



aprenderaprogramar.com

Bucles For Next Step, Do While Loop, Do Loop While (Estructuras de repetición en Visual Basic) (CU00326A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº25 del Curso Visual Basic Nivel I

29

ESTRUCTURAS DE REPETICIÓN (BUCLES) CON VISUAL BASIC

Vamos a ver cómo materializar con Visual Basic estructuras de repetición que permitirán que en nuestros programas se realice un proceso n veces. En concreto veremos las instrucciones Desde ... Siguiente (For ... Next) con su cláusula Paso (Step), la instrucción Mientras ... Hacer (Do While ... Loop) y la instrucción Hacer ... Repetir Mientras (Do ... Loop While). Muchas veces podremos optar indistintamente por usar una instrucción u otra. En otros casos, es recomendable decantarse por una de ellas por hacer el programa más legible o sencillo que usando otras opciones.

INSTRUCCIÓN DESDE ... SIGUIENTE (FOR ... NEXT) Y CLÁUSULA PASO (STEP)

La sintaxis a emplear con Visual Basic es la siguiente:

```

For VAR = Vi To Vf
    Instrucción 1
    Instrucción 2
    .
    .
    .
    Instrucción n
Next VAR
  
```

El incremento que sufre el contador es, por defecto, unitario. Es decir, el primer valor que toma VAR en el bucle será Vi, el segundo Vi + 1, el tercero (Vi + 1) + 1, etc. La modificación de dicho valor de incremento la realizaremos a través de la cláusula Step después del valor Vf. Step 2 implicará que en cada repetición del bucle el contador se incremente en dos unidades, Step 5 implicará que en cada repetición del bucle el contador se incremente en cinco unidades. Un paso negativo del tipo Step -1 supone que el contador decrece en vez de incrementarse. Si el paso es negativo, Vi necesariamente habrá de ser mayor que Vf, ya que en caso contrario no se producirá la entrada en el bucle.

Con este código se nos muestra en pantalla 3 veces hola (se ha invertido el sentido del bucle):

Código (versiones menos recientes VB):

```

Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim VAR As Integer
Dim Vi As Integer
Dim Vf As Integer
Private Sub Form_Load()
Vi = 1
Vf = 3
For VAR = Vf To Vi Step -1
    '[También supondría tres repeticiones For
VAR = Vi to Vf]
    MsgBox ("hola")
Next VAR
End Sub
  
```

Código (versiones más recientes VB):

```

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
        Dim VAR As Integer
        Dim Vi As Integer
        Dim Vf As Integer
        Vi = 1
        Vf = 3
        For VAR = Vf To Vi Step -1
            '[También supondría tres repeticiones
For VAR = Vi to Vf]
            MsgBox("hola")
        Next VAR
    End Sub
End Class
  
```

Con Visual Basic resulta admisible usar Next sin indicar la variable que está sirviendo de guía del bucle, pues todo bucle ha de tener un cierre. Sin embargo, no lo creemos recomendable pues puede dificultar la lectura y depuración de los programas.

La instrucción For ... Next es anidable dentro de sí misma o dentro de otros tipos de bucles o estructuras, debiendo seguir las pautas que ya hemos comentado.

EJERCICIO

Transformar en código el siguiente planteamiento: queremos generar un programa que pida un número del 1 al 10 y nos muestre la “tabla de multiplicar” correspondiente.

SOLUCIÓN

Hemos realizado la pregunta relativa al número cuya tabla se desea conocer utilizando un InputBox. Igualmente válido hubiera sido hacerlo con un Label contenedor de la pregunta y un TextBox donde el usuario introduce la información. La tabla la mostramos sobre un Label denominado LabelTabla.

Código (versiones menos recientes VB):

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim T As Integer
Dim i As Integer

Private Sub Form_Load()
Form1.Caption = "Tabla"
T = Val(InputBox("¿Qué tabla quiere conocer?" &
vbCrLf & _
"(Introduzca un número de 1 a 10)", "¿Número?"))
LabelTabla.FontBold = True
LabelTabla.Alignment = 2
LabelTabla = "TABLA DEL " & T & vbCrLf &
vbCrLf

For i = 1 To 10
LabelTabla = LabelTabla & T & " * " & i & " = " &
T * i & vbCrLf
Next i
End Sub
```

Código (versiones más recientes VB):

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
Dim T As Integer
Dim i As Integer

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Me.Text = "Tabla"
T = Val(InputBox("¿Qué tabla quiere
conocer?" & vbCrLf & _
"(Introduzca un número de 1 a 10)", "¿Número?"))
LabelTabla.Font = New Font("Arial", 10,
FontStyle.Bold)
LabelTabla.TextAlign = ContentAlignment.MiddleCenter
LabelTabla.Text = "TABLA DEL " & T & vbCrLf &
vbCrLf
For i = 1 To 10
LabelTabla.Text = LabelTabla.Text & T &
" * " & i & " = " & T * i & vbCrLf
Next i
End Sub
End Class
```

Gráficamente:

EJERCICIO

Transformar en código el siguiente planteamiento: queremos generar un programa que muestre todos los enteros comprendidos entre un número definido por el usuario y 100.

