



Microsoft .NET

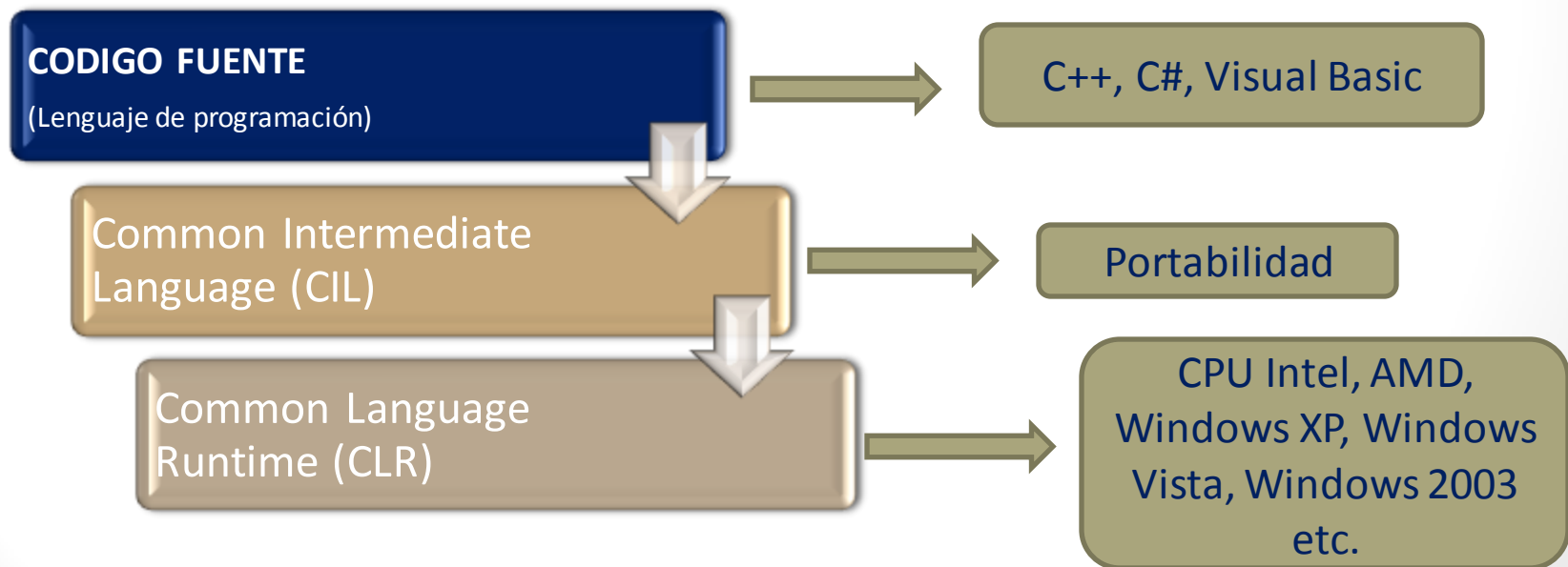
Visual Basic .NET

MICROSOFT.NET

- Microsoft. Net es una nueva tecnología que proporciona todas las herramientas para construir Aplicaciones Autónomas y Aplicaciones Web Distribuidas.
- Las aplicaciones creadas en la plataforma .NET pueden ser usadas por un gran número de clientes, tales como PC, PC de bolsillo (Palm PC), celulares, PC de juegos (PC Games), PC de auto (Auto PC), PC Televisor (Ultimate TV), etc.

MICROSOFT .NET FRAMEWORK

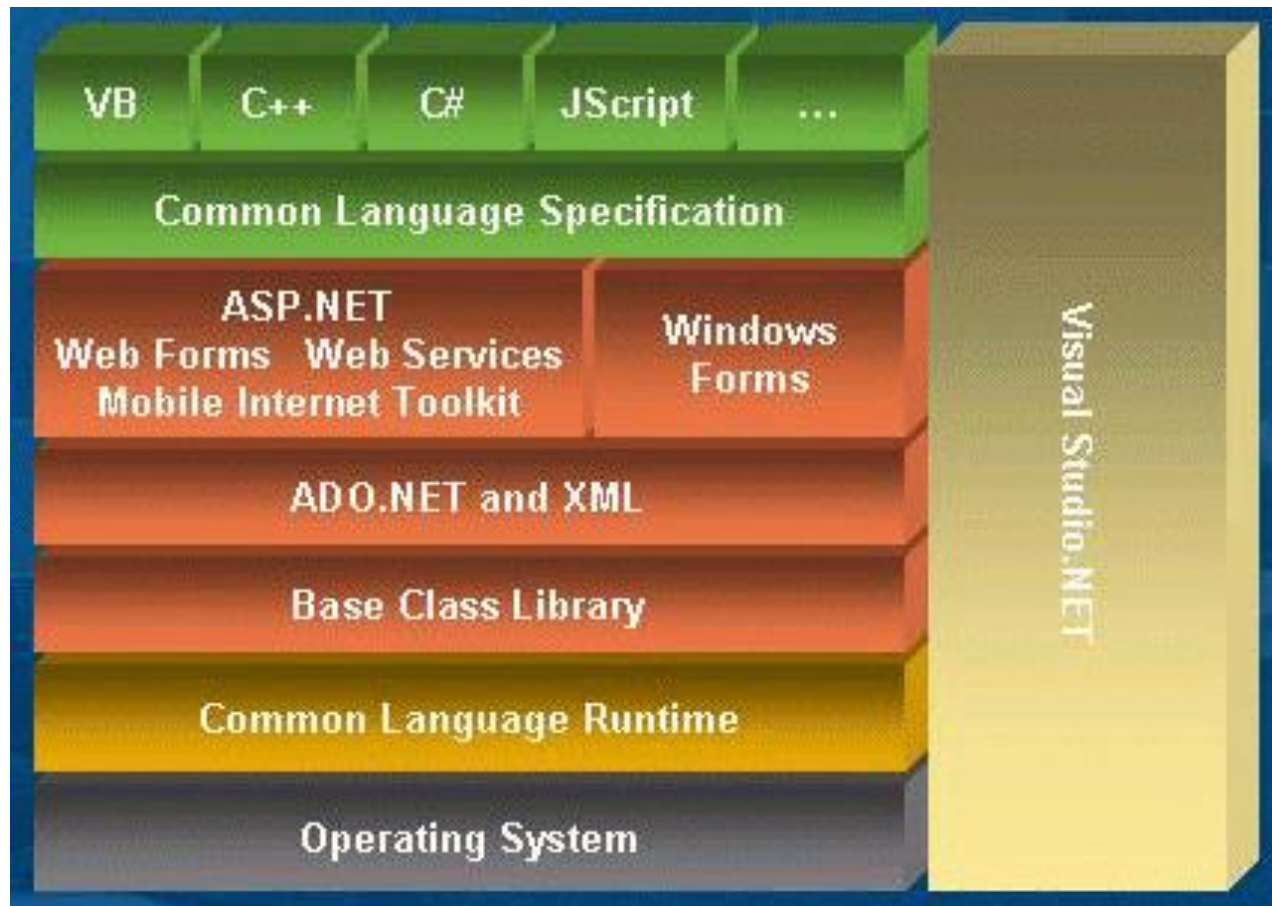
- .NET Framework es un entorno para construir, instalar y ejecutar servicios Web y otras aplicaciones. Estas son implementadas mediante **Visual Studio .NET**.
- .NET Framework consta de dos componentes principales: el Common Language Runtime (CLR) y la librería de clases .NET Framework.
- El .NET Framework es el corazón de .NET,





