

Apêndice 1: Macros e Interação com o Usuário

A1.1 Introdução

A macro é uma rotina de usuário no VBA usada para rotinas automática ou operações repetitivas no Excel. Macros são também chamadas subrotinas; usamos esses nomes alternadamente. Algumas das rotinas VBA de interação com o usuário estão cobertas nas [seções A1.3 e A1.4](#). Módulos (foram mencionados brevemente no [Capítulo 31](#)) são dados com uma cobertura completa como o [último assunto](#) do capítulo.

A1.2 Subrotines Macro

A primeira linha de uma subrotina macro dá um nome à macro e lista os parâmetros se houverem. É muito semelhante à primeira linha de uma função:

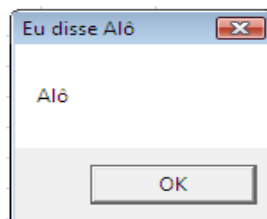
```
Sub NomeDaMacro()
```

A última linha indica o final da macro e assim se parece (muito aproximadamente) com isto:

```
End Sub
```

Separando a primeira e a última linhas estão aquelas declarações que a macro executa. O que segue é uma macro muito simples que coloca uma mensagem na tela.

```
Sub DigaAlo()  
    MsgBox "Alô", , "Eu disse Alô"  
End Sub
```



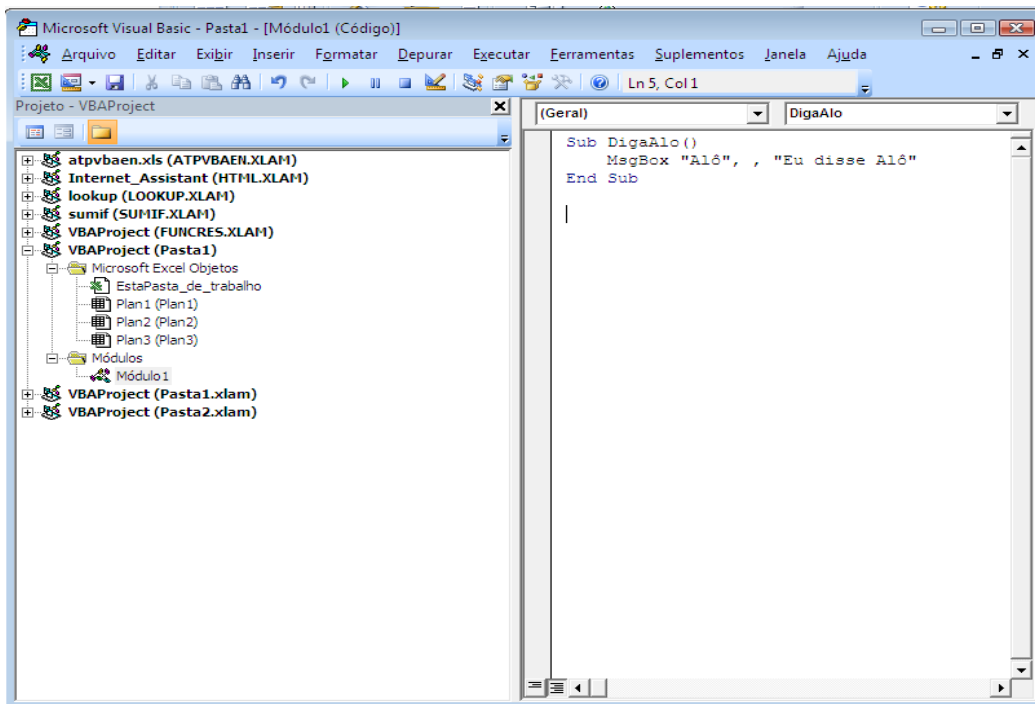
A macro introduz uma macro VBA interna chamada **MsgBox**. Ela também introduz a maneira como uma macro é ativada (chamada) de uma rotina VBA. A **MsgBox** é chamada por um comando na linha seguido por uma lista de argumentos separados por vírgulas. Note a sintaxe:

```
MsgBox "Alô", , " Eu disse Alô "
```

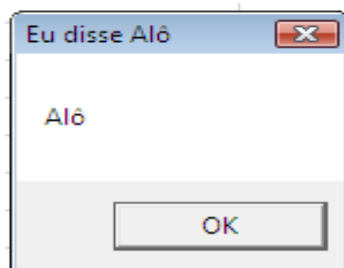
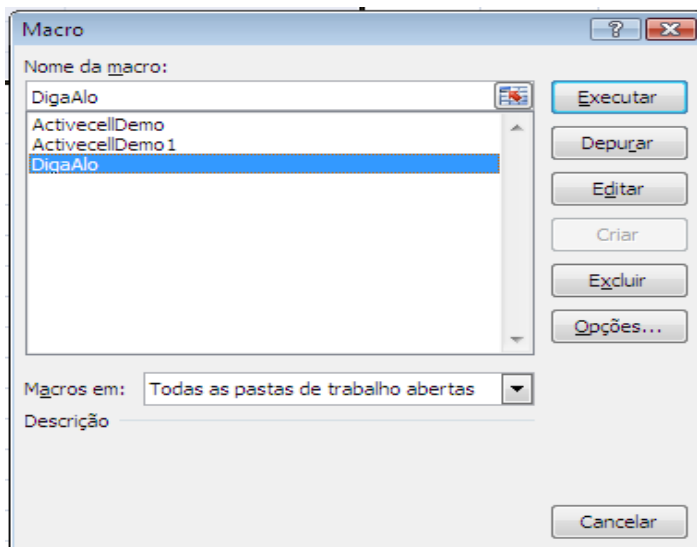
Existem três argumentos:

- "Alô" é a mensagem que será mostrada.
- O segundo argumento está vazio: note o espaço entre as vírgulas. Este argumento pode ser usado para definir botões para a caixa de mensagem.
- O terceiro argumento é: "Eu disse Alô" — este é o título da caixa de mensagem.

