

Curso: Automação de Escritórios e Secretariado

Disciplina: Inf - II Programação em VBA

VBA com Word / Excel / Access

Office 2007

(Versão 2.0)

Prof. Hamilton Martins Viana

esta apostila pode ser encontrada no site:

<http://sites.google.com/site/hamiltonmv>

Página Assunto

03	Histórico
03	Primeira parte - Word – Aula 1, Criação de uma macro
06	Inclusão de uma macro na Barra de Ferramentas de Acesso Rápido
07	Programação usando o VBA (Visual Basic for Applications)
08	O ambiente VBA - Exercícios com macros
09	Exercício de macro da Calculadora - uso da instrução Shell
10	Salvando o documento com macros
11	Aula 2 de programação com Word - Recordação
11	Uso de formulários
12	Aula 3 de programação com Word
13	Uso de caixa de textos
16	Aula 4 de programação com Word
19	Aula 5 Programação com Excel
21	Criando macros sem uso do gravador de macros
22	Uso de formulários em Excel
26	Uso dos tratadores de evento em VBA com Excel
27	Aula 6 de programação com Excel
29	Aula 7 de programação com Excel
29	Criação e uso de funções
33	Aula 8 programação em VBA com Access
34	Uso de formulários com Access
39	Alternar entre a janela do VBA e o do banco de dados (Alt + F11)
40	Aula 9
41	Aula 10
41	Inclusão de botões no formulário
44	Uso do DoCmd
46	Aula 11
49	Aula 12
51	Finalizando...
53	Automatização de tarefas.
58	Projeto final da disciplina
59	Bibliografia

Histórico:

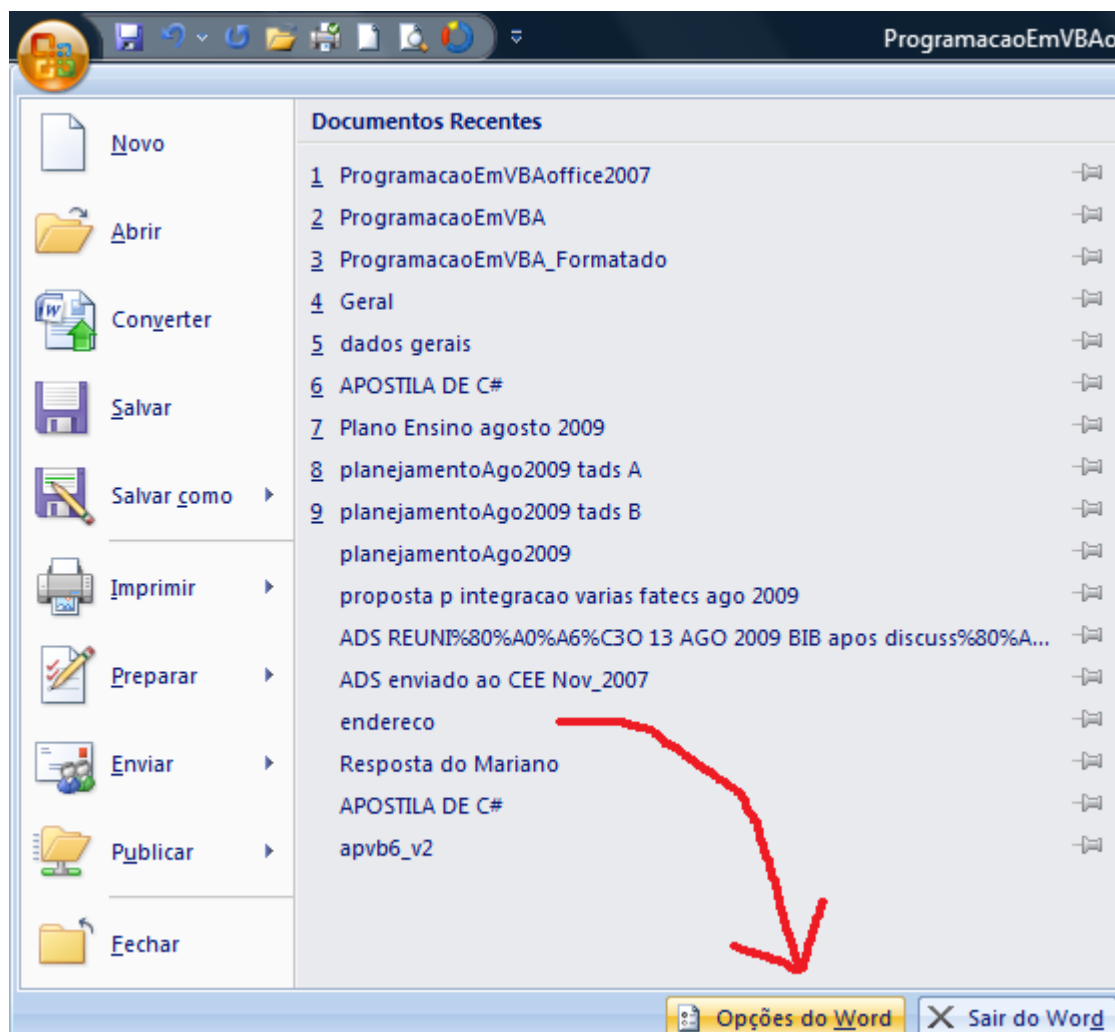
Segundo a Microsoft, foram vendidas mais de 120 milhões de licenças do “Office-2007”. O pacote “Office” é composto de quatro aplicativos (Power Point, Word, Excel e Access). Apesar de serem os programas mais utilizados em todo o mundo, poucos conhecem e usam seus recursos mais avançados que podem ser utilizados através da programação com VBA (Visual Basic for Applications).

Primeira Parte – Word – Aula 1

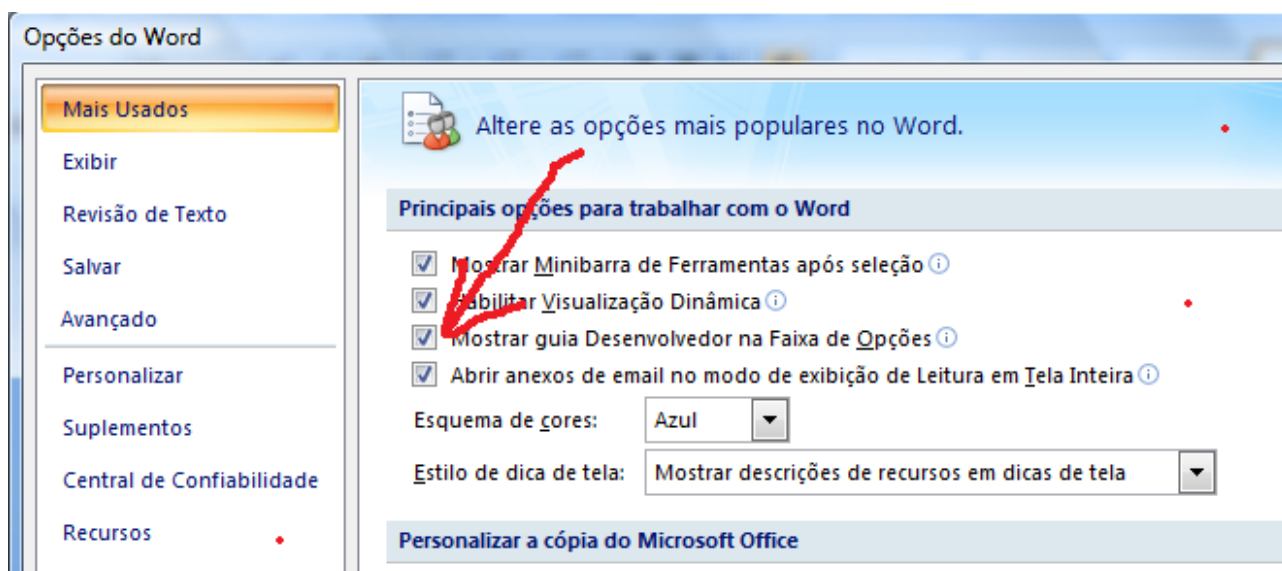
No Word, a programação é feita através de Macros, da seguinte maneira:

Criação de uma macro

Para criar uma nova macro deverá estar disponível a guia “Desenvolvedor”, que fica visível clicando-se no botão “Office” e em seguida no botão “Opções do Word”. Veja as janelas abaixo:



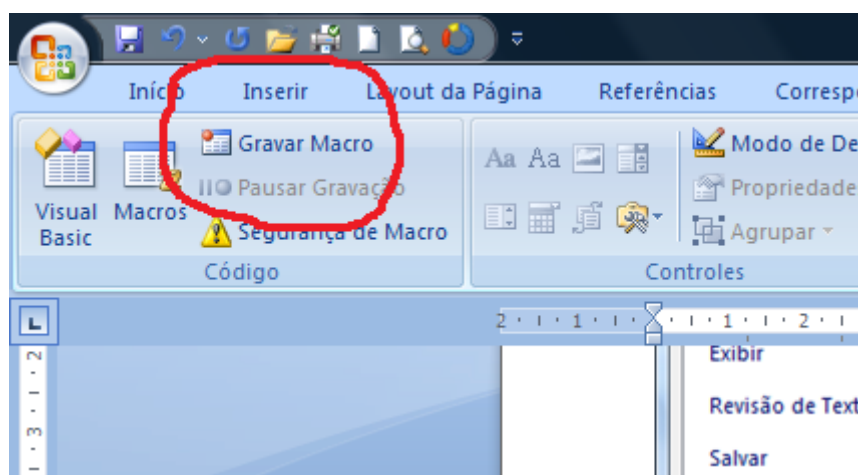
Em seguida dê um click em “Mais Usados” e ative a caixa “Mostrar guia Desenvolvedor na Faixa de Opções”.
Veja a janela a seguir:



Click em Ok p/ fechar as janelas e note que agora está disponível a guia “Desenvolvedor”, que utilizaremos em nossos trabalhos.

Uma macro contém um conjunto de instruções que serão processadas passo a passo, quando a macro for executada.

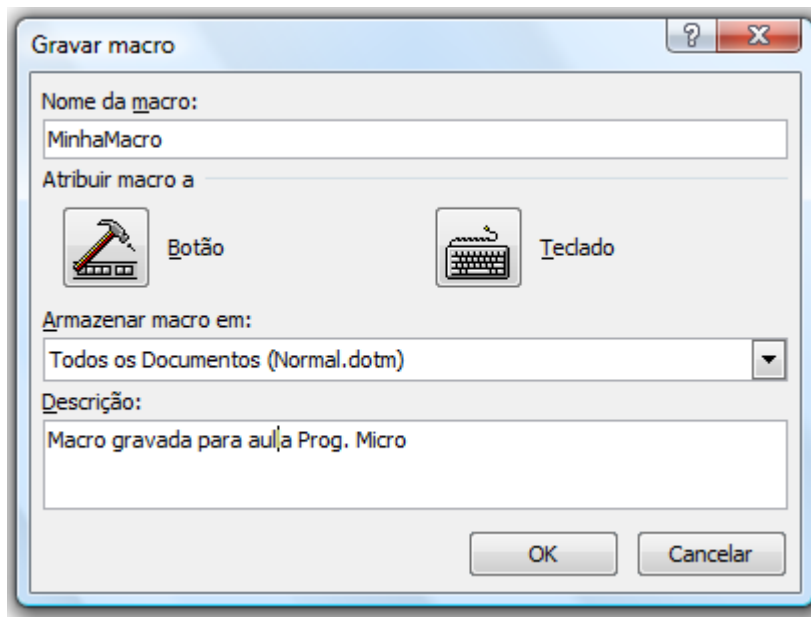
Para criar uma nova macro, dê um click na guia Desenvolvedor e um click no item “Gravar Macro”:



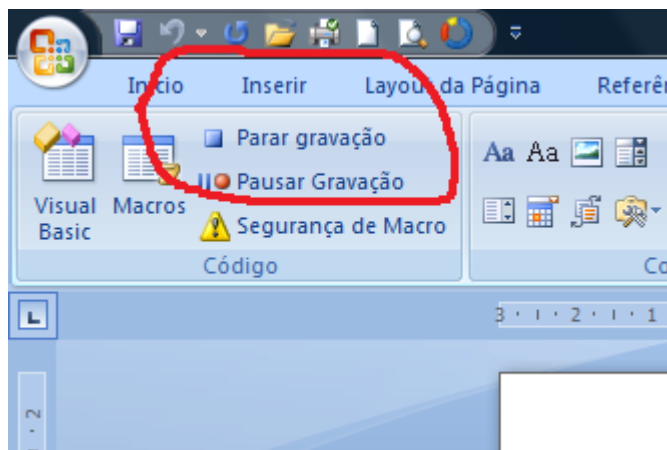
Na janela que abrir, escolha o nome da nova macro (pode ser por exemplo, “MinhaMacro”) e escolha o seu documento, na caixa “Armazenar macro em:”. Se você escolher “Todos os documentos”, a macro ficará gravada no Word que está instalado no micro.

Faremos isso e ao final desta lição, nós a apagaremos. A janela para gravação da macro, ficará como o exemplo abaixo:

Obs. Experimente digitar o nome com espaço (Minha Macro) e veja o que acontece!



Dê um click no botão OK. Note que o apontador do mouse ficou um pouco diferente. A partir desse momento todas as ações que você executar estarão sendo gravadas, até que você pare a gravação, dando um click no item retangular da janela “Parar gravação”



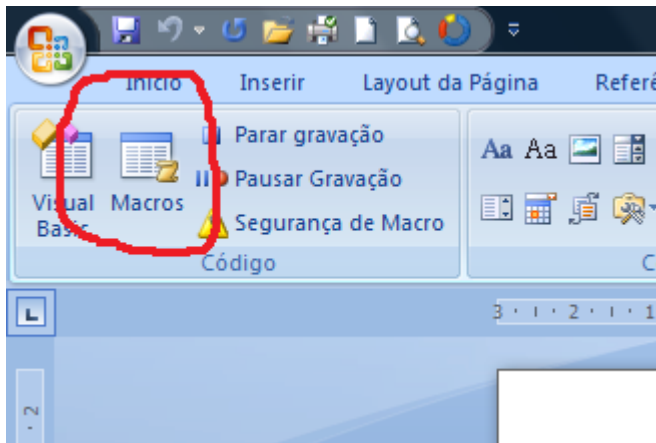
Antes de parar a gravação vamos realizar algumas ações que ficarão gravadas em nossa macro. Digite por exemplo seu nome, como por exemplo: “Ricardão, o Bãããão!!!”, formate-o com tamanho 16, tachado e cor vermelha. Pare a gravação da macro.

Obs: Como está sendo utilizado o gravador de macros, a seleção com o mouse não é possível. Para selecionar o texto, use a tecla Shift + as setas de direção.

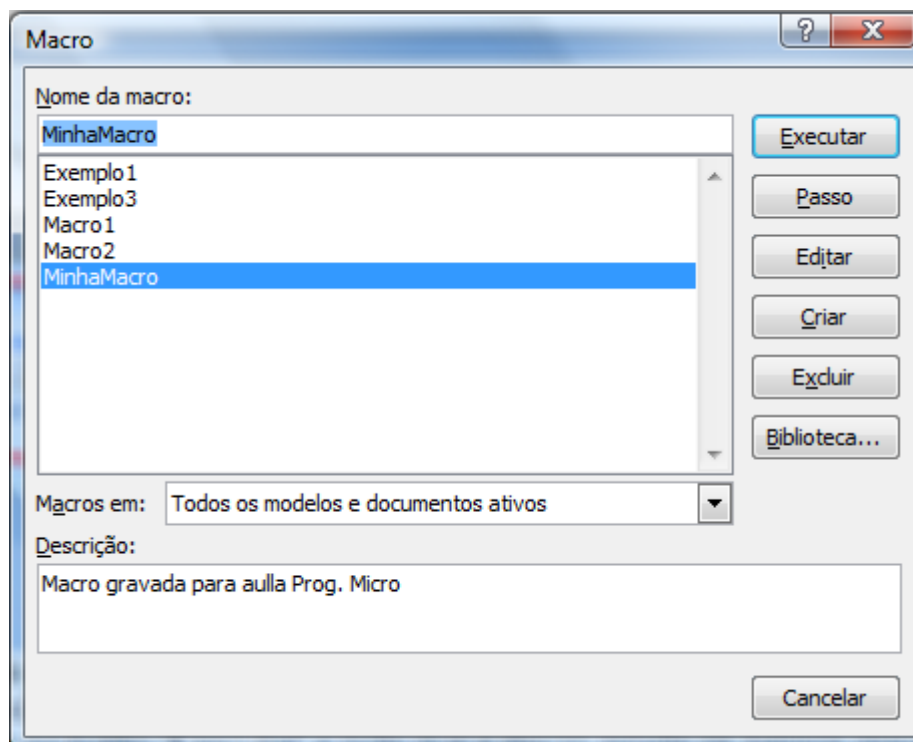
Podemos parar agora a gravação da macro. A partir de agora, toda vez que a macro for executada, repetirá as operações que você realizou.

Vamos testar a macro MinhaMacro:

- Apague o texto impresso pela macro em seu documento.
- Dê um click para acionar o item Macros, na guia Desenvolvedor e na janela que surgir, selecione a macro que quer executar.



- Selecione a macro desejada (neste caso é a macro MinhaMacro, o nome que você escolheu).
- Dê um click no botão “Executar” e note que a macro repete os passos gravados.



Inclusão de uma macro na Barra de Ferramentas de Acesso Rápido.

Vamos automatizar um pouco mais esse processo.

Podemos atribuir nossa nova macro à “Barra de Ferramentas de Acesso Rápido” para podermos executá-la com mais comodidade.

Para isso, dê um click no botão Office e em Opções do Word.

Na janela que abrir, na caixa da esquerda, click em “Personalizar”

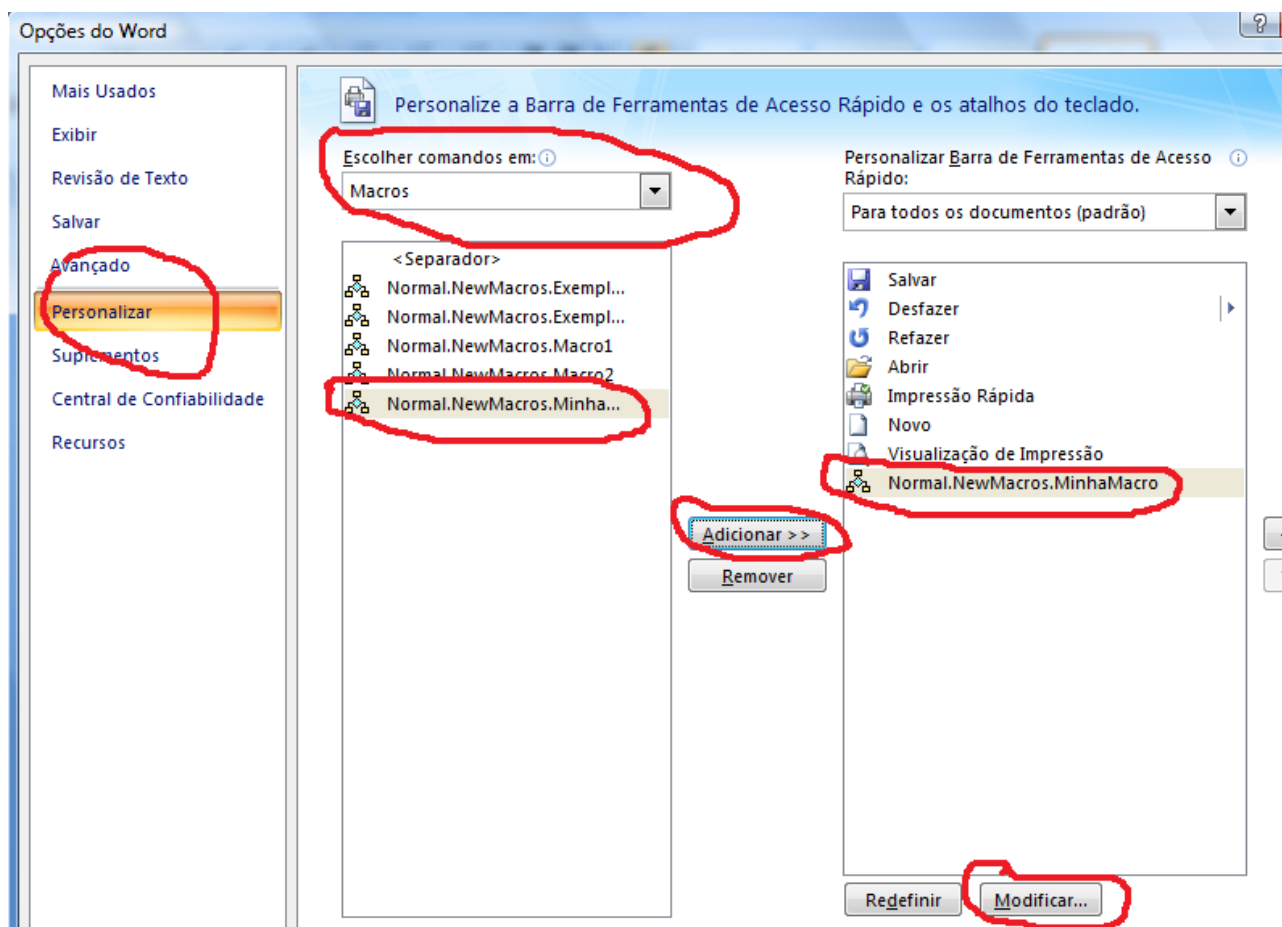
No item “Escolher comandos em:” selecione “Macros” e em seguida escolha a macro desejada (neste caso é “MinhaMacro”). Click em “Adicionar”.

