



Exemplo: MsgBox Application.Caption (Exibe o título do aplicativo)

Application.Caption = "CADASTRO DE VENDAS" (Altera o título do aplicativo)

- **ActiveWork**: Exibe o nome da pasta de trabalho ativa.

Exemplo: MsgBox Application.ActiveWorkbook.Name

(Exibe o nome do arquivo da pasta de trabalho)

- **ActiveSheet**: Exibe o nome da pasta de trabalho ativa.

Exemplo: MsgBox Application.ActiveSheet.Name (Exibe o nome da planilha)

- **ActiveCell**: Exibe o nome do endereço da célula ativa.

Exemplo: MsgBox Application.ActiveCell.Address (Exibe o nome da planilha)

✓ Métodos do Objeto Application

Os métodos do objeto Application executam funções específicas a esse objeto.

- **Quit**: Esse método fechará completamente o aplicativo Excel.

Exemplo: Application.Quit

(Fecha definitivamente o aplicativo Excel, você ainda será solicitado de salvar os arquivos não salvos)

- **Undo**: Esse método desfaz a última ação no aplicativo Excel.

Exemplo: Application.Undo (É o mesmo que o comando Editar / Desfazer...)

9.3. Objeto Workbook

O Objeto Workbook representa uma pasta de trabalho integral carregada do Excel. A pasta de trabalho padrão é a Pasta 1 (Book1). Ele está um nível abaixo do objeto Application.

Esse objeto forma uma parte do conjunto **Workbooks**, que representa todas as pastas de trabalho carregadas atualmente no Excel. Você não pode referenciar o objeto Workbook diretamente: terá que acessá-lo por meio do conjunto Workbooks.

Conhecendo alguns Métodos, Propriedades e Conjuntos do Objeto Workbook

✓ Propriedades do Objeto Workbook

- **Saved**: Essa propriedade retorna Verdadeiro (caso a pasta de trabalho tenha sido salva) ou Falso (caso a pasta de trabalho não tenha sido salvo).

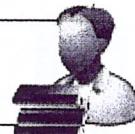




VBA

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno



Exemplo: Workbooks ("book1").Saved

- **HasPassword:** Essa propriedade retorna Verdadeiro (caso a pasta de trabalho esteja protegido por senha) ou Falso (caso a pasta de trabalho não possua senha).

Exemplo: Workbooks ("book1").HasPassword

- **Path:** Essa propriedade retorna o caminho do arquivo corrente.

Exemplo: ThisWorkbook.Path (Retorna o caminho do próprio arquivo corrente)

✓ Métodos do Objeto Workbook

- **Activate:** Esse método ativa uma pasta de trabalho específica caso haja mais de uma pasta de trabalho aberta.

Exemplo: Workbooks ("book1").Activate

- **Close:** Esse método fecha a pasta de trabalho aberta.

Exemplo: Workbooks ("book1").Close

Caso a pasta de trabalho precise ser salva é anexado parâmetros opcionais:

Exemplo: Workbooks ("book1").Close (True, "C:\Teste.xlsx")

- **Print:** Direciona a saída de uma planilha para uma impressora.

Exemplo: Workbooks("Cadastro").PrintOut
(Imprime a planilha Cadastro direto na impressora)

- **PrintPreview:** Direciona a saída de uma planilha para visualização de impressão.

Exemplo: Workbooks("Cadastro").PrintPreview
(Visualiza uma impressão da planilha Cadastro)

- **Save:** Esse método salva qualquer alteração realizada na pasta de trabalho ativa.

Exemplo: Workbooks ("book1").Save

- **SaveAs:** Esse método grava uma pasta de trabalho ativa que ainda não foi salva.

Exemplo: Workbooks ("book1").SaveAs "C:\Teste.xlsx"

- **Open:** Esse método abre um arquivo do Excel já existente.

Exemplo: Workbook.Open "C:\Teste.xlsx"





9.4. Objeto Worksheet

Esse objeto representa a planilha atual que você está trabalhando, essa planilha é membro da coleção Worksheets do objeto Workbook.

Conhecendo alguns Métodos, Propriedades e Conjuntos do Objeto Worksheet

✓ Propriedades do Objeto Worksheet

- **Count:** Essa propriedade retorna o número de planilhas existentes em um arquivo.

Exemplo: MsgBox Worksheets.Count

- **Name:** Essa propriedade retorna o nome de cada planilha.

Exemplo: MsgBox Worksheets(1).Name

✓ Métodos do Objeto Worksheet

- **Add:** Esse método adiciona novas planilhas a um arquivo aberto do Excel.

Exemplo: Worksheets.Add

É possível definir se será inserida uma planilha depois (After) ou antes (Before) de uma planilha existente.

Exemplo: Worksheets.Add before:=Worksheets(1)

- **Delete:** Esse método exclui uma planilha de um arquivo aberto do Excel.

Exemplo: Worksheets(1).Delete (O número corresponde a planilha que será excluída)

9.5. Visão Geral de Eventos

Vimos que durante a execução de algumas sub-rotinas que estavam associadas aos objetos do formulário era apresentados termos como: Click, Change ou Initialize.

Eventos são ações associados aos comportamentos que esses objetos poderão sofrer durante o manuseio.





Exemplo explicativo de um código no VBE exibindo o objeto e o evento associado:

```

Private Sub UserForm_Initialize()
    Dim linha As Integer, lista As String

    'Preenchendo a caixa de combinação CbProduto
    'Seleciona a aba onde contém as informações

```

Abaixo é apresentada uma tabela com os principais eventos do objeto Workbook:

Nome do Evento	Significado
Activate() / Desactivate()	Ativadas ou desativadas a Pasta de Trabalho
SheetChange()	Planilha Ativa
SheetSelectionChange()	Planilha Ativa Selecionada
WindowActivate() / Desactivate()	Janela Ativa ou Inativa
WindowResize()	Janela Redimensionada
Activate() / Desactivate()	Pasta de Trabalho Ativada ou Desativada
BeforeClose()	Pasta de Trabalho Fechada
BeforePrint()	Pasta de Trabalho Impressa
BeforeSave()	Pasta de Trabalho Salva
NewSheet()	Nova Planilha Criada na Pasta de Trabalho
Open()	Nova Pasta de Trabalho Criada
SheetActivate() / Desactivate()	Planilha Ativa e não ativa

Abaixo é apresentada uma tabela com os principais eventos do objeto Worksheet:

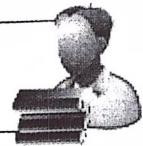
Nome do Evento	Significado
SelectionChange()	Planilha Ativa Selecionada
Activate() / Desactivate()	Janela Ativada ou Desativa
Window Resize	Janela Redimensionada
WorkbookActivate / Desactivate	Pasta de Trabalho Ativada / Desativada
Change()	Planilha Ativa

Nota: Os eventos deverão ser associados as sub-rotinas de acordo com as ações esperadas do objeto.

9.6. Aplicação Prática para o Uso de Objetos

Após essa breve abordagem feita a objetos no VBA vamos aproveitar para tornar nosso sistema de cadastro de vendas mais interessante com algumas funcionalidades que não inserimos a princípio em nosso sistema.





Criando um Encerramento para o Sistema de Registro de Vendas

Em nosso sistema o formulário já está programada para quando for cancelado o cadastro fechar o formulário e retornar para a planilha Plan1(Principal).

Vamos aperfeiçoar um pouco mais o fechamento do formulário para que ao ser cancelado ou fechado pergunte ao usuário se deseja encerrar completamente o sistema, siga os passos abaixo para programar essa nova função no formulário:

1º) Abra o módulo do código do formulário FrmRegVendas;

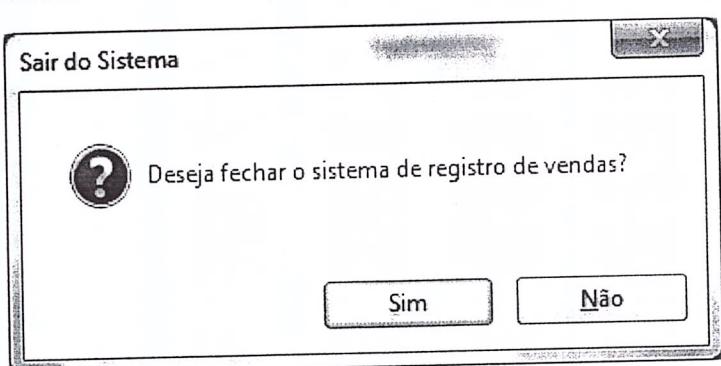
2º) Selecione o objeto Userform e o evento Terminate;

```
Microsoft Visual Basic for Applications - teste_apostila.xlsm - [Plan1 (Código)]  
Arquivo Editar Exibir Inserir Formatar Depurar Executar Ferramentas Suplementos Janela Ajuda Ln 2, Col 5  
(Geral) UserForm_Terminate  
Private Sub UserForm_Terminate()  
End Sub
```

3º) Insira o código abaixo dentro da área de código do módulo:

```
Private Sub UserForm_Terminate()  
    If MsgBox("Deseja Fechar o Sistema de Registro de Vendas?", _  
        vbQuestion + vbYesNo, "Sair do Sistema") = vbYes Then  
        ThisWorkbook.Save 'Salva a Pasta de trabalho  
        Workbooks.Close 'Fecha a Pasta de Trabalho  
    End If  
End Sub
```

4º) Execute o formulário, clique no botão cancelar ou clique no botão "X" da janela e veja se surgiu a mensagem abaixo:

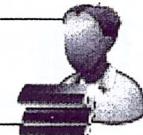


Teste Prático: Se clicado "Sim" será encerrado a pasta de trabalho aberta do nosso sistema de registro de vendas.



**VBA**

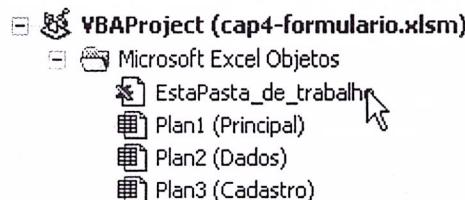
MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Abrindo o Formulário Registro de Vendas ao Abrir a Pasta de Trabalho

Esse código irá programar a execução do formulário Registro de Vendas ao abrir a pasta de trabalho para facilitar o acesso ao cadastro de vendas.

