Informática de Gestão
1º ano / 1º semestre
Ano letivo: 2014/2015

Visual Basic – VBA

As Macros (ou subrotinas) podem ser de dois tipos:

- **Macros de comandos**: Armazenam sequências de operações do utilizador no Excel.
- **Criadas pelo utilizador**: Resultam da criação do utilizador, sendo estas expressas através da linguagem de programação do VB.

Sintaxe de uma Macro

O VBA expressa uma Macro da seguinte maneira:

Sub apresentação_macro()
    <Instrução 1>
    <Instrução 2>
    ...
End sub

Quanto a sua natureza as macros podem ser **públicas** ou **privadas** de acordo com a sua **disponibilidade nos diferentes módulos**. Assim

- As macros são **privadas** quando estão apenas **disponíveis no seu módulo**;
- As macros são **públicas** quando estão **disponíveis em todos os módulos** e procedimentos do projeto.

Macro Privada	Macro Pública
Private Sub macro_privada() <Instrução 1> <Instrução 2> **End sub**	**Public Sub** macro_pública() <Instrução 1> <Instrução 2> **End sub**

Tipos de Variáveis

Enquanto que no Excel podemos dizer que o **conteúdo** de uma **célula se encontra formatado** de uma determinada maneira, no **VB** somos **obrigados** a fazê-lo. Para isso, temos então de **declarar a variável**.

Excel	VBA
12-10-2014	**Dim** valor_positivo **As Integer**
A célula pode ser formata para apresentar uma **data**, um **número inteiro**, um **número com apenas 1 casa decimal**, etc ...	No VBA é necessário declararmos as **variáveis** que vamos utilizar bem como indicar o seu **formato** (o seu tipo)

Tipos de variáveis VBA	
Tipos	**Exemplo**
Integer	Inteiros (-32768/32767)
Double	Reais Longos
Single	Reais
String	Armazena Caracteres
Long	Inteiros Longos
Date	Data
Boolean	Valores lógicos (verdadeiro ou falso)
Byte	Números sem sinal
Currency	Moeda

Para declarar uma variável é necessário 3 coisas. Em primeiro lugar temos usar a instrução "**Dim**", depois temos de indicar o nome da variável e por fim que *tipo de variável* se trata. Assim um possível exemplo de declaração de uma variável é:

Dim variável_x *As Integer*

- **Dim** – Instrução utilizada para indicar ao VB que vamos iniciar a declaração da variável;
- variável_x – Nome da variável;
- *As Integer* – Formato da variável (tipo).

Nota: O Array (ou matriz) permite armazenar várias variáveis do mesmo tipo.

Atribuição de dados as variáveis

Para fazer a atribuição de dados a variáveis é necessário que se associe os dados as variáveis. Sintaxe: **variável = valor**

Exemplo: Crie uma macro que pede ao utilizador a sua idade e o seu nome e devolve uma frase que junte os dois dados.

```
Public Sub dados_utilizador()
        Dim nome As String
        Dim idade As Integer
        nome = InputBox("Como é que te chamas?")
        idade = InputBox("Quantos anos tens?")
        MsgBox ("O nome é " & nome & " e tem " & idade & " anos")
End Sub
```

Interatividade com o utilizador

A interatividade com o utilizador pode ser realizada de duas maneiras, ou através de uma **MsgBox** ou através de uma **InputBox**.

Numa **InputBox**(“Insira um valor”)

- É uma **forma de entrada** de dados;
- Mostra **uma janela** com uma **caixa** que permite **inserir dados**;
- O utilizador deve inserir os dados ou então acionar uma maneira de sair do menu;
- A informação inserida é **guardada na variável que o utilizador indicar**.

Numa **MsgBox**(“O valor inserido é “ & valor)

- ✓ É uma **forma de saída** de dados;
- ✓ Exibe no ecrã uma janela com uma **mensagem e/o com o conteúdo de uma variável**.

Exemplo 1: Crie uma macro que pede o nome ao utilizador e devolve uma frase como “Tu chamas-te X”.

```
Sub Public nome()
        Dim nome As String
        nome = InputBox(“Como é que te chamas?”)
        MsgBox (“ Tu chamas-te “ & nome)
End Sub
```

Exemplo 2: Crie uma macro que pede ao utilizador o seu nome e a sua idade e de acordo com a sua idade irá aparecer uma mensagem personalizada como por exemplo “ O teu nome é X e tens Y anos, e és *mensagem personalizada*”.