Na caixa da direita, selecione a macro desejada e click em “Modificar”

Escolha o ícone mais adequado e mude o “Nome para exibição”. Pronto! Dê um click em Ok e veja o botão p/ acionar a macro na Barra de Ferramentas de Acesso Rápido.

Quando terminar, você pode utilizar a macro, com apenas um click do mouse.

Abaixo há um exemplo da janela “Opções do Word” com os itens a serem configurados:



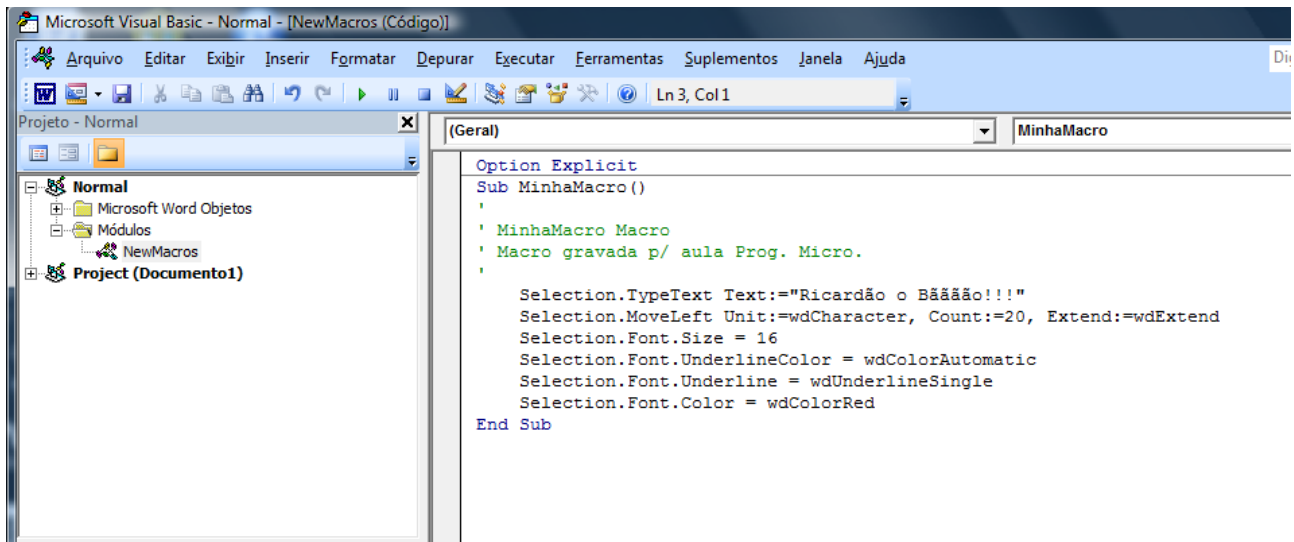
Obs: Dê um click com o botão direito do mouse na Barra de Acesso Rápido e veja as opções.

Programação usando o VBA (Visual Basic for Applications):

Vamos agora verificar a parte mais poderosa do uso de macros através do VBA.

Para isso, selecione a guia Desenvolvedor e dê um click no item “Macros”.
Selecione a macro que interessa e dê um click sobre o botão “Editar”. Nesse momento o Word abre o ambiente do VBA, que é muito parecido com o ambiente do Visual Basic 6.0.

O ambiente VBA.



Através desse ambiente, podemos fazer a programação que quisermos.

Note que as frases em cor verde são comentários (iniciam por apóstrofe) sendo utilizados apenas para documentação.

Obs.: O gravador de macros inclui, na forma de instruções, todas as ações que executamos quando criamos a macro.

Podemos modificar diretamente as instruções ou incluir novas instruções mudando o comportamento da macro.

Exercícios com a macro que acabamos de fazer:

- Através de digitação nas instruções da macro, altere o texto a ser impresso no documento;
- Altere para verde, a cor do texto que está em vermelho.
- Coloque instruções para acrescentar mais algum texto na linha seguinte (por exemplo "boa tarde pessoal...");
- Volte ao ambiente do documento (pressione Alt + F11), execute a macro para verificar as alterações efetuadas.

**Obs.: Estando no ambiente do VBA você pode obter ajuda sobre quaisquer instruções ou objetos que desejar. Para isso, selecione a instrução ou objeto e pressione a tecla F1.
Por exemplo, selecione a palavra Color e pressione F1.**

Obs.: Para alternar entre a janela do VB e a janela do Word, pressione Alt + F11.

Obs.: Note que as macros são para o VB, apenas PROCEDIMENTOS, dentro da pasta “Módulos”

Exercício : Suponha que desejamos acionar a calculadora do Windows.

Podemos criar macros usando o gravador de macros, conforme fizemos com a MinhaMacro, ou podemos digitar a macro diretamente na pasta Módulos. Faremos de acordo com essa segunda opção:

Crie uma nova macro chamada Calculadora, coloque um botão na “Barra de Ferramentas de Acesso Rápido”, com o ícone de uma calculadora, sendo que quando esse botão for acionado, é executado o programa “calc.exe”.

Vamos editar manualmente a macro Calculadora:

Após o "End Sub" da macro MinhaMacro, digite:

sub Calculadora() e pressione a tecla Enter.

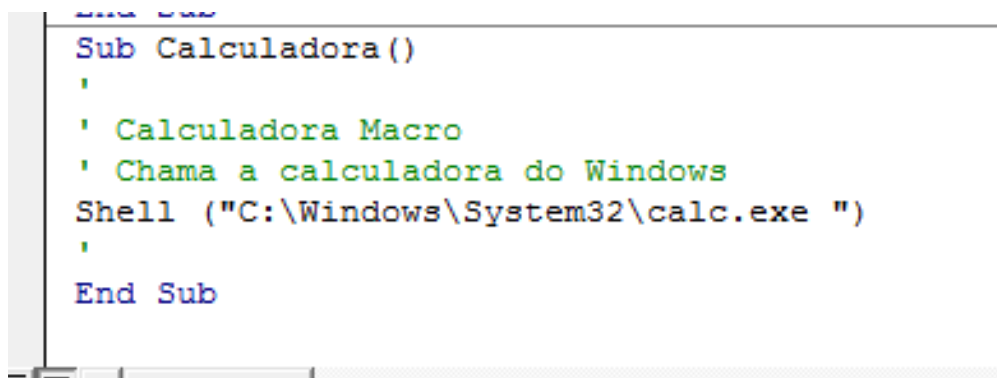
Note que o editor do VBA já incluiu p/ você o "End Sub" da sua macro Calculadora.

Digitar na Sub Calculadora(): Shell ("C:\Windows\System32\calc.exe ")

Obs.: Uso da instrução Shell:

- a) A instrução Shell(“ caminho / programa ”) executa programas externos
- b) A instrução acima executa o programa calc.exe que está na pasta c:\windows.

Nossa Macro ficará assim:



```
Sub Calculadora()  
'  
' Calculadora Macro  
' Chama a calculadora do Windows  
Shell ("C:\Windows\System32\calc.exe ")  
'  
End Sub
```

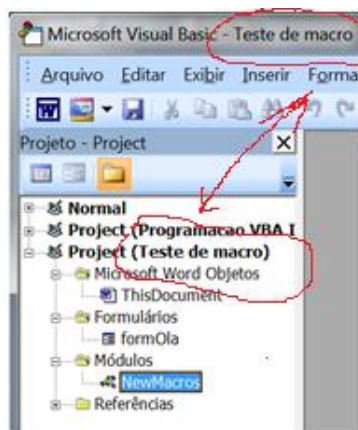
Obs.: Caso a Calculadora não esteja na pasta indicada acima, localize-a e corrija o programa.

- Voltar ao ambiente do Word (alt + F11).
- Testar a macro.

Salvando o documento com macros

Observações importantes:

- Todos os exercícios serão apresentados ao professor, para avaliação. Portanto, se você terminou, aguarde o professor.
- Ao salvar o documento, escolha salvar com o tipo “Documento Habilitado para Macro do Word(*.docm)”. Se isso não for feito, todas as macros que você fez serão perdidas.
- Há duas opções para salvar as macros: No Word que está instalado no computador ou no documento que você pode copiar p/ o pendrive, levá-lo embora e abrir posteriormente, para usar as macros em outro equipamento. Se optar por esse segundo método, a janela Project Explorer deverá ficar de acordo com a figura abaixo:



Salve o arquivo com nome ExercWordAula1, de acordo com o que foi explicado acima.

Feche o arquivo que você acabou de salvar.

Abra o arquivo e verifique se as macros ainda estão lá e se funcionam como anteriormente.

Feche o arquivo novamente e desligue o computador.

Aula 2 Programação com Word

Exercícios:

Abrir o arquivo ExercWordAula1 da aula anterior e salve-o com nome ExercWordAula2, tomando as providências para que as macros não se percam. Verifique se tudo funciona como anteriormente.

Exercício para recordação / treinamento / familiarização com o ambiente do VBA.

Fazer manualmente (sem uso do gravador de macros), uma macro de nome “TesteInputOutput” que utilize:

- variáveis - é a maneira que manipulamos valores em programação.
- inputbox – é uma função do VBA que nos permite a entrada de dados em um programa.
- msgbox – é uma função do VBA que nos permite exibir mensagens para o usuário.

Essa macro deverá declarar uma variável do tipo string que receberá uma frase digitada pelo usuário.

Em seguida, será exibida uma caixa de mensagem com a frase digitada. Quando for dado o click no botão Ok da caixa de mensagem, a frase deverá ser enviada como texto para o documento do Word onde se executou a macro.

A codificação da macro será mais ou menos como a seguir:

```
Sub TesteInputOutput()  
'  
' TesteInputOutput Macro  
' Abre InputBox, exibe MsgBox e põe string no documento  
'  
    Dim frase As String  
  
    frase = InputBox("Digite uma frase para o documento do Word:")  
  
    MsgBox "A frase digitada foi: " + frase, vbExclamation, "A T E N Ç Ã O"  
  
    Selection.TypeText Text:="A frase digitada na inputbox foi : " + frase  
End Sub
```

Complementação:

- a) Na msgbox a frase digitada deverá aparecer na segunda linha da caixa de mensagem - use chr\$(13) para concatenação de strings e mudar a linha.
- b) O string a exibir no documento do Word deve ser no tipo de fonte “Times New Roman” (Selection.Font.Name=“Times New Roman”), O tamanho deve ser 16, negrito, itálico, azul.
- c) Após a exibição do texto com as configurações acima, tudo o que o usuário digitar pelo teclado deve ter tamanho de fonte 12, preto, não itálico não negrito.

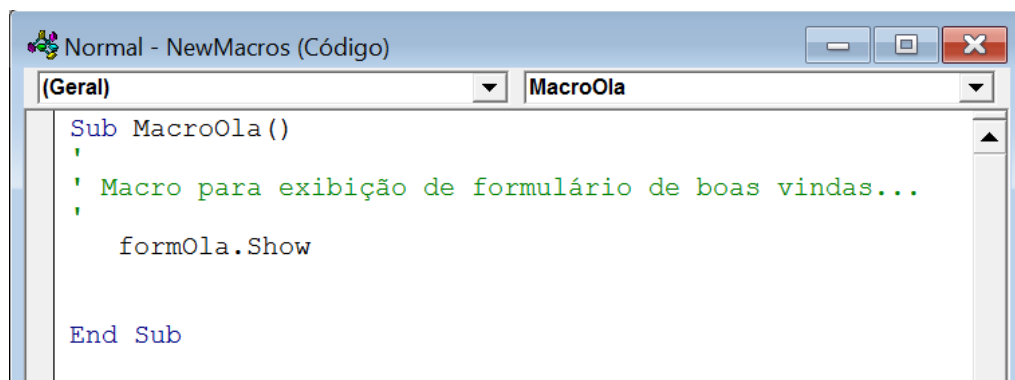
Faça todos os teste, salve o documento e aguarde o professor para verificar seu trabalho.

Aula 3 Programação com Word

Programação VBA no Word com o uso de formulários:

Abrir o arquivo ExercWordAula2 da aula anterior e salve-o com nome ExercWordAula3, tomando as providências para que as macros não se percam. Verifique se tudo funciona como anteriormente.

Fazer uma macro diretamente no Módulo (sem usar o gravador) que exiba um formulário com layout e programação mais ou menos como a seguir.



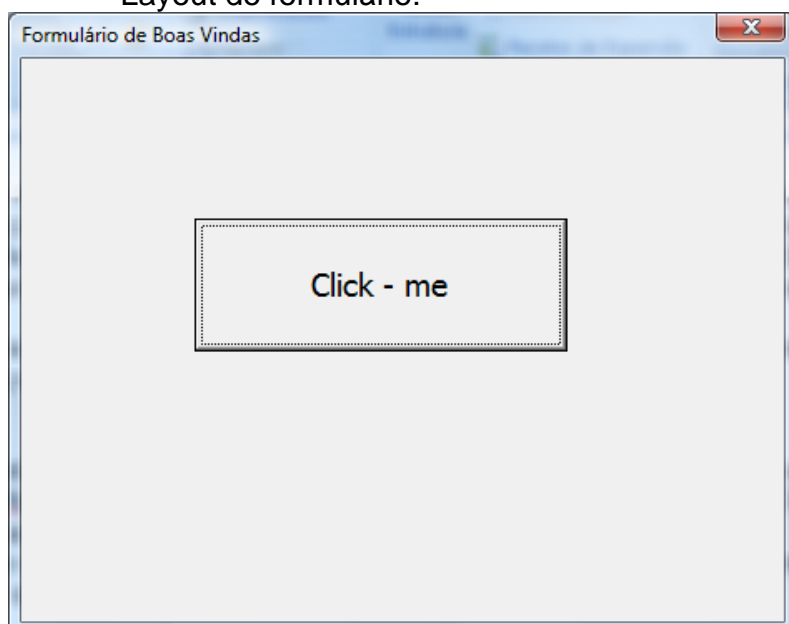
Coloque na guia de acesso rápido, um ícone para executar essa macro. Execute-a e analise o resultado.

Crie o formulário de nome "formOla" com layout igual ao abaixo.

O nome do botão (propriedade name), é cmdClickMe.

O nome dos objetos (componentes) dos formulários é importantíssimo pois é através de seu nome que programamos as ações que devem ocorrer com esses objetos.

Layout do formulário:



Quando o usuário der um click no botão do formulário acima, uma mensagem de boas vindas é exibida.

Programação associada ao evento Click no botão do formulário:

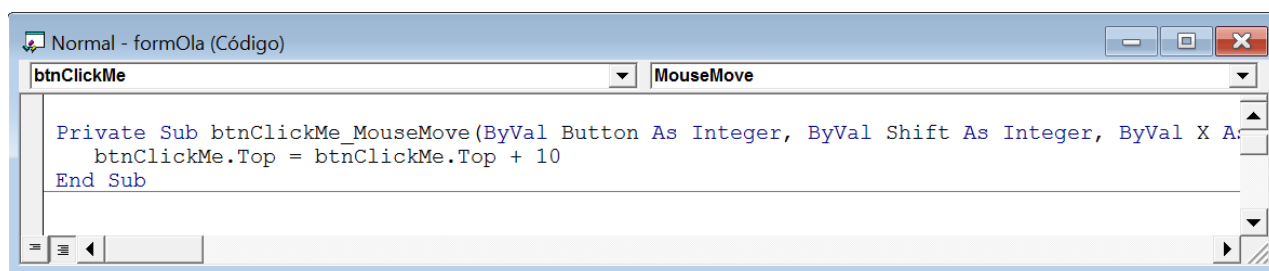
Option Explicit

```
Private Sub btnClickme_Click()  
    MsgBox "B E M   V I N D O S   A O   V B A", vbExclamation, "B O A S   V I N D A S"  
End  
End Sub
```

O procedimento acima exibe uma caixa de mensagem dando boas vindas. Ao ser dado o click no botão OK da caixa de mensagem, esta é encerrada e o processamento passa à instrução seguinte, o "End", que encerra o processamento do VBA.

Vamos fazer incluir mais uma funcionalidade em nosso botão cmdClickMe!

Inclua no tratador de evento MouseMove, a instrução abaixo:



Volte ao ambiente do documento Word, execute a macro e tente dar um click no botão...

Por que você não consegue alcançar o botão?

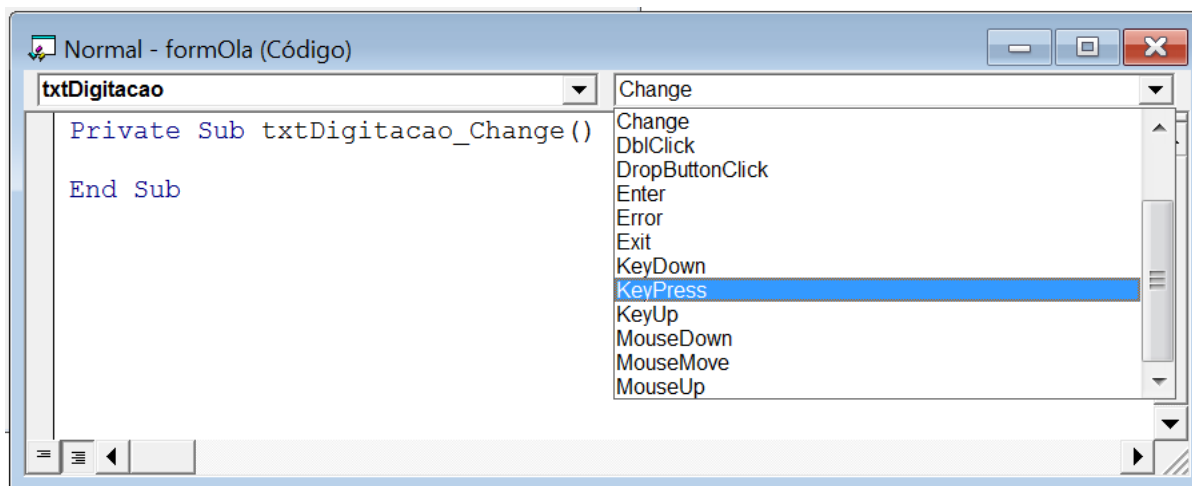
Uso de Caixa de Textos

Fazer diretamente no Módulo (sem o uso do gravador de macros), um programa que exibe uma caixa de textos de acordo com o layout abaixo. Pode usar o mesmo formulário de boas vindas do exercício anterior:

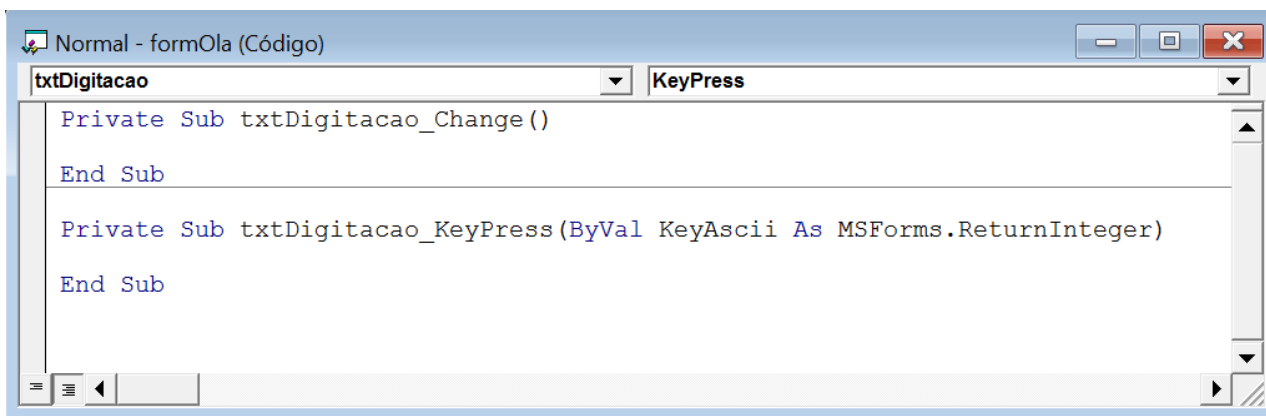
Volte ao ambiente do Word, execute a macro e digite seu nome na caixa de texto. Até aí, nada de novo!

Volte ao ambiente do VBA, dê um duplo click na caixa de texto.

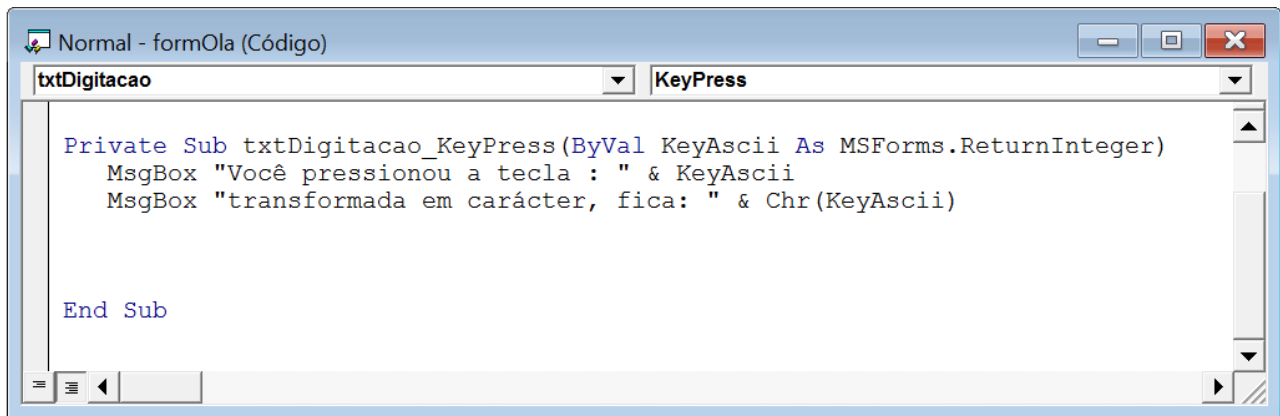
Note que é aberta a janela de código dos tratadores de eventos da caixa de textos "txtDigitacao", para tratar o evento "Change". Não queremos esse! Vamos escolher o tratador de evento "KeyPress".



