

# Estructura Iterativas

## Bucles:

For, While, do...While

### Estructura Iterativa **Bucle For**

Un bucle **for** es una estructura de control de repetición que permite escribir de manera eficiente un bucle que es necesario ejecutar un número determinado de veces. Un bucle **for** es útil cuando se sabe cuántas veces una tarea se va a repetir.



Hacer un programa en Visual Basic. Net que permita mostrar la tabla de sumar, restar, multiplicar y dividir de un número ingresado previamente.

Windows application window titled "Estructuras Operativas".

Section: Tabla aritmetica

Número:

Buttons: Nuevo, Calcular, Salir

Dropdown menus: Suma, Resta, Multiplicación, División

```

1  Public Class Form1
2
3      Private Sub limpiar()
4          cbosuma.Items.Clear()
5          cboresta.Items.Clear()
6          cbomulti.Items.Clear()
7          cbodivision.Items.Clear()
8      End Sub
9
10     Private Sub btnnuevo_Click(sender As Object, e As EventArgs) Handles btnnuevo.Click
11
12         Call limpiar()
13         txtnumero.Clear()
14
15     End Sub
16
17     Private Sub btncalcular_Click(sender As Object, e As EventArgs) Handles btncalcular.Cl
18         'Dejar en blanco los combobox
19         Call limpiar()
20
21         'Declaracion de variables
22         Dim numero As Integer
23         Dim suma As Integer
24         Dim resta As Integer
25         Dim multi As Integer
26         Dim div As Double
27         'Entrada de datos
28         numero = Val(txtnumero.Text)

```

```

29
30     'OPERACION
31     For i = 1 To 12 Step 1
32         suma = numero + i
33         resta = numero - i
34         multi = numero * i
35         div = numero / i
36         'Salida de informacion
37         cbosuma.Items.Add(numero & "+" & i & "=" & suma)
38         cboresta.Items.Add(numero & "-" & i & "=" & resta)
39         cbomulti.Items.Add(numero & "*" & i & "=" & multi)
40         cbodivision.Items.Add(numero & "/" & i & "=" & div)
41     Next
42
43 End Sub
44
45 Private Sub btnsalir_Click(sender As Object, e As EventArgs) Handles btnsalir.Click
46     Me.Close()
47 End Sub
48 End Class

```

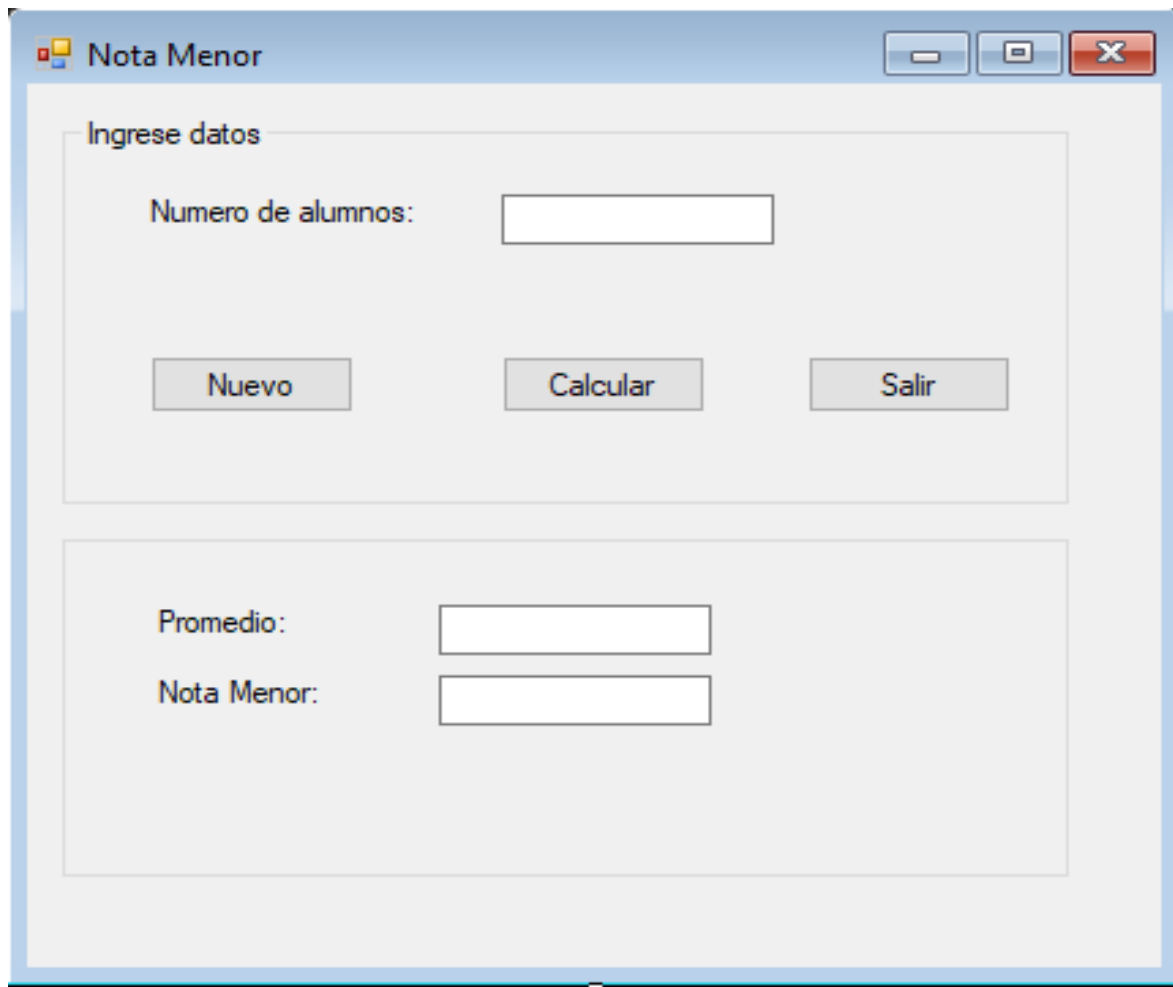
**Contador** : Son variables enteras que se incrementan (+) o decrementan (-).

