTVIEW
        if (Tdato1.Text.Trim() == "")
            return;
        String VarNombre = this.Tdato1.Text;
        int VarTelefono = Convert.ToInt32(this.Tdato2.Text);
        String VarCiudad =
        Convert.ToString(this.Combo1.SelectedItem);
        String VarEstudia = this.CheckBox1.Checked ? "Si": "No";
        ListViewItem I;
        I = this.ListView1.Items.Add(VarNombre);
        I.SubItems.Add(Convert.ToString(VarTelefono));
        I.SubItems.Add(VarCiudad);
        I.SubItems.Add(VarEstudia);
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        //Cerrar
        Application.Exit();
    }
}
```

5. GRABAR Y EJECUTAR

LISTVIEW CON ARCHIVO PLANO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

The screenshot shows a window titled "Ficha" with a close button in the top right corner. The form contains the following elements:

- Empleado:** A text input field.
- Hijos:** A numeric spinner control showing the value 0.
- Estado Civil:** A dropdown menu.
- Estudia:** A radio button group with options "Sí" and "No", where "No" is selected.
- Buttons:** Four buttons labeled "Añadir", "Nuevo", "Reporte", and "Salir".
- Table:** A table with 5 columns: "Empleado", "Hijos", "Estado Civil", "Estudia", and an empty column. The table has 5 rows.

3. Componentes

FORM

Name Form1
Text Ficha
BackColor (A su gusto)
MaximizeBox False
MinimizeBox False
StartPosition CenterScreen
WindowState Normal

LABEL

Cantidad 4

Name Mensaje1
Text Empleado
BackColor (A su gusto)

Name Mensaje2
Text Hijos
BackColor (A su gusto)

Name Mensaje3
Text Estado Civil
BackColor (A su gusto)

TEXTBOX

Cantidad	1
Name	Tdato1
Text	(Vacío)

NUMERICUPDOWN

Cantidad	1
Name	Tdato2
Mínimum	0
Máximum	12

COMBOBOX

Cantidad	1
Name	Tdato3
Items	Soltero Casado Viudo Divorciado Union Libre

GROUPBOX

Cantidad	1
Name	GroupBox1
Text	Estudia

RADIOBUTTON

Cantidad	2
Name	Opcion1
Text	SI
Name	Opcion2
Text	NO
Checked	True

BUTTON

Cantidad	4
Name	Boton1
Text	Añadir
BackColor	(A su gusto)

Name Boton2
Text Nuevo
BackColor (A su gusto)

Name Boton3
Text Reporte
BackColor (A su gusto)

Name Boton4
Text Salir
BackColor (A su gusto)

LISTVIEW

Cantidad 1

Name ListView1
Columns (Agregar 4 Columnas)
Columna1. Empleado Ancho 80
Columna2. Hijos Ancho 60
Columna3. Estado Civil Ancho 90
Columna4. Estudia Ancho 60

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        //Cursor en el Empleado
        Tdato1.Focus();
        //Vista detallada del ListView
        ListView1.View = View.Details;
        //Cuadrícula
        ListView1.GridLines = true;
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //Añadir
        //Se guardará información en texto plano
        String Resultado;
        Resultado = Opcion1.Checked ? "Si" : "No";
        System.IO.StreamWriter archivo = new
        System.IO.StreamWriter("C:\\Empleados.txt", true);
        archivo.WriteLine(Tdato1.Text + " " + Tdato2.Value + " "
        + Tdato3.Text + " " + Resultado);
        archivo.Close();
    }
}
```

```

//agrega los datos en el list view
//Solo un Dato
ListView1.Items.Add(Tdato1.Text);

//Varios Datos
String[] Vector = new String[4];
ListViewItem VarItem;
Vector[0] = Tdato1.Text;
Vector[1] = Tdato2.Text;
Vector[2] = Tdato3.Text;
Vector[3] = Resultado;
VarItem = new ListViewItem(Vector);
ListView1.Items.Add(VarItem);
}

private void Boton2_Click(object sender, EventArgs e)
{
    //NUEVO
    //Controles en blanco
    Tdato1.Text = "";
    Tdato2.Value = 0;
    Tdato3.Text = "";
    Opcion2.Checked = true;
}

private void Boton3_Click(object sender, EventArgs e)
{
    //Carga el Archivo TXT
    System.Diagnostics.Process BlocDeNotas = new
    System.Diagnostics.Process();
    BlocDeNotas.StartInfo.FileName = "notepad.exe";
    BlocDeNotas.StartInfo.Arguments="c:\\empleados.txt";
    BlocDeNotas.Start();
}

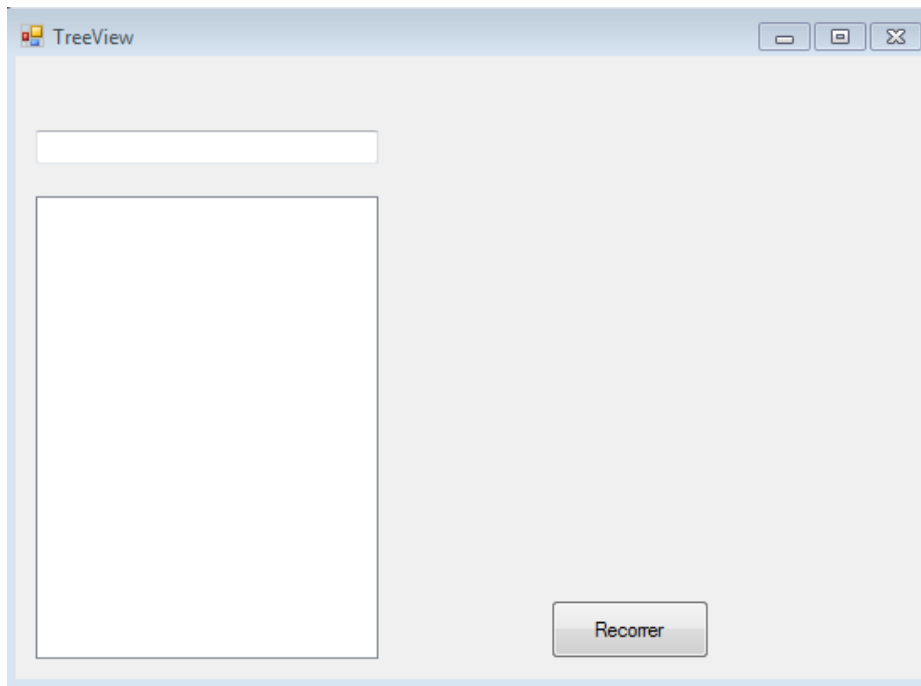
private void Salir_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}

```

5. GRABAR Y EJECUTAR

TREEVIEW

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	TreeView
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

TEXTBOX

Cantidad	1
Name	Tdato
ReadOnly	True
Text	(Vacío)

TREEVIEW

Cantidad	1
Name	TreeView1
CheckBoxes	True

BUTTON

Cantidad	1
Name	Boton1
Text	Recorrer
BackColor	(A su gusto)

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        // GENERAREMOS VALORES
        Random rnd = new Random();
        int TamA = 0;
        int TamB = 0;
        var _with1 = TreeView1;
        for (int I = 1; I <= 4; I++)
        {
            string Padre = "Padre" + I;
            TreeNode aux = _with1.Nodes.Add(Padre, "PADRE: " + I);
            TamA = Convert.ToInt32(rnd.Next() * 10);
            for (int J = 1; J <= TamA; J++)
            {
                string hijo = Padre + "\\Hijo " + J;
                TreeNode auxi = aux.Nodes.Add(hijo, "HIJO " + J);
                TamB = Convert.ToInt32(rnd.Next() * 10);
                for (int K = 1; K <= TamB; K++)
                {
                    string Nieto = hijo + "\\Nieto " + K;
                    auxi.Nodes.Add(Nieto, "NIETO:" + K);
                }
            }
        }

        private void TreeView1_AfterSelect(object sender,
        TreeViewEventArgs e)
        {
            Tdato1.Text = e.Node.FullPath;
        }

        private void Boton1_Click(object sender, EventArgs e)
        {
            // Se Declara una colección de nodos apartir de nuestro Treeview
            // que vamos a recorrer
        }
    }
}
```

```

        TreeNodeCollection nodes = TreeView1.Nodes;
        // Se recorren los nodos principales
        foreach (TreeNode n in nodes)
        {
            // Se Declara un metodo para que recorra los hijos de los principales
            // Y los hijos de los hijos...Recorrido Total en pocas palabras
            // Para ello se envía el nodo actual para evaluar si tiene hijos
            RecorrerNodos(n);
        }
    }

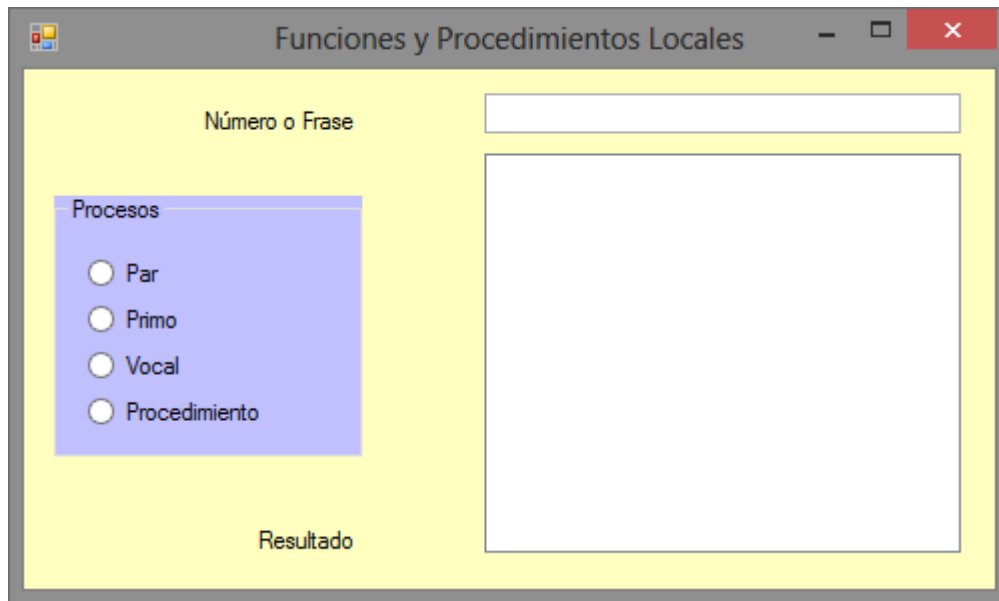
    public void RecorrerNodos(TreeNode treeNode)
    {
        try {
            // Si el nodo que recibimos tiene hijos se recorrerá
            // para luego verificar si esta o no checado
            foreach (TreeNode TN in treeNode.Nodes)
            {
                // Se Verifica si esta marcado...
                if (TN.Checked == true)
                {
                    // Si esta marcado mostramos el texto del nodo
                    MessageBox.Show(TN.Text);
                }
                // Ahora hago verificacion a los hijos del nodo actual
                // Esta iteración no acabara hasta llegar al último nodo principal
                RecorrerNodos(TN);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
    //FUNCIONAMIENTO
    //PARA PROBAR EL FUNCIONAMIENTO DEBE SELECCIONAR UN NIETO Y
    //LA INFORMACIÓN SALDRA EN EL CUADRO DE TEXTO
    //DEBE MARCA ALGUNOS HIJOS O NIETOS PARA PROBAR EL BOTÓN
}

```

5. GRABAR Y EJECUTAR

FUNCIONES Y PROCEDIMIENTOS LOCALES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Funciones y Procedimientos Locales
BackColor	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	2
Name	Mensaje1
BackColor	(A su gusto)
Text	Número o Frase
TextAlign	MiddleCenter
Name	Mensaje2
BackColor	(A su gusto)
Text	Resultado

LISTBOX

Cantidad	1
Name	Lista

GROUPBOX

Cantidad 1

Name Grupo
Backcolor (A su gusto)
Text Procesos

RADIOBUTTON (Ojo poner directamente dentro de GroupBox)

Cantidad 4

Name Opcion1
Backcolor (A su gusto)
Text Par

Name Opcion2
Backcolor (A su gusto)
Text Primo

Name Opcion3
Backcolor (A su gusto)
Text Vocal

Name Opcion4
Backcolor (A su gusto)
Text Procedimiento

TEXTBOX

Cantidad 1

Name Tdato

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Opcion1_CheckedChanged(object sender, EventArgs e)
    {
        //Par e Impar
        int Num;
        Num = Convert.ToInt16(Tdato.Text);
        Mensaje2.Text = PARIMPAR(Num);
    }

    private void Opcion2_CheckedChanged(object sender, EventArgs e)
    {
```

```

        //Primo
        int Num;
        Num = Convert.ToInt16(Tdato.Text);
        Mensaje2.Text = PRIMO(Num);
    }

private void Opcion3_CheckedChanged(object sender, EventArgs e)
{
    //Vocal
    String Frase;
    Mensaje2.Text = "Resultado";
    Frase = Tdato.Text;
    Lista.Items.Clear();
    Lista.Items.Add("Vocales A : " + (CONTAR(Frase, "A")));
    Lista.Items.Add("Vocales E : " + (CONTAR(Frase, "E")));
    Lista.Items.Add("Vocales I : " + (CONTAR(Frase, "I")));
    Lista.Items.Add("Vocales O : " + (CONTAR(Frase, "O")));
    Lista.Items.Add("Vocales U : " + (CONTAR(Frase, "U")));
    SEPARAR(Frase); //REM Invocamos a Procedimiento
}

private void Opcion4_CheckedChanged(object sender, EventArgs e)
{
    //Procedimiento
    Mensaje2.Text = "Resultado";
    Lista.Items.Clear();
    SEPARAR(Tdato.Text);
}

private String PARIMPAR(int Aux)
{
    int x;
    String Result;
    x = Aux % 2;
    if (x == 0)
        Result = "PAR";
    else
        Result = "IMPAR";
    return Result; //Forma de Retorno 1
}

private String PRIMO (int Aux)
{
    int R, sw, i;
    i = 2; sw = 0;
    while ((i < Aux) && (sw == 0))
    {
        R = Aux % i;
        if (R == 0)
            sw = 1;
        else
            i += 1;
    }
}

