

# Access ListBar

- Artigo para Access 2000, XP e 2003
- Pré-requisitos: Conhecimentos de Access e VBA

Paulo Sergio Sarraino\*

*A LISTBAR (OU MENU VERTICAL) APARECEU COM ALGUMAS VERSÕES MAIS ANTIGAS DO OUTLOOK. O CONCEITO DELA É ABRIGAR FUNÇÕES SEMELHANTES DENTRO DE GRUPOS QUE, QUANDO CLICADOS, ABREM-SE VERTICALMENTE PARA ACESSAR AS FUNÇÕES RELACIONADAS. COMO UMA BARRA DE MENU, PORÉM NA VERTICAL, EXIBE TEXTOS E ÍCONES COM CONFIGURAÇÕES DIFERENTES E ESSES ELEMENTOS PODEM SER REORDENADOS E REAGRUPADOS LIVREMENTE, A GOSTO DO USUÁRIO.*

Logo, sua utilidade foi percebida para sistemas em Access que administram grande quantidade de tipos de dados. Assim, esses tipos poderiam ser agrupados e disponibilizados numa interface mais simples para manutenção dos dados. Mas esse componente nunca foi inserido no Access (até a versão 2003) para uso em sistemas construídos nessa plataforma.

Surgiram, então, através de empresas de ferramentas de software, componentes semelhantes que permitem explorar esse recurso no Access incorporando-os aos sistemas desenvolvidos. Pesquisando na internet e no GD da FórumAccess, localizei alguns deles com custos tanto para o desenvolvimento quanto para o uso desses componentes.

Dai pensei: é possível desenvolver algo semelhante apenas usando os objetos do Access? Mãos à obra, pesquisas, testes, e hoje posso dizer: sim! Com algumas limitações, porém bem funcional.

Nesta matéria vou demonstrar isso com 2 exemplos: um usando apenas textos (com uma caixa de listagem) e outro usando ícones/textos (com um subformulário).

No arquivo disponibilizado para essa matéria há, além do mdb de demonstração, diversos bmp's para os ícones; ao descompactar o zip, mantenha todos os arquivos na mesma pasta.

Atenção, é importante destacar que a versão do Access seja 2000 ou posterior, pois há algumas referências que não são suportadas em versões anteriores (usadas no exemplo com subformulário).

## A NECESSIDADE

Em sistemas que desenvolvemos sempre há cadastros que considero como básicos: Cidades, Estados, Produtos, Funcionários, Projetos, Fornecedores, Clientes e muitos outros. Por isso, na maioria dos meus projetos sempre há uma tela de Cadastros com 2 componentes: um "grupo de opções" para cada cadastro necessário e um subformulário não-vinculado. Conforme clicamos no item do "grupo de opções" o subformulário exibe o formulário desenvolvido para o item (tipo datasheet) ou pode-se até mesmo exibir a própria tabela ou consulta dos dados (conceito usado nessa matéria). Essa estrutura fornece uma interface única, consistente e fácil para o usuário nesta parte do sistema. O problema surge quando temos muitos cadastros para gerenciar de tal forma que o espaço físico de um formulário não comporta todos os itens no "grupo de opções". O layout fica comprometido e perde a eficiência.

Com uma ListBar seria possível exibir todos os itens de cadastros que precisamos, agrupando-os sob alguma classificação e permitindo adotar o mesmo estilo de layout citado.

Para começar, é necessária uma tabela que permita registrar os itens da nossa tela de cadastros. Veja os dados da tabela tbl\_ListBar e a sua estrutura:

Campo	Tipo	Descrição
IDGrupo	Número (Inteiro) [chave]	Identifica o Grupo
Item	Texto (20) [chave]	Identifica o Item do Cadastro
Objeto	Texto (20)	Identifica o Objeto que deverá ser aberto quando o Item selecionado

Tabela 1 – Estrutura da tabela tbl\_ListBar

Nos dados da tabela, observe que existem 4 grupos identificados (de 1 a 4) e uma tabela física correspondente para cada "Objeto" citado. Algumas delas vieram do banco de dados Northwind Traders, de outros projetos, e usei também o assistente de criação de tabela do próprio Access para enriquecer ainda mais esta matéria. Para esse exemplo, a estrutura e os dados de cada tabela dessas não têm importância.

