



aprenderaprogramar.com

# Ejercicios ejemplos con if then else y operador mod en Visual Basic. Par o impar, ecuación de segundo grado, etc. (CU00324A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº23 del Curso Visual Basic Nivel I

29

## EJERCICIO EJEMPLO CON IF THEN ELSE EN VISUAL BASIC

Crear un programa que pida un número entero distinto de cero y nos muestre en pantalla un mensaje indicándonos si el número es par o impar. **Nota:** El procedimiento a emplear será basado en el uso del operador Mod.

### SOLUCIÓN

Crearemos un Label de solicitud del número, un Textbox de entrada, un Button (Command Button en las versiones menos recientes de Visual Basic) y un Label para el cálculo y el resultado. Le ponemos los nombres y organizamos los controles en pantalla como nos parezca correcto (no tiene por qué coincidir con lo que ponemos aquí). Los nombres que hemos empleado nosotros son: LabelPideNum, TextNentero, ButtonHallar y LabelResultado.

#### Código (versiones menos recientes VB):

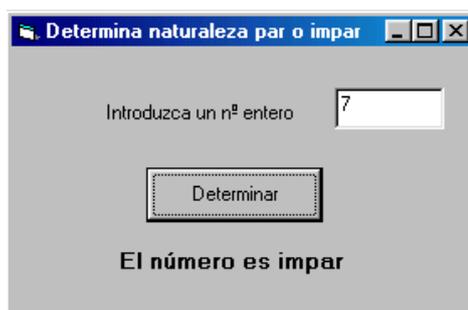
```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim Num As Integer
Dim Res As Integer
Private Sub Form_Load()
Form1.Caption = "Determina naturaleza par o
impar"
ButtonHallar.Caption = "Determinar"
LabelPideNum = "Introduzca un nº entero"
End Sub
Private Sub ButtonHallar_Click()
Num = Val(TextNentero)
Res = Num Mod 2
LabelResultado.FontSize = 10
LabelResultado.FontBold = True
If Res = 0 Then
LabelResultado = "El número es par"
Else
LabelResultado = "El número es impar"
End If
End Sub
```

#### Código (versiones más recientes VB):

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
Dim Num As Integer
Dim Res As Integer
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Me.Text = "Determina naturaleza par o impar"
ButtonHallar.Text = "Determinar"
LabelPideNum.Text = "Introduzca un nº entero"
End Sub

Private Sub ButtonHallar_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles ButtonHallar.Click
Num = Val(TextNentero.Text)
Res = Num Mod 2
LabelResultado.Font = New Font("Arial",
10, FontStyle.Bold)
If Res = 0 Then
LabelResultado.Text = "El número es par"
Else
LabelResultado.Text = "El número es impar"
End If
End Sub
End Class
```

Gráficamente:



Hemos escrito el If ... Then – Else como un bloque en vez de como una única línea. El motivo para ello es que aporta mayor claridad y facilidad de interpretación.

## EJERCICIO ECUACIÓN DE SEGUNDO GRADO (ECUACIÓN CUADRÁTICA)

Crear un programa que resuelva la ecuación cuadrática tipo  $ax^2 + bx + c$ . El planteamiento del problema y su solución en forma de pseudocódigo se pueden encontrar en el curso “Bases de la programación nivel I” de [aprenderaprogramar.com](http://aprenderaprogramar.com), dentro de los ejercicios relacionados con condicionales. Una vez creado el programa, hay que contrastar su correcto funcionamiento.

### SOLUCIÓN

Usaremos tres Labels para pedir los parámetros a, b y c denominados Labela, Labelb y Labelc, y tres TextBox para que el usuario introduzca datos (Texta, Textb y Textc) además de un botón (ButtonCalcular). Nosotros hemos añadido además un mensaje de cabecera con un label (LabelTitulo). Mostraremos primero el código para versiones de Visual Basic más recientes y luego para versiones menos recientes. Ten en cuenta que la lógica de los programas es en ambos casos igual, lo único que varía son los detalles de la sintaxis.

### CÓDIGO (PARA VERSIONES DE VISUAL BASIC MÁS RECIENTES)

