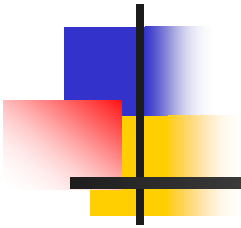




Interfaces

Curso 2009

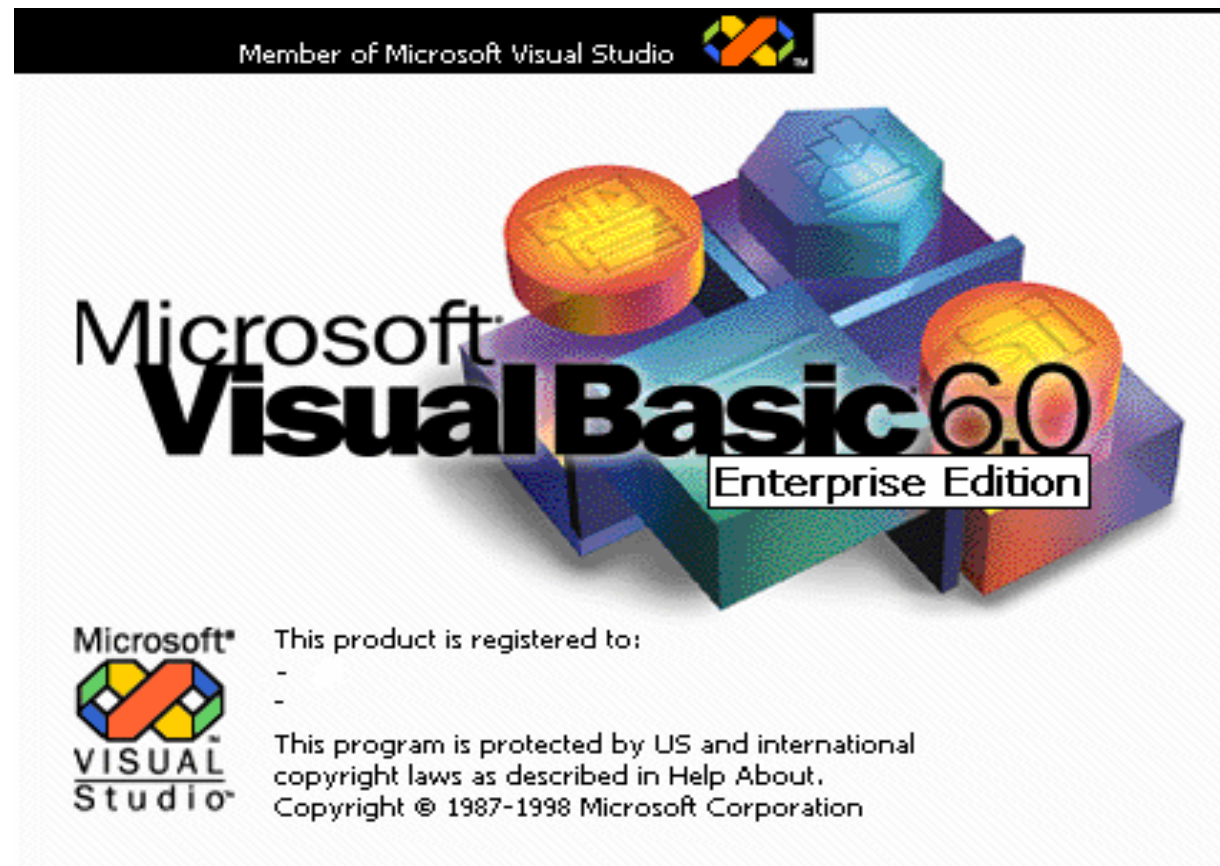


Introducción a la programación en Visual Basic 6.0

Prof. Roberto Martín Murdocca
Ing. Alejandro Nuñez
Área de Electrónica y Microprocesadores



Visual Basic 6.0





Visual Basic 6.0

- n Es un lenguaje de programación orientado a eventos.
- n En la programación orientada a eventos el usuario es el que dirige el flujo del programa.
- n El usuario interactúa con el sistema realizando eventos con los objetos de la interfaz gráfica.



Visual Basic 6.0

n Características

- n Diseñado para desarrollar aplicaciones para el entorno de Windows 9x o superior.
- n Implementa una programación orientada a objetos.
- n El desarrollo se realiza en un entorno gráfico.
- n La sintaxis del lenguaje es similar al lenguaje natural.



