



aprenderaprogramar.com

Operadores lógicos y matemáticos en Visual Basic. Math. Comentarios en VB. Rem. Ejemplos. (CU00314A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº13 del Curso Visual Basic Nivel I

29

INSERCIÓN DE COMENTARIOS EN EL CÓDIGO

Existen distintas posibilidades para insertar comentarios en el código de Visual Basic como son:

La palabra clave Rem: la presencia de la palabra clave Rem (REM en mayúsculas algunas versiones) transforma en comentario todo lo que se encuentre a continuación de ella en una línea. Rem puede encontrarse al principio de una línea, con lo cual nada de lo que aparezca en dicha línea será ejecutado, o bien después de una o varias instrucciones que se encuentren en una línea. En algunas versiones no es necesario separador, mientras que en otras Rem debe estar separado de las instrucciones anteriores en la línea mediante dos puntos.

El signo apóstrofe ': la presencia del signo apóstrofe convierte en comentario todo lo que se encuentre detrás de él en una línea dada. Puede situarse al principio de una línea, con lo cual nada de lo que aparezca en dicha línea será ejecutado, o bien después de una o varias instrucciones sin necesidad de que existan dos puntos ni ningún otro signo de separación.

La parte del código que son comentarios es marcada por el editor de Visual Basic con un color verde que lo diferencia de las instrucciones que van a ser ejecutadas. En un mismo programa pueden aparecer indistintamente instrucciones Rem y apóstrofes. Veamos ejemplos equivalentes:

Ejemplo 1 (versiones menos recientes):

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim seleccionado As Boolean

Private Sub Form_Load()
Rem A ejecutar cuando se carga el formulario
seleccionado = True
Rem Mostramos por pantalla mensaje
MsgBox ("El valor actual de seleccionado es " &
seleccionado)
End Sub
```

Ejemplo 1 (versiones más recientes):

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim seleccionado As Boolean

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
REM A ejecutar cuando se carga el formulario
seleccionado = True
REM Mostramos por pantalla mensaje
MsgBox ("El valor actual de seleccionado
es " & seleccionado)
    End Sub
End Class
```

Ejemplo 2 (versiones menos recientes):

```
'Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim seleccionado As Boolean

Private Sub Form_Load()
'A ejecutar cuando se carga el formulario
seleccionado = True
'Mostramos por pantalla mensaje
MsgBox ("El valor actual de seleccionado es " &
seleccionado)
End Sub
```

Ejemplo 2 (versiones más recientes):

```
'Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim seleccionado As Boolean

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
' A ejecutar cuando se carga el formulario
seleccionado = True
' Mostramos por pantalla mensaje
MsgBox ("El valor actual de seleccionado
es " & seleccionado)
    End Sub
End Class
```

Ejemplo 3 (versiones menos recientes):

```

Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim seleccionado As Boolean

Private Sub Form_Load()
seleccionado = True 'Ejecuta al cargar formulario
MsgBox ("El valor actual de seleccionado es " &
seleccionado) ' Mostramos por pantalla mensaje
End Sub

```

Ejemplo 3 (versiones más recientes):

```

REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim seleccionado As Boolean

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
seleccionado = True 'Ejecuta al cargar formulario
MsgBox("El valor actual de seleccionado es " &
seleccionado) 'Mostramos por pantalla mensaje
    End Sub
End Class

```

Ejemplo 4 (versiones menos recientes):

```

Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim seleccionado As Boolean

Private Sub Form_Load()
seleccionado = True 'Ejecuta al cargar formulario
MsgBox ("El valor actual de seleccionado es " &
seleccionado) : Rem Mostrar por pantalla mensaje
End Sub

```

Ejemplo 4 (versiones más recientes):

```

REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim seleccionado As Boolean

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
seleccionado = True REM Ejecuta al cargar formulario
MsgBox("El valor actual de seleccionado es " &
seleccionado) 'Mostramos por pantalla mensaje
    End Sub
End Class

```

El ejemplo 1 corresponde a un programa con comentarios insertados a través de Rem situados previos a la línea que comentan. El ejemplo 2 es igual con los Rem sustituidos por apóstrofes. En el ejemplo 3 los comentarios con apóstrofes se sitúan a continuación de las instrucciones, en la misma línea. Por último, el ejemplo 4 combina el uso de apóstrofes con Rem en la misma línea de instrucciones a ejecutar.

Los criterios para insertar comentarios pueden consultarse en el curso “Bases de la programación nivel I” de aprenderaprogramar.com. Los comentarios son una parte importante de los programas, por lo que es conveniente cuidar su estructuración y contenidos. Dado que con el pseudocódigo trabajamos con corchetes, podemos usar los corchetes detrás de los apóstrofes para "recordar" el carácter de comentario de dicho texto.

'[Esto es un comentario detrás de un apóstrofe. El corchete no es estrictamente necesario pero podemos ponerlo si queremos]

En las empresas de programación suele existir un manual de estilo donde se definen distintas reglas que deben cumplir los programadores, y la forma de comentar los programas suele ser una de ellas.