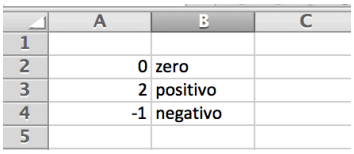
```
Sub Public nomes_idades()
        Dim nome As String
        Dim idade As Integer
        Dim personalizacao As String

        nome = InputBox("Como é que te chamas?")
        idade = InputBox("Quantos anos tens?")

        If idade < 20 Then
                personalizacao = " és muito novinho."
        ElseIf idade < 40 Then
                personalizaçao = " és maduro."
        ElseIf idade < 65 Then
                personalizacao = " és experiente."
        Else
                personalizacao = " és muito maduro e experiente."
        End If

        MsgBox (" O teu nome é " & nome & " e tens " & idade & " anos, e " &
        personalizacao)
End Sub
```

Estruturas de Controlo

Existem duas estruturas de controlo, as **estruturas condicionais** (vulgarmente conhecidas por **If**) ou **estruturas repetitivas** que são constituídas por instruções de iteração (**For**, **Do**, **While**).

Estrutura condicional

Exemplo 1: Escreva uma função e uma macro que indica se o valor é positivo, negativo ou nulo.



	A	B	C
1			
2	0	zero	
3	2	positivo	
4	-1	negativo	
5			

Função:

```
Public Function sinal( valor As Integer ) As String

    If valor > 0 Then
        Sinal = "positivo"
    Elseif valor < 0 Then
        Sinal = "negativo"
    Else
        Sinal = "nulo"
    End If
End Function
```

Macro:

```
Public Sub sinal()
    Dim valor As Integer
    valor = InputBox("Insira um número")

    If valor > 0 Then
        Sinal = "positivo"
    Elseif valor < 0 Then
        Sinal = "negativo"
    Else
        Sinal = "nulo"
    End If

    MsgBox(" O número que inseriu é " & sinal)
End sub
```

Exemplo 2: Faça uma função que calcula o valor de desconto e o valor final (valor a pagar) utilizando o ciclo IF. Tenha em consideração que no caso de a compra ser num valor inferior a 100€ existe 5% de desconto, no caso de a compra ter um valor entre 100€ e 500€ existe 10% de desconto e compras num montante superior a 500€ existe 15% de desconto.

```
Public Sub desconto ()
        Dim compra As Integer
        Dim valor_pagar, valor_desconto As Double
        compra = InputBox("Quanto é que pagou?")

        If compra < 100 Then
                valor_desconto = compra*0,05
        Elseif compra < 500 Then
                valor_desconto = compra*0,10
        Else
                valor_desconto = compra*0,15
        End if

        valor_compra = compra – valor_desconto
        MsgBox(" O desconto é de " & valor_desconto & " €. O valor final " & valor_compra)
End Sub
```

Instruções de Interação

Instrução	Descrição
For ... Next	Percorre um conjunto de valores de um dado intervalo
For Each ... Next	Percorre um conjunto de elementos de um conjunto
Do ... Loop	Repete um conjunto de instruções enquanto ou até que uma determinada condição se verifique
While ... Wend	Repete o ciclo de instruções que esta entre o While e o Wend enquanto uma determinada condição for verdadeira

- O ciclo **For ... To ... Next** vai realizar um determinado conjunto de instruções desde um **valor *x* até um valor *y***.
- O ciclo **While ... Wend** vai realizar um conjunto de instruções **enquanto a condição se verificar**.
- O ciclo **Do Until ... Loop** vai repetir um conjunto de instruções **até** que uma **determinada condição se verifique**.
- O ciclo **Do While ... Loop** vai realizar um determinado conjunto de instruções **enquanto uma determinada condição se verificar**.

Instruções de Iteração – Ciclos Repetidos

O ciclo **FOR** é utilizado quando sabemos o **número de vezes** que vamos ter de **repetir** um determinado **conjunto de instruções**.

