

# Índice analítico do Mostre-me

Visão geral do Soluções

## Como criar os formulários do Soluções

Adicionar Tudo à Lista

Caixa de Diálogo Vendas de Empregados

Editar Produtos

Encontrar Clientes

Inicialização

Inserir ou Editar Produtos

Introdução ao Soluções

Limitar Conteúdo da Lista

Mostrar Vendas

Reanexar NWIND

Sobre o Soluções

Soluções

Totais de Vendas

## Como criar os relatórios do Soluções

Carta de Vendas

Fatura

Lista de Telefones de Clientes

Médias de Vendas

Registro de Remessas

Vendas de Empregados

## Visão geral do Soluções

O Soluções é um aplicativo autodocumentado que apresenta uma coleção dos formulários, relatórios e elementos de interface mais comumente utilizados para criar aplicativos. Muitos exemplos usam recursos novos nesta versão do Microsoft Access.

### Quem deve usar o Soluções

Você achará os exemplos do Soluções mais úteis se já souber como criar consultas, formulários, relatórios e macros, e estiver familiarizado com as informações sobre criação de aplicativos da Parte 1, "Conectando os pontos", e o capítulo 5, "Fundamentos do Access Basic", no *Criando Aplicativos*.

### Dicas para usar o Soluções

Todos os exemplos deste aplicativo estão disponíveis a partir do formulário Soluções, que é exibido quando você abre o banco de dados Soluções. No formulário Soluções, selecione primeiro uma categoria de exemplos na caixa de listagem superior. Para abrir um formulário ou relatório de exemplo, clique duas vezes no exemplo da caixa de listagem inferior, ou selecione o exemplo e escolha o botão OK. O Soluções exibe automaticamente o formulário ou relatório de exemplo e seu tópico Destaques no Mostre-me, o sistema de Ajuda do Soluções.

O tópico do Mostre-me "Destaques..." identifica as técnicas usadas para criar o formulário ou relatório. Clique na técnica na qual você está interessado e, então, siga as instruções passo a passo para ver como a técnica foi aplicada no formulário ou relatório.

### Dicas para usar o Mostre-me

O tópico Mostre-me de um formulário ou relatório é exibido automaticamente quando a caixa de verificação da parte inferior do formulário Soluções está selecionada. Para exibir o Mostre-me de um formulário no modo Formulário ou de um relatório no modo Visualizar Impressão, pressione F1 ou clique no botão "Mostre-me" na barra de ferramentas.

Os recursos principais do Mostre-me são similares aos da Ajuda do Microsoft Access:

- Para usar uma frase ou palavra chave para procurar por um tópico, clique no botão Localizar da janela Mostre-me.
- Para imprimir um tópico, escolha Imprimir Tópico a partir do menu Arquivo da janela Mostre-me.
- Para manter a janela Mostre-me por cima, escolha Sempre Por Cima a partir do menu Ajuda da janela Mostre-me.

## **Destaques do formulário Limitar Conteúdo da Lista**

O formulário Limitar Conteúdo da Lista mostra como usar um grupo de opção para limitar os valores exibidos em uma caixa de listagem. Esta técnica faz o formulário Limitar Conteúdo da Lista funcionar:

- Exibindo somente os produtos da categoria selecionada.

## Limitando o conteúdo de uma lista com base no valor selecionado em um grupo de opção

Você pode usar os valores de um controle para determinar os valores exibidos em uma lista. Se o grupo de valores determinantes é estável, como as categorias do banco de dados Northwind, você pode exibí-los em um grupo de opção. Por exemplo, quando você selecionar uma categoria no formulário Limitar Conteúdo da Lista, somente produtos daquela categoria serão selecionados.

### Visão geral

Inclua o campo "Código da categoria" na consulta de origem da caixa de listagem "Selecionar produto". Defina o critério de "Código da categoria" com uma expressão que limite a lista aos produtos cujo código da categoria coincida com o código da categoria selecionado no grupo de opção "Selecionar categoria". Reconsulte a lista de produtos depois de selecionar uma categoria.

### Objetos usados neste exemplo

Tipo	Nome
Tabela	Produtos
Consulta	Limitar Lista de Produtos 2
Formulário	Limitar Conteúdo da Lista

### Passo a passo

#### 1 Crie a consulta Limitar Lista de Produtos 2.

Tabela	Campo (Classificar)	Critério
Produtos	Nome do produto (Ascendente)	--
	Identificação de produto	--
	Código da categoria	Formulários! [Limitar Conteúdo da Lista]! [Selecionar categoria]

#### 2 Estruture o formulário Limitar Conteúdo da Lista.

a) Defina as seguintes propriedades do formulário.

Propriedade	Definição
OrigemDoRegistro	Produtos
Legenda	Limitar Conteúdo da Lista
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim

- b) Crie um grupo de opção independente, e defina sua propriedade Nome como "Selecionar categoria".
- c) Para cada categoria do grupo de opção, adicione um botão de opção, e digite o nome da categoria em cada etiqueta.
- d) Defina a opção ValorDeOpção de cada botão de opção como um código de categoria. Por exemplo, o código da categoria Bebidas é 1, portanto você deve definir a propriedade ValorDeOpção do botão de opção Bebidas como 1.
- e) Defina o ValorDeOpção do grupo de opção como 1.
- f) Crie um procedimento de evento para a propriedade DepoisDeAtualizar do grupo de opção.
- g) Na seção Declarações do módulo Formulário.Limitar Conteúdo da Lista, digite:  
Option Explicit
- h) Crie uma caixa de listagem independente, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Valor</b>
Nome	Selecionar produto
TipoDeOrigemDaLinha	Tabela/Consulta
OrigemDaLinha	Limitar Lista de Produtos 2
NúmeroDeColunas	2
LargurasDasColunas	;0 cm
ColunaVinculada	2

**Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", capítulo 16, "Personalizando formulários"
Criar consultas	<i>Guia do Usuário</i> , capítulo 11, "Estruturando consultas seleção"

Fechar

Copiar

Imprimir

### **Procedimento do evento DepoisDeAtualizar para o grupo de opção Selecionar categoria**

```
Sub Selecionar_categoria_AfterUpdate ()  
    ' Reconsulta a caixa de listagem Selecionar produto.  
    Me![Selecionar produto].Requery  
End Sub
```

## **Destaques do formulário Adicionar Tudo à Lista**

O formulário Adicionar Tudo à Lista mostra como adicionar "(tudo)" à lista em uma caixa de listagem ou caixa de combinação. Esta técnica faz o formulário Adicionar Tudo à Lista funcionar:

- Adicionando "(tudo)" a uma lista.

## Adicionando "(tudo)" a uma lista

Quando você usa uma caixa de listagem ou caixa de combinação para inserir critérios de seleção, você pode querer a possibilidade de especificar todos os registros. No formulário Adicionar Tudo à Lista, a lista da caixa de combinação "Selecionar cliente" inclui "(tudo)".

### Visão geral

Crie uma função que adicione "(tudo)" à lista em uma caixa de combinação ou caixa de listagem. Crie uma caixa de listagem ou caixa de combinação, e defina sua propriedade TipoDeOrigemDaLinha como o nome da função.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Adicionar Tudo à Lista
Módulo	Adicionar Tudo à Lista

### Passo a passo

#### 1 Crie o módulo Adicionar Tudo à Lista.

- a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

- b) Crie a função AdicionarTudoÀListaDeCompanhia.

#### 2 Crie o formulário Adicionar Tudo à Lista de Companhia.

- a) Defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	Clientes
Legenda	Adicionar Tudo à Lista
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
AutoCentrar	Sim

- b) Adicione um cabeçalho de formulário ao formulário. Defina a propriedade "Altura do cabeçalho" do formulário como 0.
- c) Crie uma caixa de combinação no cabeçalho do formulário e defina suas propriedades como a seguir.

<u>Propriedade</u>	<u>Definição</u>
Nome	Selecionar cliente
TipoDeOrigemDaLinha	AdicionarTudoÀListaDeCompanhia
OrigemDaLinha	--
NúmeroDeColunas	2
LargurasDasColunas	7,62 cm;0 cm
ColunaVinculada	2
LimitarAUmaLista	Sim

DepoisDeAtualizar [Procedimento de evento]

d) Na seção Detalhe do formulário, adicione caixas de texto para os campos Código do cliente, Nome da companhia, Nome do contato, Telefone, e Fax.

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários" e capítulo 16, "Personalizando formulários"

**Fechar****Copiar****Imprimir**

## Função AdicionarTudoÀListaDeCompanhia

```
Function AdicionarTudoÀListaDeCompanhia (MyComboBox As Control, ID As Long,
Row As Long, Col As Long, Code As Integer) As Variant
    ' Adiciona "(tudo)" à lista de nomes de companhias.
    Static MyDB As Database, MyRecordset As Recordset
    Dim X As Long
    Select Case Code
        ' Inicializa.
        Case LB_INITIALIZE
            Set MyDB = DBEngine.Workspaces(0).Databases(0)
            Set MyRecordset = MyDB.OpenRecordset("SELECT [Nome da companhia],
[Código do cliente] FROM Clientes ORDER BY [Nome da companhia];")
            AdicionarTudoÀListaDeCompanhia = True
        ' Abre.
        Case LB_OPEN
            AdicionarTudoÀListaDeCompanhia = Timer
        ' Determina o número de linhas.
        Case LB_GETROWCOUNT
            MyRecordset.MoveLast
            AdicionarTudoÀListaDeCompanhia = MyRecordset.RecordCount
        ' Define o número de colunas.
        Case LB_GETCOLUMNCOUNT
            AdicionarTudoÀListaDeCompanhia = 2
        ' Define a largura da coluna.
        Case LB_GETCOLUMNWIDTH
            AdicionarTudoÀListaDeCompanhia = -1
        ' Obtém dados.
        Case LB_GETVALUE
            ' Se linha (tudo), define coluna vinculada como asterisco.
            If Row = 0 Then
                If Col = 0 Then
                    AdicionarTudoÀListaDeCompanhia = "(all)"
                Else
                    AdicionarTudoÀListaDeCompanhia = "*"
                End If
            Else
                ' Caso contrário, move para a próxima linha disponível e a
preenche.
                MyRecordset.MoveFirst
                For X = 1 To Row - 1
                    MyRecordset.MoveNext
                Next X
                If Col = 0 Then
                    AdicionarTudoÀListaDeCompanhia = MyRecordset.[Nome da
companhia]
                Else
```

```

                AdicionarTudoÀListaDeCompanhia = MyRecordset.[Código do
cliente]
                End If
            End If
            ' Fecha recordset.
            Case LB_END
                MyRecordset.Close
            End Select
        End Function
    End Function

```

Fechar

Copiar

Imprimir

### **Procedimento do evento DepoisDeAtualizar para a caixa de combinação "Selecionar cliente"**

```
Sub Selecionar_cliente_AfterUpdate ()
    ' Retorna registro(s) que coincidam com o valor selecionado na caixa de
    combinação "Selecionar cliente".
    If Me![Selecionar cliente].Column(1) = "*" Then
        DoCmd ShowAllRecords
    Else
        DoCmd ApplyFilter , "[Código do cliente] = Forms![Adicionar Tudo à
        Lista]![Selecionar cliente]"
    End If
End Sub
```

## **Destaques do formulário Reanexar NWIND**

O formulário Reanexar NWIND torna o módulo Reanexar NWIND e seu tópico Mostre-me disponíveis para visualização. Esta técnica faz a reanexação de tabelas na inicialização funcionar:

- Criando e executando código que atualize as informações de conexão das tabelas anexadas.

## Anexando tabelas na inicialização

Quando você usa múltiplos bancos de dados em um aplicativo, você pode garantir que as informações de conexão das tabelas anexadas sejam atualizadas quando o aplicativo for anexado. A macro AutoExec do banco de dados Soluções executa código que atualiza as anexações de tabelas do NWIND.MDB.

### Visão geral

Crie uma função que atualize as anexações de tabelas externas. Crie uma macro AutoExec que execute o código.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Macro	AutoExec
Módulo	Reanexar NWIND

### Passo a passo

#### 1 Crie o módulo Reanexar NWIND

- Crie a função TabelasEstãoAnexadas.
- Crie a função ObterNomeMDB.
- Crie a função ObterNomeMDB2.
- Crie a função SeqDeTamanho.

#### 2 Crie uma macro AutoExec com a seguinte ação.

<u>Condição</u>	<u>Ação</u>	<u>Argumentos</u>
Negado TabelasEstãoAnexadas()	ExecutarItem DoMenu	Barra de menu: Banco de dados Nome do Menu: Arquivo Comando: Fechar Banco de Dados

Esta ação deve ser a primeira de sua macro AutoExec. Você pode adicionar outras ações depois desta; por exemplo, você pode adicionar uma ação AbrirFormulário que exiba um formulário de inicialização ou um formulário principal.

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic" e capítulo 5, "Noções básicas sobre o Access Basic"
Criar macros	<i>Guia do Usuário</i> , capítulo 24, "Fundamentos de macros"

**Fechar****Copiar****Imprimir**

## Função TabelasEstãoAnexadas()

```
Function TabelasEstãoAnexadas () As Integer
    ' Atualiza informações de conexão em tabelas anexadas.
    '
    ' Número de tabelas anexadas para o medidor de progresso.
    Const MAXTABLES = 8
    Dim TableCount As Integer
    Dim FileName As String, SearchPath As String, Temp As String
    Dim ReturnValue As Variant, AccDir As String, I As Integer
    Dim MyTable As TableDef
    Dim MyDB As Database, MyRecords As Recordset
    Set MyDB = DBEngine.Workspaces(0).Databases(0)
    TabelasEstãoAnexadas = True
    ' Continua se as anexações estiverem quebradas.
    On Error Resume Next
    ' Abre tabela anexada para ver se as informações de conexão estão
    corretas.
    Set MyRecords = MyDB.OpenRecordset("Produtos")
    ' Sai se as informações de conexão estiverem corretas.
    If Err = 0 Then
        MyRecords.Close
        Exit Function
    End If
    ' Inicializa o medidor de progresso.
    ReturnValue = SysCmd(SYSCMD_INITMETER, "anexando tabelas", MAXTABLES)
    ' Obtém o nome do diretório onde MSACCESS.EXE está localizado.
    AccDir = SysCmd(SYSCMD_ACCESSDIR)
    ' Vê se NWIND.MDB está na localização padrão, <diretório do Access>\
    sampapps.
    ' Se não estiver, usa <diretório do Access> como ponto de partida para o
    diálogo AbrirArquivo.
    Temp = Dir$(AccDir & "sampapps\.")
    SearchPath = AccDir & IIf(Temp = "", "", "sampapps\")
    If (Dir$(SearchPath & "nwind.mdb") = "") Then
        FileName = ObterNomeMDB()           ' Exibe caixa de diálogo
        AbrirArquivo.
        FileName = Trim(FileName)
        If FileName = "" GoTo Exit_Failed   ' O usuário pressionou Cancelar.
    Else
        FileName = SearchPath & "nwind.mdb"
    End If
    ' Faz loop através de todas as tabelas, reanexando aquelas com seqüências
    de conexão de comprimento zero.
    TableCount = 1 ' Initialize TableCount for status meter.
    For I = 0 To MyDB.TableDefs.Count - 1
        Set MyTable = MyDB.TableDefs(I)
```

```

    If MyTable.Connect <> "" Then
        MyTable.Connect = ";DATABASE=" & FileName
        Err = 0
        MyTable.RefreshLink
        If Err <> 0 Then
            MsgBox "File '" & FileName & "' does not contain required
table '" & MyTable.SourceTableName & "'", 16, "Incorrect Database"
            TabelasEstãoAnexadas = False
            GoTo Exit_Failed
        End If
        TableCount = TableCount + 1
        ReturnValue = SysCmd(SYSCMD_UPDATEMETER, TableCount)
    End If
Next I
GoTo Exit_Succeeded
Exit_Failed:
    MsgBox "Você não pode executar o Soluções até localizar NWIND.MDB", 16,
"Não foi possível executar o Soluções"
    TabelasEstãoAnexadas = False
Exit_Succeeded:
    ReturnValue = SysCmd(SYSCMD_REMOVEMETER)
End Function

```

Fechar

Copiar

Imprimir

### Função ObterNomeMDB()

```
Private Function ObterNomeMDB () As String
    ' Retorna o caminho de NWIND.MDB escolhido pelo usuário na caixa de
    diálogo
    ' AbrirArquivo. (Esta função trabalha em conjunto com GetMDBName2 e
    SeqDeTamanho
    ' para exibir uma caixa de diálogo Arquivo-Abrir que peça ao usuário a
    ' localização de NWIND.MDB. Ela usa código encontrado em WZLIB.MDA).
    Const OFN_SHAREAWARE = &H4000
    Const OFN_PATHMUSTEXIST = &H800
    Const OFN_HIDEREADONLY = &H4
    Dim OFN As wlib_GetFileNameInfo
    ' Preenche estrutura ofn que é passada para wlib_GetFileName
    OFN.hWndOwner = 0
    OFN.szFilter = "Databases (*.mdb)|*.mdb|All (*.*)|*.*||"
    OFN.NFilterIndex = 1
    OFN.szTitle = "Onde está NWIND.MDB?"
    OFN.Flags = OFN_SHAREAWARE Or OFN_PATHMUSTEXIST Or OFN_HIDEREADONLY
    OFN.szDefExt = ".mdb"
    ' Chama função wlib_GetFileName e interpreta resultados.
    If (ObterNomeMDB2(OFN, True) = False) Then
        ObterNomeMDB = SeqDeTamanho(OFN.szFile)
    Else
        ObterNomeMDB = ""
    End If
End Function
```

Fechar

Copiar

Imprimir

### Função ObterNomeMDB2()

```
Private Function ObterNomeMDB2 (gfni As wlib_GetFileNameInfo, ByVal fOpen As Integer) As Long
    ' Esta função age como uma cobertura para MSAU_GetFileName em MSAU200.DLL
    ' wlib_GetFileName termina todas as seqüências da estrutura gfni com nulos e
    ' então chama a versão DLL da função. Ao retornar de MSAU200.DLL, os caracteres nulos são removidos das seqüências de gfni.
    Dim lRet As Long
    gfni.szFilter = RTrim$(gfni.szFilter) & Chr$(0)
    gfni.szCustomFilter = RTrim$(gfni.szCustomFilter) & Chr$(0)
    gfni.szFile = RTrim$(gfni.szFile) & Chr$(0)
    gfni.szFileTitle = RTrim$(gfni.szFileTitle) & Chr$(0)
    gfni.szInitialDir = RTrim$(gfni.szInitialDir) & Chr$(0)
    gfni.szTitle = RTrim$(gfni.szTitle) & Chr$(0)
    gfni.szDefExt = RTrim$(gfni.szDefExt) & Chr$(0)
    lRet = wlib_MSAU_GetFileName(gfni, fOpen)
    gfni.szFilter = SeqDeTamanho(gfni.szFilter)
    gfni.szCustomFilter = SeqDeTamanho(gfni.szCustomFilter)
    gfni.szFile = SeqDeTamanho(gfni.szFile)
    gfni.szFileTitle = SeqDeTamanho(gfni.szFileTitle)
    gfni.szInitialDir = SeqDeTamanho(gfni.szInitialDir)
    gfni.szTitle = SeqDeTamanho(gfni.szTitle)
    gfni.szDefExt = SeqDeTamanho(gfni.szDefExt)
    wlib_GetMDBName2 = lRet
End Function
```

Fechar

Copiar

Imprimir

### Função SeqDeTamanho()

```
Private Function SeqDeTamanho (szTmp As String) As String
    ' Se a seqüência termina com nulos, retorna uma seqüência truncada.
    Dim ich As Integer
    ich = InStr(szTmp, Chr$(0))
    If ich Then
        SeqDeTamanho = Left$(szTmp, ich - 1)
    Else
        SeqDeTamanho = szTmp
    End If
End Function
```

## **Destaques do formulário Introdução ao Soluções**

O formulário Introdução ao Soluções mostra como criar um formulário que os usuários possam desativar quando não quiserem mais usá-lo. Esta técnica faz o formulário Introdução ao Soluções funcionar:

- Criando um formulário que os usuários possam desativar.

## Criando um formulário que os usuários possam desativar

Você pode exibir um formulário que proporcione informações introdutórias sobre seu aplicativo na inicialização e o qual os usuários possam optar por não exibir novamente. Por exemplo, o formulário Introdução ao Soluções é exibido depois do formulário Inicialização e antes do formulário de menu principal do Soluções toda vez que você inicia o aplicativo de exemplo Soluções. Se você resolver que não o quer mais exibido, você pode selecionar a caixa de verificação na parte inferior do formulário.

### Visão geral

Crie uma tabela com um campo Sim/Não. Crie um formulário que tenha os seguintes controles: etiquetas que exibam informações sobre seu aplicativo, uma caixa de verificação vinculada ao campo Sim/Não, e um botão de comando que feche o formulário. Crie um formulário de menu principal. Crie um formulário de inicialização. Na propriedade AoTimer do formulário de inicialização, modifique o procedimento de evento para que ele use o valor do campo Sim/Não para determinar se o formulário Introdução ao Soluções ou o formulário do menu principal deve ser exibido.

### Objetos usados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Opções
Formulário	Soluções, Introdução ao Soluções, Inicialização

### Passo a passo

#### 1 Crie a tabela Opções.

- Crie um campo chamado Ocultar introdução ao Soluções, e defina seu tipo de dados como Sim/Não. (Você não precisa criar uma chave primária para a tabela).
- Altere para modo Folha de Dados, e digite Não no campo Ocultar introdução ao Soluções.

#### 2 Crie o formulário Introdução ao Soluções.

- Defina as seguintes propriedades do formulário.

<b>Propriedade</b>	<b>Definição</b>
OrigemDoRegistro	Opções
Legenda	Introdução ao Soluções
ModoPadrão	Formulário simples
VisualizaçãoAutoriza da	Formulário
EdiçãoPadrão	Impossível adicionar registros
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
EstiloDaBorda	Fino
BotãoMin	Não
BotãoMax	Não
AoFechar	<u>[Procedimento de evento]</u>

- b) Crie etiquetas para exibir informações sobre o aplicativo.
- c) Crie uma caixa de verificação. Na propriedade Legenda de sua etiqueta anexada, digite: **Não exibir este formulário novamente.** Defina então as seguintes propriedades da caixa de verificação.

<b>Propriedade</b>	<b>Definição</b>
Nome	Ocultar introdução ao Soluções
OrigemDoControle	Ocultar introdução ao Soluções
ValorPadrão	0

- d) Crie um botão de comando, e defina as seguintes propriedades.

<b>Propriedade</b>	<b>Definição</b>
Nome	OK
Legenda	OK
Padrão	Sim
AoClicar	[ <u>Procedimento de evento</u> ]

### **3 Crie o formulário Soluções.**

### **4 Crie o formulário Inicialização.**

### **5 Modifique o procedimento de evento da propriedade AoTimer do formulário Inicialização.**

#### **Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", e capítulo 16, "Personalizando formulários"



### **Procedimento do evento Fechar para o formulário Introdução ao Soluções**

```
Sub Form_Close ()  
    ' Abre o formulário Soluções.  
    DoCmd OpenForm "Soluções"  
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Clicar do botão de comando OK**

```
Sub OK_Click ()
On Error GoTo Erro_OK_Click
    ' Fecha o formulário. Este código foi criado pelo Assistente para
    Botões de Comando.
    DoCmd Close
Sai_OK_Click:
    Exit Sub
Erro_OK_Click:
    MsgBox Error$
    Resume Sai_OK_Click
End Sub
```

[Fechar](#)[Copiar](#)[Imprimir](#)

## Procedimento do evento Timer do formulário Inicialização

O texto e o código que você precisa adicionar ao procedimento de evento para abrir o formulário Introdução ao Soluções ou o formulário Soluções são exibidos em vermelho.

