



aprenderaprogramar.com

Pruebas (test) de programas. Verificación de algoritmos en Visual Basic. Cómo recargar formularios VB y .NET (CU00348A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº47 del Curso Visual Basic Nivel I

29

PROGRAMACIÓN RÁPIDA. VERIFICACIÓN DE ALGORITMOS EN VISUAL BASIC.

La programación rápida busca verificar el funcionamiento de los algoritmos antes de entrar en aspectos más formales, como pueden ser la estética, comentarios, etc. La programación rápida puede ser entendida de distintas maneras según los criterios de cada programador. Algunas prácticas de programación rápida pueden ser las siguientes:

- Se usa el formato de salida por pantalla más sencillo posible. En nuestro caso usaremos Labels, TextBox ó msgBox.
- No se piden datos al usuario. La entrada de datos es controlada directamente por el programador, bien usando la instrucción Rnd para generar datos aleatorios o disponiendo una entrada de valores a variables directa, controlada por el programador.
- La estructura del programa no es la definitiva, con los diferentes módulos o procedimientos que pueda llevar, sino provisional.

Vamos a ver un ejemplo de lo que sería programación rápida aplicada al código que se muestra a continuación (Programa SUC02), relativo al cálculo de un sumatorio del tipo:

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

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
```

SOLUCIÓN CÓDIGO DE PROGRAMACIÓN RÁPIDA

Código versiones menos recientes VB:

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

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

Private Sub Form_Load()
Label1 = ""
Database = 85 '[Línea de introducción
manual de datos]
Rem Randomize: Database = Cint(Rnd *
100 + 1) '[Línea de generación _
aleatoria de datos, actualmente
desactivada]
Dato = Database
Raíz01 = Sqr(Dato)
Raíz02 = -Raíz01
Suce = 0
Do While Dato >= 0
    Suce = Suce + Sqr(Dato)
    Dato = Dato - 1
Loop
Label1 = Label1 & "Dato base = " &
Database & vbCrLf & "Raíz01 = " & Raíz01
& " Raíz02 = " & Raíz02 & vbCrLf & "
Valor sucesión = " & Suce
End Sub
```

Código versiones más recientes VB:

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

Option Explicit On
Public Class Form1

    Dim Database, Dato, Raíz01, Raíz02, Suce As Single

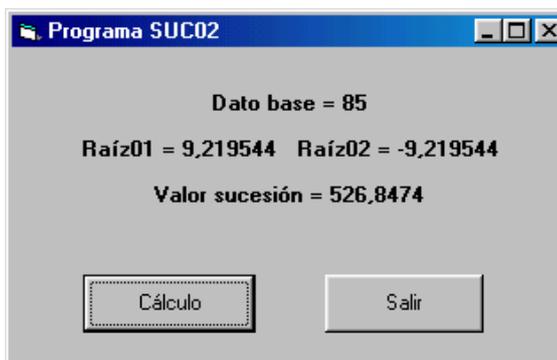
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

Label1.Text = ""
Database = 85 '[Línea introducción manual de datos]
REM generación aleatoria de datos desactivada]
'Randomize() : Database = Cint(Rnd() * 100 + 1)

    Dato = Database
    Raíz01 = Math.Sqrt(Dato)
    Raíz02 = -Raíz01
    Suce = 0
    Do While Dato >= 0
        Suce = Suce + Math.Sqrt(Dato)
        Dato = Dato - 1
    Loop
    Label1.Text = Label1.Text & "Dato base = " &
Database & vbCrLf & "Raíz01 = " & Raíz01 & " Raíz02
= " & Raíz02 & vbCrLf & " Valor sucesión = " & Suce
End Sub
End Class
```

El resultado de este código es coincidente con el que obtuvimos en el [ejercicio de referencia](#) donde creamos el programa, es decir:

```
Datobase = 85
Raíz 01 = 9,219544 Raíz 02 = -9,219544
Valorsucesión = 526,8474
```



La diferencia respecto al código que se propuso en el citado ejercicio está en que:

- Se han eliminado aspectos relacionados con la estética del programa como que se muestre un título de formulario "Programa SUC02". Ya en los programas que venimos desarrollando la atención a

cuestiones estéticas es más bien escasa, pero para la verificación de un algoritmo nos olvidamos completamente de ellas.

- La estructura del programa, que constaba de 2 opciones, Cálculo y Salir, con 3 procedimientos asociados a la opción cálculo (EntraDatos, Proceso y Resultados) se ha eliminado, existiendo un único código asociado a la carga del formulario por ser lo más rápido.
- La petición de datos al usuario con un InputBox de tipo (Dato = Cint(InputBox("Por favor, introduzca nº entero entre 0 y 100", "Dato"))) se ha eliminado. A cambio, existen dos líneas con las que el programador trabaja. Una en la que introduce datos manualmente y otra en las que genera datos aleatoriamente. De este modo puede hacer tanto comprobaciones específicas como comprobaciones aleatorias.
- Las variables que intervienen no coinciden exactamente con las del programa definitivo.

El programador introducirá los cambios que le resulten de interés para la verificación del programa. Por ejemplo, será habitual introducir un botón para que cada vez que sea pulsado se repita la muestra de resultados (generando datos aleatorios). Para ello nos basta desactivar la introducción de datos manual, activar la aleatoria e introducir un botón que se encargue de recargar el formulario. El código sería el siguiente:

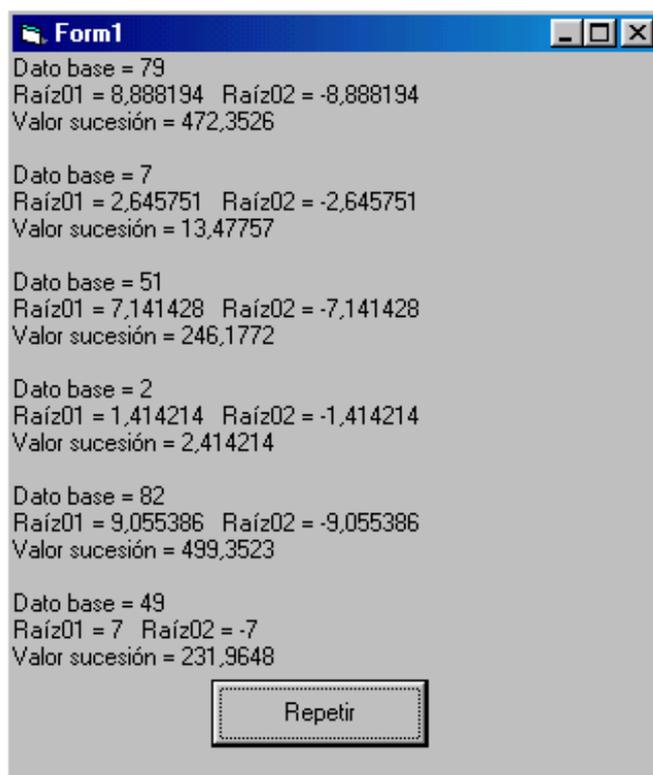
Código versiones menos recientes VB:

Código versiones más recientes VB

```
'Curso VB aprenderaprogramar.com
Private Sub CommandRepetir_Click()
Form_Load
End Sub
```

```
REM Curso Visual Basic aprenderaprogramar.com
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
Me.Form1_Load(Me, New System.EventArgs)
End Sub
```

De este modo podemos llegar a resultados como el que se muestra a continuación:



De una pantalla de este tipo un programador avezado puede obtener numerosa información respecto al funcionamiento del algoritmo. Fijémonos por ejemplo en los resultados para dato base = 49. Las raíces que nos da son 7 y -7. De ahí podemos intuir que el cálculo de raíces se está realizando correctamente. La raíz esperada para 81 sería 9. Tenemos un dato base 82 con raíz 9,05, luego con distintos datos parece que el cálculo de raíces es correcto. En cuanto a los sumatorios, podríamos compararlos con datos de que dispongamos previamente para valorar su corrección, o incluso valorarlos comparativamente. Podemos esperar que el sumatorio para un dato base 79 sea similar (ligemente inferior) al de un dato base 82. Comprobamos que es así.

Si para un dato base 49 se nos mostraran raíces 5,05 y -5,05, o el sumatorio correspondiente a 79 fuera superior a 82, estaríamos ante una situación de no superación de la verificación. Si se diera esta circunstancia, tendríamos que apuntar los datos que nos están dando lugar a resultados erróneos, revisar el algoritmo por si existen errores de sintaxis, transcripción, etc. e incluso modificar su diseño si fuera necesario. Por último, cuando creamos que está funcionando razonablemente, introduciremos de forma manual los datos que apuntamos en su momento como generadores de errores y comprobaremos que para esos datos no hay problema ninguno.

Las propuestas de verificación que hemos visto se corresponden con la verificación funcional del algoritmo, ya que mediante un número limitado de pruebas inferimos que su funcionamiento es correcto. A su vez, para la entrada manual se supone que escogemos una serie de casos representativos, lo que se correspondería con lo que hemos denominado verificación razonada. Por el contrario, la ejecución del programa con datos de entrada generados con Randomize y Rnd se correspondería con una verificación aleatoria.

La verificación total de este algoritmo consistiría en un programa que nos mostrara los resultados para todos los valores posibles (enteros comprendidos entre 0 y 100), de forma que pudiéramos comprobar la corrección de cada uno de ellos. Esto sería posible, pero en general supone un esfuerzo que se considera no necesario ya que las verificaciones funcionales bien realizadas resultan rápidas y efectivas. Además, en muchos casos una verificación completa es inviable, ya que en muchos problemas los posibles datos de entrada son infinitos (por ejemplo, los números enteros).

Próxima entrega: CU00349A

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