



Guía de Prácticas en Visual Basic Net

Ing. Mg. Patricio Medina

Primera Edición

GUÍA DE PRÁCTICAS



EN VB NET

Ing. Mg. Patricio Medina

Profesor de la Universidad Técnica de Ambato

Profesor de la Pontificia Universidad Católica del Ecuador Sede Ambato

Asesor Informático

Primera Edición

Editorial Independiente

Autor: Ing. Mg. Patricio Medina

Capacitación realizada en: Cuba, Brasil y Ecuador

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

ISBN- 978-9942-21-145-3

Certificado Instituto de Propiedad Intelectual: QUI-039697

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 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 los 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

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

PRÓLOGO

Visual Basic.Net 2010, 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 Visual Basic.Net 2010 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 y profesores, así como de incentivar a muchos en aras de conseguir una docencia de mayor calidad, tal como lo ha logrado Patricio Medina por medio de muchas herramientas como ésta.

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 Visual Basic.Net 2010.

Bienvenida esta nueva herramienta docente y estudiante, y a disfrutarla haciendo que el trabajo del profesor se refleje en el aprendizaje del estudiante.

Ing. Mg. Galo López Sevilla
DIRECTOR ESCUELA INGENIERÍA EN SISTEMAS
PUCESA - 2014

DEDICATORIA

Todo mundo quiere tener un amigo, pocos se toman la molestia de ser uno.
(Anónimo)

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

- A mi querida familia, pilar de todos mis sueños y esfuerzos
- A mis Estudiantes porque de ustedes aprendo cada día mucho más.
- Mis buenos amigos de EIS PUCESA, gracias por tan valiosa amistad.

Patricio

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	31
JUEGO DEL 7	34
CASINO	37
ENLACES	40
CARRERAS.....	42
MANEJO DE TECLAS	45
MANEJO DE TECLAS 2	48
USO DE VARIABLES	52
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	76
USO DE LISTVIEW BÁSICO	80
LISTVIEW CON ARCHIVO PLANO	83
TREEVIEW	87
FUNCIONES Y PROCEDIMIENTOS LOCALES.....	90
FUNCIONES Y PROCEDIMIENTOS GLOBALES	95
MOVIMIENTO DE FIGURA	99
MASCARAS Y ARCHIVOS DE TEXTO	102
CONTROL DE ERRORES Y AYUDA.....	105
CONECCIÓN CON BASE DE DATOS ACCESS POR CÓDIGO.....	110
CONSULTAS POR CÓDIGO	114
CONSULTAS POR COMPONENTES	121

INSERTAR REGISTROS POR CÓDIGO	127
ACTUALIZAR REGISTROS POR CÓDIGO.....	132
HA TENER PRESENTE. FORMATOS	138
REFERENCIA	145

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	2
BackColor	(A su gusto)
ForeColor	(A su gusto)
Font	(Tamaño 24, Negrita)

Name	Mensaje2
Text	“Esteban y Ariel”
BorderStyle	3
BackColor	(A su gusto)
ForeColor	(A su gusto)
Font	(Tamaño 10, Negrita)

Size	244; 43
TextAlign	MiddleCenter
Name	Mensaje3
Text	“Esteban y Ariel”
BorderStyle	1
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	1
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	3

4. CODIGO

Public Class Form1

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
Dim Aux As Byte  
Aux = MsgBox("Desea Abandonar", MsgBoxStyle.YesNo, "Aviso")  
If (Aux = 6) Then
```

```
        End
    End If
End Sub

Private Sub Lista_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles Lista.SelectedIndexChanged

    Select Case Lista.SelectedIndex
        Case 0 : Me.BackColor = Color.Red
        Case 1 : Me.BackColor = Color.Green
        Case 2 : Me.BackColor = Color.Blue

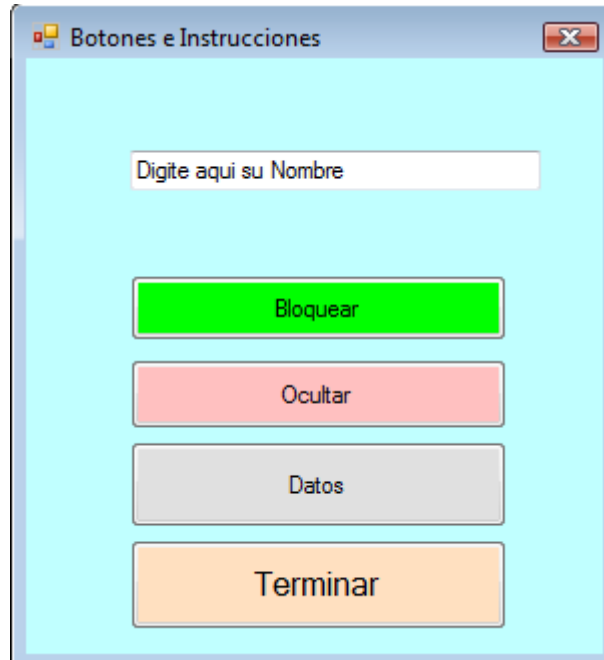
    End Select
End Sub

End Class
```

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 Class Form1

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

```
        If Tdato.Enabled Then
            Boton1.Text = "Desbloquear"
            Tdato.Enabled = False
        Else
            Boton1.Text = "Bloquear"
            Tdato.Enabled = True
        End If
    End Sub
```

Private Sub Boton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton2.Click

```
        If Tdato.Visible Then
            Boton2.Text = "Mostrar"
```

```

        Tdato.Visible = False
    Else
        Boton2.Text = "Ocultar"
        Tdato.Visible = True
    End If
End Sub

Private Sub Boton3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton3.Click

    Dim Nom As String
    REM Para un ingreso de datos rápido
    Nom = InputBox("Nombre de tu Novia")

    REM formas de visualizar rápidas
    REM Forma 1
    MsgBox("Mensaje 1: Tu novia se llama: " & Nom)

    REM Forma 2
    MessageBox.Show("Mensaje 2: Tu novia se llama: " & Nom)

End Sub

Private Sub Boton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton4.Click

    Application.Exit()

End Sub

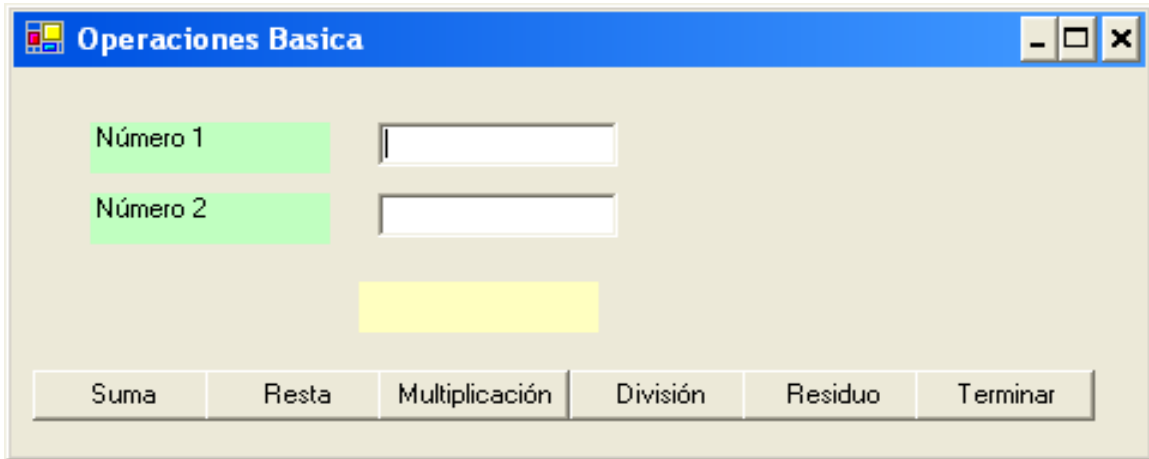
End Class

```

5. GRABAR Y EJECUTAR

OPERACIONES BÁSICAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



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

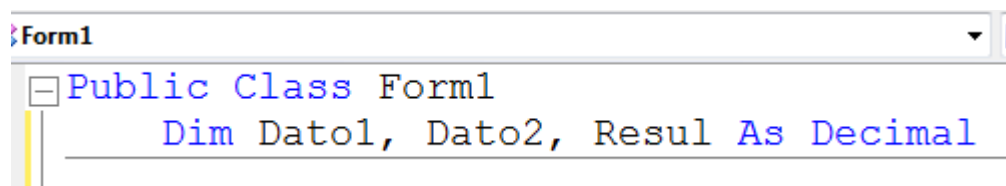
Name	Boton6
Text	Terminar

4. CODIGO

Public Class Form1

Rem Son las Variables para todo el Formulario

Dim Dato1, Dato2, Resul As Decimal



```
#Region "COMENTARIOS"
REM Tener presente que se puede colocar entre una REGION con un nombre
REM para poder extender o contraer todo lo que esta dentro de esta
REM sección
#End Region
```

```
Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles Boton1.Click
```

```
    REM Suma
    Dim Num1, Num2, Aux As Decimal
```

```

        Try
            Num1 = Convert.ToDecimal(Tdato1.Text)
            Num2 = Convert.ToDecimal(Tdato2.Text)
        Catch ex As Exception
            Num1 = 0 : Num2 = 0
        End Try
        Aux = Num1 + Num2
        Mensaje3.Text = String.Format("{0:###0.00}", Aux)
    End Sub

    Private Sub Boton2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Boton2.Click
        REM Resta
        Call Ingreso()
        Result = Dato1 - Dato2
        Mensaje3.Text = String.Format("{0:###0.00}", Result)
    End Sub

    Private Sub Boton3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Boton3.Click
        REM Multiplicacion
        Call Ingreso()
        Result = Dato1 * Dato2
        Mensaje3.Text = String.Format("{0:###0.00}", Result)
    End Sub

    Private Sub Boton4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Boton4.Click
        REM División
        Call Ingreso()
        If Dato2 <= 0 Then
            MsgBox("Número 2 es Incorrecto")
        Else
            Result = Dato1 / Dato2
        End If
        Mensaje3.Text = String.Format("{0:###0.00}", Result)
    End Sub

    Private Sub Boton5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Boton5.Click
        REM Residuo de la división
        Call Ingreso()
        If Dato2 <= 0 Then
            MsgBox("Número 2 es Incorrecto")
        Else
            Result = Dato1 Mod Dato2
        End If
        Mensaje3.Text = String.Format("{0:###0.00}", Result)
    End Sub
    Private Sub Ingreso()
        Try
            Dato1 = Convert.ToDecimal(Tdato1.Text)
            Dato2 = Convert.ToDecimal(Tdato2.Text)
        Catch ex As Exception
            Dato1 = 0 : Dato2 = 0
        End Try
    End Sub

```

```
Private Sub Boton6_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles Boton6.Click
```

```
End
```

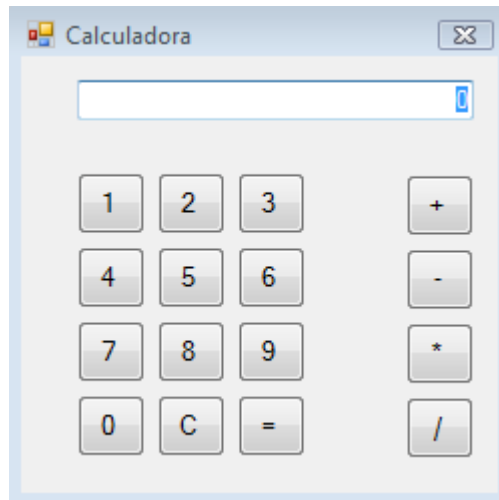
```
End Sub
```

```
END CLASS
```

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 Class Form1

```
Dim Guarda As Integer = 0
Dim Signo As String
```

```
Private Sub Btn1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn1.Click
```

```
    Calcular(1)
```

```
End Sub
```

```
    Private Sub Btn2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn2.Click
```

```
        Calcular(2)
```

```
    End Sub
```

```
    Private Sub Btn3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn3.Click
```

```
        Calcular(3)
```

```
    End Sub
```

```
    Private Sub Btn4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn4.Click
```

```
        Calcular(4)
```

```
    End Sub
```

```
    Private Sub Btn5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn5.Click
```

```
        Calcular(5)
```

```
    End Sub
```

```
    Private Sub Btn6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn6.Click
```

```

        Calculular(6)
    End Sub

    Private Sub Btn7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn7.Click
        Calculular(7)
    End Sub

    Private Sub Btn8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn8.Click
        Calculular(8)
    End Sub

    Private Sub Btn9_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn9.Click
        Calculular(9)
    End Sub

    Private Sub Btn10_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn10.Click
        Calculular(0)
    End Sub

    Private Sub BtnC_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnC.Click
        TxtPantalla.Text = "0"
    End Sub

    Private Sub BtnSuma_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnSuma.Click
        Operador("+")
    End Sub

    Private Sub BtnResta_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnResta.Click
        Operador("-")
    End Sub

```

```

    Private Sub BtnMulti_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnMulti.Click

```

```

        Operador("*")

```

```

    End Sub

```

```

    Private Sub BtnDiv_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnDiv.Click

```

```

        Operador("/")

```

```

    End Sub

```

```

    Private Sub BtnIgual_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BtnIgual.Click

```

```

Select Case Signo

```

```

    Case "+" : TxtPantalla.Text = CStr(Guarda +
CInt(TxtPantalla.Text))

```

```

    Case "-" : TxtPantalla.Text = CStr(Guarda -
CInt(TxtPantalla.Text))

```

```

    Case "*" : TxtPantalla.Text = CStr(Guarda *
CInt(TxtPantalla.Text))

```

```

    Case "/" : TxtPantalla.Text = CStr(Guarda /
CInt(TxtPantalla.Text))

```

```

    End Select

```

```

End Sub

```

```

Sub Calcular(ByVal Valor)

```

```

    TxtPantalla.Text = CStr(((CInt(TxtPantalla.Text) *
10) + Valor))

```

```

End Sub

```

```

Sub Operador(ByVal Operador)

```

```

    Signo = Operador

```

```

    Guarda = CInt(TxtPantalla.Text)

```

```

    TxtPantalla.Text = "0"

```

```

    End Sub

```

```

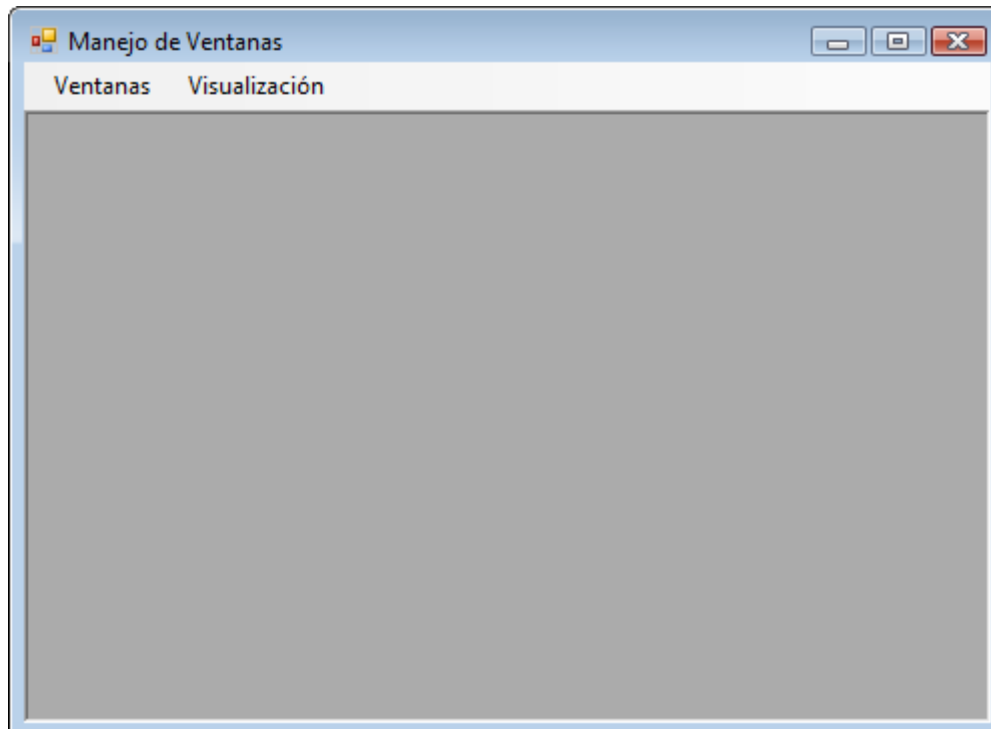
End Class

```

5. GRABAR Y EJECUTAR

MANEJO DE FORMULARIOS POR VARIABLES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



3. Componentes

FORM

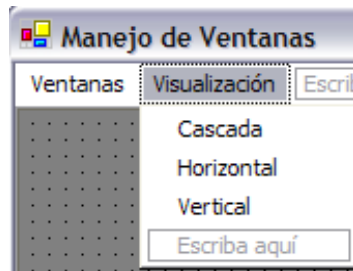
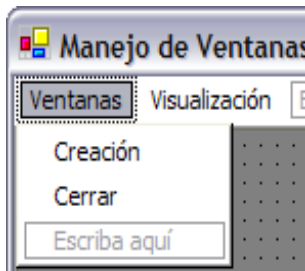
Name	Form1
Text	Manejo de Ventanas por Variables
BackColor	(A su gusto)
Icon	(A su gusto)
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Maximized
IsMdiContainer	True

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 Class Form1

