

Curso de Macro em Excel

1) Conceitos de Sub, Function e declaração de variáveis.

1.1)Integer

```
Sub declarandoInteger()  
  
    Dim x As Integer  
  
    x = 1  
  
    MsgBox "o valor é " & x  
  
End Sub
```

1.2)Double

```
Sub declarandoInteger()  
  
    Dim x As Double  
  
    x = 1.1 'Perceba que deve ser colocado ponto e não vírgula  
  
End sub
```

1.3)String

```
Sub declarandoString()  
  
    Dim localDeTrabalho As String  
  
    localDeTrabalho = "STJ"  
  
    MsgBox "Trabalho no " & localDeTrabalho  
  
End Sub
```

1.4)Boolean

```
Sub DeclaraBoolean ()
```

```

Dim verificar As Boolean

verificar = True

If verificar = True Then

    MsgBox "Encontrei verdadeiro"

Else

    MsgBox "Encontrei falso"

Endif

End Sub

```

1.5) Workbook / Worksheet

```

Sub DeclaraWbWs()

    Dim wbMeuExcel, wbNovoExcel As Workbook

    Dim wsPlan1, wsDados As Worksheet

    Set wbMeuExcel = ActiveWorkbook

    Set wsDados = wbMeuExcel.Sheets("Plan1")

    Set wbNovoExcel = Workbooks.Open(FileName:="D:\TEMP\Exercicios2.xlsm")

    wbNovoExcel.Plan1.Cells(1,1) = wsDados.Cells(1,2)

    wbMeuExcel.Activate

    wbNovoExcel.Close

End Sub

```

2) Cells e Range

2.1) Cells: Para escolher apenas uma célula -> Plan1.Cells(linha,coluna)

Uso:

```
Sub testaObjetoCells()  
    Cells(2,3).Value = "Teste na linha 2, coluna3"  
End Sub
```

2.3) Range: conjunto de células

2.3.1) Usado como uma célula

```
Sub testaObjetoRange ()  
    Range("B2").Value = "Novo Valor"  
End Sub
```

2.3.2) Usado como um conjunto de células

```
Sub testaObjetoRangeConjunto()  
    Range("A1:A4").Value = "Range Sequencial"  
End Sub
```

2.3.3) Usado como um conjunto de células não sequenciais

```
Sub testaObjetoRangeConjuntoNaoSequenciais()  
    Range("A1:A2,B3:C4,D1:E2").Value = "Range não sequencial"  
End Sub
```

2.3.4) Usado como um conjunto de células de nome definido

Primeiro, dê um nome a um conjunto de células de Selecao. Depois rode a macro.

```
Sub testaObjetoRangeConjuntoNomeDefinido()  
    Sheets("Planilha1").Range("Selecao").Value = "Range com nome definido por Selecao"  
End Sub
```

2.3.5) Usado como um conjunto de Cells

```
Sub testaObjetoRangeConjuntoCells()  
    Range(Cells(1, 1), Cells(4, 1)).Value = "Da célula 1,1 à célula 4,1"  
End Sub
```

2.3.6) Set no Range para usá-lo como variável

```
Sub testaObjetoRangeVariavel ()  
    Dim rangeDefinido As Range  
    Set rangeDefinido = Range("A1:B2")  
    rangeDefinido.Value = "testeRange definido"  
End Sub
```

2.3.6) Set no Range de um nome já definido no excel

```
Sub testaObjetoRangeVariavelComNomeDefinido ()  
    Dim rangeDefinido As Range  
    Set rangeDefinido = Range("Selecao")  
    rangeDefinido.Value = "testeRange definido"  
End Sub
```

2.3.7) Selecionando determinada linha/coluna a partir do começo de um range definido:

```
Sub testaObjetoRangeVariavelComNomeDefinido ()  
    Dim rangeDefinido As Range  
    Set rangeDefinido = Range("B5:G9")  
    rangeDefinido.Rows(3).Select 'Seleciona a terceira linha do range -  
linha 7  
    rangeDefinido.Columns(2).Select 'Seleciona a segunda coluna do range  
- coluna C
```

```
End Sub
```

2.3.8) Limpar determinado conteúdo

```
Sub limparConteudo ()  
  
    Range ("Selecao") = 5 'Coloca valores para apagarmos  
    Range ("Selecao") = "" 'Uma das formas de apagar  
    Range ("Selecao") = 10 'Coloca valores para apagarmos  
    Range("Selecao").ClearContents 'Outra forma de se apagar  
    'Essa forma é utilizada para deletar todo o conteúdo de determinada  
aba  
  
    Dim sheetResultado As Worksheet  
    sheetResultado.Select  
    sheetResultado.Cells.Select  
    Selection.Delete  
  
End Sub
```