**num+=1    num-=1    num=num+1**

**Acumulador**: Son variables de cualquier tipo que almacenan valores variables.

**num+=valor    num-=i    num=num+valor**

**Se tiene un conjunto de calificaciones de un grupo de “n” alumnos, realizar un algoritmo para calcular la calificación media y la calificación más baja de dicho grupo.**



The image shows a screenshot of a Windows application window titled "Nota Menor". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is divided into two sections. The top section, labeled "Ingrese datos", contains a label "Numero de alumnos:" followed by a text input field. Below this, there are three buttons: "Nuevo", "Calcular", and "Salir". The bottom section contains two labels, "Promedio:" and "Nota Menor:", each followed by a text input field.

Nota Menor

Ingrese datos

Numero de alumnos:

Promedio:

Nota Menor:

```

1  Public Class Form1
2  Private Sub btncalcular_Click(sender As Object, e As EventArgs) Handles btncalcular.Click
3      ...
4
5      Try
6          'Declaracion de variables
7          Dim numalumnos As Integer
8          Dim nota, sumanotas, notamenor, promedio As Double
9
10         If (Val(txtalumno.Text) - Int(Val(txtalumno.Text))) = 0 Then
11
12             'Entrada de datos
13             numalumnos = Val(txtalumno.Text)
14
15             'Inicializamos variables
16             sumanotas = 0
17             notamenor = 100
18
19             For i = 1 To numalumnos Step 1
20                 nota = InputBox("Ingrese la nota del alumno" & i, "Registro de notas")
21                 sumanotas = sumanotas + nota
22                 ...
23
24                 'Evaluamos la nota menor
25                 If (nota < notamenor) Then
26                     notamenor = nota
27                 End If
28             Next
29
30             promedio = sumanotas / numalumnos
31
32             'Salida de datos
33             txtmenor.Text = notamenor
34             txtpromedio.Text = promedio
35         End If
36     End Try
37 End Sub

```

```

32 Else
33     MessageBox.Show("Ingrese un valor entero", "Promedio de notas", MessageBoxButtons.OK, MessageBoxIcon.Error)
34
35
36
37 End If
38
39 Catch ex As Exception
40     MsgBox(ex.Message)
41 End Try
42 End Sub
43
44 Private Sub btnnuevo_Click(sender As Object, e As EventArgs) Handles btnnuevo.Click
45     txtalumno.Clear()
46     txtmenor.Clear()
47     txtpromedio.Clear()
48
49 End Sub
50
51 Private Sub btnsalir_Click(sender As Object, e As EventArgs) Handles btnsalir.Click
52     Me.Close()
53
54 End Sub
55 End Class
56

```

## Estructura Iterativa Bucle Do-While

Una estructura iterativa **Do-While** ejecuta al menos una vez su bloque repetitivo, a diferencia del **while** o del **for** que podrían no ejecutar el bloque, este bloque de instrucciones se repetirán siempre que la condición sea verdadera.