1º) Entre no módulo do código da pasta de trabalho e crie a sub-rotina, a pasta de trabalho é identificada no Project Explorer no ícone indicado abaixo:



2º) Insira o código abaixo no módulo:

```
Private Sub Workbook_Open()  
  
    ' Executa o formulário Registro de Vendas  
    ' ao abrir a pasta de trabalho no Excel  
    FrmRegVendas.Show  
  
End Sub
```

Teste Prático: Feche a pasta de trabalho que contém o formulário e o código VBA, abra a pasta de trabalho, em seguida, você verá o formulário sendo aberto automaticamente pelo VBA.

Fechando a Pasta de Trabalho e Salvando os Dados da Planilha

Esse código irá programar a execução da gravação de dados caso o usuário simplesmente feche a pasta de trabalho aberta.

2º) Insira o código abaixo no módulo da pasta de trabalho:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    ' Salva a Pasta de trabalho  
    ThisWorkbook.Save  
End Sub
```

Teste Prático: Ao fechar a pasta de trabalho todo conteúdo será salvo mesmo que você não execute nenhum comando.





CAPÍTULO 10 – CRIANDO GRÁFICOS

Gerar gráficos a partir do código VBA parece, a princípio, algo desnecessário pois o Excel nos dá todo o suporte à criação de gráficos com extrema facilidade e simplicidade através dos comandos presentes.

Embora isso seja considerado por um usuário comum uma afirmação definitiva e única você aprenderá nesse tópico que criar gráficos através do código VBA é muito enriquecedor pois a quantidade de opções disponíveis poderá superar em larga escala o que o Excel já proporciona.

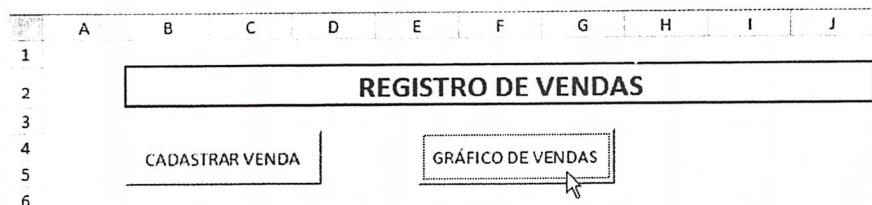
Para iniciarmos nossos estudos com gráficos faremos um cadastro um pouco maior de dados para permitir a criação de gráficos. Siga o modelo do cadastro abaixo como referência:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Registro de Vendas												
2													
3													
4	Código	Data da Venda	Vendedor	Produto	Marca	Modelo	Memória	HD	Monitor	Tipo	Acessórios	Financiado	Valor da Venda
5	1	7/6/2010	ADRIANO	NoteBook	Itautec	W-7655	4Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.904,00
6	2	8/6/2010	MARCELO	NoteBook	Asus	AS-204A	8Gb	500 Gb	15 pol.	LED	NC	36 Vezes	R\$ 3.160,00
7	3	8/6/2010	MARCELO	NetBook	Toshiba	TS-1034	4Gb	250 Gb	14 pol.	LCD	NC	À Vista	R\$ 3.860,00
8	4	8/6/2010	JULIANA	Desktop	Acer	PST-1223B	3Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.500,00
9	5	8/6/2010	JULIANA	NetBook	Toshiba	TS-1045NT	2Gb	120 Gb	14 pol.	LCD	NC	À Vista	R\$ 2.700,41
10	6	8/6/2010	ADRIANO	NetBook	Dell	ASP-1223D	2Gb	160 Gb	14 pol.	LED	NC	À Vista	R\$ 1.540,00
11	7	8/6/2010	ADRIANO	NoteBook	Acer	AC-897S	8Gb	500 Gb	17 pol.	LED	NC	36 Vezes	R\$ 4.360,00
12	8	8/6/2010	MARCELO	Desktop	Asus	AS-1030SP	3Gb	200 Gb	17 pol.	LCD	OK	À Vista	R\$ 3.480,00
13	9	8/6/2010	ADRIANO	NetBook	Positivo	PS-2040T	2Gb	120 Gb	14 pol.	LED	NC	À Vista	R\$ 1.500,00
14	10	8/6/2010	JULIANA	Desktop	Itautec	TI-1020ST	8Gb	250 Gb	17 pol.	LED	OK	24 Vezes	R\$ 2.622,40

Criando uma Planilha para a Origem de Dados do Gráfico

Vamos agora criar um código inicial que prepara uma planilha para a criação do gráfico:

1º) Insira na planilha o botão de comando Gráfico de Vendas:





2º) Insira o código abaixo dentro do módulo do botão Gráfico de Vendas:

```
Private Sub CommandButton2_Click()
    ' Conta o total de planilhas existentes
    Total = Worksheets.Count

    If Total < 4 Then

        ' Adiciona uma nova planilha após a planilha Cadastro
        Worksheets.Add After:=Sheets("Cadastro")

        ' Conta o total de planilhas existentes
        Total = Worksheets.Count

        ' Renomeia a última planilha criada
        Sheets(Total).Name = "Dados para o Gráfico"

    End If

    ' Torna a planilha que contém os dados para o gráfico ativa
    Sheets(Total).Activate

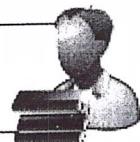
    ' Cria a Planilha de Resumo de Vendas
    With ActiveSheet
        ' Inserindo texto da planilha
        .Cells(1, 1) = "RESUMO DE VENDAS POR VENDEDOR"
        .Cells(1, 1).Font.Bold = True
        .Cells(2, 1) = "Vendedor"
        .Cells(2, 2) = "Quantidade de Vendas"
        .Cells(2, 3) = "Total de Vendas"
        .Cells(3, 1) = "ADRIANO"
        .Cells(4, 1) = "JULIANA"
        .Cells(5, 1) = "MARCELO"

        ' Formatando a planilha
        .Cells(2, 2).EntireColumn.AutoFit
        .Cells(2, 3).EntireColumn.AutoFit
        .Range("A1:C1").Merge
        .Range("A1").HorizontalAlignment = xlCenter
    End With

    ' Contando o total de vendas e soma as vendas realizadas pelos vendedores
    linha = 3
    dados = 5
    contagem = 0
    Total = 0
```

(Continua na próxima página)





3º) Continuação do código anterior:

```
Do While ActiveSheet.Cells(linha, 1) <> ""
    busca = ActiveSheet.Cells(linha, 1)

    Do While Sheets("Cadastro").Cells(dados, 3) <> ""
        procurado = Sheets("Cadastro").Cells(dados, 3)
        valor = CSng(Sheets("Cadastro").Cells(dados, 13))
        If busca Like procurado Then
            Total = Total + valor
            contagem = contagem + 1
        End If
        dados = dados + 1
    Loop
    ActiveSheet.Cells(linha, 2) = contagem
    ActiveSheet.Cells(linha, 3) = CCur(Total)
    dados = 5
    contagem = 0
    Total = 0
    linha = linha + 1
Loop

ActiveSheet.Range("B3:C5").HorizontalAlignment = xlCenter
End Sub
```

4º) Ao clicar no botão Gráfico de Vendas na planilha Principal será criada a planilha abaixo:

	A	B	C
RESUMO DE VENDAS POR VENDEDOR			
2	Vendedor	Quantidade de Vendas	Total de Vendas
3	ADRIANO	4	R\$ 9.304,00
4	JULIANA	3	R\$ 6.822,81
5	MARCELO	3	R\$ 10.500,00

Nota: Essa planilha é necessário pois os dados foram resumidos afim de criar um gráfico coerente, essa planilha será a nossa **origem de dados** para o gráfico.

Criando um Gráfico Simples

Para criar um simples gráfico faça o seguinte:

1º) Acesse o módulo de código do botão **CommandButton2_Click()**:

2º) Insira no final do código do módulo as linhas de código abaixo:

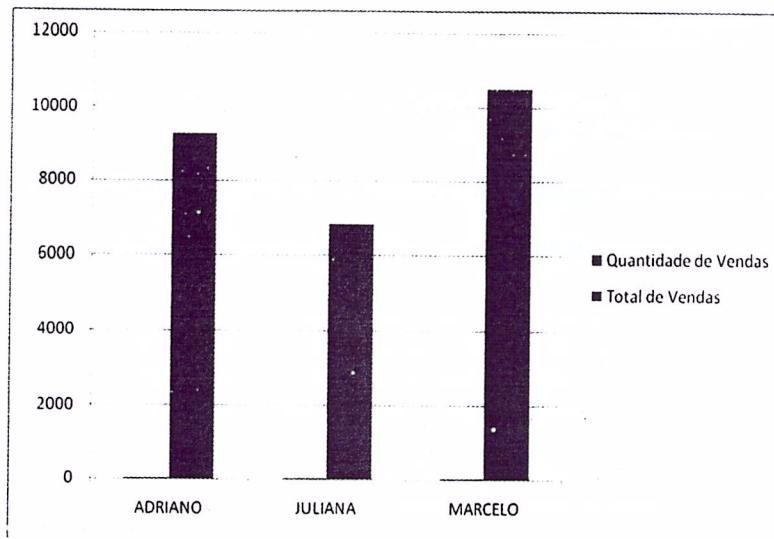
```
' Seleção de dados para o gráfico
ActiveSheet.Range("A2").Select
ActiveSheet.Range(Selection, Selection.End(xlToRight)).Select
ActiveSheet.Range(Selection, Selection.End(xlDown)).Select