Uma macro pode ser ativada (rodada) no Excel de várias maneiras. A maneira mais simples de rodar uma macro é do menu ferramentas ir para a guia **Desenvolvedor** e no Grupo **Código** escolher o botão **Visual Basic**.



Se usarmos o botão Macros, ou usando a tecla de atalho de teclado **[Alt + F8]**, aparece a caixa de seleção de macro. A caixa lista alfabeticamente todas as macros disponíveis. Encontre a sua macro, clique no seu nome, e clique no botão Executar.



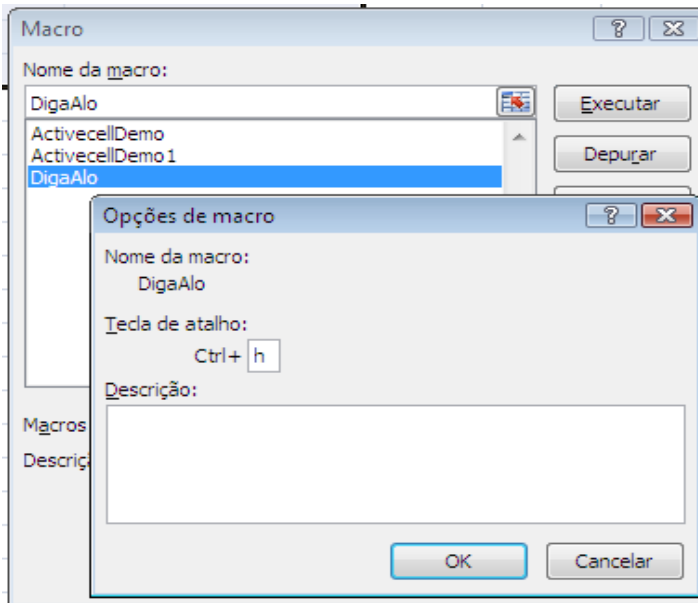
Nesse momento o Excel está travado. Você tem que clicar o botão **OK** antes de progredir.

A1.2.1 Tecla de Atalho para Macros

Usando uma tecla de atalho de teclado é um modo mais rápido de fazer a macro rodar. Para prender uma tecla de atalho a nossa macro:

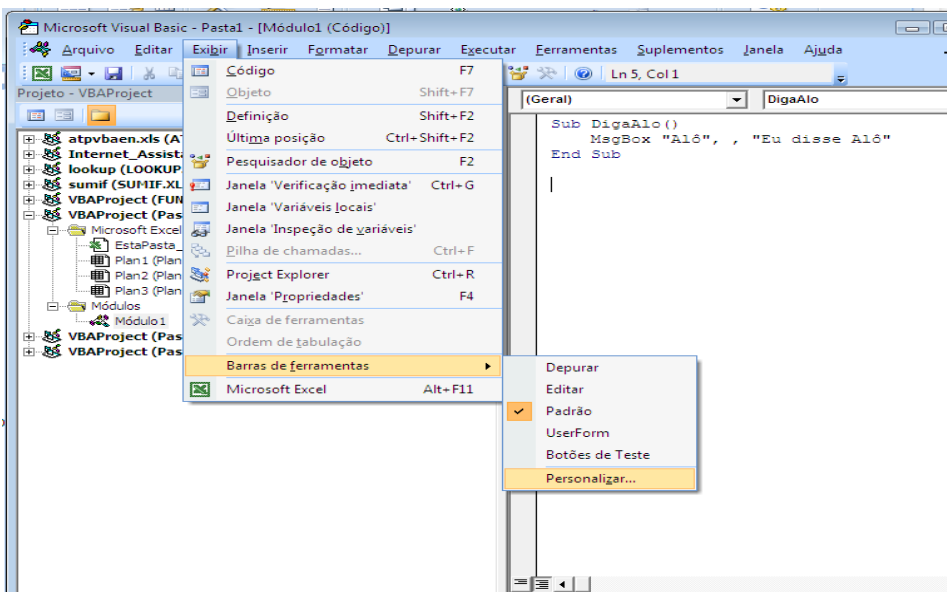
1. Selecione o botão **Opções** da caixa de seleção de macro.
2. Digite um caractere no espaço fornecido, e clique **OK**.
3. Feche a caixa de seleção de macro usando o **X** no canto superior direito.

Você pode ativar agora a macro usando a tecla de atalho ([Ctrl + h] no nosso caso).

