

05

Ajustando a tabela

Transcrição

Dando continuidade na criação da tabela, iremos realizar alguns ajustes necessários. Uma das coisas que fizemos foi deixar os dados da tabela fixos, inclusive o `Valor Total`.

Agora imagine o trabalho que nós teríamos se os valores fossem diferentes e se estivessem em uma proporção muito maior? Bom, teríamos que ter sempre uma calculadora para nos auxiliar nesses cálculos.

Neste caso, trabalharemos com os **operadores matemáticos** bem conhecidos:

- + adição
- subtração
- * multiplicação
- / divisão

O VBA segue os mesmos princípios da matemática que aprendemos na escola, ou seja, ele dará *prioridade* aos operadores de **multiplicação** e de **divisão**, e depois aos operadores de **adição** e **subtração**. Podemos usar os `()` se for necessário.

Sabendo disso, vamos acessar o script, e alterar o campo `Valor Total`.

Pegaremos o `Valor Pessoa` e somaremos com `Valor Dependentes`. Caso haja mais que um dependente, vamos *multiplicar* o `Valor Dependentes` pela `Quantidade`, e então iremos somar esse resultado com o `Valor Pessoa`. Vai ficar assim:

```
Sub Tabela_Simples()
    'Criar o título da tabela com a utilização _
    de comandos em VBA
    '...

    'Efetuar o cadastro do cliente
    'inserir informações no objeto Célula
    '...
    Range("f6").Value = 2
    Range("g6").Value = Range("d6").Value + Range("e6").Value * Range("f6").Value
End Sub
```

Então, o código irá somar com o resultado da multiplicação. Sabemos que a multiplicação terá prioridade e será executada primeiro, mas também podemos fazer o uso dos parênteses dessa forma:

```
Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)
```

Legal. Vamos apagar novamente a nossa tabela na planilha do Excel, e vamos executar a nossa macro. Clicando em qualquer linha do código, usaremos o atalho "F5".

Podemos observar que o `Valor Total` está fixo na célula, ele não pode ser apagado. Mas, nós conseguimos alterar o `Valor Pessoa`, porém o `Valor Total` continua com o mesmo valor que foi mandado do VBA para a planilha.

Com isso, concluímos que o `Valor Total` não está fazendo parte de uma fórmula matemática; é um valor fixo.

Mas, e se quiséssemos deixar **automático**?

Podemos criar uma **fórmula matemática** na macro! Fazemos assim:

```
'inserir informações no objeto Célula
'...
Range("f6").Value = 2
Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)
Range("g6").Value = "=d6+e6*f6"
End Sub
```

Precisamos usar as aspas pois não estamos querendo passar um valor numérico. E então, podemos apagar ou deixar comentada a linha com um apóstrofo `'Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)`.

O *apóstrofo* serve tanto para colocar comentários na macro, como também serve para *ignorar* linhas que você não irá usar.

Vamos apagar os dados da planilha, e dar um "F5" novamente. O que mudou? A princípio, nada! Mas, podemos observar que na célula G5 existe a fórmula que inserimos no script.

Se mudarmos o `Valor Pessoa`, veja que o `Valor Total` também é alterado automaticamente. Na verdade, se alterarmos também a `Quantidade` e o `Valor Dependente`, o `Valor Total` exibirá um novo total!

```
'inserir informações no objeto Célula
'...
Range("f6").Value = 2
'Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)
Range("g6").Value = "=d6+e6*f6"
End Sub
```

Como podemos ver, estamos definindo os valores dentro da macro, o que dificulta a tarefa de usar a tabela. Todas as vezes que queremos incluir um cadastro novo, precisamos definir o valor de cada campo diretamente na célula. Há uma forma mais amigável e simpática de inserir dados em uma célula, sem ter o contato com o código em VBA.

Nesse caso, podemos fazer o uso do `InputBox`! A função `InputBox` é uma maneira simples de capturar textos introduzidos pelo usuário. Podemos usá-lo dessa forma:

```
Sub Tabela_Simples2()
    'títulos da tabela
    'inserindo informações
    Range("a6").Value = InputBox("Insira o nome do Segurado")
    Range("b6").Value = InputBox("Cidade")
    Range("c6").Value = InputBox("Tipo de Plano")
    Range("d6").Value = InputBox("Digite o valor do Segurado")
```

```
Range("e6").Value = InputBox("Valor do Dependente")
Range("f6").Value = InputBox("Quantidade de Dependentes")
'Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)
Range("g6").Value = "=d6+e6*f6"
End Sub
```

Dessa forma, o usuário terá que colocar os valores no campo, e só passará para a próxima informação depois que o "OK" for clicado.

É interessante acrescentar um `MsgBox`, exibindo uma mensagem de que o cadastro foi realizado com sucesso.

```
Sub Tabela_Simples2()
    'títulos da tabela
    'inserindo informações
    Range("a6").Value = InputBox("Insira o nome do Segurado")
    Range("b6").Value = InputBox("Cidade")
    Range("c6").Value = InputBox("Tipo de Plano")
    Range("d6").Value = InputBox("Digite o valor do Segurado")
    Range("e6").Value = InputBox("Valor do Dependente")
    Range("f6").Value = InputBox("Quantidade de Dependentes")
    'Range("g6").Value = Range("d6").Value + (Range("e6").Value * Range("f6").Value)
    Range("g6").Value = "=d6+e6*f6"
    MsgBox "Cadastro concluído com sucesso!!!!"
End Sub
```

Na próxima aula, veremos algumas informações importantes sobre **macros** e sobre como vamos salvar o nosso arquivo.