



aprenderaprogramar.com

# Buttons (Command Buttons) en Visual Basic. Propiedades y ejemplos. Código asociado a eventos. (CU00321A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº20 del Curso Visual Basic Nivel I

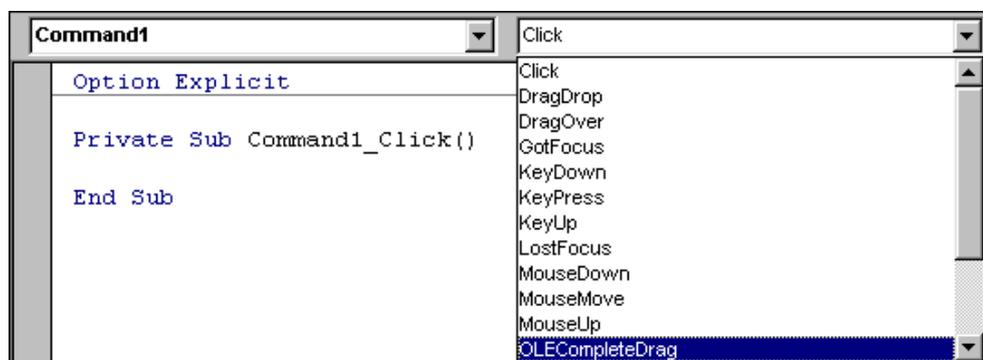
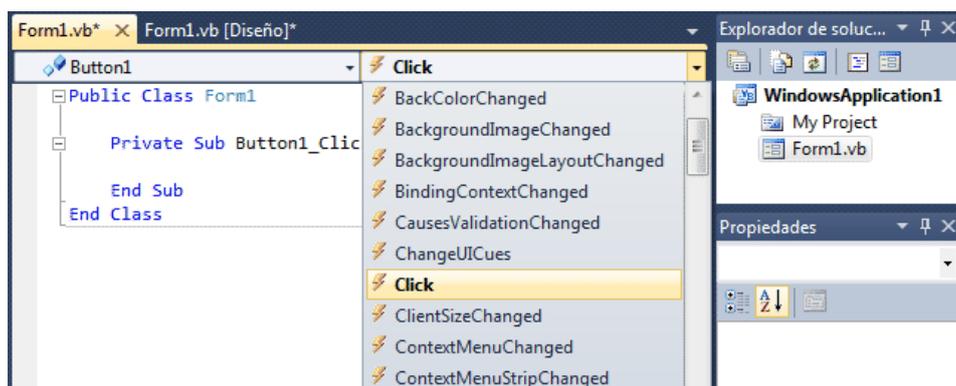
29

## CÓDIGO ASOCIADO A EVENTOS. COMMAND BUTTONS.

La programación en Visual Basic es guiada por eventos. Por tanto, cuando pedimos un dato hemos de esperar a que suceda un evento para que continúe ejecutándose el código.

Sobre un formulario, coloca un TextBox, un Label y un Button (en algunas versiones de Visual Basic se denomina Commandbutton), que es otro de los elementos de los que disponemos en la barra de herramientas, cuyo símbolo es un rectángulo y que representa “un botón”. Una vez añadidos estos tres controles al formulario vete a la ventana de código.

Observa que existen dos pestañas encima del espacio para la escritura de código. Si despliegas la pestaña de la izquierda, verás que se encuentran los distintos objetos que forman el programa y que por el momento son: el formulario o Form, la etiqueta o Label, la caja de texto o TextBox y el botón al que Visual Basic por defecto le pone nombre Button1 ó Command1 según la versión que estemos utilizando. Selecciona el Button1 (Command1). Ahora que tenemos en la pestaña izquierda seleccionado el botón, si pulsamos la pestaña derecha se muestra una lista de eventos que son detectables para el control que tenemos seleccionado. Por ejemplo, Click es el evento que se produce cuando el usuario hace click sobre el botón. MouseMove es el evento que se produce cuando el usuario mueve el puntero encima del CommandButton.



Cuando elegimos un objeto y un evento, aparece en la ventana de código el encabezado y terminación de un procedimiento de respuesta a dicho evento. Veamos un ejemplo. Agranda el objeto label sobre el formulario (establece la propiedad AutoSize a False para poder hacerlo), selecciona como control el botón en la pestaña izquierda y MouseMove como evento en la pestaña derecha y escribe lo siguiente:

*Código (versiones VB menos recientes):*

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label1.Caption = Label1.Caption & "Hola... "
End Sub
```

*Código (versiones VB más recientes):*

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On

Public Class Form1
Private Sub Button1_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs)
Handles Button1.MouseMove
Label1.Text = Label1.Text & "Hola... "
End Sub
End Class
```

Ejecuta el programa y pasa el puntero del ratón sobre el botón. Comprobarás que aparecen múltiples "Hola..." sobre el formulario (en el label). Es decir, cada vez que tiene lugar el evento, se ejecuta el código asociado.