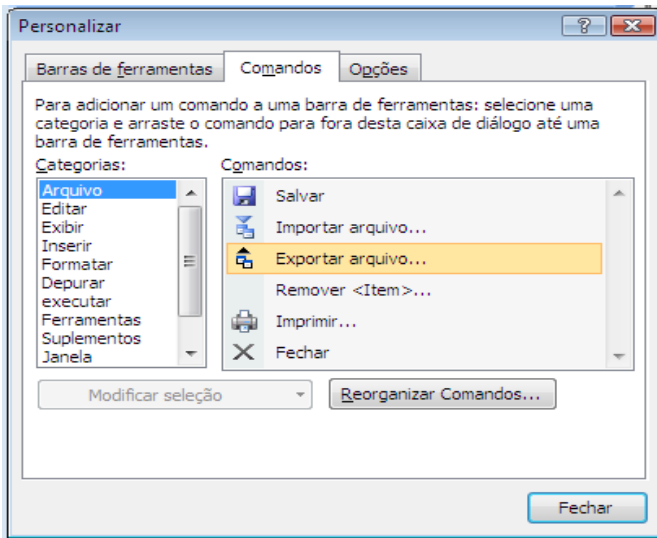


A1.2.2 Atribuindo Macros a uma Barra de Ferramentas

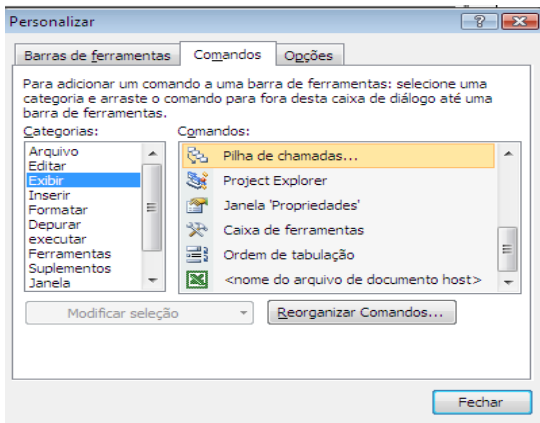
Você pode atribuir macros a um botão numa barra de ferramentas, e mais tarde ativar a macro clicando no botão. Para atribuir **DigaAlo** a um botão, abra o menu barra de ferramentas (**Exibir|Barra de Ferramentas|Personalizar...**)



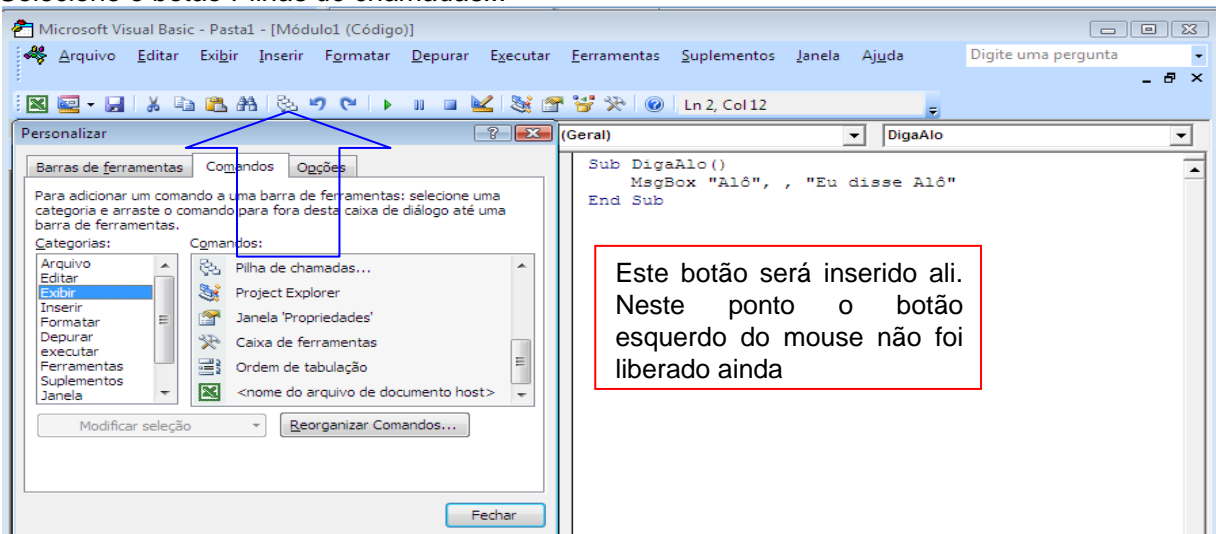
e selecione a aba do painel **Comandos**.



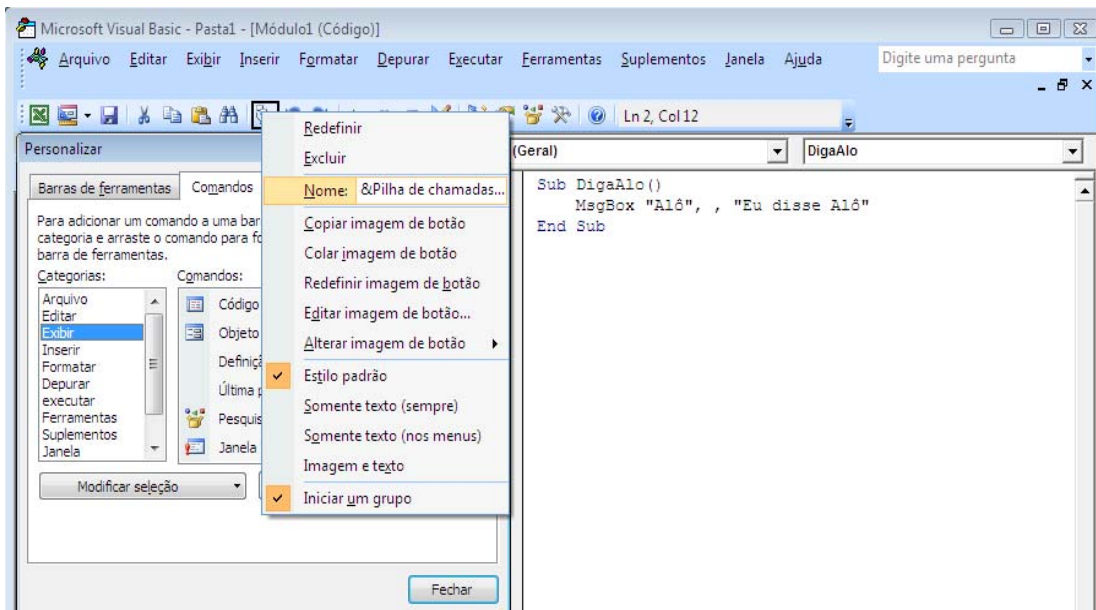
Escolha qualquer **Categorias** e qualquer **Comandos**:
Por exemplo, selecione *Exibir* e *Pilhas de chamadas...* e arraste-o para uma barra de ferramentas.
(Mantenha o botão esquerdo do mouse pressionado enquanto movimentar o mouse.)