Microsoft®

Visual Studio®



OPERADORES MATEMATICOS Y LOGICOS

OPERADORES MATEMATICOS	OPERADORES COMPARACION	OPERADORES LOGICOS “ Los mas usados”
/ División * Multiplicación + Suma - Resta mod Residuo División	= igual < Menor que > Mayor que <= menor o igual >= mayor o igual <> Distinto	And → y Or → o Not → Negación

Visual Basic .NET

TIPOS DE DATOS

TIPO DATO	MEMORIA	DESCRIPCION
Boolean	2 Bytes	Valores: True o False . Dim b As Boolean = True
Byte	1 Byte	Valores: de 0 a 255 Dim b As Byte = 129
Char	2 Bytes	Valores: de 0 a 65535 Dim c As Char = 'N'
Date	3 Bytes	Valores: desde las 0:00:00 del 1 de Enero del 0001 hasta las 23:59:59 del 31 de Diciembre del 9999. Las fechas deben representarse entre almohadillas # y por lo habitual usando el formato norteamericano: # m-d-yyyy # Dim fecha As Date = #10-27-2001#
Decimal	16 Bytes	Valores: de 0 a +/- 7.9228162514264337593543950335 con 28 lugares a la derecha del decimal Dim unDecimal As Decimal = 987654321.125
Double	8 Bytes	Valores: de -1.79769313486231570E+308 a -4.94065645841246544E-324 para valores negativos; de 4.94065645841246544E-324 a 1.79769313486231570E+308 para valores positivos. Dim unDoble As Double = 987456.0125

Visual Basic .NET

TIPOS DE DATOS

TIPO DATO	MEMORIA	DESCRIPCION
Integer	4 Bytes	Valores: de -2,147,483,648 a 2,147,483,647. Dim unEntero As Integer =250009
Long	8 Byte	Un entero largo (o grande) Valores: de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807.
Object	4 Bytes	Cualquier tipo se puede almacenar en una variable de tipo Object . Todos los datos que se manejan en .NET están basados en el tipo Object.
Short	2 Bytes	Un entero corto (sin decimales) Valores: de -32,768 a 32,767. Dim unShort As Short = 32000
String	Depende de la plataforma.	Una cadena de caracteres. Valores: de 0 a aproximadamente 2 billones (2^{31}) de caracteres. Dim miCadena As Short = " Cesar David Fernandez G "

Visual Basic.NET

AMBITO DE LAS VARIABLES

- **Dim:** Se puede utilizar a nivel de procedimiento y a nivel del formulario. Si se declara una variable a nivel de procedimiento con la palabra Dim, Visual Basic reinicializa (borra su contenido) la variable cada vez que ejecuta el procedimiento.

Ejemplo:

Dim Edad As Integer

- **Static:** Se utiliza solo a nivel del procedimiento. Cuando una variable se declara con la palabra Static, Visual Basic no reinicializa la variable cada vez que se ejecuta el procedimiento, solo lo hace la primera vez. Esto quiere decir que la variable conserva su valor entre una llamada al procedimiento y otra.

Ejemplo:

Static b As Double

- **Private:** Se utiliza solo a nivel del modulo e indica que la variable es privada; es decir, solo va a ser reconocida dentro del módulo donde ha sido declarada.

Ejemplo:

Private c As String

- **Public:** Se utiliza solo a nivel de módulo e indica que la variable es pública es decir, va a ser reconocida en toda la aplicación. Ej.:

Public d As Date



Visual Basic .NET

DATOS DEFINIDOS POR EL USUARIO

- REGISTROS:

Structure Persona

Dim Nombre As String

Dim Edad As Integer

Dim FechaNac As Date

End Structure

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

Dim Empleado As Persona

Empleado.Nombre = "Cesar David Fernandez G"

Empleado.Edad = 32

Empleado.FechaNac = #9 / 21 / 1977#

End Sub



Visual Basic .NET

FUNCIONES MATEMATICAS

Pow(b,e)	Calcula la potencia. b = BASE y e = EXPONENTE.
Abs(n)	Calcula el valor absoluto de n
Atan(n)	Calcula el arco tangente de n en radianes
Cos(n)	Calcula el coseno del ángulo n expresado en radianes
Exp(n)	Calcula el constante de e elevada a n
Sign(n)	Regresa -1 si n es menor que cero, 0 si n es cero y +1 si n es mayor a cero
Sin(n)	Calcula el seno del ángulo n expresado en radianes
Sqr(n)	Calcula la raíz cuadrada de n.
Tan(n)	Calcula la tangente del ángulo n expresado en radianes

Visual Basic .NET

FUNCIONES PARA MANIPULACION DE FECHAS Y TIEMPO

- **TimeString**: regresa la hora actual del sistema.
- **DateString**: regresa la fecha actual del sistema.
- **Now**: regresa un valor codificado que representa la hora y fecha actual del sistema.
- **Hour(hora)**: regresa el número de hora actual del sistema.
- **Minute(hora)**: regresa el número de minuto actual del sistema.
- **Second(hora)**: regresa el número del segundo actual del sistema.
- **Day(fecha)**: regresa el número del día actual del sistema.
- **Month(fecha)**: regresa el número de mes actual del sistema.
- **Year(fecha)**: regresa el año Actual del sistema.
- **Weekday(fecha)**: regresa el número que representa el día de la semana.
(1 = Domingo, 2 = Lunes, ...).