```
Sub Form_Timer ()
    ' Abre o formulário Introdução ao Soluções ou o formulário Soluções.
    ' Fecha o formulário de inicialização.
    '
    Dim MyDB As Database
    Dim MyRecordset As Recordset
    Set MyDB = DBEngine.Workspaces(0).Databases(0)
    Set MyRecordset = MyDB.TableDefs("Opções").OpenRecordset()
    ' Primeira instrução If usada pelo Soluções para reiniciar a propriedade
    IntervaloDeTempo.
    If Me.TimerInterval <> 0 Then
        TimerInterval = 0
        ' Use o valor do campo Ocultar Introdução ao Soluções da tabela
        Opções para determinar
        ' qual formulário abrir.
        If MyRecordset.Fields("Ocultar Introdução ao Soluções").Value = 0
Then
            DoCmd OpenForm "Introdução ao Soluções"
        Else
            DoCmd OpenForm "Soluções"
        End If
    End If
    ' Fecha recordset.
    MyRecordset.Close
    ' Fecha formulário de inicialização.
    DoCmd Close A_FORM, "Inicialização"
End Sub
```

## **Destaques do relatório Médias de Vendas**

O relatório Médias de Vendas mostra como apresentar múltiplos fatos para cada cabeçalho de linha em um relatório tabela de referência cruzada. Esta técnica faz o relatório Médias de Vendas funcionar:

- Criando um relatório tabela de referência cruzada de múltiplos fatos.

## Criando um relatório tabela de referência cruzada de múltiplos fatos

Você pode criar um relatório tabela de referência cruzada que imprima múltiplos fatos para cada cabeçalho de linha. Por exemplo, o relatório Médias de Vendas lista o mínimo, a média e o máximo de vendas mensais para cada vendedor.

### Visão geral

Use uma consulta seleção como origem dos registros do relatório, e crie um layout tabela de referência cruzada estruturando um relatório de múltiplas colunas. Crie uma caixa de texto independente no cabeçalho Sobrenome para usar como um indicador para determinar qual cabeçalho de linha imprimir. Crie um procedimento de evento na propriedade AoFormatar do cabeçalho Sobrenome que use a propriedade PróximoRegistro para imprimir os cabeçalhos de linha em duas colunas. Crie um procedimento de evento na propriedade AoFormatar do cabeçalho Data de envio que use a função Mês e a propriedade Esquerda para garantir que os dados corretos sejam impressos em cada coluna. Quando não houver dados para um coluna, use a propriedade PróximoRegistro para evitar o avanço para o próximo registro, e use a propriedade ImprimirSeção para evitar imprimir a próxima seção.

### Objetos usados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Empregados, Pedidos
Consulta	Subtotais dos Pedidos, Médias de Vendas
Relatório	Médias de Vendas

### Passo a passo

**1 Importe a consulta Subtotais dos Pedidos de NWIND.MB, se você já não o tiver feito.**

**2 Crie a consulta Médias de Vendas.**

<b>Tabela/ consulta</b>	<b>Campo/ expressão</b>	<b>Total</b>	<b>Critério</b>
Empregados	Sobrenome	Agrupar por	--
	Primeiro nome	Agrupar por	--
Pedidos	Número do Pedido	Agrupar por	--
	Data de envio	Agrupar por	--
	Ano: Formato([Data de envio], "aaaa")	Expressão	"1993"
Subtotais dos Pedidos	Subtotal	Agrupar por	--

**3 Estruture o relatório Médias de Vendas.**

a) Defina as seguintes propriedades de classificação e agrupamento.

<b>Campo/ expressão</b>	<b>Ordem de Classificação</b>	<b>Propriedades de grupo</b>
Sobrenome	Ascendente	Cabeçalho de Grupo: Sim Rodapé de Grupo: Sim Reagrupar: Cada Valor

Data de envio Ascendente

Número de  
Caracteres: 1  
Manter Junto: Não  
Cabeçalho de  
Grupo: Sim  
Rodapé de Grupo:  
Não  
Reagrupar: Mês  
Número de  
Caracteres: 1  
Manter Junto: Não

b) Defina a propriedade Largura do relatório como 24,841 cm.

c) Defina as seguintes opções de Configuração de Impressão.

<b>Opção</b>	<b>Definição</b>
Orientação	Paisagem
Margens	Esquerda: 1,016 cm Direita: 1,016 cm Superior: 2,540 cm Inferior: 2,540 cm
Itens através:	14
Espaçamento entre colunas	0,254 cm
Tamanho do item	Largura: 1,524 cm Altura: 0 cm
Layout do item	Horizontal

d) No cabeçalho de página, cole as etiquetas para os tipos de moeda e os nomes dos meses.

e) No cabeçalho Sobrenome, crie as seguintes caixas de texto, uma abaixo da outra.

<b>Nome</b>	<b>OrigemDoRegistro</b>
Sobrenome do vendedor	=[Sobrenome]&"
Primeiro nome	Primeiro nome

f) No cabeçalho Sobrenome, crie as seguintes etiquetas no topo das caixas de texto que você criou no Passo 3e.

<b>Nome</b>	<b>Legenda</b>
L - Mínimo	Mínimo:
L - Média	Média:
L - Máximo	Máximo:

g) No cabeçalho Sobrenome, crie as seguintes caixas de texto.

<b>Nome</b>	<b>Outras propriedades</b>
Imprimir o quê	OrigemDoRegistro: -- Visível: Não

Largura da Coluna	OrigemDoControle : =(0.6+0.1)*1440 Formato: Número geral Visível: Não
Margem Esquerda	OrigemDoControle =0.4*1440 Formato: Número geral Visível: Não

h) No cabeçalho Data de envio, crie as seguintes caixas de texto, uma abaixo da outra. Para todas as três, defina a propriedade Formato como Standard, e a propriedade CasasDecimais como 0.

Nome	OrigemDoRegistro
Mínimo do mês	=Min([Subtotal])
Média do mês	=Avg([Subtotal])
Máximo do mês	=Max([Subtotal])

i) Defina as seguintes propriedades de seção.

Seção	Propriedade	Definição
Cabeçalho de página	AoFormatar	[ <u>Procedimento de evento</u> ]
Cabeçalho Sobrenome	ManterJunto AoFormatar	Sim [ <u>Procedimento de evento</u> ]
Cabeçalho Data de envio	Manter Junto AoFormatar	Sim [ <u>Procedimento de evento</u> ]
Seção detalhe Rodapé Sobrenome	Altura NovaLinhaOuColuna Altura	0 cm Depois da seção 0 cm
Rodapé do relatório	Altura	0 cm

j) Na seção Declarações do módulo Relatório.Médias de Vendas, digite:

Option Explicit

### Referências cruzadas

Para informações sobre	Consulte
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Fundamentos do Access Basic"

Criar consultas	<i>Guia do Usuário</i> , capítulo 10, "Noções básicas de consultas", e capítulo 11, "Estruturando consultas seleção"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Noções básicas de relatórios", e capítulo 21, "Estruturando relatórios"
Definir propriedades de classificação e agrupamento	<i>Guia do Usuário</i> , capítulo 22, "Classificando e agrupando dados"

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o cabeçalho de página**

```
Sub PageHeader1_Format (Cancel As Integer, FormatCount As Integer)
    ' Define a caixa de texto "Imprimir o quê" como 0 no topo da página.
    Me![Imprimir o quê] = 0
End Sub
```

Fechar

Copiar

Imprimir

### Procedimento do evento Formatar para o cabeçalho Sobrenome

```
Sub GroupHeader2_Format (Cancel As Integer, FormatCount As Integer)
    ' Imprime as informações dos cabeçalhos de grupo na ordem correta.
    '
    ' Imprime as caixas de texto Sobrenome e Primeiro nome do vendedor,
    ' oculta as etiquetas Mínimo, Média e Máximo,
    ' define a caixa de texto "Imprimir o quê" como -1, e não avança para o
    próximo registro.
    If Me![Imprimir o quê] = 0 Then
        Me![Sobrenome do vendedor].Visible = True
        Me![Primeiro nome].Visible = True
        Me![L - Mínimo].Visible = False
        Me![L - Média].Visible = False
        Me![L - Máximo].Visible = False
        Me![Imprimir o quê] = -1
        Me.NextRecord = False
    ' Oculta as caixas de texto Sobrenome do vendedor e Primeiro nome,
    ' imprime as etiquetas Mínimo, Média e Máximo,
    ' e define a caixa de texto "Imprimir o quê" como 0.
    Else
        Me![Sobrenome do vendedor].Visible = False
        Me![Primeiro nome].Visible = False
        Me![L - Mínimo].Visible = True
        Me![L - Média].Visible = True
        Me![L - Máximo].Visible = True
        Me![Imprimir o quê] = 0
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o cabeçalho "Data de envio"**

```
Sub GroupHeader3_Format (Cancel As Integer, FormatCount As Integer)
    ' Imprime dados na coluna certa.
    '
    ' Não avança para o próximo registro nem imprime a próxima seção.
    If Me.Left < Me![Margem esquerda] + (Month(Me![Data de envio]) + 1) * Me!
[Largura da coluna] Then
        Me.NextRecord = False
        Me.PrintSection = False
    End If
End Sub
```

## **Destaques da Lista de Telefones de Clientes**

A Lista de Telefones de Clientes mostra como identificar a primeira e última companhias impressas em cada página. Esta técnica faz a Lista de Telefones de Clientes funcionar:

- Imprimindo a primeira e última entradas em uma página no cabeçalho de página

## Imprimindo a primeira e a última entrada em uma página no cabeçalho de página

Tal como os dicionários tornam fácil encontrar palavras, seus relatórios podem tornar fácil a entrada de dados identificando a primeira e última entradas em uma página. Na Lista de Telefones de Clientes, o primeiro nome de companhia em uma página é impresso no lado esquerdo do cabeçalho da página, e o último nome de companhia é impresso na direita.

### Visão geral

Crie um relatório que classifique dados por nome da companhia. No cabeçalho de página, crie uma caixa de texto para números de página que use a propriedade Páginas em sua expressão (para forçar uma reportagem de duas passagens). Crie um procedimento de evento para a propriedade AoFormatar do rodapé de página que, na primeira passagem, ponha o último registro do recordset na matriz. Crie um procedimento de evento para a propriedade AoFormatar do cabeçalho de página que, na segunda passagem, preencha as caixas de texto Primeira entrada e Última entrada.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Tabela	Clientes
Relatório	Lista de Telefones de Clientes

### Passo a passo

- **Estruture o relatório Lista de Telefones de Clientes.**
  - a) Defina a propriedade OrigemDoRegistro do relatório como Clientes.
  - b) Adicione um cabeçalho e um rodapé de relatório. Defina as propriedades Altura de ambas as seções como 0.
  - c) Na caixa Classificação e Agrupamento, selecione Nome da companhia na coluna Campo/expressão, e defina Ordem de Classificação como Ascendente.
  - d) Na seção detalhe, crie as caixas de texto Nome da companhia, Nome do contato, e Telefone. Recorte suas etiquetas anexadas, e as cole no cabeçalho de página.
  - e) Defina a propriedade ManterJunto da seção detalhe como Sim.
  - f) Crie caixas de texto nas seções seguintes e exclua suas etiquetas anexadas.

<u>Seção</u>	<u>Nome</u>	<u>OrigemDoContr ole</u>
Cabeçalho de página	Primeira entrada	--
	Última entrada	--
Rodapé de página	Número de página	=Página & " de " & Páginas & " Páginas"

- g) Torne iguais as larguras das caixas de texto Nome da companhia, Primeira entrada, e Última entrada.
- h) Na seção Declarações do módulo Relatório.Lista de Telefones de Clientes. declare variáveis.
- i) Crie procedimentos de evento para as seguintes seções.

<u>Seção</u>	<u>Propriedade</u>	<u>Definição</u>
Cabeçalho de página	AoFormatar	[ <u>Procedimento de evento</u> ]
Rodapé de página	AoFormatar	[ <u>Procedimento de evento</u> ]
Rodapé de	AoFormatar	[ <u>Procedimento de</u>

**Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios", e capítulo 21, "Estruturando relatórios"
Definir propriedades de classificação e agrupamento	<i>Guia do Usuário</i> , capítulo 22, "Classificando e agrupando dados"

Fechar

Copiar

Imprimir

### **Seção Declarações do módulo Relatório.Lista de Telefones de Clientes**

Option Compare Database 'Usa ordem do banco de dados em comparações de seqüências

Option Explicit

' Matriz para o último registro de cada página e o último registro do recordset.

Dim gLast\$()

' Indicador para o número de passagens pelo relatório.

Dim gLastPage%

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o cabeçalho de página**

```
Sub PageHeader0_Format (Cancel As Integer, FormatCount As Integer)
    ' Durante a segunda passagem, preenche as caixas de texto Primeira
    entrada e Última entrada.
    If gLastPage% = True Then
        Reports![Lista de Telefones de Clientes].[Primeira entrada] =
Reports![Lista de Telefones de Clientes]![Nome da companhia]
        Reports![Lista de Telefones de Clientes].[Última entrada] = gLast$
(Reports![Lista de Telefones de Clientes].Page)
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o rodapé de página**

```
Sub PageFooter2_Format (Cancel As Integer, FormatCount As Integer)
    ' Durante a primeira passagem, aumenta o tamanho da matriz e insere o
    último registro da página na matriz.
    If Not gLastPage% Then
        ReDim Preserve gLast$(Reports![Lista de Telefones de Clientes].Page +
1)
        gLast$(Reports![Lista de Telefones de Clientes].Page) = Reports!
[Lista de Telefones de Clientes].[Nome da companhia]
    End If
End Sub
```

Fechar

Copiar

Imprimir

### Procedimento do evento Formatar para o rodapé de relatório

```
Sub ReportFooter4_Format (Cancel As Integer, FormatCount As Integer)
    Dim MyDB As Database
    Dim MyRS As Recordset
    ' Define indicador depois da primeira passagem ter sido completada.
    gLastPage% = True
    ' Abre o recordset para o relatório.
    Set MyDB = DBEngine.WorkSpaces(0).Databases(0)
    Set MyRS = MyDB.OpenRecordset("Clientes")
    ' Move para o último registro do recordset.
    MyRS.MoveLast
    ' Insere o último registro na matriz.
    gLast$(Reports![Lista de Telefones de Clientes].Page) = MyRS.[Nome da
compañia]
End Sub
```

## Destaques do formulário Editar Produtos

O formulário Editar Produtos mostra como criar um formulário no qual você poderá editar registros existentes mas não poderá inserir novos. Estas técnicas fazem o formulário Editar Produtos funcionar:

- Impedindo os usuários de moverem-se automaticamente para outro registro.
- Criando uma barra de ferramentas e menu com botões e comandos limitados a tarefas de edição.
- Limitando os valores na caixa de combinação "Selecionar produto" pelo valor escolhido na caixa de combinação "Selecionar categoria".
- Encontrando o registro para o produto selecionado na caixa de combinação "Selecionar produto".
- Atualizando a lista na caixa de combinação "Selecionar produto" após mover-se para outro registro.

Esta técnica melhora o formulário Editar Produtos:

- Alterando a cor da caixa de texto "Unidades em estoque" e exibindo uma mensagem quando o número de unidades é inferior ao nível de novos pedidos.

## Impedindo os usuários de moverem-se automaticamente para outro registro

Em um formulário somente edição, você pode querer impedir que os usuários avancem automaticamente para outro registro. No formulário Editar Produtos, por exemplo, não se pode avançar para outro registro nas formas usuais, tais como pressionando TAB ou ENTER no último campo, utilizando os botões de navegação, ou usando comandos de menus ou botões da barra de ferramentas.

### Visão geral

Remova os botões de navegação do formulário. Crie um botão de comando transparente e utilize um procedimento de evento em sua propriedade AoEntrar que utiliza o método SetFocus para mover o foco para a caixa de combinação "Identificação do fornecedor". Posicione o botão de comando no fim da ordem de tabulação.

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nome</u>
Tabela	Produtos
Formulário	Editar Produtos

### Passo a passo

#### 1 Estructure o formulário Editar Produtos.

a) Defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Valor</u>
OrigemDoRegistro	Produtos
Legenda	Editar Produtos
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
EdiçãoPadrão	Impossível adicionar registros
MenuDeAtalho	Não
BarraDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim

b) Na seção Detalhe, adicione os campos da tabela Produtos. Utilize caixas de combinação para os campos Identificação do Fornecedor e Código da Categoria e uma caixa de verificação para o campo Descontinuado. Utilize caixas de texto para todos os outros campos.

c) Defina a propriedade Ativado para todos os controles na seção Detalhe como Não. Para impedir as etiquetas de acender em quando os controles estiverem ativados, separe as etiquetas anexadas dos controles recortando as etiquetas e depois colando-as.

d) Como é um campo Contador, faça a caixa de texto somente de exibição Identificação do Produto definindo as seguintes propriedades.

<u>Propriedade</u>	<u>Valor</u>
CorDoFundo	(mesma cor que a Cor do Fundo da seção Detalhe)

EstiloDaBorda    Limpar  
EstiloDaFonte    Negrito

**2 Na seção Detalhe, crie um botão de comando "Vá para a identificação do fornecedor."**

a) Defina suas propriedades e crie um procedimento de evento como a seguir.

<u>Propriedade</u>	<u>Valor</u>
Nome	Vá para a identificação do fornecedor
Transparente	Sim
Ativado	Não
AoEntrar	[ <u>Procedimen</u> <u>tô de</u> <u>evento</u> ]

b) Para impedir que os usuários cliquem inadvertidamente no botão de comando, oculte-o atrás da caixa de combinação "Identificação do fornecedor". (Faça o botão de comando do mesmo tamanho que a caixa de combinação, coloque-o na parte superior da caixa de combinação e, então, escolha Enviar para Trás no menu Formatar.)

**3 Na seção Declarações do módulo Form.Editar Produtos, digite:**

Option Explicit

**4 Edite a ordem de tabulação do formulário.** Certifique-se de que Identificação do Fornecedor seja o primeiro campo na lista e que Vá Para a Identificação do Fornecedor e Identificação de Produto sejam os dois últimos campos na lista.

---

**Dica** Para mover de um registro para outro neste formulário, entre critério de procura.

---

**Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários," e capítulo 16, "Personalizando formulários"

Fechar

Copiar

Imprimir

### **Inserindo procedimento de evento para o botão de comando "Identificação do Fornecedor"**

```
Sub Go_To_Supplier_ID_Enter ()  
    ' Vai a para caixa de combinação "Identificação do fornecedor" para  
    ' impedir avanço para outro registro.  
    Me![Supplier ID].SetFocus  
End Sub
```



<b>Menu</b>	<b>Comando</b>
Arquivo	Fechar
Editar	Desfazer Desfazer Campo/Registro Atual Recortar Copiar Colar

---

**Observação** A barra de menu personalizada no formulário Editar Produtos também inclui os menus Exibir, Janela e Ajuda por isso você pode alternar facilmente entre os modos e utilizar os recursos normais do menu Modo Formulário.

---

- b) Defina a propriedade BarraDeMenu do formulário como o grupo de macro, Barra de Menu Editar Produtos.

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar menus	<i>Criando Aplicativos</i> , capítulo 2, "Estruturando uma interface de usuário"
Criar barras de ferramentas	<i>Guia do Usuário</i> , Apêndice B, "Trabalhando com barras de ferramentas"



### **Ativando procedimentos de evento para o formulário Editar Produtos**

```
Sub Form_Activate ()  
    ' Ocultar barra de ferramentas Modo Formulário incorporada  
    ' Exibir barra de ferramentas Editar Produtos.  
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_NO  
    DoCmd ShowToolbar "Barra de Ferramentas Editar Produtos, A_TOOLBAR_YES  
End Sub
```

Fechar

Copiar

Imprimir

### **Desativando procedimento de evento para o formulário Editar Produtos**

```
Sub Form_Deactivate ()
    ' Ocultar barra de ferramentas Editar Produtos.
    ' Exibir barra de ferramentas Modo Formulário incorporada.
    DoCmd ShowToolbar "Barra de ferramentas Editar Produtos, A_TOOLBAR_NO
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_YES
End Sub
```

## Limitando o conteúdo de uma lista com base no valor selecionado em uma outra

Se uma lista contém vários itens, você pode achar a lista fácil de usar se puder limitar os números de itens exibidos de uma vez. Por exemplo, ao invés de exibir todos os produtos do banco de dados Northwind, a lista Selecionar produto no formulário Editar Produtos exibe somente aqueles produtos na categoria que você escolheu na lista Selecionar categoria.

### Visão geral

Crie um procedimento de evento para a propriedade DepoisDeAtualizar da caixa de combinação "Selecionar categoria" que utiliza o método Requery para reconsultar a caixa de combinação "Selecionar produto". Inclua o campo Código da Categoria na consulta de origem da caixa de combinação "Selecionar produto". Defina o critério para o campo em uma expressão que limite a lista destes produtos cujo Código da Categoria identifique-se com o Código da Categoria do valor na caixa de combinação "Selecionar categoria".

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nomes</u>
Tabela	Categorias, Produtos
Consulta	Lista de Categoria, Lista Limitar Produtos
Formulário	Editar Produtos
Módulo	Ativar Controles

### Passo a passo

#### 1 Crie a consulta Lista de Categorias.

<u>Tabela</u>	<u>Campo (Classificar)</u>
Categorias	Nome da Categoria (Ascendente) Código da Categoria

#### 2 Crie a consulta Lista Limitar de Produtos.

<u>Tabela</u>	<u>Campo (Classificar)</u>	<u>Critério</u>
Produtos	Nome do Produto (Ascendente)	--
	Identificação do Produto	--
	Código da Categoria	Forms![Editar Produtos]! [Selecionar categoria]

#### 3 Crie o módulo Ativar Controles, se você não o fez ainda.

a) Na seção Declarações, digite:

```
Option Explicit
```

b) Crie a função AtivarControles.

#### 4 Modifique o formulário Editar Produtos.

- a) Adicione um cabeçalho e um rodapé de formulário. Defina a propriedade Altura do rodapé do formulário como 0.
- b) Crie a caixa de combinação "Selecionar categoria" no cabeçalho do formulário. Defina suas propriedades e crie um procedimento de evento como se segue.

<b>Propriedade</b>	<b>Valor</b>
Nome	Selecionar categoria
TipoDeOrigemDaLinha	Tabela/Consulta
OrigemDaLinha	Lista de Categorias
NúmeroDeColunas	2
LargurasDasColunas	;0 cm.
ColunaVinculada	2
LimitarAUmaLista	Sim
DepoisDeAtualizar	<u>[Procedimento de evento]</u>

- c) Crie a caixa de combinação "Selecionar produto" no cabeçalho do formulário. Separe a etiqueta anexada recortando-a e colando-a (assim ela não acende quando Selecionar produto está ativado ou desativado). Defina as propriedades da caixa de combinação "Selecionar produto" como a seguir.

<b>Propriedade</b>	<b>Valor</b>
Nome	Selecionar produto
TipoDeOrigemDaLinha	Tabela/Consulta
OrigemDaLinha	Lista Limitar Produtos
NúmeroDeColunas	2
LargurasDasColunas	;0 cm
ColunaVinculada	2
LimitarAUmaLista	Sim
Ativado	Não

---

**Dica** Em um formulário somente edição, você pode impedir usuários de, inadvertidamente, fazerem alterações no registro errado desativando os controles na seção Detalhe enquanto eles estiverem entrando o critério de procura e ativando os controles após eles terem encontrado o registro.

---

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"

Criar  
formulários

*Guia do Usuário,*  
capítulo 16,  
"Personalizando  
formulários"

Criar consultas

*Guia do Usuário,*  
capítulo 11,  
"Estruturando  
consultas seleção"

Fechar

Copiar

Imprimir

## Função AtivarControles()

