

Testando o Método Activate

Transcrição

Vamos analisar juntos o código do vídeo anterior.

```
Sub importar02()  
    Workbooks.Open Filename:="C:\Users\caelum\Documents\VBA\janeiro.xlsx"  
    Sheets(1).Select  
    Range("a2:g11").Copy  
    Workbooks("importação de texto.xlsm").Activate  
    Range("a16").PasteSpecial xlFormats  
    Range("a16").PasteSpecial xlValues  
    Workbooks("janeiro.xlsx").Activate  
    Sheets(2).Select  
    Range("a2:g3").Copy  
    Range("a26").PasteSpecial xlFormats  
    Range("a26").PasteSpecial xlValues  
    Range("a1").Select  
    Workbooks("janeiro.xlsx").Activate  
    Workbooks("janeiro.xlsx").Close False  
End Sub
```

Nós abrimos o arquivo, paramos na planilha *um*, e copiamos a informação. Depois fomos para a planilha "Importação de texto", e colamos os dados. Voltamos para o arquivo *Janeiro*, e paramos na planilha *dois*. Depois selecionamos os dados para copiar, entretanto nós não mudamos de arquivo, ainda estamos em *Janeiro*, e precisamos ir para a "importação de texto.xlsm".

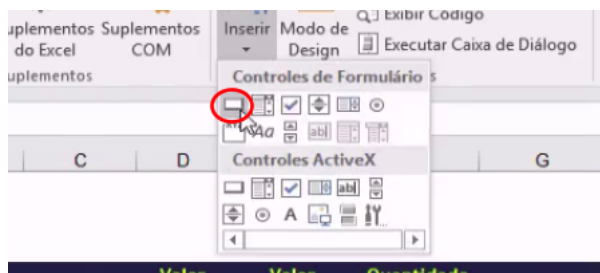
```
Sub importar02()  
    Workbooks.Open Filename:="C:\Users\caelum\Documents\VBA\janeiro.xlsx"  
    Sheets(1).Select  
    Range("a2:g11").Copy  
    Workbooks("importação de texto.xlsm").Activate  
    Range("a16").PasteSpecial xlFormats  
    Range("a16").PasteSpecial xlValues  
    Workbooks("janeiro.xlsx").Activate  
    Sheets(2).Select  
    Range("a2:g3").Copy  
    Workbooks("importação de texto.xlsm").Activate  
    Range("a26").PasteSpecial xlFormats  
    Range("a26").PasteSpecial xlValues  
    Range("a1").Select  
    Workbooks("janeiro.xlsx").Activate  
    Workbooks("janeiro.xlsx").Close False  
End Sub
```

Legal. Depois dessa alteração, mudamos novamente de arquivo, e nos posicionamos na célula *A26*. Depois colamos a informação. Paramos no *A1* e voltamos para a planilha *Janeiro* e fechamos o arquivo.

Aparentemente está tudo bem... Mas, tome cuidado com a *autoconfiança*. É possível encontrar um erro de código se executarmos a macro passo a passo ("F8"). Assim evitamos falhas.

Bom, agora nós temos a macro completa, e é possível melhorar ainda mais esse código.

Vamos colocar um botão, clicando na aba "Desenvolvedor > Inserir" e escolheremos a opção para criar um novo botão:



Depois, selecionaremos a macro `Importar02`. Para que seja melhor de entendermos, vamos apagar os dados da planilha `Base_Clientes`. Faremos algumas modificações em nosso script. Onde especificamos a célula `A16`, vamos trocar para `A6`, e onde há `A26`, trocaremos para `A16`.

```
Sub importar02()
    '...
    Range("a6").PasteSpecial xlFormats
    Range("a6").PasteSpecial xlValues
    '...
    Range("a16").PasteSpecial xlFormats
    Range("a16").PasteSpecial xlValues
    '...
End Sub
```

Depois que fizemos essas alterações, basta clicar no botão, que os dados serão trazidos automaticamente, os dez dados da planilha um e mais dois que estão na planilha dois. Se clicarmos novamente no botão, os dados serão colados em cima em todas as vezes!

Repare que a cada vez que nós clicamos no botão, a tela começa a piscar. Isso acontece porque a tela é atualizada. Então, dependendo do processamento da sua máquina, pode ser que esse tempo seja um pouco menor, ou até mesmo maior. E o que podemos fazer para melhorar?

Na primeira linha da macro, colocaremos `Application.ScreenUpdating = False`. Aqui, estamos desabilitando os saltos de tela, desligando a atualização de tela. Antes de terminar a macro colocaremos `Application.ScreenUpdating = True`.

Vamos apagar os dados da planilha para que possamos ver a diferença ao clicar no botão novamente. Como você pode ver, a tela não se atualiza mais.

Ainda podemos colocar uma mensagem para o usuário no final da nossa macro `MsgBox "Importação concluída com êxito"`.

Legal. Então, depois de clicar no botão "Importar02", vemos que a mensagem de que a importação foi concluída com êxito aparece logo após de trazer os dados para a tela.

Trabalhamos com o método `Workbooks`, vimos como fechar um arquivo e alternar entre outros arquivos. Também trabalhamos com o `Copy` e `Paste`. Vimos a diferença de quando queremos somente colar a informação, ou somente o

valor, ou até mesmo a formatação. E no final, mostramos o `Application.ScreenUpdating`, que nos permite utilizar nas macros sempre que houver um *delay* de atualização de telas.