## LISTBAR COM CAIXA DE LISTAGEM

Esta interface corresponde ao formulário "frm\_Cadastros\_listbox". Veja a Figura 1 com a estrutura do formulário na primeira fase da construção.



DISPONÍVEL PARA DOWNLOAD

**LISTBAR.ZIP** (Access 2000, XP e 2003)  
Disponível para download em [www.forumaccess.com.br](http://www.forumaccess.com.br)

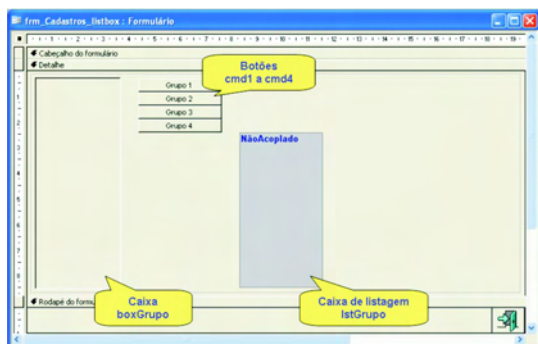


Figura 1 - Estrutura do formulário na primeira fase da construção

Segue um descritivo dos objetos utilizados:

Objeto	Tipo	Descrição
boxGrupo	caixa	Efeito visual para a ListBar
cmd1 até cmd4	botões de comando	Exibe os grupos desejados (a nomenclatura padronizada facilita a manipulação dos botões na tela)
lstGrupo	caixa de listagem	Exibe os itens do grupo selecionado através de pesquisa na tabela tbl_ListBar

Tabela 2 - Objetos utilizados

Após configurar esses objetos, coloque-os sobrepostos para montar o layout da ListBar e insira um subformulário ao lado. Veja a fase 2 na próxima figura.

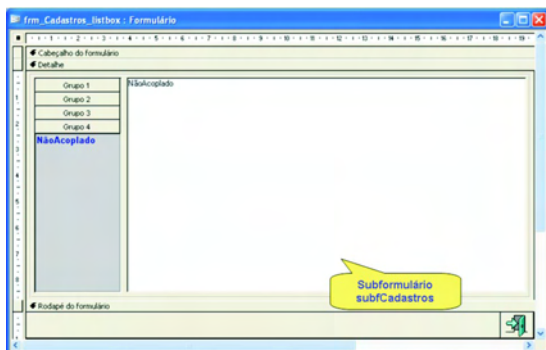


Figura 2 - Fase 2

Para movimentar os botões de Grupos, reposicionar e atualizar a caixa de listagem, há uma função chamada ListBar(Grupo), que pode ser usada para quantos grupos forem necessários. Veja, em cada botão, a propriedade "Ao clicar" (On Click).

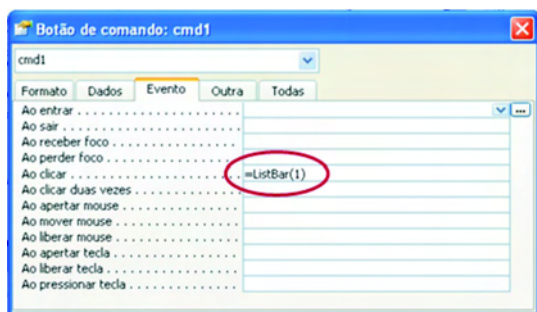


Figura 3 - Propriedade "Ao clicar" (On Click)

Detalhando melhor - a função ListBar é responsável por:

- Receber, através do parâmetro "i", o código de grupo clicado.
- Identificar, numa matriz (matGrupo), os valores de Top de cada botão de grupo que criamos.
- Ocultar a caixa de listagem lstGrupo.
- Alinhar, no topo da caixa boxGrupo, os botões de grupo até o grupo clicado (usando um For...Next com Step -30, teremos um efeito de deslize da posição atual do botão até o limite possível do topo da caixa boxGrupo).
- Alinhar, na base da caixa boxGrupo, os botões de grupo depois do grupo clicado (também usando um For...Next com Step 30 para termos o efeito de deslize do botão até o limite possível da base da caixa boxGrupo na mesma velocidade).
- Montar um SELECT da tabela tbl\_ListBar conforme o grupo clicado.
- Posicionar a caixa de listagem lstGrupo abaixo do botão de grupo clicado e reexibi-la atualizada.

```
Function ListBar(i As Integer)
    Dim matGrupo(1 To 4) As String
    Dim j As Integer, p As Long, X As Long
    Dim strListBox As String
    'matriz de grupos
    matGrupo(1) = Me.cmd1.Top
    matGrupo(2) = Me.cmd2.Top
    matGrupo(3) = Me.cmd3.Top
    matGrupo(4) = Me.cmd4.Top
    'esconde opções
    Me.lstGrupo.Visible = False
    'alinha grupos até o selecionado
    p = Me.boxGrupo.Top
    For j = 1 To i
        For X = Me("cmd" & j).Top To p Step -30
            Me("cmd" & j).Top = X
            Me.Repaint
        Next
        p = p + Me("cmd" & j).Height
    Next
    'alinha grupos após o selecionado
    p = Me.boxGrupo.Height + 190
    For j = UBound(matGrupo) To i + 1 Step -1
        For X = (Me("cmd" & j).Top) To (p - Me("cmd" & j).Height) Step 30
            Me("cmd" & j).Top = X
            Me.Repaint
        Next
        p = p - Me("cmd" & j).Height
    Next
    'monta opções conforme o grupo
    strListBox = "SELECT Item, Objeto " & _
        "FROM tbl_ListBar " & _
        "WHERE IDGrupo = " & i & " " & _
        "ORDER BY Item;"
    'reexibe opções
    With Me.lstGrupo
        .Top = Me("cmd" & j).Top + Me("cmd" & j).Height
        .RowSource = strListBox
        .Visible = True
    End With
End Function
```

Para iniciar a nossa ListBar o formulário tem no evento Ao Abrir (On Open) uma chamada à função ListBar com o parâmetro de grupo igual a 1 e assim, exibe o primeiro grupo automaticamente. As outras rotinas que precisamos são: o cmdFechar\_Click (o botão que fecha o formulário), o Form\_Load (para iniciarmos o formulário com o primeiro cadastro disponível já selecionado) e o IstGrupo\_AfterUpdate (para atualizarmos o SourceObject do subformulário conforme alteramos a seleção do item na caixa de listagem Ist\_Grupo).

```
Private Sub cmdFechar_Click()
    DoCmd.Close
End Sub
Private Sub Form_Load()
    Me.IstGrupo.Selected(0) = True
    IstGrupo_AfterUpdate
End Sub
Private Sub IstGrupo_AfterUpdate()
    With Me.subfCadastrros
        .SourceObject = "Table." & Me.IstGrupo.Column(1)
        .SetFocus
    End With
    DoCmd.RunCommand acCmdSelectRecord
End Sub
```

Veja agora o formulário em execução. Clique sobre cada grupo e veja como os controles são atualizados dinamicamente na tela. É possível perceber que facilmente podem-se criar mais grupos e manipular os itens desses grupos apenas relacionando corretamente a tabela tbl\_ListBar com os botões do formulário. A função ListBar varrerá todos os itens encontrados conforme os critérios de nomenclatura para os objetos definidos.

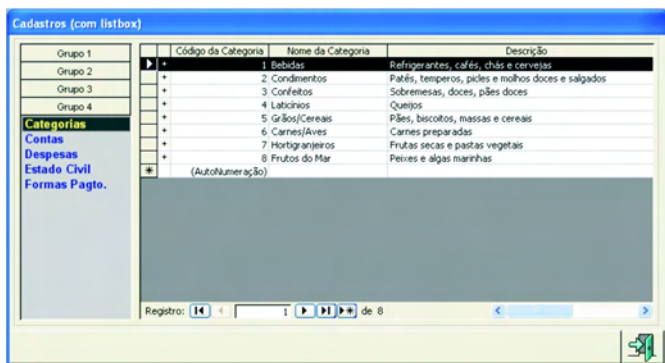


Figura 4 - Formulário em execução

### LISTBAR COM SUBFORMULÁRIO

Esta interface corresponde aos formulários "frm\_Cadastrros\_subform" e "frm\_ItemListBar".

