



aprenderaprogramar.com

Visual Basic: procedimientos Sub y funciones Function. Tipos de parámetros y tipos de retorno. Ejemplos (CU00340A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº39 del Curso Visual Basic Nivel I

29

MÓDULOS GENÉRICOS Y PARÁMETROS DE ENTRADA EN VISUAL BASIC.

Ya hemos visto que en Visual Basic una declaración de procedimiento siempre lleva unos paréntesis donde se pueden indicar parámetros de entrada. Los parámetros de entrada son indicados por el programador, o bien se generan automáticamente al tener lugar un evento y el propio ordenador envía ese parámetro de entrada al procedimiento que se dispara como consecuencia de que tenga lugar el evento. Veamos algunos ejemplos de procedimientos asociados a eventos.

Código 1 (versiones menos recientes VB)

```
Private Sub Text1_Change()
    .
    .
    .
End Sub
```

Código 1 (versiones más recientes VB)

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    .
    .
    .
End Sub
```

Código 2 (versiones menos recientes VB)

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    .
    .
    .
End Sub
```

Código 2 (versiones más recientes VB)

```
Private Sub Text1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    .
    .
    .
End Sub
```

El primer código corresponde a un procedimiento conducido por eventos que se invoca cuando el texto contenido en el TextBox Text1 cambia. En las versiones menos recientes de Visual Basic algunos procedimientos conducidos por eventos no tienen parámetros de entrada. En las versiones más recientes siempre suelen aparecer unos parámetros de entrada.

El segundo código sería el de un procedimiento conducido por eventos que se invoca cuando se produce la pulsación de una tecla en el TextBox1. En este caso, en las versiones menos recientes hay un parámetro (podría haber varios) que es enviado al procedimiento cuando tiene lugar el evento: un valor tipo Integer que se corresponde con el número asociado a la tecla pulsada según el código ASCII. En las versiones más recientes esa información viene dentro del parámetro “e” que lleva distinta información asociada relacionada con el evento. Crea un TextBox, un Label y prueba el siguiente código:

Código versiones menos recientes VB:

```
'Curso VB aprenderaprogramar.com
Private Sub Text1_KeyPress(KeyAscii As Integer)
Label 1 = ""
Label 1 = KeyAscii
End Sub
```

Código versiones más recientes VB:

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
Private Sub TextBox1_KeyPress(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyPressEventArgs)
Handles TextBox1.KeyPress
Label1.Text = ""
Label1.Text = "Valor numérico asociado a la tecla
pulsada es " & Microsoft.VisualBasic.Asc(e.KeyChar)
End Sub
End Class
```

Comprobarás que cada vez que pulsas una tecla sobre el TextBox aparece en pantalla un número: para la A, el 65, y para la a, el 97. Ese número es información que se pasa automáticamente al procedimiento cuando tiene lugar el evento. De esta forma se obtiene información útil sobre lo que está pasando (qué tecla ha pulsado el usuario, por dónde ha movido el ratón, qué botón del ratón ha pulsado, etc.) y se puede establecer un código previsto a ejecutar ante determinadas circunstancias.

Una declaración de procedimiento genérico que espera un parámetro de entrada podría ser esta:

Código versiones menos recientes VB:

```
Private Sub Raíz(Número As Single)
.
.
.
End Sub
```

Código versiones más recientes VB:

```
Private Sub Raíz(ByVal Número As Single)
.
.
.
End Sub
```

Dentro de los paréntesis se indican los parámetros de entrada requeridos, y el tipo de dato que han de ser (de momento no prestaremos atención a cuestiones adicionales como la aparición de la palabra ByVal). Si existen varios parámetros se escriben separados por comas, por ejemplo: (Número As Single, Valor As Double, Nombre As String). Si no se especifica tipo de dato para los parámetros serán del tipo por defecto (Variant en versiones menos recientes de Visual Basic u Object en las versiones más recientes). Nosotros siempre declararemos el tipo de los parámetros porque es una buena práctica de programación y no hacerlo podría dar lugar a diferentes problemas. La llamada al procedimiento podría hacerse con una de estas sintaxis:

Call Raíz(dato)

Raíz dato

Raíz (dato)

La diferencia entre usar una u otra forma de llamada la veremos en breve.

Una declaración de función genérica que espera un parámetro de entrada podría ser esta:

Código versiones menos recientes VB:

Código versiones más recientes VB:

```
Private Function Raíz(Número As Single) As Single
    .
    .
    .
End Sub
```

```
Private Function Raíz(ByVal Número As Single) As Single
    .
    .
    .
End Function
```

Las llamadas a funciones se realizan escribiendo el nombre de la función seguido de los parámetros entre paréntesis. Dado que devuelven un resultado, se pueden usar para asignar contenido a una variable, para mostrar algo en pantalla, etc. Por ejemplo:

```
Resultado = Raíz(dato)
Label1 = Raíz(dato) ó Label1.Text = Raíz(dato)
```

Ejemplos de llamadas a un módulo genérico podrían ser estas:

Código versiones menos recientes VB:

Código versiones más recientes VB:

```
'Curso VB aprenderaprogramar.com

Option Explicit
Dim Dato As Single
Dim Resultado As Single

Private Sub Form_Load()
Call Raíz(9)
Call Raíz(-144)
Dato = InputBox("Introduzca número para
calcular raíz", "Dato")
Call Raíz(Dato)
End Sub

Private Sub Raíz(Número As Single)
Número = Abs(Número)
Resultado = Sqr(Número)
Label1 = Label1 & vbCrLf & "Raíz
calculada: " & Resultado
End Sub
```

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1

    Dim Dato As Single
    Dim Resultado As Single

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
Label1.Text = ""
Call Raíz(9)
Call Raíz(-144)
Dato = InputBox("Introduzca número para
calcular raíz", "Dato")
Call Raíz(Dato)
End Sub

    Private Sub Raíz(Número As Single)
Número = Math.Abs(Número)
Resultado = Math.Sqrt(Número)
Label1.Text = Label1.Text & vbCrLf & "Raíz
calculada: " & Resultado
End Sub
End Class
```

Si introducimos como número para calcular raíz 64 el programa nos mostrará: “Raíz calculada: 3; Raíz calculada: 12; Raíz calculada: 8”, donde 3 es la raíz de 9, 12 es la raíz de 144 (valor absoluto de -144), y 8 es la raíz de 64 (número que hemos introducido por teclado).

Si tratáramos de realizar una llamada del tipo Raíz("Nueva York") nos aparecería un mensaje de error como: "Error 13. No coinciden los tipos." ó "La conversión de la cadena "Nueva york" en el tipo 'Single' no es válida."

El mismo "objetivo" cumplido utilizando funciones lo exponemos a continuación. Recuerda que una función, además de ejecutar un código, devuelve un valor.

Código versiones menos recientes VB:

```
'Curso VB aprenderaprogramar.com
'Programa con function

Option Explicit
Dim Dato As Single
Dim Resultado As Single

Private Sub Form_Load()
Label1 = ""
Mostrar(Raíz(9))
Mostrar(Raíz(-144))
Dato = InputBox("Introduzca número
para calcular raíz", "Dato")
Mostrar(Raíz(Dato))
End Sub

Private Function Raíz(Número As Single)
As Single
Número = Abs(Número)
Raíz = Sqr(Número)
End Function

Private Sub Mostrar(valor As Single)
Label1 = Label1 & vbCrLf & "Raíz
calculada: " & valor
End Sub
```

Código versiones más recientes VB

```
REM Curso Visual Basic aprenderaprogramar.com
'Programa con function
Option Explicit On
Public Class Form1

    Dim Dato As Single
    Dim Resultado As Single

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
Label1.Text = ""
Mostrar(Raíz(9))
Mostrar(Raíz(-144))
Dato = InputBox("Introduzca número para
calcular raíz", "Dato")
Mostrar(Raíz(Dato))
End Sub

    Private Function Raíz(ByVal Número As Single) As
Single
Número = Math.Abs(Número)
Raíz = Math.Sqrt(Número)
Return Raíz
End Function

    Private Sub Mostrar(ByVal valor As Single)
Label1.Text = Label1.Text & vbCrLf & "Raíz
calculada: " & valor
End Sub
End Class
```

Próxima entrega: CU00341A

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