

Mestrado Decisão Económica e Empresarial COMPUTAÇÃO

Programação em VBA.

Conceitos básicos. Macros. Variáveis e tipos de dados.

Excel/VBA

- Excel é uma folha de cálculo, uma **aplicação**, que permite criar e manusear qualquer tipo de lista com funcionalidades das mais simples às mais avançadas. Com a linguagem Visual Basic aumentam-se as potencialidades desta folha de cálculo.
- Microsoft Visual Basic for Applications (VBA) é uma linguagem de programação baseada no Microsoft Visual Basic. Permite escrever código para executar acções num documento e/ou no seu conteúdo.
- Microsoft Visual Basic é um ambiente de programação que vem com a instalação do Excel. Para o tornar facilmente acessível:

VBA acessível no Excel (2007)

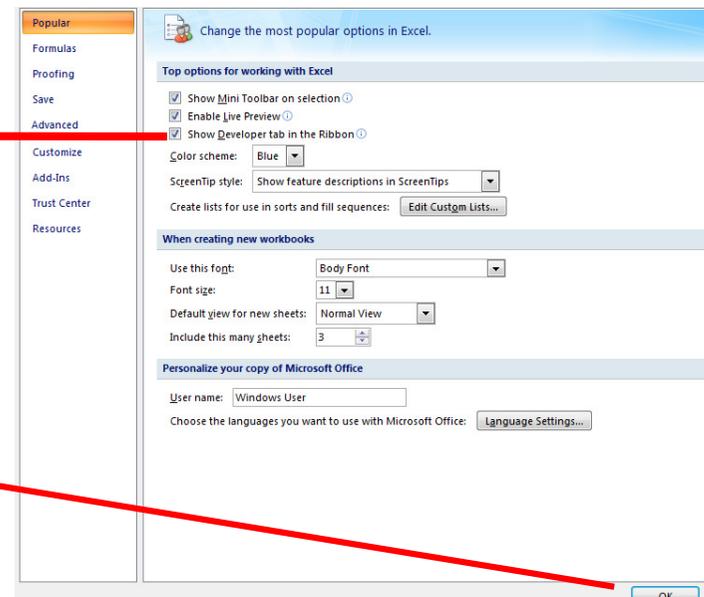
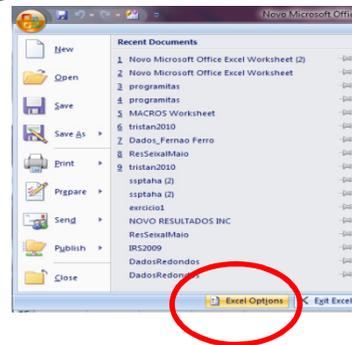