O uso de subformulário permite usar ícones, já que, nele, podemos construir um layout mais parecido com o que o Outlook exibia. Para desenvolver essa interface, a complexidade aumenta um pouco, mas não são necessárias muitas alterações nas rotinas que já desenvolvemos. Por isso, recomendo duplicar o formulário "frm\_Cadastrros\_listbox" e realizar as mudanças que descreverei em seguida.

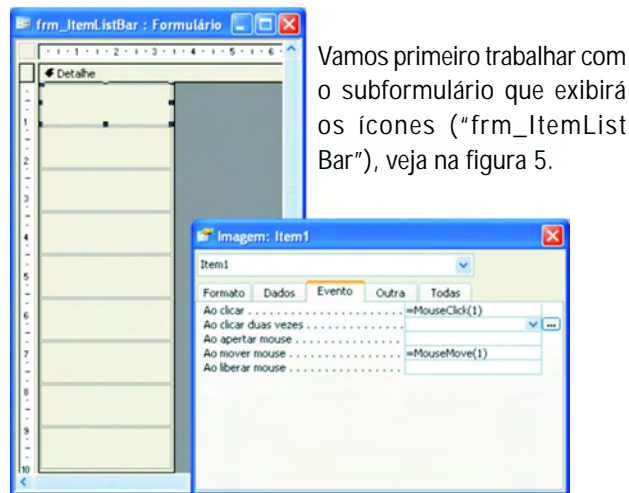


Figura 5 - Subformulário que exibirá os ícones

Defini um máximo de 9 ícones por grupo. Por isso, temos aqui 9 objetos do tipo Imagem formatados e posicionados como vimos na figura anterior. Elas não possuem bordas, não indicam nenhuma imagem inicial e são nomeadas seguindo o formato "Item<n>" (de 1 a 9) para facilitar a atualização dos objetos. Cada objeto Imagem possui chamadas para as rotinas MouseClick(n), no evento Ao Clicar (On Click), para simular o click do mouse como se fosse um botão de comando e realizar a atualização do subformulário de cadastros; e a MouseMove(n), no evento Ao Mover Mouse (On MouseMove), para simular o efeito visual de um botão alterando as bordas da Imagem. Para alterar as bordas das imagens, usa-se a propriedade SpecialEffect, onde cada número corresponde a um tipo de borda suportada pelo objeto. Veja na tabela 3 a explicação extraída do próprio Help do Access.

Configuração	Visual Basic	Descrição
Plano	0	O objeto aparece plano, com as cores padrão do sistema ou cores personalizadas que tenham sido definidas no modo de design.
Alto relevo	1	O objeto tem um brilho acima e à esquerda e uma sombra embaixo e à direita.
Baixo relevo	2	O objeto tem uma sombra acima e à esquerda e é realçado embaixo e à direita.
Esboçado	3	O objeto tem uma linha em baixo relevo ao redor do controle.
Sombreado	4	O objeto tem uma sombra abaixo e à direita do controle.
Cinzelado	5	O objeto tem uma linha em baixo relevo, abaixo do controle.

Tabela 3 – Explicação sobre bordas no Access

```
Option Compare Database
Option Explicit
Dim ctl As Control
Sub MouseClick(i As Integer)
    For Each ctl In Me.Controls
        If ctl.SpecialEffect = 2 Then ctl.SpecialEffect = 0
```

```

If ctl.SpecialEffect = 1 Then ctl.SpecialEffect = 2
Next
Me("Item" & i).SpecialEffect = 2
With Me.Parent!subfCadastrros
    .SourceObject = "Table." & Me("Item" & i).Tag
    .SetFocus
End With
DoCmd.RunCommand acCmdSelectRecord
End Sub
Sub MouseMove(i As Integer)
For Each ctl In Me.Controls
If ctl.SpecialEffect <> 2 Then
If Right(ctl.Name, 1) = i Then
ctl.SpecialEffect = 1
Else
ctl.SpecialEffect = 0
End If
End If
Next
End Sub

```

Como teremos ícones e quantidades de ícones diferentes para cada grupo, há a função `Updateltens` para que as imagens sejam atualizadas e sincronizadas com a `ListBar`. Essa rotina substitui o `RowSource` dinâmico do `IstGrupo`, que não teremos nesse exemplo.