SOLUCIÓN

Código (versiones menos recientes VB):

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim Num As Integer
Dim i As Integer

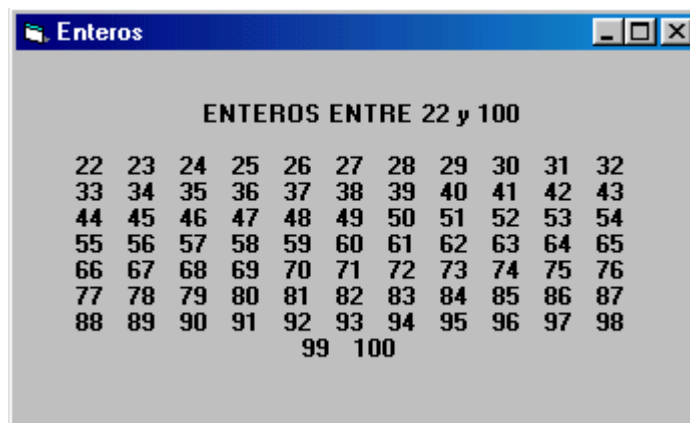
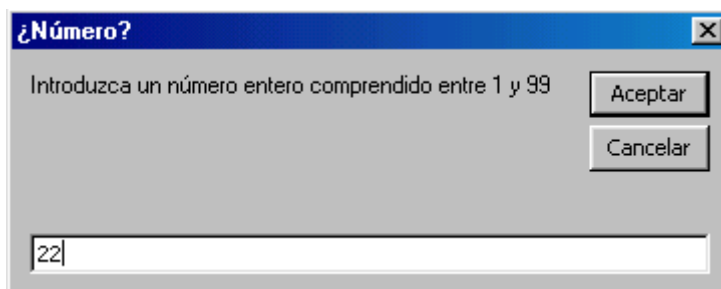
Private Sub Form_Load()
Form1.Caption = "Enteros"
Num = Val(InputBox("Introduzca un número
entero comprendido entre 1 y 99", "¿Número?"))
LabelTabla.Alignment = 2
LabelTabla.FontBold = True
LabelTabla.Text = "ENTEROS ENTRE " & Num & " y
100" & vbCrLf & vbCrLf
For i = Num To 100
LabelTabla.Text = LabelTabla.Text & i & " "
Next i
End Sub
```

Código (versiones más recientes VB):

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

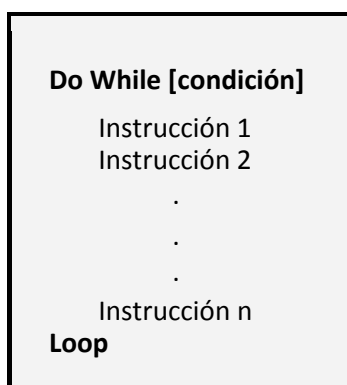
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Dim Num As Integer
Dim i As Integer
Me.Text = "Enteros"
Num = Val(InputBox("Introduzca un número
entero comprendido entre 1 y 99", "¿Número?"))
LabelTabla.TextAlign =
ContentAlignment.MiddleCenter
LabelTabla.Font = New Font("Arial", 10,
FontStyle.Bold)
LabelTabla.Text = "ENTEROS ENTRE " & Num
& " y 100" & vbCrLf & vbCrLf
For i = Num To 100
LabelTabla.Text = LabelTabla.Text & i
& " "
Next i
End Sub
End Class
```

Gráficamente:



INSTRUCCIÓN MIENTRAS ... HACER (DO WHILE ... LOOP)

La sintaxis que hemos de seguir es la siguiente:



Las instrucciones tipo Do While son anidables dentro de sí mismas o dentro de otras estructuras. Es importante verificar que los bucles diseñados con esta instrucción dispongan de una condición de salida válida.

Ejemplo: $k = 0$: Do while $k < 5$. Dentro del bucle tendremos que incrementar el valor de k , $k = k + 1$.

INSTRUCCIÓN HACER ... REPETIR MIENTRAS (DO ... LOOP WHILE)

La sintaxis a utilizar es:

```
Do
  Instrucción 1
  Instrucción 2
  .
  .
  .
  Instrucción n
Loop While [condición]
```

Un Do ... Loop While es anidable dentro de sí mismo o dentro de otras estructuras. Es importante verificar que los bucles diseñados con esta instrucción dispongan de una condición de salida válida.

Ejemplo:

```
Do
  LabelTabla.Text = LabelTabla.Text & "Iteración " & k & vbCrLf
  k = k + 1
Loop While k <= 5
```



Visual Basic admite además de la cláusula While, usar el término Until, como equivalente a "hasta que se cumpla que". Así Loop Until $i \geq 3$ significaría "Repetir hasta que i sea mayor o igual que 3". En un bucle en el que i parte de un valor cero y se incrementa unitariamente Do While $i < 3$ sería equivalente a Do Until $i \geq 3$, y Loop Until $i \geq 3$ sería equivalente a Loop While $i < 3$. Dado que podemos valernos de equivalencias, puede evitar confundirnos el usar preferentemente un mismo tipo de expresión, sabiendo que disponemos de otra equivalente.

Próxima entrega: CU00327A

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