

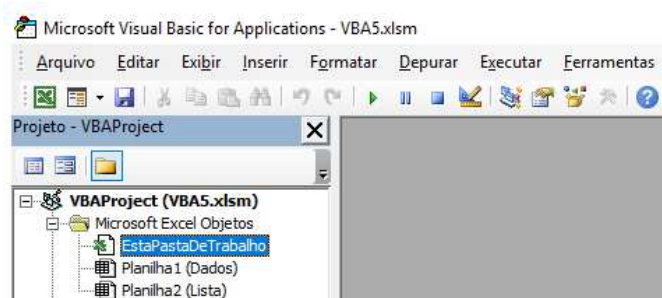
LIÇÃO 1 - EXIBIR ABERTURA POR SEGUNDOS QUANDO A PASTA FOR ABERTA

- Preparar o código para exibir o formulário **Abertura** quando a pasta for aberta.
- Criar um procedimento para fechar o formulário **Abertura** e abrir **Vendas**.
- Criar um procedimento para executar este procedimento após abrir **Abertura**.

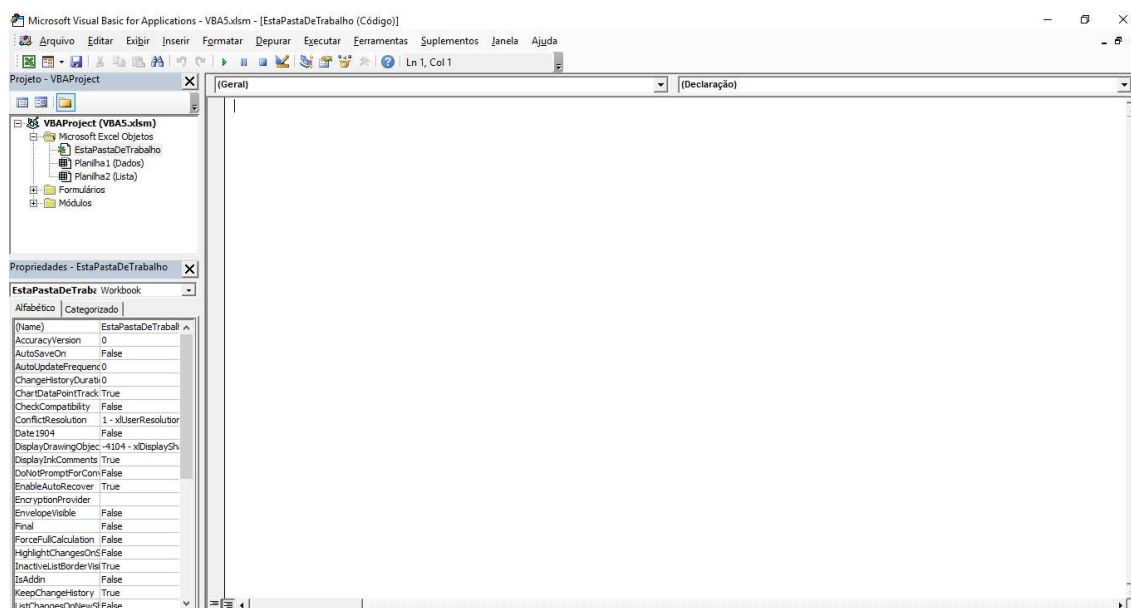
• Exibir o formulário Abertura quando a pasta for aberta



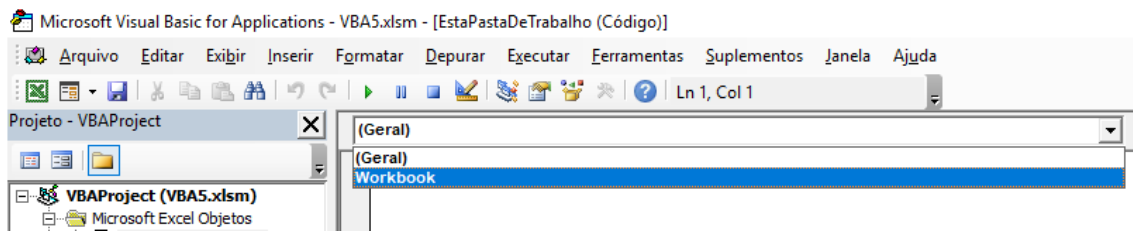
1. Abra a pasta **VBA5** e habilite seu conteúdo.
2. Pressione **ALT+F11** para abrir o editor.
3. Dê um clique duplo sobre **EstaPasta_de_trabalho** em **Microsoft Excel Objetos**.



4. Será aberta a janela de código deste objeto.



5. Na janela de código selecione **Workbook** em **Objeto**.



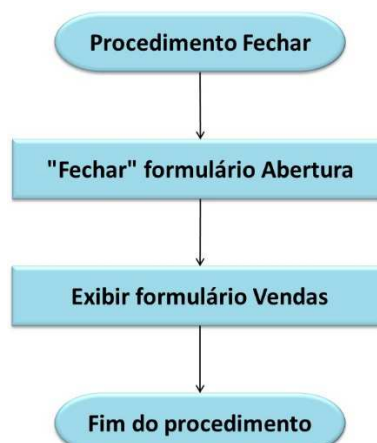
6. O procedimento para o evento padrão **Workbook_Open** é inserido.

```
Private Sub Workbook_Open()  
|  
End Sub
```

7. Digite neste procedimento as linhas destacadas em negrito (as linhas em verde representam comentários e não precisam ser digitadas).

```
Private Sub Workbook_Open()  
'Exibir o formulário Abertura  
Abertura.Show  
End Sub
```

- **Fechar o formulário Abertura e exibir o formulário Vendas**



1. Expanda **Módulos** e dê um clique duplo em **Códigos** para abrir a janela de código deste módulo.
2. Digite nesta janela o código do procedimento **Fechar** representado pelas linhas destacadas em negrito. As linhas em verde representam comentários e não precisam ser digitadas.