2.3.9) Determinar a primeira e última linha de determinado range

```
Sub encontraPrimeiraUltimaLinha ()  
  
    Dim rangeDefinido As Range  
    Dim primeiraLinha , ultimaLinha as Integer  
    Set rangeDefinido = Range ("Selecao")  
    primeiraLinha = rangeDefinido.Row  
    ultimaLinha = primeiraLinha + rangeDefinido.Rows.Count - 1  
  
End Sub
```

Definindo um Range

```
Sub SetRange()  
  
    Dim rangeDefinido as Range  
  
    Set rangeDefinido = Plan1.Range("Selecao")  
  
    Application.Goto Reference:= rangeDefinido  
  
End Sub
```

```
Sub SetRange2()  
  
    Dim rangeDefinido as Range  
  
    Set rangeDefinido = Plan1.Range("Selecao")  
  
    Plan1.select  
  
    rangeDefinido.select  
  
End Sub
```

3) Loops:

If - then /Endif

Funcionamento:

```
IF condition_1 THEN  
    'Instructions inside First IF Block  
ELSEIF condition_2 Then  
    'Instructions inside ELSEIF Block  
...  
ELSEIF condition_n Then  
    'Instructions inside nth ELSEIF Block  
ELSE
```

```
'Instructions inside Else Block
```

```
END IF
```

```
Sub testaIf()  
  
    Dim num As Integer  
  
    num = WorksheetFunction.RandBetween(1, 10)  
  
    Plan1.Cells(1, 1) = num  
  
    If num > 5 Then  
  
        Plan1.Cells(1, 2) = num & " é maior do que 5"  
  
    ElseIf num = 5 Then  
  
        Plan1.Cells(1, 2) = num & " é 5"  
  
    Else  
  
        Plan1.Cells(1, 2) = num & " é menor 5"  
  
    End If  
  
End Sub
```

While / wend

For

For Each – In / Next

```
Sub PercorreRangeComForEach()  
  
    For Each cell In rng  
  
  
  
    Next cell  
  
End Sub
```

1) Concatenar seleção e gravar em uma aba de resultado

```
Sub Concatena()  
  
    Dim selecao, rCell As Range  
  
    Dim stringConcatena As String  
  
    Dim planResultado, planDados As Worksheet  
  
    Dim ultimaLinha As Integer  
  
  
    Set planResultado = Sheets("Resultado")  
  
    Set planDados = Sheets("Plan1")  
  
    Set selecao = Selection  
  
    For Each rCell In selecao  
  
        stringConcatena = stringConcatena & rCell & " "  
  
    Next rCell  
  
    stringConcatena = Application.Trim(stringConcatena)  
  
    planResultado.Select  
  
    ultimaLinha = planResultado.Range("a2000").End(xlUp).Row  
  
    planResultado.Cells(ultimaLinha + 1, 1).Value = stringConcatena  
  
    planDados.Select  
  
End Sub
```

Do While / Loop

```
Sub testaDoWhile()  
  
    Dim sheetDados, sheetResultado As Worksheet  
  
    Dim i, j, nLinha As Integer  
  
    Dim encontraX As Range
```

```

j = 1 'Linha que será usada para a aba de resultados

Set sheetResultado = Sheets("Resultado")

Set sheetDados = Sheets("TestaDowhile")

'Limpa os resultados

sheetResultado.Select

sheetResultado.Cells.Select

Selection.Delete

sheetDados.Select

nLinha = 5

While sheetDados.Cells(nLinha, 2) <> "" 'Enquanto houver dados na coluna
B, começando da linha 5.

    sheetDados.Range(Cells(nLinha, 4), Cells(nLinha, 120)).Select
'Seleciono um range bem grande para garantir que irei pegar todas as células

    Set encontraX = Selection.Find(What:="x", LookIn:=xlFormulas _
, After:=ActiveCell, LookAt:=xlWhole, SearchOrder:=xlByRows,
SearchDirection:=xlNext, MatchCase:=False, SearchFormat:=False)

    If Not encontraX Is Nothing Then 'Se encontrou a primeira vez o
critério

        primeiroX = encontraX.Address 'Grava o endereço numa variável
para comparar depois e poder sair do loop

        Do 'Como já foi encontrado resultado, faz-se ciclo e coloca-se um
loop para as próximas ocorrências

            sheetResultado.Cells(j, 1) = sheetDados.Cells(nLinha, 2)

            sheetResultado.Cells(j, 2) = sheetDados.Cells(4,
encontraX.Column)

            j = j + 1

```

```

        Set encontraX = Selection.FindNext(encontraX) 'faz o find
novamente

        Loop While encontraX.Address <> primeiroX ' Faz o loop enquanto
não voltar para o primeiro endereço

        End If

        nLinha = nLinha + 1 'Soma-se 1 na linha para ele ir para a próxima

        Wend 'Volta para o começo do while para verificar se ainda há dados na
coluna B

End Sub