Detalhando melhor a função, ela é responsável por:

- Receber o grupo clicado através do parâmetro `X`.
- Construir um `RecordSet` (com referência ao `DAO 3.6`) da tabela `tbl_ListBar`, conforme o parâmetro `X` informado, respeitando o limite de até 9 ícones no formulário.
- Atribuir à propriedade `Picture` de cada imagem o arquivo `bmp` criado conforme a concatenação dos parâmetros.
- Atribuir à propriedade `Tag` de cada imagem o valor do campo Objeto relacionado (para ser usado posteriormente na atualização do subformulário `subfCadastrros`).
- Ocultar as imagens não utilizadas
- Redimensionar as imagens conforme as configurações definidas e ajustar a altura da seção `Detalhe` para não sobrar espaço no final do formulário (otimizando a barra de rolagem do formulário).
- E ainda, caso o subformulário de `Cadastrros` (`subfCadastrros`) esteja exibindo algum item, localizar o ícone correspondente e alterar suas bordas para indicar que o botão foi pressionado (isso não era necessário no exemplo com a caixa de listagem, pois o item selecionado fica guardado na memória).

```

Public Function Updateltens(X As Integer)
Dim rst As Recordset
Dim i As Integer, j As Integer, qt As Integer
'quantidade de ícones
qt = 9
Set rst = CurrentDb.OpenRecordset("SELECT Item, Objeto " & _
    "FROM tbl_ListBar " & _
    "WHERE IDGrupo = " & X & ";", dbOpenSnapshot)
i = 1
With rst
Do While Not .EOF
Me("Item" & i).Picture = Application.CurrentProject.Path & "img" & _
Mid(!Objeto, 5) & ".bmp"
Me("Item" & i).Tag = !Objeto
Me("Item" & i).Visible = True

```

```

i = i + 1
If i > qt Then Exit Do
.MoveNext
Loop
.Close
End With
j = i
For i = i To qt
Me("Item" & i).Visible = False
Next
Me.Section(0).Height = (qt * Me.Item1.Height) + 300
For i = 2 To qt
If Me("Item" & i).Visible Then
Me("Item" & i).Top = Me("Item" & i - 1).Top + Me("Item" & i - 1).Height + 20
Else
Me("Item" & i).Top = 0
End If
Next
Me.Section(0).Height = ((j - 1) * Me.Item1.Height) + 300
For Each ctl In Me.Controls
If InStr(Me.Parent!subfCadastrros.SourceObject, ctl.Tag) > 0 Then
ctl.SpecialEffect = 2
Else
ctl.SpecialEffect = 0
End If
Next
End Function

```

Agora voltemos ao formulário de `Cadastrros` que, após a duplicação, chamei de `frm_Cadastrros_subform`. Como dito antes, as alterações são simples.

Primeiro, apague a caixa de listagem (e a sua rotina `AfterUpdate`) e, no lugar dela, insira um subformulário do objeto `frm_ItemListBar` (sem vínculos de campos filho/mestres). Pra facilitar, chame-o de `frm_ItemListBar` e formate-o como a figura 6.

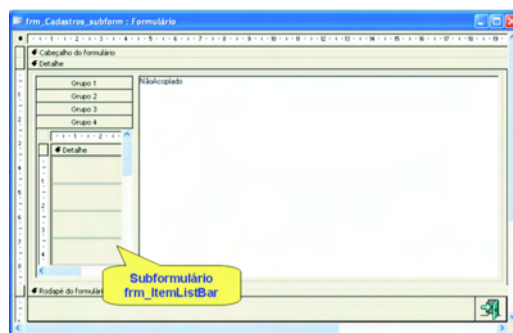


Figura 6 – Formatando o `frm_ItemListBar`

Para melhorar a visualização, aumentei a largura da `boxGrupo` e dos botões de `Grupo`.