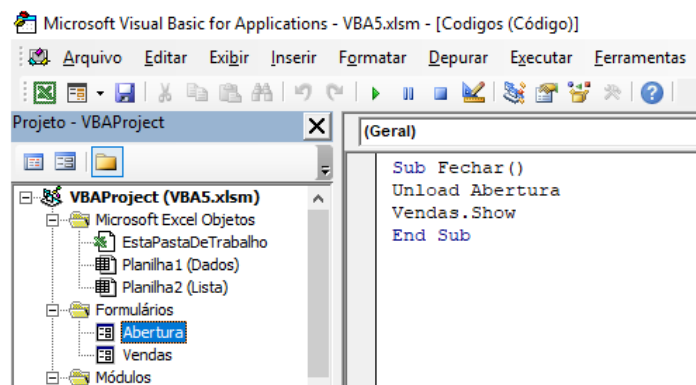
```
Sub Fechar()  
'Fechar o formulário Abertura  
Unload Abertura  
'Exibir o formulário Vendas  
Vendas.Show  
End Sub
```

3. Salve as alterações.

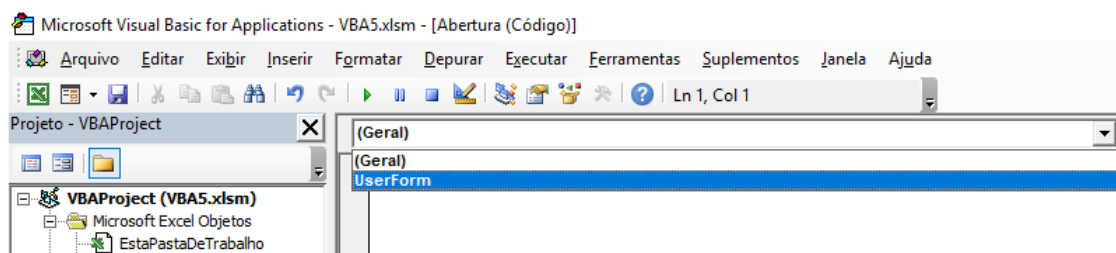
- **Executar o procedimento Fechar após abrir Abertura**



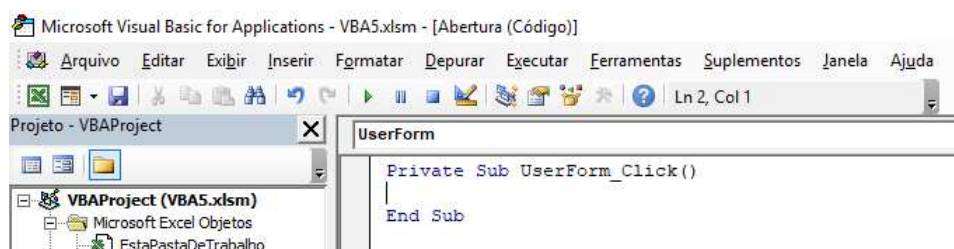
1. Selecione o formulário **Abertura** na janela **Projeto** e pressione **F7** para abrir sua janela de código.



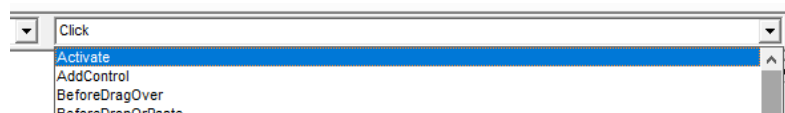
2. Selecione **UserForm** em **Objeto**.



3. Será criado um procedimento para o evento padrão deste objeto (**Click()**).



4. Selecione **Activate** em **Procedimento**.



5. Será criado o procedimento para o evento **UserForm_Activate**.

```

Private Sub UserForm_Activate()
|
End Sub
  
```

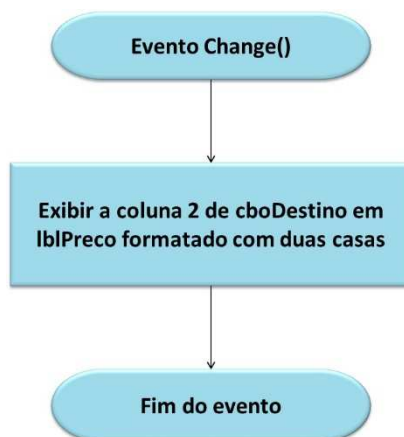
6. Digite no procedimento **UserForm_Activate** as linhas destacadas em negrito.

```
Private Sub UserForm_Activate()  
'Executa o procedimento Fechar 4 segundos após  
'o formulário Abertura ser ativado  
Application.OnTime Now + TimeValue("00:00:04"), "Fechar"  
End Sub
```

7. Salve as alterações.

LIÇÃO 2 - CODIFICANDO O USERFORM VENDAS

- Selecionar um destino em **cboDestino** e exibir o preço formatado em **lblPreco**



1. Selecione a caixa de combinação **cboDestino** e pressione **F7** para exibir o procedimento do evento padrão deste tipo de controle (**Change()**).

2. Para alterar a propriedade **Caption** de **lblPreco** com o valor correspondente à propriedade **Value** da caixa de combinação **cboDestino** formatado com duas casas decimais, complete o código do procedimento **cboDestino_Change** com as linhas abaixo. Como **Caption** é a propriedade padrão de rótulos e a propriedade **Value** é a propriedade padrão das caixas de combinação podemos omiti-las.

```
Private Sub cboDestino_Change()  
'SE cboDestino vazio  
If Then  
'Exibir 0,00 em lblPreco  
  
'SENÃO cboDestino vazio  
Else
```

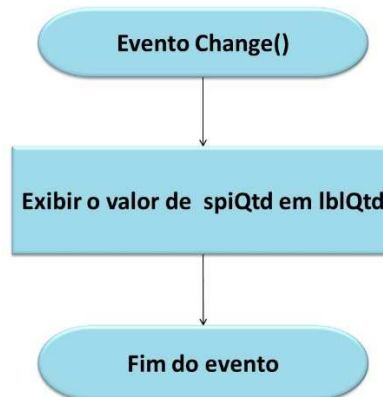
```

'Exibir coluna 2 de cboDestino em lblPreco
'formatado com duas casas
    lblPreco = FormatNumber(cboDestino, 2)
'FIM SE cboDestino vazio
End If
'Executar Total

'Fim do procedimento
End Sub