```

```

        if (sw == 0)
            return "ES PRIMO"; //REM Forma de Retorno 2
        else
            return "NO ES PRIMO";
    }

private int CONTAR(String Cadena, String Caracter)
{
    //Cuenta un Determinado Caracter
    int i, Ac;
    String Letra;
    Ac = 0;
    for (i = 0; i<= Cadena.Length-1; i++)
    {
        Letra = Cadena.Substring(i, 1);
        //MessageBox.Show(Letra);
        Letra = Letra.ToUpper();
        if (Letra == Caracter.ToUpper())
            Ac += 1;
    }
    return Ac;
}

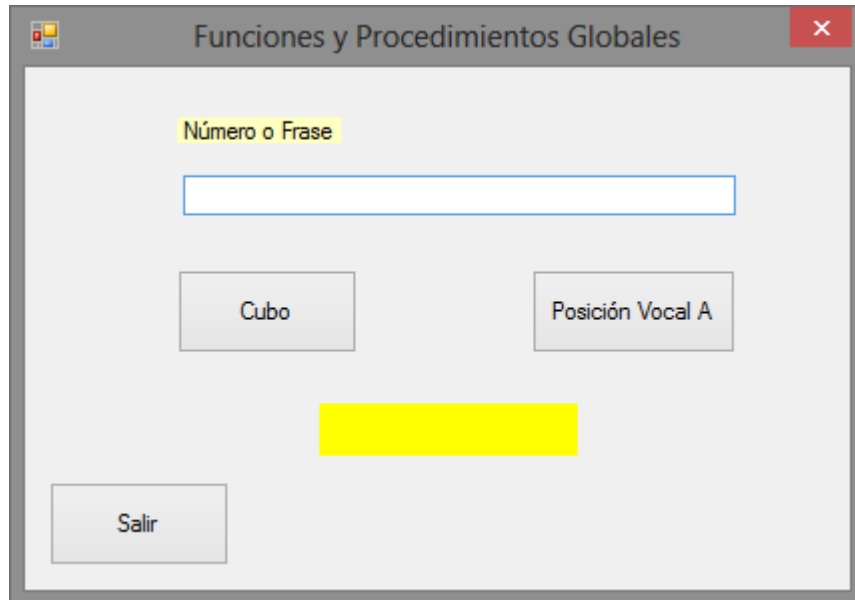
private void SEPARAR(String Cadena)
{
    int i;
    String Letra;
    Lista.Items.Add("Separar en Minusculas");
    for (i = 0; i <= Cadena.Length-1; i++)
    {
        Letra = Cadena.Substring(i, 1);
        Letra= Letra.ToLower();
        Lista.Items.Add(Letra);
    }
}
}

```

5. GRABAR Y EJECUTAR

FUNCIONES Y PROCEDIMIENTOS GLOBALES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Funciones y Procedimientos Globales
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad	2
Name	Mensaje1
BackColor	(A su gusto)
Text	Número o Frase
Name	Mensaje2
BackColor	(A su gusto)
Text	Vacío

TEXTBOX

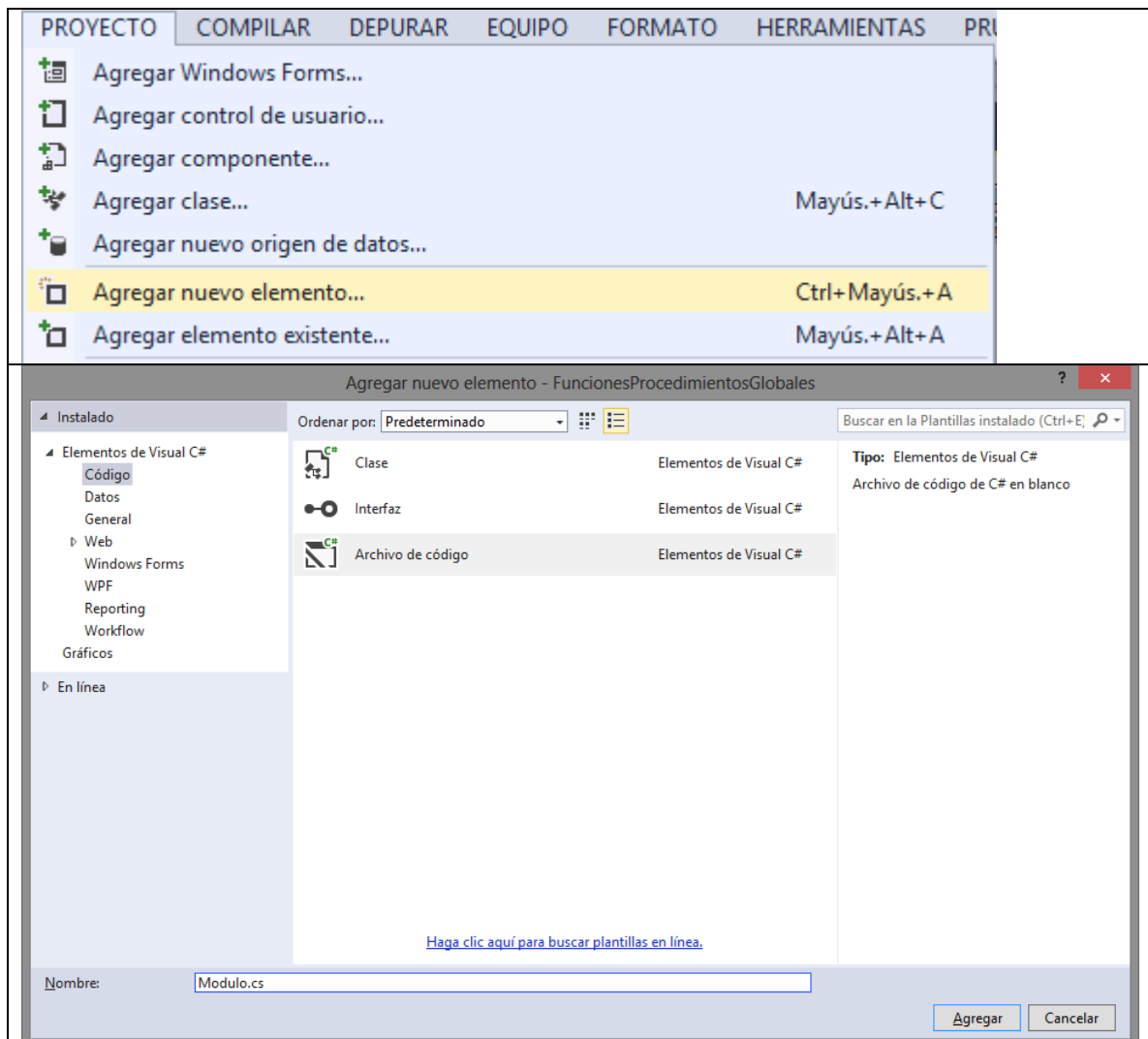
Cantidad	1
Name	Tdato
Text	Vacío

BUTTON

Cantidad	3
Name	Boton1
BackColor	(A su gusto)
Text	Cubo
Name	Boton2
BackColor	(A su gusto)
Text	Posición Vocal A
Name	Boton3
BackColor	(A su gusto)
Text	Salir

MODULO

Cantidad	1
Nombre	Modulo



4. CODIGO

EVENTOS DEL FORMULARIO

```
public partial class Form1 : Form
{
    opera x = new opera(); //Se crea una instancia de la clase
                           opera para llamar a las funciones
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        double Valor;
        String Resp;
        Valor = Convert.ToDouble(this.Tdato.Text);
        Resp = Convert.ToString(x.CUBO(Valor));
        Mensaje2.Text = Resp;
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        String Frase;
        int Resp;
        Frase = this.Tdato.Text;
        Resp = x.VOCAL(Frase, "A") + 1;
        Mensaje2.Text = Convert.ToString(Resp);
    }

    private void Boton3_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

CÓDIGO DEL MÓDULO

```
using System;

namespace operaciones{

    public class opera
    {
        public double CUBO(double Num)
        {
            double R;
            R = Math.Pow(Num, 3);
        }
    }
}
```