Nossa janela de código ficará assim:



Vamos digitar as seguintes instruções no nosso tratador do evento KeyPress:



```
Normal - formOla (Código)
txtDigitacao  KeyPress

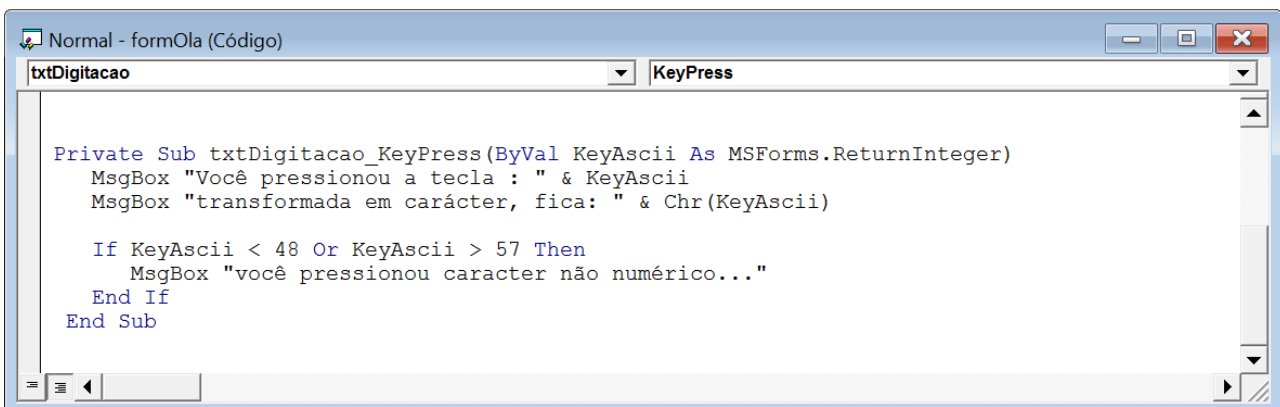
Private Sub txtDigitacao_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    MsgBox "Você pressionou a tecla : " & KeyAscii
    MsgBox "transformada em carácter, fica: " & Chr(KeyAscii)
End Sub
```

Execute o programa, pressione algumas teclas e veja o resultado.

Verifique a tabela ASCII, na em nossa pasta de trabalho. Qual o valor ASCII das tecla numéricas?

Vamos programar nossa caixa de texto para que esta somente aceite a digitação de números!

Acompanhe a programação a seguir...



```
Normal - formOla (Código)
txtDigitacao  KeyPress

Private Sub txtDigitacao_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    MsgBox "Você pressionou a tecla : " & KeyAscii
    MsgBox "transformada em carácter, fica: " & Chr(KeyAscii)

    If KeyAscii < 48 Or KeyAscii > 57 Then
        MsgBox "você pressionou caracter não numérico..."
    End If
End Sub
```

O que você faria para permitir que somente números sejam digitados em nossa caixa de textos????

Salve o arquivo pois será usado em nossa próxima aula.

Aula 4 Programação com Word

Abrir o arquivo ExercWordAula3 da aula anterior e salve-o com nome ExercWordAula4, tomando as providências para que as macros não se percam. Verifique se tudo funciona como anteriormente.

Trabalho para a Aula 4:

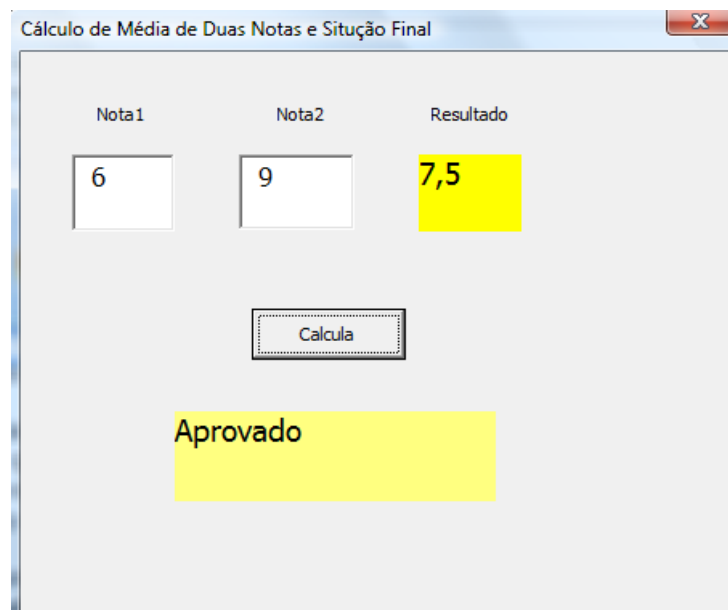
Fazer diretamente no Módulo (sem usar o gravador de macros), uma macro de nome "CalculaNota" que exibe um **formulário de acordo com o layout a seguir**. O usuário digita as duas notas de um aluno e o programa calcula a média aritmética. O programa exibe no documento do Word a nota e a informação se o aluno está "Aprovado" ou "Reprovado".

obs.: Se quiser, pode modificar o formulário já existente no documento.

A macro que exibe o formulário, será mais ou menos assim:

```
Sub CalculaNota()  
    formNotas.Show  
    MsgBox "A situação do aluno será exibida no documento do Word..."  
End Sub
```

Layout do formulário de nome formNotas:



Exemplo da programação inicial associada ao formulário acima, ao ocorrer o click do mouse no botão de nome cmdCalcula:

```
Private Sub cmdCalcula_Click()  
  
    lblResultado.Caption=txtNota1.Text + txtNota2.Text  
  
End Sub
```

Faça o teste e verifique se está tudo certo...

Talvez melhore se mudarmos a codificação para:

```
Private Sub cmdCalcula_Click()
```

```
    lblResultado.Caption=val(txtNota1.Text) + val(txtNota2.Text) 'converte texto em número
```

```
End Sub
```

Obs.: Como precisamos da média aritmética, devemos dividir por 2 o resultado. Aí deve ficar assim:

```
Private Sub cmdCalcula_Click()
```

```
    lblResultado.Caption=val(txtNota1.Text) + val(txtNota2.Text) / 2 'converte texto em número
```

```
End Sub
```

Parece que não deu muito certo! Como você resolve isso?

Exercícios:

- a) As caixas de texto txtNota1 e txtNota2 devem aceitar apenas valores numéricos. De acordo com nossa anterior, a programação é a seguinte:

```
Private Sub txtNota1_KeyPress(Byval KeyAscii As MSForms.ReturnInteger)
    If KeyAscii < 48 Or KeyAscii > 57 Then      ' aceita apenas números
        KeyAscii = 0 ' Caracteres inválidos são convertidos para carácter nulo
    End If
End Sub
```

- b) Se uma tecla inválida for digitada, o programa deverá exibir mensagem de alerta.
- c) As caixas de texto txtNota1 e txtNota2 devem aceitar no máximo 4 caracteres (propriedade maxlength da caixa de textos).
- d) Inclua no formulário do VBA uma caixa de textos para ser digitado o Nome do Aluno. Todos os caracteres devem ficar em maiúsculo. Use as funções do VBA: Asc, Ucase, Chr. Exemplo:

```
Private Sub txtNomeAluno_KeyPress(Byval KeyAscii As MSForms.ReturnInteger)
    KeyAscii = Asc ( UCase ( Chr ( KeyAscii ) ) )
End Sub
```

Obs.:

A função Chr converte um valor Ascii para Carácter;
A função Ucase transforma um carácter minúsculo para maiusculo;
A função Asc converte um carácter em seu valor numérico Ascii.

Ao ser pressionado o botão “Calcula”, exibir **no documento do Word**, o nome do aluno que está na caixa de textos e exibir a informação se o aluno está ou não aprovado.

Sugestão final proposta (até agora!):

```
Private Sub btnCalcula_Click()  
  
    Dim result As Single  
    result = (Val(txtNota1.Text) + Val(txtNota2.Text)) / 2  
    lblResultado.Caption = result  
    If result >= 5.0 Then  
        lblSituacao.Caption = "Aprovado"  
    Else  
        lblSituacao.Caption = "Reprovado"  
    End If  
  
    Selection.TypeText Text:=lblSituacao.Caption ' Exibe resultado no documento do Word  
  
End Sub
```

- Se o aluno estiver aprovado, o texto “**APROVADO**”, será na cor azul.
- Se reprovado, o texto “**REPROVADO**” deverá ser na cor vermelha.

Após efetuar seus testes, vamos eliminar as nossas macros do Word. Para isso, dê um click com o botão direito do mouse no ícone da Barra de Ferramentas e os remova.

Em seguida, dê um click no item Macros, selecione cada macro e dê um click no botão Excluir.

Aula 5 Programação em VBA com Excel

Para iniciarmos, abra o Excel, para começarmos a criação de nossa macro, que irá acrescentar um cabeçalho em qualquer planilha na qual for chamada.

No Excel o processo de criação de macro é muito parecido com o Word. Para testar, vamos seguir os mesmos passos usados no Word para criar uma nova macro:

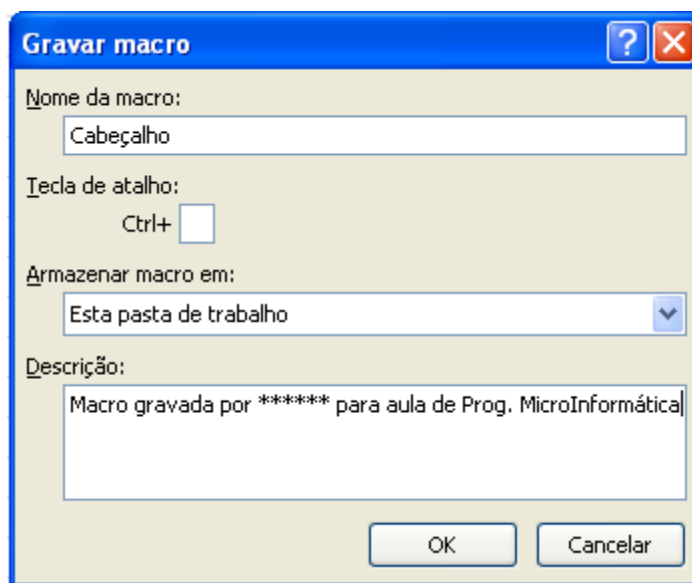
Para isso, faça ficar disponível a guia “Desenvolvedor”, selecione-a e dê um click no item “Gravar macro”.

Na janela que abrir, configure o seguinte:

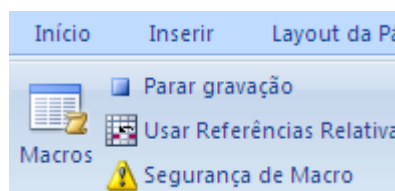
- Mude o nome da macro para “Cabeçalho”;
- No item “Armazenar macro em”, escolha “Esta pasta de trabalho”.

Obs.: Se você escolher “Pasta de trabalho pessoal de macros”, a macro ficará gravada no Excel que está instalado no micro. Não faremos isso. A macro que criarmos será disponibilizada apenas na planilha que estamos desenvolvendo. Para isso, na caixa “Armazenar macro em”, escolha “Esta pasta de trabalho”.

- Em “Descrição”, substitua o ***** por seu nome.
- Click em Ok.



A partir deste momento, tudo que você fizer estará sendo gravado na macro, até que você dê um click no botão .Parar gravação.



Quando parar a gravação, a macro será gravada na forma de comandos VBA, com o nome que você escolheu.

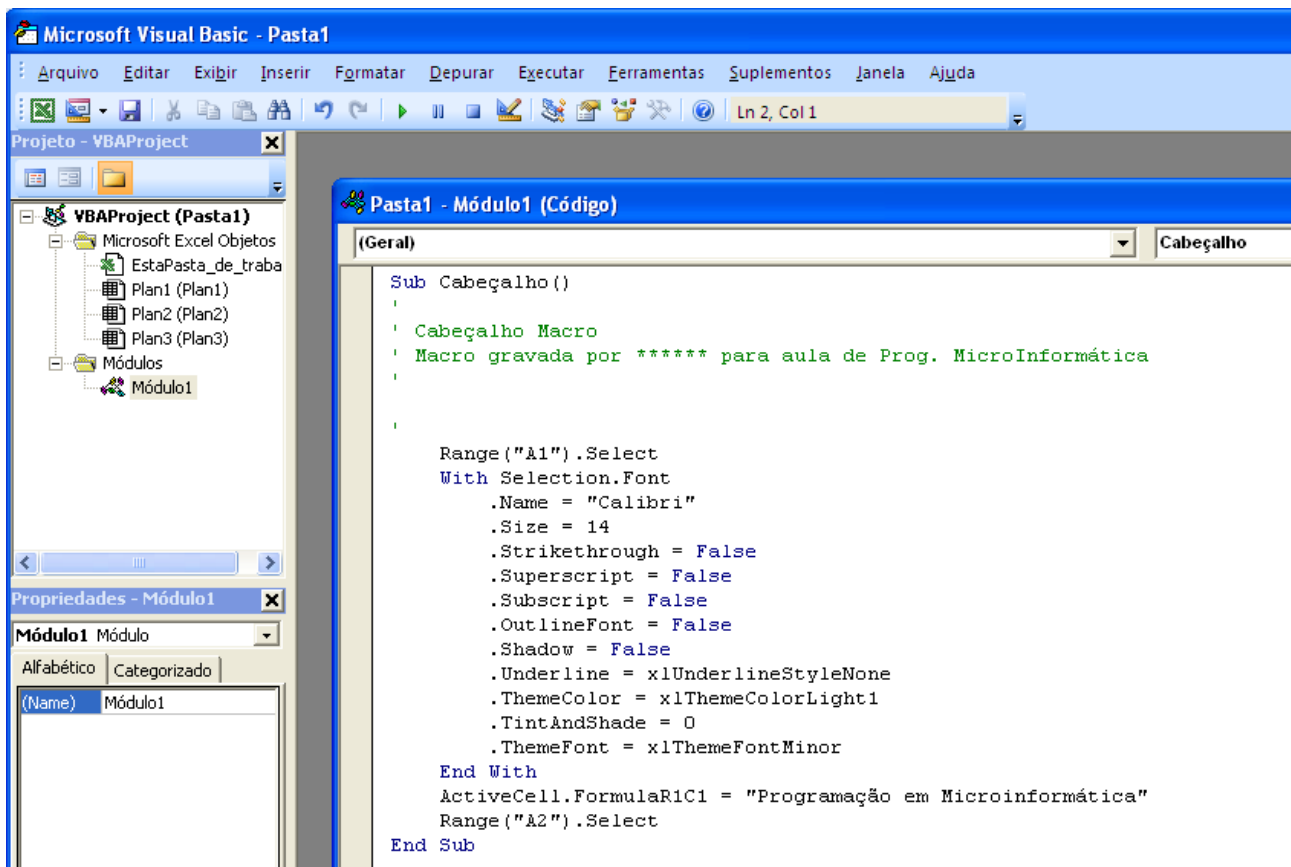
A macro que iremos criar fará o seguinte:

- Insere uma linha a partir da primeira linha'
- Posiciona o foco na célula A1 e a formata com tamanho 14;
- Digita o texto “Programação em microinformática”, na célula A1;

Então no Excel, siga os seguintes passos:

- . Selecione a primeira linha e insira uma nova linha:
- . Posicione o cursor na célula A1 e formate-a com tamanho 14;
- . Na célula A1 digite o texto “Programação em microinformática” e pressione a tecla “Enter”.
- . Encerre a gravação da macro, dando um click no botão “Parar Gravação”.

Pronto: Nossa macro “Cabeçalho” está gravada. Vamos vê-la. Dê um click no botão “Macros”, selecione a macro “Cabeçalho” e dê um click no botão “Editar”
Aparecerá uma janela parecida com a abaixo:



A figura acima mostra uma parte do ambiente de desenvolvimento do VBA. Explore a macro Cabeçalho...

Obs. Comentários: Em VB iniciam com apóstrofo, ficam na cor verde e servem apenas para a documentação do programa.

A macro “Cabeçalho” está gravada em nossa planilha do Excel. Vamos executá-la. Para voltar ao Excel ou vice-versa, use a combinação de teclas Alt + F11.

Apague o conteúdo da célula A1 (Editar / Limpar / Tudo) e execute a macro Cabeçalho que você criou, para ver seu efeito.

Para executá-la, abra selecione a guia “Desenvolvedor”, dê um click no botão “Macros”, escolha a macro “Cabeçalho” e dê um click em “Executar”.

Após executá-la, volte ao ambiente do VBA, com a combinação de teclas Alt + F11 para estudá-la um pouco melhor.

Obs. “O gravador de macros do Excel gera a parte mais “pesada” da codificação”, gerando inclusive algumas instruções desnecessárias, que deixam a execução da macro mais lenta.

Podemos analisar e modificar as instruções da macro para melhor adequá-la a nossas necessidades.

Por exemplo, podemos eliminar as instruções:

```
.Strikethrough = False  
.Superscript = False  
.Subscript = False  
.OutlineFont = False  
.Shadow = False
```

Podemos também substituir a instrução “ActiveCell.FormulaR1C1 = “Prog. Micro”,
Por ActiveCell.FormulaR1C1 = inputbox (“Digite o Cabeçalho desejado...”)

Faça essas modificações, execute a macro e veja o efeito.

Obs.: Você pode executar a macro, pressionando a tecla F5. Se você estiver no ambiente do VBA e pressionar F5, a macro será executada o foco volta ao ambiente do VBA.

- Para ver o efeito da execução da macro, use Alt + F11 (Mostra a planilha).
- Limpe novamente os textos da planilha e execute a macro novamente.

Criação de macros no Excel sem uso do gravador de macros

Criando macros sem o uso de gravador:

Abra o editor do VBA (Alt + F11), e digite após o “End Sub” do procedimento “Sub Cabeçalho”:

```
Sub Teste_Cabecalho_Com_Data()
```

Note que após você pressionar Enter, o programa inclui automaticamente o “End sub”.
Digite em seguida, os comandos abaixo, deixando a procedure da forma:

```
Sub Teste_Cabecalho_Com_Data()
```

```
    Range("A1").Select  
    ActiveCell.FormulaR1C1 = InputBox("Digite o Cabecalho : ", "A T E N Ç Ã O") + Str(Date)  
End Sub
```

Obs. : A expressão Str (Date) é concatenada à string que será digitada na InputBox. Ela é composta por duas funções do VBA:

Str – converte número em texto.

Date – retorna a data do computador.

Pressione F5 para executá-la e veja o efeito. Obs. Quando se pressiona F5, é executado o procedimento onde está o foco (cursor do editor).

Volte para o ambiente do Excel (Alt + F11) e verifique que o procedimento **Teste_Cabecalho_Com_Data** aparece junto com a macro anterior (Cabeçalho).

A qualquer momento, o ambiente do VBA dispõe de um Help, de onde se podem obter informações valiosas. Para acionar o Help, basta selecionar o objeto ou instrução que se necessita ajuda e pressionar a tecla F1. Tente com **"Inputbox"** da macro acima.

Uso de formulários em Excel:

Utilização de formulários na programação com VBA: Utilizam-se formulários de maneira similar à programação em VB. No formulário, são incluídos componentes tais como botões, caixas de texto, labels etc. que respondes a eventos provocados pelo usuário.

Para testar isso, feche o Excel, e através do "Windows Explorer" **copie para sua pasta ou seu pendrive**, o arquivo "Livros".

Abra o arquivo "Livros" de sua pasta ou pendrive, contendo a planilha "Estoque", igual à abaixo:

	A	B	C	D	E	F
1	Código	Nome	Autor	Editora	Preço	Quantidade
2	605	C++	Valmir	Mack	R\$ 67,00	10
3	805	Office	Fabiani	Erica	R\$ 92,00	21
4	1005	VB net	Viana	Makron	R\$ 77,00	6
5	104	Delphi	Cantú	Sams	R\$ 130,00	3
6	204	C Builder	Kent	Makron	R\$ 78,00	5
7	304	Access	Pádua	Érica	R\$ 45,00	2
8	505	V. Basic	Nunes	QUE	R\$ 88,00	7
9						

Obs.: No Office 2007, o nível de segurança aumentou. Ao salvar uma planilha que contenha macros, no item de menu "Salvar Como" escolha uma pasta "Pasta com Macros Excel". Se isso não for feito, a planilha será salva sem as macros e você perderá todo o seu trabalho.

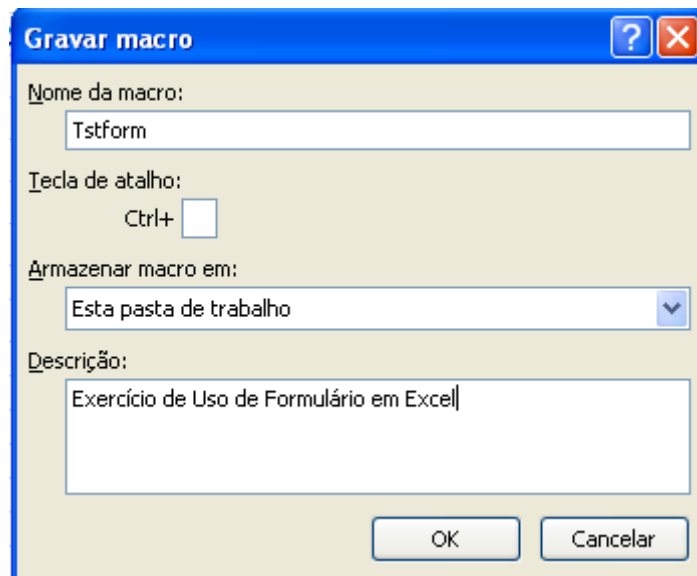
Siga o procedimento adiante para que as macros de sua planilha "Livros" fiquem sempre habilitadas:

Clique no botão Office / Clique em Opções do Excel / Central de Confiabilidade / Configurações da Central de Confiabilidade / Configurações de Macro / Habilitar todas as Macros / Ok / Ok

Salve o arquivo, feche o arquivo Livros. Note que o ícone de seu arquivo Livros ficou diferente. Abra o arquivo novamente. A partir de agora, seu arquivo estará sempre com as macros habilitadas.