```

- **Exibir o valor de spiQtd em lblQtd**



1. Localize na janela de código o evento **spiQtd_Change()**.
2. Para alterar a propriedade **Caption** de **lblQtd** com o valor correspondente à propriedade **Value** do botão de rotação **spiQtd** complete o código do procedimento **spiQtd_Change** com as linhas abaixo. Como **Caption** é a propriedade padrão de rótulos e a propriedade **Value** é a propriedade padrão dos botões de rotação podemos omiti-las.

```

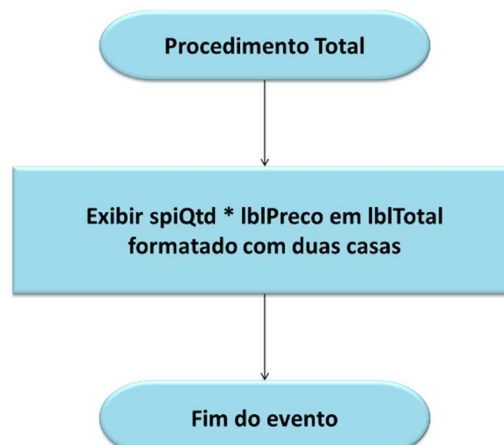
Private Sub spiQtd_Change()
'Exibir o valor de spiQtd em lblQtd
lblQtd = spiQtd
'Executar Total

'Fim do procedimento
End Sub

```

- **Calcular o total e exibir o valor formatado em lblTotal**

Como o total depende do **destino** e da **quantidade** teremos que alterar, com o mesmo código, o evento **Change** de **cboDestino** e **spiQtd**. Já que o código será repetido, vamos criar um procedimento do tipo **Sub** e desta forma defini-lo uma vez só.



1. Na janela **Código** do formulário pressione **CTRL+HOME** para levar o cursor para o início da janela.
2. Digite o código do procedimento **Total**. As propriedades padrões dos controles estão sendo omitidas.

```
Sub Total()
'Exibir spiQtd * lblPreco em lblTotal
'formatado com duas casas
lblTotal = FormatNumber(spiQtd * lblPreco, 2)
End Sub
```

- **Total conforme Destino: lblTotal conforme cboDestino**

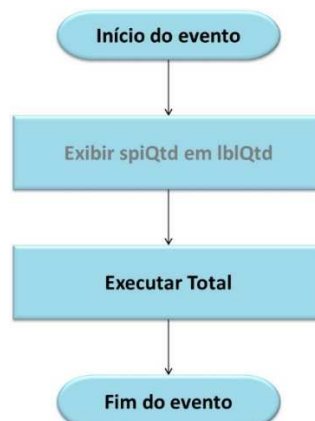


1. Localize na janela de código o evento **cboDestino_Change()**.
2. Complete o código deste evento com as linhas destacadas em **negrito**.

```
Private Sub cboDestino_Change()
'SE cboDestino vazio
If Then
'Exibir 0,00 em lblPreco

'SENÃO cboDestino vazio
Else
'Exibir coluna 2 de cboDestino em lblPreco
'formatado com duas casas
    lblPreco = FormatNumber(cboDestino, 2)
'FIM SE cboDestino vazio
End If
'Executar Total
Total
'Fim do procedimento
End Sub
```