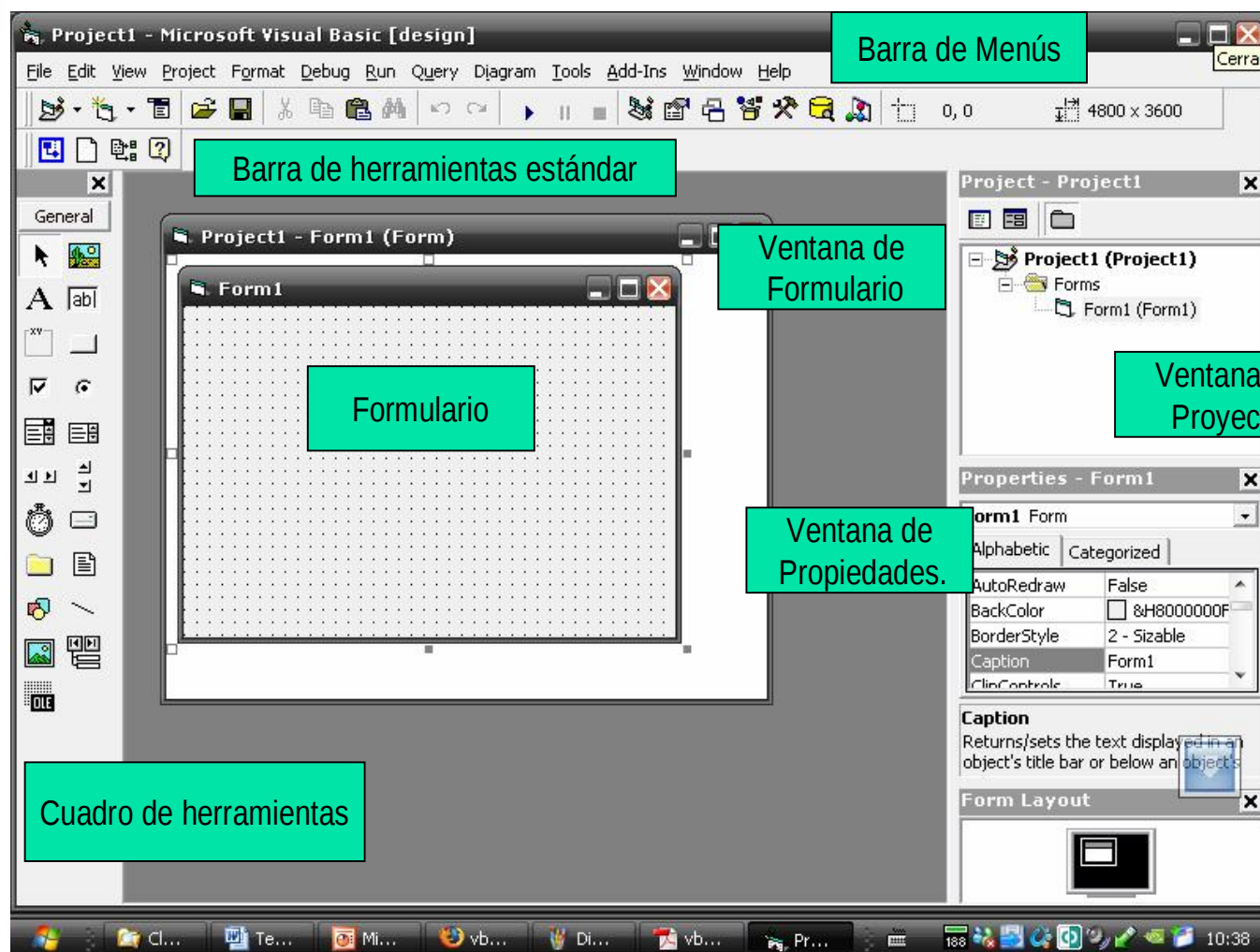
Visual Basic 6.0

n Características

- n Tiene una gran aplicación en sistemas pequeños por su alta velocidad en el desarrollo.
- n Permite el tratamiento de mensajes de Windows.
- n Gran parte del trabajo en el diseño ya está realizado, al incorporar los controles al lenguaje.
- n Permite crear controles personalizados.
- n Permite generar librerías dinámicas (DLL).



Visual Basic 6.0

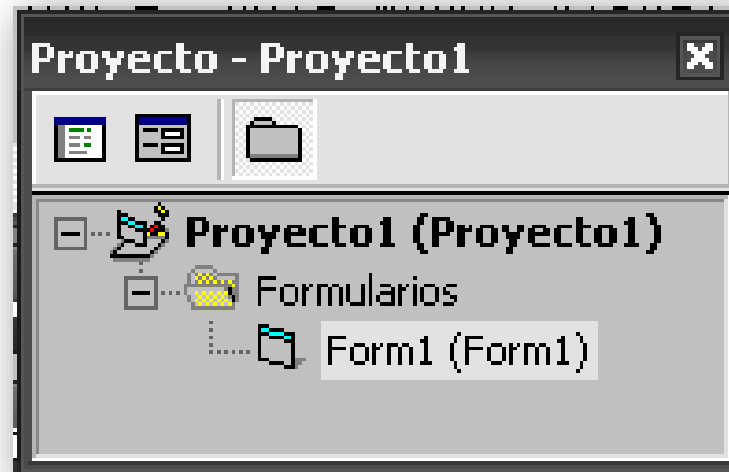




Visual Basic 6.0

n Explorador de Proyecto

- n Contiene la lista de los archivos que forman parte de la aplicación





Visual Basic 6.0

- n Explorador de Proyecto

- n Archivo de Proyecto:

- n Realiza el seguimiento de todos los archivos que forman parte de la aplicación (*.VBP).

- n Módulo de Formulario:

- n controles y código, sólo hay uno por formulario. (*.FRM)

- n Módulo Estándar:

- n Sólo pueden contener código (*.BAS)



Visual Basic 6.0

n Explorador de Proyecto

n Archivo de Recursos:

- n Aquí se guardan cadenas de texto, mapas de bits, y demás datos que puedan modificarse sin tener que volver a modificar el código. (*.RES)

n Módulo de Clase:

- n Son similares a los módulos de formulario. (*.CLS)

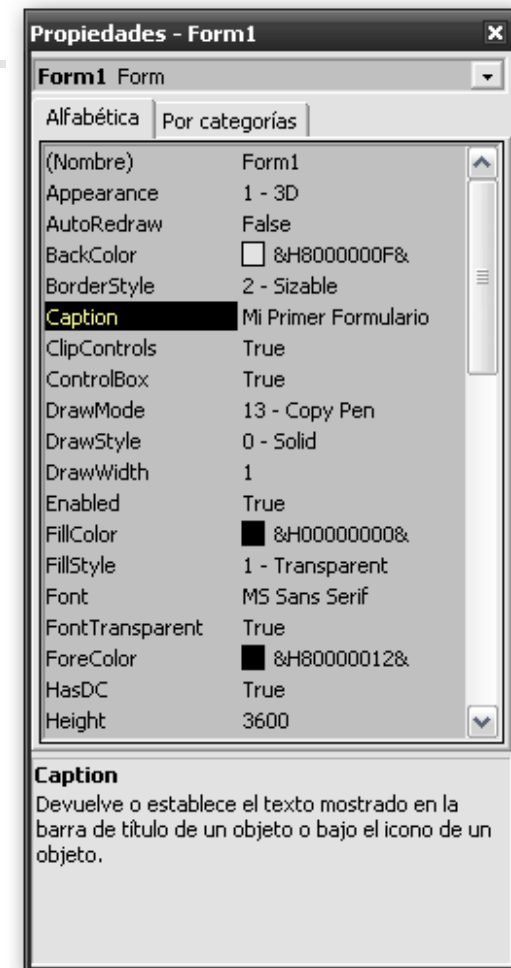
n Controles ActiveX:

- n Controles que se pueden añadir al cuadro de herramientas e incluirlos en un formulario.



Visual Basic 6.0

- n Ventana de Propiedades
 - n Los objetos tiene asociados propiedades que los describen
 - n Atributos
 - n Valores
 - n Comportamiento
 - n Apariencia





Visual Basic 6.0

n Ventana de Propiedades

n Las opciones de esta ventana son:

n Lista desplegable de objetos:

- n visualizar el nombre de los objetos contenidos por el formulario.

n Lista de propiedades del objeto seleccionado:

- n propiedades del mismo (name, visible, appearance, borderstyle, etc...).



n Cuadro de Herramientas

- n se encuentran un conjunto de herramientas que permiten insertar los objetos o controles en el formulario durante en tiempo de diseño.
- n los objetos más comunes son:
 - n botones (command)
 - n etiqueta (label)
 - n cuadros de texto (textbox)





Visual Basic 6.0

n Diseñador de Formularios

- n Es la ventana en la que se diseñará la interfaz de la aplicación.
- n En ella se pueden agregar controles gráficos e imágenes.
- n Cada formulario de una aplicación aparecerá en su propia ventana.



