

### Módulo 3

### Estruturas de Repetição I

## Conteúdo

---

- Conceito e aplicação
  - O que são e para que servem as estruturas de repetição.
- Ciclo Do . . . Loop
  - Apresentação das diversas variantes deste ciclo. Negação de expressões lógicas.
  - Exemplos de aplicação.

## Conceito

- Uma grande parte das linguagens de programação têm instruções que permitem repetir blocos de instruções. De um modo genérico, um ciclo condicional será:

- Sintaxe (geral)

```
enquanto <expressão lógica>  
  <bloco de instruções>  
repetir  
  <instrução>
```

- Explicação

- Se a expressão lógica for verdadeira é executado o bloco de instruções. Uma vez terminada a execução do bloco, volta-se a avaliar a expressão e o processo repete-se. Quando a expressão for falsa, passa-se à execução da instrução a seguir ao ciclo.

## Sintaxe do ciclo Do...Loop I

- Há quatro variantes deste ciclo. Começemos pelo teste no início do ciclo.

- DWL, teste da expressão lógica no início, executa enquanto a expressão for verdadeira

```
Do while <expressão lógica>  
  <bloco de instruções>  
Loop
```

- DUL, teste da expressão lógica no início, executa até que a expressão seja verdadeira

```
Do until <expressão lógica>  
  <bloco de instruções>  
Loop
```

## Sintaxe do ciclo Do...Loop II

- Vamos agora ver as versões com teste no fim do ciclo.

- DLW, teste da expressão lógica no fim, executa enquanto a expressão for verdadeira

```
Do
  <bloco de instruções>
Loop while <expressão lógica>
```

- DLU, teste da expressão lógica no fim, executa até que a expressão seja verdadeira

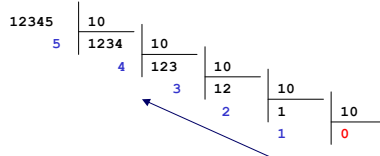
```
Do
  <bloco de instruções>
Loop until <expressão lógica>
```

## Exemplo de ciclo DUL

- Elabore um programa que leia um número e determine o resto da divisão por 9 ("noves fora") da soma dos algarismos que o compõem.

- Resolução:

- Exemplo: 12345 -> 6

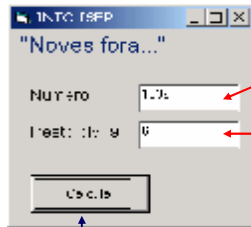


$$1+2+3+4+5 \rightarrow 15$$
$$15 \bmod 9 \rightarrow 6$$

- Divisão de número inteiros:

- $a \bmod b$  - calcula resto da divisão de a por b
- $a \setminus b$  - calcula quociente da divisão a por b

## Resolução



`txtNum` - *TextBox* para introdução do número.

`txtResto` - *TextBox* para apresentar o resultado

`cmdcalc` - *CommandButton* para executar o cálculo

`Abs()` calcula o módulo de um número e `Val()` converte o texto da *TextBox* para um valor numérico.

Option Explicit

```
Private Sub cmdCalc_Click()
    Dim n As Long, alg As Integer, s As Integer
    n = Abs(Val(txtNum.Text))
    Do Until n = 0
        alg = n Mod 10
        s = s + alg
        n = n \ 10
    Loop
    txtResto.Text = s Mod 9
End Sub
```

Pára quando `n` chegar a zero

Tem que se ir somando os algarismos à medida que são extraídos

## Execução passo a passo

- Teste com o valor 234

Iteração	n	s	n = 0	alg
	234	0		
1			False	4
		0+4 -> 4		
2	23		False	3
		4+3 -> 7		
3	2		False	2
		7+2 -> 9		
4	0		True	

## Exemplo de ciclo DWL

O mesmo problema podia ser resolvido com o ciclo DWL:

```
Option Explicit
```

```
Private Sub cmdCalc_Click()
```

```
Dim n As Long, alg As Integer, s As Integer
```

```
n = Abs(Val(txtNum.Text))
```

```
Do While n <> 0
```

```
alg = n Mod 10
```

```
s = s + alg
```

```
n = n \ 10
```

```
Loop
```

```
txtResto.Text = s Mod 9
```

```
End Sub
```

A única alteração no programa consiste em substituir o `until` por `while` e negar a expressão lógica.

Regras de negação:

`Not(A And B) ⇔ Not(A) Or Not(B)`

`Not(A Or B) ⇔ Not(A) And Not(B)`

Op.	Neg.
=	<>
>	<=
>=	<
And	Or

## Exemplo de ciclo DLU

O mesmo problema podia ser resolvido com o ciclo DLU:

```
Option Explicit
```

```
Private Sub cmdCalc_Click()
```

```
Dim n As Long, alg As Integer, s As Integer
```

```
n = Abs(Val(txtNum.Text))
```

```
Do
```

```
alg = n Mod 10
```

```
s = s + alg
```

```
n = n \ 10
```

```
Loop Until n = 0
```

```
txtResto.Text = s Mod 9
```

```
End Sub
```

Desta vez a alteração no programa, em relação ao DUL, consiste em mover a avaliação da expressão lógica para o fim do ciclo.