- **Total conforme quantidade: lblTotal conforme spiQtd**

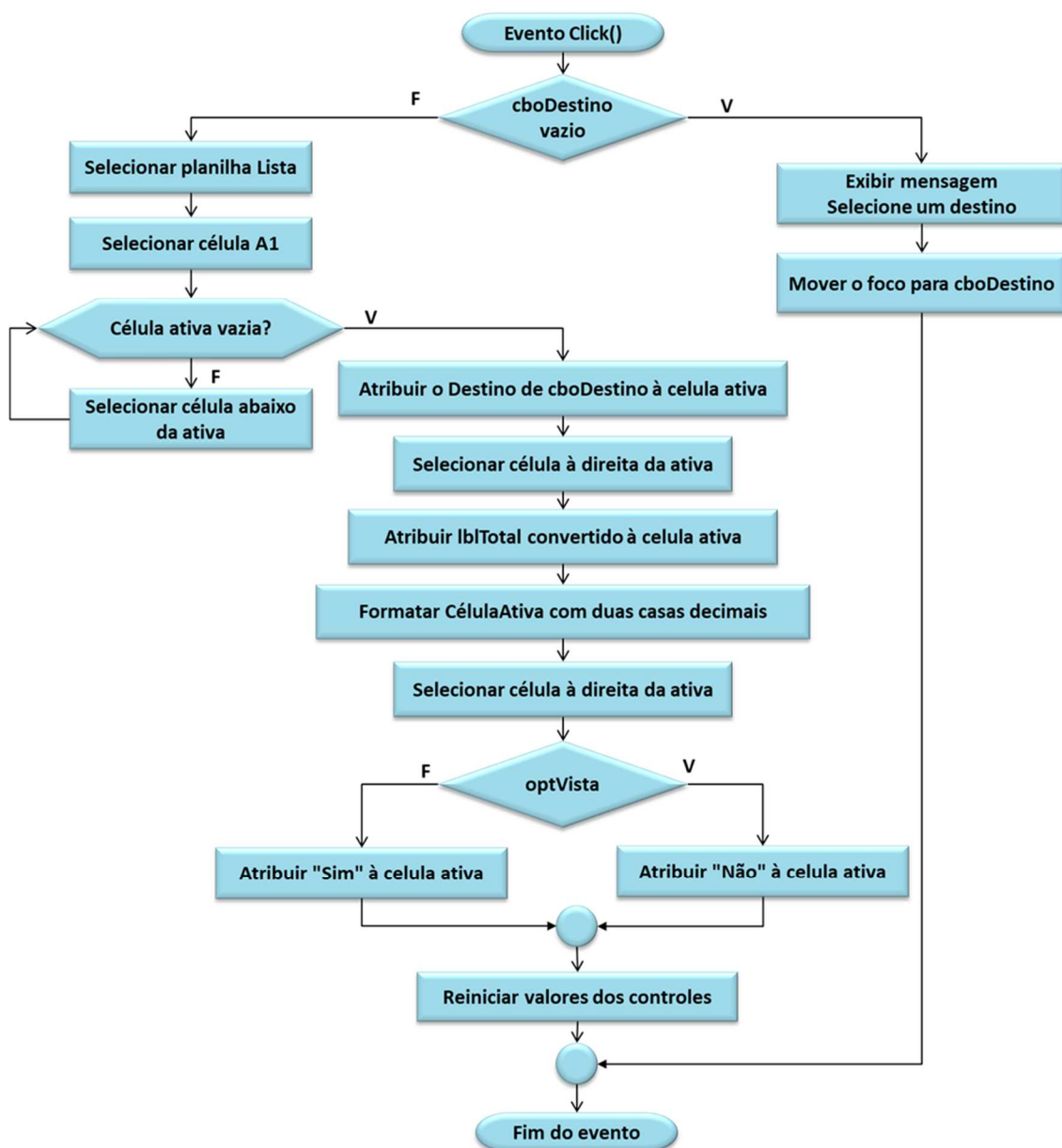


1. Localize na janela de código o evento **spiQtde_Change()**.
2. Complete o código deste evento com as linhas destacadas em negrito.

```
Private Sub spiQtde_Change()  
  'Exibir o valor de spiQtde em lblQtde  
  lblQtde = spiQtde  
  'Executar Total  
  Total  
  'Fim do procedimento  
End Sub
```

LIÇÃO 3 - TRANSFERINDO OS DADOS DO FORMULÁRIO PARA A PLANILHA LISTA

- Verificar se **Destino** foi selecionado.
- Selecionar a célula **A1** da planilha **Lista**.
- Selecionar uma célula vazia abaixo de **A1**.
- Transferir os dados para as células.
- Reiniciar os valores dos controles.

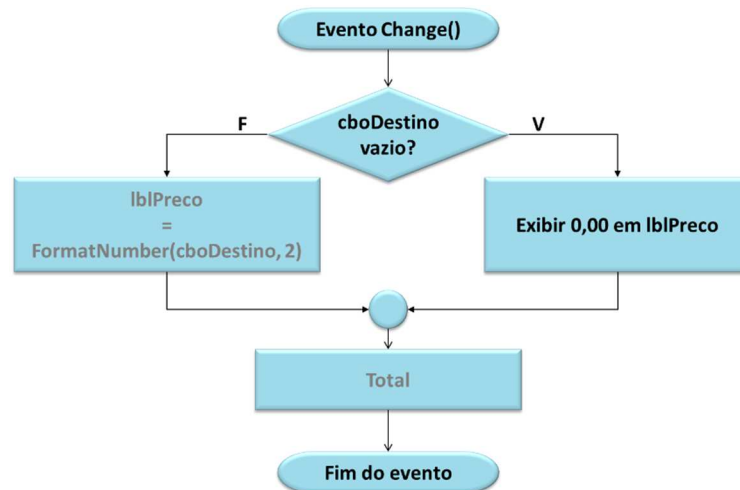


1. Localize na janela de código o evento **cmdCadastra_Click ()**.
2. Complete o código deste evento com as linhas destacadas em negrito.

```
Private Sub cmdCadastra_Click()  
  'SE cboDestino vazio  
  If IsNull(cboDestino) Then  
    'Exibir mensagem  
    MsgBox "Selecione um destino"  
    'Mover o foco para cboDestino  
    cboDestino.SetFocus  
    'SENÃO cboDestino vazio  
  Else  
    'Selecionar a planilha Lista  
    Sheets("Lista").Select  
    'Selecionar a célula A1  
    Range("A1").Select  
    'Inicio do bloco: até a célula ativa estar vazia  
    Do Until IsEmpty(ActiveCell)  
      'Selecionar a célula abaixo da ativa  
      ActiveCell.Offset(1, 0).Select  
    'Fim do bloco: até a célula ativa estar vazia  
    Loop  
    'Atribuir o valor de cboDestino à célula ativa  
    ActiveCell = cboDestino.Text  
    'Selecionar célula à direita da ativa  
    ActiveCell.Offset(0, 1).Select  
    'Atribuir lblTotal convertido à célula ativa  
    ActiveCell = CDBl(lblTotal)  
    'Formatar a célula ativa com duas casas decimais  
    ActiveCell.NumberFormat = "#,##0.00"  
    'Selecionar célula à direita da ativa  
    ActiveCell.Offset(0, 1).Select  
    'SE optVista = True  
    If optVista Then  
      'Atribuir "Não" à célula ativa  
      ActiveCell = "Não"  
    'SENÃO optVista = True  
    Else  
      'Atribuir "Sim" à célula ativa  
      ActiveCell = "Sim"  
    'Fim SE optVista = True  
    End If  
    'Reiniciar valores dos controles  
    cboDestino = Null  
    optVista = True  
    lblQtd = "1"  
    spiQtd = 1  
    lblTotal = "0,00"  
    lblPreco = "0,00"  
    'Fim cboDestino vazio  
  End If  
  'Fim do procedimento  
End Sub
```

3. Salve as alterações.

- **Prevenindo erro de execução Evento `cboDestino_Change()`**



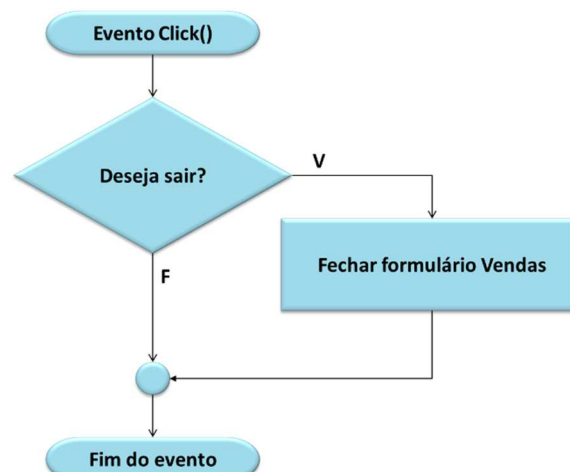
1. Localize na janela de código o evento **`cboDestino_Change()`**.
2. Complete o código deste evento com as linhas destacadas em negrito.

```

Private Sub cboDestino_Change()
'SE cboDestino vazio
If IsNull(cboDestino) Then
'Exibir 0,00 em lblPreco
  lblPreco = "0,00"
'SENÃO cboDestino vazio
Else
  'Exibir coluna 2 de cboDestino em lblPreco
  'formatado com duas casas
  lblPreco = FormatNumber(cboDestino, 2)
  'FIM SE cboDestino vazio
End If
'Executar Total
Total
'Fim do procedimento
End Sub
  