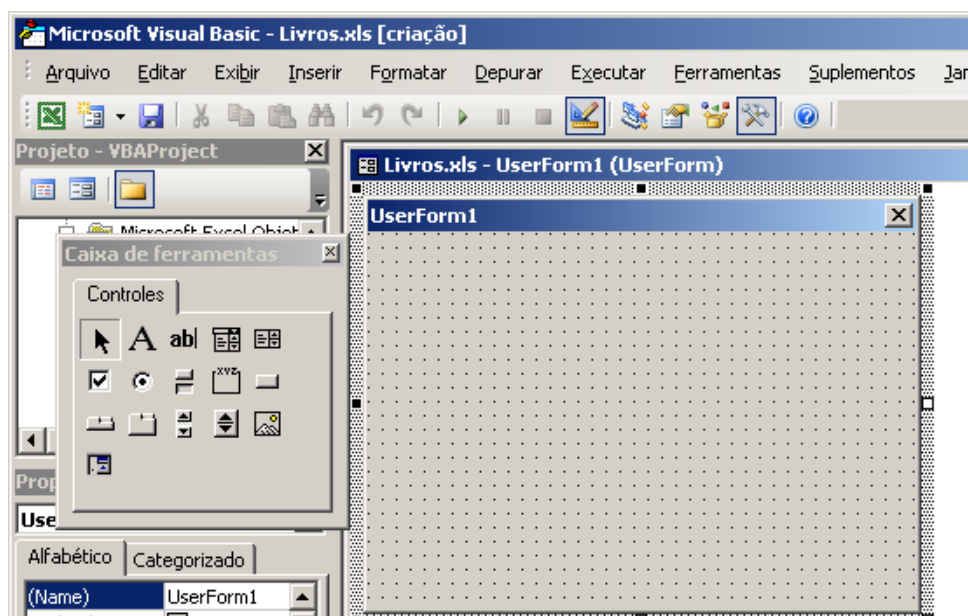
Exercício:

Crie uma macro de nome TstForm, sem nenhuma programação (click no botão “Gravar Macro”).



Pressione o botão OK e em seguida pare a gravação. Pronto! Gravamos uma macro de nome TstForm, sem nenhuma programação. Agora, abra o ambiente do VBA, com a combinação Alt + F11. Na janela “Project Explorer”, abra a pasta Módulos/Módulo1, que é onde estão as macros.

Inclua um formulário associado à sua planilha. vá ao ambiente do VBA e dê click nos itens Inserir / UserForm. Você verá a seguinte janela:



Através dessa janela você irá elaborar e programar a interface com a qual o usuário irá manipular dados na planilha.

Obs.: Estrutura de comandos e programação em VBA – aos objetos são associadas propriedades, que são as características do objeto. Por exemplo, o tamanho, cor etc. são propriedades de um objeto. Aos objetos também são associados métodos (funções) – exemplo Range(“A2:A6”).Select – deixa selecionadas as células A2 até A6.

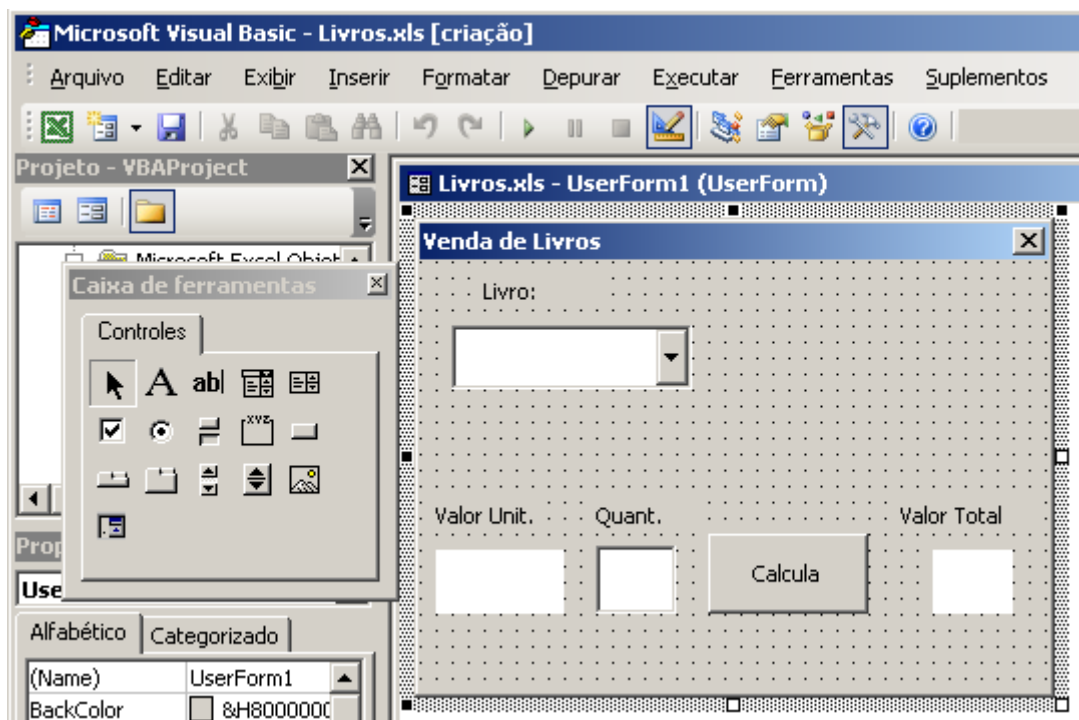
Variáveis – É a maneira pela qual manipulamos valores em programação. Exemplo de criação de variáveis em VBA :

No exemplo abaixo, ao ocorrer o click do mouse no formulário, é criada a variável x que recebe o string "Bem vindos ao VBA...", que é atribuído à barra de título do formulário.

```
Private Sub UserForm_Click()  
    Dim x As String  
    x = "Bem vindos ao VBA..."  
    FrmVendaLivros.Caption = x  
End Sub
```

Selecione o formulário e na janela “Propriedades” digite, na **propriedade Caption**, o texto “Venda de Livros”.

Inclua no formulário os componentes abaixo e configure o formulário de acordo com o seguinte layout:



Obs.: vamos mudar todos os nomes dos componentes, de acordo com seu tipo e objetivo:

Exemplos:

Formulários – inicia com Frm – exemplo FrmVendaLivros

Caixa de Combinação - Inicia com Cxc – exemplo CxcLivre

Label – Inicia com Lbl – exemplos LblValorUnit, LblValortotal

Caixa de Texto – Inicia com Txt – exemplo TxtQuantidade

Botão de comando – inicia com Cmd – exemplo CmdCalcula

Para isso, selecione cada componente e altere a propriedade Name na janela de propriedades.

Para executar o formulário, vamos criar nossa procedure, digitando diretamente na Macro TstForm:

```
Sub TstForm()
```

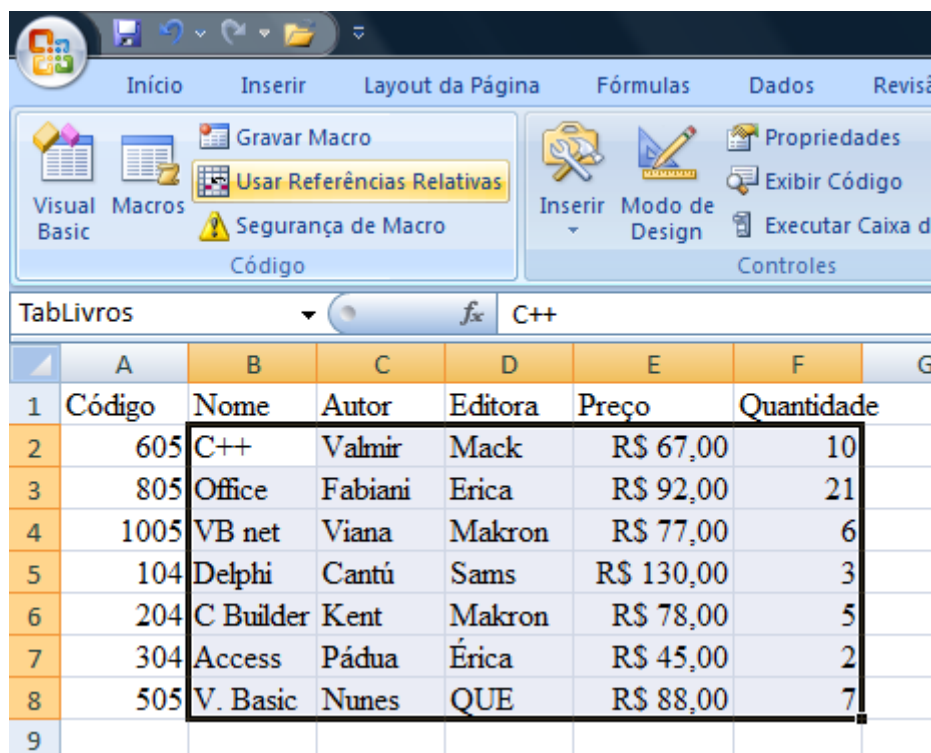
```
    FrmVendaLivros.Show ' Esta instrução exibe e executa o formulário FrmLivros.
```

```
End Sub
```

Pressione a tecla F5 para ver o programa funcionando.

Note que os componentes estão prontos e funcionando, porém não respondem aos eventos provocados pois estes (os eventos) não estão sendo tratados. Faremos o tratamento a seguir.

Feche o formulário e volte à janela da planilha (Alt + F11), selecione a área igual à abaixo, dê o nome a essa seleção de “TabLivros” e pressione a tecla Enter.



The screenshot shows the Microsoft Excel ribbon with the 'Formulas' tab selected. The ribbon includes buttons for 'Gravar Macro', 'Usar Referências Relativas', 'Segurança de Macro', 'Código', 'Inserir', 'Modo de Design', 'Propriedades', 'Exibir Código', and 'Executar Caixa de'. Below the ribbon, a table named 'TabLivros' is visible. The table has 7 columns: 'Código', 'Nome', 'Autor', 'Editora', 'Preço', and 'Quantidade'. The data rows are numbered 2 through 8. The table is highlighted with a black border.

	A	B	C	D	E	F	G
1	Código	Nome	Autor	Editora	Preço	Quantidade	
2	605	C++	Valmir	Mack	R\$ 67,00	10	
3	805	Office	Fabiani	Erica	R\$ 92,00	21	
4	1005	VB net	Viana	Makron	R\$ 77,00	6	
5	104	Delphi	Cantú	Sams	R\$ 130,00	3	
6	204	C Builder	Kent	Makron	R\$ 78,00	5	
7	304	Access	Pádua	Érica	R\$ 45,00	2	
8	505	V. Basic	Nunes	QUE	R\$ 88,00	7	
9							

Volte ao ambiente do VBA, selecione o componente CxcLivros e coloque em sua propriedade RowSource, o texto “TabLivros”, que é o nome que você colocou na região selecionada da planilha.

Execute novamente o programa (tecla F5) e dê um click no componente CxcLivro.

Note que o componente responde ao evento, trazendo a tabela TabLivros.

Vamos continuar nossa programação. Para voltar à fase de projeto, interrompa a execução do programa.

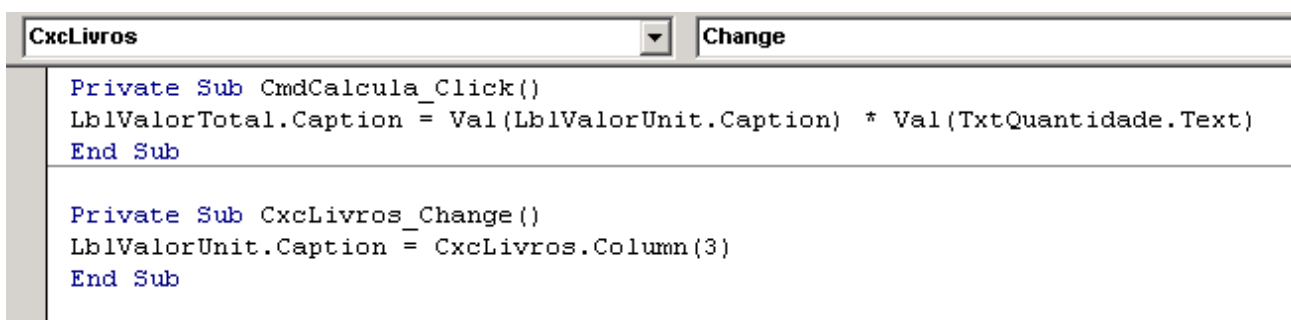
Duas funcionalidades de nosso programa serão:

- a) Quando ocorrer o click no nome do livro, deverá surgir o valor unitário no label de nome LblValorUnit.
- b) Se digitarmos a quantidade de livros e pressionarmos o botão “Calcula”, o programa deverá exibir o valor Total da compra, no label LblValorTotal.

Uso de tratadores de evento em VBA com Excel

Programação dos “Tratadores de Eventos” do VBA:

A janela de programação abre-se ao ocorrer o click duplo no componente que desejamos programar. Essa janela tem a seguinte aparência e a programação será a seguinte:



Faça o teste e verifique o funcionamento do programa.

Selecione um livro na caixa de combinação, digite a quantidade desejada e pressione o botão “Calcula”. Ao fazer isso, o tratador do evento Click do botão é executado e o valor total dos livros é calculado.

Faça todos os testes e prepare-se para apresentar o programa ao professor.

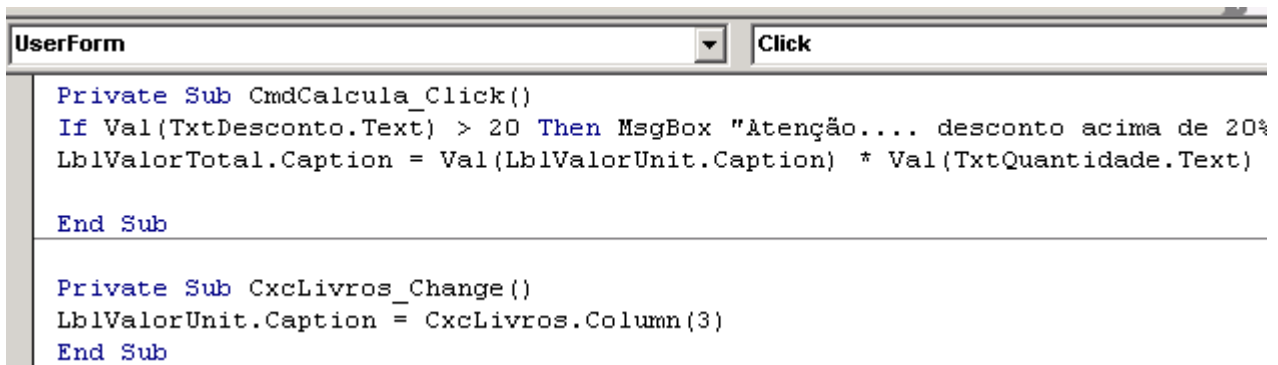
Aula 6 Programação em VBA com Excel

Abra a planilha Livros da aula anterior e teste-a p/ verificar se as macros funcionam.

Exercícios:

- 1) Fazer o valor unitário aparecer formatado, com centavos e separador de milhares. Isso pode ser feito com ajuda da instrução "format". Veja o exemplo abaixo:

```
Private Sub cxcLivro_Change()  
    lblValorUnit.Caption = Format(cxcLivro.Column(3), "R$##,###.00")  
End Sub
```
- 2) Note que o valor unitário fica muito bonito formatado, só que o programa não funciona mais. Verifique o motivo e conserte-o.
- 3) O campo "Quantidade" deve aceitar apenas números. Lembra do programa de cálculo da média que fizemos quando estudamos o VBA p/ Word? É bem parecido.
- 4) Formatar em formato de moeda (R\$) com centavos, o campo "Valor Total".
- 5) Inclua uma caixa de texto para que o operador possa oferecer um desconto no preço da compra. Caso o desconto exceda 20% o programa deve alertá-lo sobre esse fato, mas deve realizar o cálculo assim mesmo. Use a procedure do VBA msgbox. Use o Help para ver seu funcionamento.

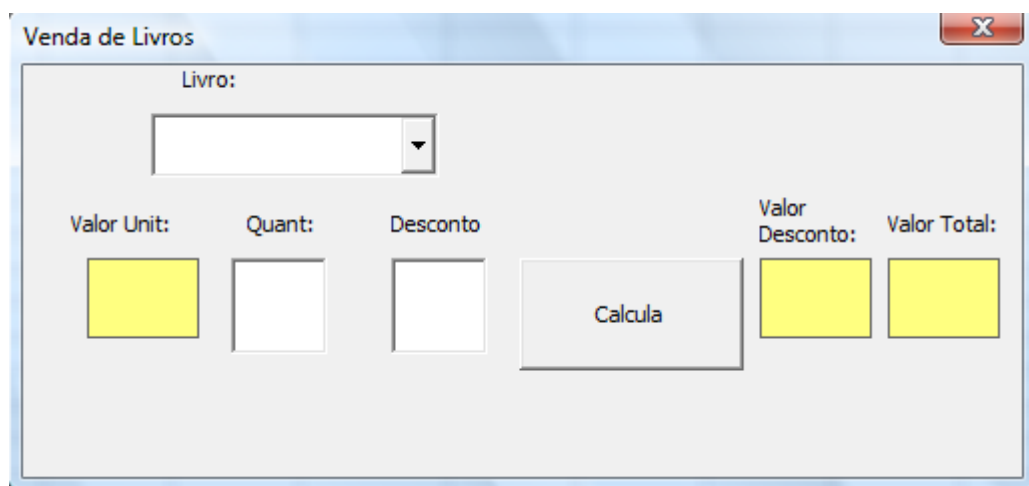


- 6) Inclua um label p/ exibir separadamente o valor do desconto em moeda.
- 7) O campo em que o operador vai informar o desconto, deve aceitar somente números e o ponto decimal. Note que o desconto deve ser um valor real. Se você for usar variáveis reais, a declaração é, por exemplo: **dim desconto as single**.
- 8) Formate adequadamente todos os valores em formato de moeda para exibição nos campos do formulário.

Use a instrução Format. Exemplo, supondo-se que valorTotal é variável real, a instrução de formatação do campo no formulário, será mais ou menos como abaixo:

LblValorTotal.Caption=Format (valorTotal, "R\$#,###.00")

Formato sugerido do formulário “Venda de Livros” até agora:



Para finalizar, selecione o botão e atribua para “True” sua propriedade “Default”. Se a propriedade default de um botão estiver True, você pode executar o programa associado ao evento Click desse botão, pressionando a tecla “Enter”.

Ainda no botão, coloque a letra “C” na propriedade “Accelerator” e note que a letra C do “Calcula”, ficou sublinhado. Agora você pode executar o programa do botão com a combinação de teclas Alt + C.

Salve a planilha, Teste o programa e prepare-se para apresentá-lo.

Aula 7 Programação em VBA com Excel

Vamos continuar desenvolvendo nossa aplicação com VBA.

Abra o arquivo Livros.xls que utilizamos na aula anterior e teste a macro que exibe o formulário "frmVendaLivros".

Teste as funcionalidades do formulário:

- A caixa de texto da quantidade, só aceita a digitação de números;
- A caixa de texto do desconto só aceita a digitação de números e do ponto decimal.
- Quando ocorre o click no botão calcula, o cálculo do preço a pagar é realizado e aparece formatado em moeda.

Exercícios:

- Ative o ambiente do VBA (Alt + F11)
- Leia o texto abaixo e faça o se pede:

Criação e uso de funções:

Podemos criar funções em VBA e utilizá-las na planilha Excel.

Por exemplo, suponha que precisamos da informação da situação de nosso estoque, da seguinte maneira:

- se o estoque estiver igual ou abaixo de 3 unidades, isso significa que **o estoque estará baixo**.
- se estiver entre 4 e 14 unidades, **estará bom (normal)**.
- com 15 unidades ou mais, **estará muito alto**.

Podemos usar as funções nativas do Excel e criar uma expressão para resolver isso. Uma solução melhor, seria criar a seguinte função em VBA, que deve ser digitada no mesmo Módulo que contém a procedure TstForm:

Obs.: As funções devem ocorrer na mesma pasta das macros, junto com as macros.