```

REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    REM Declaración de variables
    Dim aNumber, bNumber, cNumber, dNumber, eNumber As Single

    REM Contenido del formulario
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Me.Text = "Ecuación cuadrática"
        ButtonCalcular.Text = "Calcular x"
        LabelTitulo.Text = "Parámetros a, b y c para ecuación tipo a*x^2 + b*x + c = 0"

    End Sub

    REM Cálculo y muestra de resultados
    Private Sub ButtonHallar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonCalcular.Click
        LabelResultado.Font = New Font("Arial", 10, FontStyle.Bold)
        aNumber = Val(Texta.Text)
        bNumber = Val(Textb.Text)
        cNumber = Val(Textc.Text)
        dNumber = bNumber ^ 2 - 4 * aNumber * cNumber
        eNumber = 2 * aNumber

        If dNumber = 0 Then
            LabelResultado.Text = "x1 = x2 = " & -bNumber / eNumber
        Else
            If dNumber > 0 Then
                LabelResultado.Text = "x1 = " & (-bNumber + Math.Sqrt(dNumber)) / eNumber
                LabelResultado.Text = LabelResultado.Text & vbCrLf & "x2 = " & (-bNumber - Math.Sqrt(dNumber)) / eNumber
            Else
                LabelResultado.Text = "x1 = " & -bNumber / eNumber & " + " & Math.Sqrt(-dNumber) / eNumber & " * i" & Math.Sqrt(-dNumber) / eNumber & " * i"
            End If
        End If
    End Sub
End Class

```

Nota: en versiones recientes de Visual Basic algunas letras están reservadas y no pueden ser usadas como nombres de variables. Por ejemplo la letra e. De ahí que en vez de nombrar a las variables como a, b, c, d, e las hayamos nombrado aNumber, bNumber, cNumber, dNumber y eNumber.

## CÓDIGO (PARA VERSIONES DE VISUAL BASIC MENOS RECIENTES)

```

Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Rem Declaración de variables
Dim a!, b!, c!, d!, e As Single

Rem Contenido del formulario
Private Sub Form_Load()
Form1.Caption = "Ecuación cuadrática"
CommandCalcular.Caption = "Calcular x"
LabelTitulo = "Parámetros a, b y c para ecuación tipo  $a*x^2 + b*x + c = 0$ "
Labela.Caption = "Valor parámetro a"
Labelb.Caption = "Valor parámetro b"
Labelc.Caption = "Valor parámetro c"
End Sub

Rem Cálculo y muestra de resultados
Private Sub CommandCalcular_Click()
LabelResultado.FontSize = 10
LabelResultado.FontBold = True
a = Val(Texta)
b = Val(Textb)
c = Val(Textc)
d =  $b^2 - 4 * a * c$ 
e = 2 * a
If d = 0 Then
LabelResultado = "x1 = x2 = " & -b / e
Else
If d > 0 Then
LabelResultado = "x1 = " & (-b + Sqr(d)) / e
LabelResultado = LabelResultado & vbCrLf & "x2 = " & (-b - Sqr(d)) / e
Else
LabelResultado = "x1 = " & -b / e & " + " & Sqr(-d) / e & " * i"
LabelResultado = LabelResultado & vbCrLf & "x2 = " & -b / e & " - " & Sqr(-d) / e & " * i"
End If
End If
End Sub