```

        return R;
    }

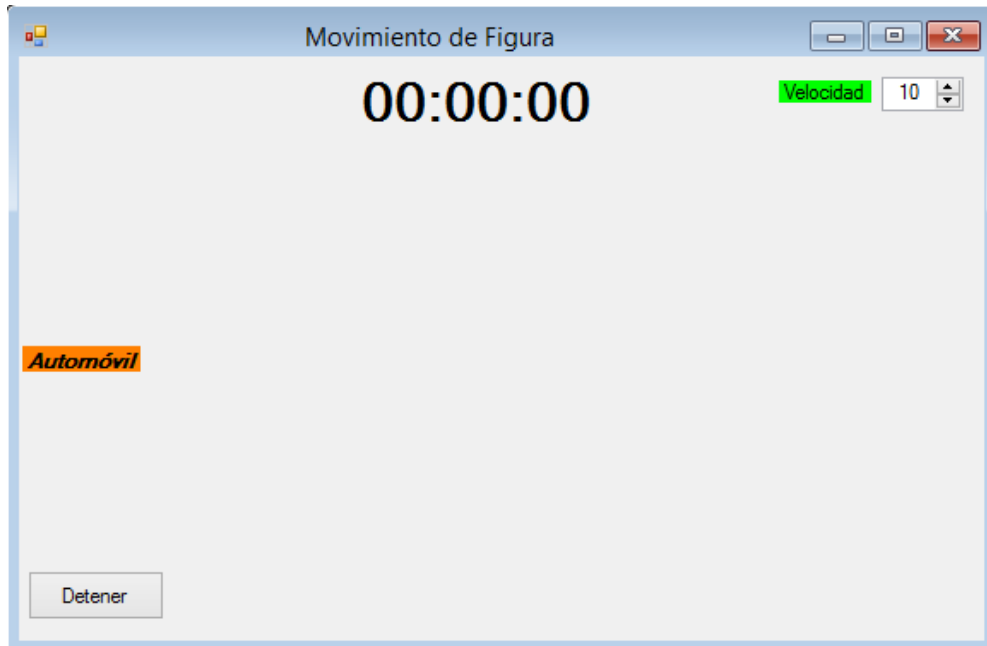
    public int VOCAL(string Texto, string Letra)
    {
        int Posi=0, I;
        String Fra;
        for (I = 1; I<= Texto.Length-1; I++)
        {
            Fra = Texto.Substring(I, 1);
            if (Fra.ToUpper() == Letra.ToUpper())
            {
                Posi = I;
                I = Texto.Length;
            }
        }
        return Posi;
    }
}
}

```

5. GRABAR Y EJECUTAR

MOVIMIENTO DE FIGURA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

Name	Form1
Text	Movimiento de una Figura
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
Size	580, 380
WindowState	Normal

LABEL

Cantidad 3

Name	Mensaje1
Text	00:00:00
BackColor	(A su gusto)
Font	(Tamaño 24, Negrita)

Name	Mensaje2
Text	Automóvil
Font	(Tamaño 10, Negrita-Cursiva)
BackColor	(A su gusto)

Name Mensaje3

Text Velocidad
Font (Tamaño 10)
BackColor (A su gusto)

TIMER

Cantidad 1

Name Reloj
Enabled True
Interval 10 (Está en milisegundos)

NUMERICUPDOWN

Cantidad 1

Name ControlNumerico
TextAlign Center
Minimum 1
Maximum 50
Increment 2
Value 10 (Similar a Timer)

BUTTON

Cantidad 1

Name Boton1
Text Detener
BackColor (A su gusto)

4. CODIGO

```
public partial class Form1 : Form
{
    int DX; //Variable para incremento horizontal
    int DY; //Variable para incremento vertical

    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //CONTROL DEL BOTON PARA DETENER O MOVER
        if (Boton1.Text == "Detener")
        {
            Boton1.Text = "Mover";
            Relej.Enabled = false;
        }
        else
        {
            Relej.Enabled = true;
            Boton1.Text = "Detener";
        }
    }
}
```

```

    }
}

private void ControlNumerico_ValueChanged(object sender,
EventArgs e)
{
    //Intervalo para el Reloj
    Reloj.Interval = Convert.ToInt16(ControlNumerico.Value);
}

private void Form1_Load(object sender, EventArgs e)
{
    //Inicialización de Variables
    DX = 3;
    DY = 4;
}

private void Reloj_Tick(object sender, EventArgs e)
{
    //CAPTURA LA HORA DEL COMPUTADOR DOS FORMAS
    Mensaje1.Text = DateTime.Now.ToString("hh:mm:ss");

    //Mensaje1.Text = Date.Now
    //MOVIMIENTO HORIZONTAL
    if (this.Mensaje2.Left >= (this.Width - this.Mensaje2.Width))
        //Controlo limite derecho
        DX = -DX;
    if (this.Mensaje2.Left <= 0) //Controlo limite izquierdo
        DX = Math.Abs(DX);
    this.Mensaje2.Left = this.Mensaje2.Left + DX;

    //MOVIMIENTO VERTICAL
    if (Mensaje2.Top >= (this.Height - this.Mensaje2.Height))
        //Controlo limite inferior
        DY = -DY;

    if (this.Mensaje2.Top <= 0) //Controlo limite superior
        DY = Math.Abs(DY);

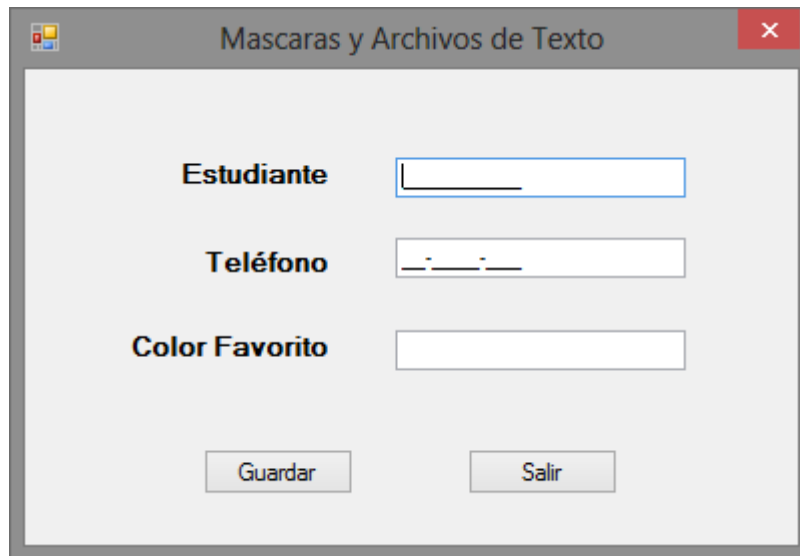
    Mensaje2.Top = Mensaje2.Top + DY;
}
}

```

5. GRABAR Y EJECUTAR

MASCARAS Y ARCHIVOS DE TEXTO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



The image shows a screenshot of a Windows application window titled "Mascaras y Archivos de Texto". The window contains a form with three text input fields. The first field is labeled "Estudiante" and is empty. The second field is labeled "Teléfono" and contains the text "1-2-3-4". The third field is labeled "Color Favorito" and is empty. Below the fields are two buttons: "Guardar" and "Salir".

3. Componentes

FORM

Name	Form1
Text	Mascaras y Archivos de Texto
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	False
MinimizeBox	Flase
StartPosition	CenterScreen
WindowState	Normal

LABEL

Cantidad 3

Name	Mensaje1
Text	Estudiante
BackColor	(A su gusto)
Font	(Tamaño 10, Negrita)

Name	Mensaje2
Text	Teléfono
Font	(Tamaño 10, Negrita)
BackColor	(A su gusto)

Name	Mensaje3
Text	Color Favorito:
Font	(Tamaño 10, Negrita)
BackColor	(A su gusto)

MASKTEXTBOX

Cantidad	2
Name	MaskTextBox1
Mask	>AAAAAAAAAAAA
Name	MaskTextBox2
Mask	>##-####-###

TEXTBOX

Cantidad	1
Name	TData1
Mask	(Vacío)

BUTTON

Cantidad	2
Name	Boton1
Text	Guardar
BackColor	(A su gusto)
Name	Boton2
Text	Salir
BackColor	(A su gusto)

4. CODIGO

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        //ASIGNAMOS MASCARA Y MARCADOR
        MaskedTextBox2.Mask = "(##)####-###";
        MaskedTextBox2.PromptChar = '0';
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //BOTON GUARDAR

        //AQUI PONEMOS LA RUTA DONDE SE CREARA EL ARCHIVO DE TEXTO
        string VarArchivo = "C:\\juegos\\MiArchivo.txt";
        int i;
        string[] MIARREGLO = new string[3];
    }
}
```

```

MIARREGLO[0] = (MaskedTextBox1.Text + ".");
MIARREGLO[1] = (MaskedTextBox2.Text + ".");
MIARREGLO[2] = (TDato1.Text + ".");

System.IO.StreamWriter VarGuardar = new
System.IO.StreamWriter(VarArchivo, true);

for (i = 0; i<3; i++)
    VarGuardar.WriteLine(MIARREGLO[i]);

VarGuardar.Close();
MessageBox.Show("Datos Almacenados con éxito");
}

private void Boton2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}

```

5. GRABAR Y EJECUTAR

CONTROL DE ERRORES Y AYUDA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

Control de Errores y Ayuda

Nombre

Apellido

Edad

Dirección (F1)

Cerrar

ErrorProvider1 HelpProvider1

3. Componentes

FORM

Name	Form1
Text	Control de Errores y Ayuda
BackColor	(A su gusto)
MaximizeBox	False
MinimizeBox	False
StartPosition	CenterScreen
WindowState	Normal
HelpButton	True

LABEL

Cantidad	4
Name	Mensaje1
BackColor	(A su gusto)
Font	(Tamaño 10)
Text	Nombre
Name	Mensaje2
BackColor	(A su gusto)
Font	(Tamaño 10)

Text	Apellido
Name	Mensaje3
BackColor	(A su gusto)
Font	(Tamaño 10)
Text	Edad
Name	Mensaje4
BackColor	(A su gusto)
Font	(Tamaño 10)
Text	Dirección (F1)

TEXTBOX

Cantidad 4

Name	Tdato1
Text	Vacío

Name	Tdato2
Text	Vacío

Name	Tdato3
Text	Vacío

Name	Tdato4
Text	Vacío

BUTTON

Cantidad 1

Name	Boton1
BackColor	(A su gusto)
Text	Cerrar

ERRORPROVIDER

Cantidad 1

Name	ErrorProvider1
ContainerControl	Form1

HELPPROVIDER

Cantidad 1

Name	HelpProvider1
------	---------------

4. CODIGO

```
public partial class Form1 : Form
{
```

```

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    //PARA ENLAZAR MEDIANTE TECLA F1
    this.HelpProvider1.HelpNamespace =
    "http://msdn.microsoft.com/library/en-
    us/vbcon/html/vbconbuttoncontroloverview.asp";

    //FORMA 2, PARA QUE SE VISUALICE ICONO DE AYUDA
    this.HelpProvider1.SetHelpString(TDato1, "Ingrese el Nombre
    del Cliente");
    this.HelpProvider1.SetHelpString(TDato2, "Digite Apellido
    del Cliente");
}

private void TDato1_Validating(object sender, CancelEventArgs
e)
{
    if (TDato1.Text.Length == 0)
        ErrorProvider1.SetError(TDato1, "Recuerde Debe llenar
        Nombre");
    else
        ErrorProvider1.SetError(TDato1, "");
}

private void TDato2_Validating(object sender, CancelEventArgs
e)
{
    if (TDato2.Text.Length == 0)
        ErrorProvider1.SetError(TDato2, "Recuerde Debe llenar
        Apellido");
    else
        ErrorProvider1.SetError(TDato2, "");
}

private void TDato3_Validating(object sender, CancelEventArgs
e)
{
    if (!IsNumeric(TDato3.Text))
    {
        //esto define la frecuencia con la que parpadeará el
        ErrorProvider1
        this.ErrorProvider1.BlinkRate = 200;

        //para que siempre esté parpadeando mientras cuando
        no se corrija el error.
        this.ErrorProvider1.BlinkStyle =
        ErrorBlinkStyle.AlwaysBlink;
    }
}

```

```

        this.ErrorProvider1.SetError(TDato3, "No es correcto el
        dato ingresado");

        //con esto hacemos que el control TextBox no pierda el
        enfoque por el momento...
        e.Cancel = true;
    }
    else
    {
        this.ErrorProvider1.SetError(TDato3, "");
        e.Cancel = false;
    }
}

private void TDato4_HelpRequested(object sender, HelpEventArgs
hlpevent)
{
    HelpProvider1.SetHelpString(TDato4, "Dirección del
    Domicilio");
}

private void Boton1_Click(object sender, EventArgs e)
{
    this.Close();
}

//Permite verificar si el contenido de TDato3 es un número
public static Boolean IsNumeric(string valor)
{
    int result;
    return int.TryParse(valor, out result);
}
}