Crea ahora un procedimiento de respuesta al click de este tipo:

*Código (versiones VB menos recientes):*

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit

Private Sub Command1_Click()
Label1.Caption = Label1.Caption & "Hola... "
End Sub
```

*Código (versiones VB más recientes):*

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On

Public Class Form1
Private Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
Label1.Text = Label1.Text & "Hola... "
End Sub
End Class
```

Comprobarás que cada vez que haces click sobre el botón se muestra el mensaje "Hola..." en el label sobre el formulario. Este tipo de procesos es vital en los programas: piensa cuando instalas un programa, cuántas veces pulsas Aceptar ó Continuar. Cada vez que lo haces, das lugar a que se ejecute un paquete de código. En cambio, si no lo haces, no tiene lugar el evento y puedes irte a tomar un café. Cuando vuelvas encontrarás que no ha sucedido nada.

Al igual que otros controles que venimos viendo, un botón tiene propiedades cuya gestión es igual a lo que hemos visto, en tiempo de diseño (trabajo gráfico sobre el formulario) o en tiempo de ejecución (a través de código). Podemos destacar las siguientes:

- **Name:** establece el nombre del botón. Recomendamos que siempre sea un prefijo común a todos los botones seguido de un término que nos permita identificar al botón. Por ejemplo, un botón para realizar un cálculo podríamos nombrarlo ButtonCalcular. Una rutina de respuesta a un click sería:

```
Private Sub ButtonCalcular_Click(...)
...
End Sub
```

- **Text ó Caption,** según la versión de Visual Basic que estemos empleando: establece el texto que se muestra al usuario sobre el botón. Por ejemplo, "Calcular".
- **AcceptButton (propiedad del formulario en versiones de Visual Basic más recientes) ó Default (propiedad del botón en versiones menos recientes):** es una propiedad interesante de los botones, ya que nos permite establecer si una pulsación de enter por parte del usuario equivale a un click a efectos de ejecución del código asociado. Es decir, supón que tienes un botón denominado ButtonCalcular y que además tienes un TextBox donde pides al usuario que introduzca un dato. Si establecemos en las propiedades del form como AcceptButton el ButtonCalcular (o establecemos Default como True para el botón en las versiones menos recientes), cuando el usuario pulse enter después de introducir el texto pedido, se ejecutará el código asociado. Prueba a establecer lo que hemos indicado con un formulario en el que hayas insertado un botón, un TextBox y un Label y comprueba que habiendo escrito un texto en el TextBox tras pulsar enter se ejecute el código previsto (que hará que se muestre "Has pulsado el botón o pulsado enter...") :

*Código (versionesVB menos recientes):*

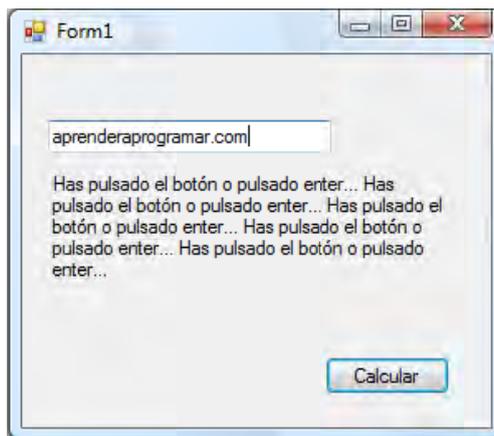
```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit

Private Sub Command1_Click()
Label1.Caption = Label1.Caption & "Has pulsado
el botón o pulsado enter... "
End Sub
```

*Código (versiones VB más recientes):*

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On

Public Class Form1
Private Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
Label1.Text = Label1.Text & "Has pulsado el
botón o pulsado enter... "
End Sub
End Class
```



Comprobarás que tiene efecto tanto hacer click sobre el Command Button como pulsar enter. Ten en cuenta que en muchas ocasiones, por ejemplo después de pedir un dato, una clave, etc. el usuario tiende a pulsar enter para dar paso a la continuación del programa. Esta propiedad te permite crear programas en los que se da relevancia a la pulsación de enter.

- **Otras propiedades:** al igual que con otros controles, existen diferentes propiedades que no hemos citado. Prueba a modificar y comprobar los efectos de algunas de ellas.

**Próxima entrega: CU00322A**

**Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com)** -- > Cursos, o en la dirección siguiente:  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=37&Itemid=61](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61)