Selecione o botão *Pilhas de chamadas...*




Clique com o botão direito do mouse no botão inserido recentemente



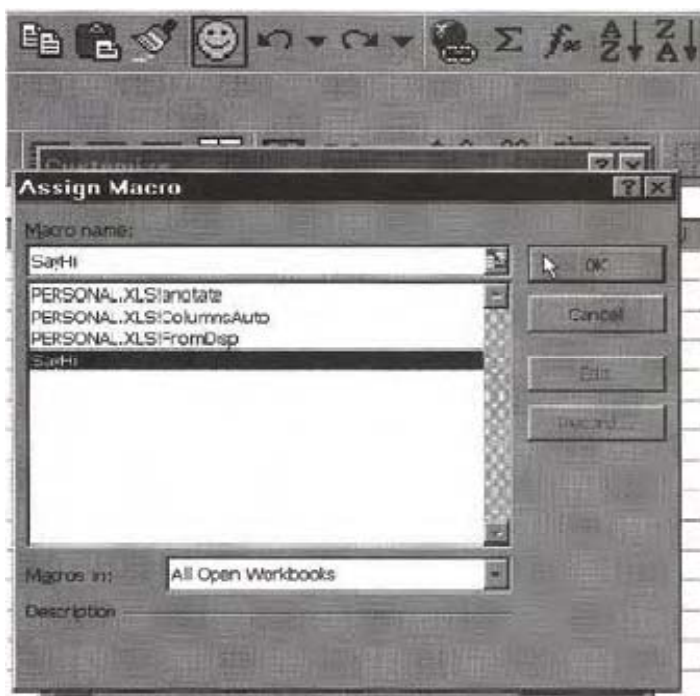
Troque o Nome do botão digitando *Botão Personalizado* na caixa à direita de **N**ome. A seguir clique em **E**ditar imagem de botão... e irá aparecer a janela Editor de botão:



Use a sua criatividade artística e desenhe outro botão do seu gosto. Eu fiz assim: . Você pode também **A**lterar imagem de botão. Feito o que você desejar, selecione **A**ssignMacro no menu.



Selecione nossa macro da lista e clique no botão **OK**. Clique o botão **Close**. **DigaAlo** é atribuída agora ao botão *smiling* na barra de ferramentas.



A imagem no botão e a ferramenta dica atribuída a ele pode também ser personalizada. Para detalhes refira-se ao arquivo Ajuda do Excel.

A1.3 User Output e a Função MsgBox

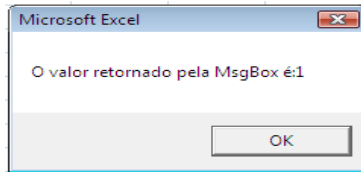
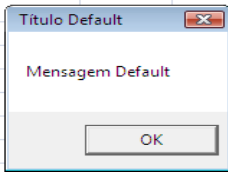
A função **MsgBox** mostra uma mensagem na tela e retorna um valor baseado no botão. Algumas das diferentes opções disponíveis com esta função estão demonstradas nas seguintes macros:

```
Sub MsgBoxDefault()
Dim Temp As Integer
Temp = MsgBox("Mensagem Default ", , _
    " Título Default ")
```

```

    MsgBox "O valor retornado pela MsgBox é: _
    " & Temp
End Sub

```

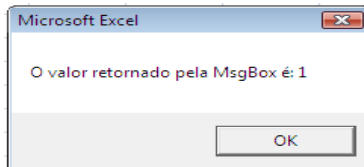
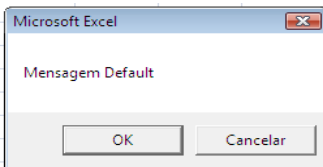


Observação: A configuração default da **MsgBox** produz um botão **OK**. O título Default é "Microsoft Excel." Clicando o botão **OK** faz a **MsgBox** retornar o valor 1.

```

Sub MsgBoxOKCancel()
    Dim Temp As Integer
    Temp = MsgBox("Mensagem Default", vbOKCancel)
    MsgBox "O valor retornado pela MsgBox é: _
    " & Temp
End Sub

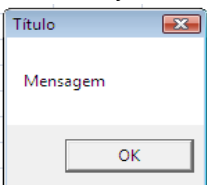
```



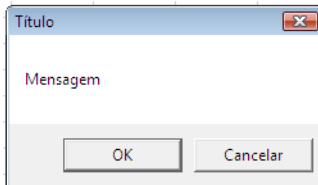
Como notado anteriormente, o segundo argumento da **MsgBox** determina quais botões serão mostrados. Esta encarnação da macro demo usa a constante **vbOKCancel** para produzir os dois botões **OK** e **Cancel**. Note que se o botão **Cancelar** for clicado, a **MsgBox** retorna o valor 2.

O que segue é uma lista de algumas constantes que podem ser usadas como o segundo argumento da **MsgBox**, juntamente com a caixa de mensagem elas produzem:

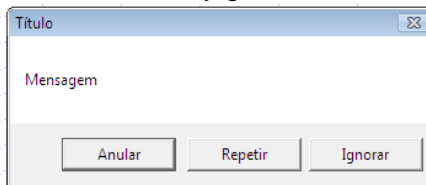
VbOKOnly



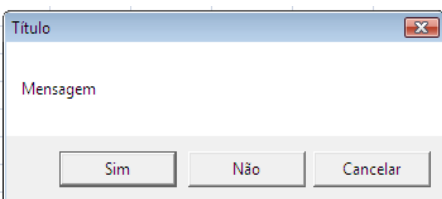
VbOKCancel



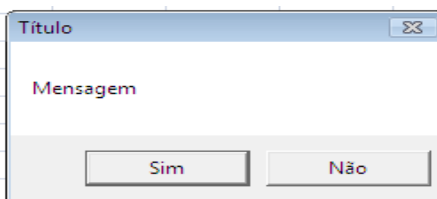
VbAbortRetryIgnore



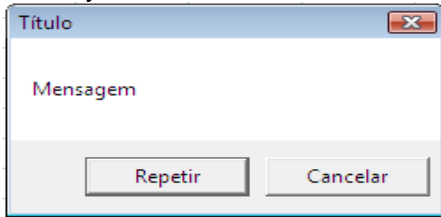
VbYesNoCancel



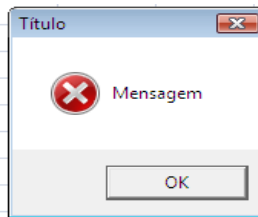
VbYesNo



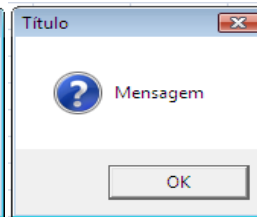
VbRetryCancel



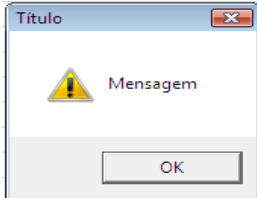
VbCritical



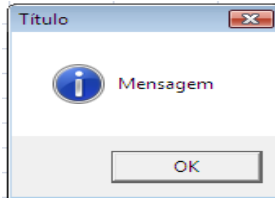
VbQuestion



VbExclamation



VbInformation



Os valores que podem ser retornados pela **MsgBox** são como os que seguem:

Constante	Valor	Descrição
vbOK	1	botão OK clicado
vbCancel	2	botão Cancel clicado
vbAbort	3	botão Abort clicado
vbRetry	4	botão Retry clicado
vbIgnore	5	botão Ignore clicado
vbYes	6	botão Yes clicado
vbNo	7	botão No clicado

A1.4 User Input e a Função InputBox

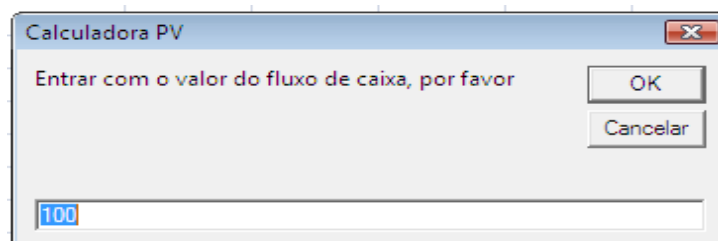
InputBox é uma função interna do VBA usada para obter informação textual do usuário para o interior de uma variável numa sub-rotina (uma macro com um outro nome ...). As operações da função estão demonstradas na seguinte macro, calculadora do valor presente. A macro **CalculadorVP** calcula , onde **FC** é um número entrado pelo usuário:

```
Sub CalculadorVP()
Dim FC
FC = InputBox("Entrar com o valor do fluxo de caixa, por favor", "Calculadora
VP", "100")
MsgBox "O valor presente de " & FC & " a 5% durante 10 períodos é: " & _
Round(Application.PV(0.05, 10, -FC), 2), vbInformation, "Calculadora PV"
End Sub
```

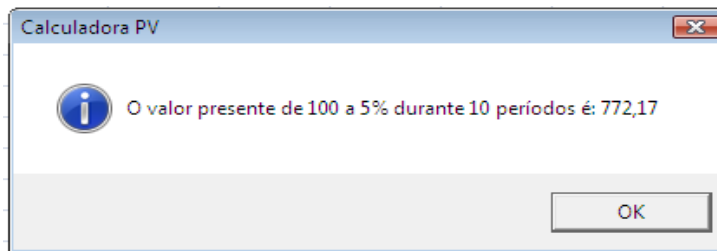
Note a sintaxe:

```
FC = InputBox("Entrar com o valor do fluxo de caixa, por favor", "Calculadora
VP", "100")
```

- "Entrar ... Por favor," o primeiro argumento para **InputBox**, é a mensagem a ser mostrada.
- "Calculadora VP," o segundo argumento, é o título para a caixa.
- "100," o terceiro argumento, é a string default a ser colocada na caixa.
- Se você não trocar o valor default por algum outro valor, rodando a macro deverá resultar no seguinte:



Neste momento você pode trocar "100" com algum outro número. (Neste exemplo, optamos por mantê-lo.)
 Clicando no botão OK produz a seguinte caixa:



A macro também introduz um novo operador VBA, o **&**. Este operador é usado para concatenar (combinar) seus dois operandos numa única string de caracteres. Ela merece uma pequena função demo por si própria. Note que operandos não string são convertidos em strings.

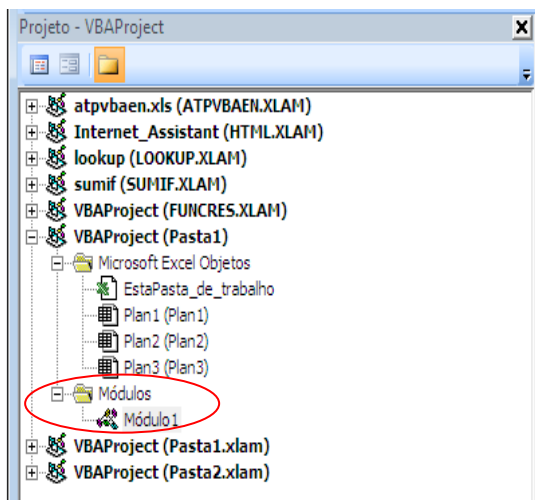
```
Function ConcatDemo(Parameter1, Parameter2)
    ConcatDemo = Parameter1 & Parameter2
End Function
```