Sintaxe do ciclo: **For** <contador=valor_inicial> **To** <valor_final> |**Step** <incrementar valor>|
<Instruções>
<Instruções>
Next contador

Exemplo 1: Utilizando uma macro calcule a soma de todos os números inteiros de 1 até ao valor que o utilizador inseriu.

```
Public Sub soma()
        Dim i, n, soma As Integer
        n = InputBox(“Insira um valor”)
        i = 1
        soma = 0

        For i=1 To n
            soma = soma + i
        Next i
        MsgBox(“ A soma de todos os valores de 1 até “ & n “ é “ & soma)
End Sub
```

Exemplo 2: Utilizando a macro anterior calcule a soma de todos os números pares (o utilizador apenas insere números pares).

```
Public Sub soma_pares()
        Dim i, n, soma As Integer
        n = InputBox(“Insira um valor”)
        i = 0
        soma = 0

        For i=0 To n Step 2
            soma = soma + i
        Next i
        MsgBox(“ A soma de todos os valores de 1 até “ & n “ é “ & soma)
End Sub
```

Nota: O **Step** no ciclo FOR vai funcionar como modificador da frequência do ciclo, ou seja, em vez de o ciclo FOR ocorrer de **um** em **um** número pode ocorrer de **n** em **n** números.

Exemplo 3: Faça uma macro que recebendo a base e a potencia calcula o respetivo valor.

```
Public Sub potencia_n()
  Dim potencia, n, i As Integer
  Dim n_potencia As Long
  n = InputBox("Insira uma base")
  potencia = InputBox("Insira o expoente")
  n_potencia = 1

  For i = 1 To potencia
    n_potencia = n_potencia * n
  Next i

  MsgBox ("O valor é " & n_potencia)
End Sub
```

Instruções de Iteração – Ciclo While

O ciclo **WHILE** repete um **conjunto de instruções**, que se encontram entre o **While e o Wend**, enquanto se verifica uma determinada condição.

Sintaxe do ciclo: **While** <condição>
<instruções>
<instruções>
Wend

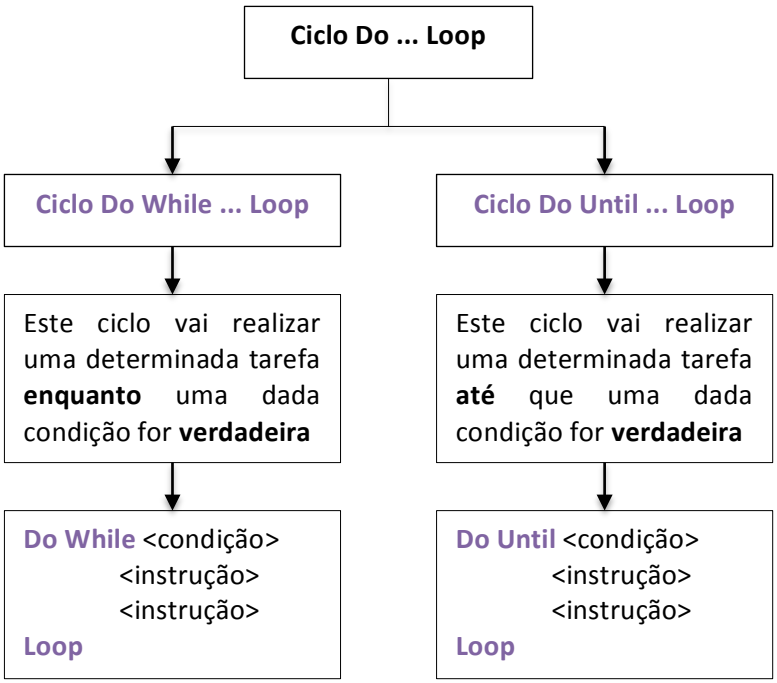
Nota: Ao contrário do que acontece com o ciclo FOR que só se utiliza quando se sabe o número de vezes que as instruções vão ser repetidas, o ciclo WHILE utiliza-se quando não se sabe quantas vezes temos de repetir as instruções.

Exemplo: Construa uma macro que soma todos os valores até que o utilizador insira um 0.

```
Public Sub ate_zero()
  Dim soma, valor As Integer
  soma = 0
  valor = InputBox("Insira um valor")

  While valor <> 0
    soma = soma + valor
    valor = InputBox("Insira outro valor")
  Wend

  MsgBox (" A soma de todos os valores inseridos é " & soma)
End Sub
```

Instruções de Iteração – Do ... Loop

O ciclo **Do ... Loop** executa uma determinada tarefa enquanto a **condição utilizada** for **verdadeira**.



Exemplo: Construa uma macro que soma todos os valores até que o utilizador insira um 0, utilizando o ciclo Do Until ... Loop e o ciclo Do While ... Loop.

