

## ¿Qué es Visual Basic?

Visual Basic (VB) es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

En 2001 Microsoft propuso abandonar el desarrollo basado en la API Win32 y pasar a un framework o marco común de librerías, independiente de la versión del sistema operativo .NET Framework, a través de Visual Basic .NET (y otros lenguajes como C Sharp(C#) de fácil transición de código entre ellos); fue el sucesor de Visual Basic 6.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando Data Access Objects, Remote Data Objects o ActiveX Data Objects.

Visual Basic contiene un entorno de desarrollo integrado o IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI.

## ¿Qué aplicación usar?

Microsoft VBA (Visual Basic para aplicaciones) es el lenguaje de macros de Microsoft Visual Basic que se utiliza para programar aplicaciones Windows y que se incluye en varias aplicaciones Microsoft. VBA permite a usuarios y programadores ampliar la funcionalidad de programas de la suite Microsoft Office. Visual Basic para Aplicaciones es un subconjunto casi completo de Visual Basic 5.0 y 6.0.

Microsoft VBA viene integrado en aplicaciones de Microsoft Office, como Outlook, Word, Excel, Access y Powerpoint. Prácticamente cualquier cosa que se pueda programar en Visual Basic 5.0 o 6.0 se puede hacer también dentro de un documento de Office, con la sola limitación que el producto final no se puede compilar separadamente del documento, hoja o base de datos en que fue creado; es decir, se convierte en una macro (o más bien súper macro). Esta macro puede instalarse o distribuirse con sólo copiar el documento, presentación o base de datos.

Su utilidad principal es automatizar tareas cotidianas, así como crear aplicaciones y servicios de bases de datos para el escritorio. Permite acceder a las funcionalidades de un lenguaje orientado a eventos con acceso a la API de Windows.

Al provenir de un lenguaje basado en Basic tiene similitudes con lenguajes incluidos en otros productos de ofimática como StarBasic y Openoffice.

# 5 Proyectos desarrollados en visual Basic:

- **EJERCICIO EJEMPLO RESUELTO Nº1**

Transformar el algoritmo que se indica a continuación, relativo a calculo de necesidades de combustible, en un programa de Visual Basic.



1. **Inicio**
2. [Definición de vehículos]
3. Turismos = 32
4. Todoterrenos = 11
5. [Fin de definición de vehículos]
6. [Definición de capacidad de depósitos]
7. Capturismos = 40
8. Captodot = 65
9. [Fin de definición de capacidad de depósitos]
10. [Cálculo de necesidades de combustible]
11. Necesidadescom = Turismos \* Capturismos + Todoterrenos \* Captodot
12. [Fin de cálculo de necesidades de combustible]
13. **Fin**

**SOLUCIÓN EJERCICIO EJEMPLO Nº 1:** El código será algo como lo que se muestra a continuación. Antes de escribir el código tenemos que crear los objetos (controles) a que vamos a hacer referencia en el código, en concreto crearemos 6 labels, 4 TextBox y un botón.

Código (versionesVB menos recientes)	Código (versiones VB más recientes)
<pre> Rem Curso Visual Basic aprenderaprogramar.com <b>Option Explicit</b> Rem Declaración de variables Dim Turismos%, Todoterrenos As Integer Dim Capturismos!, Captodot As Single Dim Necesidadescom As Single  Rem Contenido del formulario <b>Private Sub Form_Load()</b> Form1.Caption = "Cálculo de necesidades combustible" </pre>	<pre> REM Curso Visual Basic aprenderaprogramar.com <b>Option Explicit On</b> Public Class Form1     REM Declaración de variables     Dim Turismos, Todoterrenos As Integer     Dim Capturismos, Captodot As Single     Dim Necesidadescom As Single      REM Contenido del formulario     <b>Private Sub Form1_Load</b>(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles  MyBase.Load </pre>

```

Label1(0) = "Por favor introduzca estos datos"
Label1(1) = "Número de turismos"
Label1(2) = "Número de todoterrenos"
Label1(3) = "Capacidad de turismos (litros)"
Label1(4) = "Capacidad de todoterrenos (litros)"
Command1.Caption = "Aceptar"
End Sub