```

5. GRABAR Y EJECUTAR

CONEXIÓN CON BASE DE DATOS ACCESS POR CÓDIGO

1. Abrir un nuevo Proyecto
2. Estructura de la Base de Datos (Access)

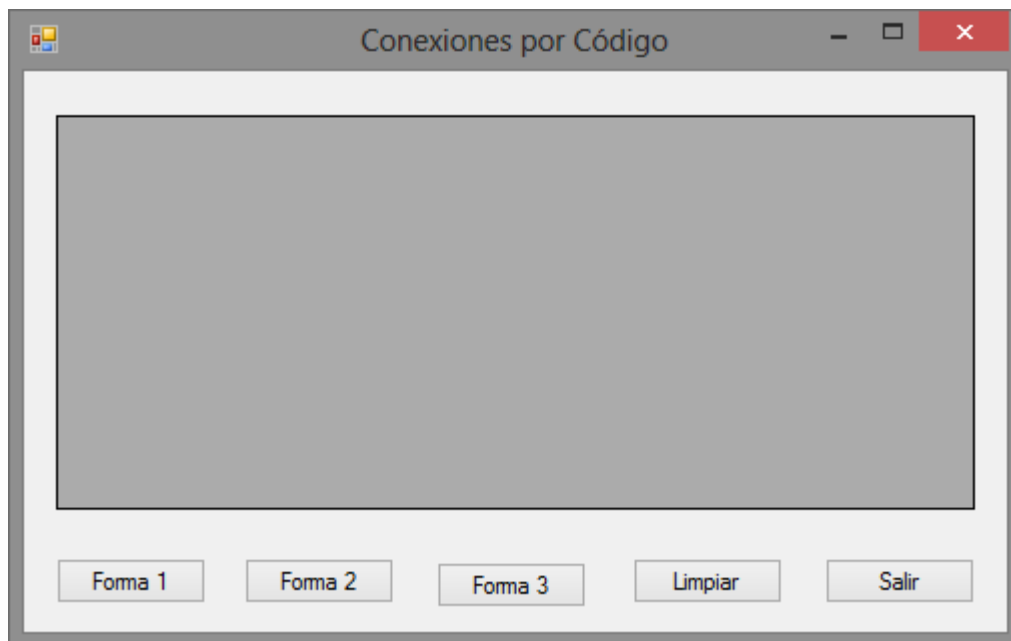
AULA.MDB VERSIÓN 2003

TBLDATOS

* Matricula	Numérico (Entero) (Campo Llave)
Alumno	Texto corto (30)
Curso	Numerico (Byte)
Sexo	Texto corto (1)
Fnace	Fecha/Hora (Fecha Corta)
Retirado	Sí/No
CostoMatricula	Moneda

(Nota: Digite la información de 5 Registros)

3. Apariencia del Formulario



4. Componentes

FORM

Name	Form1
Text	Conexiones por Código
BackColor	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	4
Name	Boton1
BackColor	(A su gusto)
Text	Forma 1
Name	Boton2
BackColor	(A su gusto)
Text	Forma 2
Name	Boton3
BackColor	(A su gusto)
Text	Forma 3
Name	Boton4
BackColor	(A su gusto)
Text	Salir
Name	Boton5
BackColor	(A su gusto)
Text	Limpiar

DATAGRIDVIEW

Cantidad	1
Name	DataGridView1

5. CODIGO

```
using System.Data.Common;
using System.Data.OleDb;

public partial class Form1 : Form
{
    DbProviderFactory DBA =
DbProviderFactories.GetFactory("System.Data.OleDb");
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //FORMA 1

        DbConnection CN_A = DBA.CreateConnection();
        DbCommand CMD_A = CN_A.CreateCommand();
```

```

        OleDbDataAdapter DA_A = DBA.CreateDataAdapter();
        DataTable DT_A = new DataTable();

        CN_A.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;
        Data Source=C:\\Aula.mdb";
        CMD_A.CommandText = "Select * From TblDatos";
        DA_A.SelectCommand = CMD_A;
        DA_A.Fill(DT_A);
        DataGridView1.DataSource = DT_A;
    }

private void Boton2_Click(object sender, EventArgs e)
{
    //FORMA 2
    OleDbConnection Conexion;
    OleDbDataAdapter Adap;
    DataSet Tabla;

    //Enlazando coneccion a la base de datos
    Conexion = new
    OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=C:\\aula.mdb");

    //Cargando el adapter con la instruccion sql
    Adap = new OleDbDataAdapter("Select * from TblDatos",
    Conexion);

    //Cargando el dataset
    Tabla = new DataSet();
    Adap.Fill(Tabla, "TblDatos");

    //Cargando el datagridview
    DataGridView1.DataSource = Tabla;
    DataGridView1.DataMember = "tbldatos";
}

private void Boton3_Click(object sender, EventArgs e)
{
    //FORMA 3

    String conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=C:\\aula.mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(conexion);
        OleDbCommand CMD = new OleDbCommand("Select *
        TblDatos", CONN);
        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();

        CONN.Open();
        DA.Fill(DS);
        CONN.Close();
    }
}

```

```
        DataGridView1.DataSource = DS.Tables[0];
    }
    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Boton4_Click(object sender, EventArgs e)
{
    this.Close();
}

private void Boton5_Click(object sender, EventArgs e)
{
    DataGridView1.Columns.Clear();
}
}
```

6. GRABAR Y EJECUTAR

CONSULTAS POR CÓDIGO

1. Abrir un nuevo Proyecto
2. Estructura de la Base de Datos (Access)

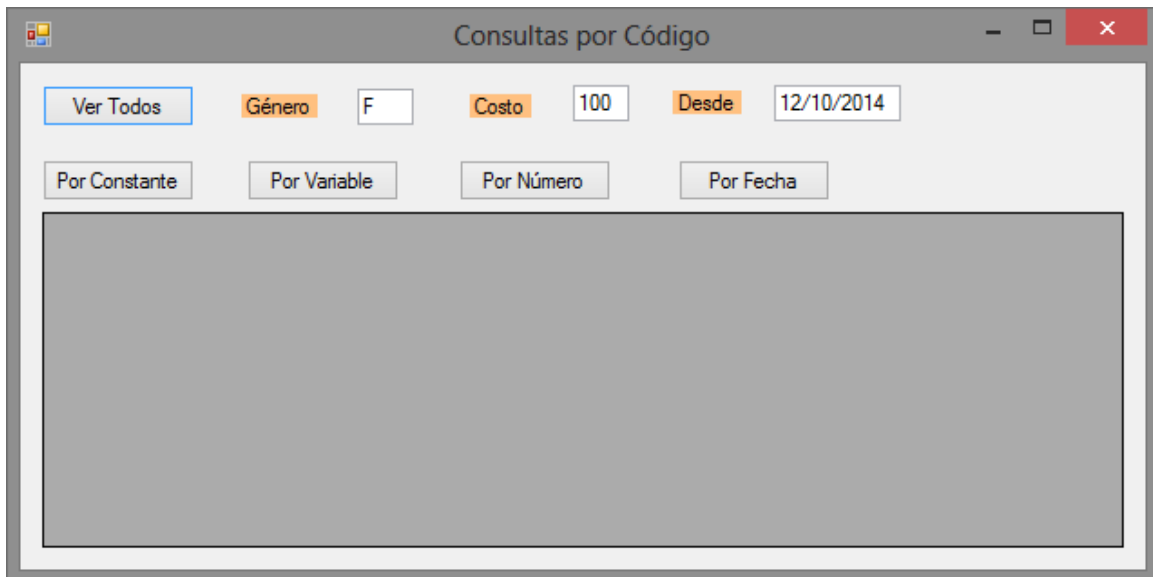
AULA.MDB

TBLDATOS

* Matricula	Numérico (Entero) (Campo Llave)
Alumno	Texto corto (30)
Curso	Numerico (Byte)
Sexo	Texto corto (1)
Fnace	Fecha/Hora (Fecha Corta)
Retirado	Sí/No
CostoMatricula	Moneda

(Nota: Digite la información de 5 Registros O puede utilizar la Base de datos del ejercicio anterior)

3. Apariencia del Formulario



4. Componentes

FORM

Name	Form1
Text	Consultas por Código
BackColor	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	5
Name	Boton1
BackColor	(A su gusto)
Text	Ver Todos
Name	Boton2
BackColor	(A su gusto)
Text	Por Constante
Name	Boton3
BackColor	(A su gusto)
Text	Por Variable
Name	Boton4
BackColor	(A su gusto)
Text	Por Número
Name	Boton5
BackColor	(A su gusto)
Text	Por Fecha

DATAGRIDVIEW

Cantidad	1
Name	DataGrid

LABEL

Cantidad	3
Name	Mensaje1
BackColor	(A su gusto)
Text	Género
Name	Mensaje2
BackColor	(A su gusto)
Text	Costo
Name	Mensaje3
BackColor	(A su gusto)
Text	Desde

TEXTBOX

Cantidad	3
Name	Tdato1
BackColor	(A su gusto)

Text	F
Name	Tdato2
BackColor	(A su gusto)
Text	100
Name	Tdato3
BackColor	(A su gusto)
Text	01/04/2010

5. CODIGO

```

using System.Data.OleDb;

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //VER TODOS
        string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

        try
        {
            OleDbConnection CONN = new OleDbConnection(Conexion);
            OleDbCommand CMD = new OleDbCommand("Select
* From TblDatos", CONN);
            OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
            DataSet DS = new DataSet();

            //Cargando el Dataset
            DA.Fill(DS, "TblDatos");

            //Cargando el Datagridview
            DataGrid.DataSource = DS;
            DataGrid.DataMember = "TblDatos";

            CONN.Open();
            DA.Fill(DS);
            CONN.Close();
            DataGrid.DataSource = DS;
        }

        catch (OleDbException ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

```

}

private void Boton2_Click(object sender, EventArgs e)
{
    //CONSULTA POR VALOR CONSTANTE
    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Select * From
TblDatos Where Sexo='M'", CONN);

        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();

        //Cargando el Dataset
        DA.Fill(DS, "TblDatos");

        //Cargando el Datagridview
        DataGrid.DataSource = DS;
        DataGrid.DataMember = "TblDatos";

        CONN.Open();
        DA.Fill(DS);
        CONN.Close();
        DataGrid.DataSource = DS;
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Boton3_Click(object sender, EventArgs e)
{
    //POR VARIABLE TIPO STRING
    string Mivar;
    Mivar = Tdato1.Text.Trim(); //Retiramos espacios en Blanco

    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Select * From
TblDatos Where Sexo='" + Mivar + "'", CONN);
        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();
    }
}

```

```

        DA.Fill(DS, "TblDatos");
        DataGrid.DataSource = DS;
        DataGrid.DataMember = "TblDatos";

        CONN.Open();
        DA.Fill(DS);
        CONN.Close();
        DataGrid.DataSource = DS;
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Boton4_Click(object sender, EventArgs e)
{
    //POR VARIABLE TIPO NUMÉRICO
    int Mivar;
    Mivar = Convert.ToInt16(Tdato2.Text); //Tomamos el dato
    ingresado convirtiéndolo en int
    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=C:\\AULA.Mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Select * From
        TblDatos Where CostoMatricula >=" + Mivar, CONN);
        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();

        DA.Fill(DS, "TblDatos");
        DataGrid.DataSource = DS ;
        DataGrid.DataMember = "TblDatos";

        CONN.Open();
        DA.Fill(DS);
        CONN.Close();
        DataGrid.DataSource = DS;
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Boton5_Click(object sender, EventArgs e)
{

```

```

//POR VARIABLE TIPO FECHA
DateTime Mivar;
Mivar = Convert.ToDateTime(Tdato3.Text); //Tomamos el dato
ingresado
string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

try
{
    OleDbConnection CONN = new OleDbConnection(Conexion);
    OleDbCommand CMD = new OleDbCommand("Select * From
TblDatos Where Fnace >=#" + Mivar + "#", CONN);
    OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
    DataSet DS = new DataSet();

    DA.Fill(DS, "TblDatos");
    DataGrid.DataSource = DS;
    DataGrid.DataMember = "TblDatos";