Visual Basic .NET

ESTRUCTURA CONDICIONAL SIMPLE

Sintaxis condicional simple:

```
If <expresión a evaluar> Then  
    <Instrucción (si la expresión evaluada devuelve Verdadero)>  
End If
```

Ejemplo:

```
If b <> 0 Then  
    Res = a / b  
    MessageBox.Show("Resultado es: " & Res)  
End If
```

Visual Basic .NET

ESTRUCTURA CONDICIONAL DOBLE

Sintaxis condicional doble:

```
If <expresión a evaluar> Then
    < Instrucción (si la expresión evaluada devuelve Verdadero)>
Else
    < Instrucción (si la expresión evaluada devuelve Falso)>
End If
```

Ejemplo:

```
If b <> 0 Then
    Res = a / b
    MessageBox.Show("Resultado es: " & Res)
Else
    MessageBox.Show("ERROR División por 0", "cuidado")
End If
```

Visual Basic .NET

ESTRUCTURA CONDICIONAL MULTIPLE

Sintaxis condicional multiple:

Sintaxis:

```
Select Case <expresión a evaluar>  
    Case <lista de expresiones>  
        Instrucciones  
    Case <otra lista de expresiones>  
        Instrucciones  
    Case Else  
        'si no se cumple ninguna de  
        las listas de expresiones  
End Select
```

Ejemplo:

```
Select Case Edad  
    Case 3 : 'Instrucción a realizar  
    Case 6 To 11  
        'Instrucción a realizar  
    Case 14, 17  
        'Instrucción a realizar  
    Case Is > 25  
        ' Instrucción a realizar  
    Case Else  
        ' Instrucción a realizar  
End Select
```

Visual Basic .NET

ESTRUCTURA REPETITIVA FOR

Sintaxis:

For <variable numérica> = <valor inicial> **To** <valor final> [**Step** <incremento>]

'contenido del bucle, lo que se va a repetir

Next

Ejemplos:

For i = 1 **To** 100 **Step** 2

'contará desde 1 hasta 100 de 2 en 2

'la variable i tomará los valores 1, 3, 5, etc.

Next

For i = 10 **To** 1 **Step** -1

'contará desde 10 hasta 1

'la variable i tomará los valores 10, 9, 8, etc.

Next

Visual Basic .NET

ESTRUCTURA REPETITIVA WHILE

Sintaxis:

```
While <expresión>  
    ' Lo que haya que hacer mientras se cumpla la expresión  
End While
```

Ejemplo:

```
While i < 10  
    MessageBox.Show("Numero = " & i)  
    i = i + 1  
End While
```



Visual Basic .NET

ESTRUCTURA REPETITIVA DO / LOOP

Sintaxis:

Do

Instrucción

Loop While <expresión>

NOTA: en la anterior expresión el ciclo se ejecuta por lo menos una sola vez en caso en que la expresión sea falsa. Esta es la diferencia con respecto al ciclo While.

Ejemplo:

Do

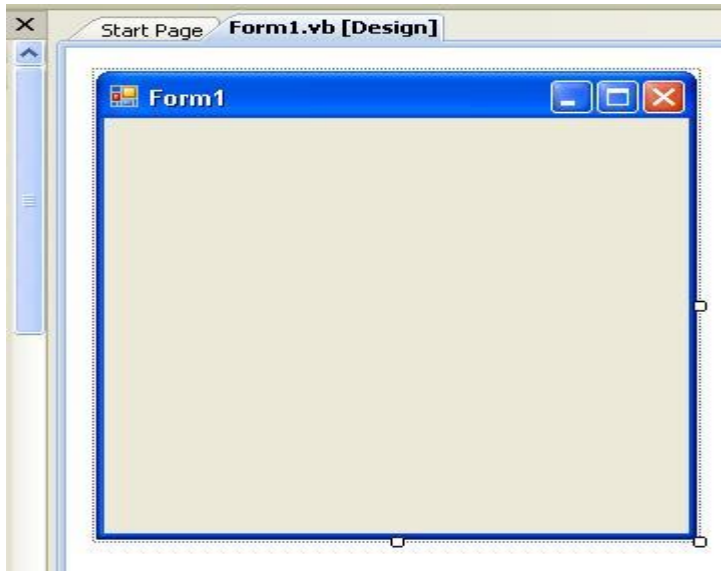
MessageBox.Show(" HOLA a TODOS", "SALUDO")

contador = contador +1

Loop While Contador < 10

Visual Basic .NET

FORMULARIOS y CONTROLES



Click en el Logo de Visual Studio
para ver controles



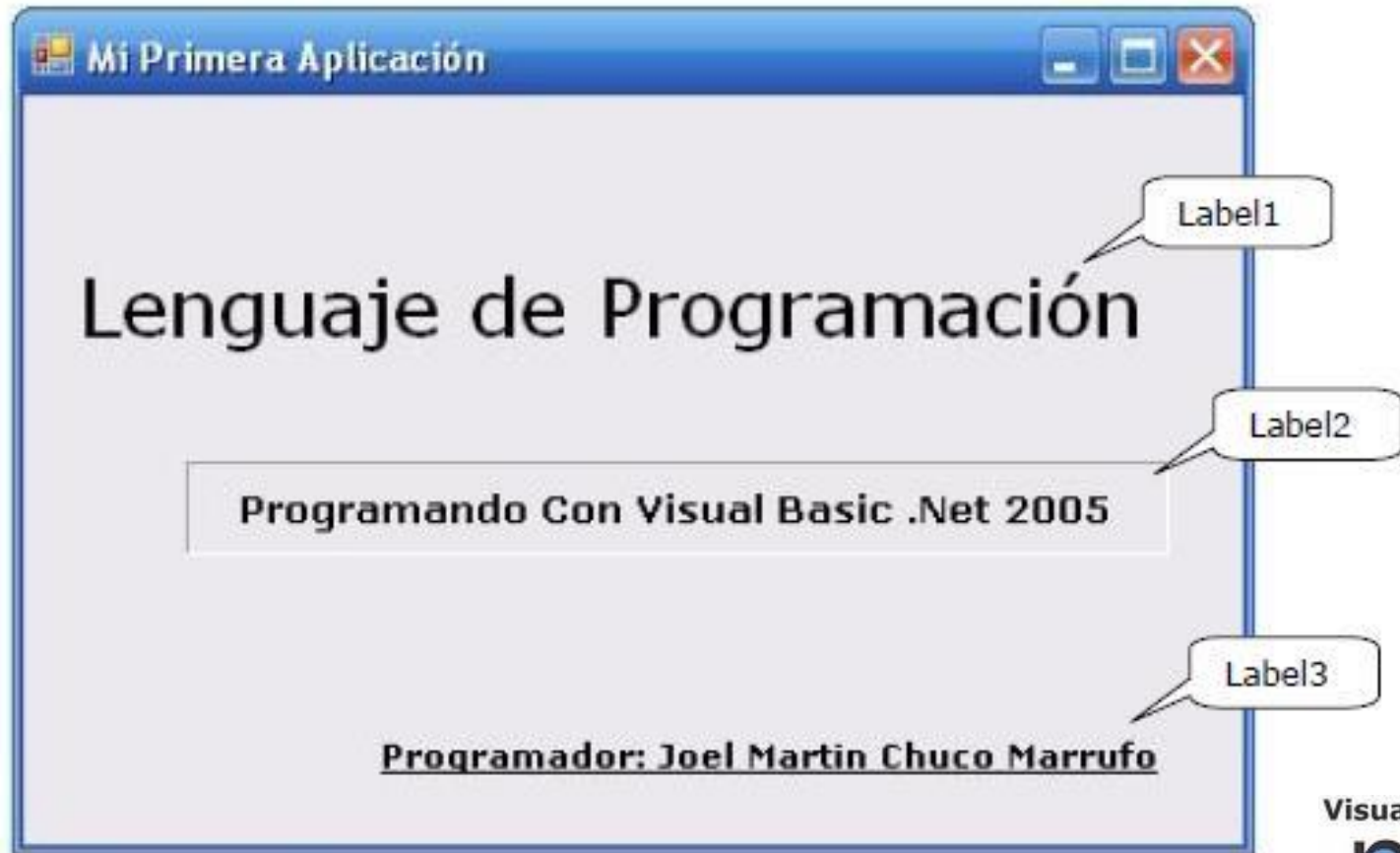
DEFINICION: Es una plantilla o ventana donde se insertan los diferentes controles de interfaz con los cuales interactúa el usuario.

