



aprenderaprogramar.com

# Visual Basic: paso de parámetros por referencia ByRef y por valor ByVal. Procedimientos y funciones. (CU00341A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº40 del Curso Visual Basic Nivel I

29

## PASO DE ARGUMENTOS POR REFERENCIA Y POR VALOR EN VISUAL BASIC.

Con Visual Basic se utilizan los términos "Paso de argumentos por Referencia (ByRef)", equivalente a lo que en el curso "Bases de la programación nivel II" de [aprenderaprogramar.com](http://aprenderaprogramar.com) hemos denominado transferencia de variable, y "Paso de argumentos por Valor (ByVal)", equivalente a lo que hemos denominado transferencia por valor.

Si no tienes claro el concepto de paso por referencia y paso por valor consulta el curso "Bases de la programación nivel II" de [aprenderaprogramar.com](http://aprenderaprogramar.com).

El comportamiento de Visual Basic depende de la versión que estemos utilizando:

- a) Con versiones menos recientes de Visual Basic: se permite no especificar cómo se pasa un parámetro. Por ejemplo Private Function suma (a As Integer, b As Integer). En caso de no estar especificado, la opción de defecto es que los parámetros pasan por referencia, con lo cual es posible que se modifique el valor de la variable inicial que se pasa como parámetro. En el último programa pasábamos la variable Dato como parámetro de la función Raíz. Al tener lugar la transformación del parámetro en valor absoluto, si introducíamos un número negativo su valor quedaba transformado. ¿Cómo hacer que la transferencia sea por valor? Bastará con hacer la declaración de función o de procedimiento de la siguiente manera:

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

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

Es decir, indicamos que el parámetro Número se procesa por valor.

- b) Con versiones más recientes de Visual Basic: no se permite dejar sin especificar cómo se pasa un parámetro. Por ejemplo no se admite Private Function suma (a As Integer, b As Integer), de hecho si escribimos esto el editor lo transformará automáticamente en Private Function suma (ByVal a As Integer, ByVal b As Integer) ya que se asume que los parámetros pasarán por defecto por valor.

Para no tener dudas, lo más sencillo será especificar siempre cómo se pasa un parámetro a un procedimiento o una función. En caso de que no sepamos cómo pasarlo, recomendamos usar siempre ByVal para evitar que se generen modificaciones indeseadas dentro del procedimiento o función.

En la hipótesis de que existieran distintos parámetros y uno se quiera procesar de una manera y otro de otra, lo indicaremos antecediendo la palabra clave ByRef o ByVal delante del nombre del parámetro como en el siguiente ejemplo:

```
Private Function Raíz(ByVal Número As Single, ByRef x As Integer)
```

En las versiones menos recientes de Visual Basic, mientras que el modo de transferencia en las funciones sólo puede indicarse en la definición de la función, en los procedimientos podemos indicarlo bien en la declaración del procedimiento bien en la forma de llamada que empleemos. En versiones menos recientes de VB, el tipo de llamada condiciona cómo se transfiere el argumento. Así tenemos que:

- a) Implican transferencia de variable llamadas del tipo: Call Raíz(dato) y Raíz dato
- b) Implican transferencia de valor llamadas del tipo: Raíz(dato)

Se observa que es una diferencia mínima de escritura la que diferencia uno y otro tipo de llamada. Sin embargo, las consecuencias de usar uno u otro tipo pueden ser relevantes, y conocer y comprender estas implicaciones nos puede evitar más de un dolor de cabeza. Visual Basic no admite sintaxis del tipo Llamar Raíz(Dato) Por Valor.

Sin embargo, cuando en la declaración del procedimiento o función hemos indicado cómo se pasan los parámetros, resulta indistinto usar uno u otro tipo de sintaxis de llamada. Por tanto, en las versiones más recientes de Visual Basic, donde el modo de paso de parámetros es siempre indicado, no tiene relevancia cómo se haga la llamada.

## EJERCICIO

Transformar en código el pseudocódigo que se muestra a continuación correspondiente a un programa denominado SUC02, cuyo objetivo es el cálculo del valor de un sumatorio del tipo:

$$\sqrt{a}, \sqrt{a-1}, \sqrt{a-2}, \dots, \sqrt{0}$$

### [PROGRAMA SUC02 Curso Visual Basic aprenderaprogramar.com]

#### Variables:

Enteras: E  
Reales: Dato, Raiz01, Raiz02, Suce

#### 1. Inicio

##### 2. Mientras E <> 2 Hacer

2.1 Mostrar "Elija 1. Cálculo 2. Salir"

2.2 Pedir E

##### 2.3 Si E = 1 Entonces

Llamar EntraDatos

Llamar Proceso(Dato) PorValor

Llamar Resultados

FinSi

Repetir

#### 3. Fin

**Módulo EntraDatos**

1. **Hacer**

Mostrar "Por favor, introduzca número entero entre 0 y 100"

Pedir Dato

Dato = Redondear(Dato)

**Repetir Mientras Dato < 0 ó Dato > 100**

**FinMódulo**

**Módulo Proceso(Num: Enteros)**

1. Raiz01 = SQR(Num)

2. Raiz02 = - Raiz01

3. Suces = 0

4. **Mientras Num >= 0 Hacer**

Suces = Suces + SQR(Num)

Num = Num - 1

**Repetir**

**FinMódulo**

**Módulo Resultados**

Mostrar "Dato base =", Dato

Mostrar "Raiz01=", Raiz01

Mostrar "Raiz02=", Raiz02

Mostrar "Valor de la suma de los términos de la sucesión =", Suces

**FinMódulo**

La solución se muestra en la siguiente página.