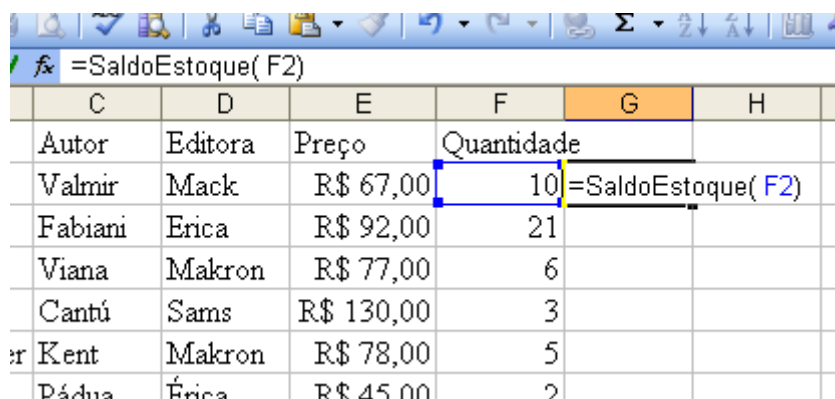
A função SaldoEstoque:

Function SaldoEstoque(estoque As Integer) As String

```
If estoque <= 3 Then SaldoEstoque = "Estoque Baixo"  
If estoque > 3 And estoque <= 15 Then SaldoEstoque = "Estoque Normal"  
If estoque > 15 Then SaldoEstoque = "Estoque Em Excesso"
```

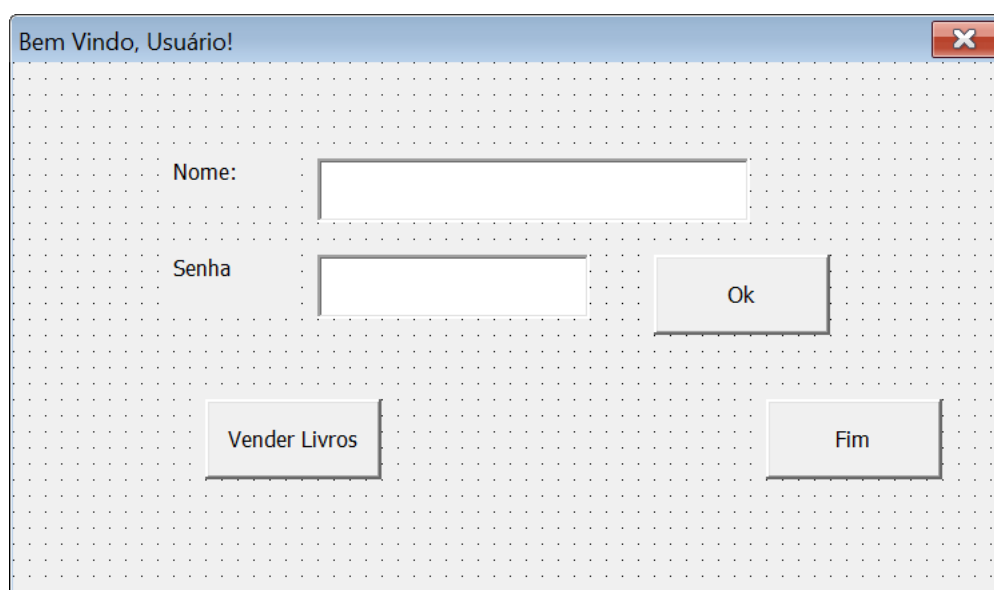
End Function

Agora basta chamar a função SaldoEstoque, na planilha e utilizá-la:



C	D	E	F	G	H
Autor	Editora	Preço	Quantidade		
Valmir	Mack	R\$ 67,00	10	=SaldoEstoque(F2)	
Fabiani	Erica	R\$ 92,00	21		
Viana	Makron	R\$ 77,00	6		
Cantú	Sams	R\$ 130,00	3		
Kent	Makron	R\$ 78,00	5		
Dádua	Erica	R\$ 45,00	2		

Continuando nosso projeto, inclua um novo formulário (Inserir UserForm) e configure-o como abaixo:



Bem Vindo, Usuário!

Nome:

Senha:

Ok

Vender Livros

Fim

Os nome dos componentes serão:

Name do formulário: FrmLivraria,

Name das caixas de texto: txtNome, txtSenha

Name dos botões: cmdOk, cmdVendas, CmdFim.

O botão cmdVendas deverá iniciar com a propriedade “Visible” false. Ou seja, não estará visível. Ficará disponível, ou seja, visível, somente quando o usuário acertar a senha.

Insira na propriedade Picture do formulário FrmLivraria, uma das imagens da pasta ...Server2000 / Prg_Micro.

Obs.: Podemos adequar o tamanho da figura ao formulário, através da propriedade PictureSizeMode.

Especificações e exercícios associados a este formulário:

- 1) Ao iniciar a execução da macro, o formulário “frmLivraria” é exibido, solicitando ao usuário que digite seu nome e senha.

A cada tecla que o usuário pressionar na caixa de texto txtSenha, iremos concatenando o carácter digitado em uma variável, para posterior verificação do que foi digitado. A programação será assim:

```
Private Sub txtSenha_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    senha = senha + Chr$(KeyAscii)
End Sub
```

Para usarmos a variável de nome "senha", esta deverá ser declarada:
dim senha as string

- 2) Ao ser pressionado o botão OK, o programa verifica se o usuário acertou ou não a senha, obedecendo as especificações abaixo:
 - a) Se a senha for correta, o programa exibe a mensagem: “Bem Vindo Senhor **nome**!”. Caso contrário, o programa exibe: “Caia Fora!...”.
 - b) Os caracteres digitados na senha aparecem como “*”. Para isso, configurar a propriedade PasswordChar.
 - c) A senha correta é “1234”.
 - d) Configure o botão “cmdFim” com a propriedade Caption Fim e com a propriedade “Accelerator” = “F”, que ao receber o click do mouse encerra o programa. A instrução é “End”.
 - e) A programação do botão cmdOk é mais ou menos assim:

```
Private Sub cmdOk_Click()

    If senha = "1234" Then
        MsgBox "Bem vindo!..."
        cmdVendas.Visible = True ' faz o botão cmdSenha ficar
visível
    Else
        MsgBox "Cai fora malandro...!"
    End If

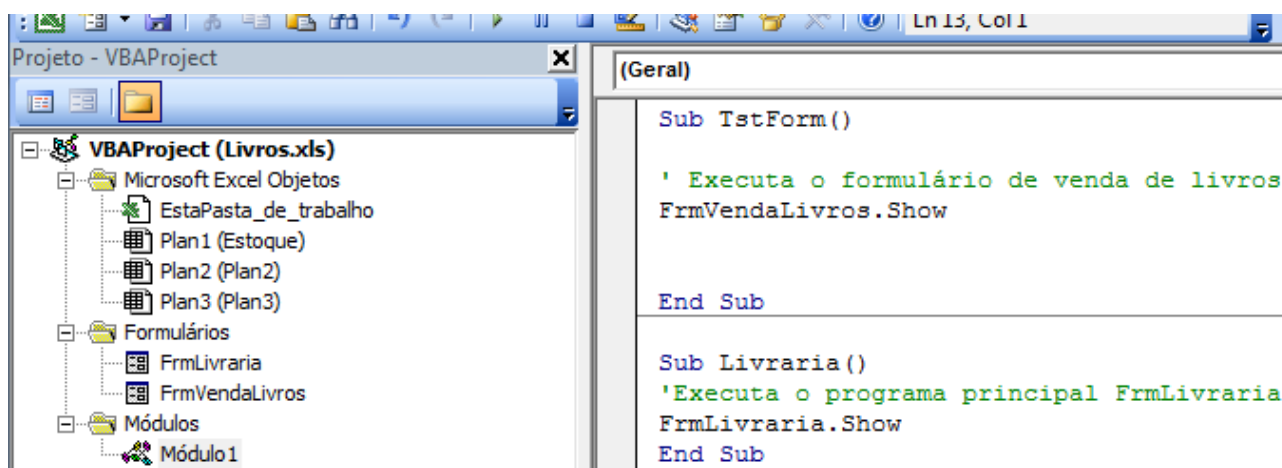
End Sub
```

- 3) Quando o usuário acertar a senha, fazer visível o botão cmdVendas.
- 4) Se o usuário o click no botão cmdVendas, este exibe o formulário frmVendaLivros. Vide o código abaixo:

```
Private Sub CmdVendas_Click()  
    FrmVendaLivros.Show 1  
End Sub
```

Obs.: O argumento “1” na instrução acima, especifica que o formulário FrmVendaLivros será exibido na forma modal, ou seja, com janela restrita. Veja o help.

5) Crie no Módulo1, a procedure “Livraria”, que ao ser executada, exhibe o formulário FrmLivraria. Veja a figura:



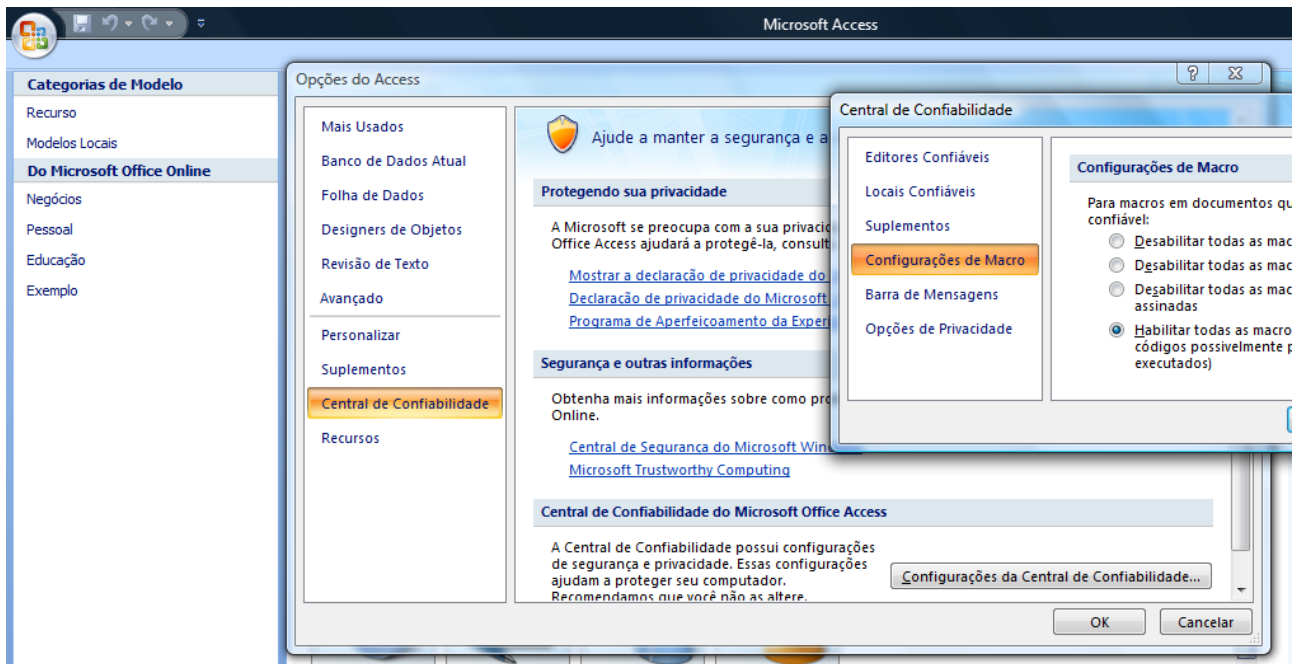
6) Inclua na barra de acesso rápido do Excel um ícone para acionar a macro **Livraria**.

Salve a planilha, Teste o programa e prepare-se para apresentá-lo.

Aula 8 Programação - em VBA com ACCESS

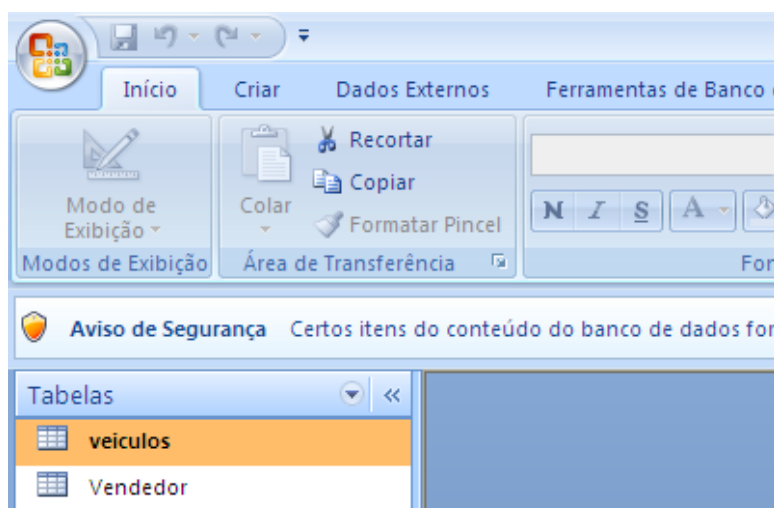
Para trabalharmos com a programação do Access, iremos inicialmente reduzir o nível de segurança para evitarmos problemas com a execução de macros. Para isso, localize e abra o Access. No botão Office, pressione o botão “Opções do Access” e selecione as janelas:

Central de Confiabilidade/Configurações da Central de Confiabilidade/Configurações de Macro. Em seguida, habilite todas as macros e **feche o Access**. Confira a figura abaixo:



Copie o arquivo do Access **Revenda.accdb** do diretório “server2000\Prg Micro” para seu pendrive e abra o arquivo dando-lhe um duplo click

Esse arquivo possui tabelas contendo registros de veículos e de vendedores em uma revenda de veículos usados. Veja na figura abaixo:



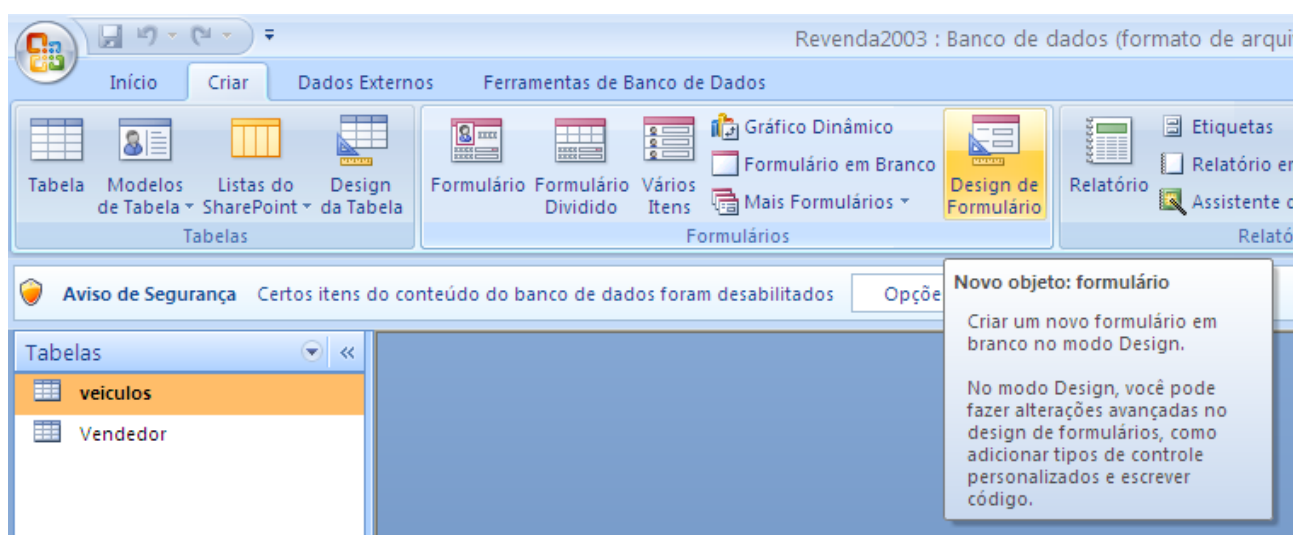
OBS.:

Você provavelmente lembra que o Access possui uma série de funcionalidades disponíveis a seus usuários, tais como a criação e uso de consultas personalizadas, filtros, vários tipos de classificação, etc.

Em nossa disciplina estaremos interessados nos recursos oferecidos através de programação em VBA, que é feito por meio de formulários.

Uso de formulários com Access:

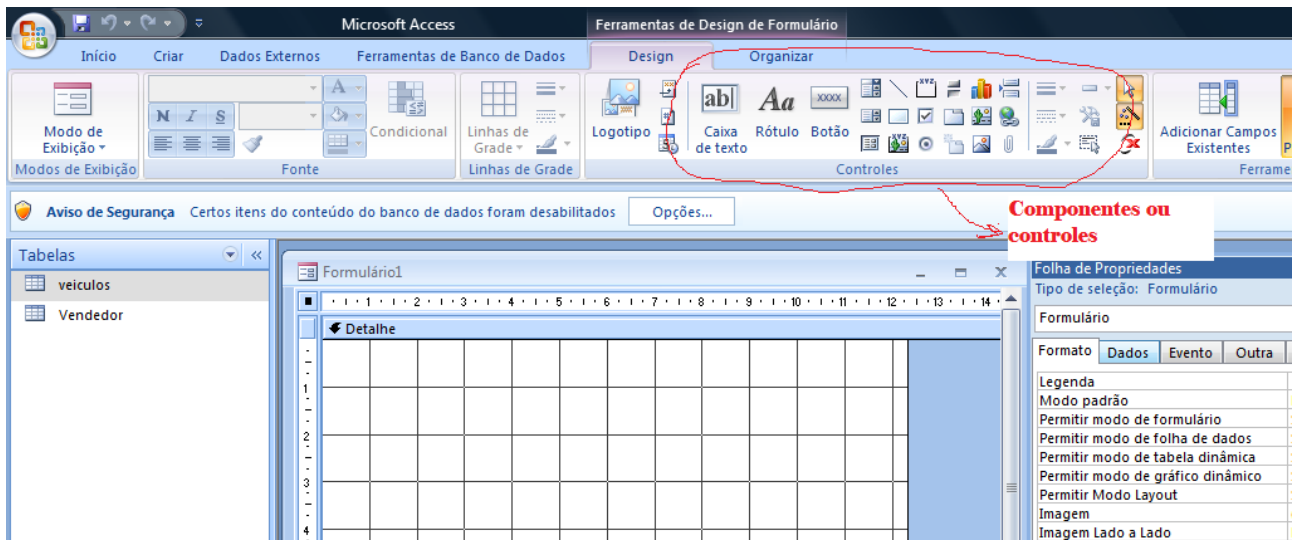
Selecione a tabela “veículos”, selecione a guia “criar” e dê um click no ícone “Design de Formulário”



Crie o formulário a seguir:

Na figura abaixo é exibida a janela “Propriedades”, o Formulário e os controles que você pode colocar em seu formulário.

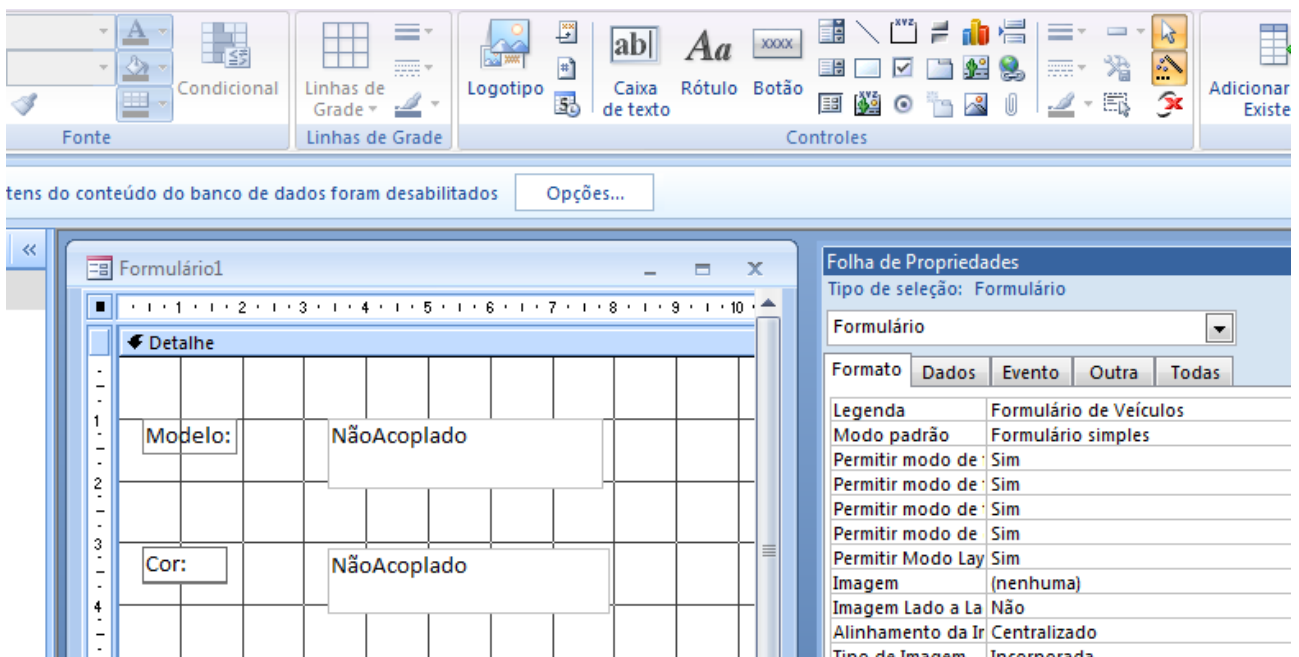
Se a janela Propriedades não estiver visível, dê um click com o botão direito do mouse no formulário e, no menu popup que abrir, selecione o item “Propriedades”.



Na caixa de listagem da janela de propriedades, selecione “Formulário” e na guia Formato selecione o item Legenda.

Digite o texto para a barra de título do formulário: “Formulário de Veículos”.

