

Percorrendo todas as linhas

Transcrição

Agora que nós utilizamos a referência relativa a partir de B6, que tal percorrermos todas as outras linhas?

No vídeo anterior, introduzimos o `While` para que ele verificasse todas as linhas, mas aconteceu que esse laço somente verificava B6. O certo seria ir mudando de linha a cada verificação sendo a cidade igual ou não ao `InputBox`.

Então, após o `If`, colocaremos o comando para descer uma linha.

```
Sub exemplo_While()  
    Dim cidade As String  
    cidade = InputBox("Digite a Cidade a ser consultada")  
    Range("b6").Select  
    'Faça enquanto  
    Do While ActiveCell <> ""  
        If ActiveCell = cidade Then  
            Range(ActiveCell.Offset(0, -1), ActiveCell.Offset(0, 5)).Interior.ColorIndex = 24  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Loop  
End Sub
```

Para descer uma linha no `Offset()`, o número será **positivo** e para subir linhas, ele será **negativo**. Como não vamos mudar de coluna, colocamos o número 0.

Vamos rodar a macro para testar se resolvemos esse problema. No `InputBox` digitaremos "Guarulhos". Conseguimos entrar no `While`, pois a célula B6 é diferente de vazio. A primeira célula que tem "São Paulo" é igual a "Guarulhos"? Não! Então, saímos do `If`, e descemos uma linha, bateu no `Loop` e voltou para o `Do While`.

A célula ativa é diferente de vazio? Sim! Essa mesma célula é igual a cidade digitada "Guarulhos"? Sim! Então, a linha que corresponde a essa cidade será pintada. E assim, o nosso laço de repetição vai confirmando linha a linha, até que ele encontre a primeira célula vazia, e o `Loop` chega ao seu fim.

Para melhorar, podemos pedir para posicionar no início da macro, no A6 por exemplo:

```
Sub exemplo_While()  
    Dim cidade As String  
    cidade = InputBox("Digite a Cidade a ser consultada")  
    Range("b6").Select  
    'Faça enquanto  
    Do While ActiveCell <> ""  
        If ActiveCell = cidade Then  
            Range(ActiveCell.Offset(0, -1), ActiveCell.Offset(0, 5)).Interior.ColorIndex = 24  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Loop  
    Range("a6").Select  
End Sub
```

Deixaremos a tabela como ela estava inicialmente, para que nós possamos visualizar claramente a macro em execução. Vamos criar um botão, que executará a macro `exemplo_While()`. O botão se chamará "Executar_While".

Vamos executar novamente informando "Guarulhos". Percebeu como foi rápido? :)

Agora, informaremos "Osasco" no `InputBox`. Reparou que além da cidade de Osasco, a cidade de Guarulhos continuou pintada? Havíamos feito anteriormente! Se executarmos a macro mais uma vez, porém indicando a cidade de São Paulo, além de estar pintado as cidades "Guarulhos" e "Osasco", "São Paulo" também estará. Ou seja, a macro está *mantendo* o que já tínhamos, e só está executando o que precisamos alterar.

O que podemos fazer para melhorar?

Bom, antes de executar todo esse `Do While`, é interessante selecionar todas as células e pintá-las com a mesma cor.

```
Sub exemplo_While()  
    Dim cidade As String  
    cidade = InputBox("Digite a Cidade a ser consultada")  
    Range("b6").Select  
    Range("a6:g36").Interior.ColorIndex = 26  
    'Faça enquanto celula ativa for diferente de vazio  
    Do While Activecell <> ""  
        If Activecell = cidade Then  
            Range(Activecell.Offset(0, -1), Activecell.Offset(0, 5)).Interior.ColorIndex = 24  
        End If  
        Activecell.Offset(1, 0).Select  
    Loop  
    Range("a6").Select  
End Sub
```

E assim, toda a vez que a macro for executada, ela irá padronizar todas as células, pintando-as de uma cor, e depois destacando de acordo com a cidade informada.

Mas como vamos saber o número que representa cada cor aqui no VBA? Sabemos que são 56 cores no total, porém não sabemos a sua numeração. A seguir, criaremos uma tabela de cores utilizando laços, para que fique mais fácil a identificação das cores.