'Cria um gráfico em uma nova aba
Charts.Add
```





3º) Clique no botão Gráfico de Vendas e observe que a nova aba Gráf1 será exibida e com um gráfico de colunas:



No código que inserimos para criar um gráfico, observe que foi feita a seleção da range de células de A2 até C5 para depois inserir o gráfico.

Podemos entender que a primeira coluna da planilha é tratada como **Eixo das categorias** e as demais colunas são tratadas como **Eixos dos Valores**; incluímos para os eixos de valores mais de um tipo de valor para a construção das colunas do gráficos.

Fazendo uma pequena modificação no código que inserimos definimos para eixo da categoria a **coluna A** (A2:A5) e para o eixo de valores a **coluna C** (C2:C5), está é uma situação comum pois muitas das vezes os dados não se encontram contínuos.

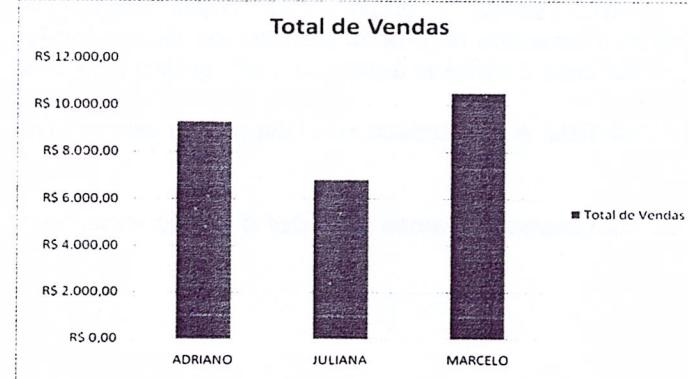
1º) Altere o código de criação do gráfico conforme abaixo:

```
' Seleção de dados para o gráfico
ActiveSheet.Range ("A2 : A5,C2 : C5").Select

'Cria um gráfico em uma nova aba
Charts.Add
```

2º) Apague a última aba de gráfico criado, caso exista e crie novamente um gráfico:

Resultado após o gráfico ser gerado:



Nota: Observe que nesse caso o gráfico vem com informações específicas, embora seja parecido com o gráfico anterior.



Criando um Gráfico mais Detalhado

Repare que criar um gráfico de imediato é simples, basta termos feito a seleção de dados e executarmos o método **Charts.Add** e um gráfico prontamente surgirá.

Criar um gráfico não se resume a esses procedimentos pois possui muitos detalhes envolvidos.

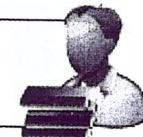
Existem alguns métodos e propriedades que permitem detalhar a criação dos gráficos vamos conhecê-las:

- **Propriedade Name:** Define o nome do gráfico sendo gerado. Trata-se de uma propriedade interna do VBA para facilitar o acesso. Não confunda com a propriedade **CharTitle**. Fique atento a essa propriedade porque não é possível criar dois gráficos com o mesmo nome. Caso o usuário execute uma macro que crie um gráfico cujo nome já está sendo atribuído a outro gráfico existente, o VBE gerará um erro.
- **Propriedade CharType:** Essa propriedade define o tipo de gráfico a ser gerado. Os principais tipos de gráficos estão sendo apresentados na tabela abaixo:

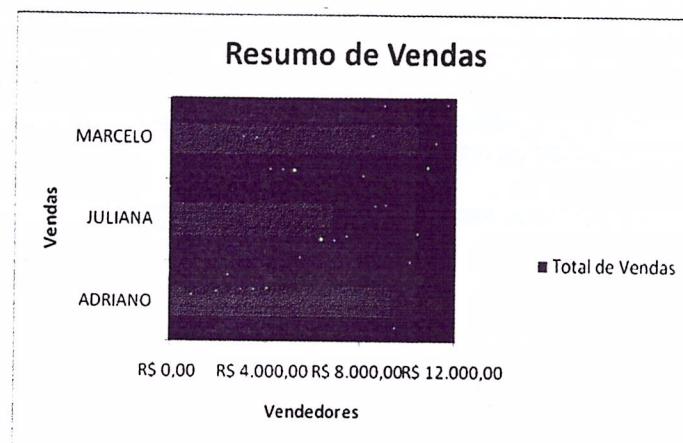
Tipo	Constante
Área	xlArea
Linha	xlLine
Barra Horizontal	xlBarClustered
Barra Vertical	xlColumnClustered
Torta (pizza)	xlPie
Bolha	xlBubble
Pirâmide	xlPiramideCol
Coluna 3D	xl3dColumn
Radar	xlRadar
Cone	xlConeCol
Superfície	xlSurface
Cilindro	xlCylinderCol
Pontos	xlCYScatter
Anéis	xlDoughnut

- **Método SetSourceData:** Indica a origem dos dados do gráfico, é possível nesse método indicar se os dados serão desenhados em **colunas** (xlColumns) ou **linhas** (xlRows);
- **Propriedade Location:** Indica aonde o gráfico será criado poderá ter seu valor definido como **xlLocationObject** (um gráfico embutido na própria planilha) ou **xlLocationAsNewSheet** (um gráfico criado em uma nova planilha), nesse caso é indicado a planilha que o gráfico será criado;
- **Propriedade HasTitle e CharTitle:** A propriedade **HasTitle** recebe valores "True" ou "False" para indicar se o gráfico terá ou não título;

Para ficar um pouco mais comprehensível vamos modificar o código inicial para criar um gráfico utilizando os métodos e propriedades citadas:



4º) Crie um novo gráfico e observe que dessa vez o gráfico surgiu com uma aparência completamente diferente:



Excluindo o Gráfico Criado

Observe que sempre que você clicar no botão Gráfico de Vendas será criado um novo gráfico, porém o gráfico antigo não é prontamente excluído gerando uma série de gráficos desnecessários.

Insira o código abaixo antes do código de criação para realizar a exclusão do gráfico já criado anteriormente:

1º) Clique no local que será inserido o código:

```
' Torna a planilha que contém os dados para o gráfico ativa
Sheets("Total").Activate

|
'Cria a Planilha de Resumo de Vendas
```

2º) Insira o código abaixo:

```
' Exclusão do Gráfico Anterior
Sheets("Dados para o Gráfico").Select

Do Until ActiveSheet.ChartObjects.Count = 0
    ActiveSheet.ChartObjects(1).Activate
    ActiveSheet.ChartObjects(1).Delete
Loop
```

3º) Teste o código criando um gráfico, observe que agora o gráfico anterior será prontamente apagado.

Nota: Esse código só funcionará caso já exista um gráfico anterior, se não houver nenhum gráfico, nada ocorrerá.





CAPÍTULO 11 – IMPRESSÃO DE DADOS

Geralmente quando se cria uma planilha o usuário do Excel tem necessidade de realizar algum tipo de impressão, isso pode ser feito de maneira simples no Excel através de configurações adequadas.

Aproveitando as facilidades que o VBA nos proporciona iremos aprender, nesse tópico iremos aprender como configurar um documento para impressão de maneira simples e rápida.

Para executar os códigos ensinados será utilizado a área de código do botão Gerar Relatório.

A área de código do módulo do botão é exibida abaixo:

```
Private Sub CommandButton3_Click()
|
End Sub
```

Demarcando a Área de Impressão

O primeiro procedimento a ser adotado em uma impressão será demarcar a área de impressão pois você poderá não querer imprimir todo o conteúdo de uma planilha.

Siga a sequência de procedimentos para criar a área de impressão:

1º) Abaixo criaremos o código que cria a área de impressão:

```
Private Sub CommandButton3_Click()
    'Seleciona a planilha Cadastro
    Sheets("Cadastro").Select

    'Seleciona a Célula Inicial da Área de Impressão
    ActiveSheet.Range("A4").Select

    'Desloca para a Última Linha da Planilha
    Selection.End(xlDown).Select

    'Obtém a Posição da Última Linha
    fim = ActiveCell.Row

    'Gera a Área de Impressão
    ActiveSheet.PageSetup.PrintArea = "A4:M" & fim
End Sub
```

2º) Execute o código clicando no botão Gerar Relatório;

3º) Após o código executado teremos a área de impressão demarcada com um pontilhado:

A	B	C	D	E	F	G	H	I	J	K	L	M	
Registro de Vendas													
4	Código	Data da Venda	Vendedor	Produto	Marca	Modelo	Memória	HD	Monitor	Tipo	Acessórios	Financiado	Valor da Venda
5	1	7/6/2010	ADRIANO	NoteBook	Itautec	W-7655	4Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	RS 1.904,00
6	2	8/6/2010	MARCELO	NoteBook	Asus	AS-204A	8Gb	500 Gb	15 pol.	LED	NC	36 Vezes	RS 3.160,00
7	3	8/6/2010	MARCELO	NetBook	Toshiba	TS-1034	4Gb	250 Gb	14 pol.	LCD	NC	Á Vista	RS 3.860,00
8	4	8/6/2010	JULIANA	Desktop	Acer	PST-1223B	3Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	RS 1.500,00
9	5	8/6/2010	JULIANA	NetBook	Toshiba	TS-1045NT	2Gb	120 Gb	14 pol.	LCD	NC	Á Vista	RS 2.700,41
10	6	8/6/2010	ADRIANO	NetBook	Dell	ASP-1223D	2Gb	160 Gb	14 pol.	LED	NC	Á Vista	RS 1.540,00
11	7	8/6/2010	ADRIANO	NoteBook	Acer	AC-897S	8Gb	500 Gb	17 pol.	LED	NC	36 Vezes	RS 4.360,00
12	8	8/6/2010	MARCELO	Desktop	Asus	AS-1030SP	3Gb	200 Gb	17 pol.	LCD	OK	Á Vista	RS 3.480,00
13	9	8/6/2010	ADRIANO	NetBook	Positivo	PS-2040T	2Gb	120 Gb	14 pol.	LED	NC	Á Vista	RS 1.500,00
14	10	8/6/2010	JULIANA	Desktop	Itautec	TI-1020ST	8Gb	250 Gb	17 pol.	LED	OK	24 Vezes	RS 2.622,40

Nota: A propriedade PageSetup.PrintPreview é recomendado para definir o conteúdo para ser impresso.





1º) Modifique o código que cria um gráfico conforme abaixo:

```
'Cria um novo gráfico
Charts.Add

'Nomeia o Gráfico
ActiveChart.Name = "Gráfico de Vendas"

'Define o Tipo de Gráfico
ActiveChart.ChartType = xlBarClustered

'Indica a Origem de Dados para o Gráfico
ActiveChart.SetSourceData Source:=Sheets("Dados para o Gráfico").Range(_
"A2:A5,C2:C5"), PlotBy:=xlColumns

'Coloca o Gráfico Embutido na Própria Planilha
'aonde se encontra os dados do gráfico
ActiveChart.Location where:=xlLocationAsObject, Name:="Dados para o Gráfico"

'Definindo as Propriedades do Gráfico
With ActiveChart
    'Ativando o título principal do gráfico
    .HasTitle = True

    'Inserindo o Título Principal do Gráfico
    .ChartTitle.Characters.Text = "Resumo de Vendas"

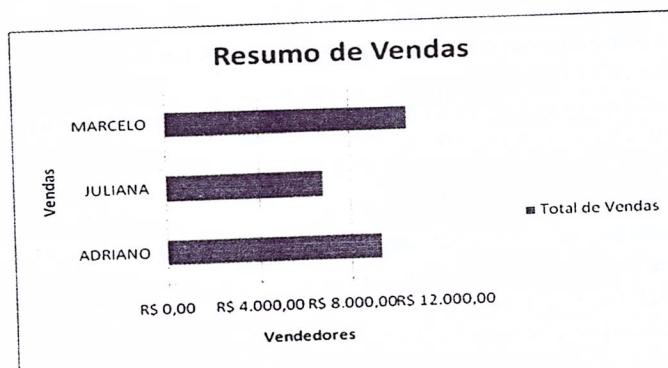
    'Ativa o Título do Eixo das Categorias (eixo X)
    .Axes(xlCategory, xlPrimary).HasTitle = True

    'Inserindo o Título do Eixo das Categorias (eixo X)
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Vendas"

    'Ativa o Título do Eixo dos Valores (eixo Y)
    .Axes(xlValue, xlPrimary).HasTitle = True

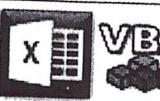
    'Inserindo o Título do Eixo dos Valores (eixo Y)
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Vendedores"
End With
```

2º) Crie o gráfico e veja se você obteve o resultado abaixo:



Nota: Embora seja mais trabalhoso o resultado compensa devido as configurações detalhadas que ajustamos no gráfico.



**VBA**

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Modificando a Aparência do Gráfico

Os gráficos, assim como as planilhas, permitem diversas formatações visuais. Vamos inserir o código abaixo após o código de criação do gráfico para alterar o visual do nosso gráfico:

1º) Clique ao final do código que altera as propriedades do gráfico:

```
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Vendedores"  
End With  
  
|  
End Sub
```

2º) Insira o código abaixo:

```
'Modificando a Aparência do Gráfico  
  
'Seleciona o Gráfico 1 da Aba  
ActiveSheet.ChartObjects(1).Activate  
  
'Seleciona a Área dos Eixos do Gráfico  
ActiveChart.PlotArea.Select  
  
'Altera a Borda Interna  
With Selection.Border  
    .ColorIndex = 55  
    .Weight = xlThin  
    .LineStyle = xlContinuous  
End With  
  
'Altera a Cor de Preenchimento do Gráfico  
With Selection.Interior  
    .ColorIndex = 55  
    .PatternColorIndex = 1  
    .Pattern = xlSolid  
End With  
  
'Aplica um Preenchimento Gradiente no Gráfico  
With Selection.Fill  
    .OneColorGradient Style:=msoGradientDiagonalDown, Variant:=1, _  
        Degree:=0.231337254901961  
End With  
  
'Aplica um Padrão de Preenchimento com Brilho no Gráfico  
With Selection  
    .Fill.Visible = True  
    .Fill.ForeColor.SchemeColor = 5  
End With
```



ADV – Curso de Informática

Av. Treze de Maio, 23 – 8º andar – Centro – Rio de Janeiro - RJ

<http://www.cursoadv.com.br/>

Telefone: (21) 2210-1180

Página 120



Controlando a Saída de Impressão

Após termos demarcado a área de impressão poderemos imprimir o conteúdo da nossa planilha, pode ser escolhido entre dois métodos de impressão: o método **PrintOut** que realiza uma saída direta para a impressora e o método **PrintPreview** que visualiza a impressão antes de imprimir.

Vamos alterar nosso código anterior para nos questionar como será a nossa saída de impressão:

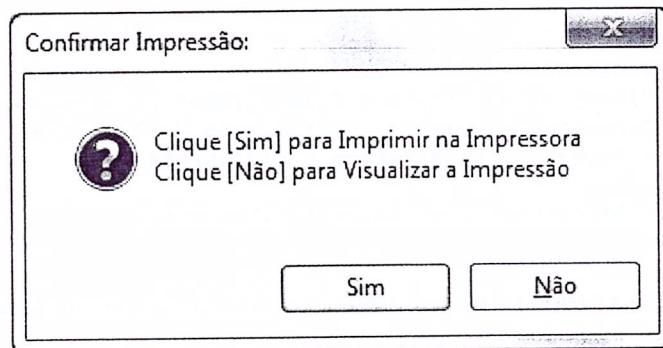
1º) Clique ao final do código que digitamos:

```
'Obtém a Posição da Última Linha  
fim = ActiveCell.Row  
  
'Gera a Área de Impressão  
ActiveSheet.PageSetup.PrintArea = "A4:M" & fim  
|  
End Sub
```

2º) Insira o código abaixo:

```
'Verificando a Saída de Impressão  
If MsgBox("Clique [Sim] para Imprimir na Impressora" & Chr(13) & _  
"Clique [Não] para Visualizar a Impressão", _  
vbQuestion + vbYesNo, "Confirmar Impressão:") = vbYes Then  
ActiveWindow.SelectedSheets.PrintOut  
Else  
    ActiveWindow.SelectedSheets.PrintPreview  
End If
```

3º) Execute o código, a caixa de diálogo abaixo será exibida questionando ao usuário se deseja imprimir ou visualizar impressão:



4º) Clicando **Sim** será feita uma cópia da planilha na impressora e clicado **Não** será apenas visualizada a impressão da planilha.



**VBA**

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Nota: O método PrintOut possui diversos parâmetros associados em sua sintaxe:

PrintOut(From,To,Copies, Preview, ActivePrinter, PrintFile, Collate, PrToFileName)

Descrição dos Parâmetros:**From:** página inicial;**To:** página final;**Copies:** quantidade de cópias;**Preview:** "True" para visualizar a impressão;**ActivePrinter:** nome da impressora. "True" indica a impressora "default";**PrintToFile:** "True" para imprimir criando um arquivo;**PrintFileName:** nome do arquivo a ser criado se PrintToFile for "True";**Collate:** "True" para agrupar cópias; e "False" para imprimir seqüência de páginas;**Ajustando a Orientação de Página para a Impressão**

Ao imprimir um documento você poderá definir se o documento será impresso no formato **Retrato** (folha em pé) ou no formato **Paisagem** (folha deitada). Existe no controle de impressão a propriedade **Orientation** que ao ser inserido os valores **xlPortrait** e **xlLandscape** ajustará a orientação do documento impresso.

Modifique o código para corresponder a esse tipo de ajuste conforme abaixo:

1º) Clique na parte do código onde iremos inserir as modificações:

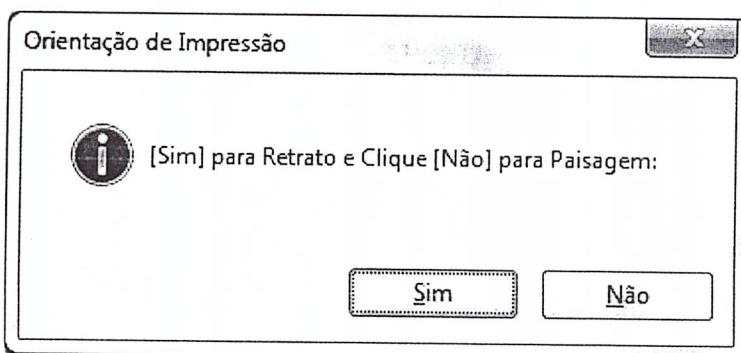
```
' Gera a Área de Impressão
ActiveSheet.PageSetup.PrintArea = "A4:M" & fim

|
' Verificando a Saída de Impressão
```

2º) Insira o código abaixo:

```
' Alterando a Orientação de Impressão
If MsgBox("[Sim] para Retrato e Clique [Não] para Paisagem:",_
vbInformation + vbYesNo, "Orientação de Impressão:") = vbYes Then
    ActiveSheet.PageSetup.Orientation = xlPortrait
Else
    ActiveSheet.PageSetup.Orientation = xlLandscape
End If
```

3º) Ao executar o código o usuário será questionado sobre a orientação de página a escolher:





Centralizando o Alinhamento de Impressão

Quando se executa a impressão é possível ajustar o alinhamento da impressão posicionando o conteúdo impresso, esse alinhamento é realizado através das propriedades **PageSetup.CenterHorizontally** (alinhamento na horizontal) e **PageSetup.CenterVertically** (alinhamento na vertical). Essas propriedades poderão ser ativadas ou desativadas através dos valores “True” ou “False”.

Realize as modificações indicadas para centralizarmos o conteúdo impresso:

1º) Clique na parte do código onde iremos inserir as modificações:

```
ActiveSheet.PageSetup.Orientation = xlLandscape
End If

|
'Verificando a Saída de Impressão
If MsgBox("Clique [Sim] para Imprimir na Impressora" & Chr(13) & _
```

2º) Insira o código abaixo:

```
'Centralizando a Impressão na Horizontal e na Vertical
ActiveSheet.PageSetup.CenterHorizontally = True
ActiveSheet.PageSetup.CenterVertically = True
```

3º) Vá executando os comandos de configuração de impressão e tente visualizar a impressão abaixo:

Código	Data da Venda	Vendedor	Produto	Marca	Modelo	Memória	HD	Monitor	Tipo	Acessórios	Financiado	Valor de Venda
1	8/6/2013 ADRIANO	Adriano	Notebook	Asus	W519S	4Gb	320 Gb 35 pol.	LCD	OK	12 Vezes	RS 1.950,00	
2	8/6/2013 MARCELO	Notebook	Asus	ASUS	ASUS	8Gb	800 Gb 15 pol.	LCD	NC	16 vezes	RS 3.150,00	
3	8/6/2013 MARCELO	Notebook	Toshiba	T510M	4Gb	250 Gb 14 pol.	LCD	NC	À Vista	RS 3.550,00		
4	8/6/2013 JULIANA	Desktop	Acer	PST2123B	3Gb	320 Gb 15 pol.	LCD	OK	12 Vezes	RS 1.500,00		
5	8/6/2013 JULIANA	NetBook	Toshiba	T5-1045MF	2Gb	120 Gb 14 pol.	LCD	NC	À Vista	RS 2.700,41		
6	8/6/2013 ADRIANO	NetBook	Dell	ASUS	1212J3D	2Gb	160 Gb 14 pol.	LCD	NC	À Vista	RS 1.540,00	
7	8/6/2013 ADRIANO	NetBook	Acer	AC-9710	1Gb	500 Gb 17 pol.	LCD	NC	36 Vezes	RS 4.360,00		
8	8/6/2013 MARCELO	Desktop	Aus	ASUS	ASUS	3Gb	200 Gb 17 pol.	LCD	OK	À Vista	RS 3.480,00	
9	8/6/2013 ADRIANO	NetBook	Festivo	P5-1242T	2Gb	120 Gb 14 pol.	LCD	NC	À Vista	RS 1.480,00		
10	8/6/2013 JULIANA	Desktop	Kawak	T-13225T	8Gb	250 Gb 17 pol.	LCD	OK	24 Vezes	RS 2.622,40		





Configurando Cabeçalho e Rodapé na Impressão

Até o momento já conseguimos imprimir um documento perfeitamente, podemos configurar outras informações como colocar cabeçalho e rodapé na página.

Existem propriedades destinadas à inserção do cabeçalho, essas propriedades também estão associadas ao alinhamento do cabeçalho:

`PageSetup.LeftHeader` (Cabeçalho a Esquerda)

`PageSetup.CenterHeader` (Cabeçalho Centralizado)

`PageSetup.RightHeader` (Cabeçalho a Direita)

Exemplo de um cabeçalho: `ActiveSheet.PageSetup.CenterHeader = "Topo da Página"`

Também existem propriedades destinadas à inserção do rodapé:

`PageSetup.LeftFooter` (Rodapé a Esquerda)

`PageSetup.CenterFooter` (Rodapé Centralizado)

`PageSetup.RightFooter` (Rodapé a Direita)

Exemplo de um rodapé: `ActiveSheet.PageSetup.CenterFooter = "Base da Página"`

Caso precise imprimir uma informação gerada sobre a impressão, como por exemplo: número de página, data atual ou hora atual, terá que utilizar um dos códigos especiais mostrados na tabela abaixo:

Código	Descrição
&L	Alinha os caracteres à esquerda.
&C	Centraliza os caracteres.
&R	Alinha os caracteres à direita.
&E	Ativa ou desativa a impressão com sublinhado duplo.
&X	Ativa ou desativa a impressão com sobreescrito.
&Y	Ativa ou desativa a impressão com subscrito.
&B	Ativa ou desativa a impressão com negrito.
&I	Ativa ou desativa a impressão com itálico.
&U	Ativa ou desativa a impressão com tachado.
&S	Ativa ou desativa a impressão com tachado.
&D	Imprime a data atual.
&T	Imprime a hora atual.
&F	Imprime o nome do documento.
&A	Imprime o nome da guia da pasta de trabalho.
&P	Imprime o número da página.
&P+Número	Imprime o número da página mais o número especificado.
&P-Número	Imprime o número da página menos o número especificado.
&&	Imprime um único um 'E' comercial.
&"fonte"	Imprime os caracteres na fonte específica. Certifique-se de incluir aspas duplas.
&nn	Imprime os caracteres no tamanho de fonte especificado. Use um número de dois dígitos para especificar um tamanho em pontos.
&N	Imprime o número total de páginas no documento.





Faremos agora alterações em nosso código para configurar o Cabeçalho e o Rodapé na página:

1º) Clique no local do código aonde iremos inserir as alterações:

```
'Centralizando a Impressão na Horizontal e na Vertical  
ActiveSheet.PageSetup.CenterHorizontally = True  
ActiveSheet.PageSetup.CenterVertically = True  
  
'Verificando a Saída de Impressão  
If MsgBox("Clique [Sim] para Imprimir na Impressora" & Chr(13) & _
```

2º) Insira o código abaixo:

```
'Inserir Cabeçalho e Rodapé na Página  
With ActiveSheet.PageSetup  
    'Inserir um Cabeçalho Centralizado, Fonte Arial, Negrito e na Cor Vermelha  
.CenterHeader = "é""Arial,Negrito""éFF0000 RELATÓRIO DE VENDAS"  
  
'Inserir a Data Atual a Direita no Cabeçalho  
.RightHeader = "éD"  
  
'Inserir a Numeração da Página Atual e o N° Total de Páginas  
.RightFooter = "Página éP de éN"  
End With
```

3º) Abaixo página resultante após a execução do código:

RELATÓRIO DE VENDAS													10/6/2010
Código	Data da Venda	Vendedor	Produto	Marca	Modelo	Memória	HD	Monitor	Tipo	Accessórios	Financiado	Valor da Venda	
1	7/6/2010	ADRIANO	NetBook	raute	W-7655	4Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.904,00	
2	8/6/2010	MARCELO	NetBook	Asus	AS-204A	8Gb	500 Gb	15 pol.	LED	NC	36 Vezes	R\$ 3.160,00	
3	8/6/2010	MARCELO	NetBook	Toshiba	TS-1034	4Gb	250 Gb	14 pol.	LCD	NC	Á Vista	R\$ 3.860,00	
4	8/6/2010	JULIANA	Desktop	Acer	PST-1223B	3Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.500,00	
5	8/6/2010	JULIANA	NetBook	Toshiba	TS-1045NT	2Gb	120 Gb	14 pol.	LCD	NC	Á Vista	R\$ 2.700,41	
6	8/6/2010	ADRIANO	NetBook	Dell	ASP-1223D	2Gb	160 Gb	14 pol.	LED	NC	Á Vista	R\$ 1.540,00	
7	8/6/2010	ADRIANO	NetBook	Acer	AC-8975	8Gb	500 Gb	17 pol.	LED	NC	36 Vezes	R\$ 4.360,00	
8	8/6/2010	MARCELO	Desktop	Asus	AS-1030SP	3Gb	200 Gb	17 pol.	LCD	OK	Á Vista	R\$ 3.480,00	
9	8/6/2010	ADRIANO	NetBook	Positivo	PS-2040T	2Gb	120 Gb	14 pol.	LED	NC	Á Vista	R\$ 1.500,00	
10	8/6/2010	JULIANA	Desktop	raute	Ti-1020ST	8Gb	250 Gb	17 pol.	LED	OK	24 Vezes	R\$ 2.622,40	

Página 1 de 1



**VBA**

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Ajustando o Tamanho da Área de Impressão

Nosso código já consegue gerar um documento impresso sem problemas, porém alguns detalhes não forma observados.

Observe que se a planilha estiver ajustada com a orientação em retrato (Portrait) será percebido que nem todas as colunas são impressas, existe a possibilidade de ajustar esse tamanho através da propriedade **PageSetup.Zoom** submetida a um determinado valor amplia o reduz a área de impressão do documento.

Vamos configurar nosso código de impressão para quando a área de impressão for ajustada para retrato (Portrait) caibam todas as colunas da planilha.

Realize as alterações indicados abaixo:

1º) Clique na parte do código aonde iremos realizar as alterações:

```
' Alterando a Orientação de Impressão
If MsgBox("[Sim] para Retrato e Clique [Não] para Paisagem:",_
vbInformation + vbYesNo, "Orientação de Impressão:") = vbYes Then
    ActiveSheet.PageSetup.Orientation = xlPortrait
|
Else
    ActiveSheet.PageSetup.Orientation = xlLandscape
End If
```

2º) Insira o código abaixo:

```
' Ajusta a Área de Impressão para 70%
ActiveSheet.PageSetup.Zoom = 70
```

3º) Teste criar uma página impressa no modo Retrato e veja se todas as colunas estão visíveis, caso contrário altere o valor até obter o ajuste ideal.

Nota: Faça o ajuste de zoom com cautela, se sua planilha têm muitas colunas não será interessante reduzir muito o tamanho do zoom pois o resultado poderá não ser satisfatório.





CAPÍTULO 12 – FILTRANDO DADOS

Atualmente você irá perceber que desenvolvemos um sistema completo de cadastro de vendas, podemos registrar vendas realizadas com facilidade.

Mas o que nos fornece aparente facilidade também irá oferecer dificuldades pois a medida que formos registrando vendas ficará difícil localizar informações específicas, portanto criaremos agora um sistema simples que nos permita descobrir vendas através de filtragem.

Criando o Formulário de Filtragem

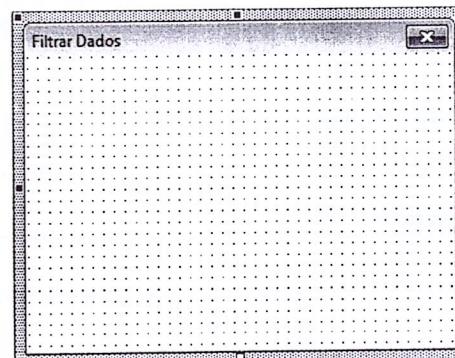
1º) Comece inserindo na planilha Principal um botão com o nome **Filtrar Dados** para executar o código a ser criado:

A	B	C	D	E	F	G	H	I	J
1	REGISTRO DE VENDAS								
2									
3									
4	CADASTRAR VENDA			GRÁFICO DE VENDAS			GERAR RELATÓRIO		
5									
6									
7	FILTRAR DADOS								
8									

2º) Insira para o botão o código abaixo:

```
Private Sub CommandButton4_Click()
    FrmFiltrar.Show
End Sub
```

3º) Crie um novo Formulário alterando as propriedades abaixo:

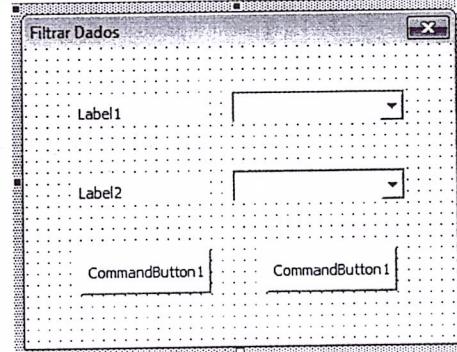


Propriedades do Formulário:

Name: FrmFiltrar

Caption: Filtrar Dados

4º) Insira e configure os componentes no formulário:





VBA

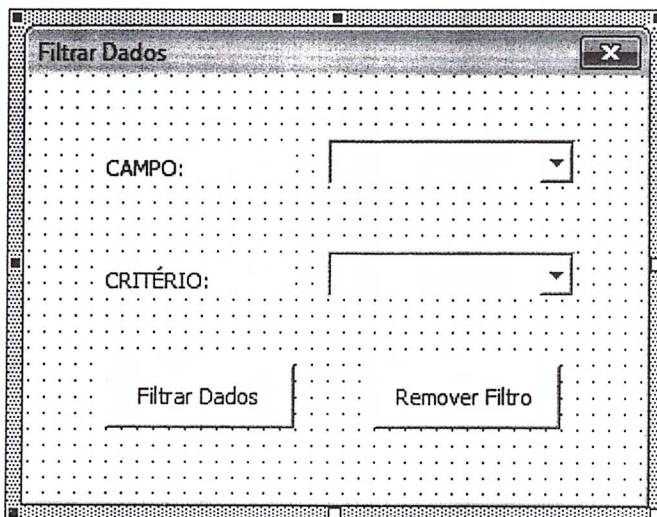
MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Configure as propriedades dos componentes conforme abaixo:

Componente	Propriedades
Label1	Name: LbCampo Caption: CAMPO:
Label2	Name: LbCriterio Caption: CRITÉRIO:
ComboBox1	Name: CbCampo
ComboBox2	Name: CbCriterio
CommandButton1	Name: CmdFiltrar Caption: Filtrar Dados
CommandButton2	Name: CmdRemover Caption: Remover Filtro

Formulário depois de configurado:



✓ Programando o Formulário de Filtragem

Configurando a Inicialização do Formulário

1º) Insira o código abaixo dentro do módulo de código para configurar a inicialização do formulário **FrmFiltrar**:

```
Private Sub UserForm_Initialize()
    ' Selecionando Cabeçalhos na Planilha
    ' Cadastro que irão preencher a ComboBox CbCampo
    Sheets("Cadastro").Select

    For lista = 2 To 10
        CbCampo.AddItem Cells(4, lista)
    Next

End Sub
```





2º) Experimente clicar no botão **Filtrar dados** que prontamente o formulário surgirá com a caixa de listagem **campo** já preenchida.

Configurando o Critério de Filtragem

1º) Insira o código abaixo no módulo do componente CbCampo:

```
Private Sub CbCampo_Change()
    Dim lista As Range

    'Armazena a informação da caixa de combinação CbCampo
    campo = CbCampo.Text

    'Limpa a caixa de combinação CbCriterio
    CbCriterio.Text = ""

    'Limpa a listagem da caixa de combinação CbCriterio
    CbCriterio.Clear

    'Verifica se a Informação da Caixa de Combinação CbCampo possui Critério
    If campo <> "Data da Venda" And campo <> "Vendedor" And campo <> "Modelo" Then

        'Seleciona a Planilha Dados
        Sheets("Dados").Select

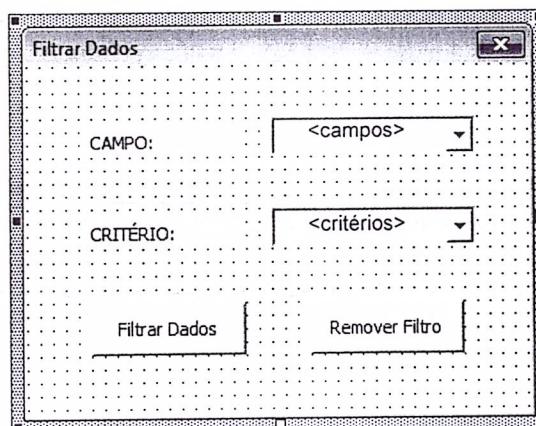
        'Realiza a Busca na Planilha Dados
        ActiveSheet.Cells.Find(campo).Select
        coluna = ActiveCell.Column
        linha = ActiveCell.Row

        'Seleciona a Região da Planilha que Contém os Dados
        ActiveSheet.Cells(linha + 1, coluna).Select
        ActiveSheet.Range(Selection, Selection.End(xlDown)).Select

        'Constroi a Listagem da Caixa de Combinação CbCriterio
        For Each lista In Selection
            CbCriterio.AddItem lista
        Next
        End If

    End Sub
```

3º) Execute o formulário de filtragem, repare que ao selecionar algum item da lista da caixa de combinação Campo será listado na caixa de combinação Critério somente os itens relacionados a busca:



**VBA**

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Programando a Execução da Filtragem pelo Formulário

Agora começaremos a programar a filtragem propriamente dita, iremos aproveitar um recurso que já existe no Excel, o AutoFiltro, porém de uma maneira totalmente diferente e personalizada.

✓ Configurando a Execução da Filtragem

1º) Abra o módulo do código do botão **Filtrar Dados** e insira o código abaixo:

```
Private Sub CmdFiltrar_Click()

    'Verifica se a Caixa de Combinação Campo está Vazia
    If CbCampo = "" Then
        MsgBox "Selecione um campo para a filtragem"
        Exit Sub
    End If

    'Verifica se a Caixa do Combinatório Criterio esta Vazia
    If CbCriterio = "" Then
        MsgBox "Selecione ou insira um critério para a filtragem"
        Exit Sub
    End If

    'Seleciona a planilha Cadastro para filtragem
    Sheets("Cadastro").Select

    'Seleciona toda a linha 4 da planilha
    ActiveSheet.Range("A4").CurrentRegion.Select

    'Ativa o Autofiltro
    Selection.AutoFilter

    'Obtem a coluna que contém o campo a ser filtrado
    ActiveSheet.Range("A4").Select
    ActiveSheet.Range(Selection, Selection.End(xlToRight)).Select
    ActiveSheet.Cells.Find(CbCampo).Select
    coluna = ActiveCell.Column

    'Seleciona toda a Área Planilha
    ActiveSheet.Range("A4").CurrentRegion.Select

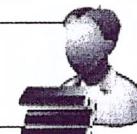
    'Executa o Autofiltro
    Selection.AutoFilter Field:=coluna, Criteria1:=CbCriterio

End Sub
```

2º) Abra o módulo do código do botão **Remover Filtro** e insira o código abaixo:

```
Private Sub CmdRemover_Click()
    'Desativa o Autofiltro
    Selection.AutoFilter
End Sub
```





Entendo o código: A maior parte dos códigos apresentados não é estranho, o que nos interessa no momento como funcionará a filtragem de dados.

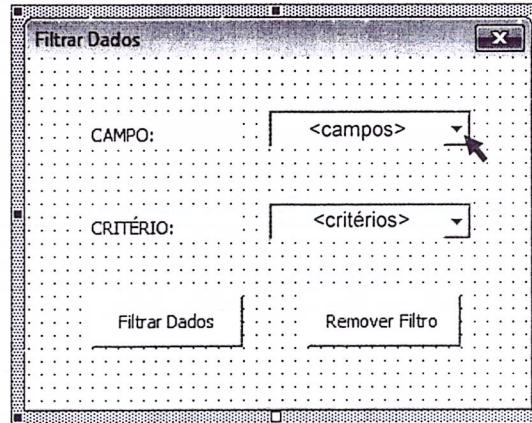
A linha de código **Selection.AutoFilter** é responsável em ativar o recurso de Auto Filtro e a linha **Selection.AutoFilter Field:=coluna, Criteria1:=critério** se encarrega de configurar o auto filtro informando o campo (Field) que será aplicado o critério de filtragem (Criteria1).

Se o código **Selection.AutoFilter** for executado uma segunda vez anula o auto filtro.

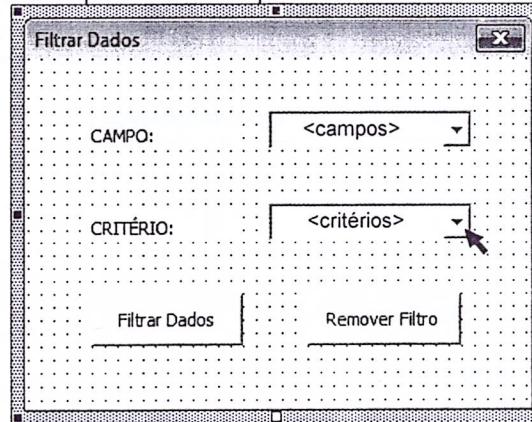
✓ Usando o Formulário de Filtragem

O uso desse formulário é extremamente simples:

1º) Selecione o campo a ser filtrado:



2º) Em seguida selecione o critério a ser aplicado ao campo:



3º) Ao ser aplicado a filtragem será feita a filtragem da planilha:

	A	B	C	D	E	F	G	H	I	J	K	L	M
Registro de Vendas													
4	Código	Data da Venda	Vendedor	Produto	Marca	Modelo	Memória	HDD	Monitor	Tipo	Acessório	Financiada	Valor da Venda
5	1	7/6/2010	ADRIANO	NoteBook	Itautec	W-7655	4Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.904,00
6	2	8/6/2010	MARCELO	NoteBook	Asus	AS-204A	8Gb	500 Gb	15 pol.	LED	NC	36 Vezes	R\$ 3.160,00
11	7	8/6/2010	ADRIANO	NoteBook	Acer	AC-897S	8Gb	500 Gb	17 pol.	LED	NC	36 Vezes	R\$ 4.360,00





4º) Caso o campo a ser filtrado não apresente listagem de critérios esse pode ser digitado manualmente:

5º) Para retirar a filtragem clique no botão **Remover Filtro** e a planilha retornará a seu estado normal.

A	B	C	D	E	F	G	H	I	J	K	L	M
Registro de Vendas												
Código Data da Venda Vendedor Produto Marca Modelo Memória HD Monitor Tipo Acessórios Financiado Valor da Venda												
1	7/6/2010	ADRIANO	NoteBook	Itautec	W-7655	4Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.904,00
2	8/6/2010	MARCELO	NoteBook	Asus	AS-204A	8Gb	500 Gb	15 pol.	LED	NC	36 Vezes	R\$ 3.160,00
3	8/6/2010	MARCELO	NetBook	Toshiba	TS-1034	4Gb	250 Gb	14 pol.	LCD	NC	Á Vista	R\$ 3.860,00
4	8/6/2010	JULIANA	Desktop	Acer	PST-1223B	3Gb	320 Gb	15 pol.	LCD	OK	12 Vezes	R\$ 1.500,00
5	8/6/2010	JULIANA	NetBook	Toshiba	TS-1045NT	2Gb	120 Gb	14 pol.	LCD	NC	Á Vista	R\$ 2.700,41
6	8/6/2010	ADRIANO	NetBook	Dell	ASP-1223D	2Gb	160 Gb	14 pol.	LED	NC	Á Vista	R\$ 1.540,00
7	8/6/2010	ADRIANO	NoteBook	Acer	AC-897S	8Gb	500 Gb	17 pol.	LED	NC	36 Vezes	R\$ 4.360,00
8	8/6/2010	MARCELO	Desktop	Asus	AS-1030SP	3Gb	200 Gb	17 pol.	LCD	OK	Á Vista	R\$ 3.480,00
9	8/6/2010	ADRIANO	NetBook	Positivo	PS-2040T	2Gb	120 Gb	14 pol.	LED	NC	Á Vista	R\$ 1.500,00
10	8/6/2010	JULIANA	Desktop	Itautec	TI-1020ST	8Gb	250 Gb	17 pol.	LED	OK	24 Vezes	R\$ 2.622,40

CAPÍTULO 13 – CRIANDO TABELA E GRÁFICOS DINÂMICOS

A Tabela Dinâmica é um dos recursos mais utilizados no Excel que permite gerar relatórios com eficiência pois é muito prático alterar a forma como os dados serão dispostos, a ideia aqui é reunir esse recurso ao VBA para aumentar as possibilidades da utilização da tabela dinâmica.

Criando um Formulário de Entrada de Dados para a Tabela Dinâmica

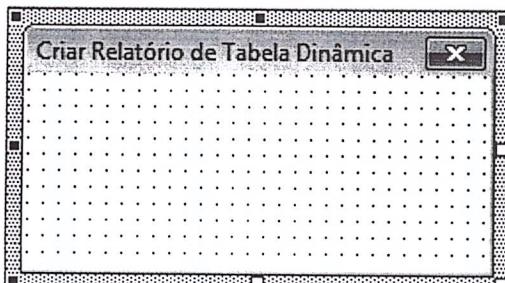
1º) Na planilha Principal insira o botão **RELATÓRIO DINÂMICO** que executará o formulário para criar tabela dinâmica:

A	B	C	D	E	F	G	H	I	J
1	REGISTRO DE VENDAS								
2									
3									
4	CADASTRAR VENDA			GRÁFICO DE VENDAS			GERAR RELATÓRIO		
5									
6									
7	FILTRAR DADOS			RELATÓRIO DINÂMICO					
8									

2º) Insira para o botão o código abaixo:

```
Private Sub CommandButton5_Click()
    FrmRelatDim.Show
End Sub
```

3º) Em seguida construa o formulário abaixo:



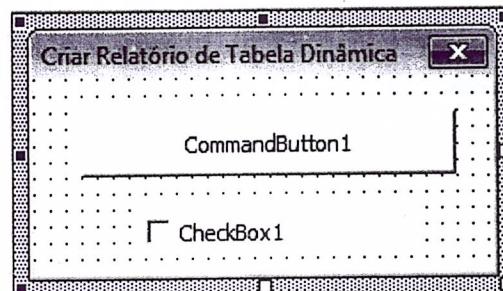


Propriedades do Formulário:

Name: FrmRelatDim

Caption: Criar Relatório de Tabela Dinâmica

4º) Insira e configure os componentes do formulário:



Configure as propriedades dos componentes conforme abaixo:

Componente	Propriedades	
CommandButton1	Name: CmdGerRelat	Caption: GERAR RELATÓRIO
Checkbox1	Caption: CkGraf	Caption: Criar Gráfico Dinâmico

Formulário depois de configurado:



**VBA**

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno

Programando o Formulário para criar o Relatório da Tabela Dinâmica

1º) Insira o código abaixo dentro do módulo de código do botão CmdGerRelat:

```
Private Sub CmdGerRelat_Click()
    'Formata a coluna Valor da Venda para garantir que seja numérica
    Dim celula As Range
    Sheets("Cadastro").Select
    ActiveSheet.Range("A4").Select
    Selection.End(xlToRight).Select
    inicio = ActiveCell.Row + 1
    coluna = ActiveCell.Column
    Selection.End(xlDown).Select
    fim = ActiveCell.Row
    Range(Cells(inicio, coluna), Cells(fim, coluna)).Select

    For Each celula In Selection
        col = celula.Column
        lin = celula.Row
        celula = Cells(lin, col).Text
        Cells(lin, col).Value = CCur(celula)
    Next

    'Seleciona os dados para a tabela dinâmica
    Range("A4").CurrentRegion.Select
    'Cria a planilha que será inserida a Tabela Dinâmica
    ultima = Sheets.Count
    Sheets.Add After:=Sheets(ultima)
    ultima = Sheets.Count
    Sheets(ultima).Name = "Relatório" & ultima
    ultima = Sheets.Count
    nome = Sheets(ultima).Name

    'Cria uma Nova Tabela Dinâmica (Tabela Pivô)
    '(Nesse momento é criado o Cache Dinâmico de Dados para a Tabela Dinâmica)
    ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
        "Cadastro!R4C1:R14C13").CreatePivotTable _
        TableDestination:=nome & "!R3C1", TableName:="Tabela dinâmica"

    'Preenche a região da "Coluna de Dados" com dados do campo DATA DA VENDA
    With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Data da Venda")
        .Orientation = xlColumnField
    End With

    'Preenche a região da "Linha de Dados" com dados do campo VENDEDOR
    With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
        .Orientation = xlRowField
    End With

    'Preenche a região da "Página de Dados" com dados do campo PRODUTO
    With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Produto")
        .Orientation = xlPageField
    End With

    'Preenche a região de "Dados" com dados do campo Valor da Venda
    ActiveSheet.PivotTables("Tabela dinâmica").AddDataField ActiveSheet. _
        PivotTables("Tabela dinâmica").PivotFields("Valor da Venda"), _
        "Soma de Valor da Venda", xlSum
```





```
'Formata a região de dados no formato Monetário  
ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Soma de Valor da Venda")  
.NumberFormat = "$ #,##0.00"  
  
'Fecha o formulário após conclusão  
Unload Me  
End Sub
```

Entendo o código: Nossa código é repleto de várias etapas, estão todas comentadas mas vamos analisar somente as partes mais importantes do códigos:

Trecho do código (Criar planilha para ser inserida a tabela dinâmica):

```
'Cria a planilha que será inserida a Tabela Dinâmica  
ultima = Sheets.Count  
Sheets.Add After:=Sheets(ultima)  
ultima = Sheets.Count  
Sheets(ultima).Name = "Relatório" & ultima  
ultima = Sheets.Count  
nome = Sheets(ultima).Name
```

Observações: Esse trecho do código tem a função de criar uma nova planilha para a tabela dinâmica, sempre que o usuário criar uma nova tabela será adicionado uma nova guia "Relatório" à pasta de trabalho.

Principal / Dados / Cadastro / Dados para o Gráfico / Relatório5 / **Relatório6** / F1

Trecho do código (Criar Cache Dinâmico de dados):

```
'Cria uma Nova Tabela Dinâmica (Tabela Pivô)  
'(Nesse momento é criado o Cache Dinâmico de Dados para a Tabela Dinâmica)  
ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=  
"Cadastro!R4C1:R14C13").CreatePivotTable  
TableDestination:=nome & "!R3C1", TableName:="Tabela dinâmica"
```

Observações: Esse trecho do código tem a função armazenar os dados que serão utilizados para a tabela dinâmica através do argumento **SourceData** e o argumento **TableDestination** informa que planilha e em que células dessa planilha será inserida a tabela dinâmica.

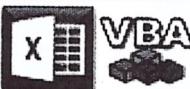
Trecho do código (Preencher dados para a região da coluna na tabela dinâmica):

```
'Preenche a região da "Coluna de Dados" com dados do campo DATA DA VENDA  
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Data da Venda")  
.Orientation = xlColumnField  
End With
```

Observações: Esse trecho do código tem a função de preencher dados do campo "Data da Venda" dentro da região **xlColumnField** (Campos da coluna), configurado pela propriedade **Orientation**.

(Explicação continua na próxima página)





VBA

MICROSOFT EXCEL 2013 COM VBA

Apostila do Aluno



Entendo o código: (Continuação)

Trecho do código (Preencher dados para a região da linha na tabela dinâmica):

