



aprenderaprogramar.com

Labels en Visual Basic. Autosize. BorderStyle. Font. Otras formas de mostrar mensajes: MsgBox, Print. (CU00319A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº18 del Curso Visual Basic Nivel I

29

MOSTRAR CON VISUAL BASIC MEDIANTE LABELS

Ya sabemos que un Label sirve para insertar mensajes en un formulario que el usuario no podrá modificar. También hemos visto cómo insertarlos en el formulario. En cuanto a propiedades de los Labels, contenido, etc. su manejo es básicamente igual que el de los TextBox, con lo que prácticamente todo lo explicado en relación a los TextBox es también válido para los Labels. Por ejemplo, podrás escribir `Label1.BackColor = Color.Green` (ó `Label1.BackColor = vbGreen` según la versión de Visual Basic que estés utilizando) de forma análoga a como lo hicimos con TextBox.

El contenido que muestra el control se establece a través de la propiedad Text (en las versiones menos recientes propiedad Caption). Podemos escribir `Label1.Text = "Introduzca aquí un entero positivo"` (ó `Label1.Caption = "Introduzca aquí un entero positivo"` en otras versiones) para obtener ese mensaje dentro del Label1.

Aparte de las propiedades que ya conocemos podemos destacar:

- **Autosize:** si se establece como True, da lugar a que el tamaño del Label se ajuste al texto que muestra. Si quieres modificar el tamaño de un label en tiempo de diseño y no puedes posiblemente se deba a que Autosize está establecido como True.
- **BorderStyle:** permite cambiar la apariencia del borde del Label.
- **Font:** permite establecer tipo de letra, su tamaño, propiedades (cursiva, negrita, etc.)

Prueba a crear un Label y establecer diferentes mensajes y propiedades para el mismo.

Para mostrar un mensaje compuesto de distintas partes en un *Label* hemos de usar el símbolo de concatenación. Por ejemplo:

```
Label1.Text = msg1 & msg2   ó   Label1.Caption = msg1 & msg2 (según versión de VB en uso)
```

daría lugar a que se mostrara el contenido derivado de las variables msg1 y msg2. En las versiones menos recientes de Visual Basic se consideraba que si se escribiría `Label = ...` nos referíamos al texto que debía mostrarse dentro del Label. Es decir, se permitía escribir:

```
Label1 = msg1 & msg2
```

Obteniendo el mismo resultado que en el caso anterior. En las versiones más recientes esto ya no se permite.

Vamos a trasladar el siguiente pseudocódigo a código usando Labels y TextBox:

```
Mostrar "Introduzca su peso en kgs por favor": Pedir Peso
```

a) A través del diseño del formulario.

Insertamos un Label en el formulario. Establecemos la propiedad Text (en otras versiones Caption) del Label como "Introduzca su peso en kgs por favor". Elegimos la propiedad Font como tamaño 10 y negrita. Creamos un TextBox llamado Text1 ó TextBox1, situado junto al mensaje.

- b) A través del código (Recuerda que si escribes código donde citas un objeto como un TextBox, etc. dicho objeto debes haberlo creado previamente sobre el formulario en ver – objeto, y tienes que invocarlo respetando su nombre. Igualmente debes prestar atención al tamaño de los objetos para que los textos no se te corten, a la propiedad Visible, etc.)

Código (versiones VB menos recientes):

```
Rem Curso Visual Basic
aprenderaprogramar.com
Option Explicit
Private Sub Form_Load()
Show
Label1.Caption = "Introduzca su peso en kgs
por favor"
Label1.Font.Size = 10
Label1.Font.Bold = True
End Sub
```

