



**Excel**

Funções definidas  
pelo utilizador

Catarina  
Oliveira

**DCT** DEPARTAMENTO CIÊNCIA  
E TECNOLOGIA

## CONTEÚDO

1. Funções definidas pelo utilizador
2. Editor de Visual Basic
3. Componentes mais comuns e importantes do Editor de Visual Basic
4. Módulos VBA
5. Linguagem VBA
  1. Estrutura de uma função
  2. Variáveis e tipos de dados
  3. Operadores e funções VBA
  4. Estruturas de controlo
    1. Estruturas Condicionais
    2. Estruturas repetitivas

## Funções definidas pelo utilizador

A criação de Funções Definidas pelo Utilizador (FDU):

- Permite reduzir a complexidade de uma folha de cálculo
  - Cálculos complexos poderão ser integrados numa função
    - Que depois é utilizada de forma mais simples (como as funções standard do Excel)
- Exige alguns conhecimentos da linguagem VBA (*Visual Basic for Applications*)
- Não podem ser gravadas como as macros de comandos

Função: programa ou algoritmo construído com o objetivo de produzir um resultado, executando um conjunto de instruções e utilizando um conjunto de argumentos fornecidos aquando da sua utilização.

Exemplo - função com o nome DOLARES, cuja finalidade é a de calcular o valor em Dólares (DOLARES) a partir do valor em Euros (EUROS) e da taxa de câmbio (TxEURUSD)

```
Function DOLARES(EUROS As Single, TxEURUSD As Single)
    DOLARES = EUROS * TxEURUSD
End Function
```

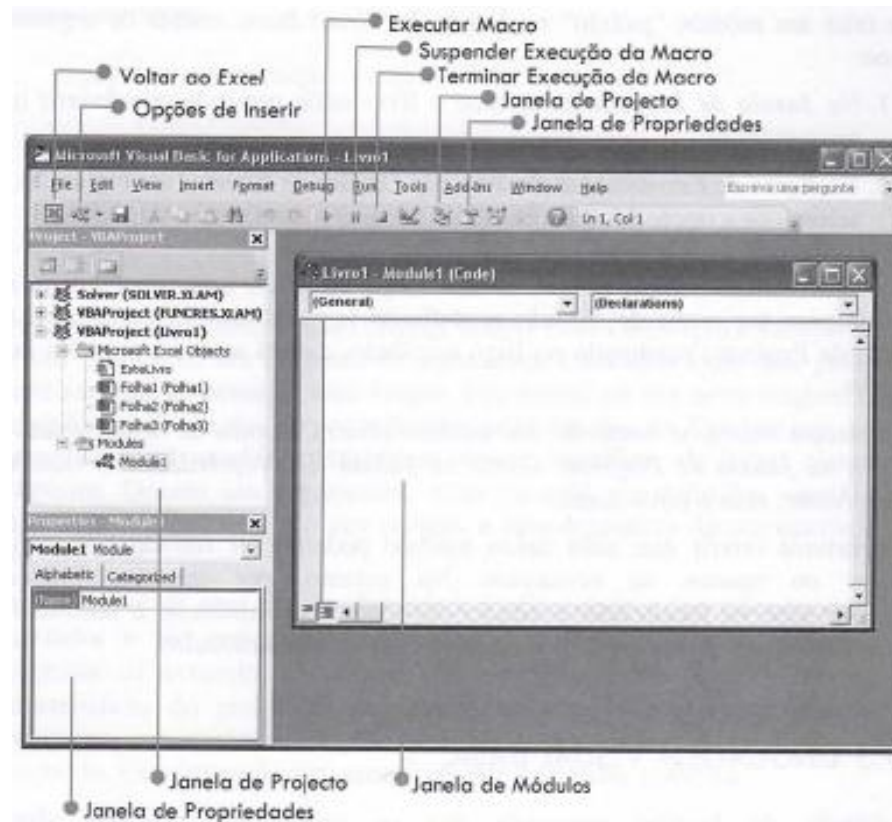
Pode ser chamada com:

- =DOLARES(1000;1,423)
- =DOLARES(B5;1,487)

## Editor de Visual Basic

Acesso ao editor:

- ALT + F11
- Separador “Programador” > Grupo “Código” > Comando “Visual Basic”