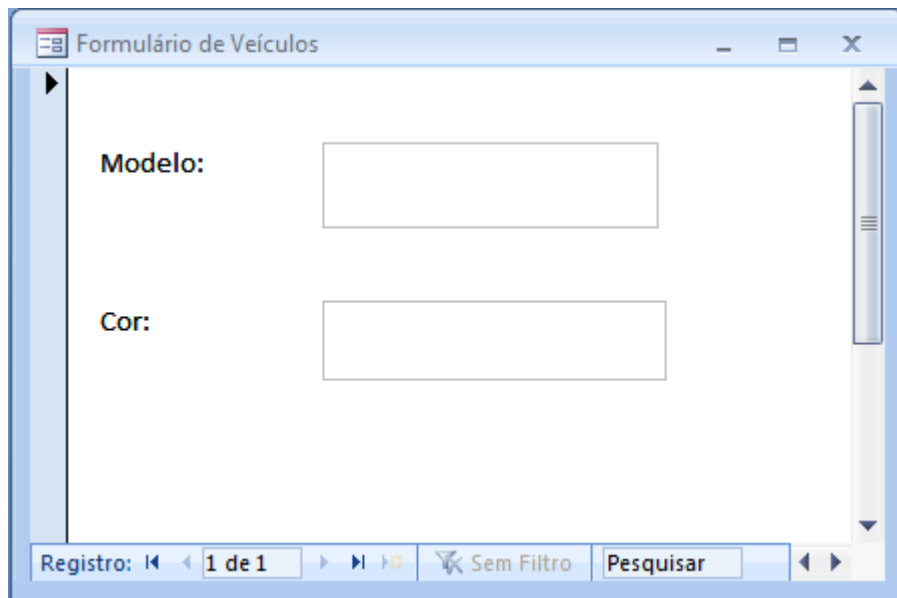
Em seguida, coloque duas caixas de texto no formulário e formate-as como abaixo:



Execute o formulário, pressionando a tecla F5. Seu formulário, em execução, deverá ficar parecido com a figura a seguir:

Obs.:

- 1) Se o formulário estiver maximizado, não será exibida a barra de título;
- 2) Se na guia "Todas", a propriedade do formulário “estilo da borda” estiver “Nenhum”, não será exibida a barra de título.
- 3) Para que o formulário não fique maximizado, na guia "Todas", configure para "Sim", a propriedade "PopUp" do formulário.



Para voltar ao modo de design, click no formulário com o botão direito do mouse e selecione “Modo Design”.

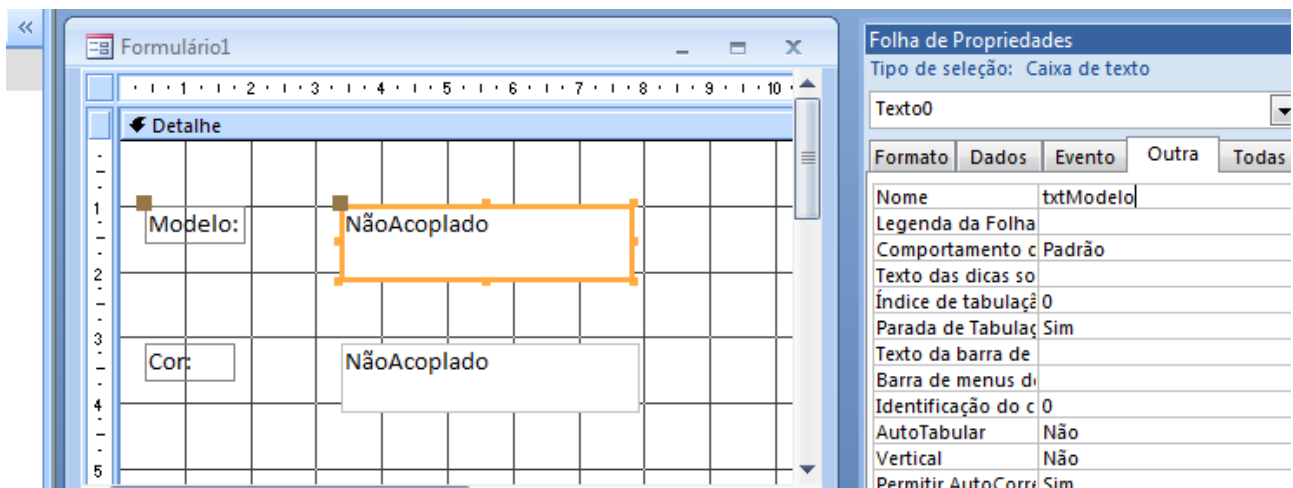
Vamos colocar nomes coerentes nos componentes do formulário. Os nomes deverão representar o tipo do componente e o que irão conter. Para representar o tipo de componente, usaremos a seguinte padronização:

- Nomes de caixa de texto devem iniciarão pelas letras txt – exemplo: txtModelo, txtCor etc.
- Nome de Label iniciará pelas letras lbl – exemplo: lblModelo, lblCor etc.
- Botão de comando iniciará pelas letras cmd – exemplo: cmdOk, cmdCancela etc.
- Botão de opção deverá iniciar pelas letras opt – exemplo: optopcao1, optopcao2 etc.

Outros tipos de nomes de componentes serão vistos oportunamente.

A padronização de nomes para componentes é muito importante para documentação e tem por objetivo facilitar o desenvolvimento e manutenção de programas.

Vamos atribuir nomes às caixas de texto de nosso formulário. Clique na caixa de texto que conterà o Modelo do veículo para selecioná-la, selecione a guia “Outra” e altere o Nome para txtModelo. Faça o mesmo para a caixa que conterà a cor do veículo.

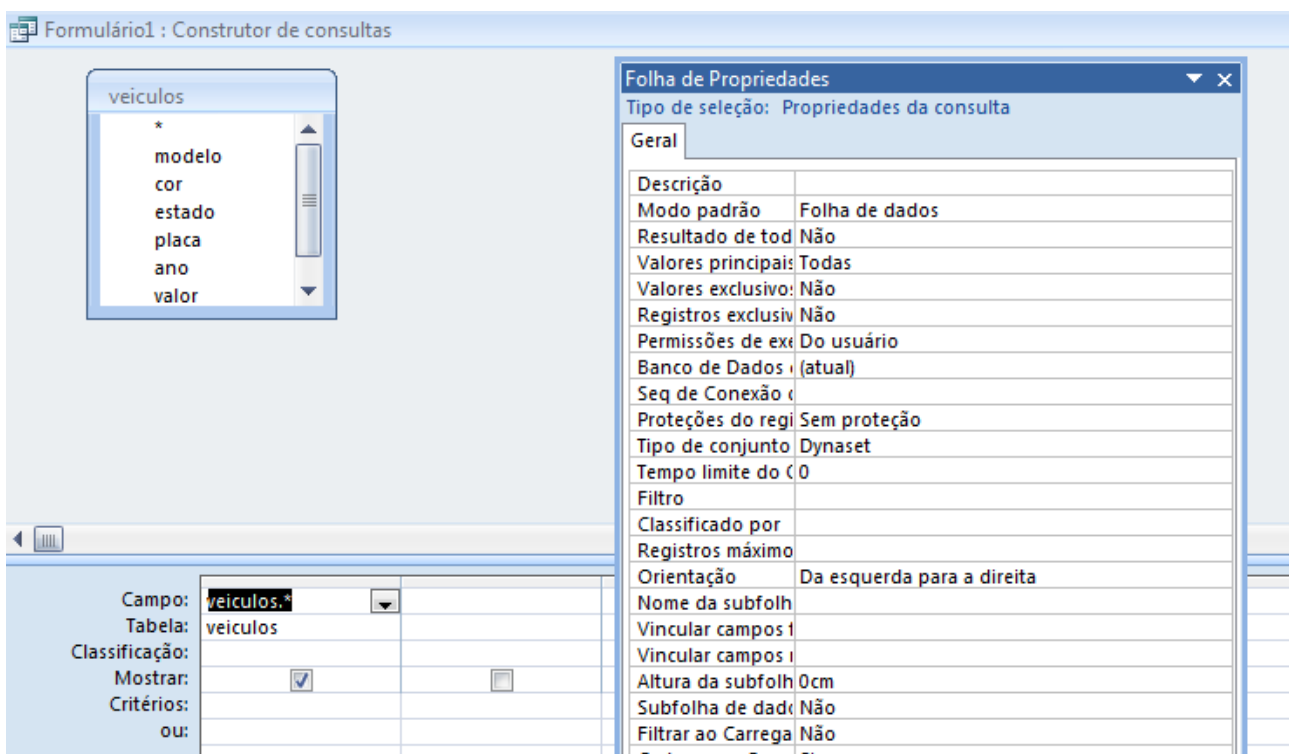


Vamos incluir em nosso formulário, a tabela de onde virão os dados que irão aparecer nas caixas de texto. Para isso, selecione o item “Formulário” na caixa de listagem da janela de propriedades, e selecione a guia “Dados”.


Selecione o item Fonte de registros e dê um click no ícone “...”. Selecione a tabela “veículos”, dê um click em Adicionar, em seguida dê um click no item Fechar.

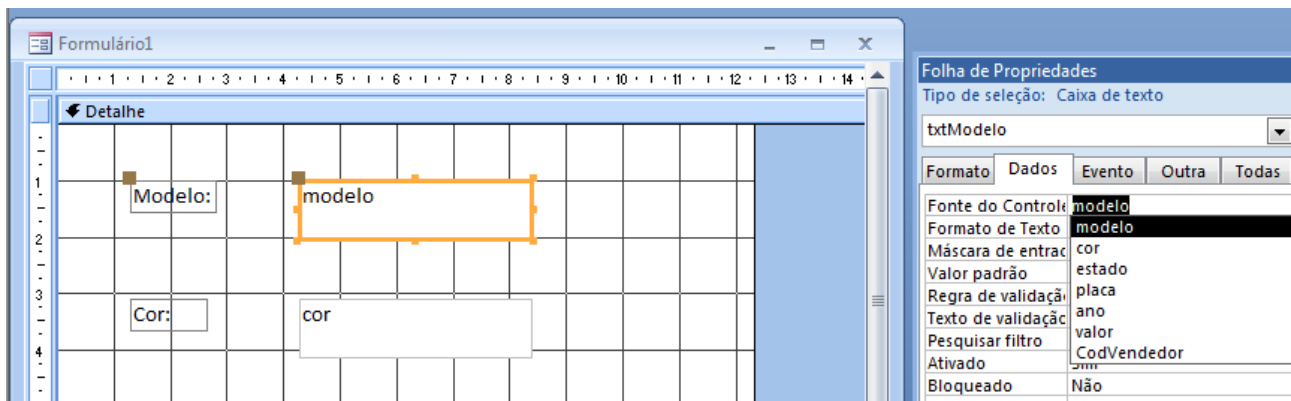
Na escolha do campo, selecione o item “veiculos.*”. Isso irá incluir em seu formulário, a tabela inteira.

Feche a janela “Formulário1 – construtor de consultas” e salve-a. Veja a figura a seguir:

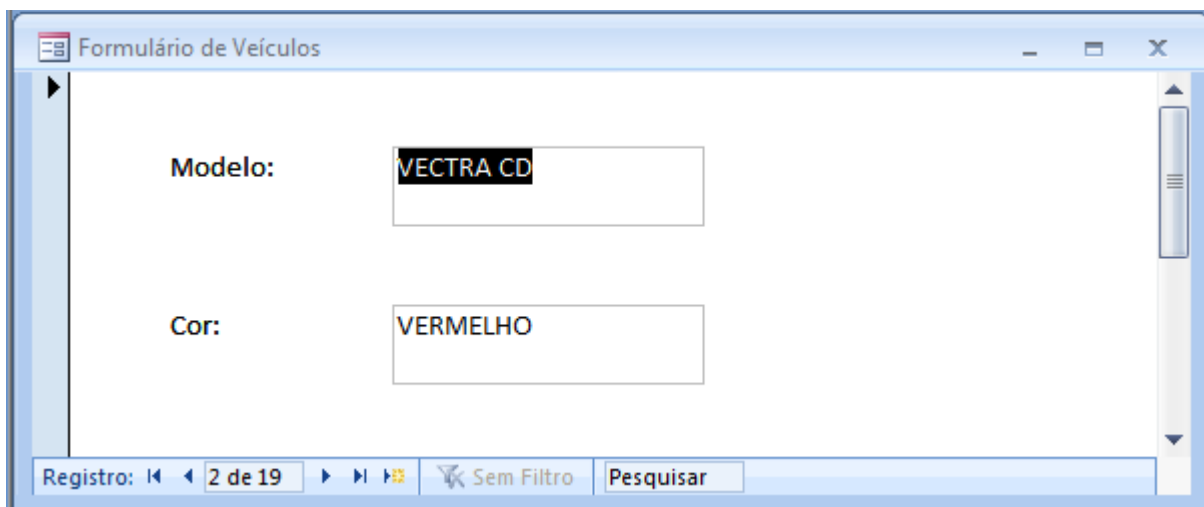


Vamos agora incluir os campos do banco de dados nos componentes do formulário:

- Selecione a caixa de texto txtModelo, clicando nela com o botão direito do mouse.
- Escolha o item propriedades.
- Na guia Dados, abra o ícone  da “Fonte de controle” e selecione o campo modelo,
- Faça o mesmo para o campo cor.



Ponha o foco no formulário e pressione a tecla F5 para ver o formulário em execução:



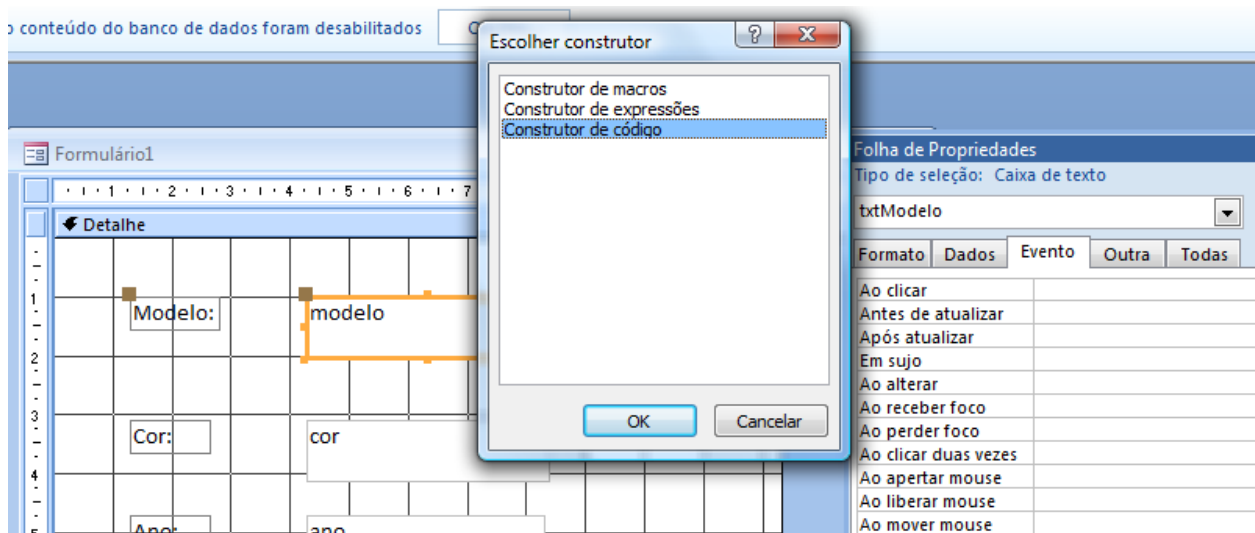
Obs. Se usarmos o assistente para criação de formulário, esses passos serão feitos automaticamente. Optamos por fazer manualmente visando a aquisição de habilidades que usaremos posteriormente.

Exercícios:

- 1 - Inclua os campos ano e valor no formulário, **com nomes txtAno e txtValor**. O campo ano deve ser no formato número geral com zero decimais e o campo valor deve ser no formato unidade monetária com 2 decimais. As especificações do tipo do campo são definidas na guia "Formato".
- 2 - Fazer programação para que ao ser inserido um novo registro, o que for digitado no campo modelo, fique em maiúsculo.

Procedimento:

- a) Selecione a caixa de texto txtModelo e na janela "Folha de Propriedades" escolha a guia Evento.
- b) Escolha o item "Ao pressionar tecla" e dê um click nos "..." da direita da linha.
- c) Na janela "Escolher construtor", selecione "Construtor de código" e dê um click no botão Ok.



Será oferecido para programação, o tratador do evento KeyPress, que deverá ser programado da seguinte forma:

```
Private Sub txtModelo_KeyPress( KeyAscii As Integer )
    KeyAscii = Asc ( UCase ( Chr$ ( KeyAscii ) ) ) ' Deixa os caracteres em maiúsculo.
End Sub
```

Obs: Para alternar entre a janela do VBA e do banco de dados, pressione Alt + F11.

3 - Fazer programação para que o ano de um novo veículo a ser digitado seja validado. Ou seja, o ano só é válido se estiver entre 1900 e o ano atual + 1.

Para isso, selecione a caixa de texto txtAno e programe o evento “Antes de atualizar”, como abaixo:

```
Private Sub txtAno_BeforeUpdate(Cancel As Integer)
    If Val(txtAno.text) < 1900 Or Val(txtAno.text) > Year(Date) + 1 Then
        MsgBox " Ano inválido... Digite o ano correto!", vbCritical, "A T E N Ç Ã O"

        Cancel = True 'Cancela a atualização
    End If
End Sub
```

4 – O campo “valor” deve aceitar apenas valores numéricos e a vírgula decimal. A programação é a seguinte:

```
Private Sub txtValor_KeyPress(KeyAscii As Integer)
    Dim TECLA As String
    TECLA = Chr$(KeyAscii)
    If (TECLA < "0" Or TECLA > "9") And TECLA <> "," Then
        Beep
        KeyAscii = 0
    End If
End Sub
```

Teste a programação feita nos itens 1 a 4 acima e chame o professor para avaliação.

Aula 9 Programação - em VBA com ACCESS

Abra o arquivo “Revenda” e teste o formulário, que deve apresentar as seguintes funcionalidades:

- 1 - Exibe os campos Modelo, Cor, Ano e Valor dos veículos do banco de dados.
- 2- Os caracteres digitados no campo Modelo ficam com caracteres maiúsculos.
- 3 – O campo Ano só aceita valores entre 1900 e o ano atual + 1.
- 4 – O campo Valor só aceita números e a vírgula decimal.

Após testar e verificar os itens acima, faça os exercícios seguintes:

- 1 – O campo Ano deve aceitar apenas a digitação de números.
- 2 – O campo Ano deve aceitar a digitação de no máximo 4 caracteres. Utilize a função LEN que retorna o comprimento de um string – use o help para verificar.
- 3 - Se no campo Ano, o usuário tentar digitar mais de 4 caracteres, exibir mensagem de alerta.
- 4 – No campo Cor todos os caracteres digitados dever ficar em maiúsculo.
- 5 – Os campos Ano e Valor devem aceitar a tecla BackSpace. O código ASCII dessa tecla é 8. Portanto, se keyAscii for igual a 8, a tecla será válida.
- 6 – Se uma tecla inválida for digitada nos campos Ano ou Valor, o programa deverá exibir mensagem de alerta, como por exemplo: “Tecla inválida... Digite apenas números!”
- 7 – Inclua no formulário, os campos Estado e Placa. Todos os caracteres digitados devem ficar em maiúsculo.
- 8 – Insira no banco de dados, através dos botões de navegação do formulário, os registros abaixo:

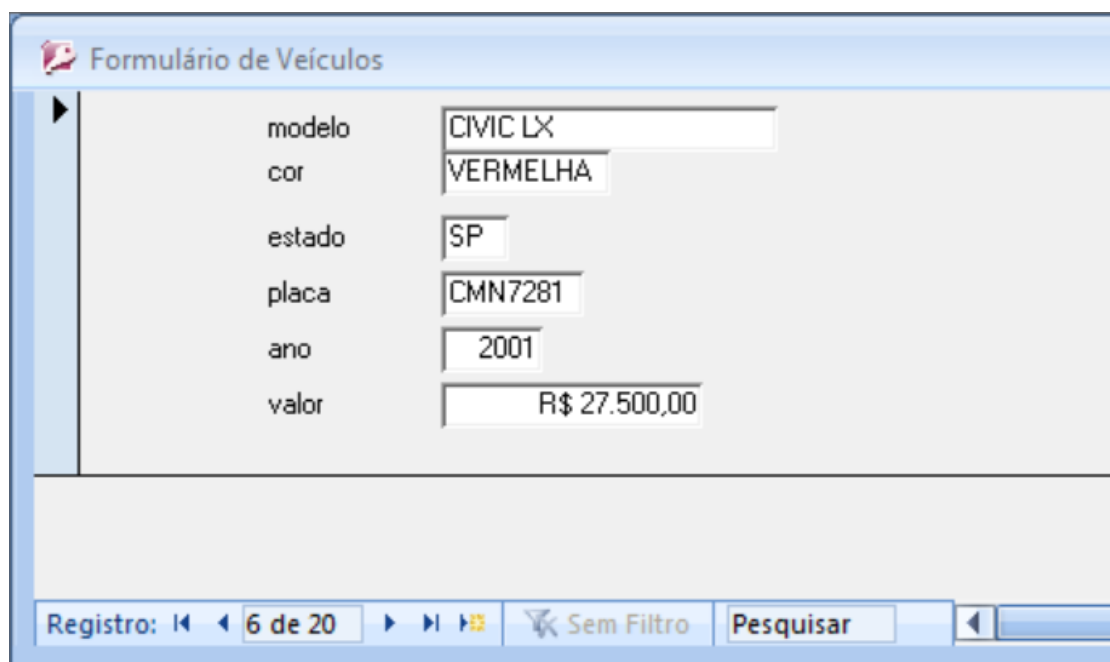
	MODELO	COR	ESTADO	PLACA	ANO	VALOR
1	CORSA	BEGE	RJ	CAU3254	1998	8000,00
2	CORSA	VERDE	SP	BBJ2429	2000	12000,00
3	CORSA	BRANCO	BA	EBF9553	2002	15000,00
4	CORSA	PRETO	SP	GTH8865	1998	7500,00

- 9 – Todos os campos devem aceitar a tecla “Tab”. O valor ascii em decimal da tecla Tab é 9. Pressionando-se a tecla Tab, a ordem de tabulação deve ser adequada ao usuário (de cima p/ baixo) – verifique.

Salvar o arquivo “Revenda” e preparar para apresentar o formulário ao professor.