TIPOS:

- **El formulario MDI:** Es un contenedor de otros formularios, por ejemplo tenemos las ventanas de Word que es un formulario MDI, ya que contiene o puede contener diversas ventanas de documentos abiertos.
- **Los formularios Child:** Son aquellos formularios hijos, en otras palabras son todos aquellos formularios que van a ser contenidos en el formulario MDI.
- **El formulario normal:** El que más vamos a usar, y el más común.
- **El formulario Parents,** que es aquel formulario que va a contener todos los controles de otro formulario,

Visual Basic .NET

EJEMPLOS DE ALGUNOS CONTROLES



Visual Basic .NET

EJEMPLOS DE ALGUNOS CONTROLES

The image shows a screenshot of a Visual Basic .NET application window titled "Ingreso de Datos". The window contains a form for entering personal data. The form has a title bar with standard Windows controls (minimize, maximize, close). The main area of the form is light gray. At the top, there is a label "Datos Personales" (Label1). Below this, there are four text boxes for entering data: "Nombre" (Label2), "Dirección" (Label3), "Teléfono" (Label4), and "Centro de Estudios" (Label5). Each text box is labeled with a corresponding label. The text boxes are labeled TextBox1, TextBox2, TextBox3, and TextBox4. Below these, there is a label "OBSERVACIONES" (Label6) and a large text box (TextBox5) for entering observations. The text boxes are labeled TextBox1, TextBox2, TextBox3, TextBox4, and TextBox5. The labels are labeled Label1, Label2, Label3, Label4, Label5, and Label6.

Visual Basic .NET

EJEMPLOS DE ALGUNOS CONTROLES

Ingreso de Datos

Datos Personales

Nombre

Dirección

Teléfono

Centro de Estudios

OBSERVACIONES

Button1

Button2

Adiciona un Nuevo Alumno

```
Private Sub Button1_Click(ByVal sender As System.Object)  
    End Sub  
End Class
```



```
TextBox1.Text = ""  
TextBox2.Text = ""  
TextBox3.Text = ""  
TextBox4.Text = ""  
TextBox5.Text = ""  
TextBox1.Focus()
```

Para cerrar aplicación:
Close()

Visual Basic .NET

EJEMPLOS DE ALGUNOS CONTROLES



Button1 (Evento Click)

Label1.Text = "Joel Martin Chuco Marrufo"

Button2 (Evento Click)

Label1.Text = "Yo Estudie en el Instituto Superior Tecnológico Argentina"

Button3 (Evento Click)

Label1.Text = "Naci el 15 de septiembre de 1984"

Button4 (Evento Click)

Close()

Visual Basic .NET

Cuadros de Dialogo MSGBOX

El MSGBOX es una ventana que nos muestra un mensaje en el centro de la pantalla. Su sintaxis es la siguiente:

MsgBox(**Mensaje**, Tipo de Mensaje, **Título**)

Por ejemplo:

```
MsgBox("¿Desea Salir de la Aplicación?", MsgBoxStyle.YesNo, "Sistema")
```

El mensaje que nos mostrara será el siguiente:



Visual Basic .NET

Cuadros de Dialogo MSGBOX

Para programar un MsgBox:

Por ejemplo:

```
If MsgBox("¿Desea Salir?", MsgBoxStyle.YesNo, "Sistema") = MsgBoxResult.Yes  
    Close()  
End If
```



Visual Basic .NET

FUNCIONES DE TIPO CADENA

- **Right:** Obtiene de una cadena, una cantidad de caracteres empezando por la derecha: Sintaxis; Microsoft.VisualBasic.Right(Cadena,N).
- **Left:** Obtiene de una cadena, una cantidad de caracteres empezando por la izquierda: Sintaxis; Microsoft.VisualBasic.Left(Cadena,N)
- **Mid:** Obtiene de una cadena, una cantidad de caracteres a partir de una posición determinada: Sintaxis; Microsoft.VisualBasic.Mid(Cadena,N).
- **Len:** Obtiene la longitud de una cadena, es decir la cantidad de caracteres que contiene incluyendo los espacios en blanco: Sintaxis; Len(Cadena).
- **StrConv:** Convierte una cadena en 3 formas diferentes, en mayúsculas = 1, minúsculas = 2, la primera letra de cada palabra en mayúsculas = 3; Sintaxis; StrConv(Cadena,Valor)
- **InStr:** Obtiene un número que indica la posición de una subcadena dentro de una cadena: Sintaxis; InStr(Cadena,SubCadena).
- **LTrim:** Quita los espacios en blanco al inicio de una cadena: Sintaxis; LTrim(Cadena)
- **UCase:** Convierte una cadena en mayúsculas: Sintaxis; UCase(Cadena)
- **LCase:** Convierte una cadena en minúsculas: Sintaxis; LCase(Cadena)



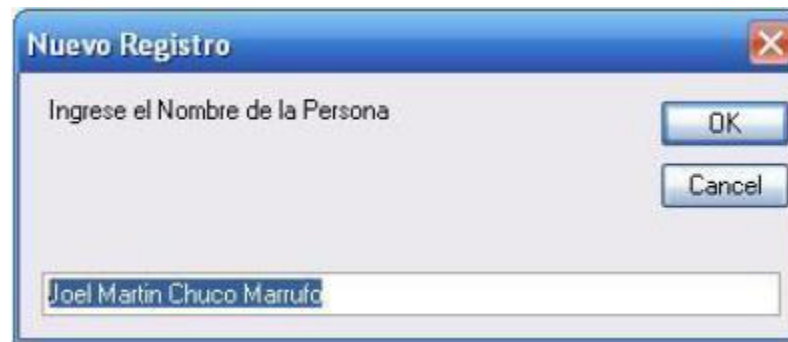
Visual Basic .NET

INPUTBOX

El INPUTBOX es una ventana que le permite al usuario el ingreso de datos. Su sintaxis es la siguiente:

InputBox(Mensaje, Titulo, Valor Pred., Columna, Fila)

- **Mensaje:** Es el mensaje que se desea mostrar cuando se pida el dato a ingresar.
- **Titulo:** Es el titulo de la ventana.
- **Valor:** Predeterminado Es el valor que se muestra en forma automática para ser ingresado.
- **Columna:** Es un valor que indica la posición horizontal del formulario, donde se desea que se muestre la ventana.
- **Fila:** Es un valor que indica la posición vertical del formulario, donde se desea que se muestre la ventana.



