

## Finalizando

### Transcrição

O que nos resta fazer em nosso código, é **fechar o arquivo de origem**.

Nos exemplos anteriores, fechávamos sempre o arquivo `janeiro.xlsx`, mas agora, a macro não sabe qual arquivo foi aberto. Então, criaremos uma variável para guardar o nome do *outro* arquivo (aquele que foi aberto). Vamos chamá-la de `nomearq2`.

Vamos entender a diferença: a variável `nomearq` guardará o nome do arquivo e seu caminho. Já a variável `nomearq2` guardará apenas o nome do arquivo. Precisamos do nome "puro" do arquivo para que possamos colocar dentro do comando `.Close`.

Então, quando abrirmos o arquivo, o `nomearq2` deverá receber o nome do arquivo ativo.

```
Sub importar03()
    Dim nomearq as String
    Dim nomearq2 as String
    nomearq = Application.GetOpenFilename
    Workbooks.Open Filename:=nomearq
    nomearq2 = ActiveWorkbook.Name

    MsgBox nomearq
    MsgBox nomearq2

    Sheets(1).Select
    Range("a2").Select
    Range(Activecell, Activecell.End(xlDown).End(xlToLeft)).Copy
    Workbooks("importação de texto.xls").Activate
    Range("a6").PasteSpecial xlFormats
    Range("a6").PasteSpecial xlValues
End Sub
```

Rode o script e observe.

Vamos escolher o arquivo `fevereiro`. Depois que a planilha de fevereiro for carregada, podemos ver que o valor que `nomearq2` contém é o nome do próprio arquivo = "fevereiro.xlsx".

Depois que avançamos com o teste, aparecerá a primeira MsgBox mostrando o caminho do arquivo, e a segunda MsgBox mostrará somente o nome do arquivo. Então, colocaremos dentro do método `.Close()` a variável `nomearq2`.

Podemos parar a macro, e apagar as MsgBox que colocamos para fins de teste.

Bom, depois de ter colocado as informações em "importação de texto", vamos acessar a planilha que está em `nomearq2`, e fechar. Acrescentaremos a seleção de `A1` para que não fique selecionado.

```
Sub importar03()
    Dim nomearq as String
    Dim nomearq2 as String
```

```

nomearq = Application.GetOpenFilename
Workbooks.Open Filename:=nomearq
nomearq2 = ActiveWorkbook.Name
Sheets(1).Select
Range("a2").Select
Range(Activecell, Activecell.End(xlDown).End(xlToRight)).Copy
Workbooks("importação de texto.xmls.").Activate
Range("a6").PasteSpecial xlFormats
Range("a6").PasteSpecial xlValues
Workbooks(nomearq2).Activate
Workbooks(nomearq2).Close
Range("a1").Select
End Sub

```

Como você pode ver, nós sempre pedimos que a macro colasse sempre a partir da linha A6 . Então, não importa se já tem informações ou não, a macro irá colar do mesmo jeito, e assim corremos o risco de perder informações.

Para resolver isso, temos que especificar para a macro um jeito de colar informações sempre a partir da última célula preenchida. Usaremos conceitos vistos nas primeiras aulas.

A ideia é ir até a última linha da planilha, e voltar! Antes de colar a informação em Range("a6").PasteSpecial xlFormats , posicionaremos na célula que queremos, que é a última! Range("a6") passa a ser **célula ativa**, e a partir dela, colaremos o formato e valores.

```

Sub importar03()
    Dim nomearq as String
    Dim nomearq2 as String
    nomearq = Application.GetOpenFilename
    Workbooks.Open Filename:=nomearq
    nomearq2 = ActiveWorkbook.Name
    Sheets(1).Select
    Range("a2").Select
    Range(Activecell, Activecell.End(xlDown).End(xlToRight)).Copy
    Workbooks("importação de texto.xmls.").Activate
    Range("a1048576").End(xlUp).Offset(1, 0).Select
    ActiveCell.PasteSpecial xlFormats
    ActiveCell.PasteSpecial xlValues
    Workbooks(nomearq2).Activate
    Workbooks(nomearq2).Close
    Range("a1").Select
End Sub

```

Vamos executar o teste passo a passo. Não se esqueça de rodar a macro pela segunda vez, e assim você poderá escolher um outro arquivo para pegar as informações.

No meio da execução da macro, algumas mensagens podem aparecer e podem atrapalhar ou ser irrelevantes. Para desabilitá-las, colocaremos o comando Application.DisplayAlerts = False logo abaixo do nome da macro, e também Application.DisplayAlerts = True antes de fechar a macro. Assim, nós somente desabilitaremos essas mensagens durante a execução da macro.

Vamos executar novamente. Você pode selecionar o arquivo de Março , caso já tenha selecionado os dados dos arquivos de Janeiro e fevereiro .

Certo. Com certeza, nós também podemos usar o botão de "Importar"! Mas antes de testá-lo, não se esqueça de apagar os dados da tabela `Base_Clientes_Completa` :)

Depois que importamos os dados a partir do **botão**, percebemos um certo *delay* na hora de carregar os dados para a planilha. Como podemos resolver isso?

Na segunda linha do nosso código, colocaremos a instrução `Application.ScreenUpdating = False`, e na última linha, colocaremos a mesma instrução, porém com `True`.

Para testar, apagaremos todos esses dados novamente, e executaremos o código a partir do botão!

Esperamos que essas informações possa te ajudar em algumas atividades diárias. Trabalhamos com arquivos, com cópias de informação, trabalhamos com algumas propriedades do `Application` e utilizamos mais variáveis.

Nos vemos no próximo capítulo!