Aula 10 Programação - em VBA com ACCESS

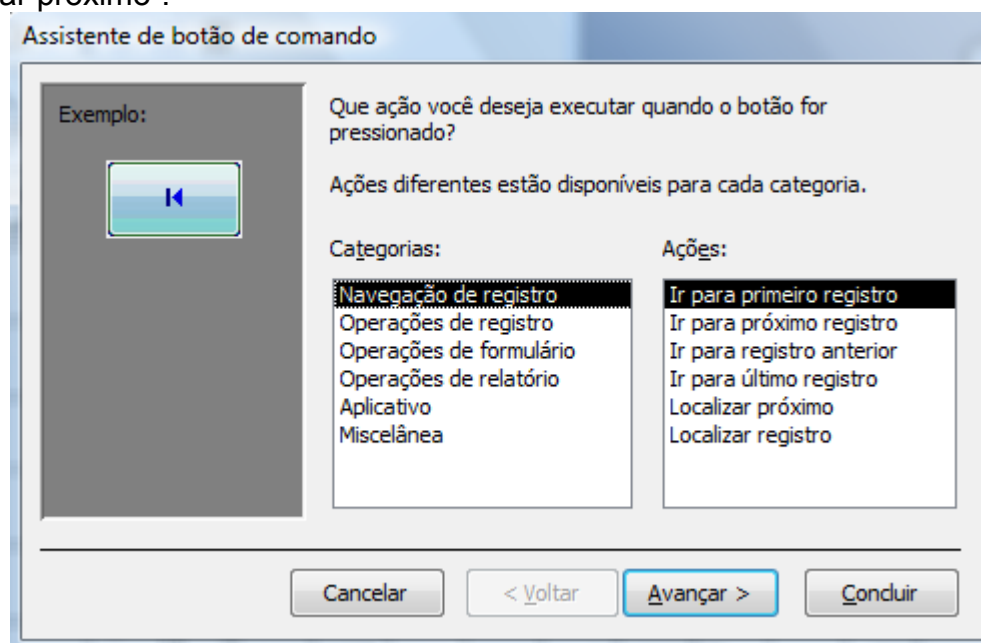
Abrir o arquivo Revenda utilizado na aula anterior. O formulário deve estar com a seguinte aparência:



Vamos personalizar um pouco nosso formulário e depois iremos remover o componente padrão de navegação de registros. Para isso, vamos selecionar o ícone do componente “botão de comando” na guia Design, e vamos incluí-lo no formulário. Note que o VBA abre uma janela para escolhermos qual a categoria de botão nós queremos. Vamos selecionar a categoria Navegação de registro.

Inclusão de botões no formulário do Access:

Vamos inserir em nosso formulário, **todos os botões dessa categoria**, exceto o botão “localizar próximo”.



Para cada botão que você inserir, click no botão Avançar e configure as propriedades do botão a seu gosto. É muito importante atribuir um nome de acordo com nossa padronização, como por exemplo, para o botão “Ir para primeiro registro”, o nome cmdPrimeiroRegistro, cmdProximo etc..

Faça isso para todos os botões.

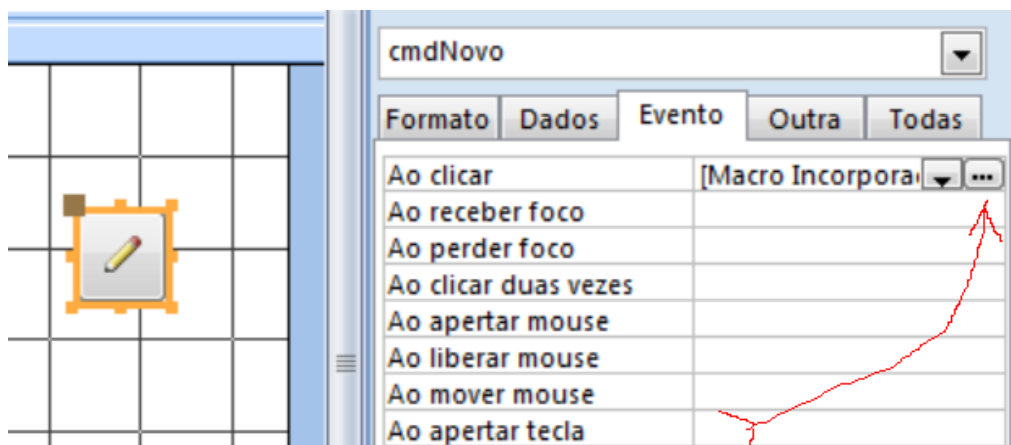
Exercício: Coloque mais dois botões da categoria Operações de registro: Adicionar novo registro e Excluir registro. Mude seus nomes para cmdAdicionaNovo e cmdExcluir, por exemplo.

Nosso formulário ficará mais ou menos assim:



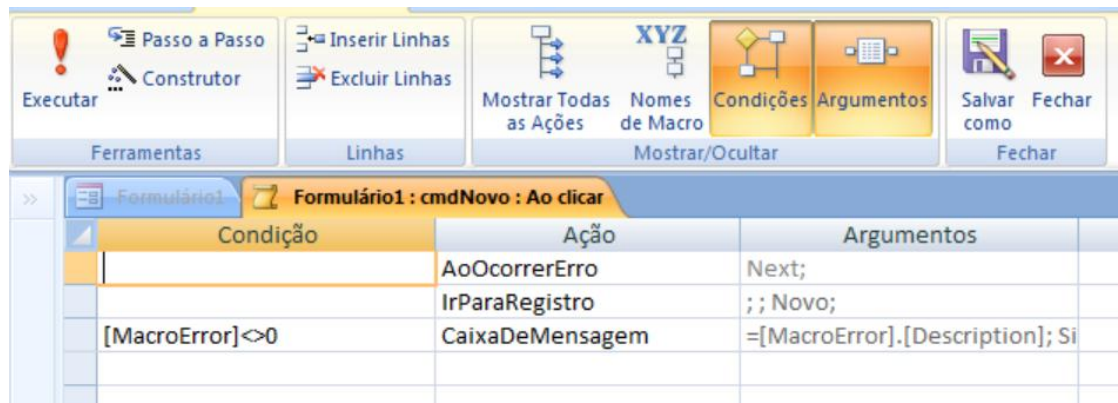
Agora que já temos nossos próprios botões de controle dos registros, não precisamos mais do controle fornecido automaticamente com o relatório. Para eliminá-lo, no modo de design do formulário, coloque “Não” no item Formulário / Formato / Botões de navegação.

Execute o formulário para verificar e teste todos os botões que você colocou. Note que para cada botão que você incluiu, o editor de VBA do Access providenciou uma "Macro Incorporada" que faz o trabalho de acordo com as funções do botão.



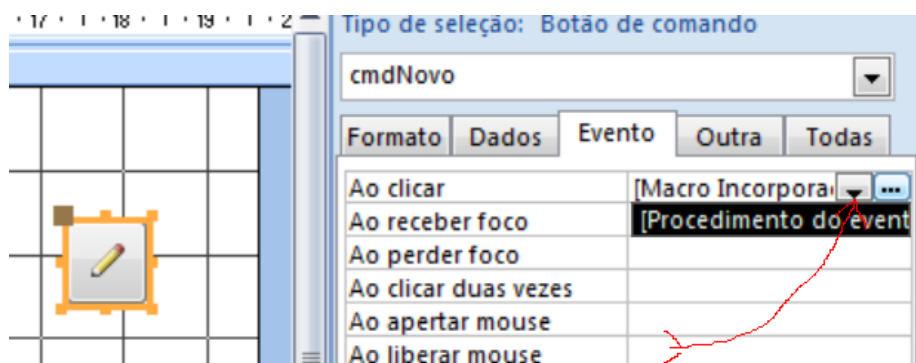
Se for essa a única ação esperada quando ocorrer o click no botão, então está tudo pronto para uso desse botão pois essa macro faz apenas isso. Veja as instruções da macro, dando um click nos (...), apontado pela seta da figura acima.

Aí está a macro incorporada! Ela é feita com instruções especiais diferentes de VBA.



Suponha que após pressionarmos o botão cmdNovo, queremos que o foco vá para a caixa de texto txtModelo ou queremos realizar algumas outras ações. Nesse caso não devemos usar o recurso de macro incorporada e sim o procedimento do evento.

Para isso, dê um click no botão apontado na figura abaixo, selecione o "Procedimento do evento" e click no botão com os (...).



E lá está nosso velho conhecido, tratador do evento Click do botão cmdNovo.

Digite nele o código abaixo e teste-o:

```
Private Sub cmdNovo_Click()
On Error GoTo MensagemErro

docmd.GoToRecord ,acNewRec

Exit Sub

MensagemErro:
MsgBox Err.Description
End Sub
```

Verifique que o código para inserir um novo registro no banco de dados funciona do mesmo modo que quando havia a macro incorporada. Porém, agora podemos incluir a codificação que quisermos, como por exemplo, após o novo registro ser criado, mandamos o foco para a caixa de texto txtNome. A instrução que faz isso é:

```
txtNome.setfocus.
```

Onde você irá incluí-la? Faça isso e teste novamente o uso do botão.

Uso de DoCmd:

Note a palavra “DoCmd” no código acima. Dizemos que DoCmd é um objeto do modelo de objetos do Access. O objeto DoCmd suporta vários “métodos” que provocam várias ações na programação em VBA para o Access.

Veja a seguir, a programação em VBA associada ao evento click no botão cmdProximo, onde o objeto DoCmd executa o método GoToRecord, com o argumento acNext:

```
Private Sub cmdProximo_Click()  
On Error GoTo MensagemErro  
  
DoCmd.GoToRecord , , acNext  
  
Exit Sub  
  
MensagemErro:  
MsgBox Err.Description  
End Sub
```

Outros exemplos de uso de DoCmd:

DoCmd.Maximize ' maximiza o formulário onde está inserido.

DoCmd.Close ' fecha o formulário.

DoCmd.Quit ' fecha o Access.

DoCmd.OpenForm "frmMenu" 'exibe o formulário de nome frmMenu.

Exercícios Access Aula 2:

Obs.: Para incluir um botão comum no formulário proceda da seguinte maneira: Inclua um botão no formulário e na janela apresentada, onde aparecem as categorias de botão oferecidas, pressione a tecla Cancelar.
Daí basta configurar suas propriedades - seu nome, etc.

De acordo com a observação acima, inclua no formulário um botão de nome cmdFim que ao ser pressionado encerra a execução do Access. Conforme visto há pouco nos exemplos de uso de DoCmd, a programação será:

```
Private Sub cmdFim_Click()  
  
DoCmd.Quit  
  
End Sub
```

Especificações para realização do próximo exercício - será realizado na próxima aula:

Se um veículo do banco de dados for vendido, o vendedor deverá informar a venda, a fim de receber sua comissão.

Para que o vendedor possa fazer isso, inclua no formulário, o botão "Vender", de nome "cmdVender" que ao ser pressionado solicita o código do vendedor. Esse código deverá ser inserido no campo "codvendedor" do banco de dados Veículos. A digitação do código poderá ser feita por uma InputBox.

Na janela de propriedades, selecione a guia Outra e mude o nome para cmdVender. Na guia Formato, mude a legenda para &Vender. Com a legenda dessa maneira, você poderá executar o programa associado ao botão, pressionando Alt + V.

O código do vendedor a ser digitado deverá possuir dígito de controle que será verificado através de cálculo pelo módulo 11.

O dígito verificador (dv), ou dígito de controle (dc) calculado pelo módulo 11 é bastante seguro, e é usado para validação de CPF, RG, CGC e vários outros códigos oficiais.

Um dígito de controle é calculado pelo módulo 11 da seguinte maneira : Por exemplo, supondo-se o código de vendedor 1210396, fazem-se os cálculos.

$$\begin{array}{r} 6 \times 1 = 6 \\ 9 \times 2 = 18 \\ 3 \times 3 = 9 \\ 0 \times 4 = 0 \\ 1 \times 5 = 5 \\ 2 \times 6 = 12 \\ 1 \times 7 = 7 \\ + \quad \text{-----} \\ 57 \end{array}$$

Pega-se o resultado da soma (57), divide-se por 11 e pega-se o resto, que neste caso é 2.

$$57 / 11 = 5 \quad \text{Resto} = 2$$

Daí, faz-se $11 - 2 = 9$. Pronto. O dígito calculado é 9, e o código do vendedor, com dígito, ficará: 1210396-9

Se o resto da divisão por 11 for 0 ou 1, o dígito é o próprio resto.

Na próxima aula implementaremos o programa que recebe um código e faz a validação desse código usando o processo do módulo 11, explicado acima.

Salve o arquivo e prepare-se para apresentar o trabalho da aula de hoje.

Aula 11 Programação - em VBA com ACCESS

Conforme explicado na aula anterior, vamos implementar em nosso projeto a validação de um código de vendedor, através do módulo 11.

Acompanhe o programa a seguir:

```
Private Sub cmdVender_Click()  
    Dim codigolido As Long  
    Dim digitolido As Integer  
    Dim codigo As Long  
    Dim digito As Integer  
    Dim soma As Integer  
    Dim mult As Integer  
    mult = 1  
    soma = 0  
  
    codigolido = Val(InputBox(" Digite o código do vendedor: ", " V e n d a d e V e í c u l o"))  
    digitolido = codigolido Mod 10  
    codigo = Int(codigolido / 10)  
  
    While codigo > 0  
        soma = soma + (codigo Mod 10) * mult  
        mult = mult + 1  
        codigo = Int(codigo / 10)  
    Wend  
  
    digito = soma Mod 11  
    If digito > 1 Then digito = 11 - digito  
  
    MsgBox "Digito lido = " & digitolido & " Digito calculado = " & digito  
  
End Sub
```

O programa acima recebe um código numérico de vendedor com dígito, separa esse dígito e calcula, com o restante do número lido, um outro dígito com a finalidade de verificar se o dígito calculado confere com o dígito lido.

Copie o programa acima para o tratador de evento click do botão cmdVender de seu formulário, execute o programa e teste-o com seu número de matrícula. Faça vários testes com outros números - inclusive com números errados.

Exercícios:

- 1 – Verificar se o usuário digitou apenas números. Caso tenha sido digitado algum caractere não numérico, exibir mensagem de erro e abandonar o procedimento. Utilize a função IsNumeric. A programação ficará assim:

```
Dim codigostr As String
codigostr = InputBox(" Digite o código do vendedor: ", "  V e n d a  d e  V e í c u l o")
' verifica se foram digitados só numeros.
If Not IsNumeric(codigostr) Then
    MsgBox "Número inválido! Digite apenas números...", vbCritical, "ATENÇÃO"
Exit Sub
End If
```

2 - Verificar se o dígito calculado é igual ao dígito lido. Se for igual, aceitar o código do vendedor e incluí-lo na tabela “Veiculos”, no campo codvendedor, e exibir mensagem “Venda efetuada com sucesso!”

3 - Se o dígito não “bater”, exibir mensagem de erro ao usuário e abandonar o procedimento.

O código será mais ou menos como abaixo:

```
If digitolido = digito Then
    codvendedor = codigolido ' codvendedor é campo do Banco de Dados.

    ' A instrução abaixo salva o registro corrente - grava o codigo do vendedor.
    DoCmd.RunCommand acCmdRefresh

    MsgBox "Venda efetuada com sucesso!"
Else
    MsgBox "Digito não bate. Tente novamente....", vbExclamation, "A T E N Ç Ã O"
Exit Sub
End If
```

4 – Se a venda for concretizada e após ser inserido com sucesso o código do vendedor, o botão cmdVender deverá ficar **desabilitado**, com caption “Vendido”. Usar a instrução cmdVender.enabled = false.

5 – Ao navegar pelo banco de dados, se o veículo estiver à venda, ou seja, se o campo de código de vendedor ainda estiver em branco, o botão cmdVender deverá ficar **habilitado**, e com caption = &Vender. Caso contrário, ou seja, se o veículo já foi vendido, consequentemente o campo “codvendedor” estará preenchido. Nesse caso, o botão cmdVender deverá estar **desabilitado**, com caption “Vendido”.
A programação será mais ou menos como a seguir:

Obs.: No Access, para verificar se um campo do banco de dados contém o caráter nulo, usamos a função IsNull. Veja o exemplo abaixo:

```
If IsNull(codvendedor) Then
    cmdVender.Caption = "&Vender"
    cmdVender.Enabled = True
Else
    cmdVender.Caption = "Vendido"
    cmdVender.Enabled = False
End If
```

A programação acima deverá ser codificada em todos os botões de navegação.

6 – Ao pressionar o botão “Inserir Registro” o foco deve ir para a caixa de texto txtModelo. Verifique se a ordem de tabulação está correta. Se não estiver, corrija-a. A propriedade que especifica a ordem de tabulação é “Índice de tabulação”. Esta inicia por zero e vai seqüencialmente de acordo com a tabulação desejada.

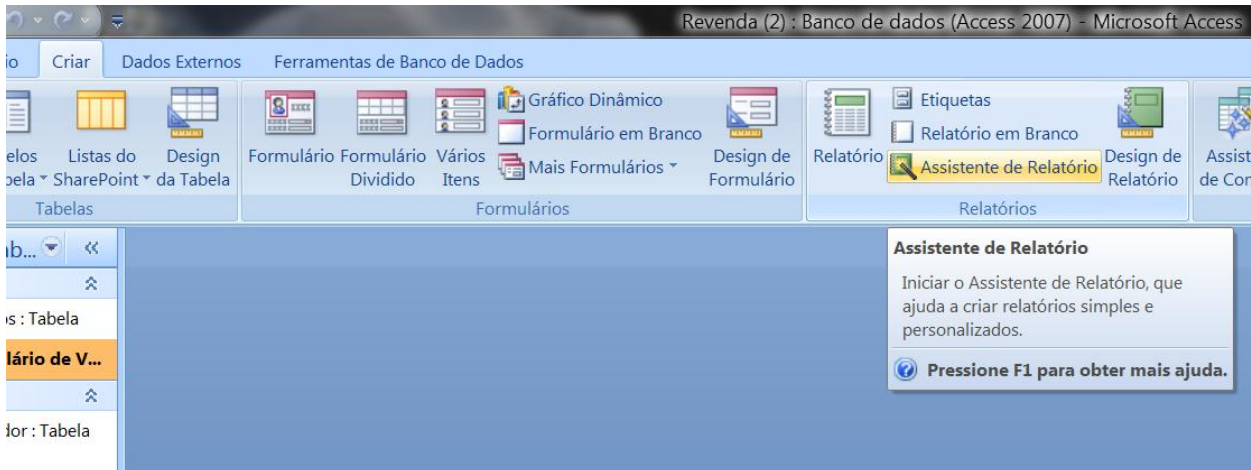
SALVE O PROJETO E PREPARE-SE P/ APRESENTÁ-LO AO PROFESSOR.

Aula 12 - Programação em VBA com ACCESS

Uso de relatórios com Access: Execute os procedimentos a seguir:

Criação do relatório de veículos.

Vamos criar um relatório que mostre os veículos de nosso banco de dados. Dê um click na guia Criar e escolha Assistente de Relatório.



Através do assistente, crie o relatório de nome Relat_Veiculos para exibição dos dados da tabela "Veiculos". Os registros deverão ser exibidos em ordem alfabética de modelo de veículo, com os agrupamentos (quebras com totais): A cada quebra de modelo deverá haver um subtotal de valor por modelo. Vide exemplo de relatório abaixo:

Mude o relatório para modo design e configure suas propriedades Pop up - Sim e Janela restrita - Sim

The screenshot shows a report titled 'Veiculos' in Design View. The report contains a table with the following data:

Modelo	Cor	Estado	Placa	Ano	Valor
ASTRA GLS	PEROLA	SP	DEM9416	2004	R\$ 32.500,00
	PEROLIZ	RJ	OQC4676	2006	R\$ 25.500,00
	CINZA	RJ	CRI6377	2011	R\$ 38.340,00
Resumo para 'Modelo' = ASTRA GLS (3 registros de detalhe)					
Soma					R\$ 96.340,00
AUDI A9	VERMELHO	SP	AAA1111	2013	R\$ 500.000,00
Resumo para 'Modelo' = AUDI A9 (1 registro de detalhe)					
Soma					R\$ 500.000,00
BLAZER STD	BEGE	RJ	CET9935	2001	R\$ 17.500,00
Resumo para 'Modelo' = BLAZER STD (1 registro de detalhe)					
Soma					R\$ 17.500,00
CIVIC LS	VERMELHA	SP	CMN7281	2004	R\$ 27.500,00
Resumo para 'Modelo' = CIVIC LS (1 registro de detalhe)					
Soma					R\$ 27.500,00
CIVIC LX	PRETA	RJ	DEEL861	2005	R\$ 22.000,00
	BRANCA	SP	CEI13544	2010	R\$ 55.000,00

Obs.: Se o relatório não estiver adequado ao ser criado pelo assistente, dê-lhe um click com o botão direito do mouse e escolha "Modo Design". Aí você pode configurá-lo como quiser.