Visual Basic .NET

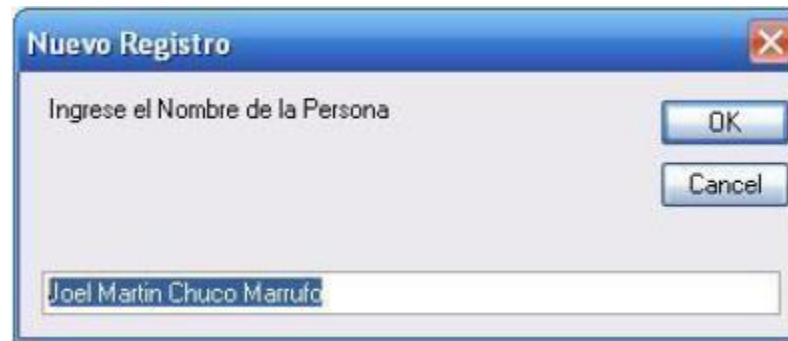
INPUTBOX

Capturando datos a través de INPUTBOX:

Dim Nombre **As** String

Nombre = InputBox("Ingrese el Nombre de la Persona", "Nuevo Registro")

En la variable **Nombre** quedara almacenado el dato digitado en el InputBox.



Visual Basic .NET

FUNCION SHELL

La función Shell se utiliza para ejecutar programas (archivos ejecutables como *.exe, *.com etc.) desde el visual Basic. La sintaxis es la siguiente:

Shell("Ruta y Nombre del programa", Estilo de Presentación)



btnCalc (Evento Click)

Shell("C:\Windows\System32\calc.exe", AppWinStyle.NormalFocus)

btnWord (Evento Click)

Shell("C:\Archivos de programa\Microsoft Office\OFFICE11\winword.exe", AppWinStyle.MaximizedFocus)

btnExcel (Evento Click)

Shell("C:\Archivos de programa\Microsoft Office\OFFICE11\excel.exe", AppWinStyle.MaximizedFocus)

btnFinalizar (Evento Click)

If MsgBox("¿Deseas salir de la aplicación?", MsgBoxStyle.YesNo, "Uso de Shell") = MsgBoxResult.Yes Then Me.Close()

Visual Basic .NET

CHECKBOX

- Este control permite activar o desactivar la casilla de verificación de manera independiente..



Visual Basic
.net

PROPIEDAD PARA SABER SI ESTA ACTIVO

```
If CheckBox1.Checked = True Then  
    MsgBox("Futbol")  
End If
```

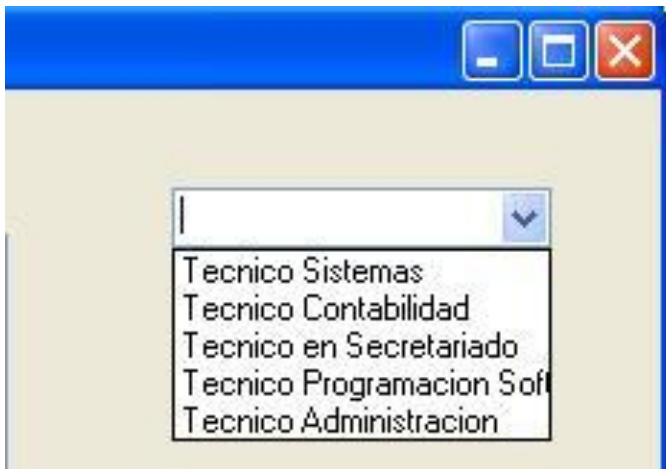
```
If CheckBox2.Checked = True Then  
    MsgBox("Natacion")  
End If
```

```
If CheckBox3.Checked = True Then  
    MsgBox("Baloncesto")  
End If
```

Visual Basic .NET

COMBOBOX

- Este control permite seleccionar una sola opción entre un conjunto de opciones. También se conoce bajo el nombre de **MENU DESPLEGABLE**.



PROPIEDAD PARA SABER SI ESTA ACTIVO

```
Dim opcion As Integer
```

```
opcion = ComboBox1.SelectedIndex()
```

```
If opcion = 0 Then
```

```
    MsgBox("Tecnico sistemas")
```

```
End If
```

```
If opcion = 1 Then
```

```
    MsgBox("Tecnico Contabilidad")
```

```
End If
```

Visual Basic .NET

RADIOBUTTON

- Este control permite crear botones de opción, cuya característica principal es que solo se puede activar un solo botón entre un conjunto de botones de opción.



PROPIEDAD PARA VERIFICAR ACTIVACION

```
If RadioButton1.Checked = True Then  
    MsgBox(" A usted le gusta la Salsa")  
End If
```

```
If RadioButton2.Checked = True Then  
    MsgBox(" A usted le gusta el Merengue")  
End If
```

Visual Basic .NET

DATETIMEPICKER

- Este control permite seleccionar una fecha de manera intuitiva mediante el despliegue de un calendario.



PROPIEDAD PARA OBTENER FECHA SELECCIONADA

`Dim F As Date`

`F = DateTimePicker1.Text`

`MsgBox(" " & F)`

`MsgBox(DateTimePicker1.Text)`

Visual Basic .NET

LINKLABEL

- Este control permite generar un hipervínculo. Para este objeto se debe crear el respectivo evento que al hacer clic sobre el hipervínculo se enlace a la respectiva dirección web.



EVENTO PARA GENERAR HIPERVINCULO

```
Private Sub Enlace_LinkClicked(ByVal sender As  
System.Object, ByVal e As  
System.Windows.Forms.LinkLabelLinkClickedEventArgs  
) Handles LinkLabel1.LinkClicked  
  
Process.Start("http://senaticscesar.blogspot.com")  
  
End Sub
```

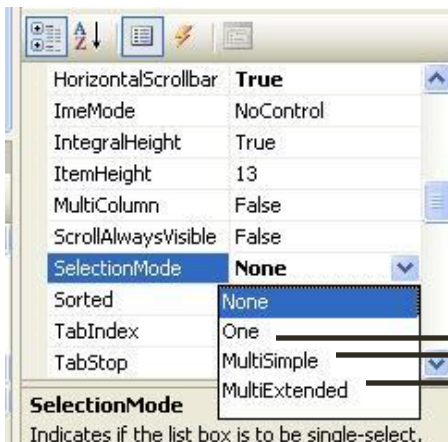
Visual Basic .NET

LISTBOX

- Este control permite seleccionar una o varias opciones de una lista de opciones.