Visual Basic 6.0

n





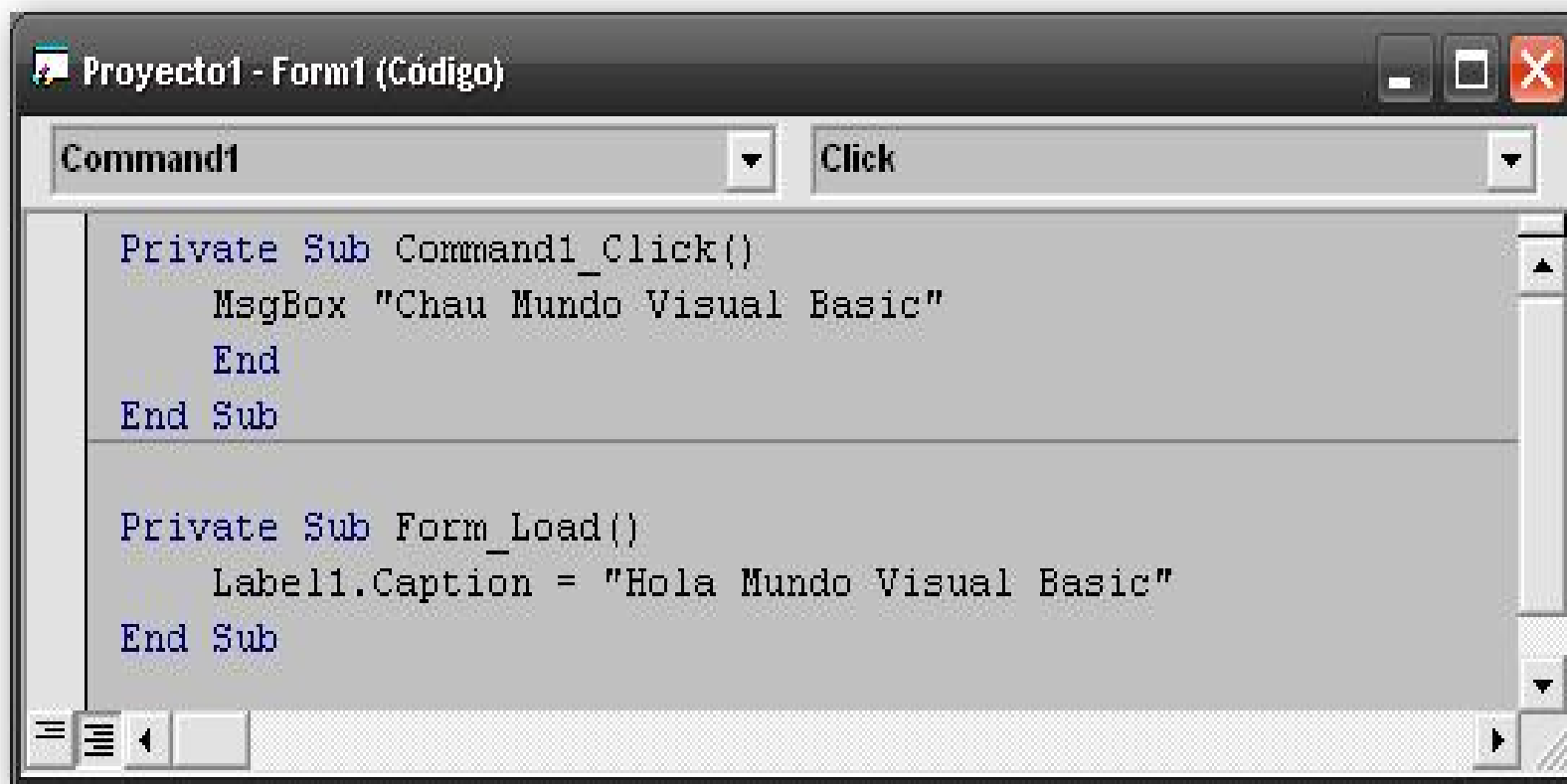
Visual Basic 6.0

n Diseñador de Código

- n En esta ventana es donde se incluye el código de la aplicación.
- n Se creará una ventana de código para cada formulario o módulo de la aplicación.



Visual Basic 6.0



The screenshot shows the Visual Basic 6.0 IDE with the 'Proyecto1 - Form1 (Código)' window open. The window has a dropdown menu set to 'Command1' and another set to 'Click'. The code editor contains the following VBA code:

```
Private Sub Command1_Click()  
    MsgBox "Chau Mundo Visual Basic"  
End  
End Sub  
  
Private Sub Form_Load()  
    Label1.Caption = "Hola Mundo Visual Basic"  
End Sub
```



OBJETOS

- n Objetos

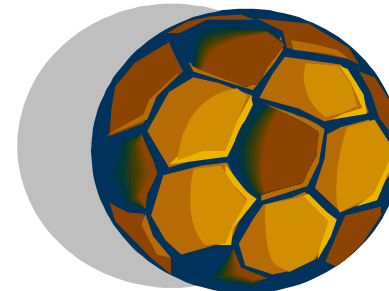
- n Un objeto es una entidad que tiene asociado un conjunto de:

- n métodos

- n propiedades

- n eventos

- n Por ejemplo una pelota





OBJETOS

n Objetos - Propiedades

n son las características de los objetos

n Por ejemplo:

n Diámetro

n Color

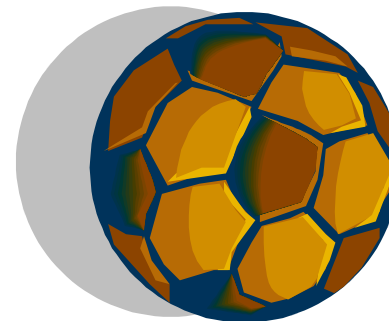
n Peso

n Presión

n Relleno

n Etiqueta

n Material





OBJETOS

n Objetos - Propiedades

n Por ejemplo:

- n Diámetro = 70 cm
- n Color = "Amarillo"
- n Peso = 450 g
- n Presión = 1005 g/cm²
- n Relleno = "Aire"
- n Etiqueta = "Mi Pelota"
- n Material = "Cuero"





OBJETOS

n Objetos - Métodos

n Son acciones que pueden, o que saben, realizar los objetos.

n Por ejemplo:

- n Botar
- n Mover $x=5$ $y=10$
- n Inflar
- n Desinflar
- n Pinchar





OBJETOS

n Objetos - Eventos

n Es un suceso que le puede ocurrir a un objeto

n Por ejemplo:

n Patearon

n Agarraron

n Lanzaron

n Apretaron





Formulario - propiedades

n Algunas propiedades:

- n BorderStyle: Establece el estilo del borde del formulario.
- n Caption: Establece el texto que aparece en la barra de título del formulario.
- n ControlBox: Permite mostrar o no el menú de control de las ventanas de Windows.
- n Enabled: Establece si el formulario puede responder o no a los eventos.
- n Font: Establece las características de los objetos de texto que se sitúen en el formulario.