1. Botão do *Office*

2. *Excel options*

3. *Show developer tab in the ribbon*

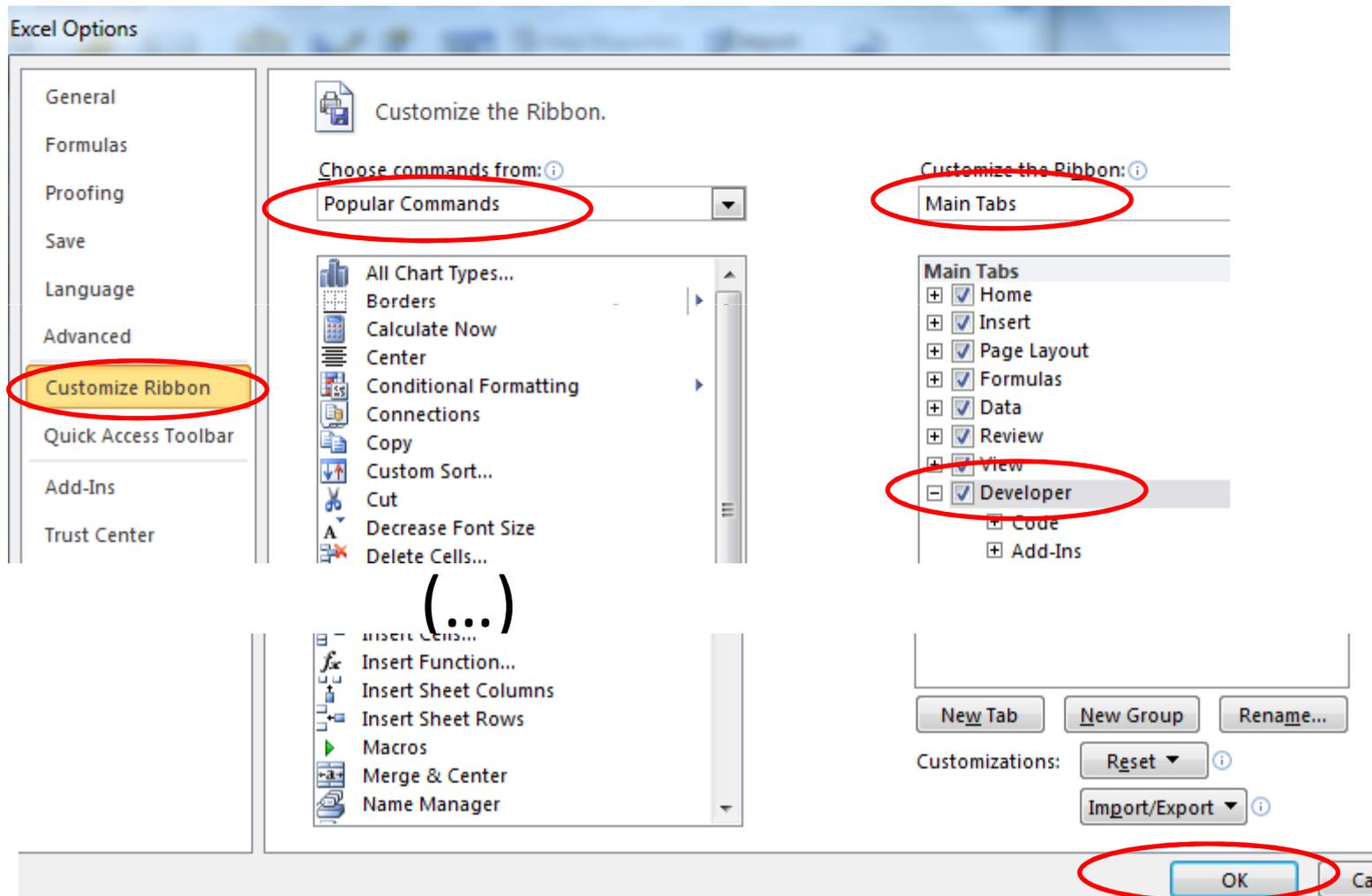
-> *OK*



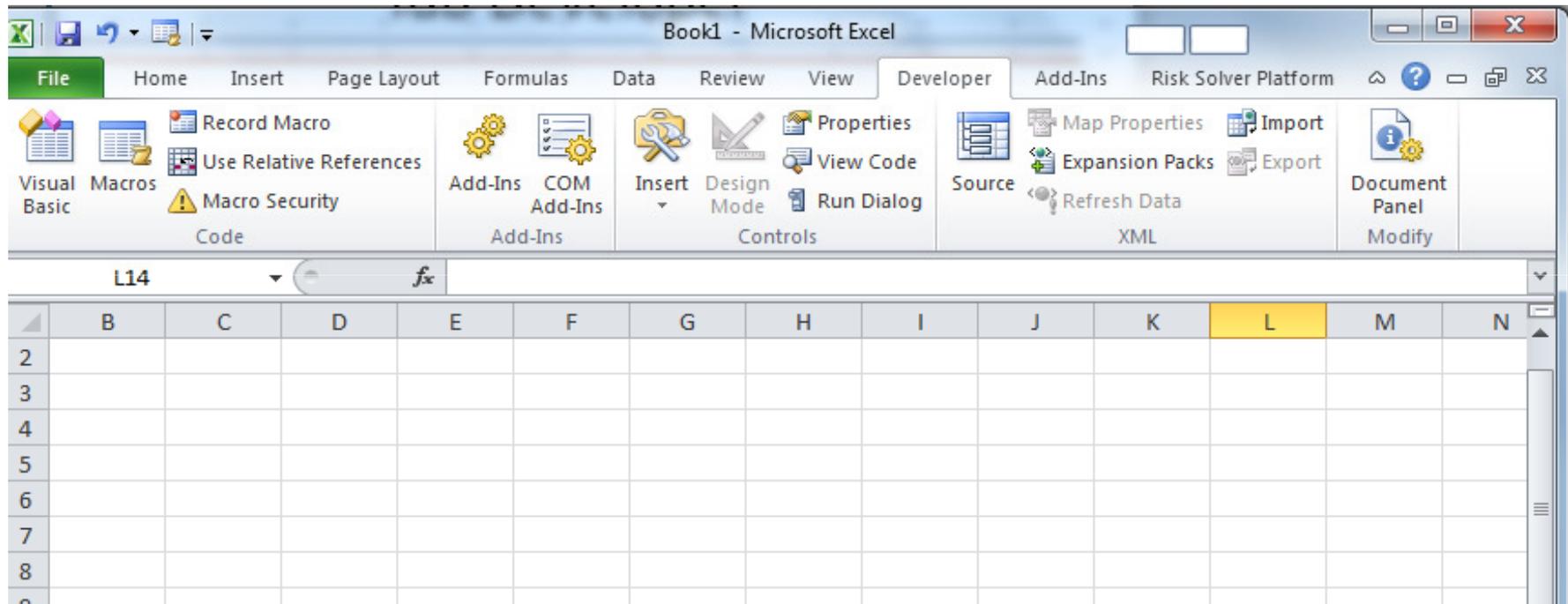
VBA acessivel no Excel (2010)

1. File -> Options

2.