    CONN.Open();
    DA.Fill(DS);
    CONN.Close();
    DataGrid.DataSource = DS;
}

catch (OleDbException ex)
{
    MessageBox.Show(ex.Message);
}
}
}

```

6. GRABAR Y EJECUTAR

CONSULTAS POR COMPONENTES

1. Abrir un nuevo Proyecto
2. Estructura de la Base de Datos (Access)

AULA.MDB

TBLDATOS

* Matricula	Numérico (Entero) (Campo Llave)
Alumno	Texto corto (30)
Curso	Numerico (Byte)
Sexo	Texto corto (1)
Fnace	Fecha/Hora (Fecha Corta)
Retirado	Sí/No
CostoMatricula	Moneda

(Nota: Digite la información de 5 Registros o puede utilizar la Base de datos del ejercicio anterior)

3. Apariencia del Formulario



4. Componentes

FORM

Name	Form1
Text	Consultas por Componentes
BackColor	(A su gusto)
StartPosition	CenterScreen
WindowState	Normal

BUTTON

Cantidad	5
Name	Boton1
BackColor	(A su gusto)
Text	Ver Todos
Name	Boton2
BackColor	(A su gusto)
Text	Por Constante
Name	Boton3
BackColor	(A su gusto)
Text	Por Texto
Name	Boton4
BackColor	(A su gusto)
Text	Por Número
Name	Boton5
BackColor	(A su gusto)
Text	Por Fecha

DATAGRIDVIEW

Cantidad	1
Name	DataGrid

LABEL

Cantidad	3
Name	Mensaje1
BackColor	(A su gusto)
Text	Género
Name	Mensaje2
BackColor	(A su gusto)
Text	Costo
Name	Mensaje3
BackColor	(A su gusto)
Text	Desde

TEXTBOX

Cantidad	3
Name	Tdato1

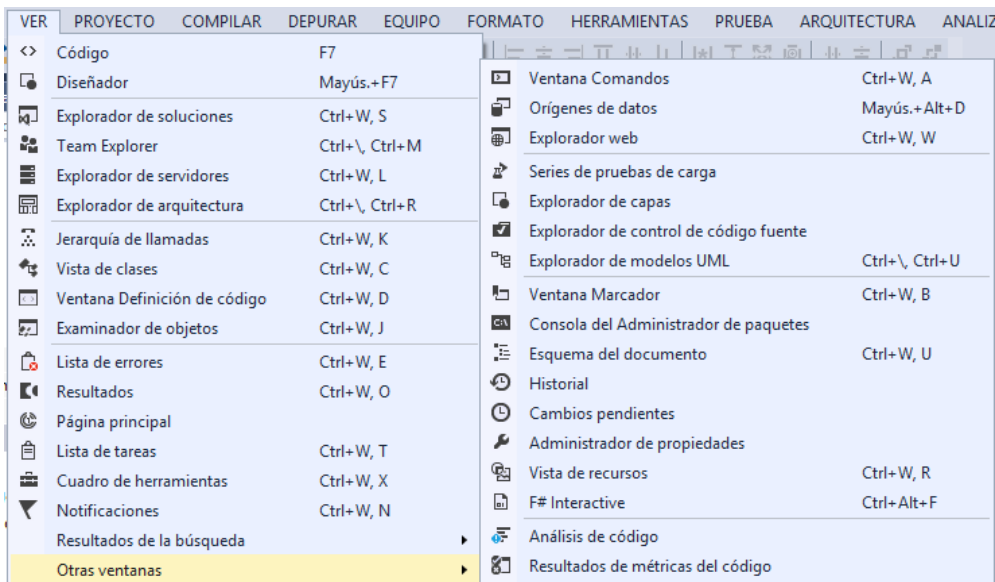
BackColor (A su gusto)
Text F