Formulario - propiedades

- n Icon: Cambia el icono que representa el formulario.
- n Left, Top: Establecen la posición del formulario en la pantalla.
- n Height, Width: Establecen las dimensiones.
- n MaxButton: Establece si se podrá maximizar el formulario.
- n MinButton: Establece si se podrá minimizar el formulario.
- n MousePointer: Modifica el puntero del ratón.



Formulario - propiedades

- n Visible: Establece si el formulario se mostrará visible en tiempo de ejecución.
- n WindowState: Establece el modo en que se carga inicialmente el formulario, normal, maximizado o minimizado.



Formulario - propiedades

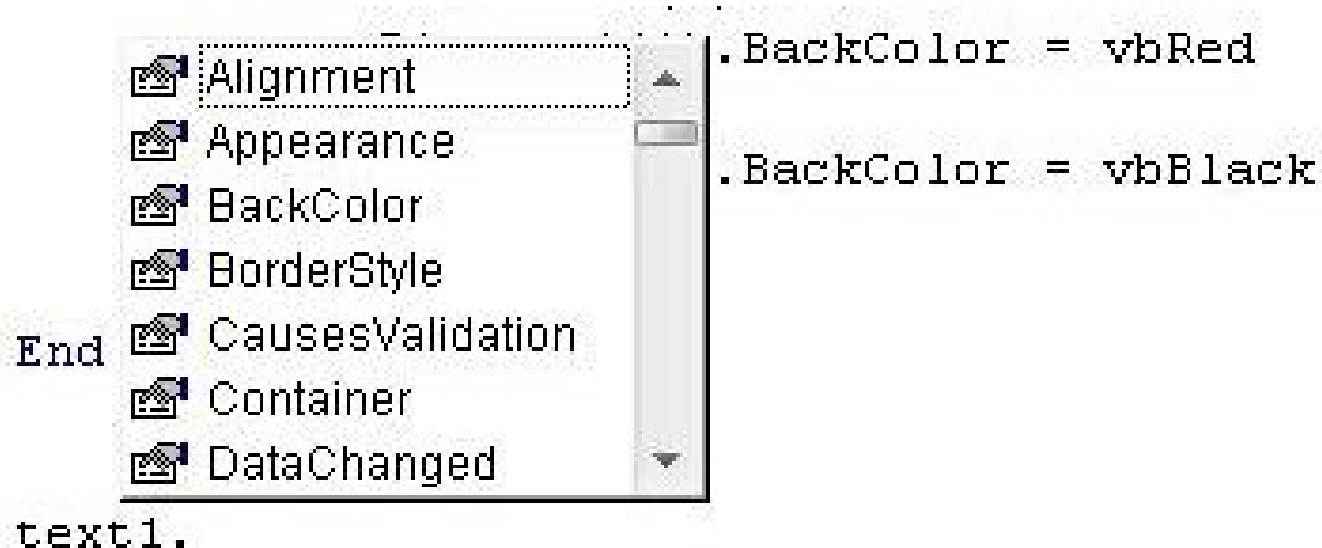
- n Todas estas propiedades están disponibles en tiempo de diseño a través de la ventana de propiedades.
- n En tiempo de ejecución se puede cambiar o consultar el valor de algunas propiedades que sólo tienen sentido en el tiempo de desarrollo.

NombreObjeto . NombrePropiedad



Formulario - propiedades

Ejemplo: en la pantalla de código el objeto text1 muestra sus propiedades





Formulario - eventos

- n Los eventos están predefinidos en Visual Basic para cada objeto.
- n Cada evento de un objeto tiene un procedimiento asignado.
- n Estos procedimientos son bloques de código que permiten especificar la respuesta del objeto ante la ocurrencia del evento.

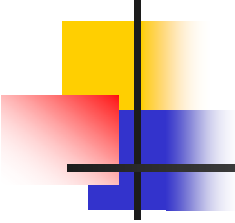


Formulario - eventos

- n La definición de un procedimiento de evento tiene la siguiente forma:

```
Private Sub NomObjeto_NomEvento ([*])  
    ...instrucciones...  
End Sub
```

[*]Lista de parámetros



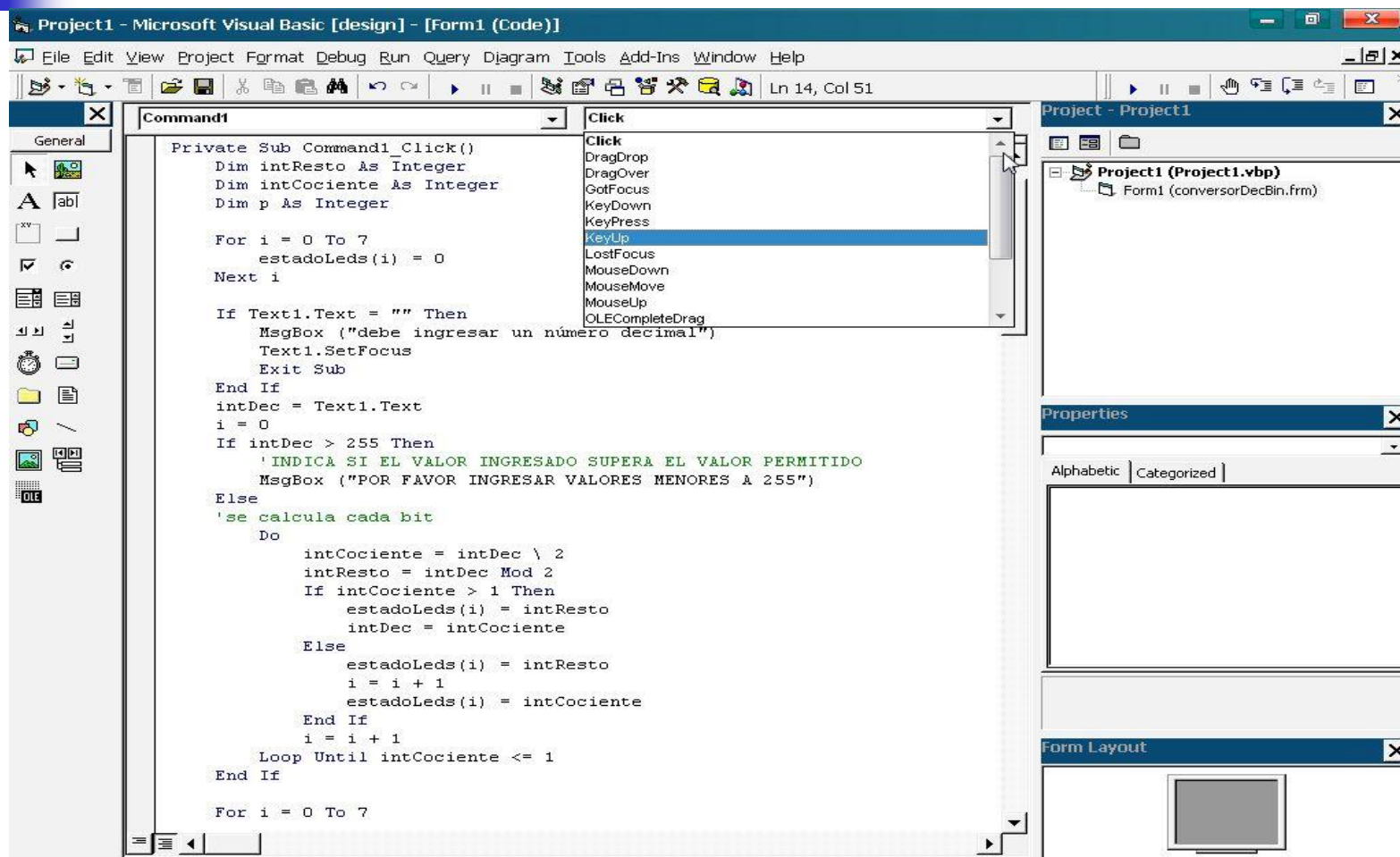
Formulario - eventos

```
Private Sub Command1_Click ()  
    MsgBox "Chau Mundo Visual Basic"  
End  
End Sub
```

```
Private Sub Form_Load()  
    Label1.Caption = "Hola Mundo Visual Basic"  
End Sub
```