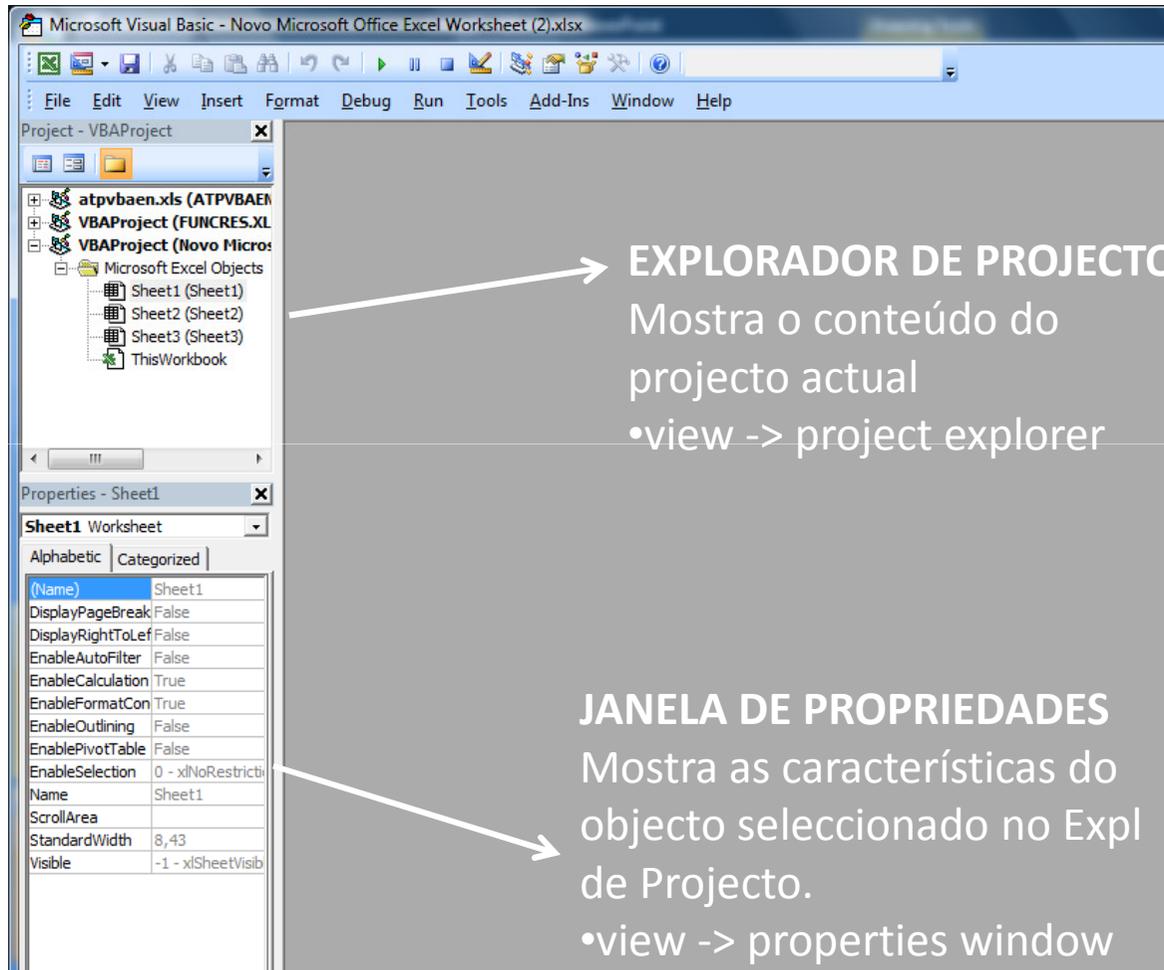
Tab Developer



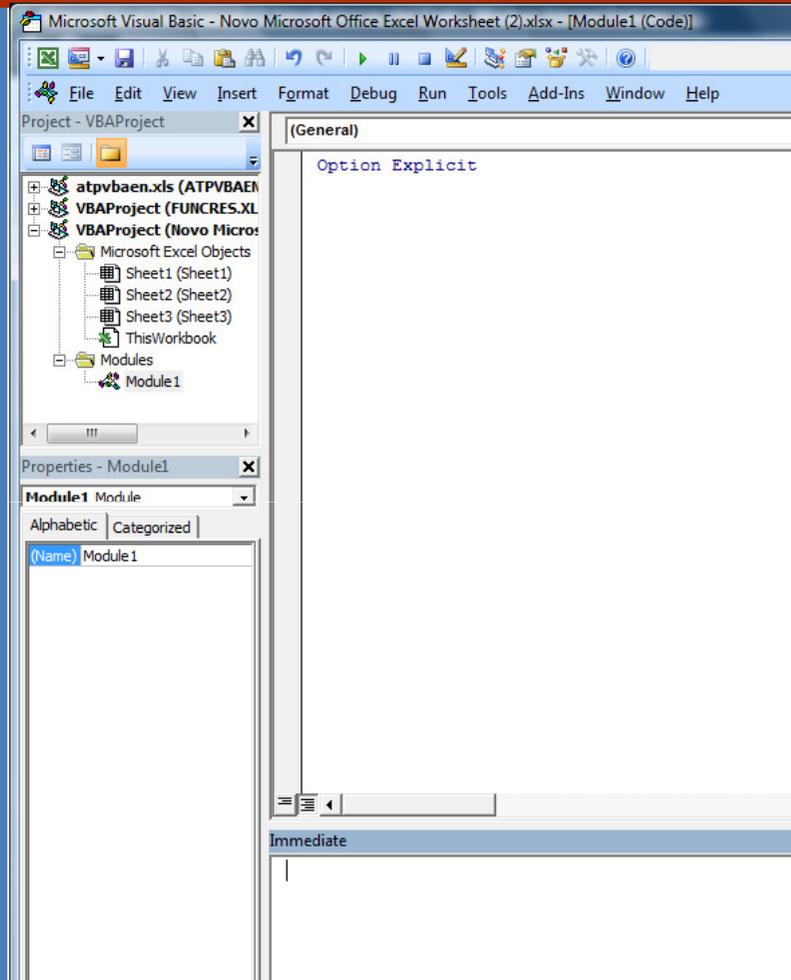
O editor do VBA

- IDE (Integrated Development Environment)
- Para aceder ao IDE partir do Excel
 - (ALT+F11) ou
 - Developer -> 
- Para voltar à folha de cálculo
 - (ALT+F11) (novamente)

O editor do VBA



O editor do VBA



Macro

É um programa escrito em VBA, que indica a uma aplicação como o Excel quais os passos a dar para atingir um objectivo específico, dito de outro modo, é a descrição formalizada das tarefas que se pretende automatizar.

- Servem para automatizar tarefas repetitivas.