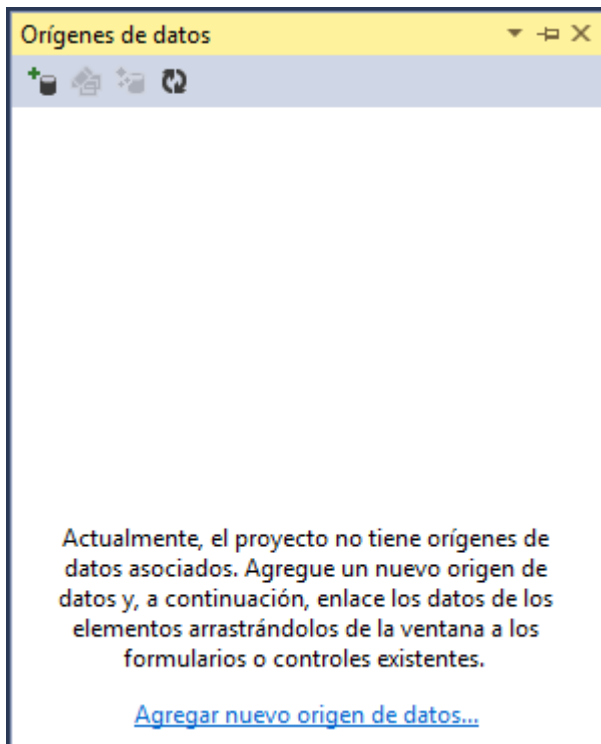
Name Tdato2
BackColor (A su gusto)
Text 120

Name Tdato3
BackColor (A su gusto)
Text 01/04/2010

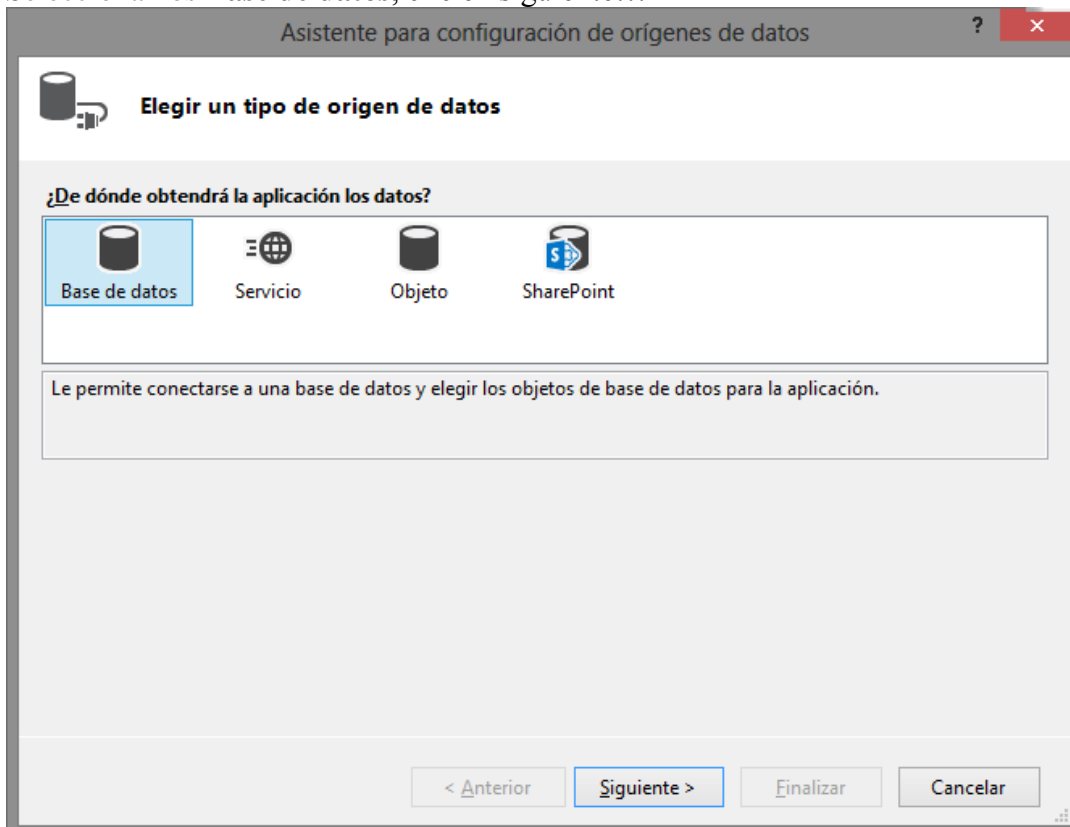
CONFIGURAR ORIGEN DE DATOS

Ir al menú ver, Otras ventanas, Orígenes de datos

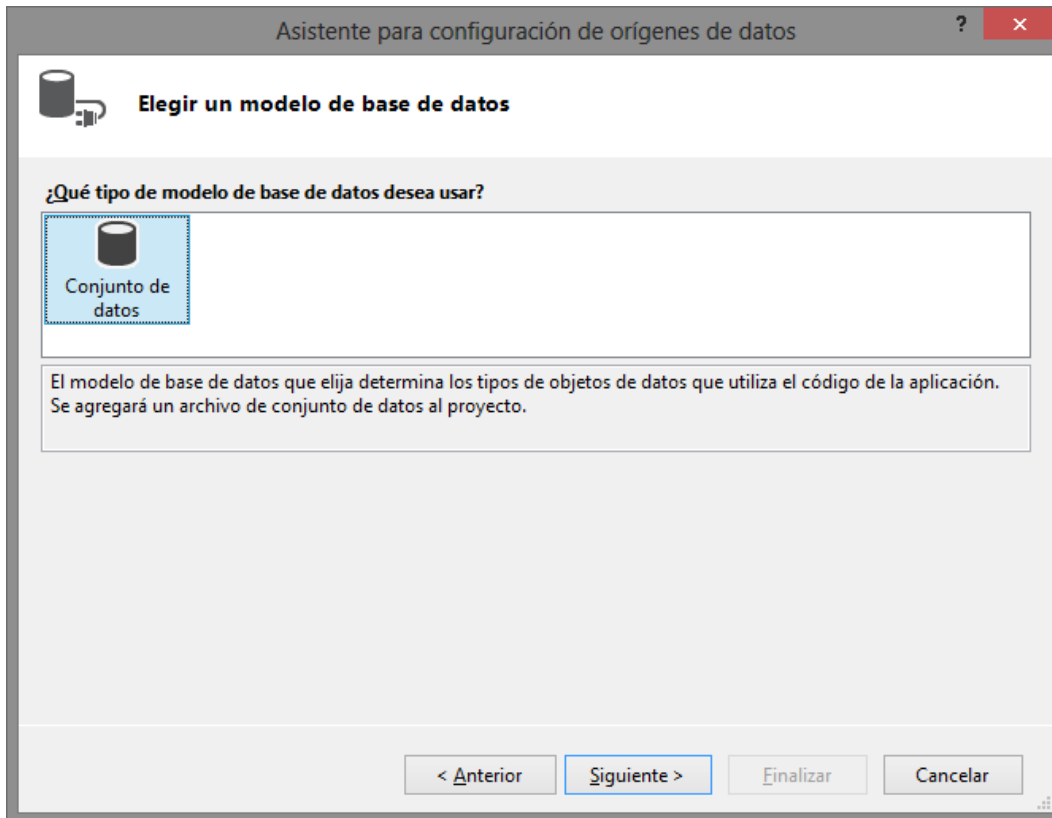




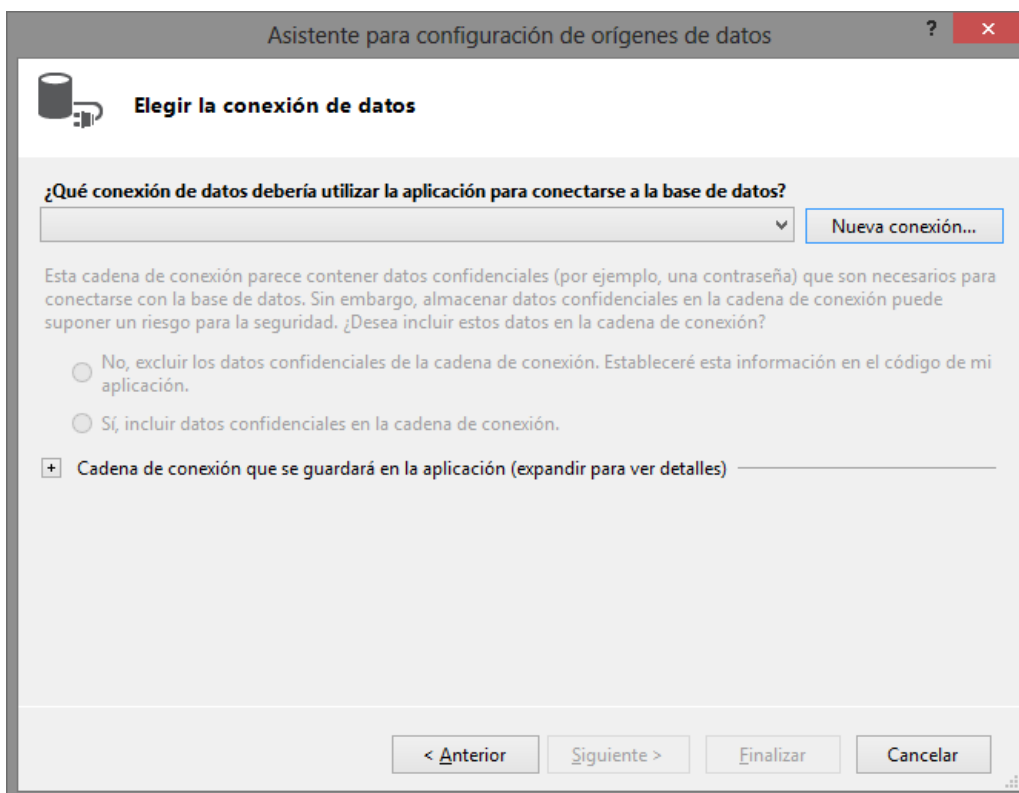
Clic en “Agregar nuevo origen de datos...”
Seleccionamos **Base de datos**, clic en siguiente...



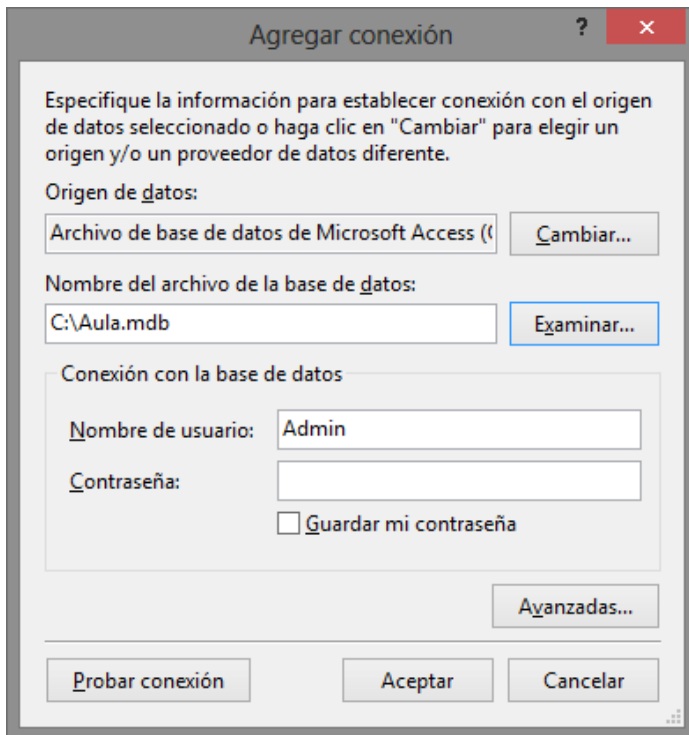
Seleccionamos **Conjunto de datos**, clic en siguiente...



Clic en **Nueva conexión...**

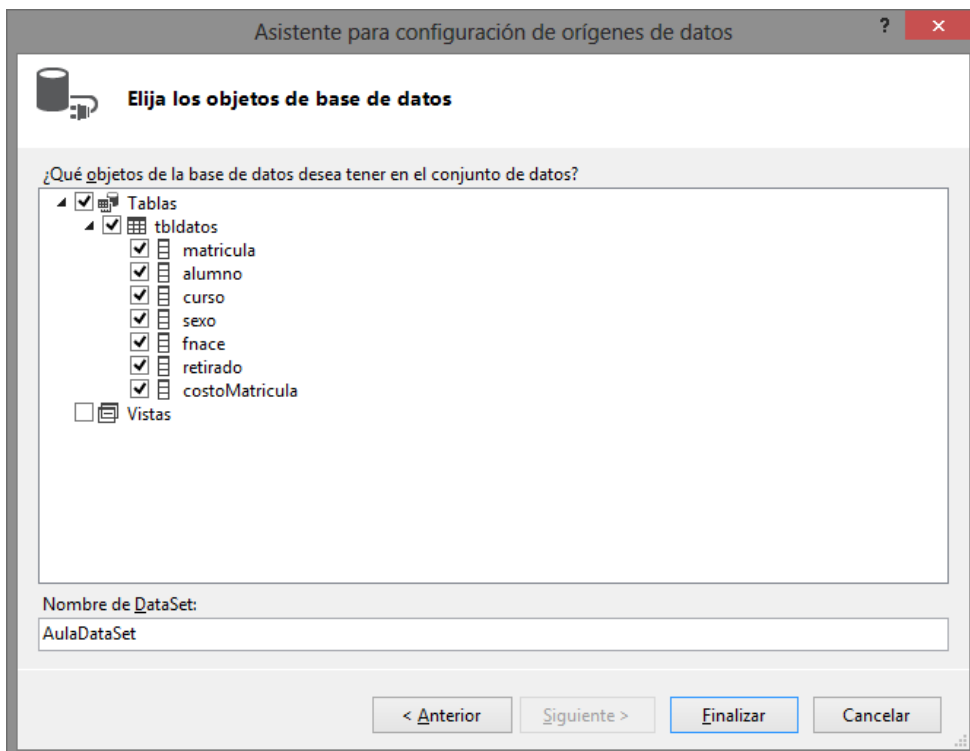


Clic en Examinar, y buscamos el archivo Aula.mdb



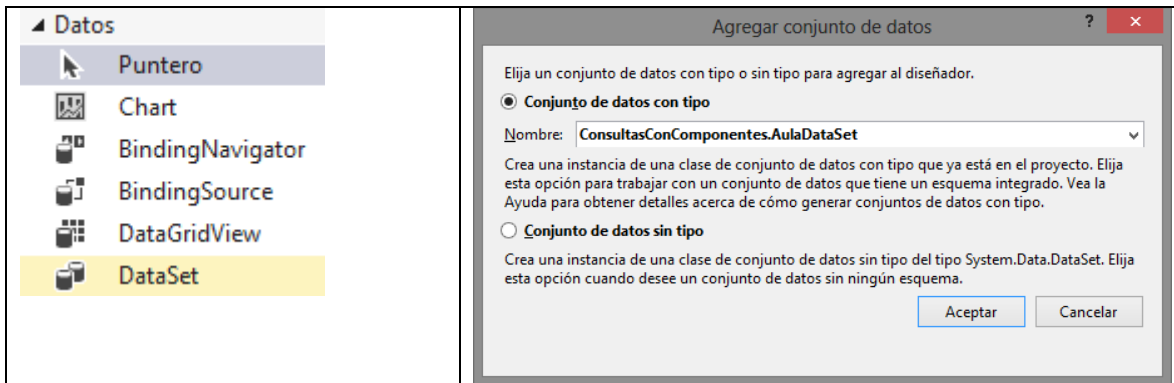
Clic en **Aceptar**, luego en **siguiente...**

Seleccionamos la tabla **tbldatos** y todos sus campos. Finalizamos.

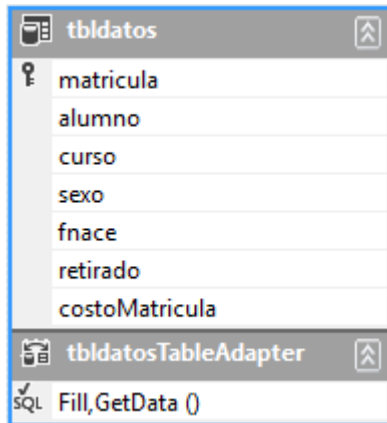


DATASET

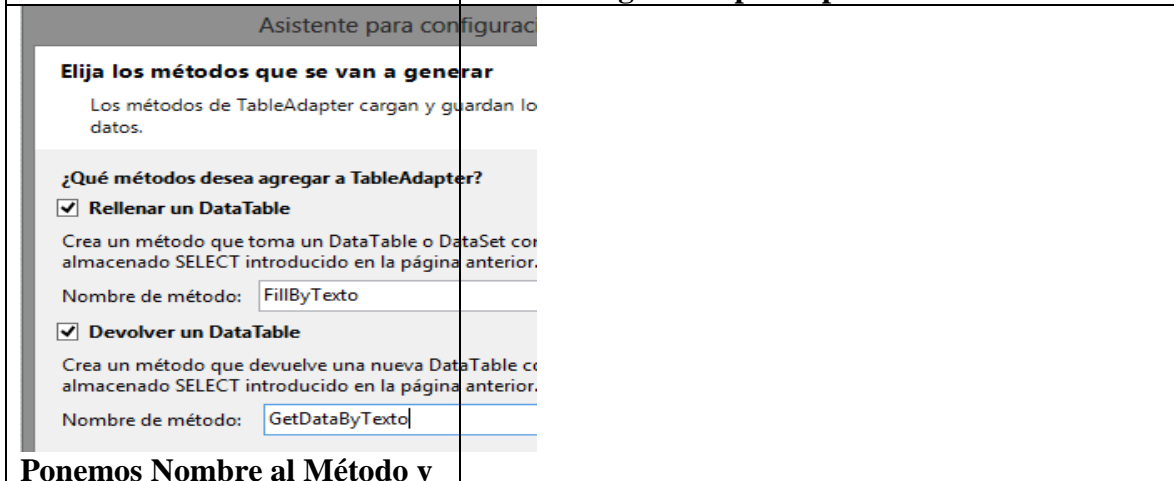
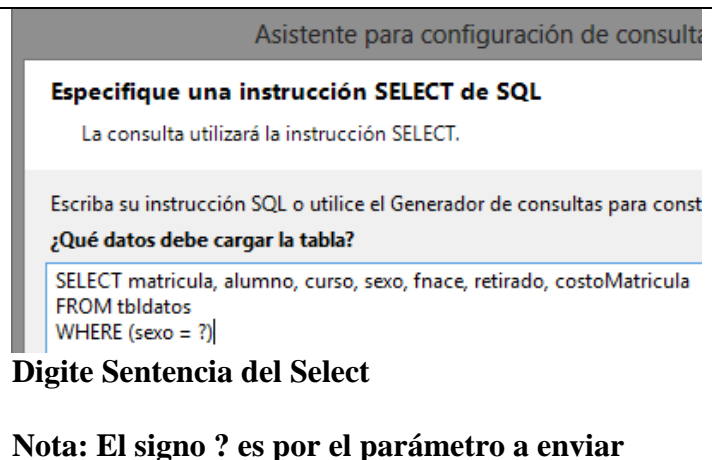
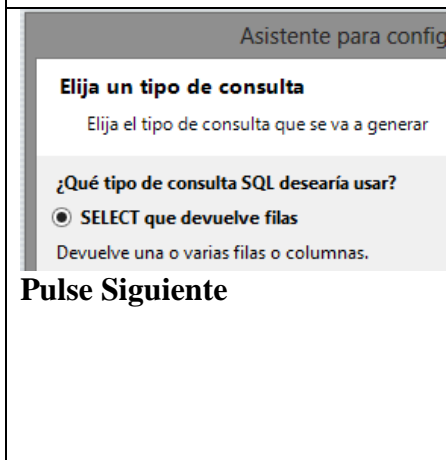
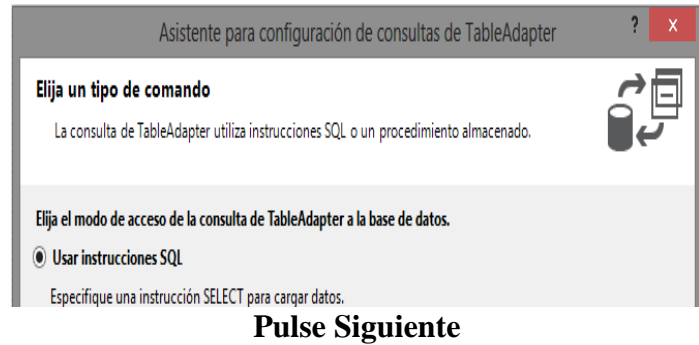
Cantidad 1



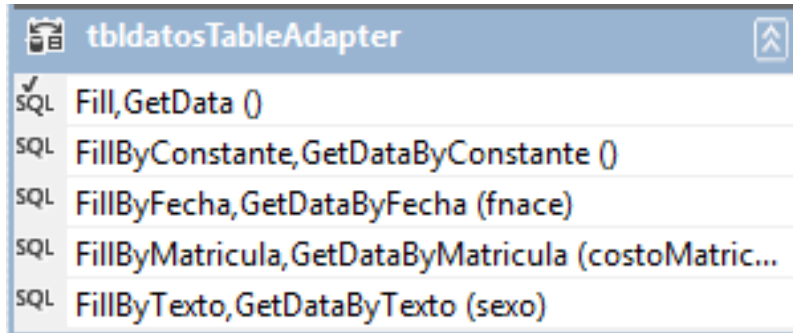
Clic derecho sobre DataSet1 y Editar



Clic derecho sobre tbldatos y escoger Agregar Consulta.



NOTA: GENERE TRES MÉTODOS PARA LOS CAMPOS FECHA, COSTO MATRICULA Y UN CAMPO CONSTANTE. QUEDARÁ:



Los Select serán:

CONSTANTE