	A	B	C	D
1	ConcatDemo em Ação			
2				
3	Parâmetro1	Parâmetro2		
4	1	2	12	<--=ConcatDemo(A4;B4)
5	Ber	tolo	Bertolo	<--=ConcatDemo(A5;B5)
6	Ber	1	Ber1	<--=ConcatDemo(A6;B6)
7	tolo	2	tolo2	<--=ConcatDemo(A7;B7)

A1.5 Módulos

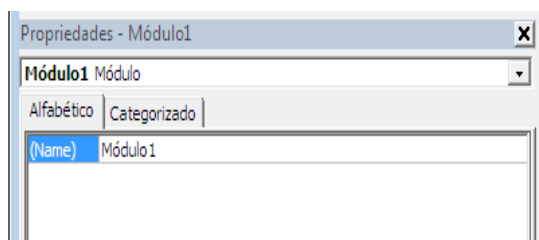
O VBA organiza as funções definidas pelo usuário e sub-rotinas em unidades chamadas módulos. Poderá (e algumas vezes, deverá) ter mais do que um módulo num projeto VBA (isto é, a parte da pasta que tem nossas funções e sub-rotinas). Os Módulos têm nomes: Por default o VBA usa o nome "Module" seguido por um número para indicar o nome do módulo, mas você poderá achar útil (como temos feito na pasta acompanhando este livro) para dar-lhes um nome um pouco mais descritivo.

Para re-nomear um módulo (no editor VBA), selecione o módulo no painel **Project Explorer**.

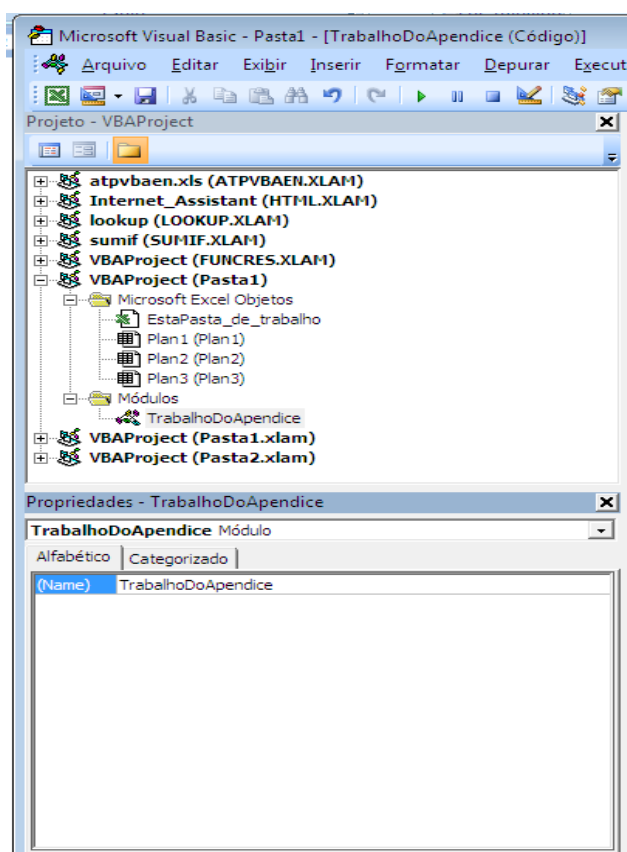


Se o painel **Project Explorer** não estiver visível, selecione o **Project Explorer** do menu **Exibir**.

Uma vez selecionado um módulo, a lista de propriedades do módulo deverá aparecer no painel propriedades. Clique no nome do módulo (ela deverá ser a única propriedade disponível) e mude-a (use uma palavra somente, e somente dígitos e caracteres alfabéticos).



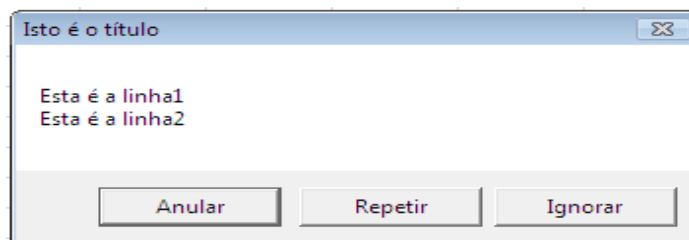
Uma vez pressionada a tecla [Enter] o nome é mudado. Note a mudança no *Project Explorer*.



Exercícios

1. Escreva uma macro que mostre a seguinte caixa de mensagem. A caixa de mensagem deverá ficar no topo de todas as outras janelas, e deverá evitar o usuário de fazer qualquer coisa em qualquer aplicação até que um dos botões seja clicado.

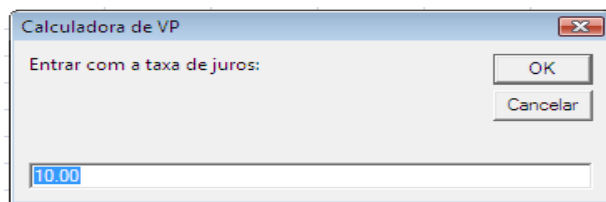
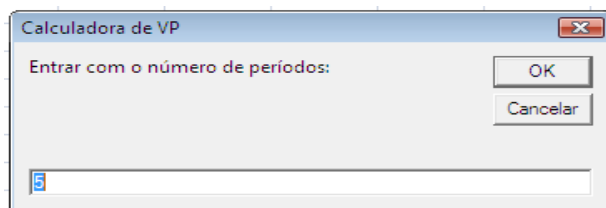
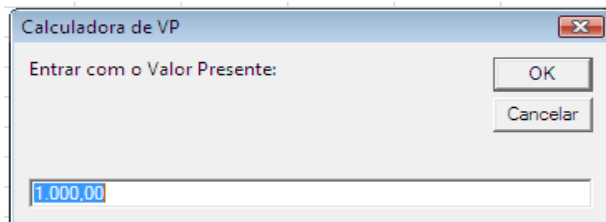
Sugestão Você precisa usar algumas opções da **MsgBox** que não foram cobertas no texto; use o sistema de Ajuda VBA.