```

Select Case

Exercícios

- 2) Escrever a mesma coisa em todas as células selecionadas:

```

Sub fazerAlgoEmTodasAsLinhas()

    Selection.Cells.Value = "Escreva em toda a seleção"

End Sub

```

- 3) Pintar as primeiras 5 colunas de todas as linhas de um determinado Range

```

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

'Pintar as linhas que foram selecionadas

    Dim celulaInicial As Range

    Dim linhaInicial, linhaFinal As Integer

    'Retira as cores de todas as células da planilha

    Cells.Interior.ColorIndex = xlNone

    'Determina qual a linha inicial e final que será pintada

```

```

linhaInicial = Target.Row

linhaFinal = Target.Row + Target.Rows.Count - 1

'Seta o range que estará sendo pintado e colore com a cor de Index 12
Set LinhasPintadas = Range(Cells(linhaInicial, 1), Cells(linhaFinal, 5))

LinhasPintadas.Interior.ColorIndex = 12

End Sub

```

4) Encontrar a última célula que foi digitada em determinada coluna:

```

Sub encontraAUltimaLinhaDeDeterminadaColuna ()

    Range("A65536").End(xlUp).Row

End Sub

```

5) Encontrar a última célula que foi digitada depois de determinar um nome no excel:

Primeiro defina o nome em Fórmulas -> Definir Nome (defina como Selecao).

```

Sub encontrarUltimaLinhaDigitadaEmRangeDefinido()

    Dim ultimaLinha As String

    Dim rangeDefinido As Range

    Set rangeDefinido = Plan1.Range("Selecao")

    rangeDefinido.Select

    'A última linha será definida como a soma do primeiro mais o número de linhas

    ultimaLinha = rangeDefinido.Row + rangeDefinido.Rows.Count - 1

    'Imprime na célula B1 o valor da última linha, apenas para controle

    Plan1.Cells(1, 2) = ultimaLinha

End Sub

```

Exemplos Práticos:

- 1) Buscar todos documentos xlsx de determinada pasta para puxar informações:

```
Sub Sub buscarArquivosPasta()  
  
    Dim FSO As New FileSystemObject  
  
    Dim pasta As Folder  
  
    Dim arquivoAtual As File  
  
    Dim caminhoPasta As String  
  
    Dim i As Integer  
  
    Dim minhaPlanilha As Worksheet  
  
    Set minhaPlanilha = Sheets("Planilha2")  
  
    caminhoPasta = minhaPlanilha.Cells(1, 1)  
  
    Set pasta = FSO.GetFolder(caminhoPasta)  
  
    i = 1  
  
    For Each arquivoAtual In pasta.Files  
        minhaPlanilha.Cells(i, 2) = arquivoAtual.Name  
  
        i = i + 1  
  
    Next  
  
End Sub
```

- a) Como saber se o arquivo existe ?

```
arquivo = "d:\teste\teste.txt"  
  
If Len(Dir(arquivo))= 0 Then
```

```
Msgbox "Arquivo " & arquivo & " não existe."  
  
Else  
  
Msgbox "Arquivo " & arquivo & " existe."  
  
End If
```

b) Como saber se o diretório existe?

```
Sub VerificaOuCriaPasta()  
  
    ' Declara uma variável para a planilha e uma para o caminho da pasta  
  
    Dim minhaPlanilha As Worksheet  
  
    Dim caminhoPasta As String  
  
    Dim resposta As Integer  
  
  
    ' Define a planilha de trabalho  
  
    Set minhaPlanilha = Sheets("Planilha2")  
  
    ' Obtém o caminho da pasta a partir da célula A1  
  
    caminhoPasta = minhaPlanilha.Cells(1, 1).Value  
  
  
    ' Verifica se a pasta existe  
  
    If Dir(caminhoPasta, vbDirectory) <> "" Then  
  
        MsgBox "A pasta existe: " & caminhoPasta, vbInformation, "Verificação  
de Pasta"  
  
    Else  
  
        ' Se a pasta não existe, pergunta ao usuário se deseja criá-la  
  
        resposta = MsgBox("A pasta não existe. Deseja criá-la?", vbYesNo +  
vbQuestion, "Criar Pasta")  
  
  
        If resposta = vbYes Then
```

```
        ' Cria a pasta se o usuário escolheu "Sim"

        Mkdir caminhoPasta

        MsgBox "Pasta criada com sucesso: " & caminhoPasta,
vbInformation, "Pasta Criada"

    Else

        ' Informa ao usuário que a pasta não foi criada

        MsgBox "A pasta não foi criada.", vbExclamation, "Ação Cancelada"

    End If

End If

End Sub
```