```

3. Salve as alterações.

- **Fechar o formulário ao clicar em Sair (`cmdFecha`)**

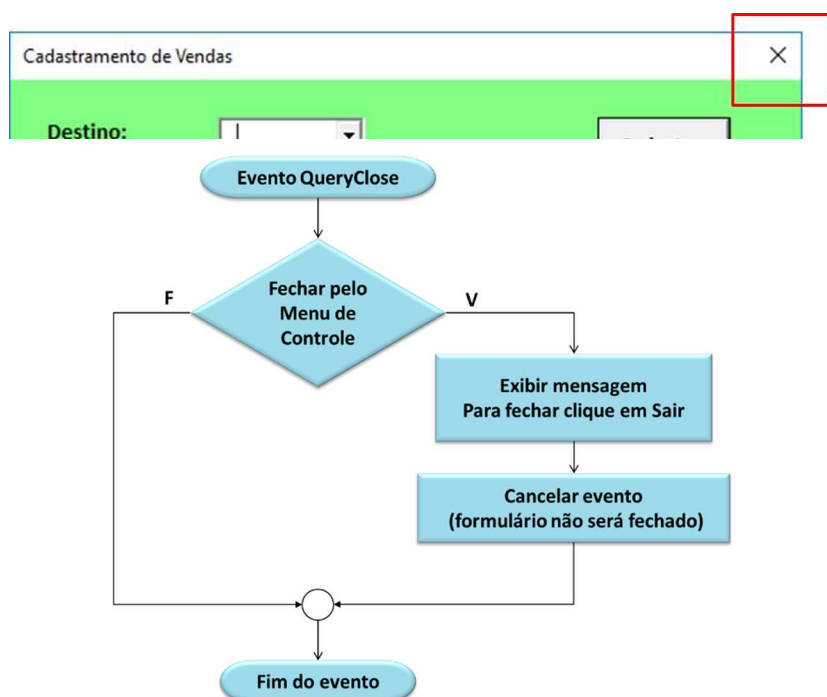


1. Localize na janela de código o evento **cmdFecha_Change()**.
2. Complete o código deste evento com as linhas destacadas em negrito.

```
Private Sub cmdFecha_Click()
'SE Deseja sair
If MsgBox("Deseja sair?", vbYesNo) = vbYes Then
'Remover o formulário ativo da memória
'(fechar o formulário)
    Unload Me
'FIM SE Deseja sair
End If
'Fim do procedimento
End Sub
```

3. Salve as alterações.

- **Desabilitando o botão Fechar do Menu de Controle da janela**



1. Localize na janela de código o evento **cmdFecha_QueryClose()**.
2. Complete o código deste evento com as linhas destacadas em negrito.

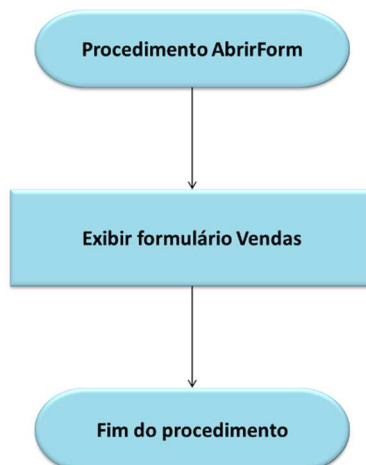
```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
'SE Fechar pelo menu de controle
If CloseMode = vbFormControlMenu Then
'Exibir mensagem
    MsgBox "Para fechar clique em Sair"
'Cancelar o evento
    Cancel = True
'FIM SE Fechar pelo menu de controle
End If
'Fim do procedimento
End Sub
```

3. Salve as alterações e pressione **F5** para exibir o formulário.
4. Teste os códigos digitados anteriormente.

LIÇÃO 4 - EXIBINDO O FORMULÁRIO VENDAS

Para que o formulário seja aberto a partir de um botão desenhado na planilha devemos criar um procedimento contendo a instrução que exhibe este formulário para, em seguida, associar a execução deste procedimento ao botão da planilha.

• Criar o procedimento para exibir o formulário Vendas



1. Dê um clique duplo sobre o módulo **Códigos** para abrir sua janela de código.
2. Insira, na janela de código deste módulo, o código do procedimento **AbrirForm** digitando as linhas destacadas em negrito.

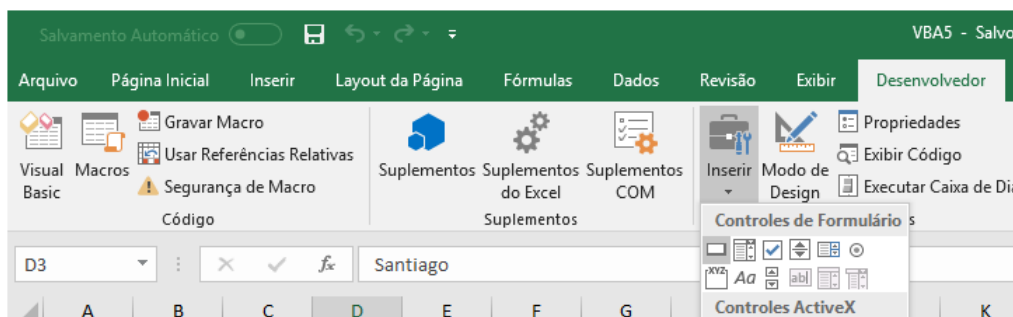
```

Sub AbrirForm()
'Exibe o formulário Vendas
Vendas.Show
End Sub
  