## SOLUCIÓN

### Código versiones menos recientes VB:

```
'Curso VB aprenderaprogramar.com
'Programa SUC02

Option Explicit
Dim Dato As Single
Dim Raíz01!, Raíz02 As Single
Dim Suce As Single

Private Sub Form_Load()
Form1.Caption = "Programa SUC02"
CommandCálculo.Caption = "Cálculo"
CommandSalir.Caption = "Salir"
End Sub

Private Sub CommandSalir_Click()
Unload Form1
End
End Sub

Private Sub CommandCálculo_Click()
Call EntraDatos
Call Proceso(Dato)
Call Resultados
End Sub

Private Sub EntraDatos()
Do
Dato = CInt(InputBox("Por favor, introduzca
nº entero entre 0 y 100", "Dato"))
Loop While Dato < 0 Or Dato > 100
End Sub

Private Sub Proceso(ByVal Num As
Integer)
Raíz01 = Sqr(Num)
Raíz02 = -Raíz01
Suce = 0
Do While Num >= 0
Suce = Suce + Sqr(Num)
Num = Num - 1
Loop
End Sub

Private Sub Resultados()
Label1.Alignment = 2
Label1.FontBold = True
Label1 = "Dato base = " & Dato & vbCrLf &
vbCrLf & " Raíz01 = " & Raíz01 & " Raíz02 = " &
Raíz02 & vbCrLf & vbCrLf & "Valor sucesión = "
& Suce
End Sub
```

### Código versiones más recientes VB:

```
REM Curso Visual Basic aprenderaprogramar.com
'Programa SUC02

Option Explicit On
Public Class Form1
Dim Dato As Single
Dim Raíz01 As Single, Raíz02 As Single
Dim Suce As Single

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
Me.Text = "Programa SUC02"
ButtonCálculo.Text = "Cálculo"
ButtonSalir.Text = "Salir"
Label1.Text = ""
End Sub

Private Sub ButtonSalir_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonSalir.Click
Me.Close()
End
End Sub

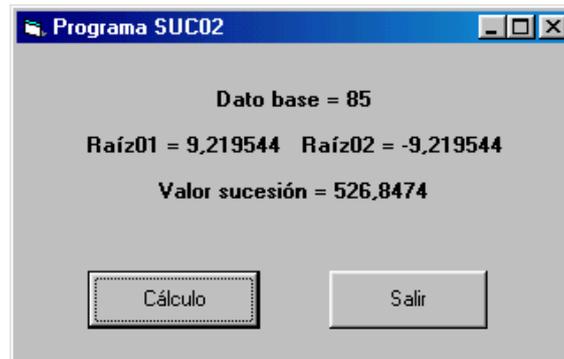
Private Sub ButtonCálculo_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonCálculo.Click
Call EntraDatos()
Call Proceso(Dato)
Call Resultados()
End Sub

Private Sub EntraDatos()
Do
Dato = CInt(InputBox("Por favor,
introduzca nº entero entre 0 y 100", "Dato"))
Loop While Dato < 0 Or Dato > 100
End Sub

Private Sub Proceso(ByVal Num As Integer)
Raíz01 = Math.Sqrt(Num)
Raíz02 = -Raíz01
Suce = 0
Do While Num >= 0
Suce = Suce + Math.Sqrt(Num)
Num = Num - 1
Loop
End Sub

Private Sub Resultados()
Label1.TextAlign = ContentAlignment.MiddleCenter
Label1.Font = New Font("Arial", 10, FontStyle.Bold)
Label1.Text = "Dato base = " & Dato & vbCrLf &
vbCrLf & " Raíz01 = " & Raíz01 & " Raíz02 = " &
Raíz02 & vbCrLf & vbCrLf & "Valor sucesión = " &
Suce
End Sub
End Class
```

El aspecto gráfico sería el siguiente:



En el curso “Bases de la programación nivel II” de [aprenderaprogramar.com](http://www.aprenderaprogramar.com) creamos una variante de pseudocódigo para este ejercicio consistente en el bloqueo del producto de los módulos. A modo de ejercicio adicional, te proponemos que crees esta variante en Visual Basic y compruebes las implicaciones que tiene.

**Próxima entrega: CU00342A**

**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)