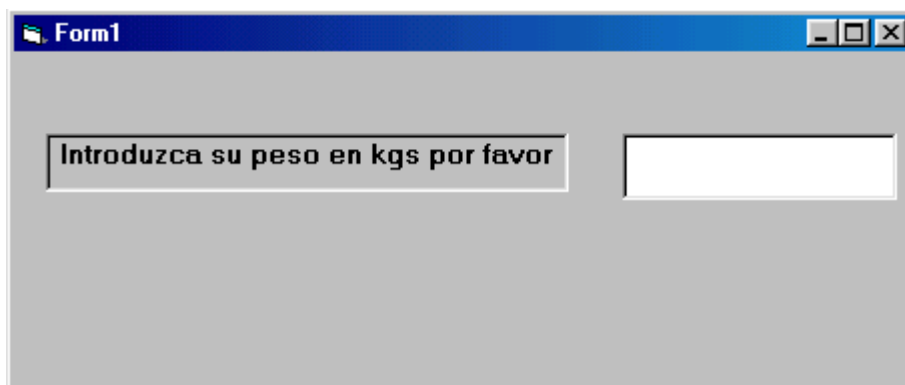
Código (versiones VB más recientes):


```
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
Label1.Text = "Introduzca su peso en kgs por favor"
Me.Label1.Font = New Font("Arial", 10, FontStyle.Bold)
    End Sub

End Class
```

El resultado esperado es algo así:



 Otra forma relevante de mostrar datos (también de pedirlos) es a través de ventanas de mensajes (MsgBox). Cuando se invoca la función MsgBox se despliega una ventana en la que se nos muestra un mensaje y una serie de botones a elegir entre Sí, No, Aceptar, Cancelar, Anular, Omitir, Reintentar...etc. La sintaxis es del tipo MsgBox("Valor no válido") ó Respuesta = MsgBox("¿Desea continuar?", vbYesNo), donde Respuesta es una variable tipo Integer donde se almacena un número asociado a la respuesta que ha dado el usuario. Por ejemplo, Respuesta valdrá 1 si el usuario ha elegido aceptar, valdrá 2 si ha elegido cancelar, valdrá 3 si ha elegido anular, etc.

Valiéndote de la Ayuda prueba a crear distintos MsgBox.

MOSTRAR CON VISUAL BASIC MEDIANTE PRINT

Hay distintos usos posibles de Print. En las versiones menos recientes existía una instrucción Print que permitía mostrar mensajes sobre un formulario y que es la que vamos a explicar a continuación. Si estás usando una versión reciente de Visual Basic esta instrucción no está disponible, por lo que puedes saltarte este apartado y pasar al siguiente. El uso de Print para mostrar mensajes sobre un formulario se basa en la escritura de texto entrecomillado y variables delimitadas por punto y coma. Ejemplo de este uso sería:

```
Print "La respuesta es"; A; "euros"
```

que daría lugar a un resultado del tipo: La respuesta es 32 euros. Utilizar una coma en lugar de un punto y coma es posible, pero la coma se traduce en la introducción de una tabulación. Así:

```
Print "La respuesta es", A, "euros"
```

da lugar a un resultado del tipo: La respuesta es 32 euros. También es posible usar el símbolo de concatenación &:

```
Print "La respuesta es " & A & " euros"
```

La diferencia respecto a usar puntos y comas estriba en que el punto y coma lleva a una introducción automática de un espacio antes y después de su ubicación, cosa que no ocurre con el símbolo &.

Podemos usar el símbolo de prolongación de línea para que dos líneas aparentes tengan consideración de una única línea:

```
Print "La respuesta es"; A; _  
"euros"
```

Como hemos visto, se puede introducir una línea con un Print sin mensaje para dejar una línea vacía, o usar Tab para localizar el contenido a mostrar en un punto, por ejemplo:

```
Print Tab(18); A
```

La ventaja de Print es que es muy rápido de usar para visualizar contenidos y su inconveniente que a efectos estéticos es poco interesante, además de que ya no está disponible en las versiones recientes de Visual Basic. En general siempre será preferible usar otro tipo de formas de presentar datos como Labels o MsgBox.

Próxima entrega: CU00320A

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