



aprenderaprogramar.com

Ubound y Lbound en Visual Basic. Conocer índice superior e inferior de arrays (arreglos) Ejemplos y ejercicios. (CU00343A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº42 del Curso Visual Basic Nivel I

29

MANEJO DE DATOS CON ARRAYS. LÍMITES SUPERIOR (UBOUND) E INFERIOR (LBOUND).

Definido un array $A(n)$ donde n define el índice más grande que puede tener un elemento del array, la función `Ubound(A)` nos devuelve el valor de n y el valor `Lbound(A)` nos devuelve el valor del índice más bajo que puede tener el array (que normalmente será 0).

En las versiones menos recientes de Visual Basic se permitían declaraciones del tipo $A(m \text{ To } n)$, estando los índices del array comprendidos entre m y n . En este caso la función `Lbound(A)` nos devuelve el valor del índice menor del array (m) y la instrucción `Ubound(A)` devuelve el valor del índice mayor (n).

En general el índice menor por defecto en un array será 0. No obstante, en las versiones menos recientes de Visual Basic se permitía establecer como primer índice de un array por defecto el 1 mediante la instrucción `Option Base` escribiendo `Option Base 1`. Si se hace uso de esa opción los resultados de `Lbound` se pueden ver afectados por el valor establecido para `Option Base`, obteniéndose un 0 o un 1 según el caso. En las versiones más recientes de Visual Basic el índice inferior será siempre el cero.

El número de elementos del array será $m - n + 1$, donde m es el límite superior y n el límite inferior. Por ejemplo un array declarado como `Dim A(3) As Integer` consta de $3 - 0 + 1 = 4$ elementos que son $A(0)$, $A(1)$, $A(2)$ y $A(3)$. Prueba el siguiente código:

Código versiones menos recientes de VB:

Código versiones más recientes de VB:

```
'Curso VB aprenderaprogramar.com
Option Explicit
Dim A(45)

Private Sub Form_Load()
Label1 = "Índice menor del array A = "
& Lbound(A) & vbCrLf & vbCrLf
Label1 = Label1 & "Índice mayor del
array A = " & Ubound(A) & vbCrLf &
vbCrLf
Label1 = Label1 & "Array con " &
Ubound(A) - Lbound(A) + 1 & "
elementos"
End Sub
```

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A(45)

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

        Label1.Text = "Índice menor del array A = " &
Lbound(A) & vbCrLf & vbCrLf
        Label1.Text = Label1.Text & "Índice mayor del
array A = " & Ubound(A) & vbCrLf & vbCrLf
        Label1.Text = Label1.Text & "Array con " &
Ubound(A) - Lbound(A) + 1 & " elementos"
    End Sub
End Class
```

En el caso de arrays multidimensionales se hace necesario indicar si queremos referirnos al primer localizador, al segundo, tercero, etc. Así `Lbound(A, 3)` nos devolvería el índice menor del array A en su dimensión 3. Por ejemplo:

Para: `Dim A(55, 20, 33, 60)`

`Lbound(A, 1)` devuelve 0, `Ubound(A, 1)` devuelve 55, `Lbound(A, 2)` devuelve 0, `Ubound(A, 2)` devuelve 20, `Lbound(A, 3)` devuelve 0, `Ubound(A, 3)` devuelve 33, `Lbound(A, 4)` devuelve 0, `Ubound(A, 4)` devuelve 60.

EJERCICIO

Desarrollar el pseudocódigo y diagrama de flujo para un algoritmo que calcule la superficie de un terreno que le corresponde a un heredero después de n generaciones, partiendo de una superficie inicial en la generación cero. Se supone que hay división a partes iguales entre herederos

SOLUCIÓN

Código versiones menos recientes de VB:

```
'Curso VB aprenderaprogramar.com
'[Superficie de terreno]

Option Explicit
Dim i%, n As Integer
Dim Supin!, Toca As Single
Dim Hgen() As Integer

Private Sub Form_Load()
Text1 = "": Text2 = ""
Form1.Caption = "Superficie de terreno"
CommandCalcular.Caption = "Calcular
superficie"
Label1.Caption = "¿Cuál es el número de
generaciones?"
Label2.Caption = "¿Cuál es la superficie inicial
(m2)?"
End Sub

