

07

## Condição Case

### Transcrição

Abordaremos um novo conceito de *instrução condicional* utilizando o mesmo exemplo dos vídeos anteriores. Vamos selecionar o código inteiro, copiar e colar logo abaixo dele. Lembrando que não podemos ter duas macros com o mesmo nome! Então, mudaremos o nome da primeira macro para `auto_open_IF_Elseif`, e deixaremos o nome da segunda de `auto_open`.

Vamos utilizar a segunda macro. Como sabemos, fizemos várias perguntas para o horário, para o objeto `Now` na função `Hour()`. Quando tivermos várias perguntas para o mesmo item, podemos utilizar uma instrução chamada de **Select Case**.

```
Sub auto_open()
    'Criar variavel info do tipo String
    Dim info As String
    Sheets("entrada").Select
    'variavel recebe conteudo digitado
    info = InputBox("Digite uma senha")
    'Se conteudo da variavel for "abc" entao...
    'If UCase(info) = "ABC" Then
    If LCase(info) = "abc" Then
        Select Case Hour(Now)

    End Select

    If Hour(Now) < 12 Then
        MsgBox "Bom dia"
    ElseIf Hour(Now) < 18 Then      'senaoSE <condicao>
        MsgBox "Boa tarde"
    Else                           ' caso contrario
        MsgBox "Boa noite"
    End If
    MsgBox "Bem vindo ao sistema!"
    Sheets("despesas").Select
Else
    MsgBox "Senha inválida!"
    Application.Quit
End If
End Sub
```

Utilizamos o `Select Case + <objeto>`, e como podemos ver, terminamos o `Select Case` com `End Select`. Dentro dele, colocaremos quantas perguntas quisermos, sendo que todas as perguntas feitas, serão sempre para o mesmo objeto.

```
If LCase(info) = "abc" Then
    Select Case Hour(Now)
        Case Is < 12

        Case Is < 18
```

```

Case Else

End Select

'Ifs para a hora do sistema
Else
    'Msgbox de senha invalida
    'Fechar o programa
End If

```

Como já foi dito, podemos ter quantos Cases nós quisermos. Então, se for menor que 12 , vamos exibir "Bom dia". Se for menor que 18 , vamos exibir "Boa tarde", e caso não seja nenhuma das opções, exibiremos "Boa noite".

## O que temos de vantagem em relação ao If?

Bom, a estrutura que fizemos com o If , foi a seguinte:

```

If Hour(Now) < 12 Then
    MsgBox "Bom dia"
ElseIf Hour(Now) < 18 Then      'senaoSE <condicao>
    MsgBox "Boa tarde"
Else                           ' caso contrario
    MsgBox "Boa noite"
End If

```

Então sempre colocamos o item (objeto) e a condição, se não temos que repetir o item novamente e também a condição. Se tivessem mais perguntas, teríamos que sempre perguntar para o objeto em todas as vezes, e assim, nós escreveríamos muito mais.

Imagine que nós iremos perguntar sobre todas as siglas de todos os estados do Brasil. Usaríamos muitos ElseIf s, não é mesmo? Com o Select Case , indicamos o item somente uma vez, e colocaremos somente as siglas embaixo. Vamos apagar a estrutura do If , e utilizar a Select Case .

```

If UCase(info) = "ABC" Then
    Select Case Hour(Now)
        Case Is < 12
            MsgBox "Bom dia"
        Case Is < 18
            MsgBox "Boa tarde"
        Case Else
            MsgBox "Boa noite"
    End Select
    MsgBox "Bem vindo ao sistema!"
    Sheets("despesas").Select
Else
    MsgBox "Senha inválida!"
    Application.Quit
End If

```

Aqui, temos uma outra estrutura diferenciada dentro de um if . E quando vamos usar a estrutura Select Case ? Apenas quando tivermos várias perguntas para o mesmo item. Assim indicaremos o item somente uma vez, e depois

vamos perguntando.

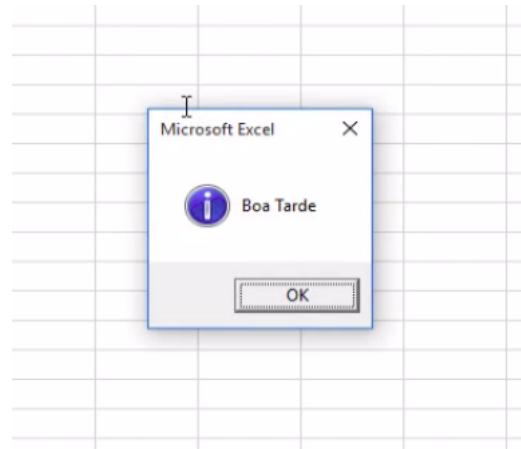
Antes de executar, vamos melhorar um pouco mais essas mensagens que serão exibidas. Nos `MsgBox`, podemos colocar uma personalização em cada caixa. Então depois da mensagem, vamos colocar uma vírgula e depois dizemos que tipo de mensagem será.

```
If UCase(info) = "ABC" Then
    Select Case Hour(Now)
        Case Is < 12
            MsgBox "Bom dia", vbInformation
        Case Is < 18
            MsgBox "Boa tarde", vbInformation
        Case Else
            MsgBox "Boa noite", vbInformation
    End Select
    MsgBox "Bem vindo ao sistema!", vbExclamation
    Sheets("despesas").Select
Else
    MsgBox "Senha inválida!", vbCritical
    Application.Quit
End If
```