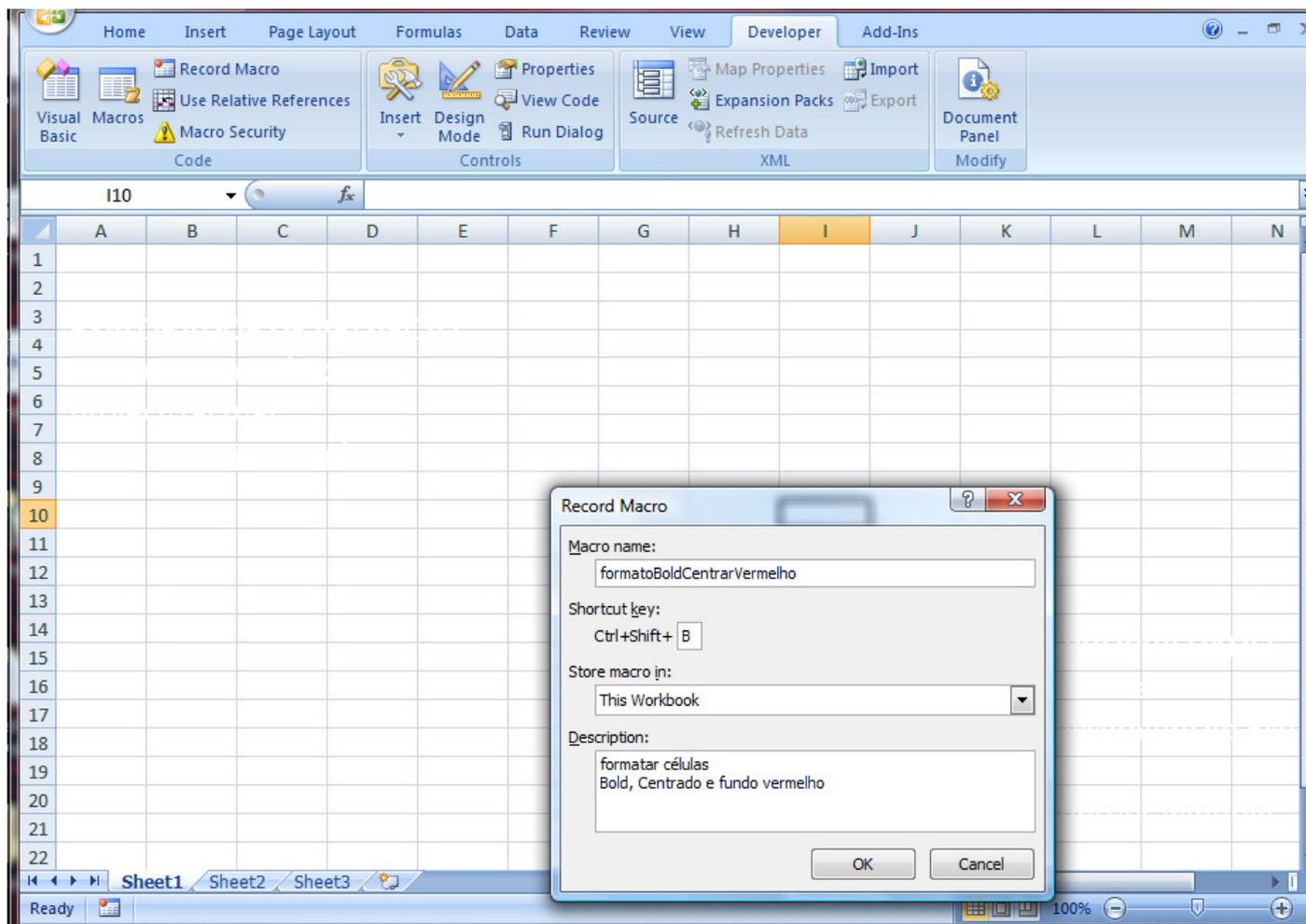
Criar um Macro

1. Abrir Excel
2. Developer -> Record a Macro
3. Preencher
 - nome,
 - escolher tecla para atalho e
 - escrever descrição
- OK
4. Executar as acções pretendidas
5. Developer -> Stop Record a Macro