PROPIEDADES DE SELECCION



EVENTO PARA OBTENER VALORES

```
For i = 0 To ListBox1.Items.Count - 1
    If (ListBox1.GetSelected(i)) Then

        Select Case i

            Case 0 : MsgBox("Abogado")
            Case 1 : MsgBox("Ingeniero")
            Case 2 : MsgBox("Odontóloga")
            Case 3 : MsgBox("Sicologa")
        End Select
    End If
Next
```

Una sola selección

Múltiple selección

Múltiple selección
Arrastrando mouse

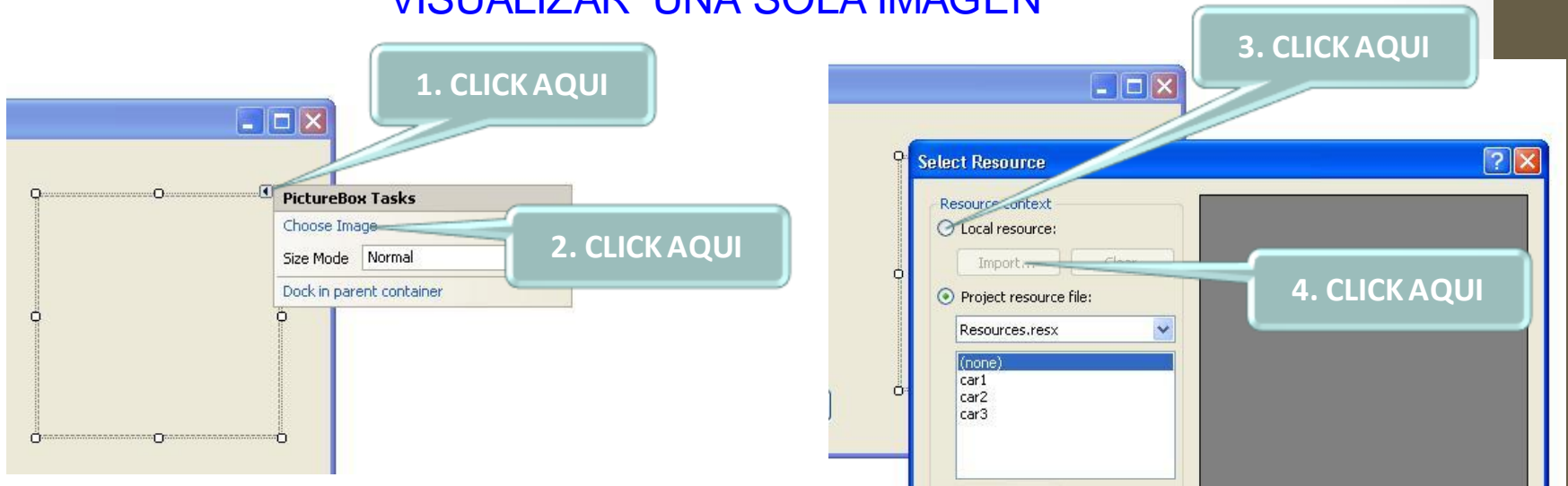


Visual Basic .NET

PICTUREBOX

- Este control permite visualizar una imagen o conjuntos de imágenes.

VISUALIZAR UNA SOLA IMAGEN



PROPIEDAD DE VISIBILIDAD

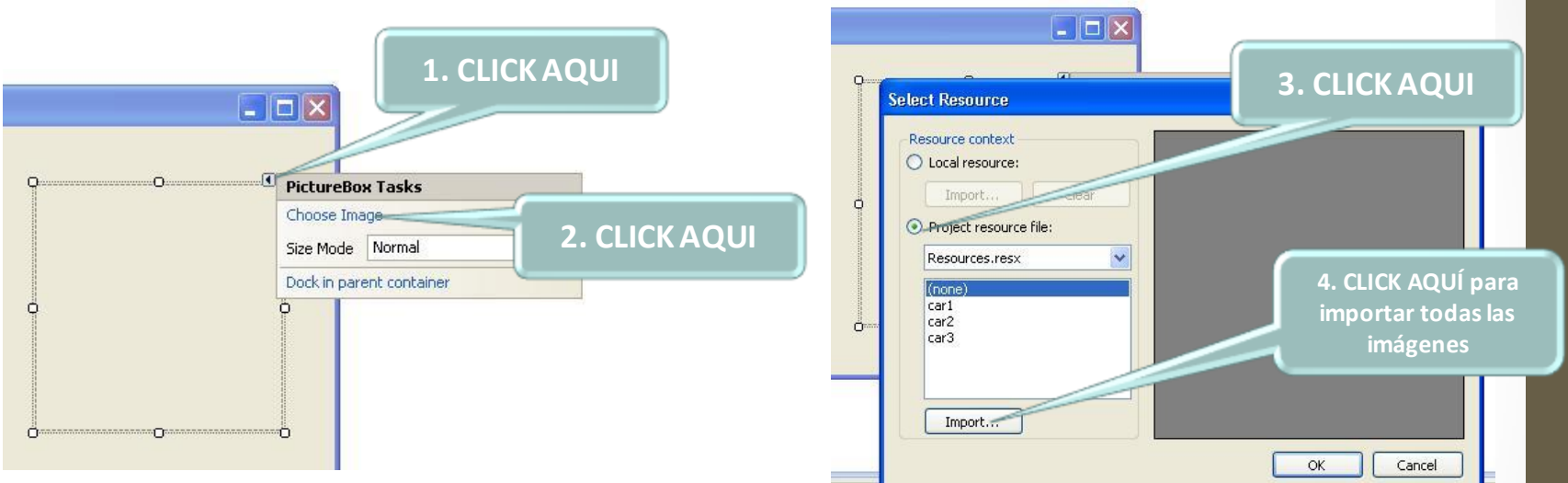
Carro.Visible = False

Carro.Visible = True

Visual Basic .NET

PICTUREBOX

VISUALIZAR VARIAS IMÁGENES: “Ejemplo con un **CheckedListBox**”




1. CLICK AQUI

2. CLICK AQUI

3. CLICK AQUI

4. CLICK AQUÍ para importar todas las imágenes



Form1

- ☒ CARRO 1
- ☐ CARRO 2
- ☐ CARRO 3

Seleccione Carro

Image of a car

```
For i = 0 To lista.Items.Count - 1
    If (lista.GetItemChecked(i)) Then

        Select Case i
            Case 0 : Carro.Image = My.Resources.car1
            Case 1 : Carro.Image = My.Resources.car2
            Case 2 : Carro.Image = My.Resources.car3
        End Select
    End If
Next
```

Visual Basic .NET

PROGRESSBAR

- Este control permite establecer una barra de progreso.