## Componentes mais comuns e importantes do Editor de Visual Basic

**Barra de ferramentas Padrão:** contém os comandos mais utilizados do editor, de onde se destacam:

- **Opções de Inserir:** decidir o tipo de objeto a inserir/criar no VBA, nomeadamente módulos (module), módulos de classe (*class module*), formulários (*form*) ou procedimentos (*procedure*)
- **Executar macro (Run):** executar e testar uma macro de comandos ou de funções
- **Suspender execução da macro (Break):** suspender/parar momentaneamente a execução de uma macro
- **Terminar execução da macro (Reset):** terminar definitivamente a execução de uma macro
- **Janela de Projeto:** essencial para a navegação entre os diferentes objetos dos projetos VBA, nomeadamente para localizar e abrir um determinado módulo; nesta janela são listados todos os livros abertos no Excel e, para cada um deles, os objetos neles contidos: folhas, módulos (*modules*) e formulários (*forms*), entre outros
- **Janela de Propriedades (properties):** apresenta uma lista de propriedades e respetivos valores do objeto do projeto selecionado
- **Janelas de módulos:** local do editor onde é introduzido e editado o código VBA das macros de comandos e de funções. Para cada um dos módulos criados de novo ou abertos o editor disponibiliza uma janela própria

## Módulos VBA

O Editor de Visual Basic permite a criação de dois tipos de módulos:

- Módulos “padrão” (*module*) ← mais simples e mais frequentemente utilizados
- Módulos de classe (*class module*)

Procedimento (criação de um módulo “padrão”):

1. Na “Janela de Projeto” selecionar o livro onde se pretende criar/inserir um novo módulo
2. Na “Barra de Ferramentas Padrão”, nas “Opções de Inserir” selecionar a opção “Module”

Deverá surgir no editor uma nova “Janela de Módulo” com um nome de módulo predefinido (ex: Module1, Module2, ...) e na “Janela de Projeto”, “Pendurado” no livro escolhido, deverá aparecer também esse módulo.

Para alterar o nome predefinido: selecionar o módulo, alterar já “Janela de Propriedades” o valor do campo “Nome”

Num único módulo podem ser inseridas várias funções ou macros de comandos

Para simplificação, aconselha-se a distribuir as várias funções/macros por módulos diferentes

## Linguagem VBA: Estrutura de uma função

**Function** NOMEFUNÇÃO(argumento1, argumento2, ...)

[Instruções]

NOMEFUNÇÃO = [Expressão ou variável]

[Instruções]

**End Function**

Nome da função:

- Deve ser original (não se pode repetir os nomes das funções do Excel ou de outra função criada)
- Deve identificar claramente o propósito da função

Argumentos

- Devem ser definidos entre dois parêntesis obrigatórios
- Número e tipo de argumentos dependem das características da função a criar

Instruções

- Dependem da finalidade e dos cálculos que serão realizados
- São compostas por variáveis, expressões, valores, operadores e estruturas de controlo
- Número de instruções depende das características do problema
- **Instrução especial obrigatória:** NOMEFUNÇÃO = [Expressão ou variável]
  - Define o valor que será devolvido

## Linguagem VBA: Variáveis e tipos de dados

Variável: local de memória ao qual é atribuído um nome e onde são temporariamente guardados os valores utilizados no processamento das funções (e das macros em geral)

Cada variável tem um nome único e um tipo de dados que pode estar definido

Tipo de dados	Significado: gama de valores possíveis
<i>Byte</i>	Inteiro muito pequeno: 0 a 255.
<i>Integer</i>	Inteiro pequeno: -32.768 a 32.767.
<i>Long</i>	Inteiro grande: -2,147,483,648 a 2,147,483,647.
<i>Single</i>	Real de precisão decimal simples: Valores negativos: -3.402823E38 a -1.401298E-45. Valores positivos: 1.401298E-45 a 3.402823E38.
<i>Double</i>	Real de precisão decimal dupla: Valores negativos: -1.79769313486231E308 a -4.94065645841247E-324. Valores positivos: 4.94065645841247E-324 a 1.79769313486232E308.
<i>Currency</i>	Moeda: -922,337,203,685,477.5808 a 922,337,203,685,477.5807.
<i>Decimal</i>	Real de grande precisão.
<i>Boolean</i>	Valor lógico: <i>TRUE</i> ou <i>FALSE</i> .
<i>Date</i>	Data: 1 Janeiro 1901 a 31 Dezembro 9999.
<i>String</i>	Texto: 1 a 65400 caracteres.
<i>Object</i>	Qualquer objecto VBA.
<i>Variant</i>	Tipo de dados genérico.