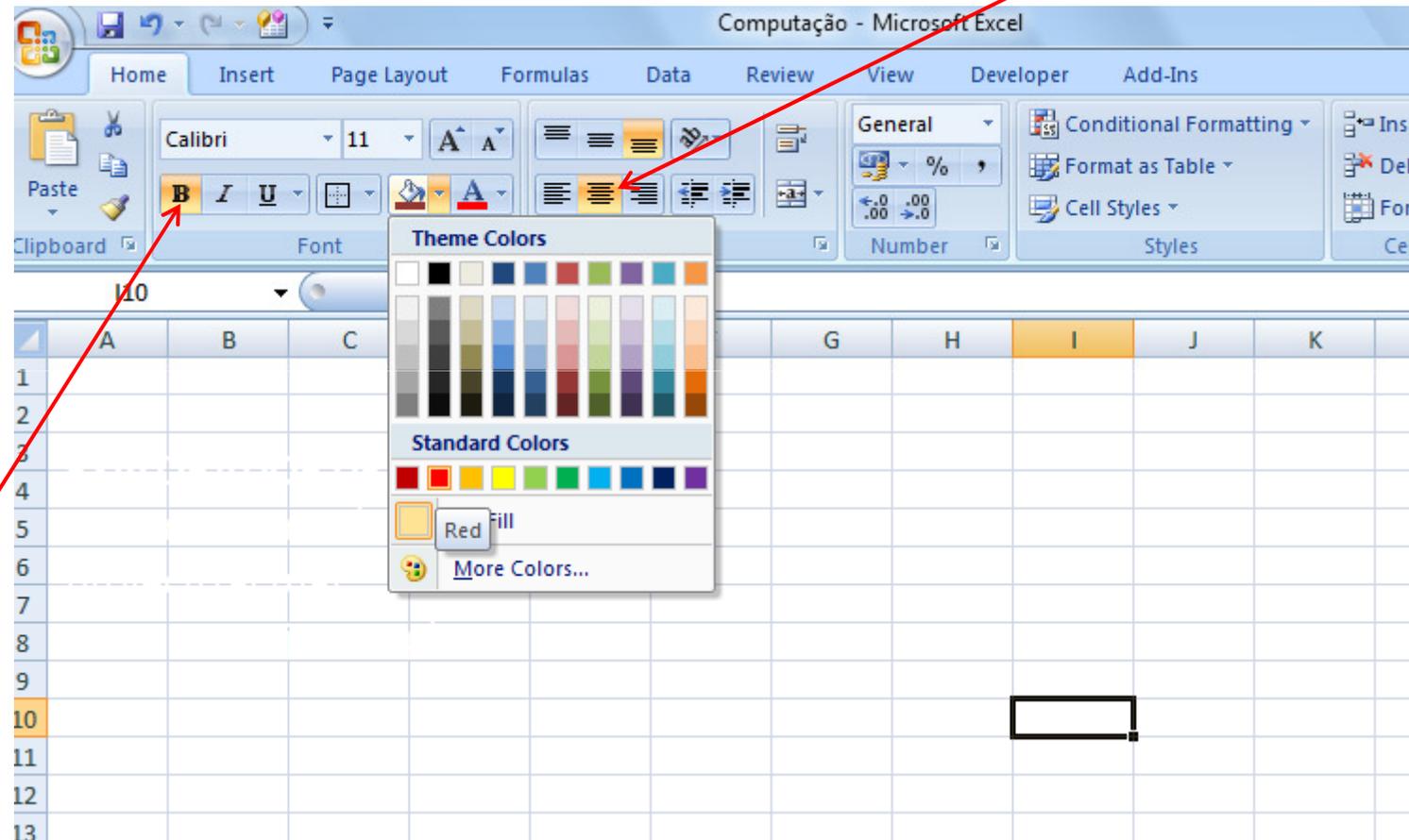
EXEMPLO

- Associar a (Ctrl+Shift+B)
- a seguinte formatação
 - Bold
 - Centrado
 - Fundo Vermelho

EXEMPLO Criar um Macro (2) (3)



EXEMPLO Criar um Macro (4)



Visualizar os Macros

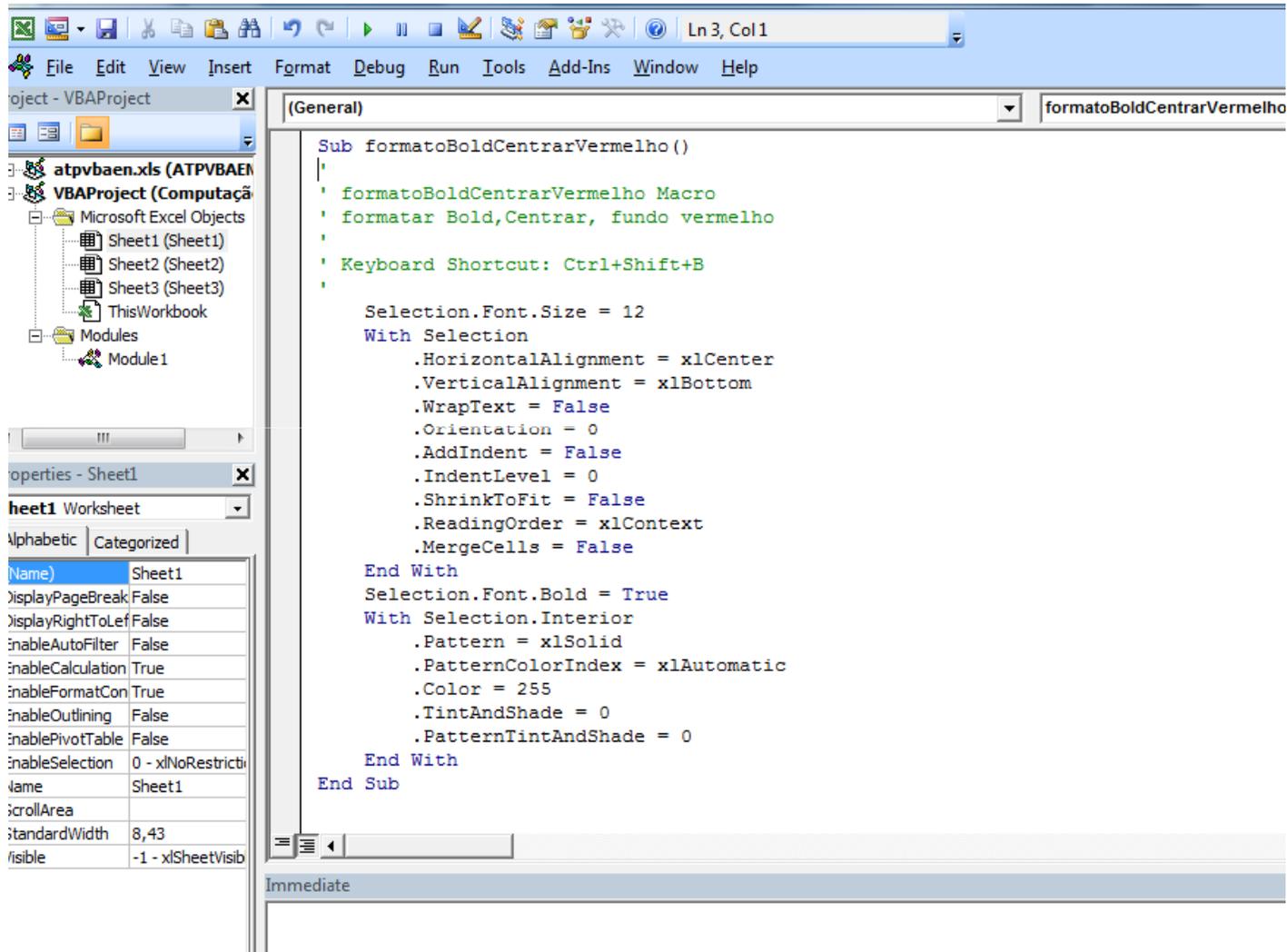
The image shows a screenshot of the Microsoft Excel VBA environment. The main window displays the Excel spreadsheet with the Developer tab selected in the ribbon. The 'View Code' button is highlighted with a red circle. A red arrow points to the 'Macro' dialog box, which is open and shows a list of macros. The 'Macro name' list contains three entries: 'formatoBoldCentrarVermelho', 'formatarItalicoCentradoLazul', and 'formatoBoldCentrarVermelho'. The 'Macro in' dropdown is set to 'All Open Workbooks'. The 'Description' field shows 'formatar Bold, Centrar, fundo vermelho'. The 'Run' button is highlighted.