```
Function EnableControls (WhichSection As String, State As Integer) As Integer
    ' Ativa ou desativa controles em uma seção especificada do formulário.
    Dim MyForm As Form
    Dim MyControl As Control
    Dim X As Integer, SelectedSection As Integer
    ' Utiliza formulário ativo. Se nenhum formulário está ativo, sai
    ' da função sem exibir mensagem de erro.
    On Error Resume Next
    Set MyForm = Screen.ActiveForm
    If Err Then
        EnableControls = False
        On Error GoTo 0
        Exit Function
    End If
    ' Define valores válidos para WhichSection.
    Select Case UCase$(WhichSection)
        Case "Cabeçalho do Formulário"
            SelectedSection = 1
        Case "Cabeçalho da Página"
            SelectedSection = 3
        Case "Detalhe"
            SelectedSection = 0
        Case "Rodapé da Página"
            SelectedSection = 4
        Case "Rodapé do Formulário"
            SelectedSection = 2
        Case Else
            MsgBox "Argumento Inválido", , "Controles Ativos"
            EnableControls = False
            Exit Function
    End Select
    ' Define estado para todos os controles na seção especificada.
    For X = 0 To MyForm.Count - 1
        Set MyControl = MyForm(X)
        If MyControl.Section = SelectedSection Then
            On Error Resume Next
            MyControl.Enabled = State
            On Error GoTo 0
        End If
    Next X
    EnableControls = True
End Function
```

Fechar

Copiar

Imprimir

### **Procedimento de evento DepoisDeAtualizar para a caixa de combinação "Selecionar categoria"**

```
Sub Select_Category_AfterUpdate ()
    ' Ativa e reconsulta a caixa de combinação "Selecionar produto".
    ' Desativa controles na seção Detalhe.
    Dim Tmp As Variant
    Me![Select Product].Enabled = True
    Me![Select Product].Requery
    Tmp = EnableControls("Detalhe", False)
End Sub
```

## Encontrando um registro com base em um valor que você selecionou de uma lista

Um meio rápido de encontrar um registro é selecionar um valor de uma lista. Por exemplo, o formulário Editar Produtos encontra o registro para o produto que você selecionou na caixa de combinação "Selecionar produto".

---

**Dica** Em um formulário somente para edição, você pode impedir os usuários de, inadvertidamente, fazerem alterações no registro errado desativando os controles na seção Detalhe enquanto eles estiverem entrando o critério de procura e ativando os controles após eles o terem entrado.

---

### Visão geral

Crie um procedimento de evento para a propriedade DepoisDeAtualizar da Caixa de combinação "Selecionar produto" que utiliza o comando ApplyFilter para encontrar o registro e uma função personalizada para ativar cada controle na seção Detalhe.

### Objetos utilizados neste exemplo

<b>Tipo</b>	<b>Nome</b>
Tabela	Produtos
Formulário	Editar Produtos
Módulo	Ativar Controles

### Passo a passo

**1 Crie as caixas de combinação "Selecionar categoria" e "Selecionar produto".**

**2 Crie o módulo Ativar Controles, se você não o fez ainda.**

a) Na seção Declarações, digite:

```
Option Explicit
```

b) Crie a função AtivarControles.

**3 Crie um procedimento de evento para a propriedade DepoisDeAtualizar da caixa de combinação "Selecionar produto".**

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"

Fechar

Copiar

Imprimir

## Procedimento de evento DepoisDeAtualizar para a caixa de combinação "Selecionar produto"

```
Sub Select_Product_AfterUpdate ()
    ' Encontrar registro para produto selecionado na caixa de combinação
    "Selecionar produto".
    ' Ativa controles na seção detalhe e desativa a caixa de texto
    "Identificação de produto".
    ' Vai para caixa de combinação "Identificação do fornecedor".
    Dim Tmp As Variant
    DoCmd ApplyFilter, "[Identificação do Produto] = Forms![Editar Produtos]!
[Selecionar produto]"
    Tmp = EnableControls("Detalhe", True)
    Me![Identificação do Produto].Enabled = False
    Me![Identificação do Fornecedor].SetFocus
End Sub
```

## Reconsultando uma lista após mover-se para outro registro

Quando você altera um valor em um campo que afeta os valores em uma lista, é essencial atualizar ou reconsultar a lista de modo que ela contenha os dados mais atuais. Por exemplo, quando você se move para outro registro no formulário Editar Produtos, a caixa de combinação "Selecionar produto" é reconsultada.

### Visão geral

Crie um procedimento de evento para a propriedade DepoisDeAtualizar do formulário que utiliza o método Requery para reconsultar a caixa de combinação "Selecionar produto".

### Objeto utilizado neste exemplo

<u>Tipo</u>	<u>Nome</u>
Formulário	Editar Produtos

### Passo a passo

► Crie um procedimento de evento na propriedade DepoisDeAtualizar do formulário Editar Produtos.

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"

Fechar

Copiar

Imprimir

### **Procedimento de evento DepoisDeAtualizar para o formulário Editar Produtos**

```
Sub Form_AfterUpdate ()  
    ' Reconsulta caixa de combinação "Selecionar produto".  
    Me![Selecionar produto].Requery  
End Sub
```

## Alterando a cor de um controle e exibindo uma mensagem com base no valor do controle

Quando o valor em um controle encontra um certo critério, você pode indicá-lo alterando sua cor e exibindo uma mensagem ao lado. No formulário Editar Produtos, por exemplo, quando as unidades em estoque ficam abaixo do nível de novos pedidos e não existem unidades pedidas, o texto da caixa de texto "Unidades em estoque" altera-se para vermelho e, além disso, uma mensagem é exibida.

### Visão geral

Crie uma etiqueta para exibir a mensagem. Crie um procedimento de evento para a propriedade NoAtual do formulário que utiliza a propriedade CorDeTexto da caixa de texto para definir a cor do texto e a propriedade Visível da etiqueta para exibir a mensagem. Crie procedimentos de evento que executem o sub Form Current nas propriedades DepoisDeAtualizar nos controles Unidades em Estoque, Unidades Pedidas, e Nível de Novos Pedidos.

### Objeto utilizado neste exemplo

<u>Tipo</u>	<u>Nome</u>
Formulário	Editar Produtos

### Passo a passo

**1 Crie uma etiqueta ao lado da caixa de texto "Unidades em estoque".** Defina suas propriedades como a seguir.

<u>Propriedade</u>	<u>Definição</u>
Nome	B_Estoque baixo
Legenda	unidades em estoque estão abaixo do nível de novos pedidos.
Visível	Não
CorDeTexto	128

**2 Crie procedimentos de evento para o formulário e controles como a seguir.**

<u>Formulário/ Controle</u>	<u>Propriedade</u>	<u>Definição</u>
Formulário Editar Produtos	NoAtual	[ <u>Procedimento de evento</u> ]
Caixa de texto "Unidades em estoque"	DepoisDeAtualizar	[ <u>Procedimento de evento</u> ]
Caixa de texto "Unidades pedidas"	DepoisDeAtualizar	[ <u>Procedimento de evento</u> ]
Caixa de texto "Nível de novos pedidos"	DepoisDeAtualizar	[ <u>Procedimento de evento</u> ]

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3,

Criar  
formulários

"Introduzindo o  
Access Basic," e  
capítulo 5,  
"Noções básicas  
do Access Basic"  
*Guia do Usuário*,  
capítulo 15,  
"Estruturando  
formulários"

Fechar

Copiar

Imprimir

### Procedimento de evento Atual para o formulário Editar Produtos

```
Sub Form_Current ()
    ' Se o valor na caixa de texto "Unidades em estoque" é menor que na
    ' caixa de texto "Novos pedidos" e o valor na caixa de texto "Unidades
pedidas" é 0, exiba
    ' o valor em Unidades em estoque em vermelho e exiba a etiqueta
B_Estoque baixo.
    If Me![Unidades em Estoque] < Me![Nível de Novos Pedidos] And Me!
[Unidades Pedidas] = 0 Then
        Me![Unidades em Estoque].ForeColor = 128
        Me![B_Estoque baixo].Visible = True
    ' Caso contrário, exiba valor na caixa de texto "Unidades em estoque" em
preto
    ' e oculte etiqueta B_Estoque baixo.
    Else
        Me![Unidades em Estoque].ForeColor = 0
        Me![B_Estoque baixo].Visible = False
    End If
End Sub
```

Fechar

Copiar

Imprimir

**Procedimento de evento DepoisDeAtualizar para caixa de texto "Unidades em estoque"**

```
Sub Units_In_Stock_AfterUpdate ()  
    ' Executa procedimento de evento sub Form_Current.  
    Form_Current  
End Sub
```

Fechar

Copiar

Imprimir

**Procedimento de evento DepoisDeAtualizar para a caixa de texto "Unidades pedidas"**

```
Sub Units_On_Order_AfterUpdate ()  
    ' Executa procedimento de evento sub Form_Current.  
    Form_Current  
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento de evento DepoisDeAtualizar para caixa de texto "Nível de novos pedidos"**

```
Sub Reorder_Level_AfterUpdate ()  
    ' Executa procedimento de evento sub Form_Current.  
    Form_Current  
End Sub
```

## **Destaques do relatório Vendas de Empregados**

O relatório Vendas de Empregados mostra como criar um relatório tabela de referência cruzada com cabeçalhos de coluna dinâmicos. Estas técnicas fazem o relatório Vendas de Empregados funcionar:

- Criando um relatório tabela de referência cruzada onde os cabeçalhos de coluna sejam dinâmicos.
- Definindo critérios de relatório em uma caixa de diálogo.

A técnica que melhora o relatório é:

- Imprimindo os critérios que você especificou na caixa de diálogo do relatório.

## Criando um relatório tabela de referência cruzada com cabeçalhos de coluna dinâmicos

Você pode variar o número de colunas a serem impressas em um relatório tabela de referência cruzada dependendo dos critérios que você insere quando o imprime. Por exemplo, no relatório Vendas de Empregados, nem todo empregado tem vendas toda semana, portanto o relatório só imprime colunas para aqueles empregados que tiveram vendas no período de tempo especificado. Este relatório também calcula totais para cada linha e coluna.

### Visão geral

Crie uma consulta tabela de referência cruzada para usar como origem dos registros do relatório. Na estrutura do relatório, crie caixas de texto independentes no cabeçalho de página (para cabeçalhos de coluna), na seção detalhe (para valores de tabela de referência cruzada), e no rodapé do relatório (para totais de coluna). Nas propriedades AoFormatar do cabeçalho de página e da seção detalhe, crie procedimentos de evento que preencham as caixas de texto com os cabeçalhos de coluna e os valores de tabela de referência cruzada e ocultem as caixas de texto não usadas. Na propriedade AoImprimir da seção detalhe, crie um procedimento de evento que calcule o total de linha, e o adicione à matriz de totais de coluna. Na propriedade AoImprimir do rodapé do relatório, crie um procedimento de evento que preencha as caixas de texto com os totais de coluna.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Empregados, Pedidos, Produtos
Consulta	Vendas de Empregados, Detalhes Adicionais do Pedido
Relatório	Vendas de Empregados

### Passo a passo

**1 Importe a consulta Detalhes Adicionais do Pedido de NWIND.MDB, se você já não o tiver feito.**

**2 Crie Vendas de Empregados, uma consulta tabela de referência cruzada.**

<u>Tabela/Consulta</u>	<u>Campo</u>	<u>Propriedades</u>
Tabela Empregados	Sobrenome	Total: Agrupar por Referência cruzada: Coluna
Tabela Produtos	Nome do produto	Total: Agrupar por Referência cruzada: Linha
Detalhes Adicionais do Pedido	Quantidade do pedido: Preço total	Total: Soma Referência cruzada: Valor
Tabela Pedidos	Data de envio	Total: Onde Critério: Entre [Forms]![Caixa de diálogo Valor de Empregados]![Data de Término]

---

**Observação** Se você observar o critério do campo Data de envio na consulta Vendas de Empregados, você verá que ele faz referência a controles na caixa de diálogo Vendas de Empregados. Os passos para modificar a consulta são abordados em Definir critérios de relatório em uma caixa de diálogo.

#### 4 Estructure o relatório Vendas de Empregados.

- Defina a propriedade OrigemDoRegistro do relatório como Vendas de Empregados
- Adicione um cabeçalho e um rodapé de relatório. Crie um título para o relatório no cabeçalho
- Crie 11 caixas de texto independentes no cabeçalho de página. Essas caixas de texto irão exibir os cabeçalhos de coluna. Defina a propriedade Nome da caixa de texto mais à esquerda como Head1, a propriedade Nome da próxima caixa de texto como Head2, e assim por diante até Head11.
- Crie 11 caixas de texto independentes na seção detalhe. (Crie uma caixa de texto para o cabeçalho de linha, nove caixas de texto para o número máximo de vendedores na Northwind Traders, e uma caixa de texto para o total de linha). Defina a propriedade Nome da caixa de texto mais à esquerda como Col1, a propriedade Nome da próxima caixa de texto como Col2, e assim por diante até Col11. Defina a propriedade Formato das caixas de texto Col2 a Col11 como Standard.
- Crie 11 caixas de texto no rodapé do relatório. Defina a propriedade Nome da caixa de texto mais à esquerda como Tot1 e sua propriedade OrigemDoControle como:
 

**= "Totais:"**

Defina a propriedade Nome das caixas de texto restantes, que exibem os totais de coluna, como sendo Tot2 a Tot11. Defina a propriedade Formato das caixas de texto de Tot2 até Tot11 como Standard.
- Na seção Geral do módulo RelatórioVendas de Empregados, declare as variáveis, crie a sub IniciaVars e a função xtabCnulls.
- Defina propriedades e crie procedimentos de evento como descrito na tabela seguinte.

<b>Relatório ou seção</b>	<b>Propriedade</b>	<b>Definição</b>
Relatório	AoAbrir	[ <u>Procedimento de evento</u> ]
	AoFechar	[ <u>Procedimento de evento</u> ]
Cabeçalho de relatório	AoFormatar	[ <u>Procedimento de evento</u> ]
Cabeçalho de página	AoFormatar	[ <u>Procedimento de evento</u> ]
Seção detalhe	ManterJunto	Sim
	AoFormatar	[ <u>Procedimento de evento</u> ]
	AoImprimir	[ <u>Procedimento de evento</u> ]
	AoRetirar	[ <u>Procedimento de evento</u> ]
Rodapé de relatório	AoImprimir	[ <u>Procedimento de evento</u> ]

#### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar consultas tabela de	<i>Guia do Usuário</i> , capítulo 12,

referência cruzada	"Consultas avançadas"
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios", e capítulo 21, "Estruturando relatórios"

Fechar

Copiar

Imprimir

### Seção declarações do módulo Relatório Vendas de Empregados

Option Compare Database 'Usa ordem do banco de dados em comparações de seqüências

Option Explicit

' Constante para o número máximo de colunas que a consulta Vendas de Empregados pode

' criar mais 1 para uma coluna Totais.

Const TOTCOLS = 11

' Variáveis para o objeto e recordset do banco de dados.

Dim RptDB As Database

Dim RptRS As Recordset

' Variáveis para o número de colunas, linhas e totais do relatório.

Dim IColCnt As Integer

Dim RgColTot(1 To TOTCOLS) As Long

Dim RptTotal As Long

Fechar

Copiar

Imprimir

### **Sub IniciaVars ()**

```
Sub IniciaVars ()
  Dim i As Integer
  ' Inicializa a variável RptTotal.
  RptTotal = 0
  ' Inicializa a matriz que armazena totais de coluna.
  For i = 1 To TOTCOLS
    RgColTot(i) = 0
  Next i
End Sub
```

Fechar

Copiar

Imprimir

### Função xtabCnulls ()

```
Function xtabCnulls (MyVal As Variant)
    ' Testa se um valor é nulo.
    If IsNull(MyVal) Then
        ' Se MyVal for nulo, define MyVal como 0.
        xtabCnulls = 0
    Else
        ' Caso contrário, retorna MyVal.
        xtabCnulls = MyVal
    End If
End Function
```

**Fechar****Copiar****Imprimir**

### Procedimento do evento Abrir para o relatório

```
Sub Report_Open (Cancel As Integer)
    ' Cria recordset de origem para o relatório usando critérios inseridos
    no formulário
    ' Caixa de Diálogo Vendas de Empregados.
    '
    Dim i As Integer
    Dim MyQuery As QueryDef
    ' Não abre o relatório se o formulário Caixa de Diálogo Vendas de
    Empregados não estiver carregado.
    If Not (EstáCarregado("Caixa de Diálogo Vendas de Empregados")) Then
        Cancel = True
        MsgBox "Para visualizar ou imprimir este relatório, você precisa
        abrir a Caixa de Diálogo Vendas de Empregados em modo Formulário.", 48,
        "Precisa abrir caixa de diálogo"
        Exit Sub
    End If
    ' Define variável do banco de dados como o banco de dados atual.
    Set RptDB = DBEngine.Workspaces(0).Databases(0)
    ' Abre QueryDef.
    Set MyQuery = RptDB.QueryDefs("Vendas de Empregados")
    ' Define parâmetros para a consulta com base em valores inseridos no
    formulário Caixa de Diálogo Vendas de Empregados.
    MyQuery.Parameters("Forms![Caixa de Diálogo Vendas de Empregados]![Data
    inicial]") = Forms![Caixa de Diálogo Vendas de Empregados]![Data inicial]
    MyQuery.Parameters("Forms![Caixa de Diálogo Vendas de Empregados]![Data
    de término]") = Forms![Caixa de Diálogo Vendas de Empregados]![Data de
    término]
    ' Abre o Recordset.
    Set RptRS = MyQuery.OpenRecordset()
    ' Se nenhum registro corresponder ao critério, exibe mensagem,
    ' fecha o recordset, e cancela o evento Abrir.
    If RptRS.RecordCount = 0 Then
        MsgBox "Nenhum registro corresponde ao critério que você inseriu.",
        48, "Nenhum registro encontrado"
        RptRS.Close
        Cancel = True
        Exit Sub
    End If
    ' Define uma variável para armazenar o número de colunas na consulta
    tabela de referência cruzada.
    IColCnt = RptRS.Fields.Count
End Sub
```



### **Procedimento do evento Fechar para o relatório**

```
Sub Report_Close ()  
    ' Fecha o recordset.  
    RptRS.Close  
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o cabeçalho do relatório**

```
Sub ReportHeader3_Format (Cancel As Integer, FormatCount As Integer)
    ' Move para o primeiro registro do recordset no início do relatório
    ' ou quando o relatório é reiniciado. (Um relatório é reiniciado quando
    ' você imprime um relatório a partir da janela Visualizar Impressão, ou
quando você volta
    ' a uma página anterior enquanto visualiza.)
    RptRS.MoveFirst
    ' Inicializa variáveis.
    IniciaVars
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Formatar para o cabeçalho de página**

```
Sub PageHeader0_Format (Cancel As Integer, FormatCount As Integer)
    Dim i As Integer
    ' Coloca cabeçalhos de coluna nas caixas de texto do cabeçalho da
    página.
    For i = 1 To IColCnt
        Me("Head" + Format$(i)) = RptRS(i - 1).Name
    Next i
    ' Torna a próxima caixa de texto disponível o cabeçalho Totais.
    Me("Head" + Format$(IColCnt + 1)) = "Totals"
    ' Oculta as caixas de texto não utilizadas do cabeçalho de página.
    For i = (IColCnt + 2) To TOTCOLS
        Me("Head" + Format$(i)).Visible = False
    Next i
End Sub
```

Fechar

Copiar

Imprimir

### Procedimento do evento Formatar para a seção detalhe

```
Sub Detail1_Format (Cancel As Integer, FormatCount As Integer)
    ' Coloca valores em caixas de texto e oculta caixas de texto não
    utilizadas.
    '
    Dim i As Integer
    ' Verifica se não é o fim do recordset.
    If Not RptRS.EOF Then
        ' Se FormatCount for 1, coloca valores do recordsSet nas caixas de
        texto
        ' da seção detalhe.
        If Me.FormatCount = 1 Then
            For i = 1 To IColCnt
                ' Converte valores nulos para 0.
                Me("Col" + Format$(i)) = xtabCnulls(RptRS(i - 1))
            Next i
            ' Oculta caixas de texto não utilizadas na seção detalhe.
            For i = IColCnt + 2 To TOTCOLS
                Me("Col" + Format$(i)).Visible = False
            Next i
            ' Move para o próximo registro do recordset.
            RptRS.MoveNext
        End If
    End If

End Sub
```

Fechar

Copiar

Imprimir

### Procedimento do evento Imprimir para a seção detalhe

```
Sub Detail1_Print (Cancel As Integer, PrintCount As Integer)
    Dim i As Integer
    Dim RowTotal As Long
    ' Se PrintCount for 1, inicializa a variável rowTotal.
    ' Adiciona aos totais de coluna.
    If Me.PrintCount = 1 Then
        RowTotal = 0
        For i = 2 To IColCnt
            ' Começando pela coluna 2 (primeira caixa de texto com valor de
            ' tabela de referência cruzada),
            ' computa o total da linha atual na seção detalhe.
            RowTotal = RowTotal + Me("Col" + Format$(i))
            ' Adiciona valor de tabela de referência cruzada ao total da
            ' coluna atual.
            RgColTot(i) = RgColTot(i) + Me("Col" + Format$(i))
        Next i
        ' Coloca o total de linha na caixa de texto da seção detalhe.
        Me("Col" + Format$(IColCnt + 1)) = RowTotal
        ' Adicionar o total de linha da linha atual ao montante final.
        RptTotal = RptTotal + RowTotal
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Sair para a seção detalhe**

```
Sub Detail1_Retreat ()  
    ' Sempre volta ao registro anterior quando a seção detalhe se retira.  
    RptRS.MovePrevious  
End Sub
```

Fechar

Copiar

Imprimir

### Procedimento do evento Imprimir para o cabeçalho do relatório

```
Sub ReportFooter4_Print (Cancel As Integer, PrintCount As Integer)
    Dim i As Integer
    ' Coloca totais de coluna nas caixas de texto do rodapé do relatório.
    ' Começa na coluna 2 (primeira caixa de texto com valor de tabela de
referência cruzada).
    For i = 2 To IColCnt
        Me("Tot" + Format$(i)) = RgColTot(i)
    Next i
    ' Coloca o montante final na caixa de texto do rodapé do relatório.
    Me("Tot" + Format$(IColCnt + 1)) = RptTotal
    ' Oculta as caixas de texto não utilizadas do rodapé do relatório.
    For i = IColCnt + 2 To TOTCOLS
        Me("Tot" + Format$(i)).Visible = False
    Next i
End Sub
```

## **Destaques da caixa de diálogo Vendas de Empregados**

O formulário Caixa de Diálogo Vendas de Empregados mostra a você como pedir ao usuário critério para o relatório Vendas de Empregados. Esta técnica faz a Caixa de Diálogo Vendas de Empregados funcionar:

- Definindo critérios de relatório em uma caixa de diálogo.

## Definindo critérios de relatório em uma caixa de diálogo

Em vez de definir o critério para um relatório em sua consulta de origem, você pode exibir uma caixa de diálogo que peça a pessoa que está usando o formulário para inserir o critério logo antes de imprimir. Um relatório é muito mais flexível quando seus critérios são definidos em tempo de execução porque os usuários têm controle sobre a gama de dados que imprimem. No formulário Caixa de Diálogo Vendas de Empregados, um usuário especifica as datas inicial e de término como critério de seleção.

### Visão geral

Crie uma caixa de diálogo que contenha os controles para inserção do critério do relatório. Adicione os botões Visualizar, Imprimir e Cancelar. Na consulta de origem do relatório, defina os parâmetros da consulta e o critério para o campo de data usando uma expressão que inclua o objeto Forms, o nome do formulário, e os nomes dos controles. No procedimento de evento para a propriedade AoAbrir do relatório, especifique os parâmetros para a DefConsulta.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Caixa de Diálogo Vendas de Empregados
Relatório	Vendas de Empregados
Módulo	Está carregado

### Passo a passo

#### **1 Crie o relatório Vendas de Empregados.**

#### **2 Crie o formulário Caixa de Diálogo Vendas de Empregados.**

a) Crie um formulário independente, e defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Definição</u>
Legenda	Vendas de Empregados
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
MenuAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
JanelaRestrita	Sim
EstiloDaBorda	Fina
BotãoMin	Não
BotãoMax	Não

b) Crie duas caixas de texto independentes. Defina suas propriedades Formatar como Data Média, e defina as propriedades Legenda das etiquetas e as propriedades Nome das caixas de texto como a seguir.

<u>Legenda para etiqueta</u>	<u>Nome para caixa de texto</u>
Insira uma data	Data inicial

inicial:

Insira uma data    Data de término  
de término:

- c) Crie os seguintes botões de comando, definindo suas propriedades Nome, Legenda e AoClicar como a seguir. Além disso, defina a propriedade Padrão do botão Visualizar como Sim.

<b>Nome e Legenda</b>	<b>AoClicar</b>
Visualizar	[ <u>Procedimento de evento</u> ]
Imprimir	[ <u>Procedimento de evento</u> ]
Cancelar	[ <u>Procedimento de evento</u> ]

- d) Na seção Declarações do módulo Formulário.Caixa de Diálogo Vendas de Empregados, digite:

```
Option Explicit
```

```
Const ERR_REP_CANCELED = 2501
```

Se você usar os Assistentes de Controle para criar os botões de comando, a Assistente adicionará automaticamente a instrução Option Explicit.

### 3 Modifique a consulta Vendas de Empregados.

- a) Na caixa Critério do campo Data de envio, digite:

**Entre [Forms]![Caixa de Diálogo Vendas de Empregados]![Data inicial] E [Forms]![Caixa de Diálogo Vendas de Empregados]![Data de término]**

- b) Defina os parâmetros da consulta como a seguir.

<b>Parâmetro</b>	<b>Tipo de Dados</b>
Forms!Caixa de Diálogo Vendas de Empregados!Data inicial	Data/Hora
Forms!Caixa de Diálogo Vendas de Empregados!Data de término	Data/Hora

### 4 Crie o módulo Está Carregado, se já não o tiver feito.

- a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

- b) Crie a função EstáCarregado.

### 5 Modifique o procedimento do evento Abrir do relatório Vendas de Empregados

#### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao

	Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 14, "Fundamentos de formulários", e capítulo 15, "Estruturando formulários"
Criar consultas	<i>Guia do Usuário</i> , capítulo 12, "Consultas avançadas"

Fechar

Copiar

Imprimir

### Procedimento do evento Clicar para o botão de comando Visualizar

```
Sub Visualizar_Click ()
' Visualiza relatório.
On Error GoTo Erro_Visualizar_Click
    Dim DocName As String
    DocName = "Vendas de Empregados"
    DoCmd OpenReport DocName$, 2
    ' Usado pelo Soluções para abrir o tópico Mostre me para o relatório.
    If EstáCarregado("Soluções") Then
        If Forms![Soluções]![Mostre-me] Then
            SendKeys "{F1}", True
        End If
    End If
Sai_Visualizar_Click:
    Exit Sub
Erro_Visualizar_Click:
    If Err = ERR_RPT_CANCELED Then
        Resume Sai_Visualizar_Click
    Else
        MsgBox Error$
        Resume Sai_Visualizar_Click
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento do evento Clicar para o botão de comando Imprimir**

```
Sub Imprimir_Click ()
' Imprime relatório.
On Error GoTo Erro_Imprimir_Click
    Dim DocName As String
    DocName = "Vendas de Empregados"
    DoCmd OpenReport DocName$, 0
Sai_Imprimir_Click:
    Exit Sub
Erro_Imprimir_Click:
    If Err = ERR_RPT_CANCELED Then
        Resume Sai_Imprimir_Click
    Else
        MsgBox Error$
        Resume Sai_Imprimir_Click
    End If
End Sub
```



### **Procedimento do evento Clicar para o botão de comando Cancelar**

```
Sub Cancelar_Click ()  
    ' Fecha formulário. (Código criado pelo Assistente para Botão de Comando.)  
    On Error GoTo Erro_Cancelar_Click  
        DoCmd Close  
Sai_Cancelar_Click:  
    Exit Sub  
Erro_Cancelar_Click:  
    MsgBox Error$  
    Resume Sai_Cancelar_Click  
End Sub
```

Fechar

Copiar

Imprimir

### Função EstáCarregado ()

```
Function EstáCarregado (MyFormName)
    ' Determina se um formulário está carregado.
    Const FORM_DESIGN = 0
    Dim i As Integer
    EstáCarregado = False
    For i = 0 To Forms.Count - 1
        If Forms(i).FormName = MyFormName Then
            If Forms(i).CurrentView <> FORM_DESIGN Then
                EstáCarregado = True
                Exit Function ' Sai da função uma vez que o formulário tenha
                sido encontrado.
            End If
        End If
    Next
End Function
```

Fechar

Copiar

Imprimir

### Procedimento do evento Abrir para o relatório Vendas de Empregados

Os comentários e o código que você precisa adicionar ao procedimento de evento para assegurar que o Caixa de Diálogo Vendas de Empregados esteja carregado, para definir parâmetros para a DefConsulta, e para exibir uma mensagem se nenhum registro coincidir com o critério são exibidos em vermelho.

```
Sub Report_Open (Cancel As Integer)
    ' Cria recordset de origem para o relatório usando critérios inseridos no
    formulário
    ' Caixa de Diálogo Vendas de Empregados.
    '
    Dim i As Integer
    Dim MyQuery As QueryDef
    ' Não abre o relatório se o formulário Caixa de Diálogo Vendas de
    Empregados não estiver carregado.
    If Not (EstáCarregado("Caixa de Diálogo Vendas de Empregados")) Then
        Cancel = True
        MsgBox "Para visualizar ou imprimir este relatório, você precisa abrir
a Caixa de Diálogo Vendas de Empregados em modo Formulário.", 48, "Precisa
abrir caixa de diálogo"
        Exit Sub
    End If
    ' Define variável do banco de dados como o banco de dados atual.
    Set RptDB = DBEngine.Workspaces(0).Databases(0)
    ' Abre QueryDef.
    Set MyQuery = RptDB.QueryDefs("Vendas de Empregados")
    ' Define parâmetros para a consulta com base em valores inseridos no
    formulário Caixa de Diálogo Vendas de Empregados.
    MyQuery.Parameters("Forms![Caixa de Diálogo Vendas de Empregados]![Data
inicial]") = Forms![Caixa de Diálogo Vendas de Empregados]![Data inicial]
    MyQuery.Parameters("Forms![Caixa de Diálogo Vendas de Empregados]![Data de
término]") = Forms![Caixa de Diálogo Vendas de Empregados]![Data de término]
    ' Abre o Recordset.
    Set RptRS = MyQuery.OpenRecordset()
    ' Se nenhum registro corresponder ao critério, exibe mensagem,
    ' fecha o recordset, e cancela o evento Abrir.
    If RptRS.RecordCount = 0 Then
        MsgBox "Nenhum registro corresponde ao critério que você inseriu.",
48, "Nenhum registro encontrado"
        RptRS.Close
        Cancel = True
        Exit Sub
    End If
    ' Define uma variável para armazenar o número de colunas na consulta
    tabela de referência cruzada.
    IColCnt = RptRS.Fields.Count
End Sub
```

## Imprimindo critério a partir da caixa de diálogo Critério de um relatório no próprio relatório

Para aumentar a funcionalidade de um relatório, você pode imprimir o critério que você digitou na caixa de diálogo de impressão do relatório no próprio relatório. Por exemplo, os valores que você insere nas caixas de texto Data inicial e Data de término da Caixa de Diálogo Vendas de Empregados aparecem no cabeçalho do relatório Vendas de Empregados.

### Visão geral

Crie uma caixa de texto independente no relatório. Defina sua propriedade OrigemDoControle com uma expressão que retorne o valor inserido nos controles da caixa de diálogo de critério do relatório.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Caixa de Diálogo Vendas de Empregados
Relatório	Vendas de Empregados

### Passo a passo

#### **1 Crie o relatório Vendas de Empregados.**

#### **2 Crie a Caixa de Diálogo Vendas de Empregados.**

**3 Modifique o relatório Vendas de Empregados.** Crie uma caixa de texto independente no cabeçalho do relatório. Defina sua propriedade Nome como Intervalo e sua propriedade OrigemDoControle como:

**= "De " & Formato(Formulários![Caixa de Diálogo Vendas de Empregados]![Data inicial], "dd-mmm-aa") & " a " & Formato(Formulários![Caixa de Diálogo Vendas de Empregados]![Data de término], "dd-mmm-aa")**

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar expressões	<i>Guia do Usuário</i> , capítulo 23, "Usando expressões em relatórios"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios", e capítulo 21, "Estruturando relatórios"

## **Destaques do formulário Inserir ou Editar Produtos**

O formulário Inserir ou Editar Produtos mostra como entrar ou editar registros no mesmo formulário. As técnicas abaixo fazem o formulário Inserir ou Editar Produtos funcionar:

- Alterando as barras de ferramentas quando você alterna entre entrar novos registros e editar registros.
- Controlando quando você vai para um novo registro.
- Atualizando a lista na caixa de combinação "Selecionar produto" depois de ter alternado para outro registro. (A técnica é a mesma que aquela no formulário Editar Produtos.)

Estas técnicas ressaltam o formulário:

- Anexando uma nova categoria à lista de categoria.
- Usando maiúscula na primeira letra do nome do produto automaticamente.
- Alterando a cor de uma seção e de seus controles em tempo de execução.

## Criando um formulário no qual você pode entrar ou editar dados

Você pode ter fácil acesso aos dados criando um formulário no qual você pode alternar rapidamente entre a entrada de novos registros e a edição de registros já existentes. Por exemplo, o formulário Inserir ou Editar Produtos abre-se no modo entrada de dados. Para editar registros, clique no botão "Editar dados" na barra de ferramentas. Para alternar de volta para a entrada de registros, clique no botão "Entrar dados" na barra de ferramentas.

### Visão geral

Estruture o formulário de modo que ele abra no modo entrada de dados. No cabeçalho do formulário, adicione as caixas de combinação nas quais você entra os dados procurados, e desative-os. Crie funções para dois botões personalizados da barra de ferramentas, "Editar dados" e "Entrar dados". Nas funções, defina a propriedade PermitirEdição do formulário de modo que, nos eventos AoAtivar e AoOcultar do formulário, você possa utilizar o valor da propriedade como um indicador para ocultar e exibir a barra de ferramentas correta. Crie macros para executar as funções. Crie duas barras de ferramentas, uma que inclui o botão "Editar dados" e que é exibida quando o formulário está no modo entrada de dados, e outra que inclui o botão "Entrar dados" e que é exibida quando o formulário está no modo edição.

### Objetos utilizados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Produtos
Formulário	Inserir ou Editar Produtos
Macro	Inserir ou Editar Produtos
Módulo	Ativar Controles, Botões Inserir ou Editar Produtos

### Passo a passo

#### 1 Estruture o formulário Inserir ou Editar Produtos.

a) Defina as seguintes propriedades de formulário.

<b>Propriedade</b>	<b>Definição</b>
OrigemDoRegistro	Produtos
Legenda	Inserir ou Editar Produtos
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
EdiçãoPadrão	Entrada de dados
MenuDeAtalho	Não
BarraDeRolagem	Nenhum
ExibirSeletor	Não

b) Na seção Detalhe, adicione os campos da tabela Produtos. Utilize uma caixa de verificação para o campo Descontinuado e caixas de combinação para os campos Identificação do Fornecedor e Código da Categoria. Use caixas de texto para todos os outros campos.

c) Para impedir que as etiquetas se acendam quando os controles na seção Detalhe estiverem ativados e desativados, separe as etiquetas anexadas dos controles recortando-as e depois colando-as.

d) Faça que a caixa de texto "Identificação de produto" seja apenas para exibição, porque é um campo Contador, definindo as seguintes propriedades:

<b>Propriedade</b>	<b>Valor</b>
Ativado	Não
CorDoFundo	(mesma cor da CorDoFundo da seção Detalhe)
EstiloDaBorda	Limpar
EstiloDaFonte	Negrito

## **2 Adicione um cabeçalho de formulário e crie as caixas de combinação**

**"Selecionar categoria" e "Selecionar produto"**. O procedimento para a criação de caixas de combinação foi escrito para o formulário Editar Produtos, então você precisará fazer as seguintes alterações.

a) Defina a propriedade OrigemDaLinha da caixa de combinação "Selecionar produto" como:

```
SELECT DISTINCTROW Produtos.[Nome do Produto], Produtos.[Identificação do Produto], Produtos.[Código da Categoria]
FROM Produtos
WHERE (Produtos.[Código da Categoria]=[Forms]![Inserir ou Editar Produtos]![Selecionar categoria])
ORDER BY Produtos.[Nome do Produto];
```

b) Crie um procedimento de evento na propriedade DepoisDeAtualizar da caixa de combinação "Selecionar produto".

c) Defina a propriedade Ativado de ambas as caixas de combinação como Não.

d) Para impedir que acendam quando as caixas de combinação estiverem ativadas ou desativadas, separe as etiquetas anexadas de ambas as caixas de combinação recortando as etiquetas e colando.

## **3 Crie o módulo Ativar Controles, se você ainda não o fez.**

a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

b) Crie a função AtivarControles.

## **4 Crie o módulo Botões Inserir ou Editar Produtos.**

a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

b) Crie a função EditarDados.

c) Crie a função InserirDados.

## **5 Crie as seguintes macros em um grupo de macro chamado Inserir ou Editar Produtos.**

<b>Nome da Macro</b>	<b>Ação</b>	<b>Argumento</b>
Entrar Dados	ExecutarCódigo	InserirDados()
Editar Dados	ExecutarCódigo	EditarDados()

## **6 Crie barras de ferramentas personalizadas.**

a) Crie uma barra de ferramenta, e nomeie-a Inserir ou Editar Produtos 1.

b) Inclua estes botões incorporados na barra de ferramentas:

Novo    Recor    Copia    Colar    Desfa    Desfazer  
tar       r       zer       campo/registro atual



c) Adicione um botão personalizado na barra de ferramentas.

<b>Macro</b>	<b>Texto do botão</b>	<b>Descrição de botão</b>
Inserir ou Editar Produtos.Editar Dados	Editar Dados	Editar Dados

- d) Crie uma barra de ferramentas e nomeie-a Inserir ou Editar Produtos 2.  
e) Inclua estes botões incorporados na barra de ferramentas: Recortar, Copiar, Colar, Desfazer, e Desfazer campo/registo atual.  
f) Adicione um botão personalizado na barra de ferramentas.

<b>Macro</b>	<b>Texto do botão</b>	<b>Descrição de botão</b>
Inserir ou Editar Produtos.Entrar Dados	Entrar Dados	Entrar Dados

**Observação** As barras de ferramentas personalizadas para os formulários no Soluções também incluem os botões "Modo Estrutura" e "Modo Formulário", dessa forma você pode alternar facilmente entre os modos enquanto estiver olhando estes exemplos. Em muitos casos, você não irá incluir estes botões nas barras de ferramentas que você criará para seus aplicativos.

### 7 Crie procedimentos de evento para as seguintes propriedades de formulário.

<b>Propriedade</b>	<b>Definição</b>
AoAbrir	[ <u>Procedimento de evento</u> ]
AoAtivar	[ <u>Procedimento de evento</u> ]
AoDesativar	[ <u>Procedimento de evento</u> ]

### 8 Na seção Declarações do módulo Inserir ou Editar Produtos, digite:

Option Explicit

#### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários," e capítulo 16, "Personalizando formulários"

Criar macros	<i>Guia do Usuário</i> , capítulo 24, "Fundamentos de macros"
Criar barras de ferramentas	<i>Guia do Usuário</i> , Apêndice B, "Trabalhando com barras de ferramentas"

Fechar

Copiar

Imprimir

### **Procedimento de evento DepoisDeAtualizar para a caixa de combinação "Selecionar produto"**

```
Sub Select_Product_AfterUpdate ()
    ' Encontra registro para produto selecionado na caixa de combinação
    "Selecionar produto",
    ' ativa controles na seção Detalhe, desativa caixa de texto
    "Identificação de produto",
    ' e vai para a caixa de combinação "Identificação do fornecedor".
    Dim Tmp As Variant
    DoCmd ApplyFilter , "[Identificação de produto] = Forms![Inserir ou
    Editar Produtos]![Selecionar produto]"
    Tmp = EnableControls("Detalhe", True)
    Me![Identificação do Produto].Enabled = False
    Me![Identificação do Fornecedor].SetFocus
End Sub
```

Fechar

Copiar

Imprimir

## Função EditarDados ()

```
Function EditarDados () As Integer
    ' Código para botão "Editar dados" na barra de ferramentas Inserir ou
    Editar Produtos 1.
    '
    On Error GoTo EditData_Err
    Dim MyForm As Form
    Dim ComboBox1 As Control
    Dim ComboBox2 As Control
    Dim EditColor As Long
    Dim Tmp As Variant

    Set MyForm = Forms![Inserir ou Editar Produtos]
    Set ComboBox1 = Forms![Inserir ou Editar Produtos]![Selecionar categoria]
    Set ComboBox2 = Forms![Inserir ou Editar Produtos]![Selecionar produto]
    EditColor = 16776960

    ' Define a propriedade PermitirEdição como Falsa para indicar modo
    edição.
    ' Define a propriedade EdiçãoPadrão como Impossível adicionar registros
    (4).
    MyForm.AllowEditing = False
    MyForm.DefaultEditing = 4
    ' Alterna barras de ferramentas.
    DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_NO
    DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_WHERE_APPROP
    ' Ativa e move para a primeira caixa de combinação.
    ComboBox1.Enabled = True
    ComboBox1.SetFocus

    ' Desativa controles na seção Detalhe.
    Tmp = EnableControls("Detalhe", False)
    Exit Function
EditData_Err:
    MsgBox Error$
    Exit Function
End Function
```

Fechar

Copiar

Imprimir

## Função InserirDados ()

```
Function InserirDados () As Integer
    ' Código para o botão "Entrar dados" na barra de ferramentas Inserir ou
    Editar Produtos 2.
    '
    On Error GoTo EnterData_Err
    Dim MyForm As Form
    Dim ComboBox1 As Control
    Dim ComboBox2 As Control
    Dim EnterColor As Long
    Dim Tmp As Variant

    Set MyForm = Forms![Inserir ou Editar Produtos]
    Set ComboBox1 = Forms![Inserir ou Editar Produtos]![Selecionar categoria]
    Set ComboBox2 = Forms![Inserir ou Editar Produtos]![Selecionar produto]
    EnterColor = 8421376

    ' Define propriedade PermitirEdição como Verdadeira para indicar modo
    entrada de dados.
    ' Define propriedade EdiçãoPadrão como entrada de dados (1).
    MyForm.AllowEditing = True
    MyForm.DefaultEditing = 1
    ' Ativa controles na seção Detalhe e desativa caixa de texto
    "Identificação de produto".
    Tmp = EnableControls("Detalhe", True)
    MyForm![Identificação do Produto].Enabled = False
    ' Vai para um novo registro e move para a caixa de combinação
    "Identificação do fornecedor".
    DoCmd GoToRecord , , A_NEWREC
    MyForm![Identificação do Fornecedor].SetFocus
    ' Alterna barras de ferramentas.
    DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_NO
    DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_WHERE_APPROP
    ' Desativa caixas de combinação no cabeçalho do formulário.
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False

    Exit Function
EnterData_Err:
    MsgBox Error$
    Exit Function
End Function
```



### **Procedimento de evento Abrir para o formulário Inserir ou Editar Produtos**

```
Sub Form_Open (Cancel As Integer)
    ' Define a propriedade PermitirEdição como Verdadeira quando o
    formulário abre.
    Me.AllowEditing = True
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento de evento AoAtivar para o formulário Inserir ou Editar Produtos**

```
Sub Form_Activate ()
    ' Oculta barra de ferramentas Modo Formulário.
    ' Usa a definição de propriedade PermitirEdição para determinar qual
    barra de ferramentas
    ' exibir.
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_NO
    If Me.AllowEditing = True Then
        DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_YES
    Else
        DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_YES
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento de evento AoDesativar para o formulário Inserir ou Editar Produtos**

```
Sub Form_Deactivate ()
    ' Usa a definição de propriedade PermitirEdição para determinar qual
    barra de ferramentas
    ' ocultar.
    ' Exibe barra de ferramentas Modo Formulário.
    If Me.AllowEditing = True Then
        DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_NO
    Else
        DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_NO
    End If
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_WHERE_APPROP
End Sub
```

## Controlando quando você vai para um novo registro

Ao invés de avançar automaticamente para um novo registro quando pressionar TAB ou ENTER no último campo em um formulário, você pode ficar em um registro até querer prosseguir. Por exemplo, quando entrar dados no formulário Inserir ou Editar Produtos, você pode verificar que a informação que você digitou está correta e, então, depois avançar para um novo registro clicando no botão "Novo" na barra de ferramentas.

### Visão geral

Remova os botões de navegação do formulário. Na seção Detalhe, crie um botão transparente, e defina um procedimento de evento em sua propriedade AoEntrar que utilize o método SetFocus para mover o foco para a caixa de combinação "Identificação do fornecedor". Posicione o botão no final da ordem de tabulação.

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nome</u>
Tabela	Produtos
Formulário	Inserir ou Editar Produtos

### Passo a passo

**1 Defina a propriedade do formulário BotõesDeNavegação como Não.**

**2 Na seção Detalhe, crie o botão de comando "Vá para a identificação do fornecedor".**

a) Defina suas propriedades e crie um procedimento de evento como a seguir.

<u>Propriedade</u>	<u>Definição</u>
Nome	Vá para a identificação do fornecedor
Transparente	Sim
AoEntrar	[ <u>Procedimento de evento</u> ]

b) Para impedir usuários de, inadvertidamente, clicarem no botão de comando, oculte-o atrás da caixa de combinação "Identificação do fornecedor". (Faça o botão de comando do mesmo tamanho que a caixa de combinação, coloque-o na parte superior da caixa de combinação e, então, selecione Enviar para Trás no menu Formatar.)

**3 Edite a ordem de tabulação do formulário.** Certifique-se de que Identificação do Fornecedor seja o primeiro campo na lista e Vá para a identificação do fornecedor e Identificação do Produto sejam os dois últimos campos na lista.

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos para evento	<i>Criando Aplicativos,</i> capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário,</i> capítulo 15, "Estruturando

Criar barras de  
ferramentas

formulários"  
*Guia do Usuário*,  
Apêndice B,  
"Trabalhando com  
barras de  
ferramentas"

## **Destaques do formulário Adicionar Categoria**

O formulário Adicionar Categoria é aberto do formulário Inserir ou Editar Produtos quando você adiciona uma nova categoria à caixa de combinação "Código da Categoria". Esta técnica faz o formulário Adicionar Categoria funcionar:

- Adicionando uma nova categoria à lista de categorias.

## Adicionando um novo registro à uma lista

Ao inserir ou editar registros, pode acontecer de você precisar adicionar um novo item à lista em uma caixa de combinação. Por exemplo, no formulário Inserir ou Editar Produtos, você pode adicionar uma nova categoria à lista digitando o nome na caixa de combinação. O formulário exibe uma caixa de mensagem perguntando-lhe se quer adicionar um novo registro. Se você escolhe Sim, ela exibe um formulário no qual você entra a informação para a nova categoria.

### Visão geral

Crie um procedimento de evento para a propriedade QuandoForaDaLista da caixa de combinação "Código da Categoria" no formulário Inserir ou Editar Produtos. (Este procedimento de evento exibe uma caixa de diálogo na qual você pode confirmar que quer adicionar uma nova categoria; se você escolhe Sim, ela abre o formulário Adicionar Categoria). Crie um procedimento de evento para o evento AoFechar do formulário Adicionar Categoria que utiliza o método Requery para reconsultar a caixa de combinação "Código da Categoria" no formulário Inserir ou Editar Produtos.