```
Dim Posi As Byte
```

```
Private Sub CreacionToolStripMenuItem_Click(ByVal  
sender As System.Object, ByVal e As  
System.EventArgs) Handles  
CreacionToolStripMenuItem.Click
```

```
REM CREACION
```

```
Dim MiNuevoForm As New Form  
MiNuevoForm.MdiParent = Me  
Posi += 1  
MiNuevoForm.Text = "Ventana" &  
Convert.ToString(Posi)  
MiNuevoForm.Show()
```

```
End Sub
```

```
Private Sub CerrarToolStripMenuItem_Click(ByVal  
sender As System.Object, ByVal e As  
System.EventArgs) Handles  
CerrarToolStripMenuItem.Click
```

```
REM CERRAR  
Me.Close()
```

```
End Sub
```

```
Private Sub CascadaToolStripMenuItem_Click(ByVal  
sender As System.Object, ByVal e As
```

**System.EventArgs) Handles
CascadaToolStripMenuItem.Click**

```
        REM CASCADA  
        Me.LayoutMdi (MdiLayout.Cascade)  
End Sub
```

**Private Sub HorizontalToolStripMenuItem_Click (ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
HorizontalToolStripMenuItem.Click**

```
        REM HORIZONTAL  
        Me.LayoutMdi (MdiLayout.TileHorizontal)  
End Sub
```

**Private Sub VerticalToolStripMenuItem_Click (ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
VerticalToolStripMenuItem.Click**

```
        REM VERTICAL  
        Me.LayoutMdi (MdiLayout.TileVertical)  
End Sub
```

**Private Sub Form1_Load (ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load**

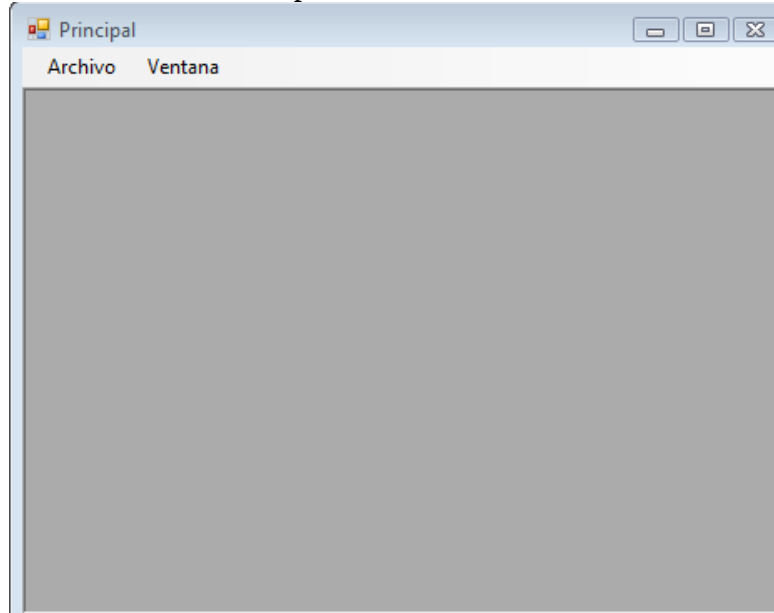
```
    REM INICIALIZACION VARIABLES CUANDO SE CARGA EL PROYECTO  
    Posi = 0
```

```
End Sub  
End Class
```

5. GRABAR Y EJECUTAR

FORMULARIOS MDI POR HERENCIA

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



3. Componentes

FORM

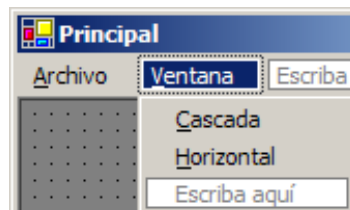
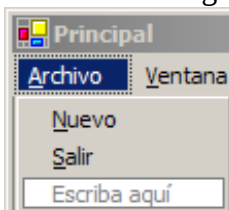
Name	FormPrincipal
Text	Principal
MaximizeBox	True
MinimizeBox	True
StartPosition	CenterScreen
WindowState	Maximized
IsMdiContainer	True

MENUSTRIP

Cantidad 1

Name MiMenu

Diseñe el siguiente esquema, teniendo seleccionado el MainMenu



4. CODIGO

Public Class FormPrincipal

Private Sub NuevoToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NuevoToolStripMenuItem.Click

REM OPCION NUEVO
Dim MiForm As FormSecundario = New FormSecundario
REM FORMSECUNDARIO ES EL NOMBRE DEL OTRO FORMULARIO
MiForm.MdiParent = Me
MiForm.show()

End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalirToolStripMenuItem.Click

REM OPCION SALIR
Me.Close()
End Sub

Private Sub CascadaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CascadaToolStripMenuItem.Click
REM OPCION CASCADA

Me.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade)

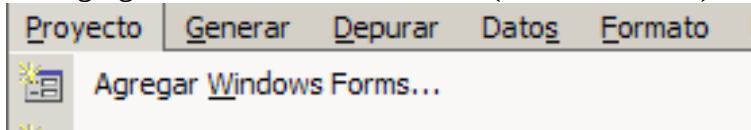
End Sub

Private Sub HorizontalToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HorizontalToolStripMenuItem.Click
REM OPCION HORIZONTAL

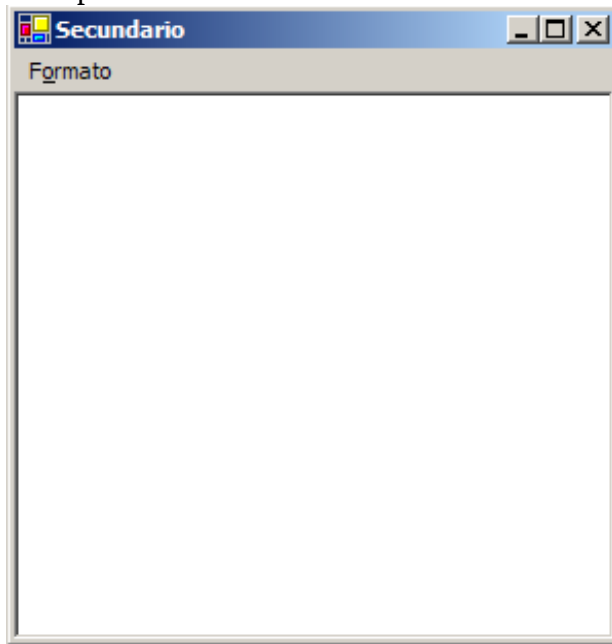
Me.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal)
End Sub

End Class

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

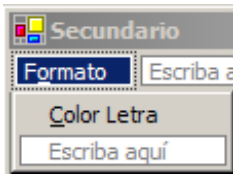
MENUSTRIP

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 LETRAMENUIITEM

8. CODIGO

EVENTOS FORMULARIO SECUNDARIO

```
Public Class FormSecundario
```

```
    Private Sub LetraMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
LetraMenuItem.Click
```

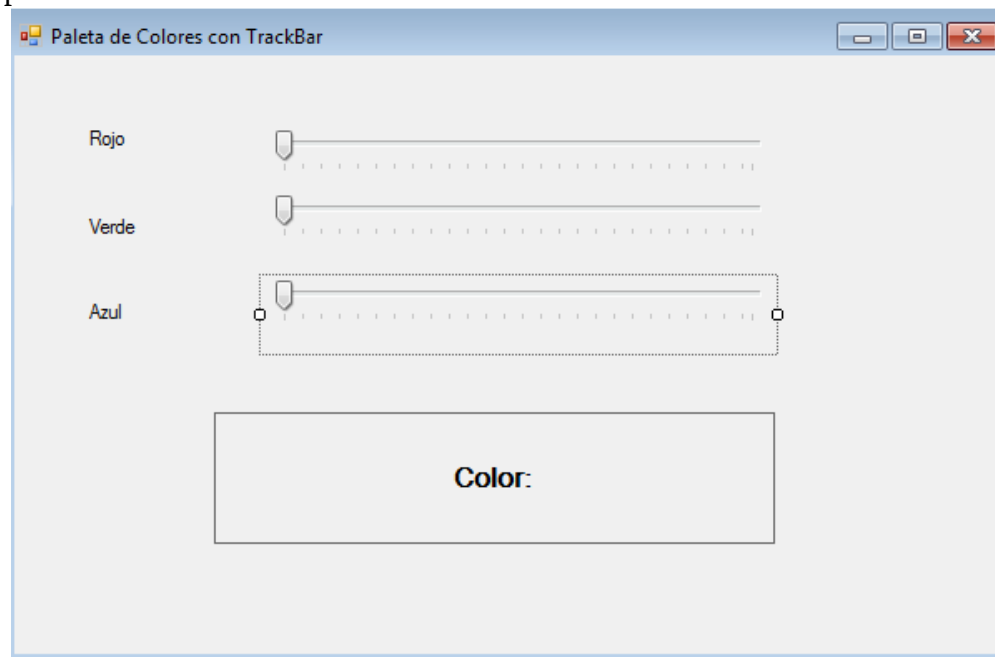
```
    If LetraMenuItem.Checked Then  
        LetraMenuItem.Checked = False  
        Tdato.ForeColor = System.Drawing.Color.Black  
    Else  
        LetraMenuItem.Checked = True  
        Tdato.ForeColor = System.Drawing.Color.Blue  
    End If
```

```
    End Sub  
End Class
```

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)
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	Opción 2
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 Class Form1

Sub MiPaleta()

REM Tomamos los valores del TrackBar

Dim CR, CV, CA As Byte

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

End Sub

```

Private Sub TrackBar1_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TrackBar1.Scroll
    REM ROJO
    Call MiPaleta()
End Sub

Private Sub TrackBar2_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TrackBar2.Scroll
    REM VERDE
    Call MiPaleta()
End Sub

Private Sub TrackBar3_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TrackBar3.Scroll
    REM AZUL
    Call MiPaleta()
End Sub
End Class

```

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 Class Form1
```

```
Private Sub TrackBarH_Scroll(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TrackBarH.Scroll
```

```
    REM Horizontal  
    Foto.Width = TrackBarH.Value  
    MensajeH.Text = TrackBarH.Value
```

```
End Sub
```

```
Private Sub TrackBarV_Scroll(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TrackBarV.Scroll
```

```
    REM Vertical  
    Foto.Height = TrackBarV.Value  
    MensajeV.Text = TrackBarV.Value
```

```
End Sub
```

```
Private Sub UpDown1_ValueChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
UpDown1.ValueChanged
```

```
    REM Horizontal  
    Foto.Width = UpDown1.Value
```

```
End Sub
```

```
Private Sub UpDown2_ValueChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
UpDown2.ValueChanged
```

```
    REM Vertical  
    Foto.Height = UpDown2.Value
```

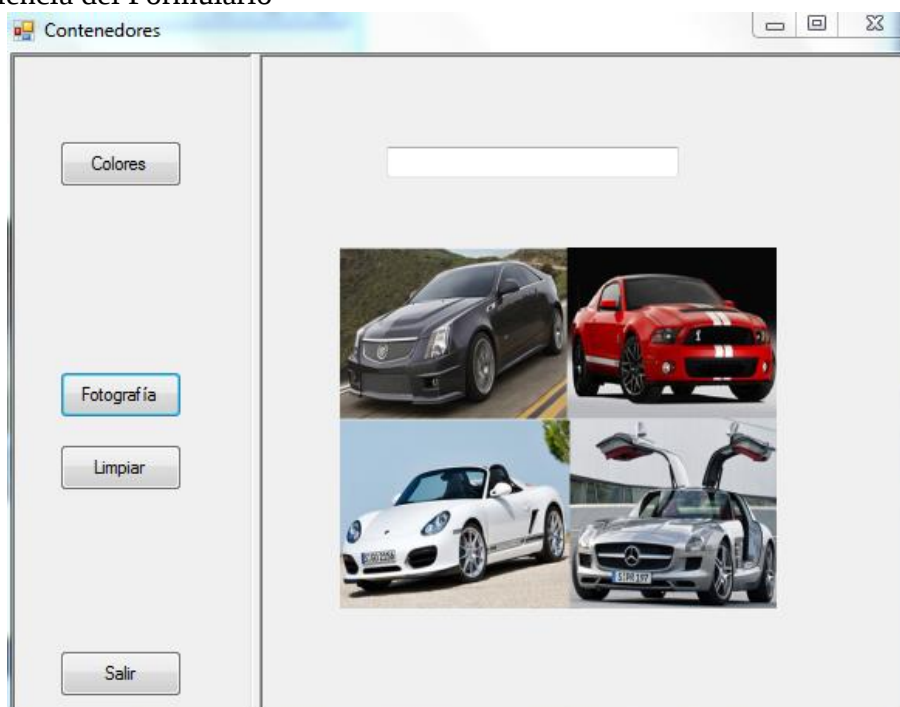
```
End Sub
```

```
End Class
```

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

4. CODIGO

Public Class Form1

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

REM Colores

If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

TDato.BackColor = ColorDialog1.Color

SplitContainer1.Panel1.BackColor = ColorDialog1.Color

End If

End Sub

Private Sub Boton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton2.Click

REM fotografía

PictureBox1.Visible = True

End Sub

Private Sub Boton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton3.Click

TDato.BackColor = Color.White

PictureBox1.Visible = False

End Sub

Private Sub Boton4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton4.Click

End

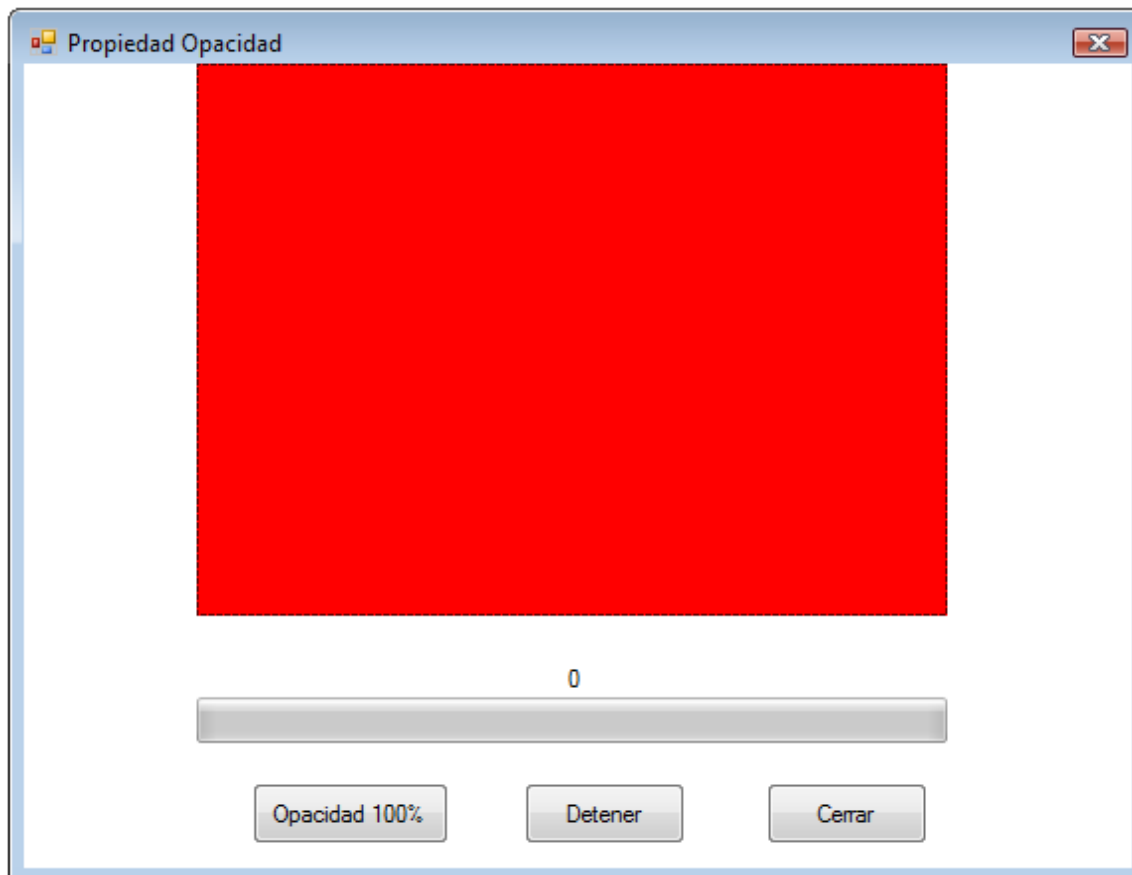
End Sub

End Class

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 Class Form1

```
Dim Aumenta As Decimal = 0.0
```

```
Private Sub Form1_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    Mensaje.Text = Me.Opacity * 100
```

```
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
    Mensaje.Text = "0"
```

```
    ProgressBar1.Value = 0
```

```
    Me.Opacity = 0
```

```
    Aumenta = 0
```

```
    Reloj.Enabled = True
```

```
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click
```

```
    Me.Close()
```

```
End Sub
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
    Reloj.Enabled = False
```

```
End Sub
```

```
Private Sub Reloj_Tick(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Reloj.Tick
```

```
    If (Aumenta > 0.5) Then
```

```
        Reloj.Stop()
```

```
        Reloj.Enabled = False
```

```
    Else
```

```
        Me.Opacity = 0.5 + Aumenta
```

```
        Aumenta = Aumenta + 0.01
```

```
        Mensaje.Text = Val(Me.Opacity) * 100
```

```
        ProgressBar1.Value = Val(Mensaje.Text)
```

```
    End If
```