• Click View. button and

• Click Develop

Any of these ad the current doc

Editar o código (VBA) de um Macro



The screenshot displays the VBA Editor interface for a macro named 'formatoBoldCentrarVermelho'. The editor is open to the 'General' tab, and the code is as follows:

```
Sub formatoBoldCentrarVermelho ()  
'  
' formatoBoldCentrarVermelho Macro  
' formatar Bold, Centrar, fundo vermelho  
'  
' Keyboard Shortcut: Ctrl+Shift+B  
'  
    Selection.Font.Size = 12  
    With Selection  
        .HorizontalAlignment = xlCenter  
        .VerticalAlignment = xlBottom  
        .WrapText = False  
        .Orientation = 0  
        .AddIndent = False  
        .IndentLevel = 0  
        .ShrinkToFit = False  
        .ReadingOrder = xlContext  
        .MergeCells = False  
    End With  
    Selection.Font.Bold = True  
    With Selection.Interior  
        .Pattern = xlSolid  
        .PatternColorIndex = xlAutomatic  
        .Color = 255  
        .TintAndShade = 0  
        .PatternTintAndShade = 0  
    End With  
End Sub
```

The interface also shows the Project Explorer on the left, listing the 'atpvbaen.xls (ATPVBAEN)' project with 'VBAProject (Computação)' containing 'Microsoft Excel Objects' (Sheet1, Sheet2, Sheet3, ThisWorkbook) and 'Modules' (Module1). The Properties window at the bottom left shows the properties for 'Sheet1', including 'Name', 'ScrollArea', and 'Visible'.

Executar um Macro

- Usando o atalho definido

ou,

- Developer -> Macros
 - Seleccionar
 - RUN

Escrever código legível

- Indentação
 - Deve permitir perceber facilmente a estrutura do código;
 - Usar os *tabs*
conselho predefinir *tab* a 4
(Menu VB -> tools -> options ...)
- Comentários
 - Devem ser inseridos para facilitar a compreensão do código
 - Uma linha que comece por ' (plica) ou Rem é ignorada pelo compilador

Variáveis

Localização de memória em que a informação pode ser guardada de modo a ser usada por um macro.

Caracterizada por

- Nome
- Tipo de dados que pode armazenar e
- Âmbito

Variáveis - declaração

Chama-se declaração à criação de uma variável.

Atribui-se-lhe: nome, tipo de dados e fixa-se o seu âmbito.

Antes de utilizar uma variável esta deve ser declarada.

Dim variável **As** tipo de dado

Variável - nome

Regras:

- Deve começar com uma letra ou com *underscore*
- Não pode contêr ponto;
- Não pode exceder 255 caracteres.
- Deve ser único na área que é usado
- Não pode coincidir com palavras reservadas

Variável - não pode ter os nomes:

And (Bitwise) And (Condition) As Boolean ByRef Byte
ByVal Call Case CBool CByte CDate
Cdbl CInt CLng Const CSng CStr
Date Dim Do Double Each Else
Elseif End EndIf Error False For
Function Get GoTo If Integer Let
Lib Long Loop Me Mid Mod
New Next Not Nothing Option Or (Bitwise)
Or (Condition) Private Public ReDim REM Resume
Select Set Single Static Step String
Sub Then To True Until vbCrLf
vbTab With While Xor

Option explicit

- Existe a possibilidade de não declarar explicitamente uma variável. Nesta caso a primeira vez que o nome é usado a variável é implicitamente declarada.

NÃO É SEGURO por isso convem trabalhar com a opção **Option explicit**

- Escrever no início do código
- Developer -> VB -> tool -> options -> require variable declaration

Variável – atribuição de valores

Sub exemplo()

Dim valor as Integer

valor = 5

End Sub

= Afectação

Tipos de dados

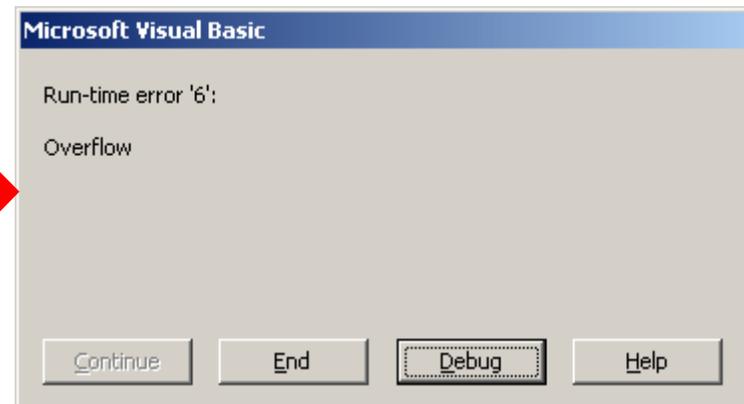
- Byte
 - Integer
 - Long
 - Single
 - Double
 - String
 - Currency
 - Date
 - Variant
- inteiros
- Virgula flutuante

Tipo Byte

Números naturais entre 0 e 255

```
Sub ExemploByte()  
    Dim Idade As Byte  
End Sub
```

```
Sub ErroByte()  
    Dim Valor As Byte  
    Valor=777  
End Sub
```



Tipo Integer

- Inteiros entre -32768 e 32767

```
Sub ExemploInt ()
```

```
    Dim Nalunos As Integer
```

```
End Sub
```

```
Sub ExemploInt ()
```

```
    Dim Nalunos%
```

```
End Sub
```

Tipo Long

Inteiros entre – 2,147,483,648 e 2,147,483,647

```
Sub ExemploLong()  
    Dim Populacao As Long  
End Sub
```

```
Sub ExemploLong()  
    Dim Populacao@  
End Sub
```

Tipo Single

Negativos de $1.401298e^{-45}$ a $3.402823e^{38}$

Positivos de $1.401298e^{-45}$ a $3.402823e^{38}$

```
Sub ExemploSingle()  
    Dim Distancia As Single  
End Sub
```

```
Sub ExemploSingle()  
    Dim Distancia!  
End Sub
```

Tipo Double

Negativos de $-1.79769313486231e^{308}$ a
 $-4.94065645841247e^{-324}$

Positivos de $4.94065645841247e^{-324}$ a
 $1.79769313486231e^{308}$

```
Sub ExemploDouble()  
    Dim Distancia As Double  
End Sub
```

```
Sub ExemploDouble()  
    Dim Distancia#  
End Sub
```

Tipo String

- É um caracter ou um conjunto de caracteres (texto)

```
Sub ExemploString()  
    Dim Nome As String  
End Sub
```

```
Sub ExemploString()  
    Dim Nome$  
End Sub
```

Tipo Currency

- Entre -922,337,203,685,477.5808 e 922,337,203,685,477.5807.
 - As vírgulas são para facilitar a leitura
 - Não colocar € ou outro símbolo

```
Sub ExemploCurrency ()  
    Dim Salario As Currency  
End Sub
```

Tipo Date

```
Sub ExemploDate()  
    Dim DataNascimento As Date  
    DataNascimento = #1/10/2010#  
End Sub
```

Tipo Variant

```
Sub ExemploVaR()  
  Dim Nome As Variant  
  Dim Contracto As Variant  
  Dim SalarioHora As Variant  
  Dim InicioContr As Variant  
  
  Nome = "Manuel Joaquim"  
  Contracto = 0  
  SalarioHora = 10  
  InicioContr = #1/1/1980#  
End Sub
```

Não é boa prática
Exige reserva de
muito espaço

Âmbito de uma variável

- Tempo de vida de uma variável

Globais/locais

Variável Global

```
Option Explicit
```

```
Dim NomeF As String
```

```
Sub ExemploGlobalLocal()
```

```
    Dim NomeP As String
```

```
    NomeP = "Maria"
```

```
    NomeF = "Silva"
```

```
End Sub
```

Variável Local

Variável -Private/Public

```
Private NomeF As String
```

```
Sub PrivPub()
```

```
    Dim NomeP As String
```

```
        NomeP = "Maria"
```

```
        NomeF = "Silva"
```

```
End Sub
```

**Só pode ser acedida
por código do
mesmo módulo**

Variável -Private/Public

```
Option Explicit
```

```
Private NomeF As String
```

```
Public NomeC As String
```

```
Sub ExemploPrivPubOK()
```

```
    Dim NomeP As String
```

```
        NomeP = "Maria"
```

```
        NomeF = "Silva"
```

```
        NomeC = NomeP & " " & NomeF
```

```
End Sub
```

**Pode ser acessada
por código do
mesmo módulo ou
fora dele**

Variável -Private/Public

```
Option Explicit
```

```
Private NomeC As String
```

```
Option Explicit
```

```
Private NomeF As String
```

```
Private NomeP As String
```

```
Sub ExemploPrivPubKO()
```

```
Dim NomeP As String
```

```
NomeP = "Maria"
```

```
NomeF = "Silva"
```

```
NomeC = NomeP & " " & NomeF
```

```
End Sub
```