```

```

Rem Cálculo y muestra resultados
Private Sub Command1_Click()
Label2.ForeColor = vbBlack
Label2.FontBold = True
Turismos = Val(Text1(0))
Todoterrenos = Val(Text1(1))
Capturismos = Val(Text1(2))
Captodot = Val(Text1(3))
Necesidadescom = Turismos * Capturismos +
Todoterrenos * Captodot
Label2 = "Las necesidades totales de
combustible son de " & Necesidadescom & "
litros"
End Sub

```

```

Me.Text = "Cálculo de necesidades combustible"
Label1.Text = "Por favor introduzca estos datos"
Label2.Text = "Número de turismos"
Label3.Text = "Número de todoterrenos"
Label4.Text = "Capacidad de turismos (litros)"
Label5.Text = "Capacidad de todoterrenos (litros)"
Button1.Text = "Aceptar"
End Sub

```

```

Rem Cálculo y muestra resultados
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
Label6.ForeColor = Color.Black
Label6.Font = New Font("Arial", 10, FontStyle.Bold)
Turismos = Val(TextBox1.Text)
Todoterrenos = Val(TextBox2.Text)
Capturismos = Val(TextBox3.Text)
Captodot = Val(TextBox4.Text)
Necesidadescom = Turismos * Capturismos +
Todoterrenos * Captodot
Label6.Text = "Las necesidades totales de
combustible son de " & Necesidadescom & " litros"
End Sub
End Class

```

Téngase en cuenta que se han declarado como código diversas órdenes que podrían haberse introducido desde el diseño de objetos previo. Ten en cuenta también que todos los objetos que invocamos en el código los hemos creado previamente sobre el formulario (menú Ver - Objeto); los nombres de los objetos los pone Visual Basic (como en este ejemplo) al crearlos, o los establecemos en la ventana Propiedades nosotros mismos si deseamos cambiarlos. Si tratas de invocar un objeto no definido o con un nombre equivocado te saldrá un mensaje de error.

En un pequeño programa como este en el que tenemos varios labels, podemos optar tanto por usar arrays de controles (en este caso los controles tendrán nombres como Label1(0), Label1(1), Label1(2), etc.) o simplemente distintos controles (en este caso los nombres serán como Label1, Label2, Label3, etc.). Lo mismo podemos decir en relación a las cajas de texto (TextBox). Recordar que en las versiones menos recientes de Visual Basic es más sencillo crear arrays de controles que en las más recientes, ya que basta copiar y pegar un control y Visual Basic pregunta directamente si queremos crear el array.

El resultado gráfico después de introducir los datos y pulsar aceptar será algo así:

## EJERCICIO EJEMPLO RESUELTO N°2

Transformar en código el pseudocódigo que se muestra a continuación, relativo a proporcionar el volumen de un cilindro dados su altura y diámetro.

### 1. Inicio

2. Mostrar "Introduzca el diámetro, en metros" :  
Pedir D
3. Mostrar "Introduzca la altura, en metros" :  
Pedir H
4.  $R = D/2$  :  $Pi = 3,141593$
5.  $V = Pi * (R ^ 2) * H$
6. Mostrar "El volumen del cilindro es de", V,  
"metros cúbicos"

### 7. Fin

## SOLUCIÓN EJERCICIO EJEMPLO N°2

Creamos un formulario donde insertamos dos Labels (podemos usar arrays de controles como Label1(0) yLabel1(1)) ó simplemente dos controles como Label1 y Label2, dos TextBox (podemos decir lo mismo que para los labels)y un Label para resultados al que vamos a ponerle nosotros el nombre y que llamaremos (LabelResultados), así como un botón al que llamaremos ButtonCalcular. El primer Label mostrará el mensaje de solicitud del diámetro y el segundo Label el correspondiente a la altura. A través de los TextBox el usuario introducirá los datos solicitados y a través del Label de resultados mostraremos los resultados. El código será algo así:

Código (versionesVB menos recientes)	Código (versiones VB más recientes)
<pre>Rem Curso Visual Basic aprenderaprogramar.com <b>Option Explicit</b> Rem Declaración de variables Const Pi As Single = 3.141592654 Dim R!, H!, V As Single  Rem Contenido del formulario <b>Private Sub Form_Load()</b> Form1.Caption = "Cálculo volumen cilindro" ButtonCalcular.Caption = "Calcular Volumen" Label1.Caption = "Introduzca aquí el diámetro, en metros"</pre>	<pre>REM Curso Visual Basic aprenderaprogramar.com <b>Option Explicit On</b> Public Class Form1     REM Declaración de variables     Const Pi As Single = 3.141592654     Dim R, H, V As Single      REM Contenido del formulario     <b>Private Sub Form1_Load</b>(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load Me.Text = "Cálculo volumen cilindro" ButtonCalcular.Text = "Calcular Volumen" Label1.Text = "Introduzca aquí el diámetro, en metros" Label2.Text = "Introduzca aquí la altura, en metros" <b>End Sub</b></pre>

```
Label2.Caption = "Introduzca aquí la  
altura, en metros"
```

```
End Sub
```

```
Rem Cálculo y muestra resultados
```

```
Private Sub ButtonCalcular_Click()
```

```
R = Val(Text1) / 2
```

```
H = Val(Text2)
```

```
V = Pi * (R ^ 2) * H
```

```
LabelResultados.FontSize = 10
```

```
LabelResultados.FontBold = True
```

```
LabelResultados.Alignment = 2
```

```
LabelResultados.Text = "El volumen del  
cilindro es de " & V & " metros cúbicos"
```

```
End Sub
```

```
REM Cálculo y muestra resultados
```

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

```
R = Val(TextBox1.Text) / 2
```

```
H = Val(TextBox2.Text)
```

```
V = Pi * (R ^ 2) * H
```

```
LabelResultados.Font = New Font("Arial", 10,  
FontStyle.Bold)
```

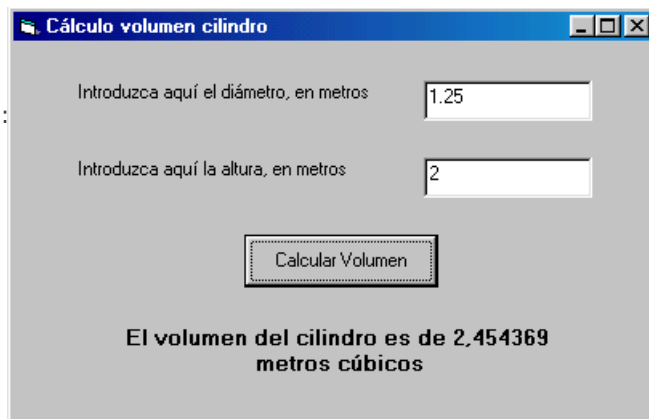
```
LabelResultados.TextAlign =  
ContentAlignment.MiddleCenter
```

```
LabelResultados.Text = "El volumen del cilindro es de "  
& V & " metros cúbicos"
```

```
End Sub
```

```
End Class
```

El resultado gráfico será similar a este:



### EJERCICIO EJEMPLO RESUELTO Nº3

Transformar en código el pseudocódigo que se indica a continuación, relativo a proporcionar el precio medio de un producto, calculado a partir del precio del mismo en tres establecimientos distintos.

#### 1. Inicio

- Mostrar "Introduzca el precio del producto en el establecimiento número 1, en euros" :  
Pedir Precio1
- Mostrar "Introduzca el precio del producto en el establecimiento número 2, en euros" :  
Pedir Precio2
- Mostrar "Introduzca el precio del producto en el establecimiento número 3, en euros" :  
Pedir Precio3
- Media = (Precio1 + Precio2 + Precio3) / 3

6. Mostrar “El precio medio del producto es”, Media, “Euros”

**7. Fin**

**SOLUCIÓN EJERCICIO EJEMPLO N°3**

En el formulario insertamos tres Labels y tres TextBox para petición de datos, así como un botón y un Label para cálculo y muestra de resultados. El código será algo así:

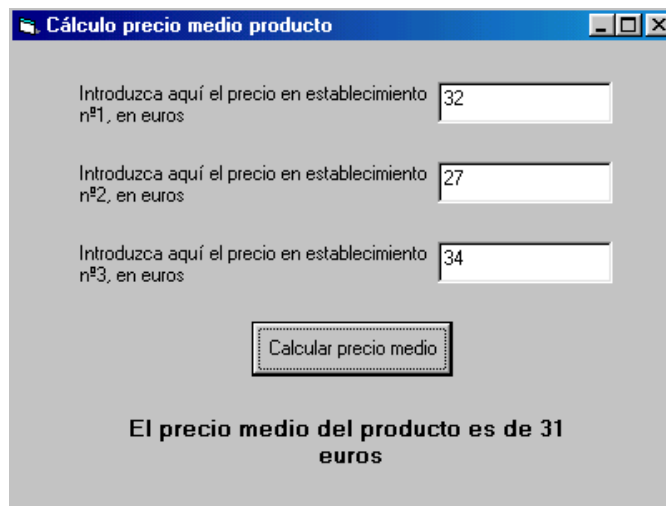
Código (versionesVB menos recientes)	Código (versiones VB más recientes)
<pre> Rem Curso Visual Basic aprenderaprogramar.com <b>Option Explicit</b> Rem Declaración de variables Dim Precio1!, Precio2!, Precio3!, Media As Single  Rem Contenido del formulario <b>Private Sub Form_Load()</b> Form1.Caption = "Cálculo precio medio producto" ButtonCalcular.Caption = "Calcular precio medio" Label1.Caption = "Introduzca aquí el precio en establecimiento n°1, en euros" Label2.Caption = "Introduzca aquí el precio en establecimiento n°2, en euros" Label3.Caption = "Introduzca aquí el precio en establecimiento n°3, en euros" <b>End Sub</b>  Rem Cálculo y muestra resultados <b>Private Sub ButtonCalcular_Click()</b> Precio1 = Val(Text1) Precio2 = Val(Text2) Precio3 = Val(Text3) Media = (Precio1 + Precio2 + Precio3) / 3 LabelResultados.FontSize = 10 LabelResultados.FontBold = True </pre>	<pre> REM Curso Visual Basic aprenderaprogramar.com <b>Option Explicit On</b> Public Class Form1     REM Declaración de variables     Dim Precio1, Precio2, Precio3, Media As Single      REM Contenido del formulario     <b>Private Sub Form1_Load</b>(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load     Me.Text = "Cálculo precio medio producto"     ButtonCalcular.Text = "Calcular precio medion"     Label1.Text = "Introduzca aquí el precio en establecimiento n°1, en euros"     Label2.Text = "Introduzca aquí el precio en establecimiento n°2, en euros"     Label3.Text = "Introduzca aquí el precio en establecimiento n°3, en euros"     <b>End Sub</b>      REM Cálculo y muestra resultados     <b>Private Sub Button1_Click</b>(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonCalcular.Click     Precio1 = Val(TextBox1.Text)     Precio2 = Val(TextBox2.Text)     Precio3 = Val(TextBox3.Text)     Media = (Precio1 + Precio2 + Precio3) / 3     LabelResultados.Font = New Font("Arial", 10, FontStyle.Bold) </pre>

```
LabelResultados.Alignment = 2
LabelResultados = "El precio medio del
producto es de " & Media & " euros"
End Sub
```

```
LabelResultados.TextAlign =
ContentAlignment.MiddleCenter
LabelResultados.Text = "El precio medio del producto es
de " & Media & " euros"
End Sub
End Class
```

Para los controles podríamos haber usado arrays de controles si hubiéramos querido. Hemos usado controles independientes tipo Label1, Label2, Label3 y Text1, Text2 y Text3 en vez de arrays de controles. Si bien en líneas generales es ventajoso el uso de arrays cuando necesitamos automatizar procesos, para aplicaciones tan sencillas como ésta no tiene trascendencia hacerlo de una u otra manera.

El resultado gráfico será similar a este:



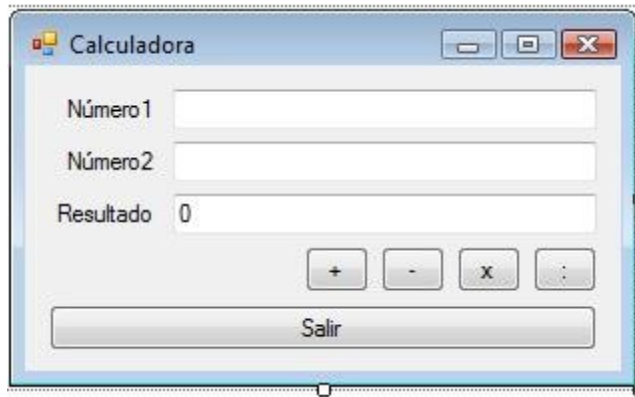
## Calculadora

Crear un formulario nuevo con los siguientes objetos (por defecto, al crear un nuevo proyecto, ya viene el Form1):