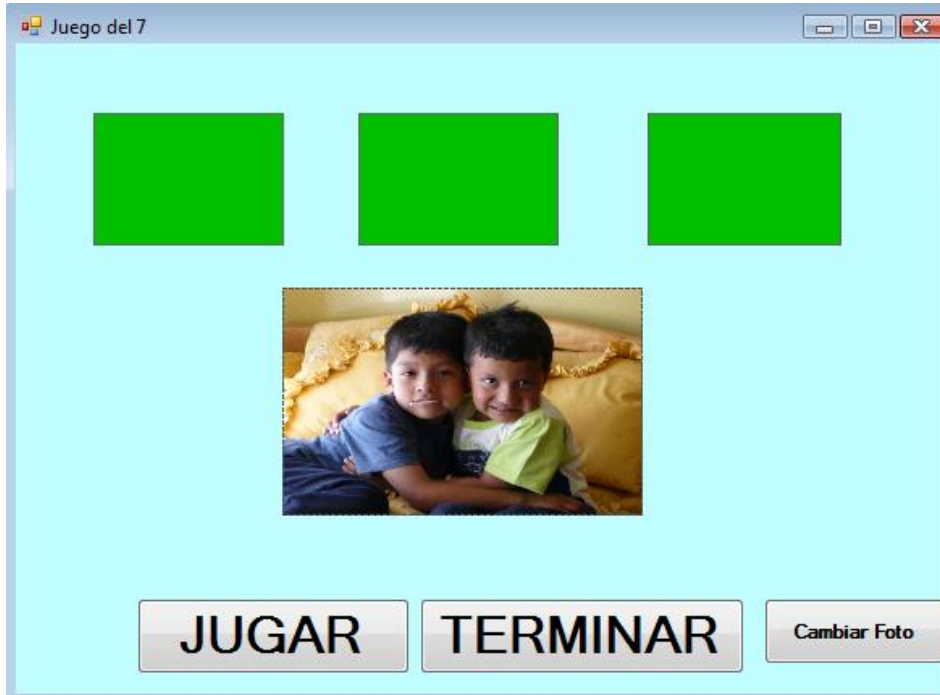
```
End Sub
```

```
End Class
```

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)
BorderStyle	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

Public Class Form1

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

REM Este juego es similar a las maquinas traga

```
REM monedas si sale un 7 en cualquier  
REM posicion GANAS
```

```
        Foto1.Visible = False  
        Mensaje1.Text = CStr(Int(Rnd() * 10))  
        Mensaje2.Text = CStr(Int(Rnd() * 10))  
        Mensaje3.Text = CStr(Int(Rnd() * 10))  
If (Mensaje1.Text = "7") Or (Mensaje2.Text = "7")  
Or (Mensaje3.Text = "7") Then  
        Beep()  
        Foto1.Visible = True  
End If  
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click
```

```
Foto1.Image =  
System.Drawing.Image.FromFile("C:\Fotos\Peques.JPG")
```

```
REM Lo que esta entre comillas es la ruta donde se  
REM encuentra otra Imágen
```

```
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

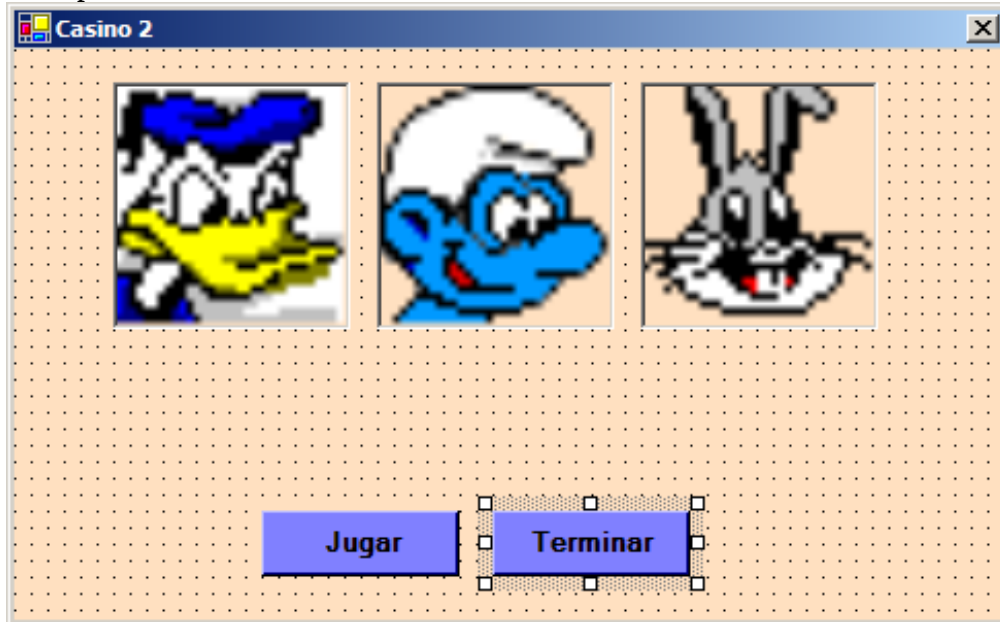
```
        End  
    End Sub
```

```
End Class
```

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 Class Form1

```
Dim Vector(3) As String
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
    Dim N1, N2, N3 As Byte
    Randomize()
    Do
        N1 = Int(Rnd() * 10)
    Loop While ((N1 < 1) Or (N1 > 3))

    Do
        N2 = Int(Rnd() * 10)
    Loop While ((N2 < 1) Or (N2 > 3))

    Do
        N3 = Int(Rnd() * 10)
```

```

        Loop While ((N3 < 1) Or (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) And (N1 = N3) Then
    MessageBox.Show("FELICITACIONES GANASTE")
Else
    MessageBox.Show("LO SIENTO INTENTA NUEVAMENTE")
End If

End Sub

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

Vector(1) = "C:\Fotos\Casino\Foto1.Ico"
Vector(2) = "C:\Fotos\Casino\Foto2.Ico"
Vector(3) = "C:\Fotos\Casino\Foto3.Ico"
End Sub

Private Sub Boton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton2.Click

        Me.Close()
End Sub

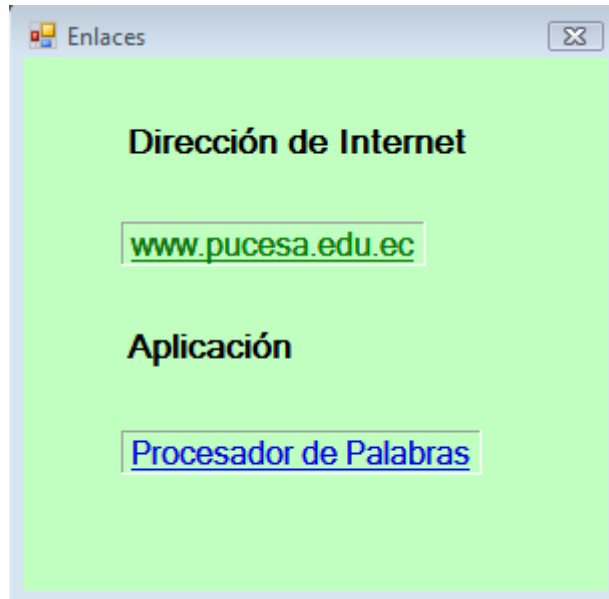
End Class

```

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
LinkColor	(Escoger Verde)
Text	www.pucesa.edu.ec
VisitedLinkColor	(Escoger Negro)

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

4. CODIGO

EVENTOS

Public Class Form1

```
Private Sub Enlace1_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
Enlace1.LinkClicked
```

```
Enlace1.LinkVisited = True
```

```
System.Diagnostics.Process.Start("http://www.pucesa.edu.ec")
```

```
REM Tenga presente el cambio de colores en este Link
End Sub
```

```
Private Sub Enlace2_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs)
Handles Enlace2.LinkClicked
```

```
Enlace2.LinkVisited = True
```

```
System.Diagnostics.Process.Start("winword.exe")
End Sub
End Class
```

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 Moto1

Image (Seleccione un archivo de imagen)

SizeMode 2

Name Moto2

Image (Seleccione un archivo de imagen)

SizeMode 2

4. CODIGO

Public Class Form1

```
Dim Mov As Double  
Dim Posi1 As Integer = 30  
Dim Posi2 As Integer = 30
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
    Reloj.Enabled = True
```

```
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
    Reloj.Enabled = False  
    Moto1.Location = New Point(30, 80)  
    Moto2.Location = New Point(30, 160)
```

```
End Sub
```

```
Sub Movimiento()
```

```
    Mov = Math.Round(Rnd() * 10)
```

```
End Sub
```

```
Private Sub Form2_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    Randomize()
```

```
End Sub
```

```
Private Sub Reloj_Tick(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Reloj.Tick
```

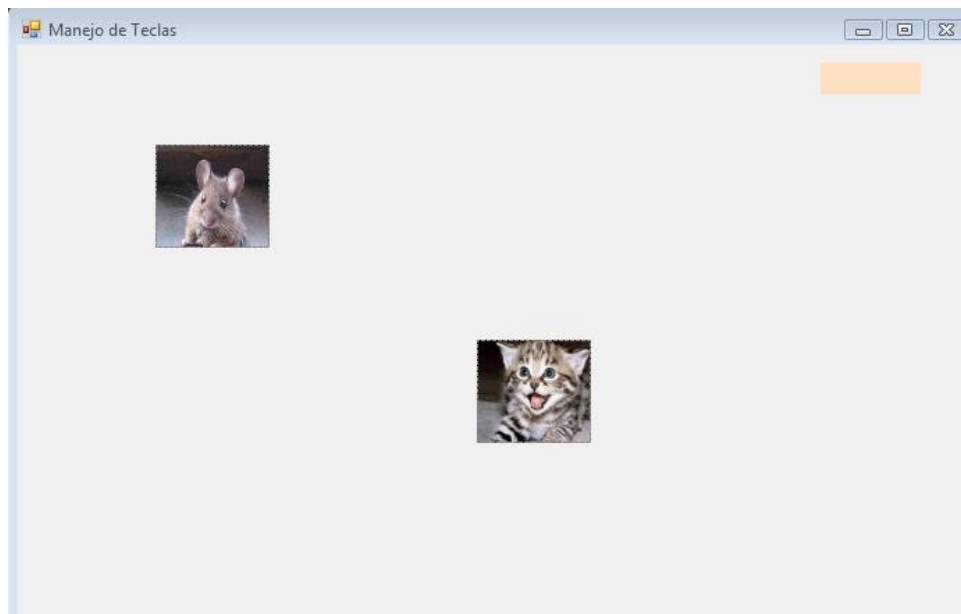
```
    For X As Integer = 1 To 2  
        If (Posi1 <= 450) And (Posi2 <= 450) Then  
            Movimiento()  
            If (Mov > 0) And (Mov < 10) Then  
                If X = 1 Then  
                    Posi2 = Posi2 + Mov  
                Moto2.Location = New Point(Posi2, 160)  
            Else  
                Posi1 = Posi1 + Mov  
                Moto1.Location = New Point(Posi1, 80)  
            End If  
        End If  
    Else  
        Reloj.Stop()  
        MsgBox("Ganó la moto: " & X)  
        Posi1 = 30  
        Posi2 = 30  
    End If  
Next  
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click  
    Me.Close()  
End Sub  
End Class
```

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 GATO
SizeMode	2

Name	Foto2
Image	(Seleccione un archivo de imagen) Ejemplo RATÓN
SizeMode	2

4. CODIGO

Public Class Form1

```
Dim Fi As Integer = 200
Dim Co As Integer = 300
Dim Puntos As Integer = 0
```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

```
Me.Width = 700
Me.Height = 600
Randomize()
```

End Sub

Private Sub Form1_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown

```
Select Case e.KeyCode
    Case 37 : Fi = Fi - 3 REM IZQUIERDA
    Case 39 : Fi = Fi + 3 REM DERECHA
    Case 38 : Co = Co - 3 REM ARRIBA
    Case 40 : Co = Co + 3 REM ABAJO
End Select
Fotol.Location = New Point(Fi, Co)
Call Ubica()
```

End Sub

Sub Ubica()

```
Dim F, C, Ancho, Largo As Integer
F = Foto2.Left : C = Foto2.Top
Ancho = Foto2.Width : Largo = Foto2.Height
```

```
If (Fotol.Left) >= F And (Fotol.Left <= F + Ancho)
And (Fotol.Top) >= C And (Fotol.Top <= C + Largo)
Then
```

```
Puntos += 1
```

```
Mensaje1.Text = Puntos
```

```
Foto2.Location = New Point(GENERA(450), GENERA(700))
```

```
End If
```

End Sub

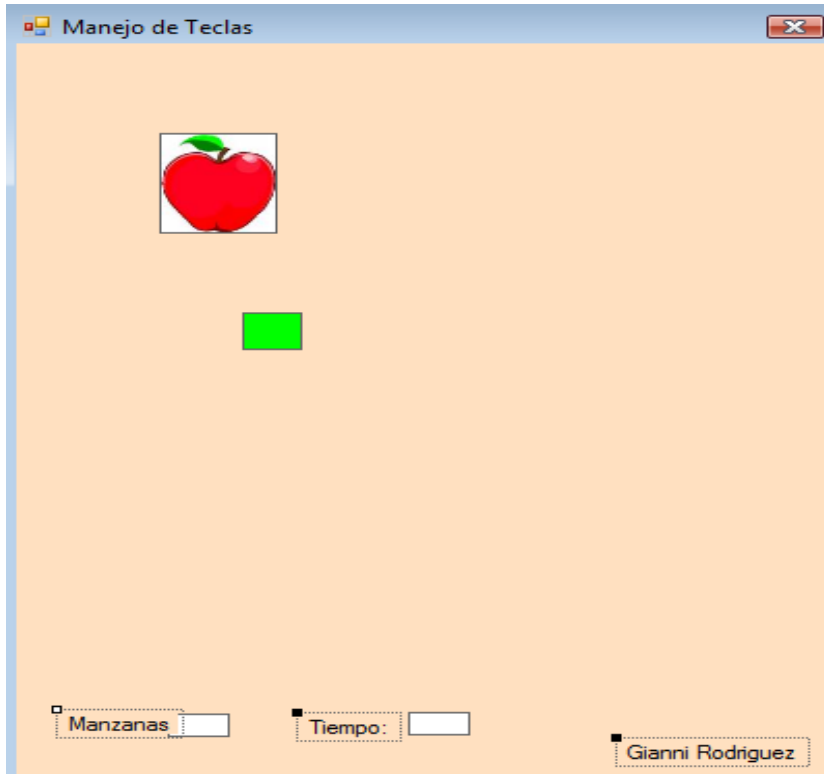
```
Function GENERA(ByVal Aux As Integer) As Integer  
    Dim N2 As Integer  
    Do  
        N2 = Int(Rnd() * 1000)  
    Loop While ((N2 < 1) Or (N2 > Aux))  
    Return N2  
End Function
```

End Class

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 Class Form1

```
Public Respuesta As Integer = 0
Public A As Integer = 20
```

```
Private Sub Form1_KeyDown1(ByVal sender As Object,  
ByVal e As System.Windows.Forms.KeyEventArgs)  
Handles Me.KeyDown
```

```
    Select Case e.KeyCode
        Case Keys.Up
            ImagenSnake.Top -= 10
        Case Keys.Down
            ImagenSnake.Top += 10
        Case Keys.Left
            ImagenSnake.Left -= 10
        Case Keys.Right
            ImagenSnake.Left += 10
    End Select
```

```

If ImagenSnake.Top > ImagenAlimento.Top And
ImagenSnake.Top < (ImagenAlimento.Top +
ImagenAlimento.Height) Then

If ImagenSnake.Left > ImagenAlimento.Left And
ImagenSnake.Left < (ImagenAlimento.Left +
ImagenAlimento.Width) Then

    Dim Alto, Ancho As Double
    Do
        Alto = Int(Rnd() * 1000)
        Loop While ((Alto < 1) Or (Alto > 300))

    Do
        Ancho = Int(Rnd() * 1000)
        Loop While ((Ancho < 1) Or (Ancho > 300))
        ImagenAlimento.Top = Alto
        ImagenAlimento.Left = Ancho
        Respuesta = Respuesta + 1
    End If
End If

Mensaje2.Text = Respuesta

If A <= 0 Then
    ImagenSnake.Visible = False
    MessageBox.Show("HAS COMIDO " & Respuesta & "
    MANZANAS EN 20 Segs", "RESULTADO!",
    MessageBoxButtons.OK)

    Timer1.Enabled = False
    Timer1.Stop()

End If

End Sub

```

```

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

```

```

    ImagenSnake.Top = 200
    ImagenSnake.Left = 200
    Randomize()

```

```

Dim Alto, Ancho As Double
Do
    Alto = Int(Rnd() * 1000)
Loop While ((Alto < 1) Or (Alto > 300))
Do
    Ancho = Int(Rnd() * 1000)
Loop While ((Ancho < 1) Or (Ancho > 300))
ImagenAlimento.Top = Alto
ImagenAlimento.Left = Ancho

```

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

