

Outro tipo de laço: For

Transcrição

Ao decorrer dessa aula, vimos dois tipos de **estruturas de repetição**: *Do While* e o *Do Until*. E da mesma forma que os laços anteriores, veremos e aplicaremos um novo conhecimento utilizando o mesmo exemplo. Neste vídeo, falaremos do **For**.

Então, temos a mesma estrutura de informações, aquela planilha de segurados. Vamos acessar o **VBA**, e fazer uma cópia do último exemplo da macro *Do Until*, e vamos colar.

Nesse caso, este será o `exemplo_For`.

Bom, o `For`, diferente dos outros laços vistos até agora, possui um contador, e podemos *incrementar* ou *decrementar* esse valor da maneira que desejarmos. Se temos um contador que vai somando valores, precisamos então, criar uma *variável*. A nossa variável se chamará `linha` e será do tipo `Integer`.

```
Sub exemplo_For()  
    Dim cidade As String  
    Dim tot_seg as Integer  
    Dim tot_val as Double  
    Dim linha as Integer  
    cidade = InputBox("Digite a Cidade a ser consultada")  
  
    'restante do código  
End Sub
```

Sabemos que o laço `For` começará na linha 6 e irá terminar até a linha 36. Então, faremos assim:

```
Sub exemplo_For()  
    Dim cidade As String  
    Dim tot_seg as Integer  
    Dim tot_val as Double  
    Dim linha as Integer  
    cidade = InputBox("Digite a Cidade a ser consultada")  
    'absoluta  
    'Range("a6:g36").Interior.ColorIndex = 19  
    'relativa  
    Range("a6").Select  
    Range(ActiveCell,ActiveCell.End(xlDown).End(xlToRight)).Interior.ColorIndex = 19  
    Range("b6").Select  
    'Para variavel comecando em 6 ate 36 - executar comandos  
    For linha = 6 to 36  
        If UCase(ActiveCell) = UCase(cidade) Then  
            Range(ActiveCell.Offset(0, -1), ActiveCell.Offset(0, 5)).Interior.ColorIndex = 24  
            tot_seg = tot_seg + 1  
            tot_val = tot_val + ActiveCell.Offset(0, 5)  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Next  
    Range("a6").Select
```

```
MsgBox "Total de Segurados em " & cidade & " - " & tot_seg, vbInformation, "Resultado Obtido"
End Sub
```

Na linha `For linha = 6 to 36`, entendemos que a variável `linha` vai de 6 até(to) 36, e o laço `For` termina com a palavra `Next`. Ao invés de indicarmos uma célula ativa, usaremos o comando `Cells`.

O comando `Cells` funciona da seguinte maneira: Para clicar na célula B6, só fazemos `Range("b6").Select`. Para realizar a mesma tarefa, porém com o `Cells`, informaremos o **número da linha**, e o **número da coluna**, e depois o comando `Select`. Então ficará assim:

```
Cells(num linha, num coluna).[método qualquer]
```

Vamos supor que queremos a célula B6, o comando inteiro utilizando `Cells` será assim:

```
Cells(6, 2).Select
```

Colocamos 2 na coluna, pois B é a segunda coluna.

Temos um outro exemplo, onde queremos pintar a célula J500, sem o `Cells`, ficaria assim:

```
Range("j500").Interior.ColorIndex = 3
```

E com `Cells`, ficaria assim:

```
Cells(500, 10).Interior.ColorIndex = 3
```

Certo. Voltando para a ideia do código de exemplo dos Segurados, se estamos deslocando as linhas usando o laço `For`, ao invés de utilizarmos o `ActiveCell`, usaremos `Cells` no `IF`:

```
Sub exemplo_For()
    Dim cidade As String
    Dim tot_seg As Integer
    Dim tot_val As Double
    Dim linha As Integer
    cidade = InputBox("Digite a Cidade a ser consultada")
    'absoluta
    'Range("a6:g36").Interior.ColorIndex = 19
    'relativa
    Range("a6").Select
    Range(ActiveCell, ActiveCell.End(xlDown).End(xlToRight)).Interior.ColorIndex = 19
    'Para variavel comecando em 6 ate 36 - executar comandos
    For linha = 6 to 36
        If UCase(Cells(linha, 2)) = UCase(cidade) Then
            Range(Cells(linha, 1), Cells(linha, 7)).Interior.ColorIndex = 24
            tot_seg = tot_seg + 1
            tot_val = tot_val + Cells(linha, 7)
        End If
    Next linha
End Sub
```

```
ActiveCell.Offset(1, 0).Select
Next
Range("a6").Select
MsgBox "Total de Segurados em " & cidade & " - " & tot_seg, vbInformation, "Resultado Obtido"
End Sub
```

Não é mais necessário parar em 86, pois já indicamos que começará na linha 6, e então podemos tirá-lo do código. Vamos executar novamente para que nós possamos ter uma noção do que vai acontecer no passo a passo.

O laço For termina com Next, e a cada passagem no Next, é somado 1 na variável linha.

Executaremos com "F8", e informaremos "Guarulhos" no InputBox. Repare que a primeira linha não é Guarulhos. E então, estamos descendo uma linha com o comando `ActiveCell.Offset(1, 0).Select`, e a princípio não é necessário, pois o comando `Next` já incrementa a linha para nós. Porém, para fins didáticos, é interessante deixá-lo, pois assim nós conseguimos ver o `For` percorrer pela tabela.

Se quisermos, podemos inserir um novo botão, e atribuir a macro `exemplo_For`.

RELEMBRANDO

O laço `For` necessita de uma variável numérica. Essa variável começa em um valor, e vai até um valor final.

A cada vez que damos um `Next`, uma linha é incrementada, e então não precisamos utilizar o `ActiveCell`.

Falando sobre os três laços, o que você consegue fazer com um, você consegue fazer com outros. E a ideia é que você aprenda a utilizá-los para que mais pra frente, você possa utilizar algum cada um com a sua responsabilidade.

Obrigada pela sua participação, e vamos para a próxima aula.