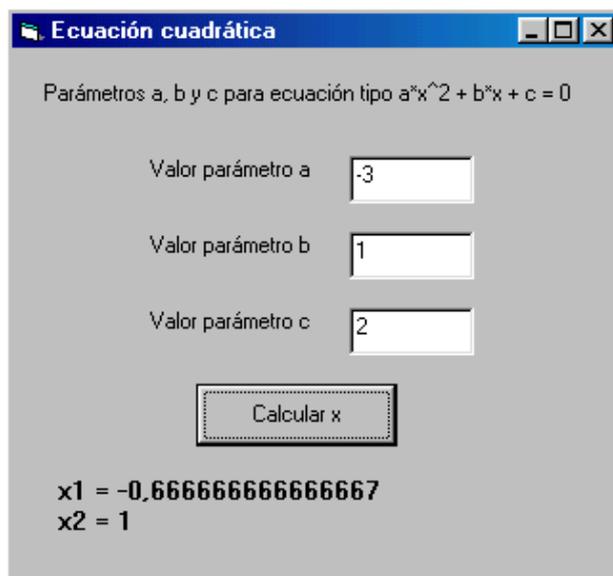
```

## COMENTARIOS

El programa (ambas versiones de código) "funciona", si bien tiene sus limitaciones. Prueba a introducir los siguientes datos:  $a = 0$ ,  $b = 3$  y  $c = 1$ . Te aparecerán resultados extraños (o un error como "Error 6 en tiempo de ejecución. Desbordamiento"). El motivo para esos resultados extraños o errores es que si  $a = 0$  implica que  $e = 0$  y al encontrarnos con una operación del tipo  $(-b + \text{Sqr}(d)) / e$  donde el denominador vale cero... ¡operación matemática no permitida! Esto ocurrirá mientras no preparemos el programa para detectar este tipo de situaciones y darles un tratamiento adecuado, cosa que veremos más adelante.

Estamos construyendo programas sencillos que en líneas generales responden bien cuando se dan las circunstancias previstas pero que fallan ante situaciones extrañas. Esto no debe preocuparnos por el momento, ya que a medida que progreseemos como programadores iremos construyendo programas cada vez más sólidos.

Aspecto gráfico:



Parámetros a, b y c para ecuación tipo  $a \cdot x^2 + b \cdot x + c = 0$

Valor parámetro a: -3

Valor parámetro b: 1

Valor parámetro c: 2

Calcular x

x1 = -0,666666666666667  
x2 = 1

## EJEMPLOS CON IF THEN ELSE EN VISUAL BASIC: PARADOJA DEL SI BURLADO POR UN INTERVALO

Reflexiona sobre el significado y prueba los siguientes códigos en tu ordenador:

**Código 1 (versiones menos recientes de VB):**

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%

Private Sub Form_Load()
A = 160
If 50 < A < 75 Then
MsgBox ("Arabia")
Else
MsgBox ("Eusebio")
End If
End Sub
```

**Código 1 (versiones más recientes de VB):**

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

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
A = 160
If 50 < A < 75 Then
MsgBox ("Arabia")
Else
MsgBox ("Eusebio")
End If
End Sub
End Class
```

**Código 2 (versiones menos recientes de VB):**

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%

Private Sub Form_Load()
A = 160
If A > 50 And A < 75 Then
    MsgBox ("Arabia")
Else
    MsgBox ("Eusebio")
End If

End Sub
```

**Código 2 (versiones más recientes de VB):**

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

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        A = 160
        If A > 50 And A < 75 Then
            MsgBox("Arabia")
        Else
            MsgBox("Eusebio")
        End If
    End Sub
End Class
```

Si bien podría esperarse obtener el mismo resultado con ambos códigos, al ejecutar el código 1 obtenemos como resultado que se muestre en pantalla "Arabia", mientras que al ejecutar el código 2 se muestra "Eusebio". La razón: Visual Basic no admite comparaciones dobles del tipo  $x < y < z$ . Ante este tipo de escritura, sólo es evaluada la primera condición de las dos pretendidas. ¿Solución? No usar comparaciones dobles.

Este "resultado inesperado" puede ocasionar un buen quebradero de cabeza, ya que en un programa extenso no será nada fácil detectar este error. Como remedio, usaremos siempre la escritura de condiciones una detrás de otra y no de forma múltiple.

**Próxima entrega: CU00325A**

**Acceso al curso completo en 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)