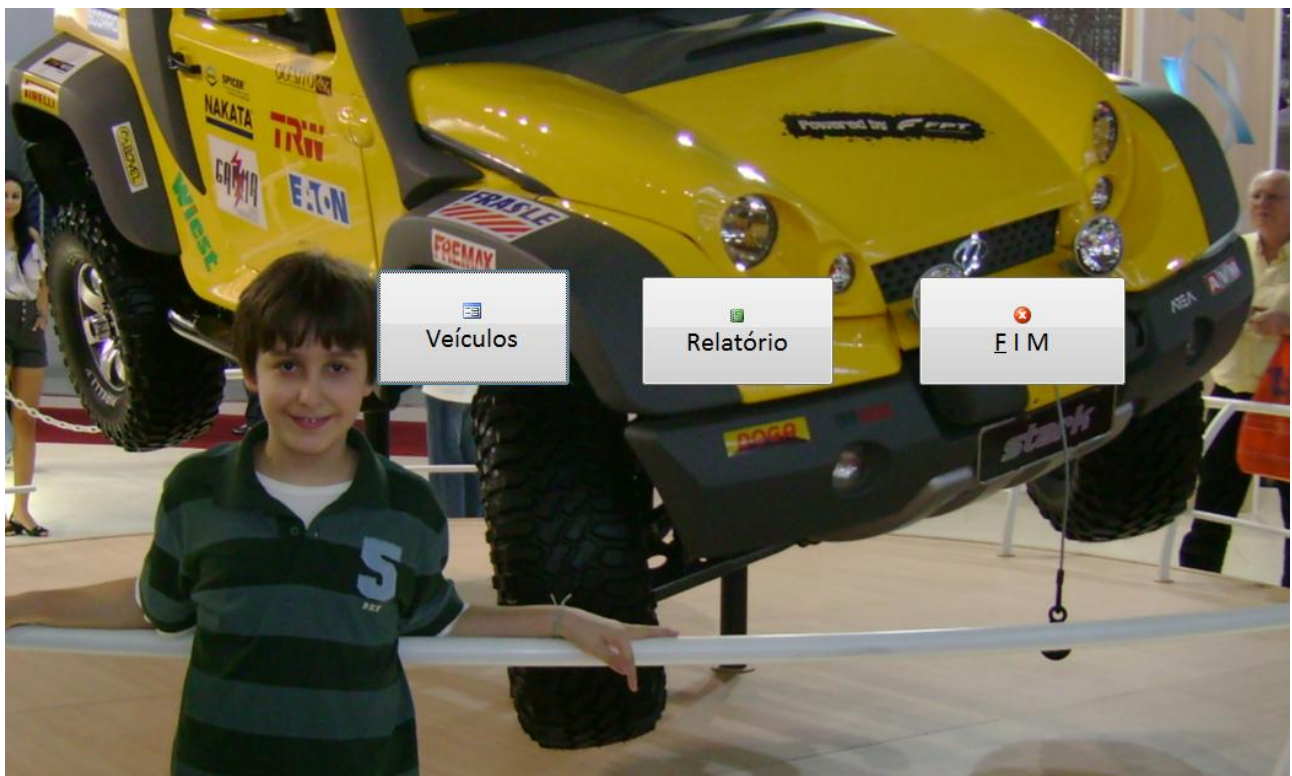
Por exemplo, em modo design, configure as caixas de texto de valores e soma de valores para unidade monetária.

Inclua mais um formulário de nome frmPrincipal, que será o formulário pelo qual a aplicação irá iniciar, quando abrirmos o arquivo “Revenda”.

Esse formulário deverá ter 3 botões, sendo:

- a) - o primeiro botão exibe o formulário frmVeiculos.
Insira o botão e escolha "Operações de formulário / Abrir formulário ..."
Escolha o formulário a abrir.
Mude o nome do botão para cmdExibeFrmVeiculos.
Na janela de propriedades, inclua a legenda Veículos.
Em Organização da legenda, escolha "Inferior".
- b) - o segundo botão exibe o relatório de veículos que você fez no início desta aula.
Insira o botão e escolha "Operações de relatório / Abrir relatório..."
Selecione o relatório a abrir (só tem um).
Mude o nome do botão para cmdExibeRelatorio.
Na janela de propriedades, inclua a legenda Relatório.
- c) - o terceiro botão encerra o processamento (nome=cmdFim).
Insira o botão e escolha Aplicativo / Encerrar aplicativo.
Siga os passos anteriores para configurar o botão.

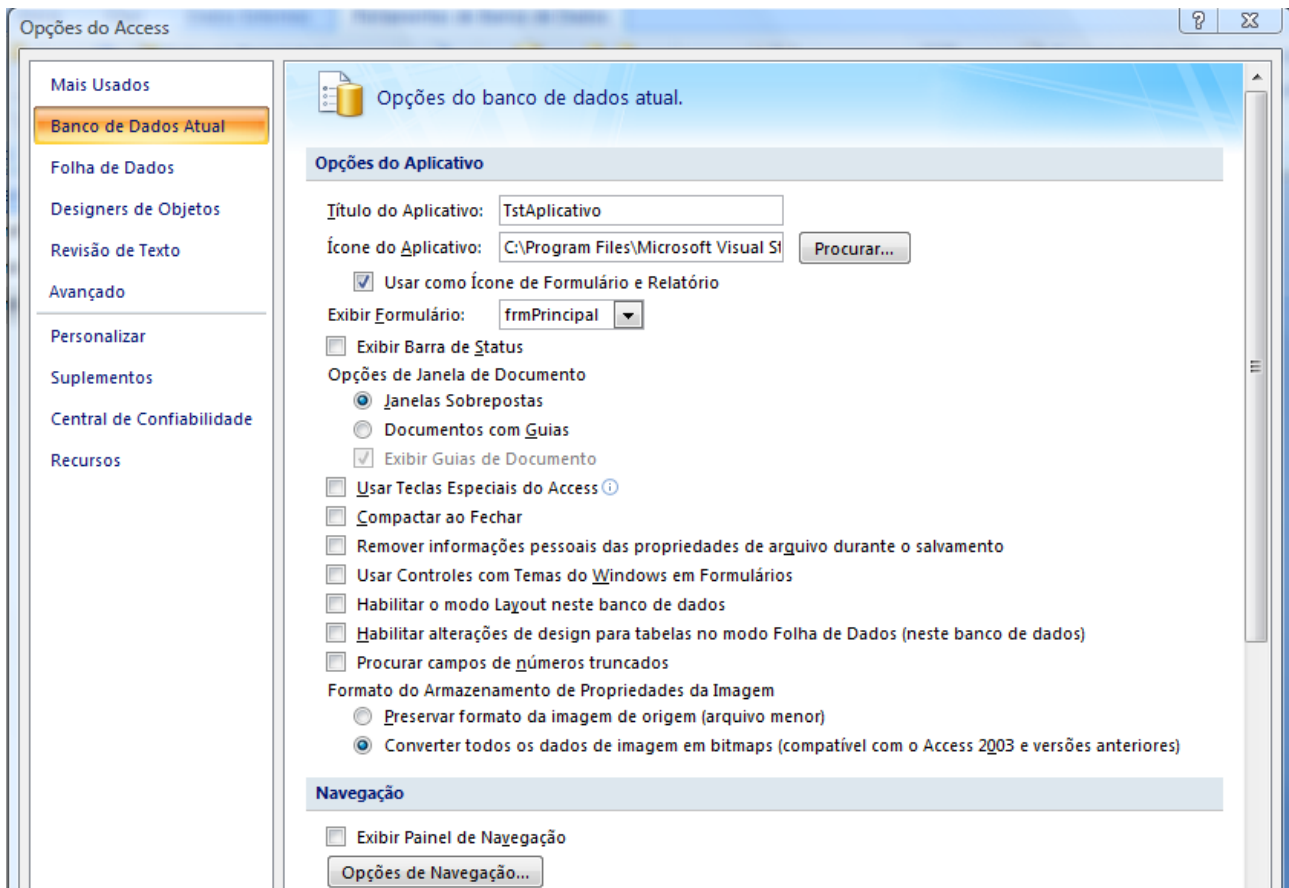
Layout do formulário frmPrincipal:



6- Finalizando...

Vamos configurar as condições de inicialização de nosso sistema.

- a) Para que o usuário não tenha acesso aos controles de formulário, selecione o formulário e configure a propriedade “Estilo da borda” para “Nenhum”. Configure também a propriedade “Seletores de registro” para “Não”
- b) Programar o evento Form_Load() do formulário, para que ao ser carregado, o mesmo fique maximizado:
- ```
Form_Load()
 doCmd.Maximize
End Sub
```
- c) Para que os relatórios fiquem visíveis (na frente do formulário que os chama), configurar suas propriedades PopUp Sim e Janela restrita Sim.
- d) Configure o Access para que o usuário não possa acessar seus recursos. Para isso, dê um click no botão Access / Opções do Access / Banco de Dados Atual, e configurar a janela apresentada de acordo com a figura a seguir.
- e) Atribuir um nome ao aplicativo, incluir um ícone no aplicativo, **desligar todas as caixas de seleção**, selecionar o formulário a iniciar a execução (frmPrincipal), dar um click no botão Ok e salvar o arquivo Revenda. Veja a janela a seguir - **Note que a imagem mostra apenas uma parte das configurações, ao descer a tela, há outras opções que devem ser desligadas.**



Feche todas as janelas, salve o arquivo e feche o Access. Abra o arquivo Revenda e note que só é possível o acesso aos formulários e ao relatório que você criou. O usuário não tem acesso direto às tabelas e aos outros componentes do projeto.

Se você quiser acessar as tabelas ou outros recursos do Access, ao abrir o arquivo, pressione a tecla Shift e a mantenha pressionada até o arquivo abrir.

***Faça um teste geral em seu sistema e verifique se tudo funciona a contento, antes de chamar o professor para avaliação.***

**Salve tudo e prepare-se para a apresentação final do projeto.**

\*\*\*\*\*

## AUTOMATIZAÇÃO DE TAREFAS

O VBA também é utilizado em arquivos de script, ou seja, um arquivo texto com extensão ".vbs", contendo instruções que são interpretadas e executadas linha a linha, pelo programa "wscript.exe" do Windows. A execução ocorre ao darmos um duplo click no arquivo com extensão ".vbs".

Exemplo:

1) Clique com o botão direito do mouse na área de trabalho selecione Novo/Documento de Texto.

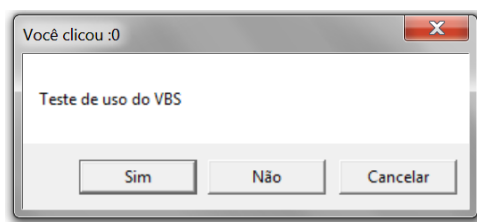
2) No novo documento, clique com o botão direito do mouse e selecione Editar.

3) Ao abrir o Bloco de Notas, digite o código abaixo (note que são instruções VBA).

```
dim variavel ' todas as variáveis em arquivos VBS são do tipo variant
variavel=0
do while variavel<4
 msgbox "Teste de uso do VBS",3,"Você clicou :" & variavel & " Vezes."
 variavel=variavel + 1
loop
```

4) Salve o arquivo: Arquivo/Salvar como.../Tipo/Todos os arquivos/teste1.VBS/Salvar

5) Execute o arquivo dando-lhe um duplo click e confira o resultado com a figura abaixo:



Note durante a execução, o ícone do programa wscript.exe na barra de tarefas do Windows.

Vamos estragar esse programa, pois somos programadores inexperientes e ainda fazemos muitas bobagens...

Altere a instrução `variavel=variavel + 1` para `variave1=variavel + 1`

Se você não prestar bastante atenção, não vai achar o erro, pois o programa não acusa nada, só que não faz o que se espera dele. Salve-o e teste novamente.

Esse tipo de problema é resolvido com o uso da cláusula Option Explicit, que obriga o programador a declarar todas as variáveis. Nosso programa agora ficará assim:

```
Option Explicit
dim variavel ' todas as variáveis em arquivos VBS são do tipo variant
variavel=0
do while variavel<4
 msgbox "Teste de uso do VBS",1,"Você clicou :" & variavel & " Vezes."
 variave1=variavel + 1
loop
```

Salve o programa e teste-o. Agora deve estar correto.

**Note que nosso programa anterior em VBScript tem um erro de lógica e não conseguimos encerrá-lo - ele está em looping infinito.**

Tente parar a execução do script pelos métodos tradicionais: Alt f4, Ctrl Break, etc.

Obs.: Caso seu programa em VBScript entre em looping, é necessário ativar o gerenciador de tarefas (Ctl+Alt+Del), acionar a guia "processos", selecionar o processo "wscript.exe" e clicar no botão "finalizar processo".

**Exercício1:** Modifique o programa para ficar com a seguinte codificação:

```
option explicit ' Obriga o programador a declarar todas as variáveis
dim variavel ' todas as variáveis em arquivos VBS são do tipo variant
dim testeTecla
variavel=0
do while variavel<4
 testeTecla=msgbox ("Teste de uso do VBS",1,"Você clicou :" & variavel)
 variavel=variavel + 1
 msgbox testeTecla
loop
```

Note que no programa acima, a variável testeTecla recebe o valor retornado pela função msgbox, quando esta recebe o click do mouse em um de seus dois botões.

Execute o script e verifique que ao ocorrer o click em um dos botões **Sim ou Cancelar**, a função msgbox retorna um dos valores 1 ou 2, respectivamente

## Exercício 2:

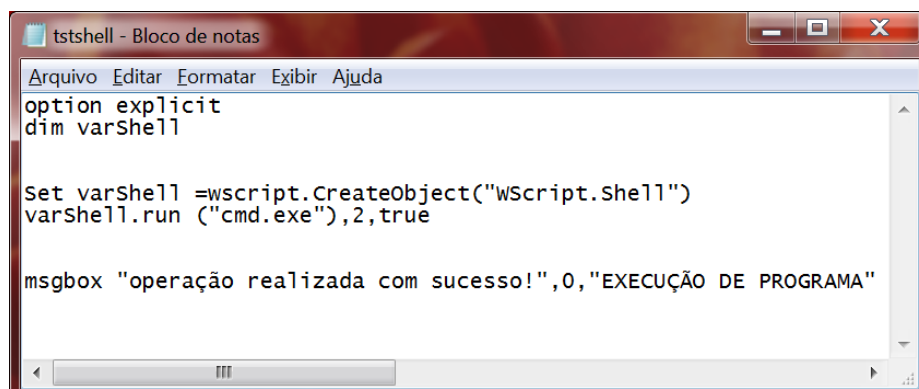
Modifique o programa para que ele continue contando os clicks por até 10 vezes. O programa deve informar, por meio de uma msgbox, em qual botão ocorreu o click. Se o click for no botão "**Cancelar**", o programa avisa que o click ocorreu nesse botão e encerra o processamento.

Obs.

1-A instrução que encerra o processamento de um script, é **wscript.quit**.

2-Tente fazer os exercícios. Se tiver dúvidas, há exercícios resolvidos no final da apostila.

Na figura abaixo vemos um exemplo de um script que executa o programa cmd.exe (abre o ambiente do ms-dos). O argumento 2 o deixa minimizado, e o argumento true significa que a execução do script é interrompida até que o programa que está sendo executado seja encerrado. Digite e teste o script abaixo.





Utilize o script que você testou acima para executar outro programa qualquer de sua preferência.

### Exercício3:

Digitar e testar o script abaixo, que executa o Access e abre o arquivo "revenda.accdb" de seu projeto das aulas anteriores.

Obs.: Para executar o Access e abrir um arquivo ".accdb", use as instruções abaixo. O argumento **"5"** no final da instrução **"varShell.run"** significa que o programa será exibido em janela normal e o argumento **"true"** significa que, as **instruções do script** que vierem após a instrução que executou o Access, serão processadas somente após o Access ser encerrado. Se quisermos que o programa a ser executado fique maximizado, o argumento é o número **"3"**.

```
option explicit
dim varShell 'Cria variável
Set varShell =wscript.CreateObject("WScript.Shell") 'Cria uma instância de objeto
varShell.run ("""msaccess.exe""C:\Users\Hamilton\Desktop\revenda.accdb"),5,true
```

Obs.: **Os nomes das pastas não podem conter espaço. Por exemplo, se o caminho for d:\meus documentos\... utilizar: d:\meusdo~1\... O nome abreviado da pasta com o "~" deve ter 8 caracteres, incluindo o "~".**

**Complemento:** Mude o argumento para **"3"** (p/ maximizar o Access na execução) e inclua algumas instruções após a chamada do Access. Verifique que com o argumento **"true"** essas instruções somente serão executadas quando encerrarmos o processamento do Access. Após esse teste, mude o argumento para **"false"** e verifique o resultado.

Para maiores informações sobre **".run"**, consulte o site:  
[http://msdn.microsoft.com/en-us/library/d5fk67ky\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/d5fk67ky(v=vs.85).aspx)

### Exercício 4:

Criar o arquivo de script de nome backUpBd que faça uma cópia backup de um banco de dados. Esse procedimento costuma ser usado e pode ser muito útil para preservar suas informações em uma cópia anterior do banco de dados, antes que esse arquivo sofra alterações.

Se houver algum problema com o banco de dados a ser atualizado, como por exemplo ficar corrompido, você tem uma cópia dele antes da atualização.

Como exemplo, o código abaixo usa o arquivo **"testebd.accdb"** que se encontra no diretório raiz do drive **"d"** e faz uma cópia no diretório **"c:\backup"** com nome **"anterior.accdb"**:

```
option explicit
dim objFSO
Set objFSO = CreateObject("Scripting.FileSystemObject")
objFSO.CopyFile "d:\testebd.accdb" , "c:\backup\anterior.accdb"
```

Adapte o script acima para o seu ambiente e teste-o.

**Obs.:**

**1-Note que o script acima não cria novas pastas, apenas copia o arquivo em pastas existentes. Para criar pastas, faremos outro exercício.**

**2-Se o arquivo a gravar já existir, este será sobrescrito.**

### **Exercício 5:**

Criar um script que irá automatizar algumas tarefas para acionar o banco de dados "revenda.accdb" de seu projeto das aulas anteriores. Esse script deverá fazer o seguinte:

a) Ao iniciar, deve exibir uma caixa de mensagem com dois botões: Ok e Cancela. Nessa caixa, além de dar boas vindas ao usuário, deverá informar que o sistema abre o banco de dados Veículos e que faz backup do arquivo, automaticamente.

b) Deverá informar ao usuário que caso ele vá fazer atualizações no banco de dados, deverá clicar no botão "Ok", que o backup será feito automaticamente.

c) Se o usuário quer simplesmente sair, deverá clicar em "**Cancela**" **que nada será feito..**

e) Após o click no botão "**Ok**", o script deve fazer backup do arquivo e executar o Access com o arquivo revenda.accdb, no modo maximizado.

g) Em seguida, o script deve exibir mensagem do sucesso ou não da operação.

h) Gravar o arquivo de script no modo "**Somente Leitura**", criar um atalho para sua execução e colocar nesse atalho, um ícone representativo das operações das tarefas a realizar.

Obs.: Este exercício é junção dos exercícios 3 e 4, devendo ficar mais ou menos como na página a seguir:



***option explicit***

***dim objFSO***  
***dim varShell      'Cria variáveis***  
***dim tecla***

***tecla=msgbox ("Sr. Usuário: Bem vindo ao nosso sistema de veículos!" &chr(13)&chr(13)  
&"Se quiser sair, pressione Cancela." & chr(13) & chr(13) & "Caso queira continuar,  
pressione OK que o BackUp será criado automaticamente",1,"A T E N Ç Ã O")***

***if tecla=2 then wscript.quit***

***Set varShell =wscript.CreateObject("WScript.Shell") 'Cria uma instância de objeto***  
***Set objFSO = CreateObject("Scripting.FileSystemObject")***

***objFSO.CopyFile "e:\testebd.accdb" , "C:\Users\User\Desktop\anterior.accdb"***

***msgbox "Backup criado... Irá abrir Banco de Dados para Consulta / atualização!..."***

***varShell.run ("""msaccess.exe""C:\Users\user\Desktop\revenda.accdb"),5,true***  
***msgbox "Operação realizada com sucesso..."***

**Para complementar este exercício:**

- a) Transforme o arquivo script em somente leitura
- b) Crie um atalho para executá-lo, com ícone significativo.

## PROJETO FINAL DA DISCIPLINA

***O projeto constitui-se na confecção de um sistema em Access para controlar as notas e faltas dos alunos da disciplina "Informática II".***

O sistema deve ser encabeçado por um formulário principal que inicia maximizado, contendo três botões: Alunos, Relatórios e Fim. Ocorrendo o click no botão Fim, o programa é encerrado.

Se o usuário der o click no botão Alunos o sistema mostra um formulário com o número do matrícula do aluno, seu nome, sua nota P1 e a quantidade de faltas.

Se o usuário der o click no botão Relatórios, o sistema apresenta um novo formulário com os botões Ordem Alfabética e Ordem de Nota.

Ao ocorrer o click em um dos botões, é apresentado o relatório correspondente contendo o número de matrícula do aluno, o nome e a nota da prova P1.

Configurar o Access para que o usuário não tenha acesso às tabelas, programação, etc.

Este projeto tem duração aproximada de três aulas e vale de 0 a 30 pontos.  
Deverá ser apresentado e entregue ao professor em cd/dvd, em data a combinar

### **Importante:**

A execução do programa inicial do projeto deve ser feita através de um script que automatize tarefas importantes para a preservação do sistema, para auxílio ao usuário ou outras que você julgue importantes ou interessantes.

Importar através do Access o arquivo em Excel fornecido pelo professor "AES INF II INTEGRAL.xls" e trabalhar esse arquivo

***Obs.: A importação pode ser feita abrindo o Access e criando um novo banco de dados (Banco de Dados em Branco). Em seguida, selecione a guia "Dados Externos" e dê um click no ícone Excel. Localize a planilha que quer importar e dê um click em OK. Siga o assistente para a importação e após importado, configure a estrutura do banco de dados a seu gosto.***

Comece a fazer o sistema agora mesmo! Dúvidas poderão ser esclarecidas na próxima aula ou por email.

A apostila também pode ser encontrada no site:

<http://sites.google.com/site/hamiltonmv>

### **Bibliografia**

MORGADO, Flávio. Programando em Microsoft Access com VBA, Rio de Janeiro, Ciência Moderna, 2008.

No desenvolvimento dos scripts da aula "AUTOMATIZAÇÃO DE TAREFAS", foram consultados os seguintes endereços web:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/wsconWSHBasics.asp>

[http://msdn.microsoft.com/en-us/library/d5fk67ky\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/d5fk67ky(v=vs.85).aspx)  
[http://www.macoratti.net/wsh\\_1.htm](http://www.macoratti.net/wsh_1.htm)