```
SELECT * FROM tbldatos WHERE Sexo='M'
```

FECHA

```
SELECT * FROM tbldatos WHERE (Fnace >= ?)
```

COSTO MATRICULA

```
SELECT * FROM tbldatos WHERE (CostoMatricula >= ?)
```

5. CODIGO

```
using System.Data.OleDb;

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        //TODO: esta línea de código carga datos en la tabla
        'MiDataSet1.TblDatos'
        //Puede moverla o quitarla según sea necesario.
        this.tbldatosAdapter1.Fill(this.aulaDataSet1.tbldatos);
    }

    private void Boton1_Click(object sender, EventArgs e)
```

```

    {
        this.tbldatosAdapter1.Fill(this.aulaDataSet1.tbldatos);
        this.DataGrid.DataSource = aulaDataSet1.tbldatos;
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        //CONSULTA POR VALOR CONSTANTE

        this.tbldatosAdapter1.FillByConstante(this.aulaDataSet1.tbldatos);
        this.DataGrid.DataSource = aulaDataSet1.tbldatos;
    }

    private void Boton3_Click(object sender, EventArgs e)
    {
        //VARIABLE TIPO STRING
        string MiVar;
        MiVar = Tdato1.Text.Trim();

        this.tbldatosAdapter1.FillByTexto(this.aulaDataSet1.tbldatos, MiVar);
        this.DataGrid.DataSource = aulaDataSet1.tbldatos;
    }

    private void Boton4_Click(object sender, EventArgs e)
    {
        //VARIABLE TIPO NUMÉRICO
        int MiVar;
        MiVar = Convert.ToInt16(Tdato2.Text);

        this.tbldatosAdapter1.FillByMatricula(this.aulaDataSet1.tbldatos,
        MiVar);
        this.DataGrid.DataSource = aulaDataSet1.tbldatos;
    }

    private void Boton5_Click(object sender, EventArgs e)
    {
        //VARIABLE TIPO FECHA
        DateTime MiVar;
        MiVar = Convert.ToDateTime(Tdato3.Text);

        this.tbldatosAdapter1.FillByFecha(this.aulaDataSet1.tbldatos, MiVar);
        this.DataGrid.DataSource = aulaDataSet1.tbldatos;
    }
}

```

6. GRABAR Y EJECUTAR

INSERTAR REGISTROS POR CÓDIGO

1. Abrir un nuevo Proyecto
2. Estructura de la Base de Datos (Access)

AULA.MDB

TBLDATOS

* Matricula	Numérico (Entero) (Campo Llave)
Alumno	Texto corto (30)
Curso	Numerico (Byte)
Sexo	Texto corto (1)
Fnace	Fecha/Hora (Fecha Corta)
Retirado	Sí/No
CostoMatricula	Moneda

(Nota: Digite la información de 5 Registros o puede utilizar la Base de datos del ejercicio anterior)

3. Apariencia del Formulario

Insertar Registros por Código

Matrícula

Estudiante

Curso

Género

Nacimiento

Retirado

Costo

Consultar Guardar

4. Componentes

FORM

Name	Form1
Text	Insertar Registros por Código

BackColor (A su gusto)
StartPosition CenterScreen
WindowState Normal

BUTTON

Cantidad 2

Name Boton1
BackColor (A su gusto)
Text Consultar

Name Boton2
BackColor (A su gusto)
Text Guardar

DATAGRIDVIEW

Cantidad 1

Name DataGrid
AllowUserToAddRows False
AllowUserToDeleteRows False
AllowUserToOrderColumns True
ReadOnly True

LABEL

Cantidad 6

Name Mensaje1
BackColor (A su gusto)
Text Matricula

Name Mensaje2
BackColor (A su gusto)
Text Estudiante

Name Mensaje3
BackColor (A su gusto)
Text Curso

Name Mensaje4
BackColor (A su gusto)
Text Género

Name Mensaje5
BackColor (A su gusto)
Text Nacimiento

Name Mensaje6
BackColor (A su gusto)
Text Retirado

Name Mensaje7
BackColor (A su gusto)
Text Costo

TEXTBOX

Cantidad 6

Name Tdato1
BackColor (A su gusto)
Text Vacio

Name Tdato2
BackColor (A su gusto)
Text Vacio

Name Tdato3
BackColor (A su gusto)
Text 1

Name Tdato4
BackColor (A su gusto)
Text M

Name Tdato5
BackColor (A su gusto)
Text 29/05/2001

Name Tdato6
BackColor (A su gusto)
Text 169

CHECKBOX

Cantidad 1

Name CheckBox1
BackColor (A su gusto)
Text Retirado
CheckAlign MiddleRight

5. CODIGO

```
using System.Data.OleDb;
```

```
public partial class Form1 : Form
```

```

{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        CargarDatos();
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //CONSULTAR
        CargarDatos();
    }

    private void Boton2_Click(object sender, EventArgs e)
    {
        //INSERTAR
        string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

        try
        {
            OleDbConnection CONN = new OleDbConnection(Conexion);
            OleDbCommand CMD = new OleDbCommand("Insert Into
TblDatos(Matricula,Alumno,Curso,Sexo,Fnace,Retirado,C
ostoMatricula) Values
(Matricula,Alumno,Curso,Sexo,Fnace,Retirado,CostoMatricula)
", CONN);

            //ASIGNAMOS DATOS
            CMD.Parameters.AddWithValue("Matricula", Tdato1.Text);
            CMD.Parameters.AddWithValue("Alumno", Tdato2.Text);
            CMD.Parameters.AddWithValue("Curso", Tdato3.Text);
            CMD.Parameters.AddWithValue("Sexo", Tdato4.Text);
            CMD.Parameters.AddWithValue("Fnace", Tdato5.Text);
            CMD.Parameters.AddWithValue("Retirado",
checkbox1.Checked);
            CMD.Parameters.AddWithValue("CostoMatricula",
Tdato6.Text);

            CONN.Open();
            CMD.ExecuteNonQuery();

            //DESDE AQUI RELLENA DATOS EN EL DATAGRID
            OleDbCommand CMD2 = new OleDbCommand("Select * From
TblDatos order by Alumno ", CONN);
            OleDbDataAdapter DA = new OleDbDataAdapter(CMD2);
            DataSet DS = new DataSet();
            DA.Fill(DS, "TblDatos");
            DataGridView.DataSource = DS;
            CONN.Close();
        }
        catch { }
    }
}

```

```

    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void CargarDatos()
{
    //VER TODOS
    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Select * From
TblDatos", CONN);

        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();

        //Cargando el Dataset
        DA.Fill(DS, "TblDatos");

        //Cargando el Datagridview
        DataGrid.DataSource = DS;
        DataGrid.DataMember = "TblDatos";

        CONN.Open();
        DA.Fill(DS);
        CONN.Close();
        DataGrid.DataSource = DS;
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}

```

6. GRABAR Y EJECUTAR

ACTUALIZAR REGISTROS POR CÓDIGO

1. Abrir un nuevo Proyecto
2. Estructura de la Base de Datos (Access)

AULA.MDB

TBLDATOS

* Matricula	Numérico (Entero) (Campo Llave)
Alumno	Texto corto (30)
Curso	Numerico (Byte)
Sexo	Texto corto (1)
Fnace	Fecha/Hora (Fecha Corta)
Retirado	Sí/No
CostoMatricula	Moneda

(Nota: Digite la información de 5 Registros o puede utilizar la Base de datos del ejercicio anterior)

3. Apariencia del Formulario

Actualización Registros por Código

Matrícula

Estudiante

Curso

Género

4. Componentes

FORM

Name	Form1
Text	Actualización Registros por Código

BackColor (A su gusto)
StartPosition CenterScreen
WindowState Normal

BUTTON

Cantidad 3

Name Boton1
BackColor (A su gusto)
Text Siguiente

Name Boton2
BackColor (A su gusto)
Text Anterior

Name Boton3
BackColor (A su gusto)
Text Actualizar

DATAGRIDVIEW

Cantidad 1

Name	DataGrid
AllowUserToAddRows	False
AllowUserToDeleteRows	False
AllowUserToOrderColumns	True
ReadOnly	True

LABEL

Cantidad 4

Name Mensaje1
BackColor (A su gusto)
Text Matricula

Name Mensaje2
BackColor (A su gusto)
Text Estudiante

Name Mensaje3
BackColor (A su gusto)
Text Curso

Name Mensaje4
BackColor (A su gusto)
Text Género

TEXTBOX

Cantidad	4
Name	Tdato1
BackColor	(A su gusto)
Enabled	False
Text	Vacio
Name	Tdato2
BackColor	(A su gusto)
Text	Vacio
Name	Tdato3
BackColor	(A su gusto)
Text	Vacio
Name	Tdato4
BackColor	(A su gusto)
Text	Vacio

5. CODIGO

```
using System.Data.OleDb;

public partial class Form1 : Form
{
    DataTable Mitabla = new DataTable();
    int FI;

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        CargarDatos();
        MostrarDatos(0);
    }

    private void Boton1_Click(object sender, EventArgs e)
    {
        //SIGUIENTE
        //Posicionarse en la fila siguiente
        int UFI = Mitabla.Rows.Count - 1;
        FI = FI + 1;
        if (FI > UFI)
            FI = UFI;
    }
}
```

```

        //MessageBox.Show(Convert.ToString(FI));
        MostrarDatos(FI);
    }

private void Boton2_Click(object sender, EventArgs e)
{
    //ANTERIOR
    //Posicionarse en la fila anterior
    int UFI = 1;
    FI = FI - 1;
    if (FI < UFI)
        FI = 1;
    //MessageBox.Show(Convert.ToString(FI));
    MostrarDatos(FI);
}

private void Boton3_Click(object sender, EventArgs e)
{
    //ACTUALIZAR
    string Alu, Cur;
    Char Sex;

    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

    try
    {
        //ASIGNAMOS DATOS
        Alu = Tdato2.Text;
        Cur = Tdato3.Text;
        Sex = Convert.ToChar(Tdato4.Text);

        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Update TblDatos
        Set Alumno= Alu, Curso=Cur, Sexo=Sex Where
        Matricula=" + Tdato1.Text, CONN);

        //EN ESTE EJERCICIO SOLO SE COLOCO TRES CAMPOS
        //PARA ACTUALIZAR, EL LECTOR PUEDE AMPLIAR

        CMD.Parameters.AddWithValue("Alumno", Alu);
        CMD.Parameters.AddWithValue("Curso", Cur);
        CMD.Parameters.AddWithValue("Sexo", Sex);