Form1

☐ CARRO 1
☐ CARRO 2
☐ CARRO 3

Seleccione Carro

NIVEL DE FOTOS CARGADAS

[Empty progress bar]

Form1

☒ CARRO 1
☐ CARRO 2
☐ CARRO 3

Seleccione Carro

NIVEL DE FOTOS CARGADAS

[Progress bar with 25 segments]

Form1

☒ CARRO 1
☒ CARRO 2
☐ CARRO 3

Seleccione Carro

NIVEL DE FOTOS CARGADAS

[Progress bar with 50 segments]

Form1

☒ CARRO 1
☒ CARRO 2
☒ CARRO 3

Seleccione Carro

NIVEL DE FOTOS CARGADAS

[Progress bar with 75 segments]

```

For i = 0 To lista.Items.Count - 1
    If (lista.GetItemChecked(i)) Then

        Select Case lista.GetItemText(i)

            Case 0 : CarrolImage.Image = My.Resources.car1
                    ProgressBar1.Value = 25
            Case 1 : CarrolImage.Image = My.Resources.car2
                    ProgressBar1.Value = 50
            Case 2 : CarrolImage.Image = My.Resources.car3
                    ProgressBar1.Value = 75
        End Select
    End If
Next
    
```

Valor máximo

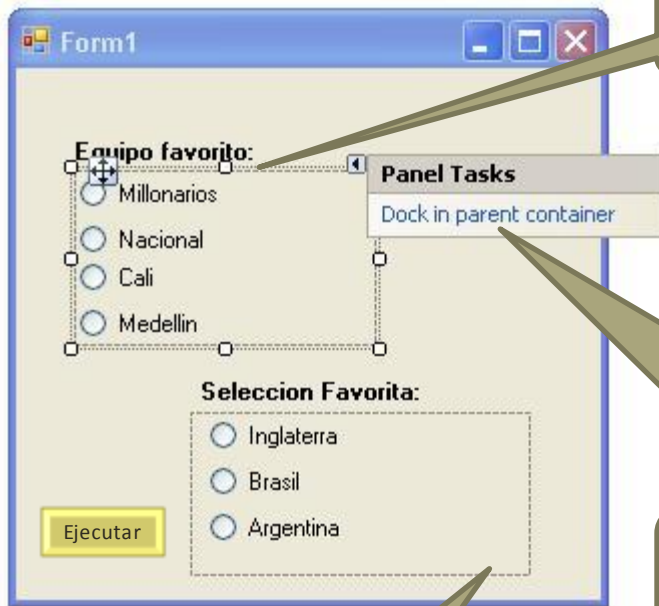
Valor mínimo

Behavior	
ContextMenuS	(none)
Enabled	True
MarqueeAnim	100
Maximum	75
Minimum	0
Step	10
Style	Blocks
TabIndex	3
Value	0

Visual Basic .NET

PANEL

- Este control permite agrupar objetos o controles permitiendo un manejo independiente entre el grupo de controles que posee cada panel del Form.



Activar y desactivar objetos Panel:

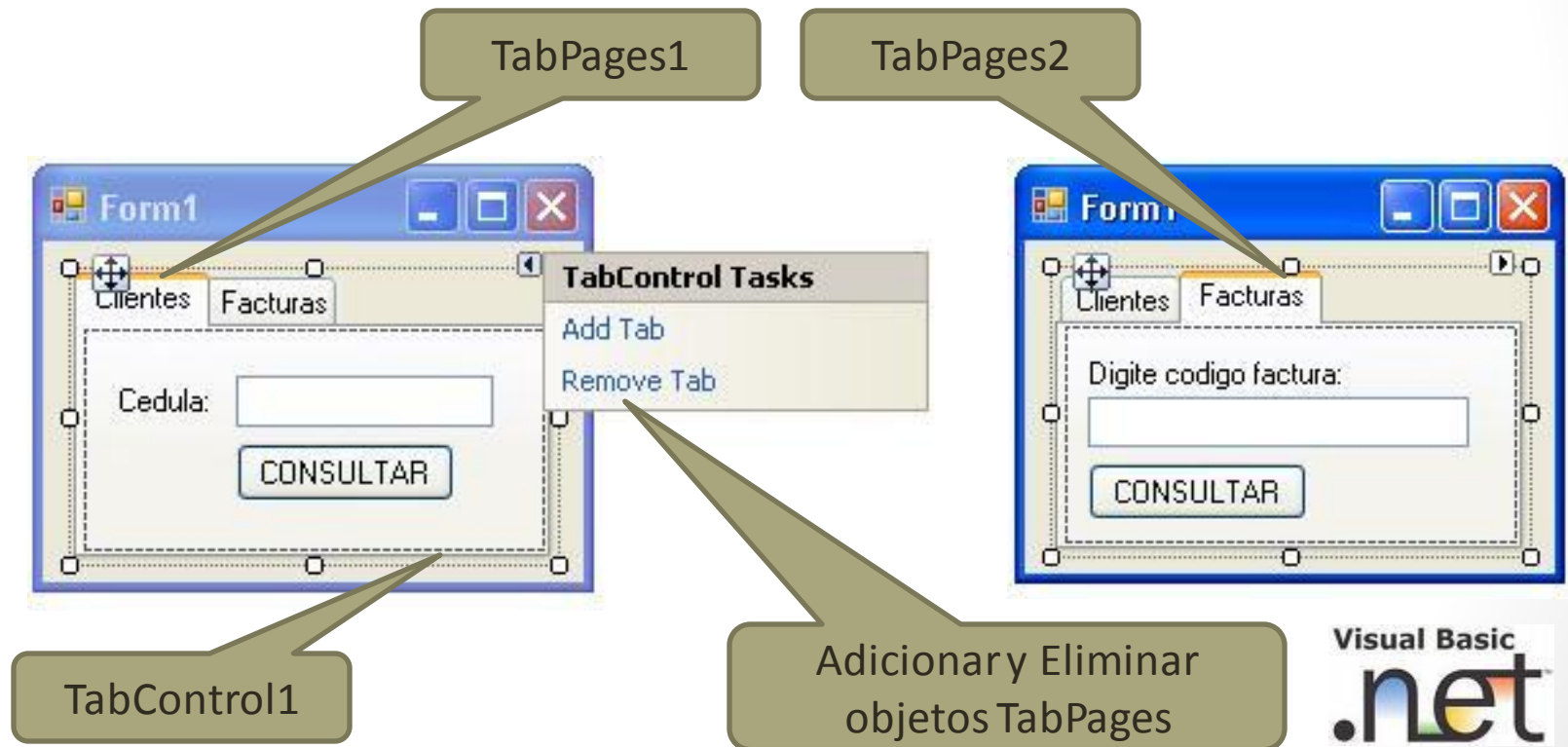
```
Private sub Button1_click()  
    If millonarios.Checked then  
        panel2.visible=false 'Desactivado  
    end if  
    If nacional.checked then  
        panel2.visible=true 'Activado  
    end if  
end sub
```