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nomes</u>
Tabela	Produtos, Categorias
Formulário	Inserir ou Editar Produtos, Adicionar Categoria
Módulo	Está Carregado

### Passo a passo

**1 No formulário Inserir ou Editar Produtos, crie um procedimento de evento na propriedade QuandoForaDaLista da caixa de combinação "Código da Categoria".**

**2 Crie o módulo EstáCarregado, se você ainda não o fez.**

a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

b) Crie a função EstáCarregado.

**3 Estructure o formulário Adicionar Categoria.**

a) Defina as propriedades do formulário e crie procedimentos de evento como a seguir.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	Categorias
Legenda	Adicionar Categoria
ModoPadrão	Formulário Simples
VisualizaçãoAutorizada	Formulário
EdiçãoPadrão	Entrada de Dados
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
RedimensionamentoAutomático	Não
AutoCentrar	Sim

AoDescarregar	<a href="#">[Procedimento de evento]</a>
AoAtivar	<a href="#">[Procedimento de evento]</a>
AoDesativar	<a href="#">[Procedimento de evento]</a>

b) Na seção Detalhe, adicione os campos Nome da Categoria, Descrição, e Imagem da tabela Categorias.

**4 Na seção Declarações do módulo Adicionar Categoria, digite:**

Option Explicit

**Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para evento	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários," e capítulo 19, "Utilizando imagens, gráficos e outros objetos"

Fechar

Copiar

Imprimir

## Procedimento de evento QuandoForaDaLista para a caixa de combinação "Código da Categoria"

```
Sub Category_ID_NotInList (NewData As String, Response As Integer)
    ' Adiciona uma nova categoria digitando um nome na
    ' caixa de combinação "Nome da Categoria".

    Dim NewCategory As Integer, TruncateName As Integer, Title As String,
    MsgBox As Integer
    Const MB_OK = 0
    Const MB_YESNO = 4
    Const MB_ICONQUESTION = 32
    Const MB_ICONEXCLAMATION = 64
    Const MB_DEFBUTTON1 = 0, IDYES = 6, IDNO = 7

    ' Exibe caixa de mensagem perguntando se o usuário quer adicionar uma
    ' nova categoria.
    Title = "Categoria fora da lista"
    MsgBox = MB_YESNO + MB_ICONQUESTION + MB_DEFBUTTON1
    NewCategory = MsgBox("Você deseja adicionar uma nova categoria?",
    MsgBox, Title)

    If NewCategory = IDYES Then
        ' Remova o novo nome da caixa de combinação "Código da Categoria" assim
        ' o controle pode ser requerido quando o usuário retornar ao
        formulário.
        DoCmd DoMenuItem A_FORMBAR, A_EDIT, A_UNDOFIELD, , A_MENU_VER20
        ' Exibe caixa de mensagem e ajusta comprimento do valor entrado na
        ' caixa de combinação "Código da Categoria".
        Title = "Nome muito longo"
        MsgBox = MB_OK + MB_ICONEXCLAMATION
        If Len(NewData) > 15 Then
            TruncateName = MsgBox("Os nomes das categorias não podem ter mais do
            que 15 caracteres. O nome que você inseriu será truncado.", MsgBox,
            Title)
            NewData = Left$(NewData, 15)
        End If
        ' Abre formulário Adicionar Categoria.
        DoCmd OpenForm "Adicionar Categoria", A_NORMAL, , , A_ADD
        Forms![Adicionar Categoria]![Nome da Categoria] = NewData
        ' Continua sem exibir mensagem de erro padrão.
        Response = DATA_ERRCONTINUE
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento de evento Descarregar para o formulário Adicionar Categoria**

```
Sub Form_Unload (Cancel As Integer)
    ' Se o formulário Inserir ou Editar Produtos está carregado,
    ' seleciona-o, reconsulta a caixa de combinação "Código da Categoria",
    ' e define valor da caixa de combinação "Código da Categoria".
    Dim MyControl As Control
    If IsLoaded("Inserir ou Editar Produtos") Then
        Set MyControl = Forms![Inserir ou Editar Produtos]![Código da
Categoria]
        DoCmd SelectObject A_FORM, "Inserir ou Editar Produtos"
        MyControl.Requery
        MyControl = Me![Código da Categoria]
    End If
End Sub
```

Fechar

Copiar

Imprimir

### **Procedimento de evento Ativado para o formulário Adicionar Categoria**

```
Sub Form_Activate ()
    ' Exibe barra de ferramentas do formulário Adicionar Categoria.
    '
    ' Oculta barra de ferramentas Modo Formulário.
    ' Exibe barra de ferramentas Modo Formulário Personalizado.
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_NO
    DoCmd ShowToolbar "Modo Formulário Personalizado", A_TOOLBAR_YES
    ' Se formulário Inserir ou Editar Produtos está carregado, oculte sua
    barra de ferramentas atual.
    If IsLoaded("Inserir ou Editar Produtos") Then
        If Forms![Inserir ou Editar Produtos].AllowEditing = True Then
            DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_NO
        Else
            DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_NO
        End If
    End If
End Sub
```



### **Procedimento de evento Desativado para o formulário Adicionar Categoria**

```
Sub Form_Deactivate ()
    ' Oculta barra de ferramentas do formulário Adicionar Categoria.
    '
    ' Oculta barra de ferramentas Modo Formulário personalizado.
    ' Exibe barra de ferramentas incorporada Modo Formulário.
    DoCmd ShowToolbar "Modo Formulário Personalizado", A_TOOLBAR_NO
    DoCmd ShowToolbar "Modo Formulário", A_TOOLBAR_WHERE_APPROP
End Sub
```

## Usando maiúscula na primeira letra em um campo

Você pode fazer formulários mais legíveis e relatórios com um aspecto mais profissional se os dados nos campos tiverem uma aparência consistente. Por exemplo, os nomes de produtos no banco de dados Northwind Traders começam todos com uma letra maiúscula. Como um nome de produto pode incluir várias palavras, algumas das quais você não quer com letras maiúsculas, é possível definir uma função que torna maiúscula somente a primeira letra da primeira palavra e deixa o resto das letras como digitadas.

### Visão geral

Crie uma função que torna maiúscula a primeira letra de um valor em um campo. Use esta função em um procedimento de evento na propriedade `DepoisDeAtualizar` do controle.

### Objetos utilizados neste exemplo

<b>Tipo</b>	<b>Nome</b>
Tabela	Produtos
Formulário	Inserir ou Editar Produtos
Módulo	Próprio

### Passo a passo

#### 1 Crie o módulo **Próprio**.

a) Na seção Declarações, digite:

```
Option Explicit
```

b) Crie a função `MaiúsculaPrimeira` function.

#### 2 No formulário **Inserir ou Editar Produtos**, crie um **procedimento de evento** na propriedade `DepoisDeAtualizar` na caixa de texto "Nome do produto".

#### 3 Na seção Declarações do módulo **Inserir ou Editar Produtos**, digite:

```
Option Explicit
```

---

**Dica** Você também pode criar uma função que torna maiúscula a primeira letra de cada palavra no campo.

---

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"



### **Função MaiúsculaPrimeira**

```
Function CapitalizeFirst (X)
    ' Torna a primeira letra do campo maiúscula; deixa outras
    ' letras como digitadas.
    Dim Temp
    Temp = Trim(X)
    CapitalizeFirst = UCase(Left(Temp, 1)) & Mid(Temp, 2)
End Function
```



### **Procedimento de evento DepoisDeAtualizar para a caixa de texto "Nome do produto"**

```
Sub Product_Name_AfterUpdate ()
    ' Torna maiúscula a primeira letra do nome do produto.
    If Not (IsNull(Me![Nome do Produto])) Then
        Me![Nome do Produto] = CapitalizeFirst((Me![Nome do Produto]))
    End If
End Sub
```



## Função Próprio

```
Function Próprio (X)
' Torna maiúscula a primeira letra de cada palavra em um campo.
' Use em um procedimento de evento DepoisDeAtualizar de controle;
' por exemplo, [Sobrenome] = Próprio([Sobrenome]).
' Nomes tais como O'Brien e Wilson-Smythe são transformados corretamente,
' mas MacDonald é mudado para Macdonald, e van Buren para Van Buren.
' Observação: Para que esta função funcione corretamente, você precisa
especificar
' Option Compare Database na seção Declarações deste módulo.
Dim Temp$, C$, OldC$, i As Integer
If IsNull(X) Then
Exit Function
Else
Temp$ = CStr(LCase(X))
' Inicializa OldC$ para um espaço simples porque a primeira
' letra precisa ser transformada em maiúscula mas não tem letra
anterior.
OldC$ = " "
For i = 1 To Len(Temp$)
C$ = Mid$(Temp$, i, 1)
If C$ >="a" And C$ <="z" And (OldC$ < "a" Or OldC$ > "z") Then
Mid$(Temp$, i, 1) = UCase$(C$)
End If
OldC$ = C$
Next i
Próprio = Temp$
End If
End Function
```

## Alterando a cor de uma seção e seus controles em tempo de execução

Alterando a cor de uma seção e seus controles em resposta a uma ação do usuário, você pode indicar se a seção está ativa ou inativa. Por exemplo, enquanto você insere novos registros no formulário Inserir ou Editar Produtos, o cabeçalho do formulário é verde e seus controles não estão disponíveis. Quando você clicar no botão "Editar Dados", o cabeçalho do formulário muda para azul claro e os controles tornam-se disponíveis. A cor mais escura indica um estado inativo enquanto que a cor mais clara indica um estado ativo.

### Visão geral

No formulário, defina a propriedade `CorDoFundo` do cabeçalho do formulário e de qualquer controle nele como a cor inativa. Para a função `EditarDados`, adicione código que defina as propriedades `CorDoFundo` do cabeçalho do formulário e cada controle nele como a cor ativa. Para a função `InserirDados`, adicione código que defina as propriedades `CorDoFundo` do cabeçalho do formulário e seus controles como a cor inativa.

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nome</u>
Tabela	Produtos
Formulário	Inserir ou Editar Produtos
Módulo	Botões Inserir ou Editar Produtos

### Passo a passo

- 1 No formulário Inserir ou Editar Produtos, defina a propriedade `CorDoFundo` do cabeçalho do formulário, e as caixas de combinação "Selecionar categoria" e "Selecionar produto" como 8421376 (Verde na paleta).**
- 2 Crie os botões "Entrar dados" e "Editar Dados", se você ainda não o fez.**
- 3 Modifique o módulo Botões Inserir ou Editar Produtos.**
  - a) Modifique a [função `EditarDados`](#).
  - b) Modifique a [função `InserirDados`](#).
- 4 Crie um [procedimento de evento](#) que redefine a cor do cabeçalho do formulário e seus controles na propriedade `AoFechar` do formulário.**

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos para eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários"
Criar macros	<i>Guia do Usuário</i> , capítulo 24, "Fundamentos de

Criar barras de  
ferramentas

macros"  
*Guia do Usuário,*  
Apêndice B,  
"Trabalhando com  
barras de  
ferramentas"



## Função EditarDados ()

O comentário e o código que você precisa adicionar à função para definir a cor do cabeçalho do formulário e seus controles estão exibidos em vermelho.

```
Function EditarDados () As Integer
    ' Código do botão "Editar dados" na barra de ferramentas Inserir ou
    Editar Produtos 1.
    '
    On Error GoTo EditData_Err
    Dim MyForm As Form
    Dim ComboBox1 As Control
    Dim ComboBox2 As Control
    Dim EditColor As Long
    Dim Tmp As Variant

    Set MyForm = Forms![Inserir ou Editar Produtos]
    Set ComboBox1 = Forms![Inserir ou Editar Produtos]![Selecionar categoria]
    Set ComboBox2 = Forms![Inserir ou Editar Produtos]![Selecionar produto]
    EditColor = 16776960

    ' Define a propriedade PermitirEdição como falsa para indicar modo
    edição.
    ' Define a propriedade EdiçãoPadrão como Impossível Adicionar Registros
    (4).
    MyForm.AllowEditing = False
    MyForm.DefaultEditing = 4
    ' Alterna barras de ferramentas.
    DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_NO
    DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_WHERE_APPROP
    ' Altera a cor do cabeçalho do formulário e das caixas de combinação.
    MyForm.Section(1).BackColor = EditColor
    ComboBox1.BackColor = EditColor
    ComboBox2.BackColor = EditColor
    ' Ativa e move para a primeira caixa de combinação.
    ComboBox1.Enabled = True
    ComboBox1.SetFocus
    ' Desativa controles na seção Detalhe.
    Tmp = EnableControls("Detalhe", False)
    Exit Function
EditData_Err:
    MsgBox Error$
    Exit Function
End Function
```



## Função InserirDados ()

O comentário e o código que você precisa adicionar à função para definir a cor do cabeçalho do formulário e seus controles estão exibidos em vermelho.

```
Function InserirDados () As Integer
    ' Código para o botão "Inserir Dados" na barra de ferramentas Inserir ou
    Editar Produtos 2.
    '
    On Error GoTo EnterData_Err
    Dim MyForm As Form
    Dim ComboBox1 As Control
    Dim ComboBox2 As Control
    Dim EnterColor As Long
    Dim Tmp As Variant

    Set MyForm = Forms![Inserir ou Editar Produtos]
    Set ComboBox1 = Forms![Inserir ou Editar Produtos]![Selecionar categoria]
    Set ComboBox2 = Forms![Inserir ou Editar Produtos]![Selecionar produto]
    EnterColor = 8421376

    ' Define a propriedade PermitirEdição como verdadeira para indicar modo
    entrada de dados.
    ' Define a propriedade EdiçãoPadrão como Entrada de Dados (1).
    MyForm.AllowEditing = True
    MyForm.DefaultEditing = 1

    ' Ativa controles da seção Detalhe e desativa a caixa de texto
    "Identificação de produto".
    Tmp = EnableControls("Detalhe", True)
    MyForm![Identificação do Produto].Enabled = False

    ' Vai para um novo registro e move para a caixa de combinação
    "Identificação do fornecedor".
    DoCmd GoToRecord , , A_NEWREC
    MyForm![Identificação do Fornecedor].SetFocus

    ' Alterna barras de ferramentas.
    DoCmd ShowToolbar "Inserir ou Editar Produtos 2", A_TOOLBAR_NO
    DoCmd ShowToolbar "Inserir ou Editar Produtos 1", A_TOOLBAR_WHERE_APPROP
    ' Altera cor do cabeçalho do formulário e das caixas de combinação.
    MyForm.Section(1).BackColor = EnterColor
    ComboBox1.BackColor = EnterColor
    ComboBox2.BackColor = EnterColor

    ' Desativa caixas de combinação no cabeçalho do formulário.
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False

    Exit Function
EnterData_Err:
    MsgBox Error$
    Exit Function
End Function
```



### **Procedimento para evento Fechar do formulário Inserir ou Editar Produtos**

```
Sub Form_Close ()  
    ' Redefine a cor do fundo do cabeçalho do formulário e seus  
    ' controles para a cor original.  
    Dim StartColor As Long  
    ' Define StartColor como verde.  
    StartColor = 8421376  
    Me.Section(1).BackColor = StartColor  
    Me![Selecionar categoria].BackColor = StartColor  
    Me![Selecionar categoria].BackColor = StartColor  
End Sub
```

## **Destaques da caixa de diálogo Sobre o Soluções**

O formulário Sobre o Soluções mostra como exibir o nome, copyright, e logotipo de um aplicativo em uma caixa de diálogo. Esta técnica faz o formulário Sobre o Soluções funcionar:



Criando uma caixa de diálogo Sobre.

## Criando uma caixa de diálogo Sobre

Você pode exibir informações sobre seu aplicativo em uma caixa de diálogo que esteja disponível a partir de uma barra de menu personalizada. A caixa de diálogo Sobre o Soluções está disponível a partir do menu Ajuda do formulário Editar Produtos.

### Visão geral

Crie um formulário independente que exiba informações sobre seu aplicativo. Crie uma macro que adicione um comando para exibir o formulário no grupo de macro do menu Ajuda.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Sobre o Soluções
Macro	Barra de Menu Editar Produtos_Ajuda

### Passo a passo

#### 1 Estruture o formulário Sobre o Soluções.

a) Defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Definição</u>
Legenda	Sobre o Soluções
ModoPadrão	Formulário simples
VisualizaçãoAutoriza da	Formulário
EdiçãoPadrão	Somente leitura
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
PopUp	Não
JanelaRestrita	Não
BotãoMin	Não
BotãoMax	Não
EstiloDaBorda	Diálogo

---

**Observação** As propriedades JanelaRestrita e PopUp do formulário Sobre o Soluções no Soluções são definidas como Não para que você possa ter acesso a comandos de barra de menu e botões de barra de ferramentas quando você olha para o formulário. Quando JanelaRestrita e PopUp estiverem definidas como Sim, o formulário terá uma borda de caixa de diálogo.

---

- b) Na seção detalhe, crie etiquetas que exibam o nome, número de versão, e copyright de seu aplicativo. Crie também uma moldura de objeto que exiba o logotipo de seu aplicativo.
- c) Crie um botão de comando. Defina suas propriedades e crie um procedimento de evento como a seguir.

<u>Propriedade</u>	<u>Definição</u>
Nome	OK

Legenda      OK  
AoClicar      [Procedimento de evento]

**2 Na seção Declarações do módulo Formulário.Sobre o Soluções, digite:**

Option Explicit

**3 Adicione uma macro ao final do grupo de macro Barra de Menu Editar Produtos\_Ajuda.** Consulte Crie uma barra de ferramentas e um menu que só incluam tarefas de edição para informações sobre criação de um menu personalizado.

<b>Nome da macro</b>	<b>Ação</b>	<b>Argumentos</b>
So&bre o Soluções	AbrirFormulário	Nome do formulário: Sobre o Soluções Exibir: Formulário Modo de dados: Somente leitura Modo da janela: Normal

---

**Dica** O aplicativo Soluções só usa um menu personalizado para o formulário Editar Produtos, portanto o Sobre o Soluções só está disponível a partir do menu Ajuda quando o formulário Editar Produtos está em modo Formulário. Se você usa barras de menu personalizadas ao longo de seu aplicativo, adicione a macro So&bre o Soluções a cada grupo de macro do menu Ajuda.

---

**Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", capítulo 16, "Personalizando formulários", e capítulo 19, "Usando imagens, gráficos e outros objetos"
Criando macros	<i>Guia do Usuário</i> , capítulo 24, "Fundamentos de macros"



### **Procedimento de evento Clicar para o botão de comando OK**

```
Sub OK_Click ()  
    ' Fecha o formulário.  
    DoCmd Close  
End Sub
```

## **Destaques do formulário Inicialização**

O formulário Inicialização mostra como exibir automaticamente um formulário por alguns segundos ao abrir um aplicativo. Esta técnica faz o formulário Inicialização funcionar:



Criando um formulário de inicialização.

## Criando um formulário de inicialização

Quando o seu aplicativo é iniciado, você pode querer exibir seu nome, copyright e logotipo por alguns instantes. No Soluções, o formulário Inicialização é exibido por vários segundos quando você abre pela primeira vez o banco de dados

### Visão geral

Crie um formulário independente que exiba informações sobre seu aplicativo. Defina a propriedade IntervaloDeTempo do formulário como um intervalo de tempo (expresso em milissegundos). Crie um procedimento de evento para a propriedade IntervaloDeTempo para fechar o formulário. Crie uma macro AutoExec para abrir o formulário.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Inicialização
Macro	AutoExec

### Passo a passo

#### 1 Estruture o formulário Inicialização.

a) Defina as seguintes propriedades do formulário.

<u>Propriedade</u>	<u>Definição</u>
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
MenuAtalho	Não
BarrasRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
EstiloDaBorda	Nenhum
AoTimer	<u>[Procedimento de evento]</u>
IntervaloDoTimer	7000

---

**Observação** Para que você possa abrir o formulário Inicialização e olhá-lo no Soluções, a propriedade IntervaloDoTimer é definida como 7000 quando você executa a macro AutoExec e reiniciada como 0 no procedimento de evento da propriedade AoFechar.

---

b) Na seção detalhe, crie etiquetas que exibam o nome, copyright e outras informações pertinentes de seu aplicativo. Além disso, crie uma moldura de objeto que exiba o logotipo de seu aplicativo.

#### 2 Na seção Declarações do módulo Formulário.Inicialização, digite:

Option Explicit

#### 3 Crie um grupo de macro chamado AutoExec para exibir automaticamente o formulário Inicialização.

<u>Ação</u>	<u>Argumento</u>
AbrirFormulário	Nome do formulário: Inicialização
	Exibir: Formulário
	Modo de dados: Somente leitura

Modo da janela:  
Normal

## Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários" capítulo 16, "Personalizando formulários", e capítulo 19, "Usando imagens, gráficos e outros objetos"
Criar macros	<i>Guia do Usuário</i> , capítulo 24, "Fundamentos de macros"



### **Procedimento do evento Timer para o formulário Inicialização**