## Execução passo a passo

- Teste com o valor 234

Iteração	n	s	n = 0	alg
	234	0		
1				4
		0+4 -> 4		
	23			
			False	
2				3
		4+3 -> 7		
	2			
			False	
3				2
		7+2 -> 9		
	0			
			True	

## Exemplo de ciclo DLW

O mesmo problema podia ser resolvido com o ciclo DLW:

```
Option Explicit

Private Sub cmdCalc_Click()
    Dim n As Long, alg As Integer, s As Integer

    n = Abs(Val(txtNum.Text))

    Do
        alg = n Mod 10
        s = s + alg
        n = n \ 10

        Loop while n <> 0

    txtResto.Text = s Mod 9
End Sub
```

← Desta vez a alteração no programa, em relação ao DLU, consiste em negar a expressão lógica.

## Selecco do ciclo adequado

- O exemplo apresentado podia ser resolvido de forma satisfatria por qualquer um dos ciclos. Nem sempre  assim.
- H duas situaes distintas:
  - O ciclo deve ser executado pelo menos uma vez.
    - O teste  executado no fim do ciclo
    - DLW e DLU
  - O ciclo pode no ser executado uma nica vez
    - Teste no incio do ciclo
    - DWL e DUL
- A escolha entre **while** e **until**  subjectiva. So equivalentes, se a expresso lgica for negada.

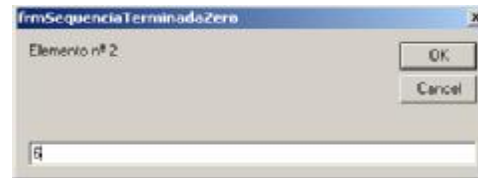
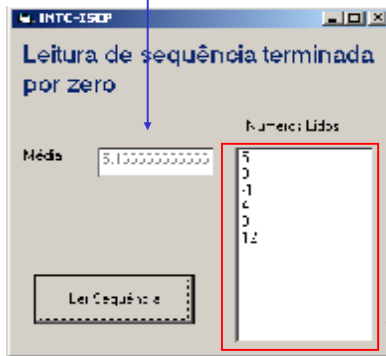
## Exemplo - Sequncia

- Elabore um programa que leia uma sequncia de nmeros terminada por zero. O programa dever dar como sada a mdia dos introduzido, sem considerar o zero. Todos os nmeros introduzidos devem ainda ser visualizados numa ListBox, com a excepo do zero.
- Vamos resolver o exerccio de duas formas, recorrendo aos ciclos:
  - DWL
  - DLU

## Resolução - Form

- ListBox – objecto que permite visualizar uma lista de elementos, um em cada linha.

txtMed – TextBox onde é visualizada a média.



InputBox para leitura dos números da sequência.

lstNum – ListBox onde são colocados os números lidos.

## Resolução DWL - Código

Option Explicit

Private Sub cmdLer\_Click()

Dim num As Integer, c As Integer, s As Long

lstNum.Clear

num = Val(InputBox("Elemento nº 1"))

Leitura do primeiro número, recorrendo a uma InputBox.

Do While num <> 0

Enquanto o número for diferente de zero. Se o primeiro for zero, não executa o ciclo.

lstNum.AddItem num

Adiciona o número à ListBox lstNum

s = s + num

c = c + 1

num = Val(InputBox("Elemento nº " & (c + 1)))

Leitura dos números seguintes, recorrendo a uma InputBox.

Loop

If c <> 0 Then

txtMed.Text = s / c

Else

txtMed.Text = ""

MsgBox "Não há elementos a considerar!"

End If

End Sub

Verifica se foi lido algum número diferente de zero.



```
Option Explicit
Private Sub cmdLer_Click()
    Dim num As Integer, c As Integer, s As Long

    lstNum.Clear ← Limpa conteúdo anterior da ListBox.
    Do
        num = Val(InputBox("Elemento n° " & (c + 1)))
        If num <> 0 Then ← Se o número for diferente de zero então
            lstNum.AddItem num ← adiciona-o e conta mais um elemento lido.
            s = s + num
            c = c + 1 ← Conta número de elementos lidos (não nulos).
        End If
    Loop Until num = 0 ← Executa o ciclo até que o valor
    lido seja 0.

    If c <> 0 Then
        txtMed.Text = s / c
    Else
        txtMed.Text = ""
        MsgBox "Não há elementos a considerar!"
    End If
End Sub
```