Formulario - eventos





Formulario - métodos

- n Síntaxis:

NombreObjeto.NombreMétodo[par1,...,parn]

- n donde la lista de parámetros [par1...,parn] depende de cada método.

- n por ejemplo:

- n mostrar un formulario:

frmPrimero.**Show**

- n ocultar un formulario:

frmPrimero.**Hide**



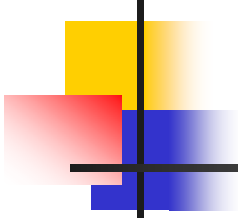
Variables

n Variables:

n espacio en memoria que almacena un conjunto de información, haciendo referencia al mismo por un nombre.

n Declaración:

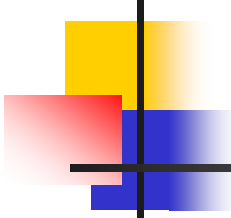
Dim nombre_variable as tipo_dato



Variables

n Tipos de datos

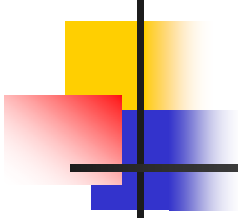
Tipo	en VB	Tamaño
Entero	Integer	2 bytes
Entero Largo	Long	4 bytes
Simple	Single	4 bytes
Doble	Double	8 bytes
Moneda	Currency	8 bytes
Cadena de caracteres	String	1 byte x caracter



Variables

n Tipos de datos

Tipo	en VB	Tamaño
Byte	Byte	1 byte
Booleano	Boolean	2 bytes
Fecha	Date	8 bytes
Objecto	Object	4 bytes
Variable	Variant	16 bytes + 1 byte x caracter



Variables

n Valores Enteros

- n Integer +/- 32,767
- n Long +/- 2,147,483,647
- n Currency +/- $9,2 \times 10^{14}$

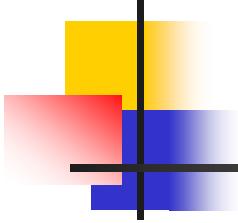
n Punto Flotante

n Single

- n +/- $1,4 \times 10^{-45}$
- n +/- $3,4 \times 10^{38}$

n Double

- n +/- $1,7 \times 10^{-324}$
- n +/- $4,9 \times 10^{308}$



Variables

- n String

- n Hasta 65356 caracteres

- n Date

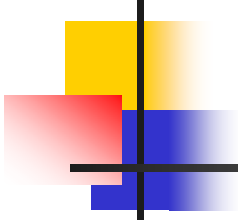
- n 1/1/100 – 31/12/9999

- n Byte

- n 0 a 255 (datos binarios)

- n Boolean

- n True o False



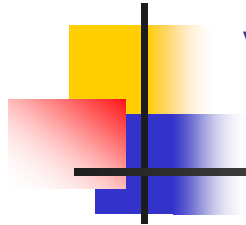
Variables

n Ejemplos

n Dim x as Integer

n Private x as Integer

n Public x as Integer

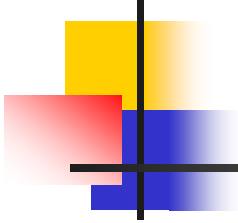


Variables

- n Constantes

- n Const MaximoValor = 500

- n Const Direccion = "C:\Miss"



Variables

Alcance de las variables

n Locales

- n** Al procedimiento o función en donde se declaren:

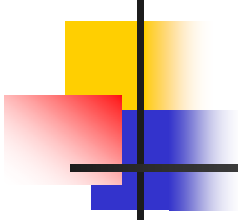
Dim nombre_variable as tipo

- n** Al formulario si se declaran antes de los Procedimientos de Eventos:

Private nombre_variable as tipo

- n** Al proyecto si se declaran como públicas:

Public nombre_variable as tipo



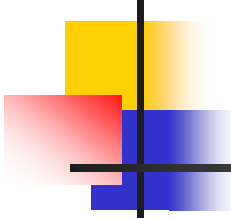
Variables

Alcance de las variables

- n Globales

- n Se deben declarar en un módulo:

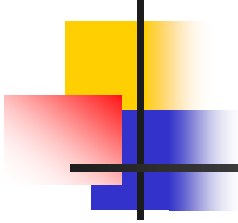
- n **Global** nombre_variable as Integer



Variables

ⁿ Operadores Matemáticos

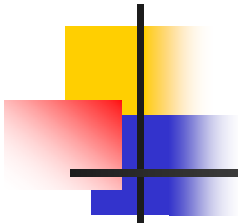
Operador	Operación
+	Suma – Concatena
-	Resta
/	División Real
\	División Entera
Mod	Resto de la División Entera
&	Concatena
=	Asignación



Variables

n Operadores de Comparación

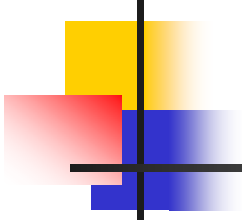
Operador	Operación
<	Menor que
<=	Menor o igual
=	Igual
>=	Mayor o igual
>	Mayor que
<>	Distinto



Variables

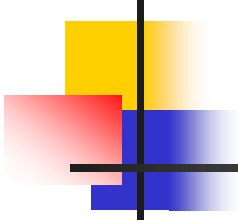
n Operadores Lógicos

Operador	Operación	En C
AND	Y lógico	&&
OR	O Lógico	
NOT	Negación	!