```

3. Salve as alterações e pressione **ALT+F11**.

• Criar um botão na planilha para executar este procedimento

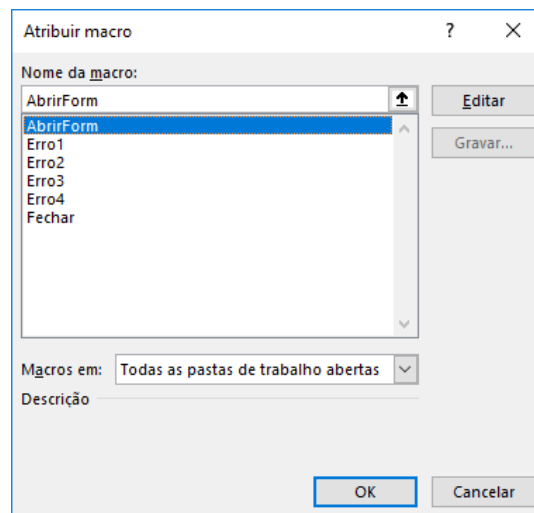
1. Selecione a planilha **Lista** e clique na guia **Desenvolvedor**.
2. Clique no botão **Inserir** do grupo **Controles** e escolha **Botão**.



3. Desenhe um botão na planilha.

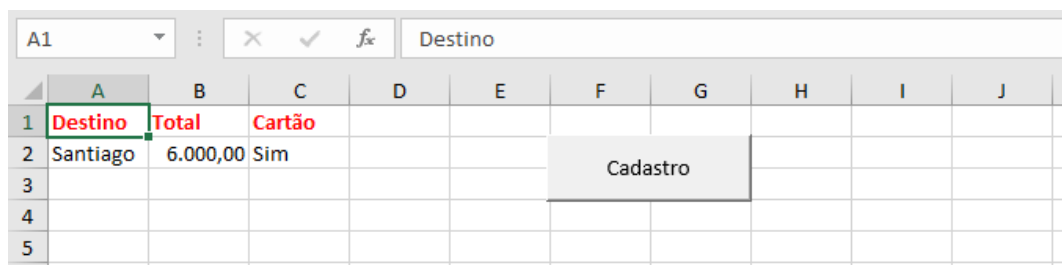
	A	B	C	D	E	F	G
1	Destino	Total	Cartão				
2	Santiago	6.000,00	Sim				
3							
4							
5							
6							

4. Na caixa **Atribuir macro** selecione **AbrirForm**.



5. Clique em **OK** para encerrar o comando e digite **Cadastro** como texto do botão.

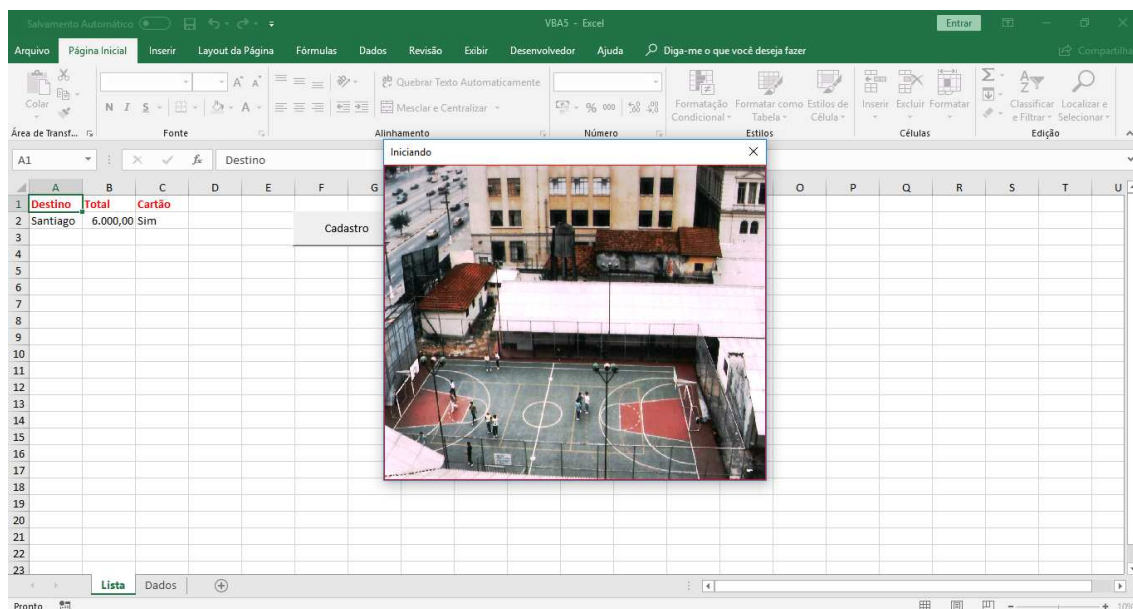
6. Clique em **A1** para encerrar e clique neste botão para exibir o formulário.