```

If (A <= 20) And (A <> 0) Then
    A = A - 1
End If
Mensaje4.Text = A

```

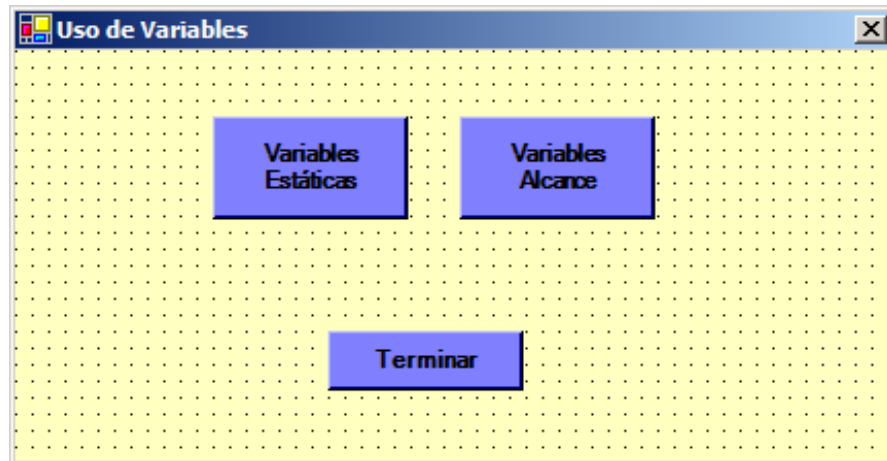
End Sub

End Class

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 MiVar As String ' Variable a nivel de solución.Prueba 3)
'Friend MiVar As String ' Variable a nivel de proyecto.Prueba 2)
```

```
'Dim MiVar As String ' Variable a nivel de módulo. (Prueba 2)
```

```
REM Probar de la siguiente manera
```

```
REM 1.- Ejecute y pruebe variable de procedimiento
```

```
REM 2.- Ejecute y pruebe variable de módulo, ponga
```

```
REM comentario a la declaración de variable anterior
```

```
REM 3.- Ejecute y pruebe variable de proyecto, ponga
```

```
REM comentario a la declaración de variable anterior
```

```
REM 4.- Ejecute y pruebe variable de solución, ponga
```

```
REM comentario a la declaración de variable anterior
```

```
Private Sub Boton2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton2.Click
```

```
    Dim MiVar As String ' Variable a nivel de procedimiento
```

```
    MiVar = "Estoy practicando alcance de Variables."
```

```
    MessageBox.Show(MiVar)
```

```
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Boton3.Click
```

```
    Application.Exit()
```

```
End Sub
```

```
Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Boton1.Click
```

```
    Static Pulsa As Integer
```

```
    Pulsa += 1
```

```
    MessageBox.Show("Número de veces que dio clic: " _
    & Pulsa)
```

```
REM Fijarse en el tipo de dato Static que no pierde su valor
```

```
REM Notar como se da continuidad de línea con la forma espacio y
signo _
```

```
End Sub
```

```
End Class
```

PD. Hacer la ejecución retirando y poniendo como comentarios a cada LÍNEA DE PRUEBA de la declaración de variables y analizar su funcionamiento.

5. GRABAR Y EJECUTAR

DECLARAR ESTRUCTURAS

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



The screenshot shows a Windows application window titled "Estructuras". The window has a yellow background. At the top center, there is a large blue button labeled "Mostrar". Below this button, there are four labels in blue boxes: "Marca", "Modelo", "Precio", and "Fecha". Each label is followed by a white text input field.

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 8, 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)

Name	Mensaje3
Text	Modelo
Backcolor	(A su gusto)

Name	Mensaje4
Text	Precio
Backcolor	(A su gusto)

Name	Mensaje5
Text	Fecha
Backcolor	(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 Class Form1

Rem Declaración de la estructura

Public Structure Carro

```

    Dim marca As String
    Dim modelo As String
    Dim precio As Single
    Dim fechafabrica As Date
End Structure

```

```
Private Sub Button1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button1.Click
```

```
    Dim Mivarcarro As Carro  
    REM asigno a miembros de estructura  
    Mivarcarro.marca = "Nissan"  
    Mivarcarro.modelo = "Cabina simple"  
    Mivarcarro.precio = 8500  
    Mivarcarro.fechafabrica = #5/29/2001#
```

```
    REM asigno a Textbox desde miembros de estructura  
    Tdato1.Text = mivarcarro.marca  
    Tdato2.Text = mivarcarro.modelo  
    Tdato3.Text = mivarcarro.precio  
    Tdato4.Text = mivarcarro.fechafabrica
```

```
End Sub
```

```
End Class
```

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 8, Negrita)

LABEL

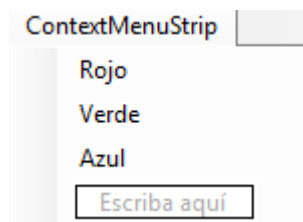
Cantidad	2
Name	Mensaje1
Text	Estaban y Ariel
Backcolor	(A su gusto)
Font	(Tamaño 24, Negrita)
Name	Mensaje2
Text	Presione Botón derecho
Forecolor	(A su gusto)

TEXTBOX

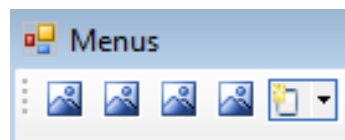
Cantidad	1
Name	Tdato1
Text	www.yahoo.es

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

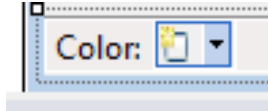
Cantidad	1
Name	WebBrowser1

STATUSSTRIP

Cantidad 1

Name StatusStrip1

Contenido (Diseñar una opción con elemento StatusLabel)



(Propiedad Text escriba **Color:**)

4. CODIGO

```
Public Class Form1
```

```
Private Sub RojoToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RojoToolStripMenuItem.Click
```

```
    REM Rojo
```

```
    Mensaje1.BackColor = Color.Red
```

```
End Sub
```

```
Private Sub VerdeToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
VerdeToolStripMenuItem.Click
```

```
    REM Verde
```

```
    Mensaje1.BackColor = Color.Green
```

```
End Sub
```

```
Private Sub AzulToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
AzulToolStripMenuItem.Click
```

```
    REM Azul
```

```
    Mensaje1.BackColor = Color.Blue
```

```
End Sub
```

```
Private Sub ToolStripButton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
ToolStripButton3.Click
```

```
    REM Rojo
```

```
    Mensaje1.BackColor = Color.Red
```

```
    ToolStripStatusLabel1.Text = "Color: ROJO"
```

```
End Sub
```

```

Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click
    REM Verde
    Mensaje1.BackColor = Color.Green
    ToolStripStatusLabel1.Text = "Color: VERDE"
End Sub

Private Sub ToolStripButton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton2.Click
    REM Azul
    Mensaje1.BackColor = Color.Blue
    ToolStripStatusLabel1.Text = "Color: AZUL"
End Sub
Private Sub ToolStripButton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton4.Click
    REM Abrir Navegador
    Process.Start("http://google.com.ec")
End Sub

Private Sub Boton1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton1.Click
    WebBrowser1.Navigate(TDato1.Text)
    WebBrowser1.Visible = True
    Me.Text = TDato1.Text
End Sub

Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    REM Espacio de la Página Web Oculta
    WebBrowser1.Visible = False
End Sub
REM COMPLETE EL TOOLSTRIP CON BOTONES PARA EL NAVEGADOR
REM RETROCEDER, AVANZAR, DETENER, REFRESCAR
End Class

```

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	ESTEBAN Y ARIEL
Font	(Negrita, Tamaño 24)
TextAlign	MiddleCenter

GROUPBOX

Cantidad	2
Name	Contenedor1
BackColor	(A su gusto)
Text	Estilo

Name	Contenedor2
BackColor	(A su gusto)
Text	Colores

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 Class Form1
```

```
    Private Sub Opcion1_Click(ByVal sender As  
Object, ByVal e As System.EventArgs) Handles  
Opcion1.Click
```

```
        REM ROJO
```

```
        Mensaje1.BackColor = Color.Red
```

```
    End Sub
```

```
    Private Sub Opcion2_Click(ByVal sender As  
Object, ByVal e As System.EventArgs) Handles  
Opcion2.Click
```

```
        REM VERDE
```

```
        Mensaje1.BackColor = Color.Green
```

```
    End Sub
```

```
    Private Sub Opcion3_Click(ByVal sender As  
Object, ByVal e As System.EventArgs) Handles  
Opcion3.Click
```

```
        REM AZUL
```

```
        Mensaje1.BackColor = Color.Blue
```

```
    End Sub
```

```
    Private Sub Opcion4_Click(ByVal sender As  
Object, ByVal e As System.EventArgs) Handles  
Opcion4.Click
```

```
        REM PALETA DE COLORES
```

```
    If DialogoColor.ShowDialog() =  
Windows.Forms.DialogResult.OK Then
```

```
        Mensaje1.BackColor = DialogoColor.Color
```

```
    End If
```

```
    End Sub
```

```

Private Sub Casilla1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Casilla1.CheckedChanged
    REM NEGRILLA
    If Casilla1.Checked Then
        Mensaje1.Font = New Font("Microsoft Sans Serif", 24, FontStyle.Bold)

    Else
        Mensaje1.Font = New Font("Microsoft Sans Serif", 24, FontStyle.Regular)
    End If
End Sub

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click
    Me.Close()
End Sub

Private Sub Boton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    REM DIALOGO FUENTE
    If DialogoFuente.ShowDialog() = Windows.Forms.DialogResult.OK Then

        Mensaje1.Font = DialogoFuente.Font
    End If

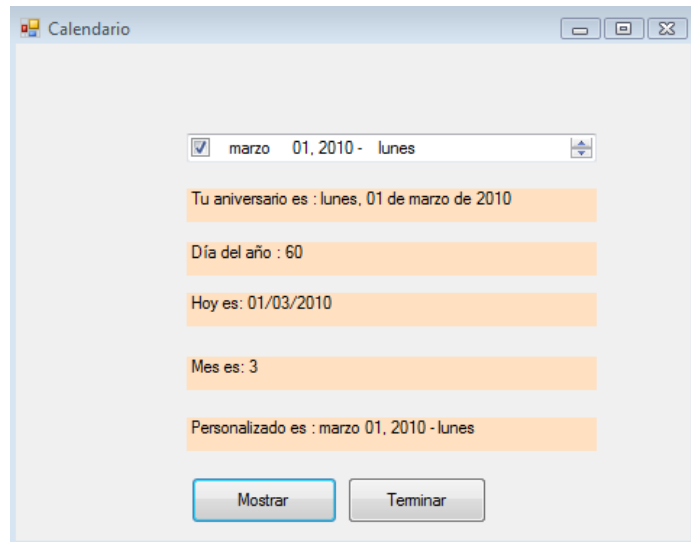
    REM Otra forma sin poner el objeto a probar
    'Dim FD As New FontDialog
    'FD.ShowDialog()
    'Mensaje1.Font = FD.Font
End Sub
End Class

```

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 Class Form1

**Private Sub Boton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton1.Click**

```
Mensaje1.Text = "Tu aniversario es : " & Calendario.Text
```

```
REM Día del año a partir del 1 de Enero  
Mensaje2.Text = "Día del año : " &  
Calendario.Value.DayOfYear.ToString
```

```
REM 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
```

```
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
    Me.Close()
```

```
End Sub
```

```
End Class
```

5. GRABAR Y EJECUTAR

USO DE LISTBOX

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

The screenshot shows a Windows application window titled "ListBox". Inside the window, there is a text input field with the placeholder text "Digite un Nombre". Below this field is a button labeled "Agregar". To the left of the window is a list box labeled "Lista1", and to the right is a list box labeled "Lista2". Between these two list boxes are two buttons: "Pasar Todo" and "Pasar Uno". Below "Lista1" are three orange rectangular bars. At the bottom of the window are three buttons: "Borrar Item", "Borrar Todo", and "Insertar".

3. Componentes

FORM

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

BUTTON

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

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 Class Form1

```
Private Sub Form1_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    Lista1.Items.Add("Lorena")  
    Lista1.Items.Add("Patricio")  
    Lista1.Items.Add("Esteban")  
    Lista1.Items.Add("Ariel")
```

```
End Sub
```

```
Private Sub Lista1_SelectedIndexChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Lista1.SelectedIndexChanged
```

```
Mensaje2.Text = "Posición : " &  
Lista1.SelectedIndex
```

```
Mensaje3.Text = "Nombre : " & Lista1.SelectedItem
```

```
Mensaje4.Text = "Elementos: " & Lista1.Items.Count  
End Sub
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
    Lista1.Items.Add(Tdato.Text)  
    Tdato.Text = ""  
    Tdato.Focus()
```

```
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
    Try
```

```
        Lista1.Items.RemoveAt(Lista1.SelectedIndex)
```

```
        Catch ex As Exception  
            MessageBox.Show("Primero seleccione un Nombre")  
        End Try
```

```
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click
```

```
    Lista1.Items.Clear()  
End Sub
```

```
Private Sub Boton4_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton4.Click
```

```
    Lista1.Items.Insert(2, "RICARDO")  
End Sub
```

```
Private Sub Boton5_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton5.Click
```

```
    Dim Posi As Byte  
    For Posi = 0 To (Lista1.Items.Count - 1)  
  
Lista2.Items.Add(Lista1.Items.Item(Posi))  
        Next Posi  
End Sub
```

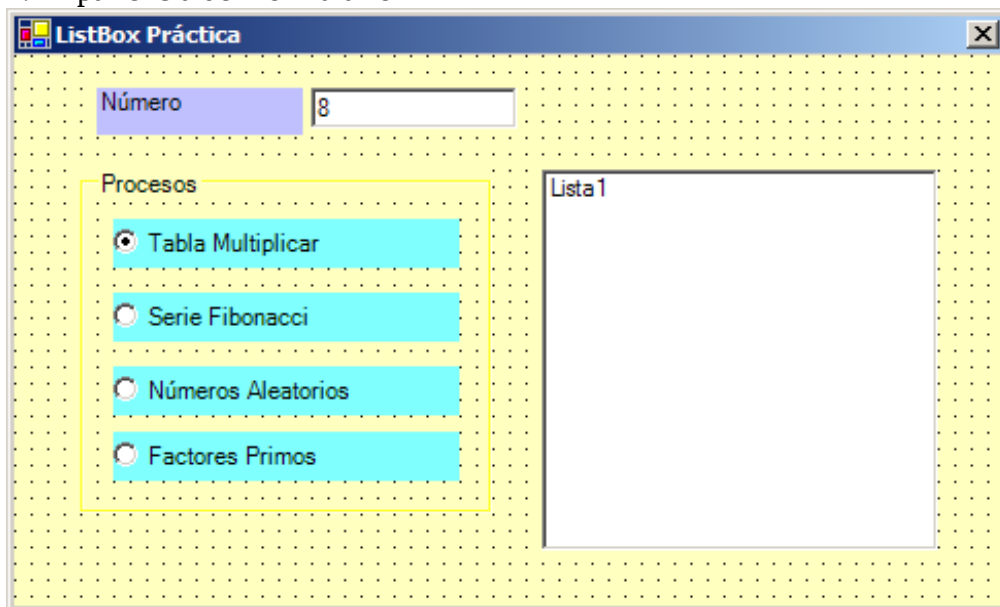
```
Private Sub Boton6_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton6.Click
```

```
    Try  
        Lista2.Items.Add(Lista1.SelectedItem)  
  
        Catch ex As Exception  
            MessageBox.Show("Primero seleccione un Nombre")  
        End Try  
End Sub  
End Class
```

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 Class Form1

**Private Sub Opcion1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Opcion1.CheckedChanged**

```
REM TABLA DE MULTIPLICAR
Dim NUM, I, RES As Byte
NUM = Val(Tdato.Text)
Lista1.Items.Clear()
For I = 1 To 12
    RES = NUM * I
    Lista1.Items.Add(NUM & " * " & I & " = " & RES)
Next I
```

End Sub

**Private Sub Opcion2_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Opcion2.CheckedChanged**

```
REM SERIE DE FIBONACCI
```

```

Dim A, B, AUX, X, NUM As Integer
NUM = Val(Tdato.Text)
A = 1 : B = 1
Lista1.Items.Clear()
Lista1.Items.Add(A) : Lista1.Items.Add(B)
For X = 3 To NUM
    AUX = A + B
    Lista1.Items.Add(AUX)
    A = B
    B = AUX
Next

```

End Sub

Private Sub Opcion3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Opcion3.CheckedChanged

```

REM NUMEROS ALEATORIOS
Dim NUM, X, AUX As Integer
NUM = Val(Tdato.Text)
Lista1.Items.Clear()
For X = 1 To NUM
    AUX = Rnd() * 1000
    Lista1.Items.Add(AUX)
Next

```

End Sub

Private Sub Opcion4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles opcion4.CheckedChanged