```
Sub Form_Timer ()
    ' Abre o formulário Introdução ao Soluções ou o formulário Soluções.
    ' Fecha o formulário de inicialização.
    '
    Dim MyDB As Database
    Dim MyRecordset As Recordset
    Set MyDB = DBEngine.Workspaces(0).Databases(0)
    Set MyRecordset = MyDB.TableDefs("Opções").OpenRecordset()
    ' Primeira instrução If usada pelo Soluções para reiniciar a propriedade
IntervaloDeTempo.
    If Me.TimerInterval <> 0 Then
        TimerInterval = 0
        ' Use o valor do campo Ocultar Introdução ao Soluções da tabela
Opções para determinar
        ' qual formulário abrir.
        If MyRecordset.Fields("Ocultar Introdução ao Soluções").Value = 0
Then
            DoCmd OpenForm "Introdução ao Soluções"
        Else
            DoCmd OpenForm "Soluções"
        End If
    End If
End Sub
```

## Destaques do relatório Fatura

O relatório Fatura mostra como criar cartas formulário personalizadas que resumem as vendas totais para cada vendedor. Esta técnica faz o relatório Fatura funcionar:



Criando um relatório que é impresso em um formulário pré-impresso.

Esta técnica melhora o relatório:



Imprimindo "Continua na próxima página" na parte inferior do formulário pré-impresso.

## Criando um relatório que é impresso em um formulário pré-impresso

Alguns relatórios necessitam que você imprima os dados em locais predefinidos. Dados de faturas em um formulário predefinido, por exemplo, precisam aparecer em áreas apropriadas do formulário. Se uma fatura requer mais de uma página, algumas informações, tal como o nome e o endereço do cliente, precisa ser impresso em todas as páginas. Outras informações, tal como o total da fatura, necessita de ser impresso somente na última página.

### Visão geral

Coloque os controles do nome, endereço do cliente, e outros dados que se repetem no cabeçalho da página. Coloque os controles do subtotal, frete, e total no rodapé da página. Crie uma caixa de texto não-vinculada no rodapé Número do Pedido. Crie um procedimento de evento na propriedade AoFormatar do rodapé do grupo que define o valor da caixa de texto como Verdadeiro. Crie um procedimento de evento na propriedade AoFormatar do rodapé da página que imprime o rodapé da página somente depois de todos os detalhes de registros terem sido impressos (isto é, quando o valor na caixa de texto não-vinculada for Verdadeiro).

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nomes</u>
Tabela	Clientes, Empregados, Pedidos
Consulta	Fatura, Detalhes Adicionais do Pedido
Relatório	Fatura

### Passo a passo

**1 Importe a consulta Detalhes Adicionais do Pedido do NWIND.MDB, se você ainda não o fez.**

**2 Crie a consulta Fatura.**

a) Use as seguintes tabelas e campos.

<u>Tabela/ Consulta</u>	<u>Campos</u>
Tabela Clientes	Nome da companhia, Endereço, Cidade, Região, Código postal, País
Tabela Empregados	Sobrenome, Primeiro nome
Tabela Pedidos	Número do pedido, Código do cliente, Nome do destinatário, Endereço do comprador, Cidade de destino, Região do destinatário, Código postal do destinatário, País de destino, Enviar por via, Data do

	pedido, Data de entrega, Data de envio, Frete
Consulta Detalhes Adicionais do Pedido	Nome do produto, Identificação do produto, Preço unitário, Quantidade, Desconto, Preço total

b) Defina o critério para o campo Data do pedido como: Entre #1/2/94# e #30/3/94#.

---

**Dica** Em seu aplicativo, você provavelmente deseja avisar os usuários para o critério de data quando eles visualizem ou imprimam o relatório ao invés de especificar as datas na consulta. Para um exemplo de definição de critério em tempo de execução, consulte o relatório Vendas de Empregados.

---

### 3 Estructure o relatório Fatura.

a) Defina as seguintes propriedades de classificação e de grupos.

<b>Campo/ Expressão</b>	<b>Ordem de Classificação</b>	<b>Propriedades de Grupo</b>
Número do pedido	Descendente	Cabeçalho do Grupo: Sim Rodapé do Grupo: Sim Reagrupar: Cada Valor Intervalo de Grupo: 1 Manter Juntos: Não
Identificação do produto	Ascendente	(Definições padrão)

---

**Observação** Número do pedido é classificado em ordem descendente no relatório Fatura, desse modo, a fatura que necessitar de várias páginas aparecerá primeiro.

---

b) Na caixa de diálogo Configurar Impressora, defina as margens Esquerda, Direita, Superior, e Inferior como 2,540 cm (1 pol).

c) Defina a propriedade Altura para as seguintes seções.

<b>Seção</b>	<b>Altura</b>
Cabeçalho da página	9,36625 cm. (3,6875 pol)
Seção Detalhe	0,635 cm. (0,25 pol.)
Rodapé da página	2,540 cm. (1 pol.)

O cabeçalho e rodapé Número do pedido não são visíveis no relatório impresso, por isso você não precisa definir suas propriedades Altura.

d) No cabeçalho de página, crie caixas de texto para o nome, endereço do cliente, e qualquer outra informação que você queira que apareça em todas as páginas.

e) Na seção Detalhe, crie caixas de texto com suas propriedades definidas como a seguir.

<b>Nome</b>	<b>OrigemDoContr</b>
-------------	----------------------

	<b>ole</b>
Identificação do produto	Identificação do produto
Nome do produto	Nome do produto
Quantidade	Quantidade
Preço unitário	Preço unitário
Desconto	Desconto
Preço total	Preço total

- f) No rodapé Número do pedido, crie uma caixa de texto não-vinculada. Defina sua propriedade Nome como "Imprimir rodapé da página".
- g) No rodapé Número do pedido, crie uma caixa de texto que calcule o subtotal da fatura. Defina sua propriedade Nome como Subtotal e sua propriedade OrigemDoControle como: = Soma([Preço Total])
- h) Crie caixas de texto no rodapé da página com suas propriedades definidas como a seguir.

<b>Nome</b>	<b>OrigemDoControle</b>
Subtotal da fatura	= [Subtotal]
Frete	Frete
Total da fatura	= [Subtotal da fatura] + [Frete]

- i) Defina as propriedades e crie procedimentos de eventos para as seguintes seções.

<b>Seção</b>	<b>Propriedade</b>	<b>Definição</b>
Cabeçalho Número do pedido	Visível	Não
Rodapé Número do pedido	ForçaQuebraDePágina	Depois da seção
	Visível	Não
	AoFormatar	[ <u>Procedimento de evento</u> ]
Rodapé da página	AoFormatar	[ <u>Procedimento de evento</u> ]

- j) Na seção Declarações do módulo Report.Fatura, digite:

Option Explicit

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"

Criar consultas	<i>Guia do Usuário</i> , capítulo 10, "Fundamentos de tabelas," e capítulo 11, "Estruturando consultas seleção"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios," e capítulo 21, "Estruturando relatórios"
Definir propriedades de classificação e agrupamento	<i>Guia do Usuário</i> , capítulo 22, "Classificando e agrupando dados"



### **Procedimento de evento Formatar para rodapé Número do Pedido**

```
Sub GroupFooter3_Format (Cancel As Integer, FormatCount As Integer)
    ' Redefine a caixa de texto do "Imprimir rodapé de página" que mostra
    ' todos os detalhes
    ' de registros que foram impressos na ordem atual. O relatório
    ' imprimirá
    ' rodapé da página quando o valor do "Imprimir rodapé de página" for
    ' Verdadeiro.
    Me![Imprimir rodapé de página] = True
End Sub
```



### **Procedimento de evento Formatar para Rodapé da página**

```
Sub PageFooter4_Format (Cancel As Integer, FormatCount As Integer)
    ' Determina se o Rodapé da página será impresso.
    '
    ' Imprime rodapé da página se todos os detalhes de registro na
    ' ordem atual foram impressos.
    If Me![Imprimir rodapé de página] Then
        Me.PrintSection = True
        Me![Imprimir rodapé de página] = False
    Else
        ' Não imprime rodapé da página se todos os detalhes de registro
        ' na ordem atual não foram impressos.
        Me.PrintSection = False
    End If
End Sub
```

## Imprimindo "Continua na próxima página" na parte inferior de um formulário pré-impresso

Quando existem muitos detalhes de registros para um grupo, você pode imprimir "Continua na próxima página" na parte inferior da página para indicar que mais registros seguirão na próxima página. Por exemplo, no relatório Fatura, há espaço para imprimir 17 detalhes de registro em uma página. Se a fatura tem mais de 17 detalhes de registros, o relatório imprimirá 16 detalhes de registro e "Continua na próxima página" na parte inferior da primeira página onde o detalhe de registro 17 teria sido impresso, e então imprimirá os detalhes de registro 17 na próxima página.

### Visão geral

Determine o número máximo de seções Detalhe que podem ser impressas em uma página. Quando o relatório formatar a última seção Detalhe em uma página (a 17ª seção Detalhe para este relatório), oculte os controles no detalhe de registro, imprima "Continua na próxima página", e não avance para o próximo registro.

### Objetos utilizados neste exemplo

<u>Tipo</u>	<u>Nomes</u>
Tabela	Clientes, Empregados, Pedidos
Consulta	Fatura, Detalhes Adicionais do Pedido
Relatório	Fatura

### Passo a passo

#### **1 Crie um relatório para um formulário pré-impresso.**

**2 No cabeçalho da página, crie uma caixa de texto não-vinculada.** Defina sua propriedade Nome como Linhas Por Página, sua propriedade OrigemDoControle como : = 17, e sua propriedade Visível como Não.

#### **3 No cabeçalho Número do Pedido, crie duas caixas de texto não-vinculadas.**

- Defina a propriedade Nome de uma das caixas de texto como Contador de Linhas, e defina sua propriedade OrigemDoControle como: =Contar([Identificação do produto])
- Defina a propriedade Nome da outra caixa de texto como Ordem da Página, e deixe sua propriedade OrigemDoControle em branco.

---

**Dica** No relatório Fatura, o cabeçalho Número do Pedido não é visível. Se você cria um relatório onde está visível, esteja certo de ter definido a propriedade Visível para estes controles como Não.

---

#### **4 Faça as seguintes alterações na seção Detalhe.**

- Crie uma caixa de texto não-vinculada atrás dos outros controles no centro da seção Detalhe. Defina sua propriedade Nome como Continua, e sua propriedade FonteEmltálico como Sim, e sua propriedade OrigemDoControle como: ="Continua na próxima página..."
- Crie uma caixa de texto não-vinculada na borda direitada seção Detalhe. Defina sua propriedade Nome como Linha e sua propriedade Visível como Não.
- Crie um controle de quebra de página na parte inferior da seção. Defina sua propriedade Nome como Quebra de Página Detalhe.

#### **5 Crie procedimentos para evento para as seguintes seções.**

<u>Seção</u>	<u>Propriedade</u>	<u>Definição</u>
Cabeçalho da página	AoFormatar	[ <u>Procedimento de evento</u> ]

Cabeçalho Número do Pedido	AoFormatar	<a href="#">[Procedimento de evento]</a>
Seção Detalhe	AoFormatar	<a href="#">[Procedimento de evento]</a>

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar controles	<i>Guia do Usuário</i> , capítulo 21, "Estruturando relatórios"
Criar procedimentos de eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar expressões	<i>Guia do Usuário</i> , capítulo 23, "Usando expressões em relatórios"



### **Procedimento de evento Formatar para Cabeçalho da Página**

```
Sub PageHeader0_Format (Cancel As Integer, FormatCount As Integer)
' Na parte superior da página, inicialize os controles na seção Detalhe.
'
' Defina o valor da caixa de texto "Linha" como 0.
' Defina propriedade Visível para exibir controles no detalhe de registro
' e ocultar controles Continua e Quebra de Página Detalhe.
Me![Linha] = 0
Me![Continua].Visible = False
Me![Quebra de Página Detalhe].Visible = False
Me![Identificação do Produto].Visible = True
Me![Nome do Produto].Visible = True
Me![Quantidade].Visible = True
Me![Preço Unitário].Visible = True
Me![Desconto].Visible = True
Me![Preço Total].Visible = True
End Sub
```



### Procedimento de evento Formatar para a seção Detalhe

```
Sub Detail2_Format (Cancel As Integer, FormatCount As Integer)
    ' Determina se imprime o detalhe de registro ou "Continua."
    '
    ' Mostra caixa de texto "Continua" se está no número máximo de
    ' detalhes de registro para a página.
    If (Me![Linha] = Me![Ordem da Página] * (Me![Linhas Por Página] - 1) + 1)
And Me![Linha] <> Me![Contagem de Linhas] Then
        Me![Continua].Visible = True
    End If
    ' Exibe quebra de página e oculta controles nos detalhes de registro.
    If Me![Continua].Visible Then
        Me![Quebra de Página Detalhe].Visible = True
        Me![Identificação do Produto].Visible = False
        Me![Nome do Produto].Visible = False
        Me![Quantidade].Visible = False
        Me![Preço Unitário].Visible = False
        Me![Desconto].Visible = False
        Me![Preço Total].Visible = False
        ' Aumenta valor na Ordem da Página.
        Me.NextRecord = False
        Me![Ordem da Página] = Me![Ordem da Página] + 1
        ' Amplia Contagem de linhas se o detalhe de registro foi impresso.
    Else
        Me![Linha] = Me![Linha] + 1
    End If
End Sub
```



### **Procedimento de evento Formatar para o Cabeçalho do Número do Pedido**

```
Sub GroupHeader1_Format (Cancel As Integer, FormatCount As Integer)
    ' Define valor da caixa de texto "Linha" como 1 para marcar início
    ' do grupo da ordem atual.
    ' Define valor da caixa de texto "Ordem da página" como 1 para mostrar
que
    ' está é a primeira página de pedidos.
    Me![Linha] = 1
    Me![Ordem da Página] = 1
    ' Oculta controles Continua e Quebra de Página Detalhe.
    Me![Continua].Visible = False
    Me![Quebra de Página Detalhe].Visible = False
End Sub
```

## Destaques do relatório Registro de Remessas

O relatório Registro de Remessas mostra como exibir os dados atuais em formato de colunas de tipo jornalístico, ou estilo lista telefônica. Esta técnica faz com que o relatório Registro de Remessas funcione:



Criando um relatório de colunas múltiplas.

Estas técnicas ressaltam o relatório:



Repetindo o nome do grupo na parte superior de uma coluna ou página.



Ocultando um rodapé de grupo quando ele aparece no topo de uma página ou coluna.

## Criando um relatório de colunas múltiplas

Você pode criar um relatório no estilo lista telefônica imprimindo dados em duas ou mais colunas de tipo jornalístico em uma página. Por exemplo, o relatório Registro de Remessas imprime seus dados em duas colunas.

### Visão geral

Defina a largura do relatório como a largura da área de impressão na página (a área dentro das margens). Defina o número e a largura das colunas da página nas propriedades Configurar Impressora do relatório.

### Objetos utilizados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Pedidos
Consulta	Subtotais de Pedidos, Registro de Remessas
Relatório	Registro de Remessas

### Passo a passo

**1 Importe a consulta Subtotais de Pedidos do NWIND.MDB, se você ainda não o fez.**

**2 Crie a consulta Registro de Remessas.**

<u>Tabela</u>	<u>Campo</u>	<u>Critério</u>
Tabela Pedidos	Data do envio	Entre #1/2/92# e #31/3/92#
	Número do pedido	--
Consulta Subtotais de Pedidos	Subtotal	--

---

**Dica** Em seu aplicativo, você provavelmente quer avisar os usuários do critério de datas quando eles visualizarem ou imprimirem o relatório ao invés de especificarem as datas na consulta. Para um exemplo de definição de critério em tempo de execução, consulte o relatório [Vendas de Empregados](#).

---

**3 Estructure o relatório Registro de Remessas.**

a) Defina as seguintes propriedades de classificação e de grupo.

<u>Campo/Expressão</u>	<u>Ordem de Classificação</u>	<u>Propriedades de Grupo</u>
Data de envio	Descendente	Cabeçalho de Grupo: Sim Rodapé de Grupo: Sim Reagrupar: Cada Valor Intervalo de Grupo: 1 Manter Junto: Com primeiro detalhe
Número do pedido	Ascendente	(Definições padrão)

- b) Defina a propriedade Largura do relatório para 22,86 cm (9 polegadas).  
 c) Defina as seguintes opções de Configurar Impressora.

<b>Opção</b>	<b>Definição</b>
Orientação	Paisagem
Margens	Esquerda: 2,54 cm (1 pol) Direita: 2,54 cm (1 pol) Superior: 2,54 cm (1 pol) Inferior: 2,54 cm (1 pol)
Itens através	2
Espaçamento entre colunas	0.75 in
Tamanho do item	largura: 3.25 in altura: 0.2299 in
Layout do item	Vertical

- d) Coloque etiquetas com as seguintes legendas no cabeçalho de página: Data de envio, Número do pedido, e Quantidade do pedido.  
 e) Coloque as seguintes caixas de texto na seguintes seções do relatório, definindo suas propriedades Nome e OrigemDoControle como a seguir.

<b>Seção</b>	<b>Nome</b>	<b>OrigemDoControle</b>
Cabeçalho Data do Envio	Data do envio	Data do envio
Detalhe	Número do pedido Subtotal	Número do pedido Subtotal

- f) Copie as etiquetas Data do envio, Número do pedido, e Quantidade do pedido no cabeçalho da página, e cole-as à direita do primeiro grupo (a propriedade Esquerda da segunda etiqueta Data de envio deve ser definida para 10,16 cm (4 pol)).

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar consultas	<i>Guia do Usuário</i> , capítulo 10, "Fundamentos de tabelas," e capítulo 11, "Estruturando consultas seleção"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios," e capítulo 21, "Estruturando relatórios"

Definir  
propriedades de  
classificação e  
de  
agrupamento

*Guia do Usuário,*  
capítulo 22,  
"Classificando e  
agrupando dados"

## Repetindo o nome do grupo na parte superior de uma coluna ou página

O Microsoft Access imprime um cabeçalho de grupo imediatamente no início de um grupo, desse modo você pode facilmente identificar onde um grupo começa. Contudo, se os detalhes de registros em um grupo apresentam várias colunas ou páginas, é útil imprimir o nome do grupo na parte superior de cada nova coluna ou página. Por exemplo, no relatório Registro de Remessas, Pedidos estão agrupados pela data de envio. Quando um grupo continua em outra coluna ou página, a data de envio é impressa novamente.

### Visão geral

Use a propriedade Superior para determinar se uma seção Detalhe está sendo formatada na parte superior de uma coluna ou página. Se está, exiba a caixa de texto que repete o nome do grupo, oculte os outros controles na seção Detalhe, e não avance para o próximo registro.

### Objetos utilizados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Pedidos
Consulta	Subtotais de Pedidos, Registro de Remessas
Relatório	Registro de Remessas

### Passo a passo

#### 1 Crie o relatório Registro de Remessas.

#### 2 Adicione os seguintes controles ao relatório.

- No cabeçalho Data de envio, crie uma caixa de texto não-vinculada. Defina sua propriedade Nome como Superior1 e sua propriedade Visível como Não.
- Na seção Detalhe, crie uma caixa de texto não-vinculada. Defina sua propriedade Nome como Continua, sua propriedade FonteEmItálico como Sim, e sua propriedade OrigemDoControle como:

=Format([Data de envio], "dd-mmm-aa") & " continua"

#### 3 Crie procedimentos de eventos para as seguintes seções.

<u>Seção</u>	<u>Propriedade</u>	<u>Definição</u>
Cabeçalho da página	AoFormatar	[ <u>Procedimento de evento</u> ]
Cabeçalho Data de envio	AoFormatar	[ <u>Procedimento de evento</u> ]
Detalhe	AoFormatar	[ <u>Procedimento de evento</u> ]

#### 4 Na seção Declarações do módulo Report.Registro de Remessas, digite:

Option Explicit

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos de eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o

Criar relatórios

Access Basic," e capítulo 5, "Noções básicas do Access Basic" *Guia do Usuário*, capítulo 20, "Fundamentos de relatórios," e capítulo 21, "Estruturando relatórios"



### **Procedimento de evento Formatar para o Cabeçalho da Página**

```
Sub PageHeader1_Format (Cancel As Integer, FormatCount As Integer)
    ' Na parte superior da página, defina o valor da caixa de texto
    "Superior1" como 0.
    Me![Superior1] = 0
End Sub
```



### **Procedimento de evento Formatar para o cabeçalho Data de Envio**

```
Sub GroupHeader2_Format (Cancel As Integer, FormatCount As Integer)
    ' Define valor da caixa de texto "Superior1" para o valor da propriedade
    ' Superior localizar a posição do cabeçalho do grupo na página.
    If Me![Superior1] = 0 Then
        Me![Superior1] = Me.Top
    End If
End Sub
```



### **Procedimento de evento Formatar para a seção Detalhe**

```
Sub Detail3_Format (Cancel As Integer, FormatCount As Integer)
    ' Determina se a seção Detalhe é a primeira seção
    ' formatada após o cabeçalho da página. Se é, oculte controles
    ' no detalhe de registro e exiba caixa de texto "Continua".
    '
    ' Define o valor da caixa de texto "Superior1" para o valor da
propriedade Superior
    ' para localizar a posição do detalhe de registro na página.
    If Me![Superior1] = 0 Then
        Me![Superior1] = Me.Top
    End If
    ' Exibe a caixa de texto "Continua" e oculta caixas de texto no
    ' detalhe de registro. Não avança para o próximo registro.
    If Me.Top = Me![Superior1] Then
        Me![Continua].Visible = True
        Me![Número do pedido].Visible = False
        Me![Subtotal].Visible = False
        Me.NextRecord = False
    ' Oculta caixa de texto "Continua" e exibe caixas de texto no
    ' detalhe do registro.
    Else
        Me![Continua].Visible = False
        Me![Número do pedido].Visible = True
        Me![Subtotal].Visible = True
    End If
End Sub
```

## Ocultando uma seção quando ela é impressa na parte superior de uma página ou coluna

Utilizar um rodapé de grupo em branco para separar grupos é útil quando você quer que a divisão entre eles seja bem óbvia. Contudo, você não quer um rodapé de grupo em branco impresso quando ele é a primeira seção em uma página ou em uma coluna. No relatório Registro de Remessas, o rodapé Data de envio é uma seção em branco que não é impressa se ela cair na parte superior de uma página ou coluna.

### Visão geral

Utilize a propriedade Superior para determinar se o rodapé do grupo é a primeira seção a ser formatada na parte superior de uma página ou coluna. Se ela é, não imprima a seção e não mova para o próximo local de impressão na página.

### Objetos utilizados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Pedidos
Consulta	Subtotais de Pedidos, Registro de Remessas
Relatório	Registro de Remessas

### Passo a passo

- 1 Crie um relatório de colunas múltiplas.**
- 2 Adicione a caixa de texto "Superior1" ao relatório, e crie procedimentos de eventos para o cabeçalho da página e para o cabeçalho Data de Envio como descrito em Repetindo o nome do Grupo na parte superior de uma coluna ou página.**
- 3 Crie um procedimento de evento para a propriedade AoFormatar do rodapé Data de envio.**

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introducing Access Basic," e capítulo 5, "Noções básicas do Access Basic"



### **Procedimento de evento Formatar para o rodapé Data de Envio**