## Linguagem VBA: Variáveis e tipos de dados

Nas funções, as variáveis poderão ser utilizadas em diferentes contextos:

- Para representar e guardar os valores dos argumentos
- Para guardar valores de cálculos intermédios
- Para guardar o resultado da própria função

Declaração de variáveis

- Permite associá-las a um nome e a um tipo de dados (contextualiza e condiciona a utilização)
- Nas funções, a forma como são declaradas depende do tipo de utilização:
  - Argumentos: declaração na sintaxe da própria função com a instrução `NomeVariável As TipoDados`
  - Restantes variáveis: sintaxe `Dim NomeVariável As TipoDados`

```
Function TESTE(NUM1 As Integer, NUM2 As Integer)
```

```
    Dim DIVISÃO As Single
```

```
    DIVISÃO = NUM1 / NUM2
```

```
    TESTE = DIVISÃO * 100
```

```
End Function
```

Função **TESTE** (devolver o resultado de uma divisão multiplicado por 100)

- Variáveis **NUM1** e **NUM2**: dois argumentos do tipo inteiro
- Variável **DIVISÃO**: variável auxiliar utilizada para guardar o resultado de um calculo parcial
- Variável **TESTE**: variável com o nome da função, que permite devolver o resultado desejado

## Linguagem VBA: Operadores e funções VBA

Categoria de operadores	Operadores
Atribuição	=
Aritméticos	+, -, *, /, ^ MOD (resto da divisão)
Lógicos	AND (conjunção) OR (disjunção) NOT (negação) XOR (exclusão) EQV (equivalência) IMP (implicação)
Comparação	=, >, <, >=, <= LIKE (igualdade de cadeias de caracteres)
Concatenação	&

Função/Sintaxe	Descrição/Finalidade
<i>Sqr</i> (número)	Calcula a raiz quadrada de um número.
<i>Date</i>	Devolve a data actual.
<i>Year</i> (data)	Devolve o ano de uma data.
<i>Round</i> (valor, casasdecimais)	Arredonda um número ou expressão para <i>n</i> casas decimais.

## Linguagem VBA: Estruturas de controle

Em programação, de forma a controlar o fluxo de execução das instruções num algoritmo, são normalmente utilizados três tipos de estruturas:

- **Estruturas sequenciais:** tipo mais básico de estrutura. Traduz-se na execução sequencial de um conjunto de instruções, considerando apenas a sua ordem. Exemplo: as instruções de uma macro gravada
- **Estruturas condicionais** (de decisão ou de seleção): as instruções são agrupadas e executadas em função do resultado de uma ou mais condições
- **Estruturas repetitivas:** permitem repetir a execução de um conjunto de instruções enquanto determinadas condições se verificarem

## Estruturas Condicionais

```
If condição Then
    [instruções se condição verdadeira]
Else
    [instruções se condição falsa]
End If
```

```
If x > 0 Then
    resultado = "Positivo"
Else
    resultado = "Negativo ou zero"
End If
```

```
Select Case expressão
Case valor1
    [instruções]
Case valor2
    [instruções]
Case else
    [instruções]
End Select
```

```
Select Case dia
Case 1
    resultado = "Domingo"
Case 2
    resultado = "Segunda feira"
...
Case else
    resultado = "Dia inválido"
End Select
```

## Estruturas repetitivas

```
For contador = valorInicial To valorFinal  
    [instruções]  
Next [contador]
```

```
Do While condição  
    [instruções]  
Loop
```

```
Do  
    [instruções]  
Loop While condição
```

```
For cont = 1 To 3  
    resultado = resultado + cont  
Next cont
```

```
Dim cont As Integer  
Cont = 1  
Do While cont <= 3  
    resultado = resultado + cont  
    cont = cont + 1  
Loop
```

```
Dim cont As Integer  
Cont = 1  
Do  
    resultado = resultado + cont  
    cont = cont + 1  
Loop While cont <= 3
```



UNIVERSIDADE  
PORTUCALENSE

Do conhecimento à prática.