```

REM FACTORES PRIMOS DE UN NÚMERO
Dim NUM, X, AUX, RESI As Integer
NUM = Val(Tdato.Text)
X = 2
Lista1.Items.Clear()
While (NUM <> 1)
    RESI = NUM Mod X
    AUX = NUM \ X
    If (RESI = 0) Then
        Lista1.Items.Add(X)
        NUM = AUX
    Else
        X += 1
    End If
End While

```

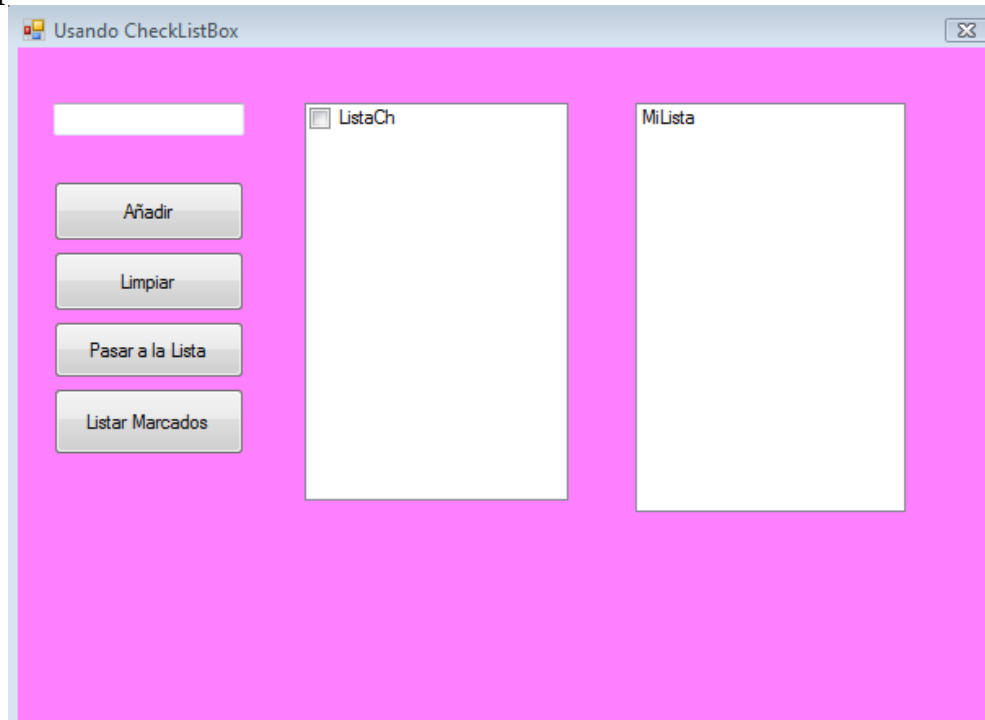
End Sub

End Class

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)

CHECKLISTBOX

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 Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

```
REM AÑADIMOS NOMBRES POR DEFECTO
ListaCh.Items.Add("Tebby")
ListaCh.SetItemChecked(0, True)
ListaCh.Items.Add("Arito")
ListaCh.SetItemChecked(1, True)
Tdato.Focus()
```

End Sub

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

```
If Tdato.Text <> "" Then
REM VERIFICA SI YA EXISTE EL TEXTO PARA NO REPETIR
If ListaCh.CheckedItems.Contains(Tdato.Text) = False Then
ListaCh.Items.Add(Tdato.Text, CheckState.Checked)
```

```

Else
    MsgBox("Ya Existe Mercado")
End If

Tdato.Text = ""
End If
Tdato.Focus()
End Sub

Private Sub Boton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton2.Click
    REM LIMPIAR MARCADOS
    Dim n As Integer
    For n = 0 To ListaCh.Items.Count - 1
        ListaCh.SetItemChecked(n, False)
    Next
End Sub

Private Sub Boton3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton3.Click

    REM PASA ITEMS MARCADOS DEL CHECKEDLISTBOX A EL LISTBOX
    MiLista.Items.Clear()
    Dim i As Integer
    For i = 0 To ListaCh.CheckedItems.Count - 1

        MiLista.Items.Add(ListaCh.CheckedItems(i))
    Next i
End Sub

Private Sub Boton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton4.Click
    REM LISTAR MARCADOS
    Dim I As Integer
    Dim Valor As String = ""
    Dim Dato As String
    For I = 0 To Me.ListaCh.CheckedItems.Count - 1
        Dato = ListaCh.CheckedItems(I)
        Valor = Valor + Chr(13) + Dato
    Next

```

```

        MessageBox.Show(Valor)
    End Sub

    Private Sub ListaCh_ItemCheck(ByVal sender As
Object, ByVal e As
System.Windows.Forms.ItemCheckEventArgs) Handles
ListaCh.ItemCheck

    REM validar el evento de cuando un item es marcado o no
        If e.NewValue = CheckState.Unchecked Then
            If ListaCh.CheckedItems.Count = 1 Then
                MsgBox("Todavía existen items activos")
            End If
        End If
    End Sub

End Class

```

5. GRABAR Y EJECUTAR

USO DE LISTVIEW BÁSICO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

The screenshot shows a Windows application window titled "ListView". Inside the window, there is a form with the following controls:

- NOMBRE**: A text input field.
- TELEFONO**: A text input field.
- CIUDAD**: A dropdown menu.
- ESTUDIA**: A checkbox that is currently checked.

Below the form, there are two buttons: **AÑADIR** and **SALIR**.

At the bottom of the window, there is a **ListView** control. Its header row has four columns: **Nombre**, **Teléfono**, **Ciudad**, and **Estudia**. The body of the list is currently empty.

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	Estudia
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 Class Form1

**Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load**
 REM SE DEFINE QUE COLUMNAS TENDRÁ LA MATRIZ

 Me.ListView1.Columns.Add("Nombre")
 Me.ListView1.Columns.Add("Teléfono")
 Me.ListView1.Columns.Add("Ciudad")
 Me.ListView1.Columns.Add("Estudia")
 Me.ListView1.View = View.Details

End Sub

**Private Sub Boton1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton1.Click**
 REM PASAR LA INFORMACIÓN AL LISTVIEW

 If TDato1.Text.Trim = "" Then Exit Sub

 Dim VarNombre As String = Me.TDato1.Text
 Dim VarTelefono As Integer = Me.TDato2.Text
 Dim VarCiudad As String = Me.ComboBox1.SelectedItem
 Dim VarEstudia As String = IIf(Me.CheckBox1.Checked, "Si", "No")

 Dim I As ListViewItem
 I = Me.ListView1.Items.Add(VarNombre)
 I.SubItems.Add(VarTelefono)
 I.SubItems.Add(VarCiudad)
 I.SubItems.Add(VarEstudia)

End Sub

**Private Sub Boton2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton2.Click**

 REM CERRAR
 Me.Close()

End Sub

End Class

5. GRABAR Y EJECUTAR

LISTVIEW CON ARCHIVO PLANO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

Empleado	Hijos	Estado Civil	Estudia	

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)

Name	Mensaje4
Text	Estudia
BackColor	(A su gusto)

TEXTBOX

Cantidad 1

Name Tdato1
Text (Vacío)

NUMERICUPDOWN

Cantidad 1

Name Tdato2
Text (Vacío)
Minimum 0
Máximo 12

COMBOBOX

Cantidad 1

Name Tdato3
Items Soltero
Casado
Viudo
Divorciado
Union Libre

GROUPBOX

Cantidad 1

Name GroupBox1

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 Class Form1

**Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load**

 REM Cursor en el Empleado
 TDato1.Focus()
 REM Vista detallada del ListView
 ListView1.View = View.Details
 REM Cuadricula
 ListView1.GridLines = True

End Sub

 REM AÑADIR

**Private Sub Boton1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton1.Click**

 REM Se guardará información en texto plano
 Dim Resultado As String
 Resultado = IIf(Opcion1.Checked, "Si", "No")
 FileOpen(1, "D:\Empleados.txt",
OpenMode.Append)WriteLine(1, TDato1.Text, TDato2.Value,
TDato3.Text, Resultado)
 FileClose(1)
 REM agrega los datos en el list view
 REM Solo un Dato

```

        ListView1.Items.Add(TDato1.Text)

        REM Varios Datos
        Dim Vector(4) As String
        Dim VarItem As ListViewItem
        Vector(0) = TDato1.Text
        Vector(1) = TDato2.Text
        Vector(2) = TDato3.Text
        Vector(3) = Resultado
        VarItem = New ListViewItem(Vector)
        ListView1.Items.Add(VarItem)
    End Sub

    REM NUEVO
    Private Sub Boton2_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Boton2.Click
        REM Controles en blanco
        TDato1.Text = ""
        TDato2.Value = 0
        TDato3.Text = ""
        Opcion2.Checked = True
    End Sub

    Private Sub Boton3_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Boton3.Click
        REM Carga el Archivo TXT
        Shell("Notepad D:\Empleados.TXT", AppWinStyle.MaximizedFocus)
    End Sub

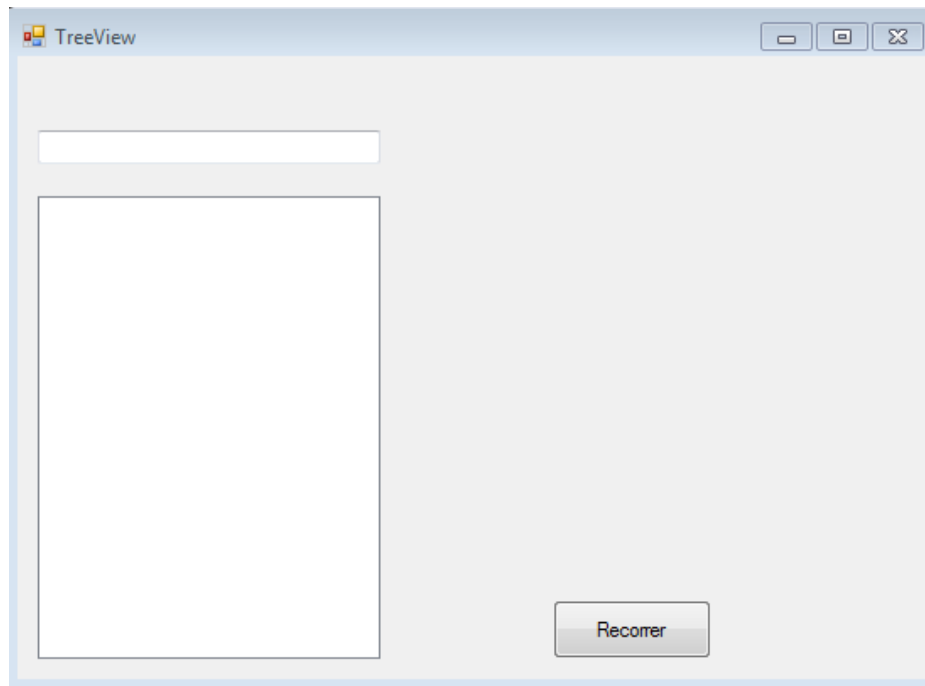
    Private Sub Boton4_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Boton4.Click
        End
    End Sub
End Class

```

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
CheckedBoxes	True

BUTTON

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

4. CODIGO

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

```
    REM GENERAREMOS VALORES ALEATORIOS
    Randomize()
    Dim TamA, TamB As Byte
With TreeView1
    For I As Integer = 1 To 4
        Dim Padre As String = "Padre" & I
        Dim aux As TreeNode = .Nodes.Add(Padre, "PADRE: " & I)
        TamA = Int(Rnd() * 10)
        For J As Integer = 1 To TamA
            Dim hijo As String = Padre & "\Hijo " & J
            Dim auxi As TreeNode = aux.Nodes.Add(hijo, "HIJO " & J)
            TamB = Int(Rnd() * 10)
            For K As Integer = 1 To TamB
                Dim Nieto As String = hijo & "\Nieto " & K

                auxi.Nodes.Add(Nieto, "NIETO:" & K)

            Next K

        Next J
    Next I
End With
End Sub
```

Private Sub TreeView1_AfterSelect(ByVal sender As System.Object, ByVal e As System.Windows.Forms.TreeViewEventArgs) Handles TreeView1.AfterSelect

```
    Tdato.Text = e.Node.FullPath
End Sub
```

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

```
    REM Se Declara una colección de nodos apartir de nuestro Treeview
    REM que vamos a recorrer
    Dim nodes As TreeNodeCollection = TreeView1.Nodes
    REM Se recorren los nodos principales
```

```

For Each n As TreeNode In nodes
    REM Se Declara un metodo para que recorra los hijos de los principales
    REM Y los hijos de los hijos....Recorrido Total en pocas palabras
    REM Para ello se envía el nodo actual para evaluar si tiene hijos
        RecorrerNodos(n)
    Next
End Sub

Private Sub RecorrerNodos(ByVal treeNode As TreeNode)
    Try
        REM Si el nodo que recibimos tiene hijos se recorrerá
        REM para luego verificar si esta o no checado
        For Each TN As TreeNode In treeNode.Nodes
            REM Se Verifica si esta marcado...
            If TN.Checked = True Then
                REM Si esta marcado mostramos el texto del nodo
                MessageBox.Show(TN.Text)
            End If
            REM Ahora hago verificacion a los hijos del nodo actual
            REM Esta iteración no acabara hasta llegar al último nodo principal
            RecorrerNodos(TN)
        Next
        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try
    End Sub

    REM FUNCIONAMIENTO
    REM PARA PROBAR EL FUNCIONAMIENTO DEBE SELECCIONAR UN NIETO Y
    REM LA INFORMACIÓN SALDRA EN EL CUADRO DE TEXTO

    REM DEBE MARCA ALGUNOS HIJOS O NIETOS PARA PROBAR EL BOTÓN
    REM APOYO EN http://ka0stj.wordpress.com/2011/06/04/recorrer-treeview-con-checkbox-e-imprimir-marcados/
End Class

```

5. GRABAR Y EJECUTAR

FUNCIONES Y PROCEDIMIENTOS LOCALES

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

The screenshot shows a Windows application window titled "Funciones y Procedimientos Locales". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main area of the window has a light blue background with a small dot pattern. At the top, there is a label "Número o Frase" in a blue box, followed by a white text input field. Below this, there is a group box titled "Procesos" with a blue border. Inside this group box, there are four radio buttons, each with a label: "Par", "Primo", "Vocal", and "Procedimiento". To the right of the "Procesos" group box is a list box titled "Lista". At the bottom of the window, there is a label "Resultado" in a blue box, followed by a white text output field.

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 Class Form1

Private Sub Opcion1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Opcion1.CheckedChanged

```
REM Par e Impar
    Dim Num As Integer
    Num = Val(Tdato.Text)
    Mensaje2.Text = PARIMPAR(Num)
```

End Sub

```
Private Sub Opcion2_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
Handles Opcion2.CheckedChanged
```

```
    REM Primo  
    Dim Num As Integer  
    Num = Val(Tdato.Text)  
    Mensaje2.Text = PRIMO(Num)
```

```
End Sub
```

```
Private Sub Opcion3_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
Handles Opcion3.CheckedChanged
```

```
    REM Vocal  
    Dim Frase As String  
    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
```

```
End Sub
```

```
Private Sub Opcion4_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
Handles Opcion4.CheckedChanged
```

```
    REM Procedimiento  
    Mensaje2.Text = "Resultado"  
    Lista.Items.Clear()  
    SEPARAR(Tdato.Text)
```

```
End Sub
```

```
Private Function PARIMPAR(ByVal Aux As Integer) As String
```

```
    Dim x As Integer  
    Dim Resul As String  
    X=Aux Mod 2  
    If (x = 0) Then  
        Resul = "PAR"  
    Else  
        Resul = "IMPAR"  
    End If
```

```
Return Resul REM Forma de Retorno 1  
End Function
```

```
Private Function PRIMO(ByVal Aux As Integer) As String  
    Dim R, sw, i As Integer  
    i = 2 : sw = 0  
    While ((i < Aux) And (sw = 0))  
        R = Aux Mod i  
        If (R = 0) Then  
            sw = 1  
        Else  
            i += 1  
        End If  
    End While  
    If (sw = 0) Then  
        PRIMO = "ES PRIMO" REM Forma de Retorno 2  
    Else  
        PRIMO = "NO ES PRIMO"  
    End If  
End Function
```

```
Private Function CONTAR(ByVal Cadena As String,  
ByVal Caracter As String) As Byte  
    REM Cuenta un Determinado Caracter  
    Dim i, Ac As Integer  
    Dim Letra As String  
    Ac = 0  
    For i = 1 To Len(Cadena)  
        Letra = UCase(Mid(Cadena, i, 1))  
        If Letra = UCase(Caracter) Then  
            Ac += 1  
        End If  
    Next i  
    Return Ac  
End Function
```