Private Sub CommandCalcular_Click()
n = Val(Text1)
ReDim Hgen(n)
Supin = Val(Text2)
Toca = Supin
For i = 1 To n
    Hgen(i) = InputBox("¿Cuál es el número de
herederos de la generación " & i & "?", "Nº
herederos")
    Toca = Toca / Hgen(i)
Next i
LabelFinal.Alignment = 2
LabelFinal.FontBold = True
LabelFinal = vbCrLf & "Al heredero actual le
corresponde una superficie de " & Toca & " m2"
End Sub
```

Código versiones más recientes de VB:

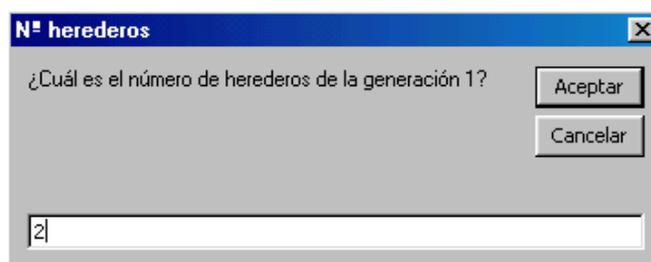
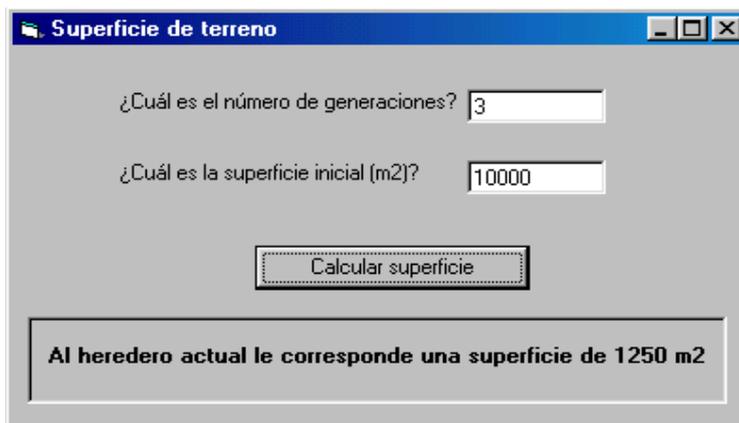
```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        TextBox1.Text = "" : TextBox2.Text = ""
        Me.Text = "Superficie de terreno"
        ButtonCalcular.Text = "Calcular
superficie"
        Label1.Text = "¿Cuál es el número de
generaciones?"
        Label2.Text = "¿Cuál es la superficie
inicial (m2)?"
    End Sub

    Private Sub ButtonCalcular_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles ButtonCalcular.Click
        Dim i, n As Integer
        Dim Supin, Toca As Single
        Dim Hgen() As Integer

        n = Val(TextBox1.Text)
        ReDim Hgen(n)
        Supin = Val(TextBox2.Text)
        Toca = Supin
        For i = 1 To n
            Hgen(i) = InputBox("¿Cuál es el número de
herederos de la generación " & i & "?", "Nº
herederos")
            Toca = Toca / Hgen(i)
        Next i
        LabelFinal.Font = New Font("Arial", 12,
FontStyle.Bold)
        LabelFinal.TextAlign = ContentAlignment.MiddleCenter
        LabelFinal.text = vbCrLf & "Al heredero
actual le corresponde una superficie de " & Toca &
" m2"
    End Sub
End Class
```

Comentarios: Para la generación 1, 2, 3,... el número de herederos correspondiente se almacena en el vector $Hgen(i)$, que se declara como array dinámico por no ser conocido el dato de número de herederos a priori. Gráficamente el programa sería este:



A modo de ejemplo, si los datos de entrada son 3 generaciones, superficie inicial 10000 m2, y en cada generación hay dos herederos, obtenemos como resultado que al heredero actual le corresponde una superficie de 1250 m2.

EJERCICIO

Nos han facilitado el siguiente pseudocódigo y una explicación relativa al objetivo de un programa que debemos realizar:

```

1. Inicio [Extracción de datos p100 curso VB aprenderaprogramar.com]
2. n = 1
3. Mientras Esperado = Falso Hacer
    3.1 Desde i = n hasta n + 99 Hacer
        Leer Dato(i)
        Si Dato(i) > 600 y Dato(i) < 700 Entonces
            j = j + 1
        FinSi
    Siguiete
    3.2 Si j > 100 Entonces
        Esperado = Verdadero
    FinSi
    3.3 Si Esperado = Verdadero Entonces
        Mostrar "Se cumple lo previsto habiendo
        cumplido", j, "datos de un total de", n + 99,
        "datos extraídos"
    SiNo
        Mostrar "Extracción de un nuevo paquete"
        n = n + 100
    FinSi
Repetir
4. Fin
    
```

Nos indican que se trata de extraer datos de un archivo existente en “paquetes” de 100 unidades. Si el número de datos extraídos con valor superior a 600 y menor de 700 es mayor de 100, se debe mostrar el mensaje “Se cumple lo previsto”, indicando el número de datos que se extrajeron cumpliendo los requisitos frente al total extraído. En caso contrario, continuar extrayendo paquetes de datos (se consideran ilimitados). Considerar que los datos se encuentran en un archivo creado previamente.

SOLUCIÓN

Para generar una simulación del archivo con los datos vamos a usar el siguiente código (atención a poner una ruta de archivo correcta, donde queramos guardar el archivo):

Código versiones menos recientes de VB:

```
'Curso VB aprenderaprogramar.com
Option Explicit
Dim Canal As Integer
Dim i As Integer

Private Sub Form_Load()
Canal = FreeFile
Open "C:\misdatos.dat" For Output As Canal
For i = 1 To 300
Randomize
Write #Canal, Int((750 - 550 + 1) * Rnd + 550)
Next i
Close
Label1 = "Datos generados correctamente"
End Sub
```

Código versiones más recientes de 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 myFileToWrite As New
System.IO.StreamWriter("C:\Users\Asus\Desktop\misdat
osnet.dat", False)
For i = 1 To 300
Randomize()
myFileToWrite.WriteLine(Int((750 - 550 +
1) * Rnd() + 550))
Next i
myFileToWrite.Close()
Label1.Text = "Datos generados correctamente"
End Sub
End Class
```

Hemos establecido un rango de datos entre 550 y 750 para provocar que los datos estén en el entorno de los valores esperados. En caso contrario se puede producir que no exista un número suficiente de datos válidos y se genere un error al tratar de extraer más datos de los realmente existentes en el archivo. El programa se muestra a continuación:

Código versiones menos recientes de VB:

Código versiones más recientes de VB

```
'Curso VB aprenderaprogramar.com
'[Extracción datos p100]
Option Explicit
Dim Canal As Integer
Dim i%, j%, n As Integer
Dim Esperado As Boolean
Dim Dato() As Integer

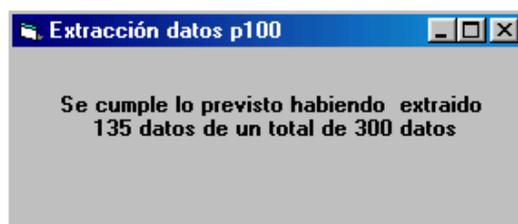
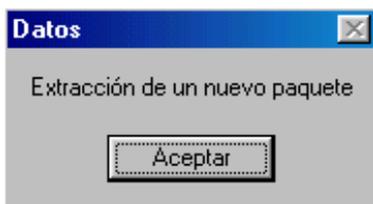
Private Sub Form_Load()
Form1.Caption = "Extracción datos p100"
LabelResultado.Alignment = 2
LabelResultado.FontBold = True
n = 1
Canal = FreeFile
Open "C:\misdatos.dat" For Input As Canal
Do While Esperado = False
ReDim Dato(n + 99)
For i = n To n + 99
Input #Canal, Dato(i)
If Dato(i) > 600 And Dato(i) < 700 Then
j = j + 1
End If
Next i
If j > 100 Then
Esperado = True
End If
If Esperado = True Then
LabelResultado = "Se cumple lo previsto
habiendo cumplido " & j & " datos de un total
de " & n + 99 & " datos extraidos"
Else
MsgBox("Extracción de un nuevo paquete")
n = n + 100
End If
Loop
Close
End Sub
```

```
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 i, j, n As Integer
Dim Esperado As Boolean
Dim Dato() As Integer

Me.Text = "Extracción datos p100"
LabelResultado.Font = New Font("Arial", 12,
FontStyle.Bold)
LabelResultado.TextAlign =
ContentAlignment.MiddleCenter
n = 1
Dim myFileToRead As New
System.IO.StreamReader("C:\Users\Asus\Desktop\misdat
osnet.dat", False)
Do While Esperado = False
ReDim Dato(n + 99)
For i = n To n + 99
Dato(i) = myFileToRead.ReadLine()
If Dato(i) > 600 And Dato(i) < 700 Then
j = j + 1
End If
Next i
If j > 100 Then
Esperado = True
End If
If Esperado = True Then
LabelResultado.Text = "Se cumple lo
previsto habiendo cumplido " & j & " datos de un
total de " & n + 99 & " datos extraidos"
Else
MsgBox("Extracción de un nuevo paquete")
n = n + 100
End If
Loop
myFileToRead.Close()
End Sub
End Class
```

Gráficamente:



Próxima entrega: CU00344A

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