```
' Preenche a região da "Linha de Dados" com dados do campo VENDEDOR
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .Orientation = xlRowField
End With
```

Observações: Esse trecho do código tem a função de preencher dados do campo "Vendedor" dentro da região **xlRowField** (Campos da linha), configurado pela propriedade **Orientation**.

Trecho do código (Preencher dados para a região de página da tabela dinâmica):

```
' Preenche a região da "Página de Dados" com dados do campo PRODUTO
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Produto")
    .Orientation = xlPageField
End With
```

Observações: Esse trecho do código tem a função de preencher dados do campo "Produto" dentro da região **xlPageField** (Campos da página), configurado pela propriedade **Orientation**.

Trecho do código (Preencher dados para a região de dados da tabela dinâmica):

```
' Preenche a região de "Dados" com dados do campo Valor da Venda
ActiveSheet.PivotTables("Tabela dinâmica").AddDataField ActiveSheet. _
    PivotTables("Tabela dinâmica").PivotFields("Valor da Venda"), _
    "Soma de Valor da Venda", xlSum
```

Observações: Esse trecho do código tem a função de preencher dados do campo "Valor da Venda" dentro da região de dados.

A principal diferença dessa região da tabela dinâmica é que geralmente ela armazena valores, pode também armazenar dados numéricos;

Porém essa região armazena tem a função de "Resumo de dados" ou "Área de Subtotal"; será nomeado com o identificador do resultado, a exemplo, "Soma de Valor da Venda", esse nome é obrigatório.

É atribuída a essa região um operador de resumo, os principais operadores são mostrados abaixo:

xlSum (Soma), xlAverage (Média), xlCount(Contagem), Min (Menor valor), Max (Maior Valor)

(Explicação continua na próxima página)





Entendo o código: (Continuação)

Trecho do código (Formatar a região de dados da tabela dinâmica):

```
'Formata a região de dados no formato Monetário
ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Soma de Valor da Venda") _
.NumberFormat = "$ #,##0.00#"
```

Observações: É possível aplicar uma formatação a região de dados através da propriedade **NumberFormat**, nesse caso, o formato aplicado foi o monetário.

3º) Clique no botão Relatório Dinâmico e veja se obteve o resultado abaixo:

	A	B	C	D
1	Produto	(Tudo)		
2				
3	Soma de Valor da Venda	Rótulos de Coluna		
4	Rótulos de Linha	7/6/2010	8/6/2010	Total geral
5	ADRIANO	R\$ 1.904,00	R\$ 7.400,00	R\$ 9.304,00
6	JULIANA		R\$ 6.822,81	R\$ 6.822,81
7	MARCELO		R\$ 10.500,00	R\$ 10.500,00
8	Total geral	R\$ 1.904,00	R\$ 24.722,81	R\$ 26.626,81

Inserindo mais Dados em uma Região da Tabela Dinâmica

É possível ter mais de um tipo de dado na tabela dinâmica apenas realizando algumas alterações iniciais em nosso código:

Vamos acrescentar agora o campo Modelo à região da linha de dados, acrescente as alterações abaixo:

```
'Preenche a região da "Linha de Dados" com dados do campo VENDEDOR
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .Orientation = xlRowField
    .Position = 1
End With

'Preenche a região da "Linha de Dados" com dados do campo MODELO
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Modelo")
    .Orientation = xlRowField
    .Position = 2
End With
```

Repare que a alteração que fizemos em nosso código é bem simples, o código para o campo modelo é semelhante ao código feito para o campo Vendedor, a diferença fica em parte pela propriedade **Position** que indica a ordem dos campos dentro da região de dados.





Execute o código e veja se você obteve o resultado abaixo:

A	B	C	D
1 Produto	(Tudo)		
2			
3 Soma de Valor da Venda	Rótulos de Coluna		
4 Rótulos de Linha	7/6/2010	8/6/2010	Total geral
5 =ADRIANO	R\$ 1.904,00	R\$ 7.400,00	R\$ 9.304,00
6 AC-897S		R\$ 4.360,00	R\$ 4.360,00
7 ASP-1223D		R\$ 1.540,00	R\$ 1.540,00
8 PS-2040T		R\$ 1.500,00	R\$ 1.500,00
9 W-7655	R\$ 1.904,00		R\$ 1.904,00
10 =JULIANA		R\$ 6.822,81	R\$ 6.822,81
11 PST-1223B		R\$ 1.500,00	R\$ 1.500,00
12 TI-1020ST		R\$ 2.622,40	R\$ 2.622,40
13 TS-1045NT		R\$ 2.700,41	R\$ 2.700,41
14 =MARCELO		R\$ 10.500,00	R\$ 10.500,00
15 AS-1030SP		R\$ 3.480,00	R\$ 3.480,00
16 AS-204A		R\$ 3.160,00	R\$ 3.160,00
17 TS-1034		R\$ 3.860,00	R\$ 3.860,00
18 Total geral	R\$ 1.904,00	R\$ 24.722,81	R\$ 26.626,81

Ocultando Dados dos Campos da Tabela Dinâmica

O conceito de ocultar dados em um determinado campo da tabela dinâmica é na verdade uma simples filtragem de dados que submetemos a tabela dinâmica.

Abaixo, um exemplo de ocultação de dados deixando visível somente o vendedor Adriano:

1º) Clique no local aonde iremos inserir o código:

```
'Preenche a região da "Linha de Dados" com dados do campo VENDEDOR
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .Orientation = xlRowField
    .Position = 1
End With
```

|

2º) Insira o código abaixo:

```
'Ocultando os vendedores Juliana e Marcelo na Tabela Dinâmica
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .PivotItems("JULIANA").Visible = False
    .PivotItems("MARCELO").Visible = False
End With
```

A propriedade **Visible** quando apresenta o valor "False" atuará sobre a propriedade **PivotItems** que age ocultando os dados citados dentro dessa propriedade.





3º) Execute o código e veja se é obtido o resultado abaixo:

	A	B	C	D
1	Produto	(Tudo)		
2				
3	Soma de Valor da Venda Rótulos de Coluna	<input checked="" type="checkbox"/>		
4	Rótulos de Linha	<input checked="" type="checkbox"/>	7/6/2010	8/6/2010 Total geral
5	ADRIANO		R\$ 1.904,00	R\$ 7.400,00 R\$ 9.304,00
6	AC-897S			R\$ 4.360,00 R\$ 4.360,00
7	ASP-1223D			R\$ 1.540,00 R\$ 1.540,00
8	PS-2040T			R\$ 1.500,00 R\$ 1.500,00
9	W-7655			R\$ 1.904,00
10	Total geral		R\$ 1.904,00	R\$ 7.400,00 R\$ 9.304,00

Ocultando Detalhes dos Campos da Tabela Dinâmica

Quando se tem mais de um campo associado na mesma região o segundo campo funciona como uma "área de detalhe", se não quiser exibir esses detalhes poderemos ocultá-los através da propriedade **ShowDetail**, a propriedade ShowDetail ao ser atribuída o valor "True" irá ocultar dados do campo.

Faça as alterações abaixo para ocultar detalhes do campo:

1º) Clique no local aonde iremos inserir alterações no código:

```
'Ocultando os vendedores Juliana e Marcelo na Tabela Dinâmica
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .PivotItems("JULIANA").Visible = False
    .PivotItems("MARCELO").Visible = False
End With
```

2º) Insira o código abaixo:

```
'Ocultando detalhes do campo VENDEDOR
With ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Vendedor")
    .PivotItems("ADRIANO").ShowDetail = False
End With
```

3º) Execute o código verificando a tabela dinâmica resultante:

	A	B	C	D
1	Produto	(Tudo)		
2				
3	Soma de Valor da Venda Rótulos de Coluna	<input checked="" type="checkbox"/>		
4	Rótulos de Linha	<input checked="" type="checkbox"/>	7/6/2010	8/6/2010 Total geral
5	ADRIANO		R\$ 1.904,00	R\$ 7.400,00 R\$ 9.304,00
6	Total geral		R\$ 1.904,00	R\$ 7.400,00 R\$ 9.304,00





Gerando um Gráfico a partir da Tabela Dinâmica

Podemos constatar até o momento que a tabela dinâmica apresenta muita eficiência quando se trata de manipulação de dados, essa eficiência também poderá ser vista criando gráficos, quando se cria um gráfico a partir de uma tabela dinâmica o gráfico recebe a definição de **Gráfico Dinâmico**.

Vamos fazer agora com que seja possível o usuário controlar a criação de um gráfico dinâmico, siga os passos abaixo para realizarmos essa tarefa:

1º) Altere o código para que nossa tabela dinâmica quando criada assuma a aparência abaixo:

	A	B	C	D
1	Produto	(Tudo)		
2				
3	Soma de Valor da Venda	Rótulos de Coluna		
4	Rótulos de Linha	7/6/2010	8/6/2010	Total geral
5	ADRIANO	R\$ 1.904,00	R\$ 7.400,00	R\$ 9.304,00
6	JULIANA		R\$ 6.822,81	R\$ 6.822,81
7	MARCELO		R\$ 10.500,00	R\$ 10.500,00
8	Total geral	R\$ 1.904,00	R\$ 24.722,81	R\$ 26.626,81

2º) Posicione o cursor no local aonde iremos inserir o código para a construção do gráfico dinâmico:

```
'Formata a região de dados no formato Monetário
ActiveSheet.PivotTables("Tabela dinâmica").PivotFields("Soma de Valor da Venda") _
.NumberFormat = "$ #,##0.00#"

|
'Fecha o formulário após conclusão
Unload Me
End Sub
```

3º) Insira o código abaixo:

```
'Verifica se o usuário confirma a criação do gráfico dinâmico, se essa for positiva
If CkGraf = True Then
    'Reserva memória para armazenar uma range
    Dim endereco As Range

    'Seleciona toda a planilha
    Range("A3").CurrentRegion.Select

    'Obtem informações da planilha
    celulas = Selection.Address
    planilha = ActiveSheet.Name
```

(Continua na próxima página)