2. Escreva uma macro calculadora de valor presente semelhante aquela uma que aparece na [seção A1.4](#). Entretanto — como ilustrado aqui — sua macro deverá perguntar ao usuário pelos valores dos fluxos de

caixa, da taxa de juros, e do número de períodos. Ela deverá então mostrar o resultado numa caixa de mensagem. Valores default lógicos deverão ser fornecidos para todos os argumentos. Não use a função Excel **PV**; escreva a sua própria função valor presente e use-a. Um lembrete:

$$VP(FC, i, n) = \sum_{k=1}^n \frac{FC}{(1 + i)^k}$$



	A	B	C
1			
2			
3	100		
4	10,0%		
5	5		
6	R\$ 379,08	<--=VP(A4;A5;-A3)	
7			
8			
9			
10			
11			
12			
13			
14			
15			

Você pode usar a função **VP** fornecida pelo Excel, como fizemos, para verificar a exatidão de sua macro.

3. Reescreva a macro do exercício anterior de modo que a interface do usuário seja como a demonstrada na seguinte *screen shots*. Algumas das funções necessárias para escrever a macro não foram cobertas no texto. Usamos as funções seguintes:

- **Val**— Uma função usada para converter uma string de dígitos para um número.
- **Left**— Uma função usada para retornar a parte esquerda de uma string.
- **Right**— Uma função usada para retornar a parte direita de uma string.
- **FormatPercent**— Uma função usada para formatar um número.
- **FormatCurrency**— Uma função usada para formatar um número.

Mais informações sobre estas funções estão disponíveis no arquivo Ajuda do VBA; recomendamos você usá-las.

Nova Calculadora de VP

Entrar com o Valor Presente:

OK

Cancelar

100,25

Nova Calculadora de VP

O Valor Presente entrado foi: 100,25

Entrar com o número de períodos:

OK

Cancelar

13

Nova Calculadora de VP

O Valor Presente entrado foi: 100,25

O número de Períodos entrado foi: 13

Entrar com a taxa de juros:

OK

Cancelar

12,45

	A	B	C
1			
2			
3	R\$ 100,25		
4	12,45%		
5	13		
6	R\$ 630,06	<--=VP(A4;A5;-A3)	
7			
8			
9			
10			
11			
12			
13			

Nova Calculadora de VP

O Novo VP é = R\$ 630,06

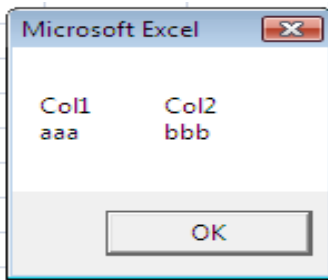
OK

Observação: Seu computador poderá mostrar um símbolo de moeda diferente.

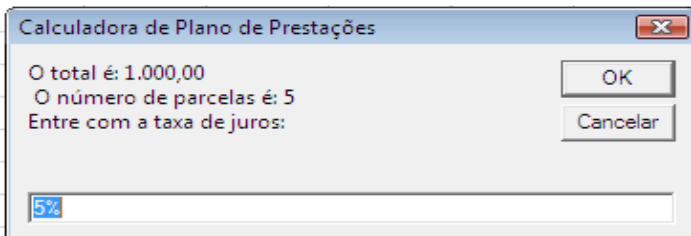
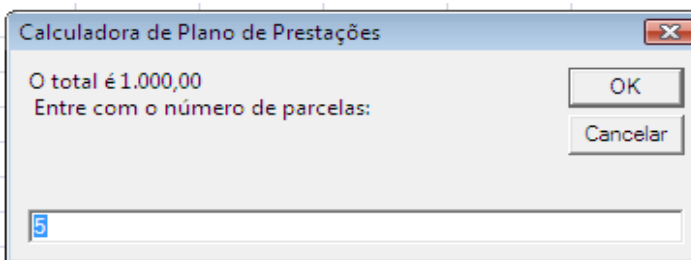
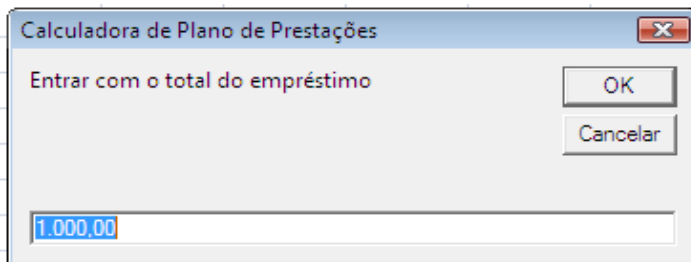
- Reescreva a macro do exercício anterior de modo que ele trate apropriadamente com o botão **Cancel**.
 - Uma versão simples da nova macro abortará a macro se **Cancel** for clicado em qualquer estágio.
 - Uma versão mais sofisticada da nova macro permitirá o usuário entrar novamente com os dados do ponto de partida.
 - A versão mais sofisticada da nova macro permitirá entrar novamente com os dados usando os dados antigos como default.
- Escreva uma macro calculadora de plano de prestações; a macro perguntará ao usuário pela soma do empréstimo, o número de parcelas, e a taxa de juros. Assuma que o pagamento seja no final do período. A saída deverá se parecer com o exemplo seguinte:

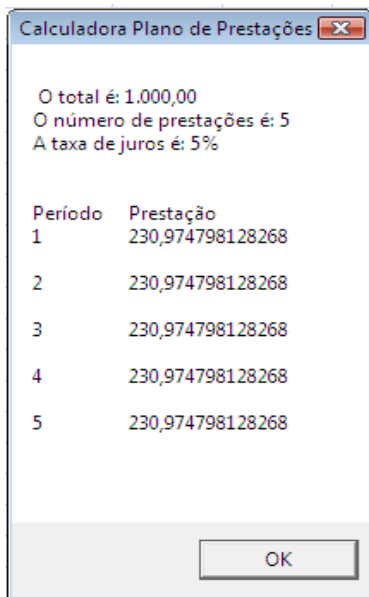
```
Sub StringConcat()
    Dim S As String
    S = "Col1" & Chr(9) & "Col2" & Chr(9) & " _
    is the Tab
    S = S & Chr(13) & "aaa" & Chr(9) & "bbb"
```

```
MsgBox S  
End Sub
```

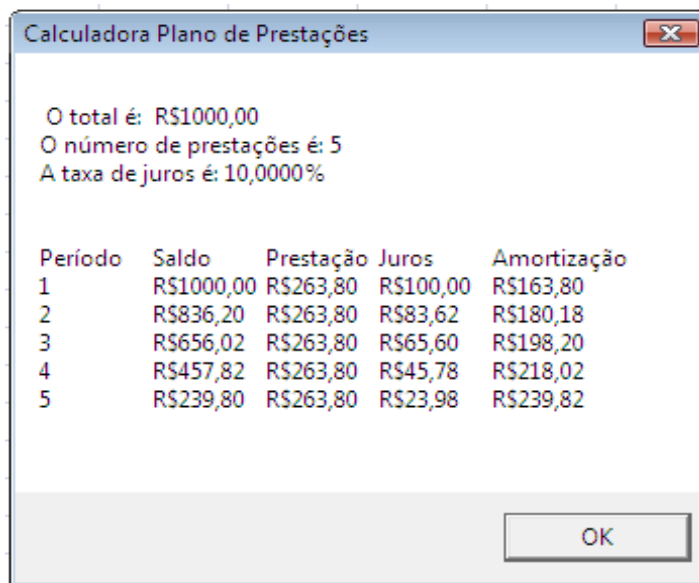


Aqui está um exemplo da macro solicitada em ação:





6. Reescreva a macro - Calculadora de Plano de Prestações - de modo que ela mostre os pagamentos segregados em juros e pagamentos do capital (amortizações). A caixa de entrada no exemplo foi removida por solidez.



7. Reescreva a macro, calculadora de plano de pagamentos; a macro perguntará pela soma do empréstimo, os pagamentos, e a taxa de juros. Assuma o pagamento no final do período. A macro deverá mostrar os pagamentos segregados em juros e amortizações. Obviamente, o último pagamento pode ser menor (mas não maior) que o pagamento fornecido pelo usuário. A saída parecerá com o seguinte exemplo (caixas de entrada removidas por solidez).

Payment Schedule Calculator

Sum of loan £1,000.00 Payment £240.00 Rate 10.0000%

Period	Balance	Payment	Interest	Capital
1	£1,000.00	£240.00	£100.00	£140.00
2	£860.00	£240.00	£86.00	£154.00
3	£706.60	£240.00	£70.60	£169.40
4	£536.60	£240.00	£53.66	£186.34
5	£350.26	£240.00	£35.03	£204.97
6	£145.29	£159.81	£14.53	£145.29

OK

8. Uma versão um pouco mais complicada da macro no exercício 7 produziria os seguintes resultados melhores apresentados. Escreva esta versão da macro.

9. Um plano móvel de pagamentos envolve pagamentos que variam por um percentual fixo durante a vida do empréstimo. Escreva a versão móvel do plano de pagamento da calculadora do exercício 8. Além de todas as entradas descritas anteriormente, a macro obterá uma taxa de pagamento de troca (em porcentagem) do usuário. Isto é o que ela deverá se parecer numa ação.

Observação Uma rápida olhada no arquivo Ajuda da função **Format** deverá ser vantajoso neste momento.

Payment Schedule Calculator

Sum of loan £1,000.00 Payment £240.00 Rate 10.0000%

Period	Balance	Payment	Interest	Capital
1	£1,000.00	£0,240.00	£0,100.00	£0,140.00
2	£0,860.00	£0,240.00	£0,086.00	£0,154.00
3	£0,706.60	£0,240.00	£0,070.60	£0,169.40
4	£0,536.60	£0,240.00	£0,053.66	£0,186.34
5	£0,350.26	£0,240.00	£0,035.03	£0,204.97
6	£0,145.29	£0,159.81	£0,014.53	£0,145.29

OK

9. Um plano de pagamento deslizante (sliding) envolve pagamentos que variam por uma porcentagem fixa durante a vida do empréstimo. Escreva uma versão prestação deslizante do plano de prestações da calculadora do exercício 8. Além disso, para todas as entradas descritas anteriormente, a macro pedirá uma taxa de variação de prestação (como porcentagem) ao usuário. Isto é o que deve parecer em ação.

Payment Schedule Calculator

Sum of loan £10,000.00 Payment £2,000.00 Rate 10.00% Payment rate 10.00%

Period	Balance	Payment	Interest	Capital
1	£10,000.00	£2,000.00	£1,000.00	£1,000.00
2	£8,900.00	£2,200.00	£890.00	£1,380.00
3	£7,700.00	£2,420.00	£770.00	£1,650.00
4	£6,380.00	£2,660.00	£638.00	£2,057.00
5	£4,930.00	£2,920.00	£493.00	£2,526.90
6	£1,464.10	£1,610.51	£146.41	£1,464.10

OK