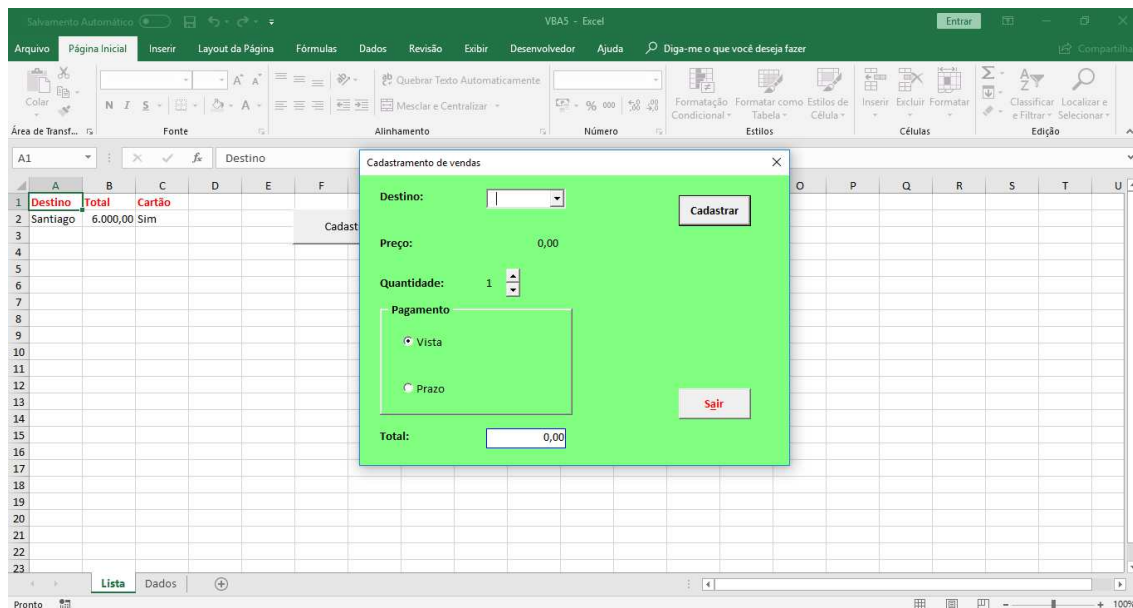


7. Clique em **Cadastro**, clique em **Sair** e confirme.

8. Salve as alterações e feche **VBA5**.

9. Abra novamente **VBA5** e aguarde a exibição dos formulários **Abertura** e **Vendas**.

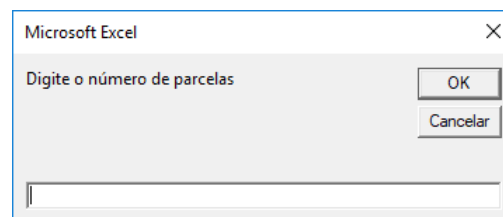




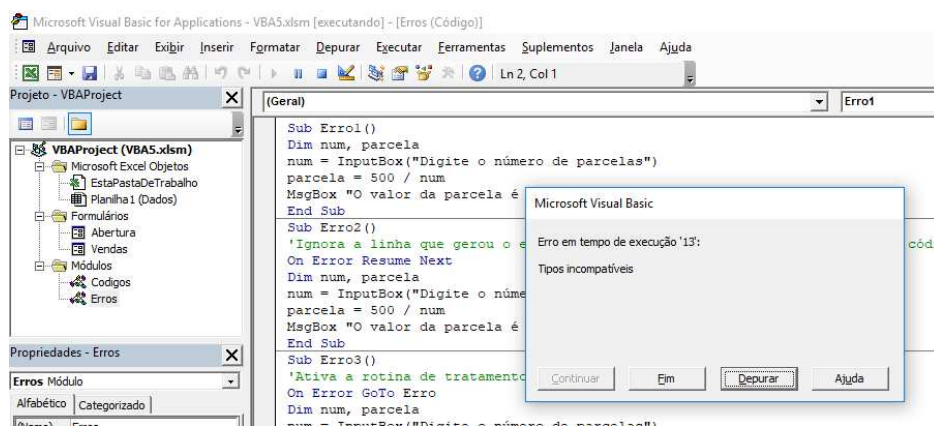
LIÇÃO 5 - MANIPULAÇÃO DE ERROS EM TEMPO DE EXECUÇÃO

• On Error Resume Next

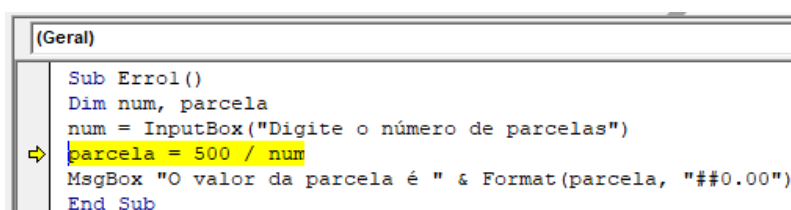
1. Feche o formulário **Vendas** e abra a janela de código do módulo **Erros** no **Editor**.
2. Execute o procedimento **Erro1** e na mensagem **Digite o número de parcelas** clique em **OK**.



3. Esta ação provocará um erro de execução.
4. Na mensagem clique em **Depurar**.



5. A linha que gerou o erro aparece destacada em amarelo.
6. Clique em **Redefinir** para interromper a execução.



7. Complete o código do procedimento **Erro2**.

```
Sub Erro2()  
'Ignora a linha que gerou o erro e continua a execução na próxima linha de código  
On Error Resume Next  
Dim num, parcela  
num = InputBox("Digite o número de parcelas")  
parcela = 500 / num  
MsgBox "O valor da parcela é " & Format(parcela, "##0.00")  
End Sub
```

8. Execute este procedimento. Na mensagem **Digite o número de parcelas** clique em **OK**. Esta ação provocará um erro de execução e a linha inserida anteriormente (**On Error Resume Next**) repassará a execução para a próxima linha e a mensagem **O valor da parcela é** será exibida.

9. Clique em **OK** para finalizar a execução do procedimento.

- **On Error GoTo <linha> e Resume <linha>**

1. Complete o código do procedimento **Erro3**.

```
Sub Erro3()  
'Ativa a rotina de tratamento de erro que inicia na linha Erro:  
On Error GoTo Erro  
Dim num, parcela  
num = InputBox("Digite o número de parcelas")  
parcela = 500 / num  
MsgBox "O valor da parcela é " & Format(parcela, "##0.00")  
'Linha rotulada ErroFim  
ErroFim:  
'Interrompe o procedimento  
Exit Sub  
'Rotina de tratamento de Erro  
Erro:  
MsgBox "Número inválido"  
'Desvia a execução para a linha ErroFim  
Resume ErroFim  
End Sub
```

2. Execute este procedimento. Na mensagem **Digite o número de parcelas** clique em **OK**. Note que neste caso a mensagem **Número inválido** será exibida e o procedimento se encerra.

3. Execute novamente e na mensagem **Digite o número de parcelas** digite **2** e clique em **OK**. Note que neste caso a mensagem **O valor é** é exibida e o procedimento se encerra.