```
Sub GroupFooter4_Format (Cancel As Integer, FormatCount As Integer)
    ' Determina se o rodapé de grupo é a primeira seção formatada
    ' na parte superior da página ou coluna.
    '
    ' Não imprime o rodapé do grupo se qualquer instrução for verdadeira.
    If Me![Superior1] = 0 Or Me.Top = Me![Superior1] Then Cancel = True
End Sub
```

## Destaques do relatório Carta de Vendas

O relatório Carta de Vendas é uma carta modelo que resume um total de vendas de empregados. Esta técnica faz com que o relatório Carta de Vendas funcione:



Criando uma carta modelo.

Esta técnica ressalta o relatório:



Realçando dados colocando um círculo ao seu redor.

## Criando uma carta modelo

Quando você precisa enviar a mesma carta para várias pessoas, pode utilizar um relatório para criar uma carta modelo. Utilizando os dados em seu banco de dados, você pode personalizar o nome, endereço e outras informações de cada carta. Por exemplo, o relatório Carta de Vendas imprime a mesma mensagem de congratulações em todas as cartas, mas o nome, meta e totais do trimestre são especificados para cada vendedor.

### Visão geral

Crie as consultas de origem para o relatório. Coloque o nome, endereço e outros controles para o relatório principal em sua seção Detalhe. Crie um sub-relatório para os totais do trimestre e adicione-o à seção Detalhe do relatório principal.

### Objetos utilizados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Empregados, Pedidos, Metas de Vendas
Consulta	Subtotais dos Pedidos, Carta de Vendas 1, Carta de Vendas 2
Relatório	Carta de Vendas, Sub-relatório Carta de Vendas

### Passo a passo

**1 Importe a consulta Subtotais de Pedidos do NWIND.MDB, se você ainda não o fez.**

**2 Crie Carta de Vendas 1, uma consulta tabela de referência cruzada.**

a) Utilize as seguintes tabelas e campos.

<u>Tabela/ Consulta</u>	<u>Campo/ Expressão</u>	<u>Propriedades</u>
Tabela Empregados	Número do empregado	Total: Agrupar por Referência cruzada: Linha
	Nome: [Sobrenome] & ", " & [Primeiro nome]	Total: Agrupar por Referência cruzada: Linha
Tabela Pedidos	Ano: PartData("aaaa", [Data de envio])	Total: Agrupar por Referência cruzada: Linha
	Expr1: "Trim" & PartData("b", [Data de envio])	Total: Agrupar por Referência cruzada: Coluna
	PartData("aaaa", [Data de envio])	Total: Onde Critério: 1993
Consulta Subtotais de Pedidos	Subtotal	Total: Soma Referência cruzada: Valor

---

**Dica** Em seu aplicativo, você irá provavelmente querer avisar aos usuários quanto ao critério de datas quando eles visualizarem ou imprimirem o relatório ao invés de

especificar as datas na consulta. Para um exemplo de definição de critério em tempo de execução, consulte o relatório Vendas de Empregados.

- b) Defina a propriedade `TítulosDeColuna` da consulta com os seguintes valores: "Bim1", "Bim2", "Bim3", "Bim4"

### 3 Crie a consulta Carta de Vendas 2.

Tabela/Consulta	Campo/Expressão	Propriedades
Tabela Empregados	Sobrenome	Classificação: Ascendente
	Salesperson: [Primeiro nome] & " " & [Sobrenome]	--
	Número do empregado, Endereço, Cidade, Região, Código Postal, País	--
Consulta Metas de Vendas	Meta	--
	Ano da meta: PartData("aaaa", [Metas de Vendas].[Ano])	Critério: [Carta de Vendas 1].[Ano]
Consulta Carta de Vendas1	Trim1, Trim2, Trim3, Trim4	--

### 4 Crie o relatório Carta de Vendas.

- Defina a propriedade `OrigemDoRegistro` do relatório como Carta de Vendas 2.
- Remova o cabeçalho e rodapé da página. (Você só precisa da seção Detalhe para este relatório.)
- Na caixa Classificar e Agrupar, selecione Sobrenome na coluna Campo/Expressão e defina Ordem de Classificação como Ascendente.
- Defina a propriedade `ForçaQuebraDePágina` da seção Detalhe como "Depois da seção".
- Crie caixas de texto para o nome, endereço, saudação, e o texto da carta. Para a saudação, utilize uma expressão como a seguinte:

= "Caro(a) " & [Vendedor] & " :"

Para concatenar o texto e os dados em um parágrafo, utilize uma expressão como:

= "Congratulações! Você cumpriu ou excedeu sua meta de vendas trimestrais de " & Format([Meta], "\$#,##0") & " em " & [Ano da meta] & ". No resumo seguinte de suas vendas, cada semestre dentro de 90% de sua meta está circundado."

Se uma caixa de texto contém um ou mais parágrafos, faça a caixa de texto ter a altura de uma linha de texto e defina sua propriedade `PodeAmpliar` como Sim.

### 5 Crie um sub-relatório Carta de Vendas.

- Defina a propriedade `OrigemDoRegistro` como Carta de Vendas 2.
- Remova o cabeçalho e o rodapé da página. (Você só precisa da seção Detalhe.)
- Crie as caixas de texto para os totais do trimestre e para a meta. Faça que as caixas de texto tenham a mesma largura e altura, e espace-as evenly across a seção Detalhe. Por exemplo, para cada caixa de texto, defina a propriedade `Largura` como 2,54 cm (1 pol) e a propriedade `Altura` como 0,2027 cm (0,1979 pol). Deixe 0,3175 cm (0,125 pol) entre cada caixa de texto.

d) Salve o relatório.

### 6 Adicione o sub-relatório ao relatório principal.

- Selecione o sub-relatório Carta de Vendas na janela Banco de Dados, e arraste-o para a seção Detalhe do relatório Carta de Vendas.
- Exclua a etiqueta para o controle do sub-relatório e ajuste o tamanho do controle do sub-relatório. Por exemplo, faça o sub-relatório ter a altura de uma linha de texto.
- Verifique que as propriedades VincularAPartirDe e VincularA do controle do sub-relatório Carta de Vendas estão definidas como Número do empregado.

### 7 Ajuste a colocação dos controles no relatório principal.

Por exemplo, tenha certeza que as partes da carta, como endereço, saudação, corpo do texto, e fechamento, estão colocadas de acordo com o estilo de carta que você está seguindo.

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"
Criar consultas	<i>Guia do Usuário</i> , capítulo 11, "Estruturando consultas seleção" e capítulo 12, "Consultas avançadas"
Criar relatórios	<i>Guia do Usuário</i> , capítulo 20, "Fundamentos de relatórios," e capítulo 21, "Estruturando relatórios"
Criar sub-relatórios	<i>Guia do Usuário</i> , capítulo 22, "Classificando e agrupando dados"
Criar expressões	<i>Guia do Usuário</i> , capítulo 23, "Usando expressões em relatórios"
Definir propriedades de classificação e de grupos	<i>Guia do Usuário</i> , capítulo 22, "Classificando e agrupando dados"

## Realçando dados colocando um círculo ao seu redor

Você pode chamara atenção para dados específicos destacando-os de alguma forma. Por exemplo, o relatório Carta de Vendas destaca o total do trimestre quando este é de 90 por cento ou maior do que a meta colocando um círculo ao seu redor.

### Visão geral

Utilize o método Circle para desenhar um círculo ou elipse ao redor de um controle que encontra o critério que você especifica.

### Objetos utilizados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Empregados, Pedidos, Metas de Vendas
Consulta	Subtotais de Pedidos, Carta de Vendas 1, Carta de Vendas 2
Relatório	Carta de Vendas, Sub-relatório Carta de Vendas

### Passo a passo

- 1 Crie o relatório Carta de Vendas.**
- 2 No sub-relatório Carta de Vendas, crie um procedimento de evento para a propriedade AoImprimirda seção Detalhe.**
- 3 Na seção Declarações do módulo Report.Cartas de Vendas digite:**

Option Explicit

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos para eventos	<i>Criando Aplicativos</i> , capítulo 3, "Introduzindo o Access Basic," e capítulo 5, "Noções básicas do Access Basic"



## Procedimento de evento Imprimir para a seção Detalhe

```
Sub Detail0_Print (Cancel As Integer, PrintCount As Integer)
    ' Desenha elipse ao redor dos controles que encontrem um critério
    específico.
    Dim MyControl As Control, MyReport As Report, MySubreport As Control
    Dim PrintCircle As Integer
    Dim Aspect As Single, YOffset As Single
    Dim EllipseHeight As Integer, EllipseWidth As Integer
    Dim XCoord As Single, YCoord As Single
    ' Para produzir-se a elipse desejada, defina Aspecto e YOffset
    ' de acordo com o tamanho e localização das caixas de texto no
    ' sub-relatório.
    Aspect = .23
    YOffset = 200
    ' Determina altura e largura da elipse.
    EllipseHeight = Me![Goal].Height * 1.1
    EllipseWidth = Me![Goal].Width * 1.04
    YCoord = (Me![Goal].Top + Me![Goal].Height) \ 2 + YOffset
    ' Identifique os controles que podem ser circundados.
    Set MyControl = Me![Qtr1]
    GoSub PrintCircle
    Set MyControl = Me![Qtr2]
    GoSub PrintCircle
    Set MyControl = Me![Qtr3]
    GoSub PrintCircle
    Set MyControl = Me![Qtr4]
    GoSub PrintCircle
Exit Sub

PrintCircle:
    ' Se MyControl não é nulo, então determine se o valor é de 90% ou
    ' maior que o valor na caixa de texto "Meta".
    If Not IsNull(MyControl) Then
        PrintCircle = (MyControl >= .9 * Me![Meta])
        ' Determina coordenadas de MyControl e desenha elipse.
        If PrintCircle Then
            XCoord = MyControl.Left + (MyControl.Width \ 2)
            Me.Circle (XCoord, YCoord), EllipseWidth \ 2, , , , Aspect
        End If
    End If
    Return
End Sub
```

## Destaques do formulário Encontrar Clientes

O formulário Encontrar Clientes mostra como inserir critérios de procura em um ou mais campos. Esta técnica faz o formulário Encontrar Clientes funcionar.



Encontrando registros com base em critérios que você insere nos campos Companhia, Nome do contato, Cidade e País.

Esta técnica aperfeiçoa o formulário Encontrar Clientes.



Limpendo os campos de procura e o subformulário com o botão Limpar.

## Encontrando registros com base nos critérios que você insere em um ou mais campos

Quando você procura por registros, o critério de procura pode variar cada vez que você procura. No formulário Encontrar Clientes, por exemplo, você pode encontrar um registro de clientes inserindo um valor em qualquer um ou mais dos campos do cabeçalho do formulário: Companhia, Nome do contato, Cidade ou País. Além disso, você pode ter somente parte do critério de procura de que necessita - por exemplo, você pode se lembrar somente de que o nome da companhia começa com "B". No formulário Encontrar Clientes, você pode inserir "B" no campo Companhia e então procurar pelos registros por todas as companhias que comecem com "B".

### Visão geral

Crie um subformulário que contenha os campos a partir da tabela Clientes. Crie um formulário principal independente que contenha os campos de procura e um botão de comando. Na propriedade AoClicar do botão de comando, crie um procedimento de evento que crie uma instrução SELECT baseada no critério inserido nos campos de procura e então defina a propriedade OrigemDoRegistro do formulário com a instrução SELECT.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Clientes
Formulário	Encontrar Clientes, Subformulário Encontrar Clientes
Módulo	Ativar controles

### Passo a passo

#### 1 Crie o módulo Ativar Controles, se você já não o tiver feito.

a) Na seção Declarações do módulo, digite:

```
Option Explicit
```

b) Crie a função AtivarControles.

#### 2 Crie o formulário Encontrar Clientes.

a) Crie um formulário independente, e defina suas propriedades como a seguir.

<u>Propriedade</u>	<u>Definição</u>
Legenda	Encontrar Clientes
ModoPadrão	Formulário simples
VisualizaçãoAutoriza da	Formulário
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
AoAbrir	<u>[Procedimento de evento]</u>

b) Adicione um cabeçalho e um rodapé de formulário. Defina a propriedade Altura do rodapé do formulário como 0.

c) Na seção Declarações do módulo Formulário.Encontrar Clientes, digite:

```
Option Explicit
```

d) Na seção Geral do módulo Formulário.Encontrar Clientes, crie o sub AdicionarAWhere e

o sub DesativarControle.

- e) No cabeçalho do formulário, crie quatro caixas de texto independentes. Defina suas propriedades como a seguir.

<u>Nome</u>	<u>DepoisDeAtualizar</u>
Procurar por companhia	[ <u>Procedimento de evento</u> ]
Procurar por contato	[ <u>Procedimento de evento</u> ]
Procurar por cidade	[ <u>Procedimento de evento</u> ]
Procurar por país	[ <u>Procedimento de evento</u> ]

- f) No cabeçalho do formulário, crie um botão de comando, e defina suas propriedades como a seguir.

<u>Propriedade</u>	<u>Valor</u>
Nome	Mostrar clientes
Legenda	Mostrar clientes
Padrão	Sim
AoClicar	[ <u>Procedimento de evento</u> ]

### 3 Crie o subformulário Encontrar Clientes

- a) Defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	SELECT * FROM Clientes WHERE False
ModoPadrão	Folha de Dados

- b) Na seção Detalhe, crie caixas de texto para todos os campos da lista de campos.

---

**Dica** Para ajustar a largura das colunas em modo Folha de Dados, recorte a instrução WHERE False da propriedade OrigemDoRegistro, alterne para modo Folha de Dados, ajuste as larguras das colunas, e então salve o formulário. Volte ao modo Estrutura, cole a instrução WHERE False, e salve o formulário novamente.

---

### 4 Adicione o Subformulário Encontrar Clientes ao formulário Encontrar Clientes.

- a) Selecione o Subformulário Encontrar Clientes na janela Banco de Dados, e o arraste para a seção detalhe do formulário Encontrar Clientes.  
b) Exclua a etiqueta anexada ao controle de subformulário.

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access"

Criar  
formulários

Basic"  
*Guia do Usuário*,  
capítulo 15,  
"Estruturando  
formulários",  
capítulo 16,  
"Personalizando  
formulários, e  
capítulo 17,  
"Criando  
formulários  
baseados em mais  
de uma tabela"



### **Procedimento do evento Abrir para o formulário Encontrar Clientes**

```
Sub Form_Open (Cancel As Integer)
```

```
    ' Move o ponto de inserção para a caixa de texto "Procurar por  
    companhia" quando o formulário é aberto.
```

```
    Me![Procurar por companhia].SetFocus
```

```
End Sub
```



### **sub AdicionarAWhere**

```
Sub AdicionarAWhere (FieldValue As Variant, FieldName As String, MyCriteria  
As String, ArgCount As Integer)
```

```
    ' Cria critério para a cláusula WHERE.
```

```
    If FieldValue <> "" Then
```

```
        ' Adiciona "and" se existir outro critério.
```

```
        If ArgCount > 0 Then
```

```
            MyCriteria = MyCriteria & " and "
```

```
        End If
```

```
        ' Anexa o critério ao critério já existente.
```

```
        ' Coloca FieldValue e o asterisco entre aspas.
```

```
        MyCriteria = (MyCriteria & FieldName & " Like " & Chr(39) &  
FieldValue & Chr(42) & Chr(39))
```

```
        ' Aumenta a contagem de argumentos.
```

```
        ArgCount = ArgCount + 1
```

```
    End If
```

```
End Sub
```



### **sub DesativarControle**

```
Sub DesativarControle ()
    ' Se ativado, desativa o controle na seção detalhe depois de alterar o
    critério de procura.
    Dim Tmp As Variant
    If Me![Subformulário Encontrar Clientes].Enabled Then
        Tmp = AtivarControles("Detail", False)
    End If
End Sub
```



### **Procedimento do evento DepoisDeAtualizar para a caixa de texto "Procurar por companhia"**

```
Sub Procurar_por_companh_AfterUpdate ()  
    ' Desativa o controle na seção detalhe depois de alterar o critério de  
    procura.  
    DesativarControle  
End Sub
```



### **Procedimento do evento DepoisDeAtualizar para a caixa de texto "Procurar por contato"**

```
Sub Procurar_por_contato_AfterUpdate ()  
    ' Desativa o controle na seção detalhe depois de alterar o critério de  
    procura.  
    DesativarControle  
End Sub
```



### **Procedimento do evento DepoisDeAtualizar para a caixa de texto "Procurar por cidade"**

```
Sub Procurar_por_cidade_AfterUpdate ()  
    ' Desativa o controle na seção detalhe depois de alterar o critério de  
    procura.  
    DesativarControle  
End Sub
```



### **Procedimento do evento DepoisDeAtualizar para a caixa de texto "Procurar por país"**

```
Sub Procurar_por_país_AfterUpdate ()  
    ' Desativa o controle na seção detalhe depois de alterar o critério de  
    procura.  
    DesativarControle  
End Sub
```



## Procedimento do evento Clicar para o botão de comando "Mostrar clientes"

```
Sub Mostrar_clientes_Click ()
    ' Cria uma cláusula WHERE usando critérios de procura inseridos pelo
    usuário e
    ' define a propriedade OrigemDoRegistro do subformulário Encontrar
    Clientes.
    Dim MySQL As String, MyCriteria As String, MyRecordSource As String
    Dim ArgCount As Integer
    Dim Tmp As Variant
    ' Inicializa a contagem de argumentos.
    ArgCount = 0
    ' Inicializa instrução SELECT.
    MySQL = "SELECT * FROM Clientes WHERE "
    MyCriteria = ""
    ' Usa valores inseridos nas caixas de texto do cabeçalho do formulário
    para criar critérios para a cláusula WHERE.
    AdicionarAWhere [Procurar por companhia], "[Nome da companhia]",
    MyCriteria, ArgCount
    AdicionarAWhere [Procurar por contato], "[Nome do contato]", MyCriteria,
    ArgCount
    AdicionarAWhere [Procurar por cidade], "[Cidade]", MyCriteria, ArgCount
    AdicionarAWhere [Procurar por país], "[País]", MyCriteria, ArgCount
    ' Se não há critério especificado, retorna todos os registros.
    If MyCriteria = "" Then
        MyCriteria = "True"
    End If
    ' Cria instrução SELECT.
    MyRecordSource = MySQL & MyCriteria
    ' Define a propriedade OrigemDoRegistro de Subformulário Encontrar
    Clientes.
    Me![Subformulário Encontrar Clientes].Form.RecordSource = MyRecordSource
    ' Se nenhum registro corresponder ao critério, exibe mensagem.
    ' Move o foco para o botão Limpar.
    If Me![Subformulário Encontrar Clientes].Form.RecordsetClone.RecordCount
    = 0 Then
        MsgBox "Nenhum registro corresponde ao critério que você inseriu.",
        48, "Nenhum registro encontrado"
        Me!Limpar.SetFocus
    Else
        ' Ativa controle na seção detalhe.
        Tmp = AtivarControles("Detail", True)
        ' Move o ponto de inserção para o Subformulário Encontrar Produtos.
        Me![Subformulário Encontrar Clientes].SetFocus
    End If
End Sub
```

## Limpendo os valores em caixas de texto independentes e em um subformulário

Em um formulário onde você possa usar os valores de vários campos para procurar registros, você pode querer limpar os campos de procura antes de inserir seu próximo conjunto de critérios. Para evitar confundir os registros retornados pelo conjunto de critérios anterior com os que possam ser retornados pelo conjunto atual, você pode querer limpar os registros do subformulário. No formulário Encontrar Clientes, clicar no botão Limpar limpa todos os campos de procura do cabeçalho do formulário e limpa os registros do subformulário na seção Detalhe.

### Visão geral

Crie o botão de comando Limpar no cabeçalho do formulário. Na propriedade AoClicar do botão de comando, crie um procedimento de evento que defina cada caixa de texto do cabeçalho de formulário como nula, reinicie a propriedade OrigemDoRegistro do formulário, e mova o ponto de inserção para a primeira caixa de texto no cabeçalho do formulário.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Tabela	Clientes
Formulário	Encontrar Clientes, Subformulário Encontrar Clientes

### Passo a passo

#### 1 Crie o formulário Encontrar Clientes.

#### 2 No cabeçalho do formulário Encontrar Clientes, crie um botão de comando, e defina suas propriedades como a seguir.

<u>Propriedade</u>	<u>Valor</u>
Nome	Limpar
Legenda	Limpar
AoClicar	<u>[Procedimento de evento]</u>

#### 3 Modifique o procedimento de evento da propriedade AoClicar do botão de comando Mostrar clientes.

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", capítulo 16,

"Personalizando  
formulários"



### **Procedimento do evento Clicar para o botão de comando Limpar**

```
Sub Limpar_Click ()
    ' Limpa controles no cabeçalho do formulário e remove registros do
    subformulário.
    '
    Dim MySQL As String
    Dim Tmp As Variant
    MySQL = "SELECT * FROM Clientes WHERE False"
    ' Limpa caixas de texto de procura.
    Me![Procurar por companhia] = Null
    Me![Procurar por contato] = Null
    Me![Procurar por cidade] = Null
    Me![Procurar por país] = Null
    ' Reinicia a propriedade OrigemDoRegistro do subformulário para remover
    registros.
    Me![Subformulário Encontrar Clientes].Form.RecordSource = MySQL
    ' Move o ponto de inserção para a caixa de texto "Procurar por
    companhia".
    Me![Procurar por companhia].SetFocus
End Sub
```



## Procedimento do evento Clicar para o botão de comando "Mostrar clientes"

O texto e o código que você precisa adicionar ao procedimento de evento para definir o foco para o botão Limpar são exibidos em vermelho.

```
Sub Mostrar_clientes_Click ()
    ' Cria uma cláusula WHERE usando critérios de procura inseridos pelo
    usuário e
    ' define a propriedade OrigemDoRegistro do subformulário Encontrar
    Clientes.
    Dim MySQL As String, MyCriteria As String, MyRecordSource As String
    Dim ArgCount As Integer
    Dim Tmp As Variant
    ' Inicializa a contagem de argumentos.
    ArgCount = 0
    ' Inicializa instrução SELECT.
    MySQL = "SELECT * FROM Clientes WHERE "
    MyCriteria = ""
    ' Usa valores inseridos nas caixas de texto do cabeçalho do formulário
    para criar critérios para a cláusula WHERE.
    AdicionarAWhere [Procurar por companhia], "[Nome da companhia]",
    MyCriteria, ArgCount
    AdicionarAWhere [Procurar por contato], "[Nome do contato]", MyCriteria,
    ArgCount
    AdicionarAWhere [Procurar por cidade], "[Cidade]", MyCriteria, ArgCount
    AdicionarAWhere [Procurar por país], "[País]", MyCriteria, ArgCount
    ' Se não há critério especificado, retorna todos os registros.
    If MyCriteria = "" Then
        MyCriteria = "True"
    End If
    ' Cria instrução SELECT.
    MyRecordSource = MySQL & MyCriteria
    ' Define a propriedade OrigemDoRegistro de Subformulário Encontrar
    Clientes.
    Me![Subformulário Encontrar Clientes].Form.RecordSource = MyRecordSource
    ' Se nenhum registro corresponder ao critério, exibe mensagem.
    ' Move o foco para o botão Limpar.
    If Me![Subformulário Encontrar Clientes].Form.RecordsetClone.RecordCount
    = 0 Then
        MsgBox "Nenhum registro corresponde ao critério que você inseriu.",
        48, "Nenhum registro encontrado"
        Me!Limpar.SetFocus
    Else
        ' Ativa controle na seção detalhe.
        Tmp = AtivarControles("Detail", True)
        ' Move o ponto de inserção para o Subformulário Encontrar Produtos.
        Me![Subformulário Encontrar Clientes].SetFocus
    End If
End Sub
```



## Destaques do formulário Soluções

O formulário Soluções mostra como acessar todos os exemplos do aplicativo de exemplo Soluções. Esta técnica faz o formulário Soluções funcionar:



Abrindo um formulário ou relatório a partir de um formulário principal.

Esta técnica aperfeiçoa o formulário:



Fechando todos os objetos de banco de dados quando você sai do aplicativo

## Abrindo um formulário ou relatório a partir de um formulário principal

Se existem muitos formulários e relatórios em um aplicativo, você pode querer proporcionar um formulário principal que os organize por tarefa ou função executada. O formulário Soluções é o formulário principal do aplicativo de exemplo Soluções. Ele agrupa os exemplos por categoria de desenvolvimento de aplicativos.

### Visão geral

Crie um formulário independente. Crie uma caixa de listagem que liste as categorias. Crie uma segunda caixa de listagem que liste os exemplos pertencentes à segunda categoria. Crie um botão de comando. Em sua propriedade AoClicar, crie um procedimento de evento que use a propriedade Coluna para abrir o formulário ou relatório de exemplo da caixa de listagem de exemplos.

### Objetos usados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Objetos de exemplo, Tópicos de exemplo, Exemplos
Consulta	Lista de exemplos, Lista de tópicos de exemplo
Formulário	Soluções

### Passo a passo

#### 1 Crie a consulta Lista de Tópicos de Exemplo.

<b>Tabela</b>	<b>Campo (Ascendente)</b>	<b>Exibe</b>
Tópicos de exemplo	Identificação do tópico	Sim
	Descrição	Sim
	Ordem de classificação (Ascendente)	Não

#### 2 Crie a consulta Lista de Exemplos.

<b>Tabela</b>	<b>Campo (Ascendente)</b>	<b>Exibe</b>	<b>Critério</b>
Exemplos	Descrição (Ascendente)	Sim	--
Objetos de exemplo	Nome do objeto	Sim	--
	Tipo de objeto	Sim	--
	Identificação de tópico	Não	Forms![Soluções]! [Selecionar tópico]

#### 3 Estruture o formulário Soluções.

a) Crie um formulário independente, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Legenda	Soluções
ModoPadrão	Formulário simples

VisualizaçãoAutoriza da	Formulário
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim

b) Crie uma caixa de listagem independente, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Selecionar tópico
TipoDeOrigemDaLinha	Tabela/consulta
OrigemDaLinha	Lista de Tópicos de Exemplo
NúmeroDeColunas	2
LargurasDasColunas	0 cm;12,699 cm
ColunaVinculada	1
TextoDaBarraDeStatus	Selecione uma categoria de exemplos.
DepoisDeAtualizar	[ <u>Procedimento de evento</u> ]

c) Crie uma outra caixa de listagem independente, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Selecionar exemplo
TipoDeOrigemDaLinha	Tabela/consulta
OrigemDaLinha	Lista de Exemplos
NúmeroDeColunas	3
LargurasDasColunas	12,699 cm; 0 cm; 0 cm
ColunaVinculada	2
TextoDaBarraDeStatus	Selecione um exemplo.
EmDuploClique	[ <u>Procedimento de evento</u> ]

d) Crie um botão de comando, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	OK
Legenda	OK
Padrão	Sim
AoClicar	[ <u>Procedimento de evento</u> ]

## Referências cruzadas

**Para informações**

**Consulte**

**sobre**

---

Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas sobre o Access Basic"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", e capítulo 16, "Personalizando formulários"
Criar consultas	<i>Guia do Usuário</i> , capítulo 11, "Estruturando consultas seleção"



### **Procedimento do evento DepoisDeAtualizar da caixa de listagem "Selecionar t3pico"**

```
Sub Selecionar_t3pico_AfterUpdate ()  
    ' Reconsulta a caixa de listagem Selecionar exemplo.  
    Me![Selecionar exemplo].Requery  
End Sub
```



### **Procedimento do evento DuploClique da caixa de listagem Selecionar Exemplo**

