



aprenderaprogramar.com

# **TextBox de Visual Basic. Propiedades Multiline, Scrollbars, Maxlength, TextAlign, BorderStyle, Enabled, Visible. (CU00317A)**

**Sección: Cursos**

**Categoría: Curso Visual Basic Nivel I**

**Fecha revisión: 2029**

**Autor: Mario R. Rancel**

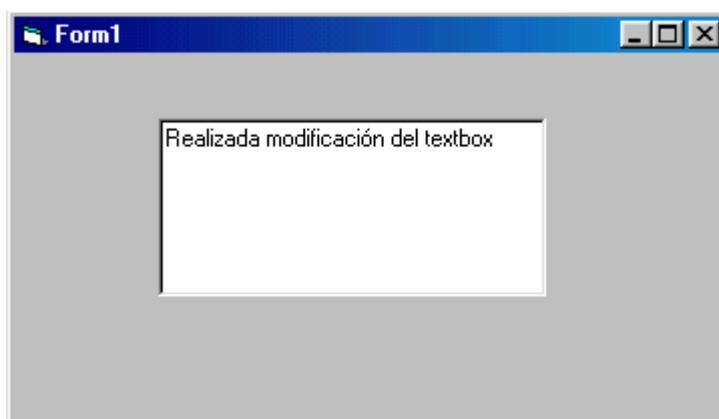
**Resumen: Entrega nº16 del Curso Visual Basic Nivel I**

29

## INSTRUCCIÓN PEDIR CON VISUAL BASIC. LOS TEXTBOX.

La forma de pedir datos con Visual Basic resultará un tanto distinta a la que se utiliza cuando se trabaja con pseudocódigo puro o lenguajes estructurados como C. Una petición de datos no necesitará ser dispuesta en el código, sino que se obtendrá a través de un objeto prefabricado: un TextBox. Vamos a aprender a utilizarlo.

Visualiza el formulario y vete en el cuadro de herramientas al botón correspondiente a TextBox, representado por un símbolo ab|. Pulsando dos veces sobre él, o pulsando una vez y después insertándolo en el formulario, tendrás un TextBox cuyo nombre por defecto será Text1, TextBox1 ó similar. Agrándalo un poco y sin haber escrito código para el programa, da orden de ejecutar. El resultado es que te aparece el formulario con el TextBox sobre él. Puede que te aparezca vacío o que muestre un texto por defecto como Text1 dentro del TextBox. En la vista del diseño del formulario podemos escribir el mensaje que queremos dentro del TextBox. Borra el contenido de texto en el TextBox (si es que había alguno) y escribe por ejemplo "Realizada modificación del textbox" y pulsa enter. Quizás no hayas podido agrandar el TextBox en sentido vertical, ahora veremos cómo se puede hacer.



Un TextBox es un control que principalmente utilizamos para que el usuario introduzca datos, pero también puede mostrar una respuesta predeterminada o incluso servir para mostrar en vez de para pedir datos.

Un TextBox tiene una serie de propiedades, como puedes ver pulsando sobre el Textbox y recorriendo la lista que te aparece en la ventana Propiedades (si no tienes activada esta ventana, actívala). Prueba por ejemplo a cambiar el color de fondo (BackColor). El mensaje inicial que aparecerá sobre el TextBox cuando comience el programa también es una propiedad: la propiedad Text. Prueba a cambiar esta propiedad (puedes tanto escribir un texto como dejar vacío el TextBox). Comprobarás que si escribes un contenido y pulsas enter, cuando ejecutas dicho texto aparece dentro del TextBox sobre el formulario. Te bastará con establecer Text vacío para que no haya ningún mensaje inicial cuando ejecutes el programa.

Muchas propiedades pueden establecerse en tiempo de diseño (cuando diseñamos el formulario) o en tiempo de ejecución (contenidos del código que se realizan cuando ejecutamos el programa). Escribe como texto para el TextBox en el diseño del formulario el texto "Text1". Escribe el siguiente código:

**Código (versiones VB menos recientes):**

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

Private Sub Form_Load()
Text1.Text = ""
Text1.BackColor = vbGreen
End Sub
```

**Código (versiones VB más recientes):**

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        TextBox1.Text = ""
        TextBox1.BackColor = Color.Green
    End Sub
End Class
```

El resultado es que el TextBox se muestra vacío y con color de fondo verde. El mostrarse vacío se debe a que hemos indicado que cuando se cargue el formulario la propiedad Text del control (un elemento como un TextBox se denomina "control") se establezca a cadena vacía (""). En versiones menos recientes de Visual Basic incluso se permitía escribir simplemente `Text1 = ""`, ya que se consideraba que si no se especificaba propiedad era para referirnos a la propiedad Text. En las versiones más recientes es obligatorio indicar siempre la propiedad a modificar.