Estructuras de Control

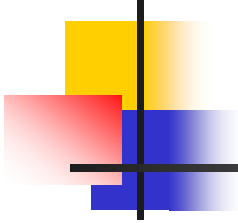
- n Permiten modificar el flujo de ejecución de las instrucciones de un programa.
- n Tipos:
 - n De acuerdo a una condición, ejecutar un grupo u otro de sentencias (If-Then-Else y Select-Case)
 - n Ejecutar un grupo de sentencias mientras se cumpla una condición (Do-While)
 - n Ejecutar un grupo de sentencias hasta que se cumpla una condición (Do-Until)
 - n Ejecutar un grupo de sentencias un número determinado de veces (For-Next)



Estructuras de Control

- n Condicional, If
 - If *condición* Then
 - Instrucciones... (Condición Verdadera)
 - End If

 - If *condición1* Then
 - Instrucciones... (Condición1 Verdadera)
 - ElseIf *condición2* Then
 - Instrucciones... (Condición2 Verdadera)
 - Else
 - Instrucciones... (Condición Falsa)
 - End If



Estructuras de Control

n Condicional, If

If $a > b$ Then

 mayor = a

End If

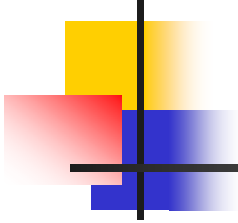
If $a > b$ Then

 mayor = a

Else

 mayor = b

End If



Estructuras de Control

n Condicional, If

If $a > b$ Then

mayor = a

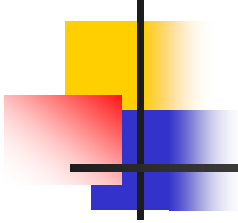
ElseIf $a < b$ then

mayor = b

Else

mayor = 0

End If



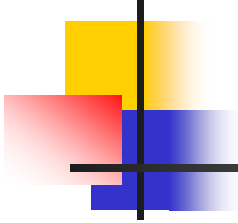
Estructuras de Control

n Repetitiva, For

For *variable* = *inicio* To *final*

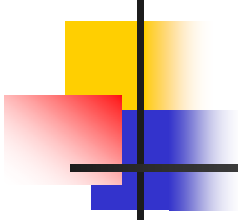
Instrucciones...

Next *variable*



Estructuras de Control

```
n Repetitiva,      For
    Dim numero as Integer
    Dim i as Integer
    For i = 1 To 10
        numero = numero * 2
    Next
    MsgBox str(numero)
```



Estructuras de Control

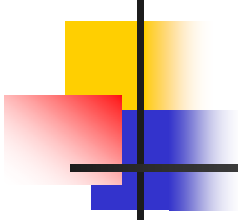
n Repetitiva, For

n variante, incremento distinto de 1 o decremento

For ... To *final* step *incremento*

Instrucciones...

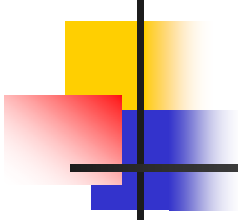
Next *variable*



Estructuras de Control

n Repetitiva, For

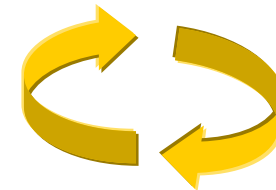
```
Dim i As Integer
Dim par As String
For i = 2 To 50 Step 2
    par = par + str(i)
Next i
MsgBox par
```



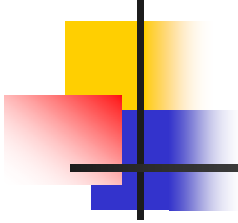
Estructuras de Control

- n Repetitiva, While ... Wend
 - n hacer mientras que

While (condición)
instrucciones ...
Wend

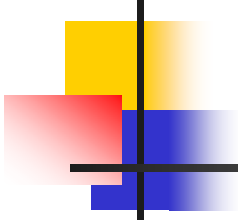


Condición Falsa o Verdadera



Estructuras de Control

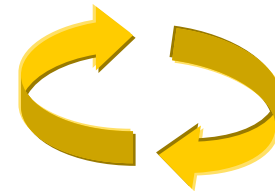
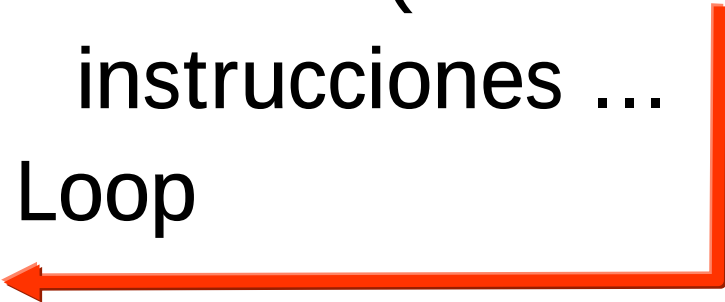
```
n Repetitiva, While ... Wend
  n Mientras que
    Dim i As Integer
    Dim pares As String
    While (i < 50)
      pares = pares + ", " + str(i)
      i = i + 2
    Wend
    MsgBox pares
```



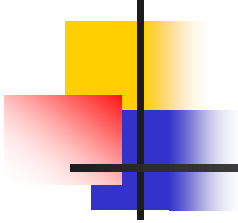
Estructuras de Control

- n Repetitiva, Do While ... Loop
 - n hacer mientras que

Do While (condición)
instrucciones ...
Loop



Condición Falsa o Verdadera



Estructuras de Control

n Repetitiva, Do While ... Loop

n hacer mientras que

Dim i As Integer

Dim pares As String

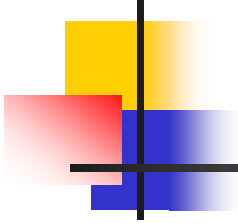
Do While (i < 50)

pares = pares + ", " + str(i)

i = i + 2

Loop

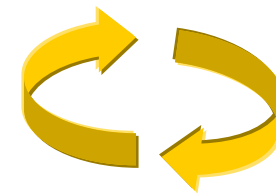
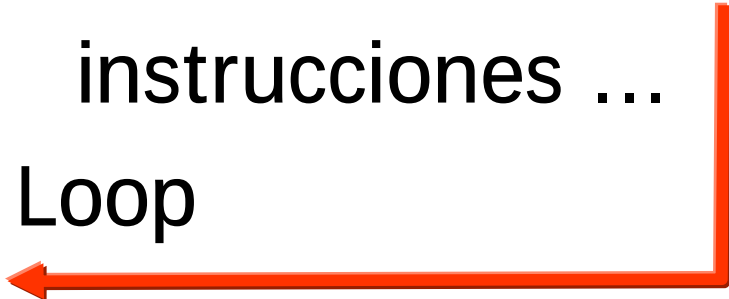
MsgBox pares



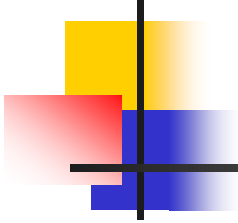
Estructuras de Control

- n Repetitiva, Do Until ... Loop
 - n hacer hasta que

Do Until (condición)
instrucciones ...
Loop



Condición Verdadera o Falsa



Estructuras de Control

n Repetitiva, Do Until ... Loop

n hacer hasta que

Dim i As Integer

Dim pares as String

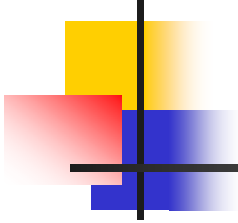
Do Until (i = 50)

pares = pares + ", " + str(i)

i = i + 2

Loop

MsgBox pares



Estructuras de Control

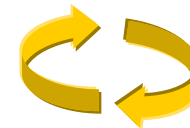
n Repetitiva, Do ... Loop While

n Repetir mientras que

Do

Falsa o Verdadera

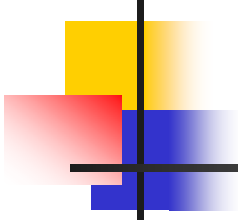
instrucciones ...