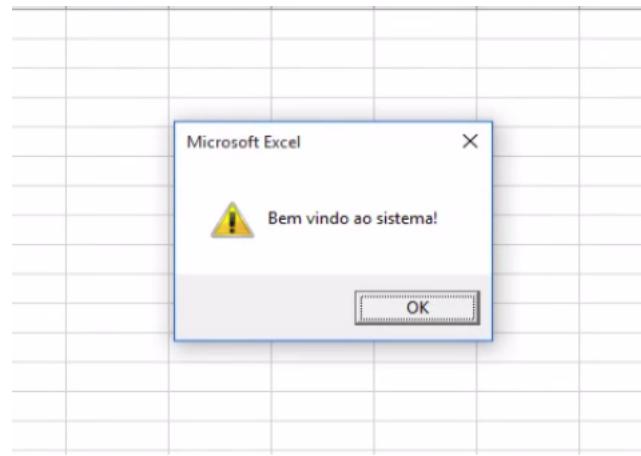
O `vbExclamation` personaliza a mensagem de forma que ela aparece dentro de uma caixa com uma !. Podemos também, utilizar o `vbInformation` é uma mensagem de informação, e será mostrado um ícone dentro da caixa.

Em "Senha inválida" usamos `vbCritical`, que mostra uma mensagem de erro. Vamos salvar e executar essa macro para ver como vai ficar com essas alterações.

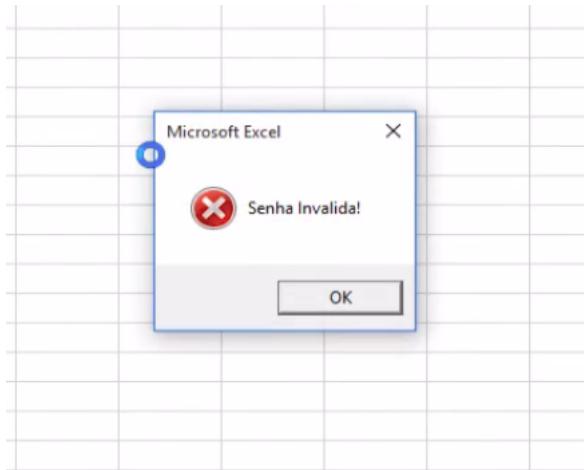
Após ter colocado a senha corretamente, é exibido um ícone diferente na caixa de "Boa tarde":



E logo depois, a mensagem de "Bem vindo ao sistema" é mostrada em uma caixa com uma exclamação:



Agora, vamos rodar novamente, porém não vamos acertar a senha.



Como podemos ver, agora aparece um "X" de que houve um erro.

Bom, precisamos relembrar que é executado a macro `auto_open`. E se quisermos que a outra macro seja executada, é necessário o uso do "F5" ou "F8". Assim como vimos o `auto_open()`, também podemos ter o `auto_close()`. E o que faz a macro `auto_close()`?

Essa macro irá executar alguma ação, ou exibir alguma informação sempre que fecharmos o arquivo. Por exemplo, quando vamos fechar o arquivo e nos esquecemos de salvar. Nesse caso, podemos pedir para a macro salvar o nosso arquivo automaticamente, independente da pergunta que o Excel nos fará, afim de salvarmos o arquivo.

Podemos mostrar uma mensagem ao usuário, através da macro `auto_close()`. Vamos fazer isso:

```
Sub auto_close()
    MsgBox "Até Breve!"
End Sub
```

Legal, agora vamos pedir para que a macro salve automaticamente o arquivo para nós. Em VBA, como chamamos os arquivos? **Workbook!** Então, para salvar esse workbook, fazemos assim:

```
Sub auto_close()
    ActiveWorkbook.Save
    MsgBox "Até Breve!"
End Sub
```

Com isso, ele salvará o arquivo que está ativo, e depois o Excel será fechado. Vamos fazer um teste. Mude alguma coisa na planilha de `Despesas`. Agora, ao invés de salvar, vamos deixar que a macro faça isso por nós. Clicamos em "Arquivo > Fechar". Assim como o nosso código, a mensagem de "Até Breve" apareceu. Então, isso quer dizer que o arquivo já foi salvo! Pois a mensagem deve aparecer somente depois que houver o salvamento do arquivo. Reparamos que não foi pedido para salvarmos novamente.

Percebemos que o Excel nos mostra uma variedades de objetos e comandos diferenciados para nos auxiliar em um determinado objetivo. Vimos a questão de variáveis, navegação entre planilhas, instruções condicionais através do comando `If` e do `Select Case`. Vimos também que podemos colocar uma instrução dentro da outra. Vimos um pouco do `auto_open()` e do `auto_close()` do arquivo, e para concluir, vimos o objeto `ActiveWorkbook`, que é o objeto para tratamento de arquivo, onde podemos salvar automaticamente o arquivo antes de fechar.