Estructura  
Iterativa

Do

$s=s+1$

Instrucciones  
que se repetirán

Loop While (**nota<0 Or nota>20**)

Expresión Lógica

Hacer un programa en Visual Basic que permita validar el ingreso de notas (70 a 90) para “n” alumnos, es decir mientras la nota este fuera de ese rango el programa deberá indicar al usuario ingresar la nota, hasta que se ingrese la nota correcta; al final mostrar el promedio.

```

1 Public Class Form1
2     Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click
3
4     End Sub
5
6     Private Sub btningresar_Click(sender As Object, e As EventArgs) Handles btningresar.Click
7         'Declaracion de variables
8
9         Dim cantAlumnos As Integer, nota As Integer
10        Dim suma As Integer
11        Dim promedio As Double
12
13        'Iniamos variables
14        suma = 0
15
16        'Entrada de datos
17        cantAlumnos = Val(txtalumnos.Text)
18        'Proceso
19        'Implementar un ciclo para pedir las notas con input box
20        For i As Integer = 1 To cantAlumnos Step 1
21            Do
22                nota = InputBox("Ingrese la nota del estudiante" & i, "Notas")
23
24                Loop While (nota < 70 Or nota > 90)
25                suma = suma + nota
26            Next
27        promedio = suma / cantAlumnos
28

```

```

29        'salida de informacion
30        txtpromedio.Text = promedio
31
32    End Sub
33
34    Private Sub btnnuevo_Click(sender As Object, e As EventArgs) Handles btnnuevo.Click
35
36        txtpromedio.Clear()
37        txtalumnos.Clear()
38    End Sub
39
40    Private Sub btnsalir_Click(sender As Object, e As EventArgs) Handles btnsalir.Click
41
42
43    End
44    End Sub
45 End Class
46

```



Una estructura iterativa **While** permite repetir una o más instrucciones mientras la condición (Expresión lógica) sea verdadera, cuando la condición es falsa sale del bucle.

Estructura  
Iterativa

Expresión Lógica

Do **While** (**stock >= 10**)  
cantP=cantP+1

Loop

Instrucciones  
que se repetirán

Una organización tiene punto de repartos de vacunas, cada día se registra con cuantas vacunas se inicia.

Crear un programa en Visual Basic que gestione las entregas (validar que la cantidad a entregar no sea mayor que el stock) y de un aviso cuando el stock es inferior a 10 unidades.

Reparto de vacunas

Ingreso de Datos

Stock inicial:

Nuevo Repartir Salir

Existencias

Stock:

Cantidad de personas:

```

5
6 Private Sub btnrepartir_Click(sender As Object, e As EventArgs) Handles btnrepartir.Click
7     'DECLARACION DE VARIABLES
8     Dim stock As Integer
9     Dim cantPersonas As Integer
10    Dim cantEntrega As Integer
11
12    'INICIALIZACION DE VARIABLES
13    cantPersonas = 0
14
15    'ENTRADA DE DATOS
16    stock = Val(txtStock_inicial.Text)
17    'REPARTIR VACUNAS
18    Do While (stock >= 10)
19        cantEntrega = InputBox("Ingrese cant. entregar " & (cantPersonas + 1), "Reparto de vacunas")
20        'Validar que la cant. entregada sea menor o igual al stock disponible
21        If (stock >= cantEntrega) Then
22            cantPersonas = cantPersonas + 1
23            stock = stock - cantEntrega
24        Else
25            MessageBox.Show("No hay suficiente stock", "Reparto de vacunas", MessageBoxButtons.OK, MessageBoxIcon.Error)
26        End If
27    Loop
28
29    MessageBox.Show("El stock es menor a 10 uds", "Reparto de vacunas", MessageBoxButtons.OK, MessageBoxIcon.Information)
30
31    'SALIDA DE INFORMACION
32

```

```
33     txtStock.Text = stock
34     txtCant_personas.Text = cantPersonas
35
36
37
38 End Sub
39
40 Private Sub btnnuevo_Click(sender As Object, e As EventArgs) Handles btnnuevo.Click
41
42     txtCant_personas.Clear()
43     txtStock.Clear()
44     txtStock_inicial.Clear()
45 End Sub
46
47 Private Sub btnsalir_Click(sender As Object, e As EventArgs) Handles btnsalir.Click
48
49     End
50 End Sub
51 End Class
52
```