Permite maximizar el panel cubriendo todo el formulario

Visual Basic .NET

TABCONTROL

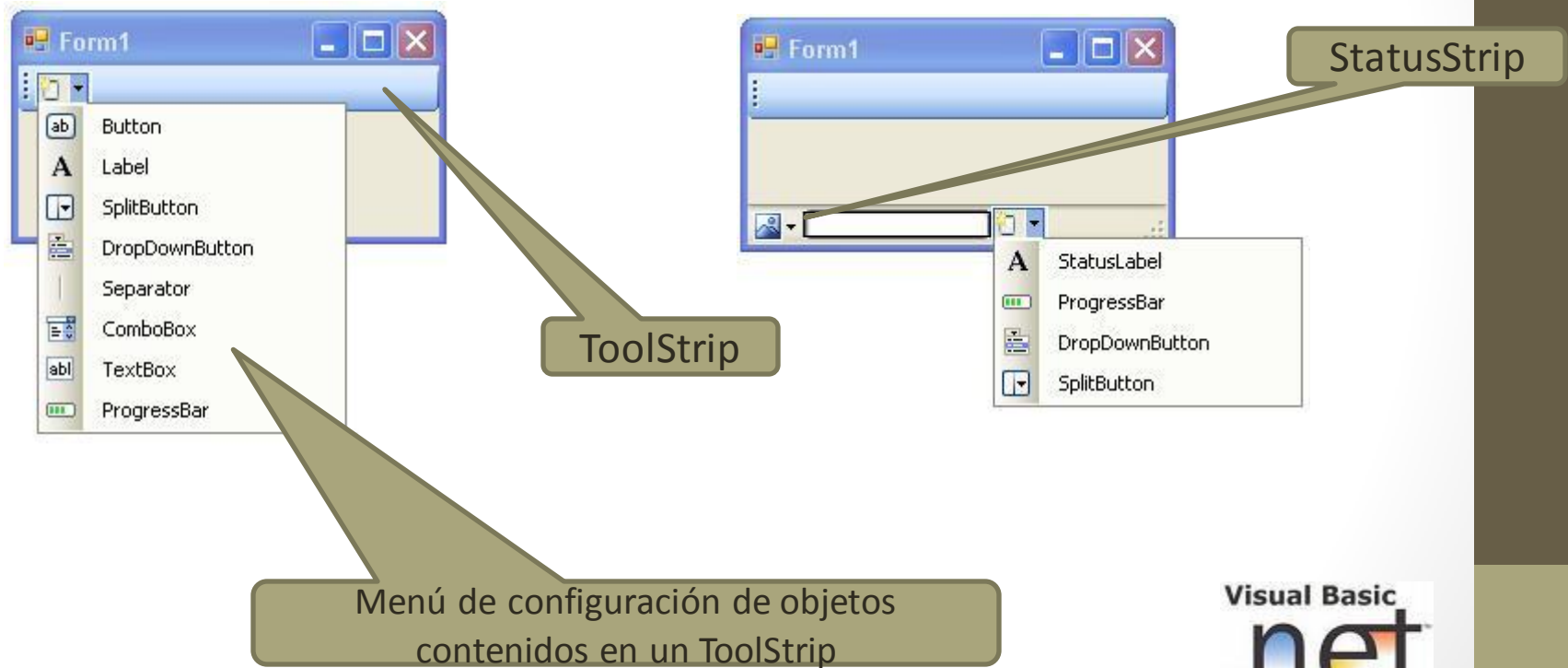
- Este control puede ser utilizado cuando se tienen demasiados datos en un solo formulario y es posible separarlos por medio de fichas u objetos llamados **TabPages**.



Visual Basic .NET

ToolStrip y StatusStrip

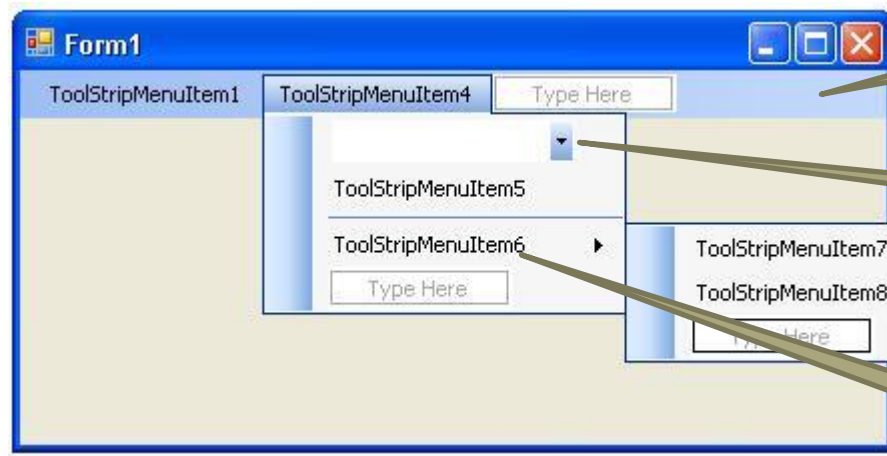
- Estos objetos permiten crear una barra de herramientas y una barra de estado. Cada control de estos puede contener objetos especificados en los menús de configuración.



Visual Basic .NET

MenuStrip

- Este objeto permite crear barra de menús.



MenuStrip

ToolStripComboBox

Submenús
ToolStripMenuItem

Visual Basic .NET

TIMER

- Este objeto se visualiza en tiempo de diseño mas no en tiempo de ejecución. Permite configurar tiempos de procesamiento de instrucciones.



```
Public Class Form1
    Dim incremento As Integer = 0
    PrivateSub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        incremento = incremento + 10
        ProgressBar1.Value = incremento
        If incremento = 100 Then
            Timer1.Enabled = False
            incremento = 0 'listo para iniciar de nuevo
        End If
    End Sub

    PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Timer1.Enabled = True
    End Sub

    PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        ProgressBar1.Value = 0
    End Sub
End Class
```

INQUIETUDES O DUDAS ?
MUCHAS GRACIAS

PROXIMAMENTE PROGRAMACION ORIENTADA A OBJETOS

Ing. Cesar David Fernández Grueso.
CENTRO DE TELEINFORMATICA Y PRODUCCION INDUSTRIAL
SENA REGIONAL CAUCA