Loop While *condición*



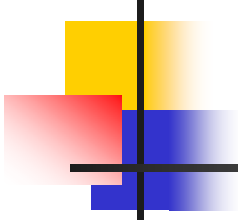
n Las instrucciones se ejecutan por lo menos una vez.



Estructuras de Control

n Repetitiva, Do ... Loop While

```
Dim i As Integer
Dim pares as String
Do
    pares = pares + str(i)
    i = i + 2
Loop While (i < 50)
MsgBox pares
```



Estructuras de Control

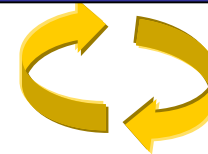
n Repetitiva, Do ... Loop Until

n Repetir hasta que

Do

Verdadera o Falsa

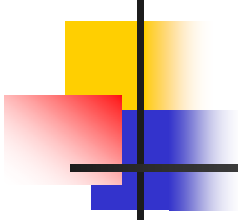
instrucciones ...



Loop While *condición*



n Las instrucciones se ejecutan por lo menos una vez.



Estructuras de Control

n Repetitiva

Do ... Loop Until

n Repetir hasta que

Dim i As Integer

Dim pares As String

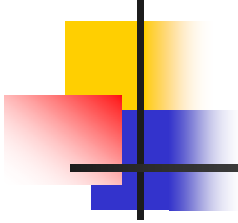
Do

pares = pares + str(i)

i = i + 2

Loop Until (i < 50)

MsgBox pares



Estructuras de Control

- n Condicional Selectiva, Select Case

Select Case *variable*

Case *valor_1*

instrucciones...

Case *valor_2*

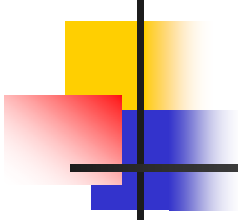
instrucciones...

...

Case *Else*

instrucciones ...

End Select



Estructuras de Control

```
Select Case mes
```

```
  Case 2
```

```
    If (año Mod 4 = 0) Then
```

```
      max_dia = 29
```

```
    Else
```

```
      max_dia = 28
```

```
    End If
```

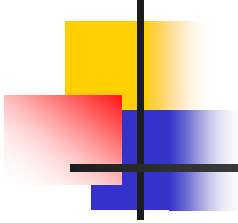
```
  Case 1, 3, 5, 7, 8, 10, 12
```

```
    max_dia = 31
```

```
  Case 4, 6, 9, 11
```

```
    max_dia = 30
```

```
End Select
```



Estructuras de Control

```
Select Case mes
  Case "febrero"
    If (año Mod 4 = 0) Then
      max_dia = 29
    Else
      max_dia = 28
    End If
  Case "enero", "marzo", ...
    max_dia = 31
  Case "abril", "junio", ...
    max_dia = 30
End Select
```



Nombre de los Objetos

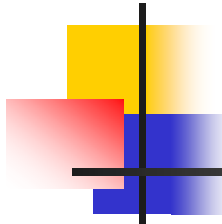
Objeto	Traducción	Prefijo
Form	Formulario	frm
CheckBox	Casilla de verificación	chk
ComboBox	Cuadro combinado	cbo
Command Button	Botón de comando	btn
Frame	Marco	fra
Grid	Rejilla	grd
Image	Imagen	img
Label	Etiqueta	lbl
PictureBox	Cuadro de imagen	pic
Shape	Forma	shp
Text Box	Cuadro de texto	txt
Timer	Temporizador	tmr



Nombre de los Objetos

Abreviatura	Control	Abreviatura	Control
chk	CheckBox	cbo	Combo y Drop-ListBox
cmd	CommandButton	dir	DirListBox
drv	DriveListBox	fil	FileListBox
frm	Form	fra	Frame
hsb	HorizontalScrollBar	img	Image
lbl	Label	lin	Line
lst	List	mnu	Menu
opt	OptionButton	pct	PictureBox
shp	Shape	txt	TExtEditBox
tmr	Timer	vsb	VerticalScrollBar

Abreviaturas para los controles más usuales.

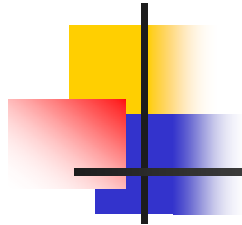


Leer Puerto Paralelo

Para tener acceso al puerto paralelo se utiliza una dll llamada io.dll para xp y INPOUT32.DLL para windows 98. Las funciones de esta dll se accede redefiniéndolas en vb

```
Public Declare Sub Out Lib "io.dll" Alias "PortOut" (ByVal Port As Integer, ByVal Data As Byte)
Public Declare Sub PortWordOut Lib "io.dll" (ByVal Port As Integer, ByVal Data As Integer)
Public Declare Sub PortDWordOut Lib "io.dll" (ByVal Port As Integer, ByVal Data As Long)
Public Declare Function Inp Lib "io.dll" Alias "PortIn" (ByVal Port As Integer) As Byte
Public Declare Function PortWordIn Lib "io.dll" (ByVal Port As Integer) As Integer
Public Declare Function PortDWordIn Lib "io.dll" (ByVal Port As Integer) As Long
Public Declare Sub SetPortBit Lib "io.dll" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Sub ClrPortBit Lib "io.dll" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Sub NotPortBit Lib "io.dll" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Function GetPortBit Lib "io.dll" (ByVal Port As Integer, ByVal Bit As Byte) As Boolean
Public Declare Function RightPortShift Lib "io.dll" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Public Declare Function LeftPortShift Lib "io.dll" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Public Declare Function IsDriverInstalled Lib "io.dll" () As Boolean
```

La dll debe estar en la misma carpeta en donde se guarda el proyecto. Las funciones "out", "in" etc se redefinen en un módulo o en el mismo formulario como Public.