```
Private Sub SEPARAR(ByVal Cadena As String)  
    Dim i As Integer  
    Dim Letra As String  
    Lista.Items.Add("Separar en Minusculas")  
    For i = 1 To Len(Cadena)  
        Letra = LCase(Mid(Cadena, i, 1))
```

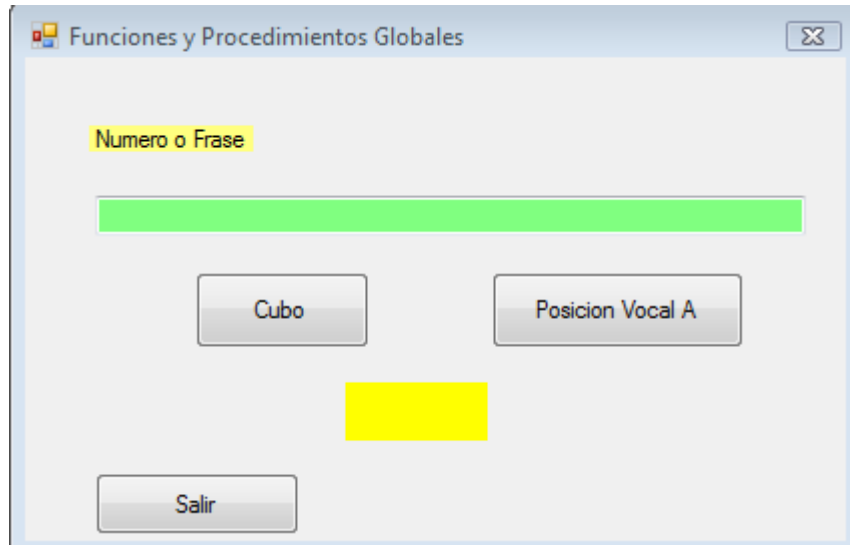
```
        Lista.Items.Add(Letra)
    Next i
End Sub
```

End Class

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

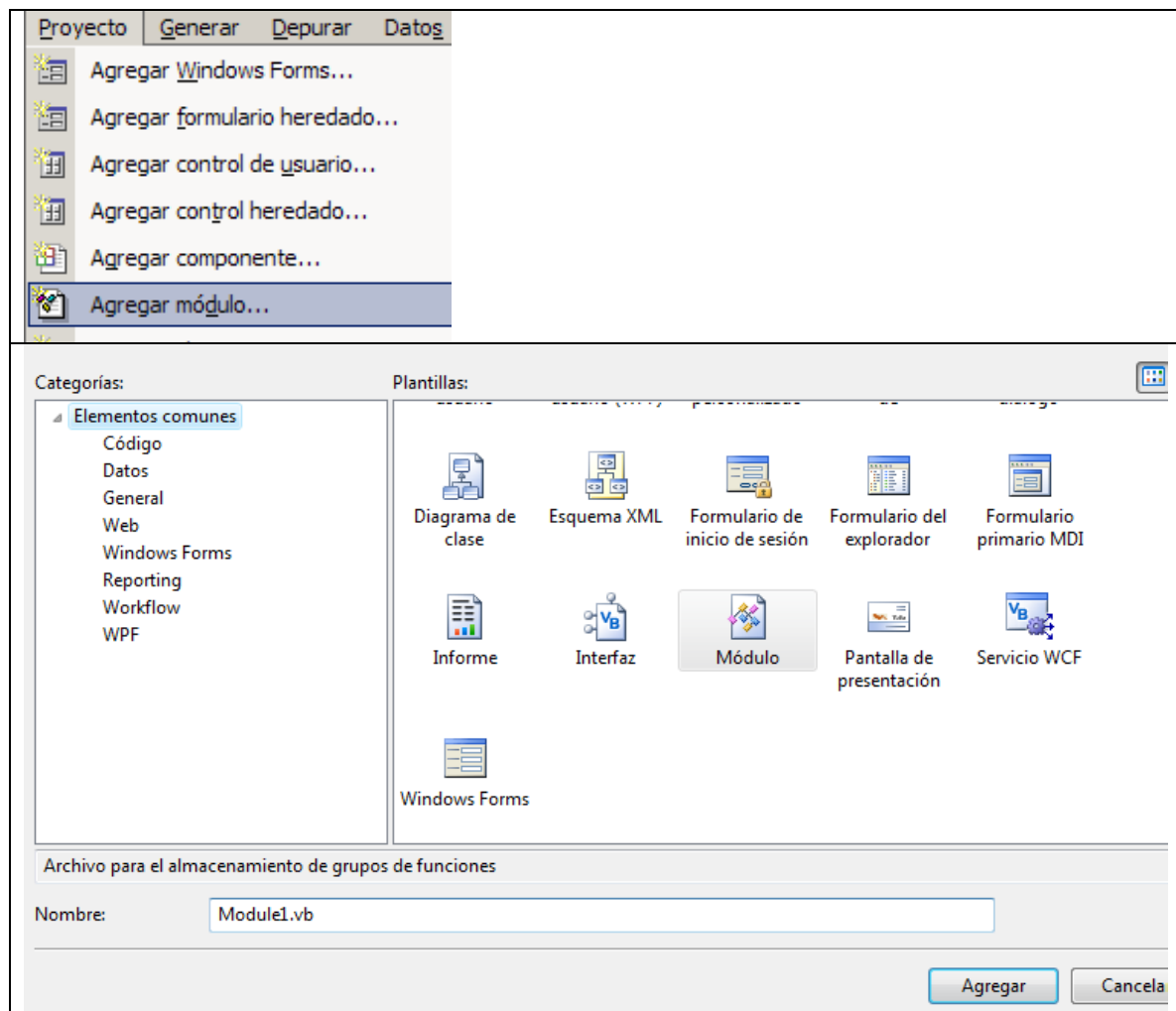
Cantidad	3
----------	---

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 Class Form1
```

```
    Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
        Dim Valor, Resp As Double  
        Valor = Val(Me.Tdato.Text)  
        Resp = CUBO(Valor)  
        Mensaje2.Text = Resp
```

```
    End Sub
```

```
    Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
        Dim Frase As String  
        Dim Resp As Integer  
        Frase = Me.Tdato.Text  
        Resp = VOCAL(Frase, "A")  
        Mensaje2.Text = Resp
```

```
    End Sub
```

```
    Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click
```

```
        Me.Close()
```

```
    End Sub
```

```
End Class
```

CÓDIGO DEL MÓDULO

```
Module Modulo
```

```
Public Function CUBO(ByVal Num As Double) As Double
```

```
    Dim R As Double  
    R = Math.Pow(Num, 3)  
    Return R
```

```
End Function
```

```
Public Function VOCAL(ByVal Texto As String, ByVal  
Letra As String) As Integer
```

```
    Dim Posi, I As Integer
```

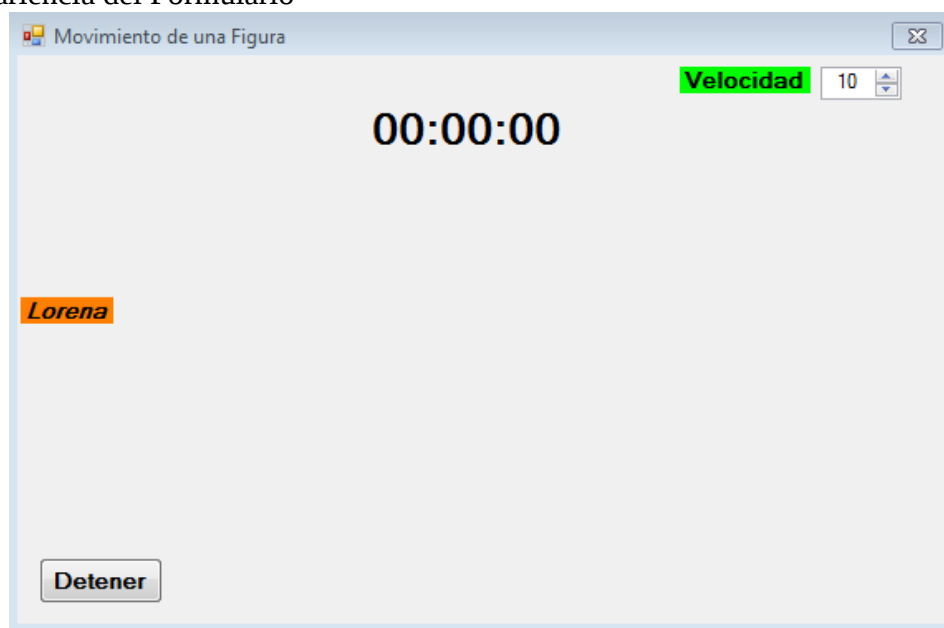
```
Dim Fra As String
For I = 1 To Len(Texto)
    Fra = Mid(Texto, I, 1)
    If UCase(Fra) = UCase(Letra) Then
        Posi = I
        I = Len(Texto)
    End If
Next I
Return Posi
```

```
End Function
End Module
```

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	2
----------	---

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

Name	Mensaje2
Text	Lorena
Font	(Tamaño 10, Negrita-Cursiva)
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 Class Form1

```
Dim DX As Integer 'Variable para incremento horizontal
Dim DY As Integer 'Variable para incremento vertical
```

```
Private Sub Reloj_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Reloj.Tick
```

```
    REM CAPTURA LA HORA DEL COMPUTADOR DOS FORMAS
    Mensaje1.Text = TimeOfDay
    REM Mensaje1.Text = Date.Now
    REM MOVIMIENTO HORIZONTAL
    If Mensaje2.Left >= (Me.Width - Mensaje2.Width) Then 'Controlo limite derecho
        DX = -DX
    End If

    If Mensaje2.Left <= 0 Then 'Controlo limite izquierdo
        DX = Math.Abs(DX)
    End If

    Mensaje2.Left = Mensaje2.Left + DX

    REM MOVIMIENTO VERTICAL
    If Mensaje2.Top >= (Me.Height - Mensaje2.Height) Then 'Controlo limite inferior
        DY = -DY
    End If

    If Mensaje2.Top <= 0 Then 'Controlo limite superior
        DY = Math.Abs(DY)
    End If

    Mensaje2.Top = Mensaje2.Top + DY

End Sub
```

```

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        REM Inicialización de Variables
        DX = 3
        DY = 4
    End Sub

    Private Sub ControlNumerico_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ControlNumerico.ValueChanged
        REM Intervalo para el Reloj
        Reloj.Interval = ControlNumerico.Value

    End Sub

    Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Boton1.Click

        REM CONTROL DEL BOTON PARA DETENER O MOVER
        If Boton1.Text = "Detener" Then
            Boton1.Text = "Mover"
            Reloj.Enabled = False
        Else
            Reloj.Enabled = True
            Boton1.Text = "Detener"
        End If

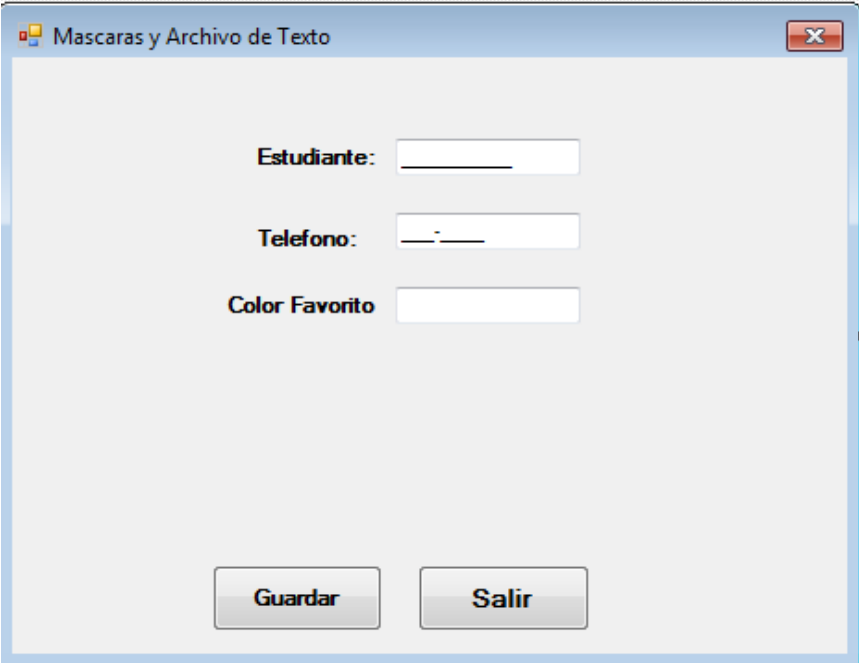
    End Sub
End Class

```

5. GRABAR Y EJECUTAR

MASCARAS Y ARCHIVOS DE TEXTO

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario



The screenshot shows a Windows application window titled "Mascaras y Archivo de Texto". Inside the window, there are three text input fields arranged vertically. The first field is labeled "Estudiante:", the second is labeled "Telefono:", and the third is labeled "Color Favorito:". Below these fields, there are two buttons: "Guardar" (Save) and "Salir" (Exit). The window has a standard Windows XP-style title bar with a close button (X) in the top right corner.

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	1
----------	---

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 MaskedTextBox1
Mask >AAAAAAAAAAA

Name MaskedTextBox2
Mask (Vacío)

BUTTON

Cantidad 2

Name Boton1
Text Guardar
BackColor (A su gusto)

Name Boton2
Text Salir
BackColor (A su gusto)

4. CODIGO

Public Class Form1

Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

REM ASIGNAMOS MASCARA Y MARCADOR

MaskedTextBox2.Mask = "(##)####-###"

MaskedTextBox2.PromptChar = "0"

End Sub

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

REM BOTON GUARDAR

REM AQUI PONEMOS LA RUTA DONDE SE CREARA EL ARCHIVO DE TEXTO

Dim VarArchivo As String = "D:\MiArchivo.txt"

Dim I As Integer

Dim MIARREGLO(3) As String

```

        MIARREGLO(0) = (MaskedTextBox1.Text & ".")
        MIARREGLO(1) = (MaskedTextBox2.Text & ".")
        MIARREGLO(2) = (TDato1.Text & ".")

Dim VarGuardar As New System.IO.StreamWriter(VarArchivo, True)

    For i = 0 To 3
        VarGuardar.WriteLine(MIARREGLO(I))
    Next

    VarGuardar.Close()
    MsgBox("Datos Almacenados con éxito")
End Sub

Private Sub Boton2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Boton2.Click
    End
End Sub
End Class

```

5. GRABAR Y EJECUTAR

CONTROL DE ERRORES Y AYUDA

1. Abrir un nuevo Proyecto
2. Apariencia del Formulario

The screenshot shows a Windows application window titled "Control de Errores y Ayuda". The window has a light blue title bar with a question mark icon and a close button. The main area has an orange background and contains four text input fields labeled "Nombre", "Apellido", "Edad", and "Dirección (F1)". Below these fields is a "Cerrar" button. At the bottom of the window, there is a status bar with two icons: a red circle with a white exclamation mark labeled "ErrorProvider1" and a blue circle with a white "F1" labeled "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)
Text Nombre

Name Mensaje2
BackColor (A su gusto)
Text Apellido
Name Mensaje3
BackColor (A su gusto)

Text	Edad
Name	Mensaje4
BackColor	(A su gusto)
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 Class Form1

```
Private Sub Tdato1_Validating(ByVal sender As  
Object, ByVal e As
```

**System.ComponentModel.CancelEventArgs) Handles
TDato1.Validating**

```
If TDato1.Text.Length = 0 Then  
    ErrorProvider1.SetError(TDato1, "Recuerde Debe  
llenar Nombre")
```

```
Else  
    ErrorProvider1.SetError(TDato1, "")  
End If  
End Sub
```

**Private Sub Tdato2_Validating(ByVal sender As
Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles
Tdato2.Validating**

```
If Tdato2.Text.Length = 0 Then  
    ErrorProvider1.SetError(Tdato2, "Recuerde Debe  
llenar Apellido")
```

```
Else  
    ErrorProvider1.SetError(Tdato2, "")  
End If  
End Sub
```

**Private Sub Tdato3_Validating(ByVal sender As
Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles
Tdato3.Validating**

```
If Not IsNumeric(Tdato3.Text) Then  
REM esto viene la frecuencia con la que parpadeará el  
ErrorProvider1
```

```
        ErrorProvider1.BlinkRate = 200  
REM para que siempre esté parpadeando mientras  
cuando no se corrija el error.
```

```
ErrorProvider1.BlinkStyle = ErrorBlinkStyle.AlwaysBlink  
ErrorProvider1.SetError(Tdato3, "No es correcto el  
dato ingresado")
```

```
REM con esto hacemos que el control TextBox no pierda el  
enfoque por el momento...
```

```

        e.Cancel = True

Else
        ErrorProvider1.SetError(Tdato3, "")
        e.Cancel = False
End If

End Sub

```

REM OTRO CASO

```

Private Sub Tdato4_HelpRequested(ByVal sender As  

Object, ByVal hlpevent As  

System.Windows.Forms.HelpEventArgs) Handles  

Tdato4.HelpRequested

```

```

    HelpProvider1.SetHelpString(Tdato4, "Dirección del  

    Domicilio")

```

```

End Sub

```

```

Private Sub Form1_Load(ByVal sender As  

System.Object, ByVal e As System.EventArgs) Handles  

MyBase.Load

```

```

        REM PARA ENLAZAR MEDIANTE TECLA F1

```

```

    HelpProvider1.HelpNamespace =  

    "http://msdn.microsoft.com/library/en-  

    us/vbcon/html/vbconbuttoncontroloverview.asp"

```

```

    REM FORMA 2, PARA QUE SE VISUALICE ICONO DE AYUDA

```

```

    HelpProvider1.SetHelpString(TDato1, "Ingresa el  

    Nombre del Cliente")

```

```

    HelpProvider1.SetHelpString(Tdato2, "Digite  

    Apellido del Cliente")