        CONN.Open();
        CMD.ExecuteNonQuery();

        //DESDE AQUI RELLENA DATOS EN EL DATAGRID

        OleDbCommand CMD2 = new OleDbCommand("Select * From
        TblDatos order by Matricula", CONN);
        OleDbDataAdapter DA = new OleDbDataAdapter(CMD2);
        DataSet DS = new DataSet();
        DA.Fill(DS, "TblDatos");
    }
}

```

```

        //REFRESCAMOS LOS DATOS ACTUALIZADOS
        Mitabla.Clear();
        DA.Fill(Mitabla);

        DataGrid.DataSource = DS;
        CONN.Close();
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void CargarDatos()
{
    //VER TODOS
    string Conexion = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\AULA.Mdb";

    try
    {
        OleDbConnection CONN = new OleDbConnection(Conexion);
        OleDbCommand CMD = new OleDbCommand("Select * From
        TblDatos Order By Matricula", CONN);

        OleDbDataAdapter DA = new OleDbDataAdapter(CMD);
        DataSet DS = new DataSet();

        //Cargando el Dataset
        DA.Fill(DS, "TblDatos");

        //Cargando el Datagridview
        DataGrid.DataSource = DS;
        DataGrid.DataMember = "TblDatos";
        Mitabla = new DataTable();

        //Llenar la tabla con los datos indicados
        DA.Fill(Mitabla);

        CONN.Open();
        //DA.Fill(DS);
        CONN.Close();
        DataGrid.DataSource = DS;
    }

    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void MostrarDatos(int fila)

```

```
{
    //ASIGNACIÓN DE CAMPOS HACIA LOS TEXTBOX
    int Ufila = Mitabla.Rows.Count - 1;
    if ((fila < 0) || (Ufila < 0))
        return;

    DataRow MiCampo = Mitabla.Rows[fila];
    Tdato1.Text = Convert.ToString(MiCampo[0]);
    Tdato2.Text = Convert.ToString(MiCampo[1]);
    Tdato3.Text = Convert.ToString(MiCampo[2]);
    Tdato4.Text = Convert.ToString(MiCampo[3]);
}
}
```

6. GRABAR Y EJECUTAR

HA TENER PRESENTE. FORMATOS

¿QUÉ ES VISUAL C#?

Es un lenguaje orientado a objetos y eventos que soporta encapsulación, herencia y polimorfismo. Forma parte de Visual Studio .NET y compartiendo el entorno de desarrollo con Visual C++ .NET, Microsoft Visual Basic .NET, etc.

DECLARACIÓN DE VARIABLES

Existen diversas formas para declarar una variable o inicializar la misma con un valor constante pero antes conoceremos la forma básica:

- **Estructura:**
 - “tipo de dato” “nombre de variable” ;
- **Ejemplos:**
 - **Byte** num;
 - **int** x;
 - **string** a,b;

INICIALIZAR VARIABLES

- Para inicializar se realiza de la siguiente manera:
 - **int** x;
x = 0;
 - **int** x = 0;

DECLARAR UNA CONSTANTE

- Una constante es aquella que almacena un valor que no es posible cambiar durante la ejecución del programa.

Se recomienda que el nombre de una constante sea escrito en su totalidad en **MAYÚSCULAS**.

const double PI = 3.14159265;

TIPOS DE DATOS O VARIABLES

Tipo de Dato	Escritura en C#	Asignación de almacenamiento nominal	Intervalo de valores
Boolean	bool Boolean	En función de la plataforma de implementación	True o False
Byte	Byte Byte	1 byte	0 a 255 (sin signo)
Char (carácter individual)	Char Char	2 bytes	0 a 65535 (sin signo)
Date	DateTime	8 bytes	0:00:00 (medianoche) del 1 de enero de 0001 a 11:59:59 p.m. del 31 de diciembre de 9999.
Decimal	decimal Decimal	16 bytes	0 a +/- 79.228.162.514.264.337.593.543.950.335
Double (punto flotante de precisión doble)	double Double	8 bytes	-1,79769313486231570E+308 a -4,94065645841246544E-324 para los valores negativos; 4,94065645841246544E-324 a 1,79769313486231570E+308 para los valores positivos
Integer	int	4 bytes	-2.147.483.648 a 2.147.483.647 (con signo)
Long (entero largo)	long	8 bytes	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 (9,2...E+18 †) (con signo)
Object	object (clase)	4 bytes en plataforma de 32 bits 8 bytes en plataforma de 64 bits	Cualquier tipo puede almacenarse en una variable de tipo Object

SByte	sbyte SByte	1 byte	-128 a 127 (con signo)
Short (entero corto)	int16	2 bytes	-32.768 a 32.767 (con signo)
Single (punto flotante de precisión simple)	Single	4 bytes	-3,4028235E+38 a -1,401298E-45 † para los valores negativos; 1,401298E-45 a 3,4028235E+38 † para los valores positivos
String (longitud variable)	String (clase)	En función de la plataforma de implementación	0 a 2.000 millones de caracteres Unicode aprox.
UInteger	UInt32	4 bytes	0 a 4.294.967.295 (sin signo)
ULong	UInt64	8 bytes	0 a 18.446.744.073.709.551.615 (1,8...E+19 †) (sin signo)
User-Defined (estructura)	(hereda de ValueType)	En función de la plataforma de implementación	Cada miembro de la estructura tiene un intervalo de valores determinado por su tipo de datos y es independiente de los intervalos de valores correspondientes a los demás miembros.
UShort	UInt16	2 bytes	0 a 65.535 (sin signo)

OPERADORES

C# dispone de los siguientes operadores matemáticos:

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
%	Residuo (resto de la división entera)
+	Concatenación de Cadenas

OPERADORES ARITMÉTICOS

Los operadores aritméticos son (*, /, %, +, -) y producen resultados numéricos.

Operadores String

El operador de concatenación (+) String requiere operadores String y producen resultados String.

Operadores Comparativos

Los operadores Comparativos son (>, <, >=, <=, ==, !=) y producen un resultado lógico (True or False).

Operadores Lógicos

Los operadores Lógicos son (!, &&, ||) y producen un resultado lógico (True or False) y se utilizan en conjunto con los operadores comparativos.

Operadores Abreviados

C# incorpora nuevos operadores abreviados que facilitan la escritura de código, por ejemplo:

`x = x + 1` 'Antes escribía

`x += 1` 'Ahora puede escribir

Operadores Abreviados
<code>+=</code>
<code>-=</code>
<code>*=</code>
<code>/=</code>

Precedencia de Operadores

A continuación se muestra el orden de precedencia/prioridad/jerarquía de los operadores en C#

Precedencia de Operadores
<code>()</code>
<code>^</code>
<code>-</code>
<code>* /</code>
<code>%</code>
<code>+ -</code>

Sentencias de Control

A manera de resumen recordaremos que el ciclo For se utiliza cuando se conoce de antemano el número de iteraciones a ejecutarse. While se utiliza cuando no se conoce de antemano el número de iteraciones a ejecutarse. Si al menos debe ejecutarse 1 iteración es necesario utilizar Do-While con la expresión condicional al final del ciclo. Evite ciclos infinitos, para ello asegúrese de que estos ciclos tengan una expresión condicional de salida.

Las condiciones de los ciclos mencionados, pueden también hacer uso de los operadores lógicos:

Operadores Lógicos	Descripción
&&	Las 2 expresiones deben ser verdaderas
	Alguna de las 2 expresiones es verdadera
!	Negación del resultado de la expresión

CondicionaL IF

Esta estructura permite ejecutar condicionalmente una o más sentencias y puede escribirse de dos formas. La primera ocupa sólo una línea y tiene la forma siguiente:

```
if (condición) (sentencia1) else (sentencia2);
```

La segunda es más general y se muestra a continuación:

```
if (condición) {  
    sentencia(s); }  
else {  
    sentencia(s); }
```

ARREGLOS

Concepto: Los arreglos se utilizan para almacenar un conjunto de variables, que sean del mismo tipo de dato, y todas estas bajo un mismo nombre.

Declaración en el código

Para crear un arreglo se debe en primer lugar declararlo como cualquier otra variable, con las siguientes consideraciones:

- Colocar un par de corchetes [] luego del tipo de dato.
- Inicializar el arreglo llamando al constructor de la clase tipo de dato a través de la sentencia “new”.
- Indicar la cantidad de elementos que contendrá el arreglo (entre []) en el tipo de dato invocado.

Observe los ejemplos:

Ejemplo 1: La sentencia siguiente crea un arreglo de 16 elementos de tipo integer.	<pre>int[] miArreglo = new int[16];</pre>
Ejemplo 2: La sentencia siguiente crea un arreglo de 11 elementos de tipo String.	<pre>string[] miArreglo = new string[11];</pre>

Acceder a los datos de un Arreglo

Para acceder a los datos de un arreglo o vector, debemos hacerlo mediante el índice o número del elemento del mismo, colocandolo entre [].

ARREGLOS DE DOS DIMENSIONES (MATRICES)

Concepto: Una matriz es un conjunto de elementos contiguos, todos del mismo tipo, que comparten un nombre común, ubicados en un espacio de dos dimensiones, a los que se puede acceder por la posición (índice) que ocupa cada uno de ellos dentro de la matriz (fila, columna). Esta disposición permitirá escribir código más pequeño y simple, ya que se pueden establecer bucles mediante el número del índice. Cada elemento es una variable que puede contener un dato numérico o una cadena de caracteres, dependiendo esto del tipo de matriz. La representación de las matrices se hace mediante variables suscritas o de subíndices.

Declaración de una Matriz

La declaración de una matriz especifica el nombre de la matriz, el número de filas y columnas de la lista y el tipo de éstos.

Ejemplo:

```
string[,] matriz = new string[5,8];    (matriz de 5 filas y 8 columnas de
                                     tipo string)

byte[,] numeros = new byte[3,3];    (matriz de 3 filas y 3 columnas de tipo
                                     byte)
```

REFERENCIA

Birnios, M. (2002). Microsoft Visual Basic .Net Guia del Programador. MP Ediciones S.A.

Cerezo, Y., Pañalba, O., & Caballero, R. (2007). *Iniaciación a la Programación en C# un enfoque práctico*. Madrid: Delta Publicaciones.

Canal Visual Basic;. (20 de Febrero de 2010). *Visual Basic*. Obtenido de Manuales: <http://www.canalvisualbasic.net/manual-net/vb-net/>

Duran, L. (2007). Bases de datos con Visual Basic. Marcombo.

Foros Aprender a programar. (21 de Enero de 2011). *Foros Aprender a programar*; .
Obtenido de Visual Basic: <http://www.aprenderaprogramar.com/foros/>

Foros del Web. (10 de Marzo de 2010). *Foros*. Obtenido de Conectar a Bases de datos:
<http://www.forosdelweb.com/>

Medina, P. (2012). *Guía práctica en VB Net*. Ambato.

Microsoft. (10 de Noviembre de 2014). *Microsoft Developer Network*. Obtenido de
<http://msdn.microsoft.com/es-es/library/aa287558%28v=vs.71%29.aspx>

Microsoft. (11 de Marzo de 2009). *Msdn.Microsoft*. Obtenido de Libreria:
<http://msdn.microsoft.com/es-es/library/2x7h1hfk%28v=vs.80%29.aspx>

El hecho de que C# sea una herramienta joven en evolución hace que se requieran guías para su explotación y aprendizaje. Esta obra ofrece una nueva forma de involucrar a los estudiantes de forma activa en el área de programación, a través del manejo eficiente de lenguaje C# en entorno visual.

Esta guía de prácticas ha sido diseñada como herramienta de auto aprendizaje, contiene un variado conjunto de ejercicios prácticos que ilustran el uso de controles, fortalece su manejo y crea una experiencia autónoma enriquecedora en el estudiante, además de constituirse en un recurso didáctico para docentes de áreas afines.

ISBN- 978-9942-21-146-0