HERRAMIENTAS LÓGICAS Y MATEMÁTICAS EN VISUAL BASIC

Visual Basic clasifica los operadores con que trabaja de la siguiente manera:

- **Operadores aritméticos:** operadores utilizados para ejecutar cálculos matemáticos.
- **Operadores de comparación:** operadores utilizados para efectuar comparaciones.
- **Operadores de concatenación:** operadores utilizados para combinar cadenas de caracteres.
- **Operadores lógicos:** operadores usados para realizar operaciones lógicas.

Operadores Aritméticos	Operadores de comparación	Operaciones de concatenación	Operadores lógicos
Operador ^	< (Menor que)	&	Operador And
Operador *	<= (Menor o igual que)	+	Operador Eqv
Operador /	> (Mayor que)		Operador Imp
Operador \	>= (Mayor o igual que)		Operador Not
Operador Mod	= (Igual a)		Operador Or
Operador +	<> (Distinto de)		Operador Xor
Operador -	Is		
	Like		

La mayoría de estos operadores han sido explicados en el curso “Bases de la programación nivel I” de aprenderaprogramar.com. El operador & tiene interés para concatenar cadenas. Por ejemplo "Hoy puede" & "ser un gran día" equivale a "Hoy puede ser un gran día". Te remitimos a la ayuda de Visual Basic para obtener mayor información sobre operadores. Ten en cuenta que puede haber pequeñas diferencias entre las distintas versiones de Visual Basic.

En cuanto a las funciones matemáticas, algunas de las funciones disponibles son las siguientes:

Funciones Matemáticas	Significado	Funciones Matemáticas	Significado
Abs	Valor absoluto	Log	Logaritmo neperiano
Atn ó Atan según versión	Arcotangente	Rnd	Generación de número aleatorio
Cos	Coseno	Sgn ó Sign según versión	Signo de un número
Exp	Exponenciación	Sin	Seno

Funciones Matemáticas	Significado	Funciones Matemáticas	Significado
Fix	Truncamiento tipo 1	Sqr ó Sqrt según versión	Raíz cuadrada
Int	Truncamiento tipo 2	Tan	Tangente

Las funciones relacionadas con ángulos trabajan en radianes. Por tanto si queremos operar con grados habremos de realizar la conversión oportuna.

Vamos a centrarnos en algunas funciones que no se han descrito en el curso “Bases de la programación nivel I”. Empezando por Fix e Int, se trata de funciones que devuelven la parte entera de un número, que es diferente a redondear. La diferencia entre ambas es que si el número es negativo Int devuelve el primer entero menor o igual mientras que Fix devuelve el primer entero mayor o igual.

Ejemplo:

$$\text{Int}(8.7) = 8 \qquad \text{Int}(-8.7) = -9$$

$$\text{Fix}(8.7) = 8 \qquad \text{Fix}(-8.7) = -8$$

La función *Rnd* permite generar números aleatorios. Por su importancia para la programación y para la verificación de programas, vamos a desarrollarla un poco más extensamente.

En las versiones más recientes de Visual Basic para usar una función matemática debemos escribir `Math.nombreDeLaFuncion(valorACalcular)`. Por ejemplo, este es un pequeño programa para calcular la raíz cuadrada de un número.

Versiones menos recientes:

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim num As Double

Private Sub Form_Load()
num = 1344
MsgBox ("La raíz cuadrada de " & num & " es " &
Sqr(num))
End Sub
```

Versiones más recientes:

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
Dim num As Double

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
num = 1344
MsgBox("La raíz cuadrada de " & num & " es " &
Math.Sqrt(num))
End Sub
End Class
```

La función *Sgn* (`Math.Sign` en las versiones más recientes) devuelve el signo de un número o de una variable numérica tomando valor 1 para un número positivo, - 1 para un número negativo y cero cuando el número es cero. Supongamos $A = 7.44$, $B = -23$, $C = 0$.

Versiones menos recientes: $Sgn(A) = 1$ $Sgn(43) = 1$
 $Sgn(B) = -1$ $Sgn(-453.33) = -1$
 $Sgn(C) = 0$ $Sgn(0) = 0$

Versiones más recientes: $Math.Sign(A) = 1$ $Math.Sign(43) = 1$
 $Math.Sign(B) = -1$ $Math.Sign(-453.33) = -1$
 $Math.Sign(C) = 0$ $Math.Sign(0) = 0$



Si vas a trabajar con constantes físicas o matemáticas te resultará de interés la instrucción Const para declaración de constantes. La ventaja de declarar una constante en vez de una variable es que la constante no puede variar en el transcurso del programa. Por tanto, se impide que por error pueda tener un valor no válido en un momento dado. Las constantes facilitan la documentación del programa y lo hacen fácil de modificar. Una declaración tipo de constante podría ser la siguiente:

```
Const Pi As Double = 3.14159265358979
```

El redondeo de un decimal a un entero se realiza en Visual Basic a través de la función CInt, correspondiente a un grupo de funciones denominadas "Funciones de conversión de tipos" que no vamos a entrar a estudiar. CInt(A) devuelve el valor de A redondeado al número entero más próximo. Prueba a visualizar el resultado de este código:

```
Dim A As Double  
A = 7.32  
MsgBox("El redondeo de 7.32 es " & CInt(A))
```

Próxima entrega: CU00315A

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