```

```

End Sub

```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
Me.Close()
```

```
End Sub
```

```
End Class
```

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)

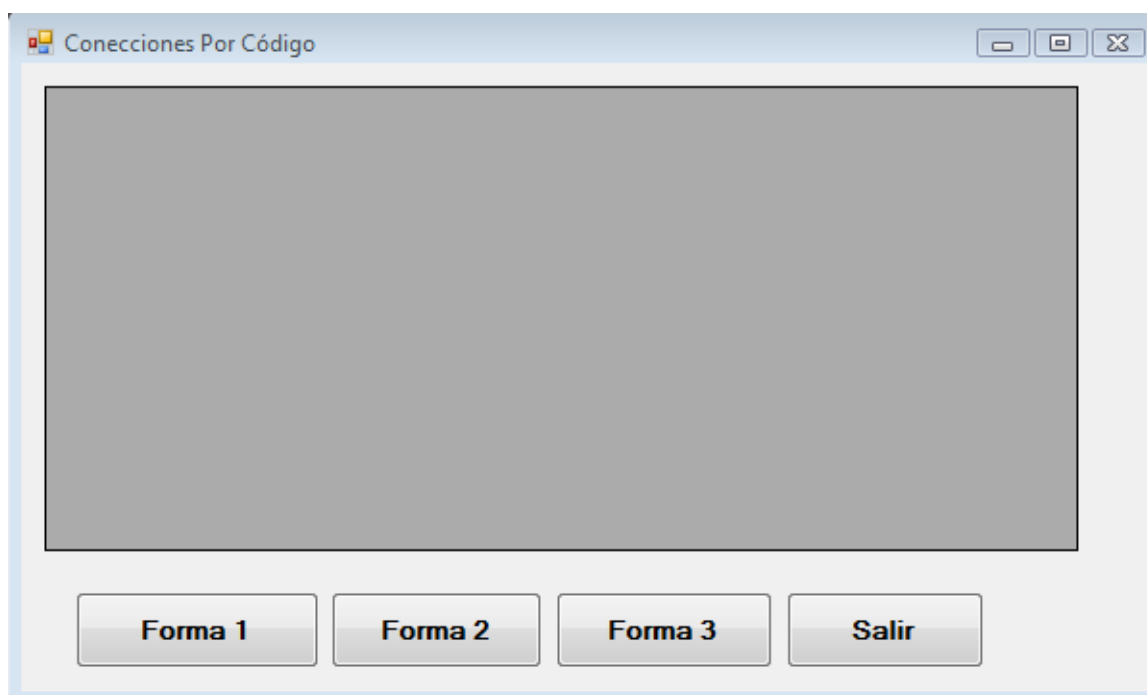
AULA.MDB VERSIÓN 2003

TBLDATOS

* Matricula	N Numérico (Entero) (Campo Llave)
Alumno	Texto (30)
Curso	N Numérico (Byte)
Sexo	Texto (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	Conecciones 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

DATAGRIDVIEW

Cantidad 1

Name DataGridView1

5. CODIGO

```
Imports System.Data.Common REM Para Forma 1
Imports System.Data.OleDb REM Para Form 2
```

Public Class Form1

```
Dim DBA As DbProviderFactory =
DbProviderFactories.GetFactory("System.Data.OleDb")
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
REM FORMA 1
```

```
Dim CN_A As DbConnection = DBA.CreateConnection()
Dim CMD_A As DbCommand = CN_A.CreateCommand
Dim DA_A As DbDataAdapter = DBA.CreateDataAdapter
Dim DT_A As New DataTable
```

```

CN_A.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2008\AULA.Mdb"

CMD_A.CommandText = "Select * From TblDatos"
    DA_A.SelectCommand = CMD_A
    DA_A.Fill(DT_A)
    DataGridView1.DataSource = DT_A
End Sub

Private Sub Boton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    REM FORMA 2
    Dim Coneccion As OleDbConnection
    Dim Adap As OleDbDataAdapter
    Dim Tabla As DataSet

    REM Enlazando coneccion a la base de datos
    Coneccion = New
    OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=c:\ProyectosNet2008\AULA.Mdb")

    REM Cargando el adapter con la instruccion sql

    Adap = New OleDbDataAdapter("Select * from
    TblDatos", Coneccion)

    REM Cargando el dataset

    Tabla = New DataSet()
    Adap.Fill(tabla, "TblDatos")

    REM Cargando el datagridview
    DataGridView1.DataSource = Tabla
    DataGridView1.DataMember = "TblDatos"
End Sub

Private Sub Boton3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton3.Click
    REM FORMA 3

```

```

Dim Coneccion As String =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2008\AULA.Mdb"

Try
    Dim CONN As New OleDb.OleDbConnection(Coneccion)
    Dim CMD As New OleDb.OleDbCommand("Select * From
TblDatos", CONN)

    Dim DA As New OleDb.OleDbDataAdapter(CMD)
        Dim DS As New DataSet
        CONN.Open()
        DA.Fill(DS)
        CONN.Close()
        DataGridView1.DataSource = DS.Tables(0)
    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub Boton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton4.Click
    REM Salir
    Me.Close()

    End Sub
End Class

```

6. GRABAR Y EJECUTAR

CONSULTAS POR CÓDIGO

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

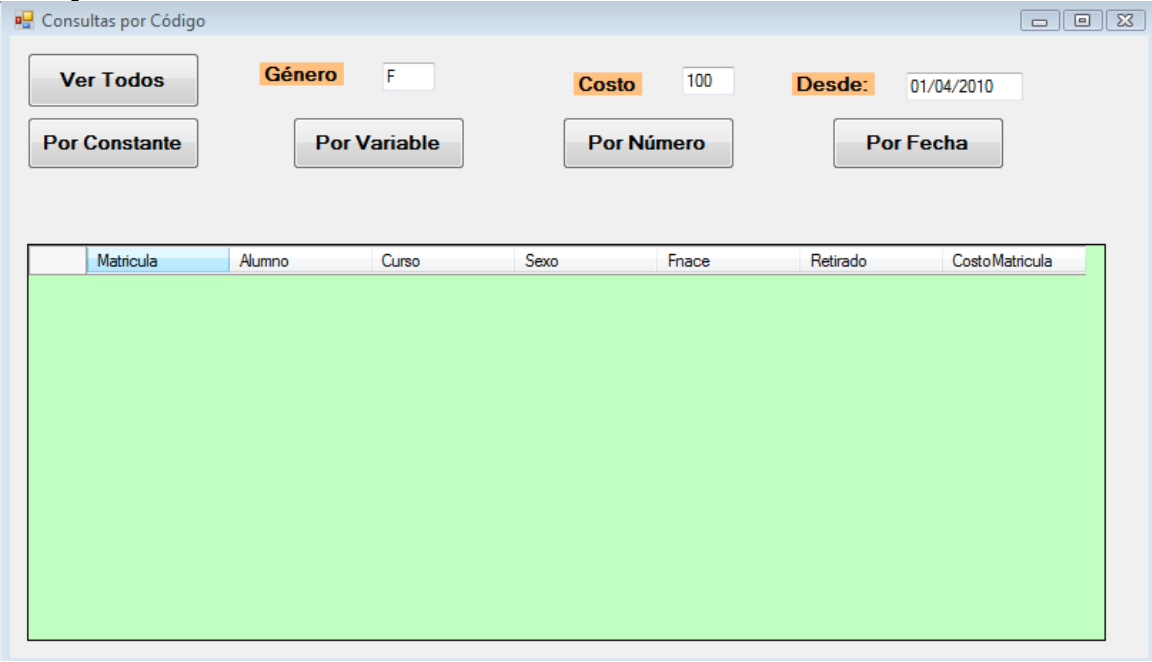
AULA.MDB

TBLDATOS

* Matricula	N Numérico (Entero) (Campo Llave)
Alumno	T Texto (30)
Curso	N Numérico (Byte)
Sexo	T Texto (1)
Fnace	F Fecha/Hora (Fecha Corta)
Retirado	S Sí/No
CostoMatricula	M 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

Imports System.Data.OleDb

Public Class Form1

Private Sub Boton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

```

        REM VER TODOS
        Dim Coneccion As String =
        "Provider=Microsoft.Jet.OLEDB.4.0; Data
        Source=c:\ProyectosNet2008\AULA.Mdb"

        Try
        Dim CONN As New OleDb.OleDbConnection(Coneccion)
        Dim CMD As New OleDb.OleDbCommand("Select * From
        TblDatos", CONN)

        Dim DA As New OleDb.OleDbDataAdapter(CMD)
            Dim DS As DataSet
            REM Cargando el Dataset

            DS = New DataSet()
            DA.Fill(DS, "TblDatos")

            REM Cargando el Datagridview
            DataGrid.DataSource = DS
            DataGrid.DataMember = "TblDatos"

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

```

```

        DataGridView.DataSource = DS
    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try
End Sub

```

Private Sub Boton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton2.Click

```

    REM CONSULTA POR VALOR CONSTANTE
    Dim Coneccion As String =
    "Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=c:\ProyectosNet2008\AULA.Mdb"

    Try
        Dim CONN As New OleDb.OleDbConnection(Coneccion)
        Dim CMD As New OleDb.OleDbCommand("Select * From
        TblDatos Where Sexo='M'", CONN)

        Dim DA As New OleDb.OleDbDataAdapter(CMD)
        Dim DS As DataSet
        REM Cargando el Dataset

        DS = New DataSet()
        DA.Fill(DS, "TblDatos")

        REM Cargando el Datagridview
        DataGridView.DataSource = DS
        DataGridView.DataMember = "TblDatos"

        CONN.Open()
        DA.Fill(DS)
        CONN.Close()
        DataGridView.DataSource = DS
    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try

End Sub

```

Private Sub Boton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton3.Click

```

        REM POR VARIABLE TIPO STRING
        Dim Mivar As String
Mivar = Trim(Tdato1.Text) REM Retiramos espacios en Blanco

Dim Coneccion As String =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2008\AULA.Mdb"

Try
Dim CONN As New OleDb.OleDbConnection(Coneccion)
Dim CMD As New OleDb.OleDbCommand("Select * From
TblDatos Where Sexo='" & Mivar & "'", CONN)

Dim DA As New OleDb.OleDbDataAdapter(CMD)
    Dim DS As New DataSet
    REM DS = New DataSet()
    DA.Fill(DS, "TblDatos")

    DataGridView.DataSource = DS
    DataGridView.DataMember = "TblDatos"

        CONN.Open()
        DA.Fill(DS)
        CONN.Close()
        DataGridView.DataSource = DS

    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try

End Sub

Private Sub Boton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton4.Click
        REM POR VARIABLE TIPO NUMÉRICO
        Dim Mivar As Integer
Mivar = Tdato2.Text REM Tomamos el dato ingresado
Dim Coneccion As String =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2008\AULA.Mdb"

Try

```

```

        Dim CONN As New
OleDb.OleDbConnection(Coneccion)
        Dim CMD As New
OleDb.OleDbCommand("Select * From TblDatos Where
CostoMatricula >=" & Mivar, CONN)

Dim DA As New OleDb.OleDbDataAdapter(CMD)
        Dim DS As New DataSet
        REM DS = New DataSet()
        DA.Fill(DS, "TblDatos")

DataGridView.DataSource = DS
DataGridView.DataMember = "TblDatos"

        CONN.Open()
        DA.Fill(DS)
        CONN.Close()
        DataGridView.DataSource = DS

Catch ex As OleDbException
        MsgBox(ex.Message)
End Try
End Sub

Private Sub Boton5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton5.Click
        REM POR VARIABLE TIPO FECHA
        Dim Mivar As Date
Mivar = Tdato3.Text REM Tomamos el dato ingresado

Dim Coneccion As String =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2008\AULA.Mdb"

Try
        Dim CONN As New OleDb.OleDbConnection(Coneccion)
        Dim CMD As New OleDb.OleDbCommand("Select * From
TblDatos Where Fnace >=#" & Mivar & "#", CONN)
Dim DA As New OleDb.OleDbDataAdapter(CMD)
        Dim DS As New DataSet
        REM DS = New DataSet()
        DA.Fill(DS, "TblDatos")

```

```
DataGrid.DataSource = DS
DataGrid.DataMember = "TblDatos"

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

    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try

End Sub
End Class
```

6. GRABAR Y EJECUTAR

CONSULTAS POR COMPONENTES

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

AULA.MDB

TBLDATOS

* Matricula	N Numérico (Entero) (Campo Llave)
Alumno	T Texto (30)
Curso	N Numérico (Byte)
Sexo	T Texto (1)
Fnace	F Fecha/Hora (Fecha Corta)
Retirado	S Sí/No
CostoMatricula	M Moneda

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

3. Apariencia del Formulario

Consultas con Componentes

Ver Todos Genero F Costo 120 Desde: 01/04/2010

Por Constante Por Texto Por Número Por Fecha

Matricula	Alumno	Curso	Sexo	Fnace	Retirado	CostoMatricula
-----------	--------	-------	------	-------	----------	----------------

MiDataSet1 MiBindingSource TAdatos

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
----------	---

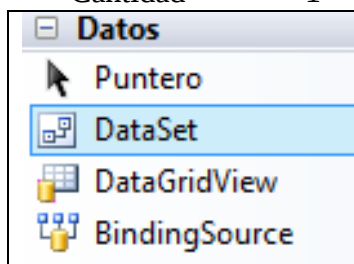
Name Tdato1
 BackColor (A su gusto)
 Text F

Name Tdato2
 BackColor (A su gusto)
 Text 120

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

DATASET

Cantidad 1



Agregar conjunto de datos

Elija un conjunto de datos con tipo o sin tipo para agregar al diseño.

☒ Conjunto de datos con tipo

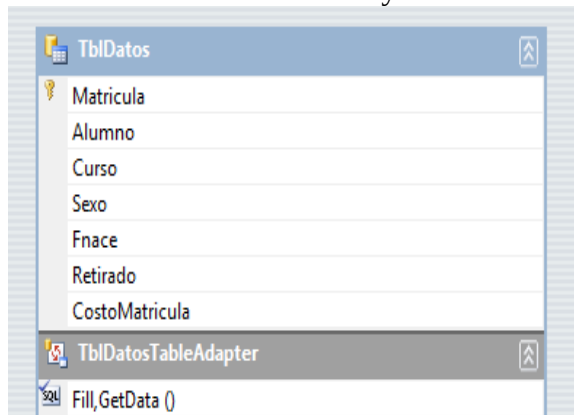
Nombre: BaseDatos.DataSet1

Crea una instancia de una clase de conjunto de datos con tipo que y esta opción para trabajar con un conjunto de datos que tiene un esquema. Ayuda para obtener detalles acerca de cómo generar conjuntos de datos.

☐ Conjunto de datos sin tipo

Crea una instancia de una clase de conjunto de datos sin tipo del tipo esta opción cuando desee un conjunto de datos sin ningún esquema.

Clic derecho sobre DataSet1 y Editor



Clic derecho sobre TblDatos 1 y escoger Agregar Query.

Asistente para la configuración de consultas de TableAdapter

Elija un tipo de comando

La consulta de TableAdapter utiliza instrucciones SQL o un procedimiento almacenado.

Elija el modo de acceso de la consulta de TableAdapter a la base de datos

☒ Usar instrucciones SQL

Especifique una instrucción SELECT para cargar datos.

☐ Crear un nuevo procedimiento almacenado

Especifique una instrucción SELECT y el asistente generará un nuevo registro.

Pulse Siguiente

Asistente para la configuración de consultas de TableAdapter

Elija un tipo de consulta

Elija el tipo de consulta que se va a generar

¿Qué tipo de consulta SQL desearía usar?

☒ SELECT que devuelve filas

Devuelve una o varias filas o columnas.

Pulse Siguiente

Escriba su instrucción SQL o utilice el Generador de consultas para construirlo.

¿Qué datos debe cargar la tabla?

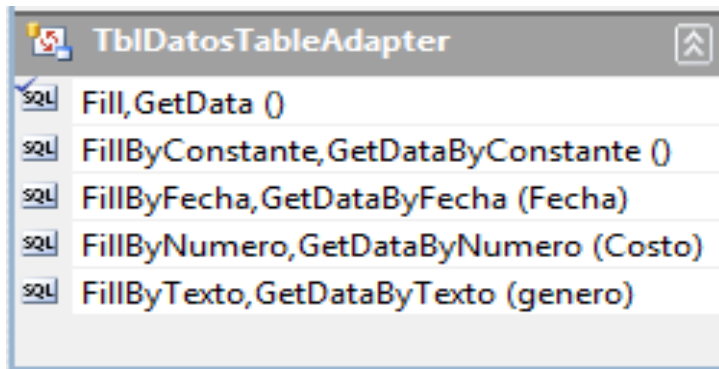
```
SELECT Matricula, Alumno, Curso, Sexo, Fnace, Retirado, CostoMatricula
FROM TblDatos
WHERE (Sexo = ?)
```

Digite Sentencia del Select

Nota: El signo ? es por el parámetro a enviar

¿Qué métodos desea agregar a TableAdapter? ☒ Rellenar un DataTable Crea un método que toma un DataTable o DataSet como parámetro almacenado SELECT introducido en la página anterior. Nombre de método: FillByTexto ☒ Devolver un DataTable Crea un método que devuelve una nueva DataTable con los datos almacenados SELECT introducido en la página anterior. Nombre de método: GetDataByTexto Ponemos Nombre al Método y Siguiente	
---	--

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

```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
Private Sub FormConsultas_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
'TODO: esta línea de código carga datos en la tabla  
'MiDataSet1.TblDatos' Puede moverla o quitarla  
según sea necesario.
```

```
Me.TAdatos.Fill(Me.MiDataSet1.TblDatos)  
End Sub
```

```
Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
TAdatos.Fill(Me.MiDataSet1.TblDatos)  
End Sub
```