Leer Puerto Paralelo

Lectura de una dirección llamada STINR

Do

 cont = Inp(STINR) And &H4

Loop Until cont = 4

Do

 cont = Inp(STINR) And &H4

Loop Until cont = 0

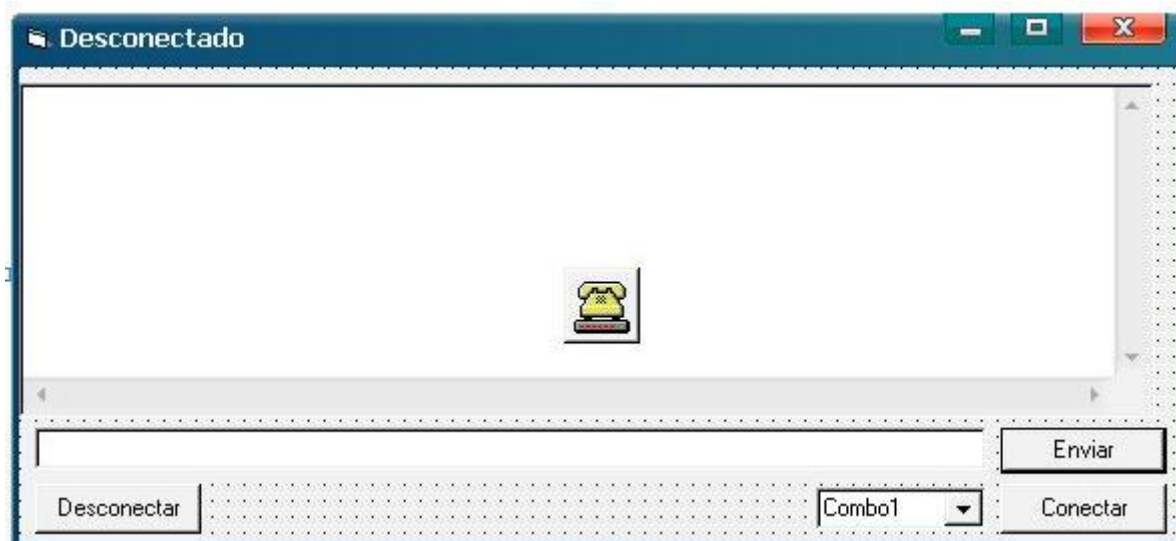
Escritura en dos direcciones llamadas COWOR y CONT2

Out COWOR, &HB0

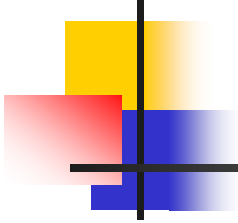
Out CONT2, &H8

Out CONT2, &H0

Leer Puerto Serie

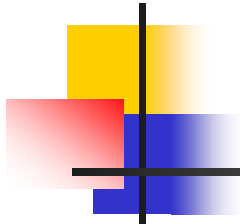


El objeto con forma de teléfono es el encargado de realizar la comunicación con el puerto serie



Leer Puerto Serie

```
Private Sub Conectar_Click()  
    'comprueba que el puerto este cerrado para poder abrirlo  
    If MSComm1.PortOpen = False Then  
        'determina el puerto que hemos seleccionado  
        If Combo1.ListIndex = 0 Then  
            MSComm1.CommPort = 1  
        Else  
            MSComm1.CommPort = 2  
        End If  
        'determina: 9600-Velocidad en Baudios, N-No utiliza ninguna paridad,  
        '8-Cantidad de bits de envio y recepcion por paquete,  
        '1-Determina los bits de parada  
        MSComm1.Settings = "9600,N,8,1"  
        'lee todo el buffer de entrada para que quede vacio  
        MSComm1.InputLen = 0  
        'Abre el puerto seleccionado  
        MSComm1.PortOpen = True  
        Me.Caption = "Conectado por el puerto " & MSComm1.CommPort  
    End If  
End Sub
```

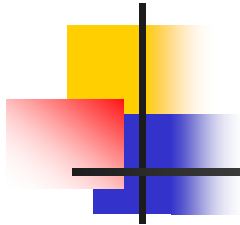


Leer Puerto Serie

```
Private Sub Desconectar_Click()  
    If MSComm1.PortOpen Then  
        'cierra el puerto  
        MSComm1.PortOpen = False  
        Me.Caption = "Desconectado"  
    End If  
End Sub
```

```
Private Sub Enviar_Click()  
    'envia el texto escrito.  
    MSComm1.Output = TextoEnviar.Text & vbCrLf  
    'coloca el texto que enviamos en la pantalla  
    Texto.Text = Texto.Text & vbCrLf & "<Ordenador 1>" & TextoEnviar.Text  
    TextoEnviar.SetFocus  
End Sub
```

Crear y escribir un archivo de texto

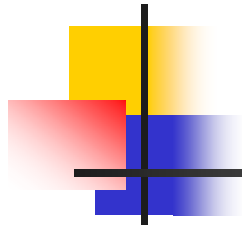


```
Private Sub guardapulsos()  
Dim fso, pulsodato, en  
Dim t As Single
```

```
m = 2 * k  
Set fso = CreateObject("scripting.filesystemobject")  
Set pulsodato = fso.createtextfile((nombrepulsos & k & ".txt"), True)
```

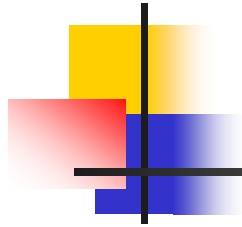
```
For t = 1 To CMuestras * CPulsos  
    pulsodato.WriteLine valorV(m, t) & ";" & valorV(m + 1, t) & ";" & valorI(m, t) & ";" & valorI(m + 1, t) & ";"  
Next
```

```
pulsodato.Close  
End Sub
```



Leer un archivo de texto

```
Open (nombrepulso & n & ".txt") For Input As 1
  For i = 1 To bb
    Line Input #1, lineaDeTexto
    x = Split(lineaDeTexto, ";")
    Vx = CDBl(x(0))
    Vy = CDBl(x(1))
    Picture0.CurrentX = Vx
    Picture0.CurrentY = Vy
    Picture0.ForeColor = vbRed
    Picture0.DrawWidth = 1
    Picture0.Scale (0, 1)-(dd, 0)
    Picture0.DrawStyle = 0
    Picture0.Line (Vx0, Vy0)-(Vx, Vy)
    Vx0 = Vx
    Vy0 = Vy
  Next
Close
```



Graficar un Función

```
For i = 1 To bb
```

```
    lxx = valorl(jj, i)
```

```
    lyy = valorl(zz, i)
```

```
    Picture0.CurrentX = lxx
```

```
    Picture0.CurrentY = lyy
```

```
    Picture0.ForeColor = vbRed
```

```
    Picture0.DrawWidth = 1
```

```
    Picture0.Scale (0, 1)-(dd, 0)
```

```
    Picture0.DrawStyle = 0
```

```
    Picture0.Line (lxx0, lyy0)-(lxx, lyy)
```

```
    lxx0 = lxx
```

```
    lyy0 = lyy
```

```
Next
```