Hemos cambiado la propiedad de color indicándole que cambie a `vbGreen` ó `Color.Green`. `VbGreen` ó `Color.Green` se corresponden con constantes predefinidas por Visual Basic que tenemos disponibles para ser usadas cuando estemos escribiendo código.



*Visual Basic* define constantes para simplificar la programación. Puedes consultar "Constantes de *Visual Basic*" para conocer todas las constantes definidas, que se agrupan en constantes de color, Calendar, del compilador, de fecha, Dir, GetAttr y SetAttr, IMEStatus, Instr, StrComp, de código de teclas, MsgBox, QueryClose, Shell, StrConv, de color del sistema, VarType y otras.

Sólo citaremos algunas constantes que juzgamos interesantes. Una de ellas es `vbCrLf`, una constante que supone una indicación de salto de línea y retorno de carro. Cuando se introduce esta constante, equivale a una pulsación de enter. Su equivalente usando código ASCII es `Chr(13) + Chr(10)`. Es decir, podemos escribir tanto una cosa como otra, resultando equivalentes. El interés de las constantes es que suelen resultar más sencillas de usar que su escritura con otro tipo de código. Prueba estos códigos y comprobarás que hacen lo mismo:

a) Caso 1

```
MsgBox ("22 salones" & vbCrLf & "4321 camarotes")
```

b) Caso 2

```
MsgBox ("22 salones" & Chr(13) & Chr(10) & "4321 camarotes")
```

En la ventana Propiedades puedes modificar la denominación del control (Text1 ó TextBox1) por otra denominación. Supón que en ese cuadro de texto se espera que el usuario introduzca una altura. Podemos establecer la propiedad Name como Textaltura. Así, podríamos escribir por ejemplo Textaltura.BackColor = vbGreen ó Textaltura.BackColor = Color.Green. También podríamos llamar al TextBox simplemente altura y escribir altura.BackColor = ... , aunque este tipo de denominaciones son poco recomendables porque son menos claras. Conviene siempre respetar al menos un texto inicial que nos permita saber que esa denominación corresponde a un TextBox, ya que en caso contrario puede ser muy difícil de leer un programa. Por ejemplo, si vamos a pedir 4 datos que son nombre de edificio, longitud, anchura y altura podemos nombrar los TextBox como TextNomEdificio, TextLongitud, TextAnchura y TextAltura. Un mismo prefijo nos identifica todos los controles que son del tipo TextBox. Las normas para nombrar objetos deben seguir razonamientos similares a lo que hemos descrito para variables, tendentes a ser lo más claros y explicativos posibles.

Para terminar esta visión rápida de lo que son los TextBox, citaremos algunas propiedades que nos parecen interesantes:

- **TextAlign ó Alignment:** permite que el texto se alinee a la izquierda, a la derecha, o que aparezca centrado.
- **BorderStyle ó Appearance:** cambia la apariencia que puede ser tridimensional o plana.
- **Enabled:** si está establecido a True permite que se pueda escribir sobre el TextBox. Si está establecida a False, impide que el usuario pueda modificar el contenido del TextBox.
- **Maxlength:** limita el número máximo de caracteres que puede admitir un TextBox. Si escribimos el valor 0, esto indica la ausencia de un máximo distinto del que imponga la memoria del sistema del usuario (lo cual será un valor muy grande, prácticamente equivalente a decir “se admite cualquier longitud de texto”). Un valor mayor que 0 indica el número máximo de caracteres. Por ejemplo si establecemos el valor en 5, el usuario podrá escribir “perro” porque tiene 5 letras, pero no podrá escribir “serpiente” por tener más de 5 letras, sólo podría escribir << serpi >>
- **Multiline:** establecido a False supone que el TextBox consta de una línea única. Establecido a True permite que el TextBox conste de múltiples líneas. En algunas versiones es necesario establecer Multiline como true si queremos agrandar verticalmente el formulario.
- **Scrollbars:** permite añadir barras de desplazamiento verticales, horizontales o ambas al TextBox.
- **Visible:** determina si el TextBox resultará visible para el usuario o no.

Trata de modificar los valores de estas propiedades y observa los resultados. Haz lo mismo actuando en el código en vez de en el diseño del formulario. Podrás comprobar que existen muchas más propiedades además de las que hemos citado. Prueba a modificar algunas de ellas para comprender qué es lo que controlan.

**Próxima entrega: CU00318A**

**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)