```
Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
REM CONSULTA POR VALOR CONSTANTE  
  
TAdatos.FillByConstante(Me.MiDataSet1.TblDatos)  
End Sub
```

```
Private Sub Boton3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton3.Click
```

```
REM VARIABLE TIPO STRING  
Dim MiVar As String  
MiVar = Trim(Tdato1.Text)  
TAdatos.FillByTexto(Me.MiDataSet1.TblDatos, MiVar)  
End Sub
```

```
Private Sub Boton4_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton4.Click
```

```
REM VARIABLE TIPO NUMÉRICO  
Dim MiVar As Integer  
MiVar = Tdato2.Text  
  
TAdatos.FillByNumero(Me.MiDataSet1.TblDatos, MiVar)  
End Sub
```

```
Private Sub Boton5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton5.Click
    REM VARIABLE TIPO FECHA
    Dim MiVar As Date
    MiVar = Tdato3.Text
    TAdatos.FillByFecha(Me.MiDataSet1.TblDatos, MiVar)
End Sub
End Class
```

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	N Numérico (Entero) (Campo Llave)
Alumno	T Texto (30)
Curso	N Numerico (Byte)
Sexo	T Texto (1)
Fnace	F Fecha/Hora (Fecha Corta)
Retirado	S Sí/No
CostoMatricula	M 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

Matricula		Nacimiento	29/05/2001
Estudiante		Retirado	☐
Curso	1	Costo	169
Genero	M		

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	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

Imports System.Data.OleDb

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Call CargarDatos()

End Sub

Private Sub Boton1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Boton1.Click

```

        REM CONSULTAR
        Call CARGARDATOS()
    End Sub

Private Sub Boton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton2.Click
        REM INSERTAR
        Dim Coneccion As String =
        "Provider=Microsoft.Jet.OLEDB.4.0; Data
        Source=c:\ProyectosNet2010\AULA.Mdb"
        Try

            Dim CONN As New OleDb.OleDbConnection(Coneccion)
            Dim CMD As New OleDb.OleDbCommand("Insert Into
            TblDatos(Matricula,Alumno,Curso,Sexo,Fnace,Retirado
            ,CostoMatricula) Values
            (Matricula,Alumno,Curso,Sexo,Fnace,Retirado,CostoMa
            trricula)", CONN)

            REM 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()
            REM DESDE AQUI RELLENA DATOS EN EL DATAGRID

            Dim CMD2 As New OleDb.OleDbCommand("Select * From
            TblDatos order by Alumno ", CONN)

            Dim DA As New OleDb.OleDbDataAdapter(CMD2)
                Dim DS As New DataSet
                DA.Fill(DS, "TblDatos")
                DataGridView1.DataSource = DS
                CONN.Close()

            Catch ex As OleDbException
                MsgBox(ex.Message)
            End Try

```

End Sub

Sub CARGARDATOS ()

REM VER TODOS

Dim Coneccion As String =

"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2010\AULA.Mdb"

Try

Dim CONN As New OleDb.OleDbConnection(Coneccion)

Dim CMD As New OleDb.OleDbCommand("Select * From
TblDatos", CONN)

Dim DA As New OleDb.OleDbDataAdapter(CMD)

Dim DS As DataSet

REM Cargando el Dataset

DS = New DataSet()

DA.Fill(DS, "TblDatos")

REM Cargando el Datagridview

DataGrid.DataSource = DS

DataGrid.DataMember = "TblDatos"

CONN.Open()

DA.Fill(DS)

CONN.Close()

DataGrid.DataSource = DS

Catch ex As OleDbException

MsgBox(ex.Message)

End Try

End Sub

End Class

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	N Numérico (Entero) (Campo Llave)
Alumno	T Texto (30)
Curso	N Numérico (Byte)
Sexo	T Texto (1)
Fnace	D Fecha/Hora (Fecha Corta)
Retirado	S Sí/No
CostoMatricula	M Moneda

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

3. Apariencia del Formulario

The screenshot shows a Windows application window titled "Actualización por Código". The window has a light blue background. On the left side, there are four labels: "Matricula", "Alumno", "Curso", and "Genero", each followed by a text input box. Below these input boxes are three buttons: "Siguiete", "Anterior", and "Actualizar". At the bottom of the window is a large, empty gray rectangular area, likely a placeholder for a list or data grid.

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	DataGridView
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

```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
    Dim Mitabla As DataTable
```

```
    Dim FI As Integer
```

```
    Private Sub FormActualiza_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
        Call CARGARDATOS()
```

```
        Call MOSTRARDATOS(0)
```

```
End Sub
```

```
    Private Sub Boton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton1.Click
```

```
        REM SIGUIENTE
```

```
        REM Posicionarse en la fila siguiente
```

```
        Dim UFI As Integer = Mitabla.Rows.Count - 1
```

```
        FI = FI + 1
```

```
        If FI > UFI Then FI = UFI
```

```
        MOSTRARDATOS(FI)
```

```
End Sub
```

```
    Private Sub Boton2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Boton2.Click
```

```
        REM SIGUIENTE
```

```

        REM Posicionarse en la fila anterior
        Dim UFI As Integer = Mitabla.Rows.Count - 1
        FI = FI - 1
        If FI > UFI Then FI = 1
        MOSTRARDATOS(FI)
End Sub

Private Sub Boton3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Boton3.Click
        REM ACTUALIZAR
        Dim Alu, Cur As String
        Dim Sex As Char

Dim Coneccion As String =
"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\ProyectosNet2010\AULA.Mdb"

        Try

                REM ASIGNAMOS DATOS
                Alu = Tdato2.Text
                Cur = Tdato3.Text
                Sex = Tdato4.Text

Dim CONN As New OleDb.OleDbConnection(Coneccion)
Dim CMD As New OleDb.OleDbCommand("Update TblDatos
Set Alumno= Alu, Curso=Cur, Sexo=Sex Where
Matricula=" & Tdato1.Text, CONN)

REM EN ESTE EJERCICIO SOLO SE COLOCO TRES CAMPOS
REM 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()
        REM DESDE AQUI RELLENA DATOS EN EL DATAGRID

Dim CMD2 As New OleDb.OleDbCommand("Select * From
TblDatos order by Matricula", CONN)

Dim DA As New OleDb.OleDbDataAdapter(CMD2)

```

```

        Dim DS As New DataSet
        DA.Fill(DS, "TblDatos")
        DataGrid.DataSource = DS
        CONN.Close()

    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try
End Sub

```

Sub CARGARDATOS ()

```

    REM VER TODOS
    Dim Coneccion As String =
    "Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=c:\ProyectosNet2010\AULA.Mdb"

    Try
        Dim CONN As New OleDb.OleDbConnection(Coneccion)
        Dim CMD As New OleDb.OleDbCommand("Select * From
        TblDatos Order By Matricula", CONN)

        Dim DA As New OleDb.OleDbDataAdapter(CMD)
        Dim DS As DataSet

        REM Cargando el Dataset

        DS = New DataSet()
        DA.Fill(DS, "TblDatos")

        REM Cargando el Datagridview
        DataGrid.DataSource = DS
        DataGrid.DataMember = "TblDatos"

        Mitabla = New DataTable
        REM Llenar la tabla con los datos indicados
        DA.Fill(Mitabla)

        CONN.Open()
        REM DA.Fill(DS)
        CONN.Close()
        DataGrid.DataSource = DS
    Catch ex As OleDbException
        MsgBox(ex.Message)
    End Try

```

End Sub

Private Sub MOSTRARDATOS (ByVal Fila As Integer)

REM ASIGNACIÓN DE CAMPOS HACIA LOS TEXTBOX

Dim Ufila As Integer = Mitabla.Rows.Count - 1

If Fila < 0 OrElse Ufila < 0 Then Exit Sub

Dim MiCampo As DataRow = Mitabla.Rows(Fila)

Tdato1.Text = MiCampo("Matricula").ToString

Tdato2.Text = MiCampo("Alumno").ToString

Tdato3.Text = MiCampo("Curso").ToString

Tdato4.Text = MiCampo("Sexo").ToString

End Sub

End Class

6. GRABAR Y EJECUTAR

HA TENER PRESENTE. FORMATOS

¿QUÉ ES VISUAL BASIC .NET

Del portal (Canal Visual Basic, 2010), es un lenguaje orientado a objetos y eventos que soporta encapsulación, herencia y polimorfismo. Es una mejora a Visual Basic formando parte de Visual Studio y compartiendo el entorno de desarrollo con Microsoft Visual C++ .NET, Microsoft Visual C# .NET, etc.

DECLARACIÓN DE VARIABLES

Existen diversas formas para declarar una variable o inicializar la misma con un valor constante pero antes conoceremos los significados de estas dos palabras:

Dim es la abreviatura de **Dimensión**, que sirve para reservar espacio para la variable. La sentencia **Dim** especifica el nombre de la variable y su tipo de dato asociado.

As de la instrucción de declaración permite definir el tipo de datos o de objetos de la variable que se está declarando.

- **Estructura:**
 - **Dim** “nombre de variable” **As** “tipo de dato”
- **Ejemplos:**
 - **Dim** num **As** Byte
 - **Dim** x **As** Integer
 - **Dim** a,b **As** String

INICIALIZAR VARIABLES

- Para inicializar se realiza de la siguiente manera:
 - **Dim** x **As** Integer
x=0
 - **Dim** x **As** Integer=0
 - **Dim** x **As** Integer : x=0
- Por lo general este tipo de declaraciones se utiliza en la estructura de ciclos manuales, ya que ahí se declara y se inicializa la variable para luego utilizarlo en el ciclo.

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 PI **As** Double = 3.14159265

TIPOS DE DATOS O VARIABLES

Tipo de Dato	Escritura en Visual Net	Asignación de almacenamiento nominal	Intervalo de valores
Boolean	Boolean	En función de la plataforma de implementación	True o False
Byte	Byte	1 byte	0 a 255 (sin signo)
Char (carácter individual)	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	16 bytes	0 a +/- 79.228.162.514.264.337.593.543.950.335
Double (punto flotante de precisión doble)	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	Int32	4 bytes	-2.147.483.648 a 2.147.483.647 (con signo)
Long (entero largo)	Int64	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	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

flotante de precisión simple)			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

Visual Basic .NET dispone de los siguientes operadores matemáticos:

Operador	Descripción
+	Suma
-	Resta
*	Multipliación
/	División
	División entera (parte entera de la división)
Mod	Residuo (resto de la división entera)
^	Exponenciación (elevar a una potencia)
&	Concatenación de Cadenas

OPERADORES ARITMÉTICOS

Los operadores aritméticos requieren operadores numéricos (^,*,/,Mod,+,-) y producen resultados numéricos.

Operadores String

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

Operadores Comparativos

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

Operadores Lógicos

Los operadores Lógicos requieren operadores Lógicos (NOT,AND,OR,XOR) y producen un resultado lógico.

Operadores Abreviados

Visual Basic .NET incorpora nuevos operadores abreviados que facilitan la escritura de código, por ejemplo:

`x = x + 1` 'Antes escribía

`x += 1` 'Ahora puede escribir

Operadores Abreviados
+=
-=
*=
/=
=
^
&=

Precedencia de Operadores

A continuación se muestra el orden de precedencia/prioridad/jerarquía de los operadores en Visual Basic .NET

Precedencia de Operadores
()
^
-
* /
Mod
+ -

Sentencias de Control

A manera de resumen recordaremos que For se utiliza cuando se conoce de antemano el número de iteraciones a ejecutarse. Do se utiliza cuando no se conoce de antemano el número de iteraciones a ejecutarse. Si al menos debe ejecutarse 1 iteración es necesario colocar la expresión condicional al final del ciclo. Evite ciclos infinitos, para ello asegúrese de que estos ciclos tengan una expresión condicional de salida. Utilice Until a diferencia de While cuando requiera que la expresión condicional sea la contraria, por ejemplo con While la expresión condicional podría ser $A \neq B$ y con Until podría ser $A = B$.

La única sentencia de control que tiene un cambio relevante sintacticamente es While ya que antes se escribía While ... Wend ahora es necesario escribir While ... End While.

Visual Basic .NET incorpora 2 nuevos operadores lógicos para utilizarlos en las sentencias de control, a continuación se mostrará la tabla de operadores lógicos anteriores y nuevos:

Operadores Lógicos	Descripción
And	Las 2 expresiones deben ser verdaderas
Or	Alguna de las 2 expresiones es verdadera
Not	Negación del resultado de la expresión
Xor	Si 1 y sólo 1 de las expresiones es verdadera
AndAlso	Si la primer y segunda condición son verdaderas
OrElse	Si la primer o segunda condición es verdadera

Los nuevos operadores lógicos AndAlso y OrElse reciben el nombre de sistema de cortocircuito.

```
Dim nCalificacion As Integer = 0
If nCalificacion <= 0 AndAlso nCalificacion / 0 Then
    MsgBox("Error de lógica")
Else
    MsgBox("Continuar cálculo")
End If
```

Parece que el operador lógico AndAlso como OrElse nos ahorran escribir un If dentro de otro If/else y quizá evitar un error en tiempo de ejecución.

```
Dim nCalificacion As Integer = 0
If nCalificacion < 0 OrElse nCalificacion / 0 Then
    MsgBox("Error de lógica")
Else
    MsgBox("Continuar cálculo")
End If
```

Ciclo 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) Then (sentencia1) Else (sentencia2)

La segunda es más general y se muestra a continuación:

If (condición) Then

 sentencia(s)

Else

 sentencia(s)

End If

Además existe el formato en una sola instrucción directa

IIF (condición, Sentencia Verdadero, Sentencia Falso)

Do ... Loop

La sentencia de control Do ... Loop es la sentencia general iterativa, la cual permite ejecutar repetitivamente un grupo de sentencias hasta que una condición sea cumpla.

Existen cuatro versiones de la sentencia Do ... Loop :

1. Do *While* condición
2. sentencias
3. Loop
- 4.
5. Do *Until* condición
6. sentencias
7. Loop
- 8.
9. Do
10. sentencias
11. Loop *While* condición
- 12.
- 13.
14. Do
15. sentencias
16. Loop *Until* condición
- 17.

Exit

La sentencia Exit permite salir inmediatamente de una decisión, ciclo o procedimiento.

ARREGLOS

Concepto: Los vectores 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, la única diferencia es que debemos indicar la cantidad de elementos que contendrá el arreglo, colocando el número de índice entre paréntesis. Observe los ejemplos:

Acceder a los datos de un Vector

Ejemplo 1: La instrucción siguiente crea un arreglo de 16 elementos o datos de tipo integer:	<code>Dim miArreglo(0 To 15) As Integer</code>
Ejemplo 2: La sentencia a continuación crea un arreglo de 11 elementos o datos de tipo String:	<code>Dim empleados (10) as String</code>

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

MATRICES

Concepto: Una matriz es un conjunto de elementos contiguos, todos del mismo tipo, que comparten un nombre común, a los que se puede acceder por la posición (índice) que ocupa cada uno de ellos dentro de la matriz.. 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 y pueden tener una o varias dimensiones. A las matrices de una dimensión se les llama también listas y a las de dos dimensiones tablas. En general, se utiliza la denominación de matriz (array en inglés).

Declaración de una Matriz

La declaración de una matriz especifica el nombre de la matriz, el número de elementos de la lista y el tipo de éstos. En función del número de dimensiones, distinguimos entre matrices unidimensionales y multidimensionales.

Sentencia Dim

La sentencia Dim, además de para declarar variables, se puede utilizar también para declarar matrices. Su sintaxis es:

`Dim variable [(dims)] [As tipo] [variable [(dims)] [as tipo]]...`

Variable es un identificador que nombra la matriz.

Dim es una de expresiones numéricas, separadas por comas y que definen las dimensiones de una matriz. Esta lista puede ser de la forma siguiente:

`[Inferior to] superior [, [inferior to] superior]...`

As tipo define el tipo de variable. Puede ser un tipo simple (Integer, log, single, double, string, etc.) o un tipo definido por el usuario.

Por ejemplo, la siguiente línea de código define una matriz unidimensional llamada temp con subíndices definidos en el rango de 0 a 59. Esto es, los elementos de la matriz son: temp (0), temp (1),..., temp (59). El límite inferior predeterminado es 0.

`Dim temp (1 to 60) as string * 40`

La sentencia siguiente declara una matriz bidimensional de 10 por 10 elementos de tipo Double:

`Dim ma (9,9) as double`

El espacio de almacenamiento total que necesita una matriz depende de su número de elementos y del tipo de cada elemento. Por lo tanto, hay que utilizar las matrices multidimensionales con sumo cuidado. También hay que tener especial cuidado en las matrices de tipo Variant, ya que este tipo de datos requiere más espacio de almacenamiento que los otros tipos de datos.

REFERENCIA

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

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.

Evangelos Petroutsos. (1999). Visual Basic 6. España: Anaya Multimedia. Recuperado a partir de <http://dialnet.unirioja.es/servlet/libro?codigo=319544>

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/>

Microsoft. (11 de Marzo de 2009). *Msdn.Microsoft*. Obtenido de Libreria:

<http://msdn.microsoft.com/es-es/library/2x7h1hfk%28v=vs.80%29.aspx>

La importancia de la programación visual 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 la Visual Basic NET.

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- 978-9942-21-145-3