```
Sub Selecionar_exemplo_DblClick (Cancel As Integer)
    ' Executa o procedimento para evento em sub OK_Click.
    OK_Click
End Sub
```



## Procedimento do evento Clique do botão de comando OK

```
Sub OK_Click ()
    ' Abre o formulário ou relatório do exemplo selecionado.
    ' Só o Soluções: Exibe o tópico Mostre me se a caixa de verificação
Mostre me estiver selecionada.
    '
    ' Abre o formulário do exemplo selecionado.
    If Me![Selecionar exemplo].Column(2) = "Form" Then
        ' Usado pelo Soluções para abrir o formulário e exibir seu tópico
Mostre me.
        If Me![Mostre me] = True Then
            DoCmd OpenForm Forms![Soluções]![Selecionar exemplo].Column(1), ,
, , 0
            SendKeys "{F1}", True
            ' Abre o formulário.
        Else
            DoCmd OpenForm Forms![Soluções]![Selecionar exemplo].Column(1), ,
, , 0
        End If
        ' Abre o relatório do exemplo selecionado.
    ElseIf Me![Selecionar exemplo].Column(2) = "Report" Then
        ' Usado pelo Soluções para abrir o relatório e exibir seu tópico
Mostre me.
        If Me![Mostre me] = True Then
            DoCmd OpenReport Forms![Soluções]![Selecionar exemplo].Column(1),
A_PREVIEW
            SendKeys "{F1}", True
            ' Abre o relatório.
        Else
            DoCmd OpenReport Forms![Soluções]![Selecionar exemplo].Column(1),
A_PREVIEW
        End If
        ' Usado pelo Soluções para abrir a caixa de diálogo para o relatório
Vendas de Empregados.
    ElseIf Me![Selecionar exemplo].Column(2) = "Dialog" Then
        DoCmd OpenForm "Caixa de Diálogo Vendas de Empregados"
    End If
End Sub
```

## Fechando todos os objetos de banco de dados ao sair de um aplicativo

Quando você sai de seu aplicativo, todos os objetos de banco de dados abertos podem ser fechados automaticamente. Se você tiver feito quaisquer alterações a qualquer objeto, você pode fazer com que o seu aplicativo peça a você para salvar ou descartar as alterações. Por exemplo, quando você clica no botão Sair do formulário Soluções, o Soluções tenta fechar todos os objetos abertos. Se você tiver alterado a estrutura de um objeto, o Microsoft Access pedirá que você salve ou descarte as alterações ou que você cancele. Se você clicar em Cancelar, o formulário Soluções exibirá uma mensagem dizendo que você precisa salvar ou descartar as alterações.

### Visão geral

Crie uma função que feche quaisquer objetos abertos. Adicione um botão de comando ao formulário. Crie um procedimento de evento na propriedade AoClicar que feche cada tipo de objeto de banco de dados e então feche o banco de dados.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nome</u>
Formulário	Soluções

### Passo a passo

**1 Na seção Geral do módulo Formulário.Soluções, crie a função FecharObjeto.**

**2 Na seção Declarações do módulo Formulário.Soluções, digite:**

```
Option Explicit
```

**3 Adicione um botão de comando ao formulário Soluções e defina suas propriedades como a seguir.**

<u>Propriedade</u>	<u>Definição</u>
Nome	Sair
Legenda	Sair
AoClicar	<u>[Procedimento de evento]</u>

---

**Observação** O procedimento de evento Clique para o botão de comando Sair não funciona quando você descarta alterações em um módulo. Quando você opta por não salvar alterações em um módulo, o Microsoft Access fecha a janela do módulo mas suspende o procedimento do evento Clique do botão de comando Sair, e você próprio precisa fechar o banco de dados.

---

### Referências cruzadas

<u>Para informações sobre</u>	<u>Consulte</u>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas do Access Basic"
Criar formulários	capítulo 16, "Personalizando formulários"





### **Função FecharObjeto()**

```
Function FecharObjeto (ContainerName As String, ContainerType As Integer)
```

```
    ' Fecha os objetos de banco de dados abertos.  
    Dim MyDatabase As Database, MyContainer As Container  
    Dim I As Integer  
    Set MyDatabase = DBEngine.Workspaces(0).Databases(0)  
    Set MyContainer = MyDatabase.Containers(ContainerName)  
    For I = 0 To MyContainer.Documents.Count - 1  
        DoCmd Close ContainerType, MyContainer.Documents(I).Name  
    Next I  
End Function
```



## Procedimento do evento Clicar do botão de comando Sair

```
Sub Sair_Click ()
' Fecha todos os objetos de banco de dados abertos, e então sai do
aplicativo.
'
' Ativa a interceptação de erros.
On Error GoTo ErrHandler
    Dim Tmp As Variant
    Dim ReturnValue As Integer
    Tmp = FecharObjeto("Tables", A_TABLE)
    Tmp = FecharObjeto("Tables", A_QUERY)
    Tmp = FecharObjeto("Forms", A_FORM)
    Tmp = FecharObjeto("Reports", A_REPORT)
    Tmp = FecharObjeto("Scripts", A_MACRO)
    Tmp = FecharObjeto("Modules", A_MODULE)
    ' Sai do aplicativo.
    DoCmd DoMenuItem 1, 0, 2, , A_MENU_VER20
    Exit Sub
' Desvia para aqui se ocorrer um erro.
ErrHandler:
    MsgBox "Você precisa salvar ou descartar as alterações no objeto
aberto.", 48, "Precisa fechar o objeto"
    Resume Next
End Sub
```

## **Destaques do formulário Totais de Vendas**

O formulário Totais de Vendas mostra como exibir progressivamente mais detalhes sobre as vendas de um empregado. Esta técnica faz o formulário Totais de Vendas funcionar:



Criando um formulário um por vários por vários.

## Criando um formulário um por vários por vários

Quando você examina os dados, você costuma querer ter uma visão de cima bem como uma visão detalhada. No formulário Totais de Vendas, você pode revisar os totais de vendas anuais do vendedor selecionado e então visualizar seus totais de vendas trimestrais para o ano selecionado.

### Visão geral

Crie um subformulário que exiba totais anuais e um segundo subformulário que exiba totais trimestrais. Crie um formulário principal vinculado à tabela Empregados. No cabeçalho do formulário principal, crie uma caixa de combinação independente a partir da qual você encontre registros para o empregado selecionado. Na seção detalhe do formulário principal, crie uma caixa de texto independente que vincule o segundo subformulário ao primeiro subformulário. Defina a propriedade OrigemDoControle da caixa de texto com uma expressão que exiba o valor a partir de Ano fiscal, o campo de vinculação no primeiro subformulário. Defina ambas as propriedades VincularAPartirDe e VincularA do segundo subformulário com os campos que identificam o empregado e o ano – Identificação do empregado e Ano fiscal.

### Objetos usados neste exemplo

<b>Objeto</b>	<b>Nomes</b>
Tabela	Empregados, Pedidos, Detalhes do Pedido
Consulta	Totais de Vendas (Trim), Totais de Vendas (Ano), Vendedores
Formulário	Totais de Vendas, Subformulário Totais de Vendas 1, Subformulário Totais de Vendas 2
Módulo	Ativar Controles

### Passo a passo

#### 1 Crie a consulta Vendedores.

<b>Tabela</b>	<b>Campo/ expressão</b>	<b>Classificar</b>
Empregados	Vendedor: [Sobrenome] & ", " & [Primeiro nome]	Ascendente
	Identificação do empregado	--
Pedidos	--	--

---

**Observação** Você não usa nenhum campo da tabela Pedidos nesta consulta; você só usa a tabela Pedidos para limitar os nomes dos empregados retornados aos que têm registros na tabela Pedidos.

---

#### 2 Crie a consulta Totais de Vendas (Ano).

<b>Tabela</b>	<b>Campo/expressão</b>	<b>Total</b>	<b>Critério</b>
Pedidos	Número do empregado	Agrupar por	--

	Ano Fiscal: Ano([Data de envio])	Agrupar por	É Negado Nulo
Detalhes do Pedido	Total: CMoeda(Soma(CLog([Preço unitário]*[Quantidade]*(1-[Desconto])*100)/100))	Expressão	--

### 3 Crie a consulta Totais de Vendas (Trim).

<u>Tabela</u>	<u>Campo/expressão</u>	<u>Total</u>	<u>Critério</u>
Pedidos	Número do empregado	Agrupar por	--
	Ano Fiscal: Ano([Data de envio])	Agrupar por	É Negado Nulo
	Trimestre: PartData("t",[Data de envio])	Agrupar por	É Negado Nulo
Detalhes do Pedido	Subtotal: CMoeda(Soma(CLog([Preço Unitário]*[Quantidade]*(1-[Desconto])*100/100))	Expressão	--

### 4 Estruture o Subformulário Totais de Vendas 1.

a) Defina as seguintes propriedades do formulário.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	Totais de Vendas (Ano)
ModoPadrão	Folha de Dados

b) Crie duas caixas de texto na seção Detalhe, e defina suas propriedades Nome e OrigemDoControle como a seguir.

<u>Nome</u>	<u>OrigemDoControle</u>
Ano fiscal	Ano fiscal
Total	Total

c) Alterne para modo Folha de Dados. Ajuste as larguras das colunas, e salve o formulário.

### 5 Estruture o subformulário Totais de Vendas 2.

a) Defina as seguintes propriedades do formulário.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	Totais de Vendas (Trim)
ModoPadrão	Folha de Dados

b) Crie duas caixas de texto na seção detalhe, e defina suas propriedades Nome e OrigemDoControle como a seguir.

<u>Nome</u>	<u>OrigemDoControle</u>

Trimestre	Trimestre
Subtotal	Subtotal

c) Alterne para modo Folha de Dados. Ajuste as larguras das colunas, e salve o formulário.

## 6 Crie o módulo Ativar Controles, se você já não o tiver feito.

a) Na seção Declarações, digite:

Option Explicit

b) Crie a função Ativar Controles.

## 7 Estructure o formulário Totais de Vendas.

a) Defina as seguintes propriedades de formulário.

<b>Propriedade</b>	<b>Definição</b>
OrigemDoRegistro	Empregados
ModoPadrão	Formulário Simples
Visualização Autorizada	Formulário
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim

b) Adicione um cabeçalho e um rodapé de formulário. Defina a propriedade Altura do rodapé do formulário como 0.

c) No cabeçalho do formulário, crie uma caixa de combinação independente, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Selecionar vendedor
TipoDeOrigemDaLinha	Tabela/consulta
OrigemDaLinha	Vendedores
NúmeroDeColunas	2
LargurasDasColunas	5,08 cm; 0 cm
ColunaVinculada	2
LimitarAUmaLista	Sim
DepoisDeAtualizar	[ <u>Procedimento de evento</u> ]

d) Na seção Declarações do módulo Formulário.Totais de Vendas, digite:

Option Explicit

e) Na seção detalhe, crie uma caixa de texto, e desanexe sua etiqueta. Defina então as propriedades da caixa de texto como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Vendedor
OrigemDoControle	=[Sobrenome] &","&[Primeiro nome]

Ativado	Não
ParadaDeTabulaçã o	Não
CorDoFundo	8388608 (azul escuro)
EstiloDaBorda	Limpar
CorDeTexto	16777215 (branco)

f) Na seção detalhe, crie uma caixa de texto oculta com suas propriedades definidas como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Ano fiscal
OrigemDoControl e	=[Subformulário Totais de Vendas 1].Formulário![Ano fiscal]
Visível	Não
ParadaDeTabulaçã o	Não

---

**Observação** O Microsoft Access usa o valor desta caixa de texto para vincular os valores do Subformulário Totais de Vendas 2 à linha selecionada no subformulário Totais de Vendas 1.

---

### **8 Adicione o Subformulário Totais de Vendas 1 ao formulário Totais de Vendas.**

- Selecione o Subformulário Totais de Vendas 1 na janela Banco de Dados, e arraste-o para a seção detalhe do formulário Totais de Vendas.
- Altere a etiqueta do controle do subformulário para **Selecionar ano:** e ajuste o tamanho do controle do subformulário.
- Verifique que as propriedades VincularAPartirDe e VincularA do controle Subformulário Totais de Vendas 1 estejam ambas definidas como Identificação do empregado.
- Defina a propriedade Ativado do controle Subformulário Totais de Vendas 1 como Não.

### **9 Adicione o Subformulário Totais de Vendas 2 ao formulário Totais de Vendas.**

- Selecione o subformulário Totais de Vendas 2 na janela Banco de Dados, e arraste-o para a seção detalhe do formulário Totais de Vendas.
- Altere a etiqueta do controle de subformulário para **Totais trimestrais para o ano selecionado:** e ajuste o tamanho do controle de subformulário.
- Defina ambas as propriedades VincularAPartirDe e VincularA do controle Subformulário Totais de Vendas 2 como: **Ano fiscal;Identificação do empregado**
- Defina a propriedade Ativado do controle subformulário Totais de Vendas 2 como Não.

### **Referências cruzadas**

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de evento	<i>Criando Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções Básicas sobre o Access"

Criar formulários	Basic" <i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", capítulo 16, "Personalizando formulários", e capítulo 17, "Criando formulários com base em mais de uma tabela"
Criar consultas	<i>Guia do Usuário</i> , capítulo 11, "Estruturando consultas seleção"



### **Procedimento de evento DepoisDeAtualizar para a caixa de combinação "Selecionar vendedor"**

```
Sub Selecionar_vendedor_AfterUpdate ()
    ' Encontra registros para o vendedor selecionado na caixa de combinação
    "Vendedor".
    ' Ativa controles da seção detalhe.
    ' Vai para o Subformulário Totais de Vendas 1.
    Dim Tmp As Variant
    DoCmd ApplyFilter , "[Número do empregado] = Forms![Totais de Vendas]!
[Selecionar vendedor]"
    Tmp = AtivarControles("Detail", True)
    Me![Subformulário Totais de Vendas 1].SetFocus
End Sub
```

## **Destaques do formulário Mostrar Vendas**

Mostrar Vendas é uma consulta por formulário que mostra como usar caixas de texto e um grupo de opção para especificar critérios. Esta técnica faz o formulário Mostrar Vendas funcionar:



Usando uma função personalizada em uma instrução SQL SELECT.

## Usando uma função personalizada na instrução SELECT (SQL) de um formulário

Em uma consulta por formulário, você pode querer que os usuários escolham uma opção a partir de um conjunto específico de critérios. No formulário Mostrar Vendas, você pode especificar um intervalo de datas e se você deseja ver todos os registros no intervalo de datas ou registros no intervalo de datas onde as vendas tenham sido menores ou maiores que 1000.

### Visão geral

Crie uma consulta por formulário onde você insira critérios em caixas de texto e um grupo de opção no cabeçalho do formulário principal, e que exiba registros em um subformulário na seção detalhe. Crie uma função que retorne o valor selecionado no grupo de opção. Crie um botão de comando no cabeçalho de formulário do formulário principal. Na propriedade AoClicar do botão de comando, crie um procedimento de evento que crie uma instrução SQL SELECT que use a função personalizada para retornar o critério do usuário e defina a propriedade OrigemDoRegistro do subformulário como a instrução SELECT.

### Objetos usados neste exemplo

<u>Objeto</u>	<u>Nomes</u>
Tabela	Clientes, Pedidos
Consulta	Subtotais dos Pedidos, Mostrar Vendas
Formulário	Mostrar Vendas, Subformulário Mostrar Vendas
Módulo	Ativar Controles

### Passo a passo

**1 Importe a consulta Subtotais dos Pedidos de NWIND.MDB, se você já não o tiver feito.**

**2 Crie a consulta Mostrar Vendas.**

<u>Tabela/consulta</u>	<u>Campos</u>	<u>Classificar</u>
Clientes	Nome da companhia	--
Pedidos	Data de envio	--
Subtotais dos Pedidos	Número do pedido	--
	Subtotal	Descendente

**3 Crie o Subformulário Mostrar Vendas.**

a) Defina as seguintes propriedades de formulário.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	Mostrar Vendas
ModoPadrão	Folha de Dados
VisualizaçãoAutorizada	Folha de Dados
MenuDeAtalho	Não

b) Crie caixas de texto para os seguintes campos da seção detalhe: Nome da companhia, Número do pedido e Subtotal.

c) Alterne para modo Folha de Dados. Ajuste as larguras das colunas, e salve o

formulário.

- d) Reinicie a propriedade OrigemDoRegistro do formulário com a seguinte instrução SQL:  
SELECT [Nome da companhia] FROM Clientes WHERE False

---

**Observação** Definir a propriedade OrigemDoRegistro com essa instrução SQL irá exibir o subformulário como uma folha de dados vazia quando o formulário Mostrar Vendas for aberto.

---

#### 4 Crie o formulário Mostrar Vendas.

- a) Defina as seguintes propriedades.

<u>Propriedade</u>	<u>Definição</u>
OrigemDoRegistro	--
Legenda	Mostrar Vendas
ModoPadrão	Formulário simples
VisualizaçãoAutorizada	Formulário
MenuDeAtalho	Não
BarrasDeRolagem	Nenhum
ExibirSeletor	Não
BotõesDeNavegação	Não
AutoCentrar	Sim
AoAbrir	<u>[Procedimento de evento]</u>

- b) Selecione o Subformulário Mostrar Vendas na janela Banco de Dados, e o arraste para a seção detalhe do formulário Mostrar Vendas. Ajuste o tamanho e lugar do controle de subformulário, e remova sua etiqueta anexa. (Deixe as propriedades VincularAPartirDe e Vincular A em branco).
- c) Adicione um cabeçalho e um rodapé de formulário. Defina a propriedade Altura do rodapé do formulário como 0.
- d) No cabeçalho do formulário, crie duas caixas de texto independentes, e defina suas propriedades Nome e Formato como a seguir.

<u>Nome</u>	<u>Formato</u>
Data inicial	Data média
Data de término	Data média

- e) Digite o nome de cada caixa de texto na propriedade Legenda de sua etiqueta anexada.
- f) No cabeçalho do formulário, crie um grupo de opção independente, e defina sua propriedade Nome como Vendas e sua propriedade ValorPadrão como 1.
- g) Digite **Vendas** na propriedade Legenda da etiqueta anexada do grupo de opção.
- h) Crie três botões de opção no grupo de opção Vendas, e defina suas propriedades Nome e ValorDeOpção como a seguir.

<u>Nome</u>	<u>ValorDeOpção</u>
Menores que 1000	1
Maiores que 1000	2

- i) Digite o nome de cada botão de opção na propriedade Legenda de sua etiqueta anexada.

### 5 Modifique a consulta **Mostrar Vendas**.

- a) Especifique o seguinte critério para o campo Data de envio:  
Entre Forms![Mostrar Vendas]![Data inicial] e [Forms]![Mostrar Vendas]![Data de término]
- b) Para o campo Subtotal, especifique ordem de classificação descendente.

---

**Dica** Você pode copiar a instrução SQL a partir da janela SQL da consulta para usar no procedimento de evento do botão de comando criado no passo 8.

---

### 6 Na seção **Geral do módulo Formulário.Mostrar Vendas**, crie a **função Mostrar Vendas**.

### 7 Na seção **Declarações do módulo Formulário.Mostrar Vendas**, digite:

Option Explicit

### 8 Crie o módulo **Ativar Controles**, e você já não o tiver feito.

- a) Na seção Declarações do módulo, digite:

Option Explicit

- b) Crie a **função AtivarControles**.

### 9 No cabeçalho do formulário **Mostrar Vendas**, crie um botão de comando, e defina suas propriedades como a seguir.

<b>Propriedade</b>	<b>Definição</b>
Nome	Mostrar Vendas
Legenda	Mostrar Vendas
Padrão	Sim
AoClicar	[ <u>Procedimento de evento</u> ]

### Referências cruzadas

<b>Para informações sobre</b>	<b>Consulte</b>
Criar procedimentos de eventos	<i>Construindo Aplicativos</i> , capítulo 3, "Introdução ao Access Basic", e capítulo 5, "Noções básicas do Access"
Criar formulários	<i>Guia do Usuário</i> , capítulo 15, "Estruturando formulários", e capítulo 16, "Personalizando formulários"
Criar consultas	<i>Guia do Usuário</i> , capítulo 10, "Fundamentos de consultas" e capítulo 11,

"Estruturando  
consultas seleção"



### **Procedimento do evento Abrir para o formulário Mostrar Vendas**

```
Sub Form_Open (Cancel As Integer)
    ' Move o ponto de inserção para o campo Data inicial.
    Me![Data inicial].SetFocus
    ' Usado pelo Soluções para retornar registros por exemplo.
    Me![Data inicial] = #2/01/93#
    Me![Data de término] = #16/01/93#
End Sub
```



### **função MostrarVendas()**

```
Function MostrarVendas (OptionGroupValue As Long) As String
    ' Retorna o valor selecionado no grupo de opção Vendas.
    '
    ' Define constantes para valores do grupo de opção.
    Const SALES_UNDER_1000 = 1
    Const SALES_OVER_1000 = 2
    Const ALL_SALES = 3
    ' Cria restrição baseada no valor do grupo de opção.
    Select Case OptionGroupValue
        Case SALES_UNDER_1000:
            MostrarVendas = "[Subtotais dos Pedidos].Subtotal < 1000"
        Case SALES_OVER_1000:
            MostrarVendas = "[Subtotais dos Pedidos].Subtotal >= 1000"
        Case Else
            MostrarVendas = "[Subtotais dos Pedidos].Subtotal = True"
    End Select
End Function
```



## Procedimento do evento Clicar para o botão de comando Mostrar Vendas

```
Sub Mostrar_vendas_Click ()
    ' Cria uma instrução SQL usando critérios de procura inseridos pelo
    usuário e
    ' define a propriedade OrigemDoRegistro do Subformulário Mostrar Vendas.
    Dim MySQL As String, Restrict As String
    Dim x As Long
    Dim Tmp As Variant
    x = Forms![Mostrar Vendas]![Vendas].Value
    Restrict = MostrarVendas(x)
    ' Cria instrução SELECT.
    MySQL = "SELECT DISTINCTROW Clientes.[Nome da companhia], [Subtotais dos
    Pedidos].[Número do pedido], "
    MySQL = MySQL & "[Subtotais dos Pedidos].Subtotal , Pedidos.[Data de
    envio]"
    MySQL = MySQL & "FROM Clientes INNER JOIN ([Subtotais dos Pedidos] INNER
    JOIN Pedidos ON "
    MySQL = MySQL & "[Subtotais dos Pedidos].[Número do pedido] = Pedidos.
    [Número do pedido]) ON "
    MySQL = MySQL & "Clientes.[Código do cliente] = Pedidos.[Código do
    cliente] "
    MySQL = MySQL & "WHERE (Pedidos.[Data de envio] Between [Forms]![Mostrar
    Vendas]![Data inicial] "
    MySQL = MySQL & "And [Forms]![Mostrar Vendas]![Data de término]) "
    MySQL = MySQL & "And " & Restrict
    MySQL = MySQL & " ORDER BY [Subtotais dos Pedidos].Subtotal DESC;"
    ' Define a propriedade OrigemDoRegistro do Subformulário Mostrar Vendas.
    Me![Subformulário Mostrar Vendas].Form.RecordSource = MySQL
    ' Se nenhum registro corresponder ao critério, reinicia a propriedade
    OrigemDoRegistro do subformulário,
    ' exibe a mensagem, e move o foco para a caixa de texto "Data inicial".
    If Me![Subformulário Mostrar Vendas].Form.RecordsetClone.RecordCount = 0
    Then
        Me![Subformulário Mostrar Vendas].Form.RecordSource = "SELECT [Nome
        da companhia] FROM Clientes WHERE False;"
        MsgBox "Nenhum registro corresponde ao critério que você inseriu.",
        48, "Nenhum registro encontrado"
        Me![Data inicial].SetFocus
    Else
        ' Ativa controle na seção detalhe.
        Tmp = AtivarControles("Detail", True)
        ' Move o ponto de inserção para o Subformulário Mostrar Vendas.
        Me![Subformulário Mostrar Vendas].SetFocus
    End If
End Sub
```