- **Testando o número do erro**

1. Complete o código do procedimento **Erro4**.

```
Sub Erro4()  
'Ativa a rotina de tratamento de erro que inicia na linha Erro:  
On Error GoTo Erro  
Dim num, parcela  
num = InputBox("Digite o número de parcelas")  
parcela = 500 / num  
MsgBox "O valor da parcela é " & Format(parcela, "##0.00")  
'Linha rotulada ErroFim  
ErroFim:
```

```

'Interrompe o procedimento
Exit Sub
'Rotina de tratamento de Erro
Erro:
'Testar o número do erro e enviar mensagem
If Err = 11 Then
    MsgBox "O número não pode ser zero"
ElseIf Err = 13 Then
    MsgBox "Digite apenas números"
End If
'Desvia a execução para a linha ErroFim
Resume ErroFim
End Sub

```

2. Execute este procedimento. Na mensagem **Digite o número de parcelas** teste as diferentes possibilidades.
3. Feche o editor, salve as alterações e saia do **Excel**.

RESUMO: PROPRIEDADES, EVENTOS, MÉTODOS, INSTRUÇÕES (FONTE: AJUDA DO APLICATIVO)

PROPRIEDADES: definem as características de um objeto.

- **Caption:** texto descritivo que aparece em um controle/objeto para identificá-lo ou descrevê-lo.
- **Value:** especifica o conteúdo do controle.
- **Offset(Linhas,Colunas):** representa o deslocamento em relação ao intervalo especificado. A opção Linhas define o deslocamento vertical. A opção Colunas define o deslocamento horizontal.
- **Text:** representa o texto exibido na caixa de combinação.
- **NumberFormat:** define o formato de número aplicado a uma célula

OBJETOS: representam um elemento específico no Excel.

- **Application:** representa o próprio **Excel**.
- **Sheets:** representa a coleção de todas as planilhas e gráficos da pasta especificada ou ativa. **Sheets ("Nome")** representa a planilha/gráfico especificado.
- **Range ("Intervalo"):** representa a célula ou intervalo especificado.
- **ActiveCell:** representa a célula ativa.

MÉTODOS: representam ações aplicadas a um objeto.

- **Show:** exibe um objeto **UserForm**.
- **OnTime:** programa um procedimento para ser executado em um momento especificado no futuro (seja em uma determinada hora do dia ou após uma quantidade específica de tempo decorrido). Use **Now + TimeValue(time)** para programar algo para ser executado quando uma quantidade de tempo específica, contando a partir de agora, tiver decorrido. Use **TimeValue(time)** para programar algo para ser executado em um momento específico.
- **SetFocus:** move o foco para o objeto/controle especificado.
- **Select:** seleciona um objeto.

FUNÇÕES

- **Now:** retorna uma variant (Date) que especifica a data e hora atuais de acordo com a data e hora do sistema.
- **TimeValue("hora"):** retorna uma variant (Date) que contém a hora.
- **MsgBox ("Prompt"):** exibe uma mensagem em uma caixa de diálogo.
- **FormatNumber (Expressão, Formato):** Retorna uma expressão formatada como um número.
- **IsEmpty():** retorna True quando a expressão não tiver sido inicializada ou estiver vazia (empty).
- **Cdbl:** converte uma expressão para número tipo double

INSTRUÇÕES

- **Unload:** remove um objeto da memória. Para se referir ao próprio objeto ativo, substitua o nome do objeto pela palavra chave **Me**.
- **IF...Else...Endif:** executa condicionalmente um grupo de instruções, dependendo do valor de uma expressão.
- **Do Until...Loop:** repete um bloco de instruções até que uma condição se torne verdadeira.
- **On Error:** ativa uma rotina de tratamento de erro e especifica o local da rotina dentro de um procedimento. Pode ser usada também para desativar uma rotina de tratamento de erro.
- **On Error Resume Next:** especifica que, quando ocorrer um erro em tempo de execução, o controle passará para a instrução imediatamente seguinte à instrução onde ocorreu o erro e a execução continua.
- **On Error GoTo <linha>:** ativa a rotina de tratamento de erro que inicia na linha especificada do argumento **linha** obrigatório. O argumento **linha** é qualquer rótulo de linha ou número de linha. Se ocorrer um erro em tempo de execução, o controle desvia para **linha**, tornando o tratamento de erro ativo. A **linha** especificada deve estar no mesmo procedimento em que se encontra a instrução **On Error**, caso contrário, ocorrerá um erro em tempo de compilação.

No momento em que um erro em tempo de execução é gerado um código de erro representado por um número inteiro. Este código fornecido pelo sistema pode ser testado através do objeto **Err**. Quando ocorre um erro, as propriedades do objeto **Err** são preenchidas com informações que identificam com exclusividade o erro. A propriedade padrão do objeto **Err** é a **Number**. Ela retorna o número de identificação do erro ocorrido. A propriedade **Description** do objeto **Err** retorna a cadeia de caracteres que descreve o erro ocorrido. Quando nenhum erro é gerado, a instrução **Err** irá retornar o valor n.

- **Resume:** continua a execução após a conclusão de uma rotina de tratamento de erro.
- **Resume Next:** continuar a execução na próxima linha de código.
- **Resume <linha>:** continuar a execução na linha especificada.

EVENTOS: representam ações que ocorrem durante o uso do aplicativo.

- **Workbook_Open():** ocorre quando a pasta de trabalho é aberta.
- **Activate():** ocorre quando um objeto torna-se a janela ativa. Um objeto pode se tornar ativo utilizando-se o método Show no código.
- **Click:** ocorre quando o usuário clica em um controle com o mouse.
- **QueryClose(Cancel as Integer, CloseMode as Integer):** ocorre antes que um **UserForm** feche. Este evento possui dois argumentos: **Cancel** e **CloseMode**. Definir o argumento **Cancel** com qualquer valor diferente de 0 interrompe o evento **QueryClose** em todos os formulários de usuário carregados e evita que o **UserForm** e o aplicativo fechem. O argumento **CloseMode** indica a causa do evento **QueryClose**.

Constante	Valor	Descrição
VbFormControlMenu	0	O usuário escolheu o comando Close a partir do menu Control no UserForm .
VbFormCode	1	A instrução Unload é invocada a partir do código.
VbAppWindows	2	A sessão do ambiente operacional Windows atual está terminando. (Somente para o Visual Basic 5.0.)
VbAppTaskManager	3	O Gerenciador de tarefas do Windows está fechando o aplicativo. (Somente para o Visual Basic 5.0.)

- **Change():** ocorre quando a propriedade **Value** é alterada.