- Label (3)
- Button (5)
- TextBox (3)

**Nota:** Cambia la propiedad "(Name)" de los TextBox a "Numero1", "Numero2" y "Resultado" respectivamente.

Distribuido de esta forma:



*\*Distribución de los objetos.*

### **Código:**

Para escribir código de un botón u otro objeto, doble clic sobre el botón (o algún otro objeto).

Aparecerá algo así:

```
Private Sub [...] 'Esto lo pone el mismo programa.  
'Aquí escribes el código de lo que va a hacer ese botón o objeto.  
End Sub 'Esto también lo pone el programa.
```

*\*Los comentarios en código se ponen con una '*

En el botón "Salir":

```
Private Sub [...]  
End 'El comando End, cierra todos los formularios y el programa.  
End Sub
```

Sobre el Botón "+":

```
Private Sub [...]  
Resultado.Text = (Cdbl(Numero1.Text) + Cdbl(Numero2.Text)) 'Resultado  
= Numero1 + Numero2  
End Sub
```

Sobre el botón "-":

```
Private Sub [...]  
Resultado.Text = (Cdbl(Numero1.Text) - Cdbl(Numero2.Text)) 'Resultado  
= Numero1 - Numero2  
End Sub
```

Sobre el botón "x":

```
Private Sub [...]  
Resultado.Text = (Cdbl(Numero1.Text) * Cdbl(Numero2.Text)) 'Resultado  
= Numero1 x Numero2  
'El * indica multiplicar, aunque en el botón se haya puesto una x  
End Sub
```

Sobre el botón ":":

```
Private Sub [...]  
Resultado.Text = (Cdbl(Numero1.Text) / Cdbl(Numero2.Text)) 'Resultado  
= Numero1 : Numero2
```

```
'Aunque en el botón pongamos :, para dividir se usa una /  
End Sub
```

Opcional: Se puede poner un sexto botón que ponga AC o Nuevo calculo con el siguiente código:

```
Private Sub[...]  
Numero1.Text = "" 'El TextBox del número1 se queda en blanco.  
Numero2.Text = "" 'El TextBox del número2 se queda en blanco.  
Resultado.Text = "0" 'El TextBox del resultado se queda a "0".  
End Sub
```

Para saber cómo generar un .exe con ese código, al final del tutorial se explica.

## Convertor de unidades de temperatura

Antes de nada, como no todo el mundo sabe la unidad de temperatura del SI (Sistema internacional) que muchos creen que es el °C, voy a explicar un poco:

La temperatura mínima en °C, son -273°C lo que equivale en K a 0 (En K no hay temperatura negativa).

Conversión:

- De °C a K:  $x + 273$  (x es el número de °C)
- De K a °C:  $x - 273$  (x es el número de K)

Objetos:

- TextBox (2)
- RadioButto (2)
- GroupBox (1)
- Button (3)
- Label (3)

En la calculadora, hemos puesto nombres un poco largos, aquí no escribiremos tanto:

**(Name):** N y R (A los TextBox)



*\*Distribución de los objetos.*

**Código:**

Al botón "Calcular":

```
Private Sub [...]  
If RadioButton1.Checked Then 'Abre una condición  
R.Text = (Cdbl(N.Text + 273)) 'Si está seleccionado el RadioButton1,  
suma 273  
ElseIf RadioButton2.Checked Then  
R.Text = (Cdbl(N.Text - 273)) 'Si está seleccionado el RadioButton2,  
resta 273  
End If 'Cierra la condición  
End Sub
```

Al botón "Borrar":

```
Private Sub [...]  
N.Text = "" 'El cuadro de texto N (El del número) se borra  
R.Text = "0" 'El cuadro de texto R (El del resultado) se pone en 0  
End Sub
```

Al botón "Salir":

```
Private Sub [...]  
End 'Se cierra completamente el programa  
End Sub
```

## Instalación:

