# Excel

## Macros

Catarina
Oliveira

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

UPT UNIVERSIDADE PORTUCALENSE

# CONTENT

1. Macros
2. Activating the Developer tab
3. Macros’ security
4. Record and execute macros
5. Relative vs. absolute references
6. Macros’ execution
    1. Macro dialog
    2. Shortcut key
    3. Use a button or object
7. Macro editing
8. Macro deletion

# Macros

Allow automation of tasks that are repeated on a spreadsheet with some frequency

Set of sequentially executed steps that, as a whole, constitute and perform a certain task

Can also:

- Facilitate and accelerate the execution of tasks
- Ensure that certain tasks are always performed in the same way, avoiding errors and/or inconsistencies

Developed using VBA (*Visual Basic for Applications*) code, which can be generated in two ways:

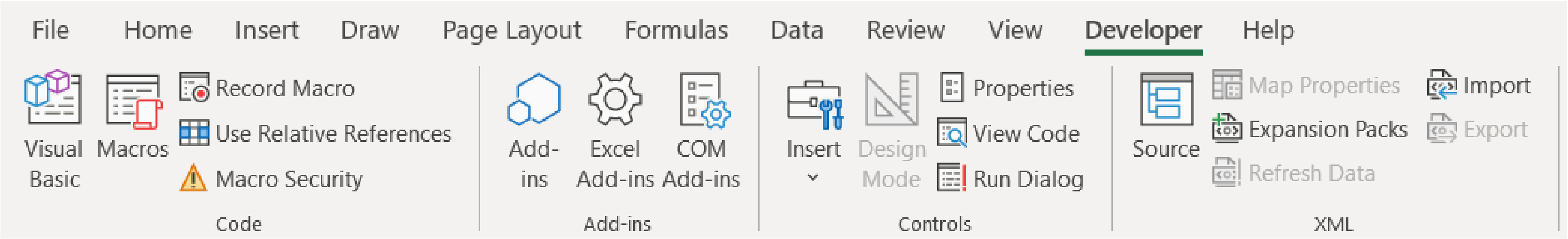
- Using **Record Macro** that automatically translates the user actions into VBA code
    - Advantage: the simplest way, does not require knowledge of VBA language
    - Limitation: only allows automation of tasks whose steps can be fully performed manually
- Writing VBA code directly into the **Visual Basic Editor** (next Slide)
    - Although limited to the user's experience and knowledge of programming techniques and the VBA language itself
        - Allows creation of any type of macro, simple or complex

Using VBA, we can create two types of macros:

- **Command macros:** programs that sequentially execute a set of excel commands
- **Function macros:** programs that perform an operation and return a result.

# Activating the Developer tab

1. File > Options
2. Dialog “Excel Options” > Select “Customise Ribbon”
3. Select “Developer”
4. OK



# Macros’ security

To be able to create macros, the Excel file must be saved with the extension .xlsm (or .xlsb) instead of .xlsx

To consult/change macro definitions in Excel:

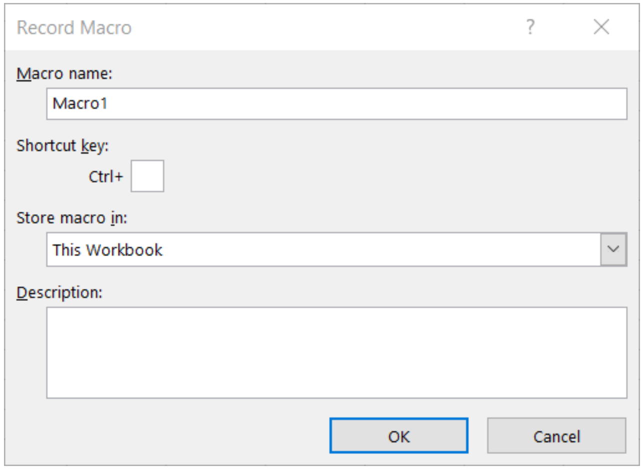
1. Tab “Developer” > Group “Code” > Command “Macro Security”
2. On the dialog “Trust Centre”
    1. Select “Macro Settings”
    2. In “Macro settings” select one of the following options
        - **Disable all macros without notification:** Disables all macro security warnings and automatically prevents any type of macro from running
        - **Disable runs macros with notification:** disables the macros, but keeps the security warning that will allow you to manually enable these macros (the safest)
        - **Disable all macros except digitally signed:** Enables macros digitally signed by a trusted publisher and already accepts and disables the rest. If the macro maker has not yet been trusted, a prompt will appear to do so.
        - **Enable all macros:** Enables all macros in the workbook, however this carries the risk of allowing macros whose code could be malicious to the system to run
    3. In section “Developer Macro Settings” we may allow Trusted Access to the VBA project object model by programmers
    4. OK

# Record and execute macros

Before starting: “train” the command to automate so that unnecessary steps are not also recorded

**How to (record)**:

1. Tab “Developer” > Group “Code” > Command “Record Macro”
2. Complete the dialog
    - **Macro name:** define the name by which the macro will be identified and with which the VBA code will be generated
    - **Shortcut key:** identify the key that can be used as a shortcut when pressed simultaneously with the CTRL key (case sensitive)
    - **Save macro in:** identify the location where the macro will be generated and saved. By default it will be in the book where it is generated (In This Book), but it can be saved in another book
    - **Description:** additional information about the macro
3. Execute the commands to record
4. Tab “Developer” > Group “Code” > Command “Stop Recording”



Record Macro

Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

Description:

**How to (execute)**: Once recorded, it can be performed in 3 ways:

- Through the "Macro" dialog box (slide 8)
- Using the shortcut key associated with the macro (slide 9)
- Using a button or an object (slide 10)

# Relative vs. absolute references

There is a difference between recording macros with relative references and with absolute references :

- **With absolute references:** commands are executed on the active sheet, in exactly the same cells where they were recorded
- **With relative references:** when a macro starts to be recorded, Excel saves the Active Cell as the initial reference (position) and from then on, all offsets in the spreadsheet are saved with their relative addresses. When running a macro with relative references, the commands are applied depending on the currently active cell

Defining record type:

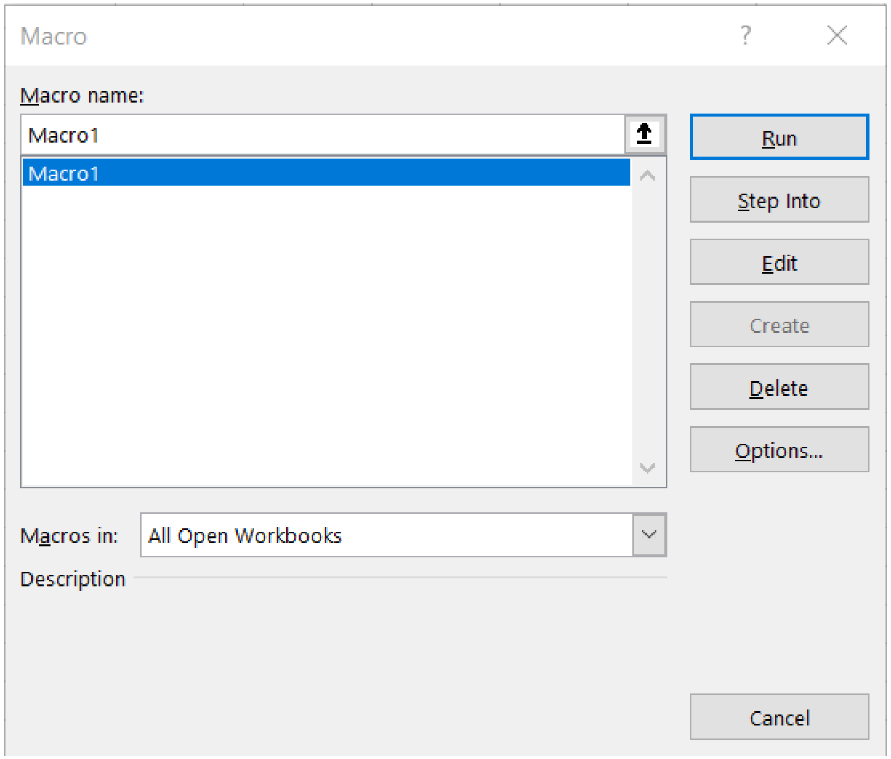
Tab “Developer” > Group “Code” > Command “Use relative references”

If the command is selected, recording is done with relative references. If not, it is done with absolute references

# Macro execution: Macro dialog

**How to**:

1. Tab “Developer” > Group “Code” > Command “Macros”
2. On the dialog:
   1. Select the name of the macro to run
   2. Click “Run”



Macro

Macro name: Macro1

Macro1

Macros in: All Open Workbooks

Description

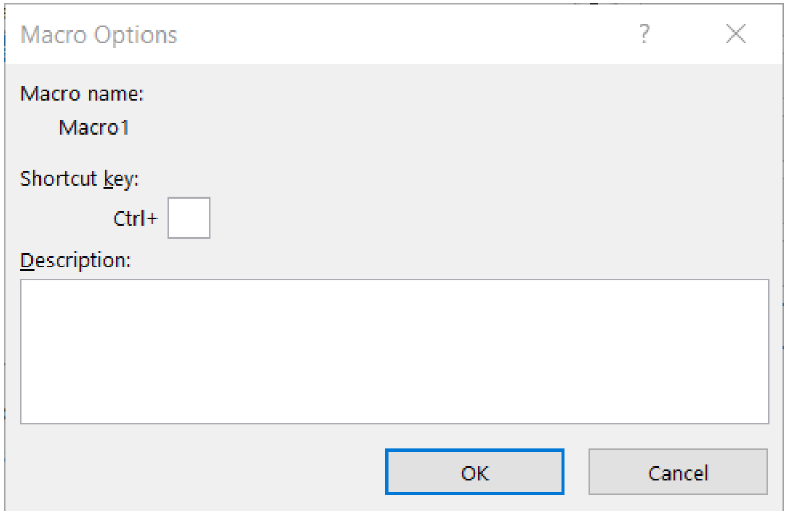
# Macro execution: shortcut key

It can be used if a shortcut key has been defined in the macro definition.

Alternatively, it can be defined later.

**How to add shortcut key:**

1. Tab “Developer” > Group “Code” > Command “Macros”
2. On the Macro dialog
    1. Select the macro
    2. Click “Options”
    3. On the “Macro Options” dialog:
        1. Insert the key in “Shortcut key”
        2. Ok
    4. Ok



Macro name:

Macro1

Shortcut key:

Ctrl+

Description:

How to (run):

If lowercase letter is defined, press Ctrl and shortcut key simultaneously

If capital letter is defined, press Ctrl, Shift and shortcut key simultaneously

# Macro execution: using a button or object

First you need to define the button/object

Excel allows the creation of 2 types of buttons :

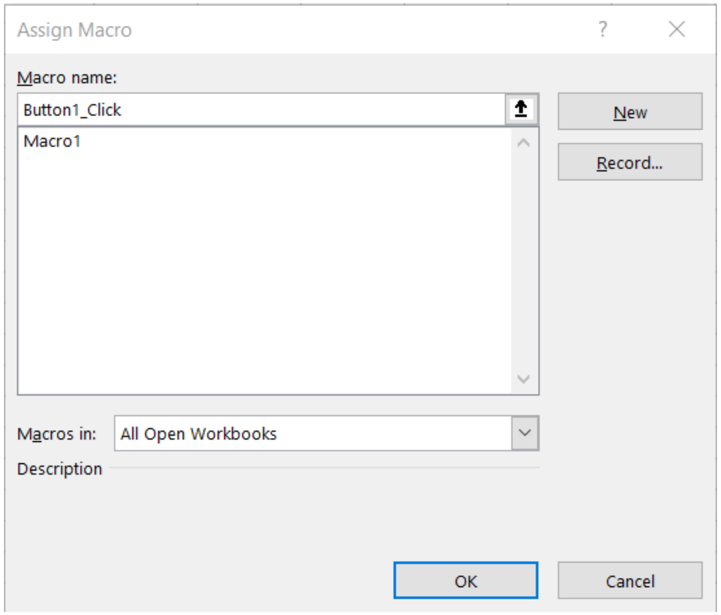
- Form control: with only one event type (mouse click)
- ActiveX controls: various types of events (eg one click, double click, move mouse, …)

**How to (insert button)**:

1. Tab “Developer” > Group “Controls” > Commando “Insert” > Option “Button”
2. Draw the button
3. On the dialog “Assign Macro”
    1. Select the name of the macro
    2. Ok
4. Double click the button to change its text

**How to (run macro)**:

1. Click the button


Assign Macro

Macro name: Button1_Click

Macro1

Macros in: All Open Workbooks

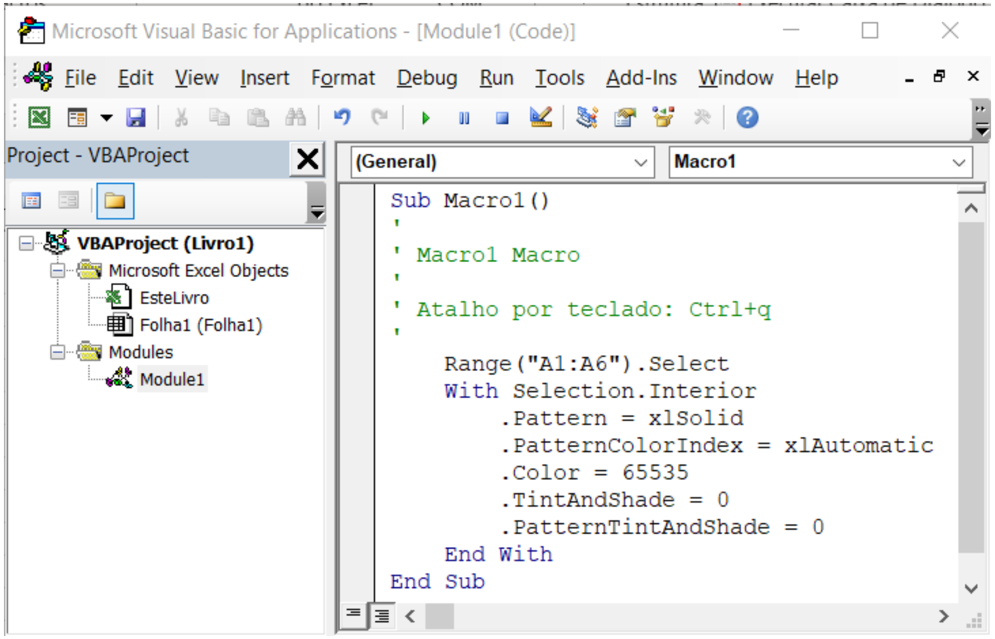
Description

# Macro editing

After recording a macro, the VBA code automatically generated by Excel can be consulted and edited in one of the modules in the Visual Basic Editor

**How to**:

1. Tab “Developer” > Group “Code” > Command “Macros”
2. On the “Macro” dialog select the macro to edit and click “Edit”


Microsoft Visual Basic for Applications - [Module1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - VBAProject
VBAProject (Livro1)
Microsoft Excel Objects
EsteLivro
Folha1 (Folha1)
Modules
Module1
(General)
Macro1

```
Sub Macro1()
'
' Macro1 Macro
'
' Atalho por teclado: Ctrl+q
'
    Range("A1:A6").Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
End Sub
```

# Macro deleting

**How to**:

1. Tab “Developer” > Group “Code” > Command “Macros”
2. On the “Macro” dialog select the macro to delete and click “Delete”

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.