Do Until ... Loop	Do While ... Loop
Public Sub janelas_n() **Dim** soma, valor **As Integer** soma = 0 valor = **InputBox**("Insira um valor") **Do Until** valor = 0 soma = soma + valor valor = **InputBox**("Insira outro valor") **Loop** **MsgBox** (" A soma de todos os valores inseridos é " & soma) End Sub	Public Sub janelas_n() **Dim** soma, valor **As Integer** soma = 0 valor = **InputBox**("Insira um valor") **Do While** valor <> 0 soma = soma + valor valor = **InputBox**("Insira outro valor") **Loop** **MsgBox** (" A soma de todos os valores inseridos é " & soma) End Sub

Objetos e Métodos

Em VB todos os **componentes que são programáveis** são designados por **objetos**, enquanto que o **comportamento** de um **objeto** designa-se por **método**.

Sintaxe: ***Objeto*.Método**

Objetos	**Métodos**
Worksheets	Select
Range	Count
ActiveCell	Formula
Selection	Value
Workbooks	Offset
Cells	Clear
Rows	Activate

Sequências de ***Objetos*.Métodos** mais utilizadas:

- ✓ ***Worksheets***(“Sheet2”).**Activate** – Serve para ativar a folha de cálculo de um ficheiro Excel.
- ✓ ***Range***(“F2”).**Activate** – Serve para ativar os dados de uma célula (ou conjunto de células no caso de ser apresentado segundo a forma Range(“F2:F8”)).
- ✓ ***ActiveCell*.Offset**(i,0)**.Value** - Serve para aceder ao valor que está colocado numa determinada célula (i,0 - corresponde as posições)
- ✓ ***ActiveCell*.Offset**(i, 0).**Formula = "=SUM(B2:B8)"** – Serve para somar os valores de uma coluna, enquanto i cumprir uma determinada condição que o utilizador definiu.

  Nota: Esta sequência tanto serve para fazer somas como para outras funções. Para isso basta mudar a função SUM por outra qualquer.

- ✓ ***Selection*.Count** – Serve para contar uma determinada seleção anteriormente efetuada

- ✓ ***ActivateCell*.Offset**(i, 1).**Activate** – Move o conteúdo para a célula da coluna seguinte mantendo a mesma linha.
- ✓ n = ***ActiveCell***.**Value** – Serve para atribuir o valor de uma célula a uma variável.

	(-1,0)			
(0,-1)	**Célula Origem** (0,0)	(0,1)		
	(1,0)	(1,1)		(1,3)
	(2,0)			
(3,-1)	(3,0)		(3,2)	

Seta azul – O conteúdo move-se 3 linhas
Seta verde – O conteúdo move-se 3 linhas e 2 colunas para a direita
Seta roxa – O conteúdo move-se 3 linhas e 1 coluna para a esquerda

Exemplo 1: Construa uma macro que indica se os valores que estão inseridos nas células de G2 até G8, e no caso de serem maiores de 10 deve indicar na coluna ao lado a mensagem ">10X". Quando a macro terminar o seu trabalho esta deve devolver uma mensagem como "acabei".

```
Public Sub escreve_x()
   Worksheets("Sheet2").Activate
   Range("G2:G8").Activate
   Dim i As Integer

   For i = 0 To (Selection.Count - 1)
      If ActiveCell.Offset(i, 0).Value > 10 Then
         ActiveCell.Offset(i, 1).Value = " > 10 X"
      Else
         ActiveCell.Offset(i, 1).Value = ""
      End If
   Next i

MsgBox ("Já trabalhei tudo")
End Sub
```

Exercício 2: Construa uma macro que verifica os números inseridas nas células B2:B8, devolvendo na coluna C, se o valor é positivo ou não positivo. No célula B9 deve inserir a soma de todos os números inseridos de B2:B8.

```
Public Sub escreve_3a()
   Worksheets("Sheet2").Activate
   Range("B2:B8").Activate
   Dim i As Integer

   For i = 0 To Selection.Count - 1
      If ActiveCell.Offset(i, 0).Value > 0 Then
         ActiveCell.Offset(i, 1).Value = "Positivo"
      Else
         ActiveCell.Offset(i, 1).Value = "N_Positivo"
      End If
   Next i

   ActiveCell.Offset(i, 0).Formula = "=SUM(B2:B8)"
End Sub
```