No evento `Form_Load`, exclua as linhas existentes e inclua apenas uma, chamando a função `MouseDown` do subformulário para o grupo 1, conforme código:

```

Private Sub Form_Load()
Me.frm_ItemListBar.Form.MouseDown (1)
End Sub

```

Acrescente também uma rotina para o evento `Ao Mover Mouse` da seção `Detalhe` (`Detail_MouseMove`) para realinhar as bordas das imagens quando navegamos pelo formulário:

```

Private Sub Detail_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    Dim ctl As Control
    For Each ctl In Me.frm_ItemListBar.Controls
        If ctl.SpecialEffect <> 2 Then
            ctl.SpecialEffect = 0
        End If
    Next
End Sub

```

Para a função ListBar, faça as seguintes alterações:

- Mude as referências da caixa de listagem lstGrupo para o subformulário frm\_ItemListBar.
- Exclua a definição e a montagem da variável strListBox.
- Mude a linha ".RowSource = strListBox" para ".Form.UpdateTens (i)".

Atenção para o acesso a propriedades e métodos, já que utilizei blocos With...End With em algumas partes do aplicativo. Compile e veja o formulário em execução (figura 7). Clique sobre cada grupo e veja como os controles são atualizados dinamicamente na tela. Todas as observações notadas no exemplo com a caixa de listagem também podem ser aplicadas aqui.

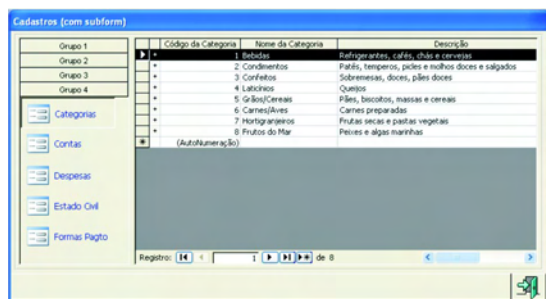


Figura 7 - Formulário em execução

## OPS, ALGO ESTRANHO OCORREU, O QUE EU FIZ DE ERRADO?

Mesmo que você tenha seguido corretamente os passos demonstrados neste exemplo, poderá se deparar com a tela seguinte ao abrir o formulário frm\_Cadastros\_subform e passar o mouse sobre a barra lateral (figura 8).

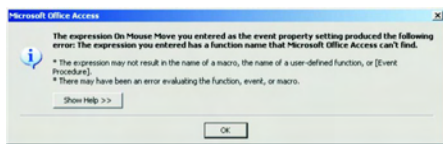


Figura 8 – Mensagem de erro

A explicação para o erro acima está na própria ajuda que obtemos ao clicar no botão Show Help dessa tela. O problema ocorre, pois o nível de segurança do Access está configurado para "Médio" ou "Alto" e o service pack 8 ou superior do Jet não está instalado.

Para começar, feche a caixa de mensagem. Ela poderá aparecer novamente se você mover o mouse sobre a barra lateral ou clicar em um de seus botões.

Para solucionar o problema, você pode escolher entre instalar a atualização do Service Pack mais recente (recomendado) ou mudar o nível de segurança do Access para "Baixo". Note que essa última opção expõe o computador a macros maliciosos, portanto, fique bastante atento para o risco envolvido neste procedimento. O nível de segurança pode ser verificado ou modificado através do menu Ferramentas ► Macro ► Nível de segurança (figura 9).

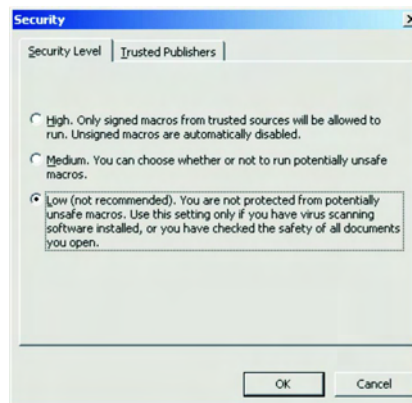


Figura 9 – Modificando Nível de Segurança

Depois de mudar o nível de segurança para "Baixo", o Access mostrará a mensagem de alerta da figura 10.

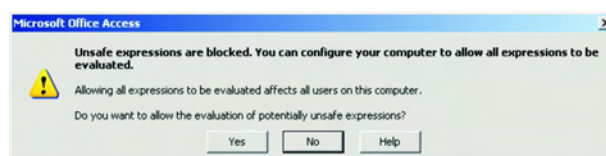


Figura 10 - Mensagem de alerta

Após a confirmação, uma nova mensagem indica que a aplicação precisa ser reiniciada (fechar e abrir novamente o Access). Veja na figura 11.

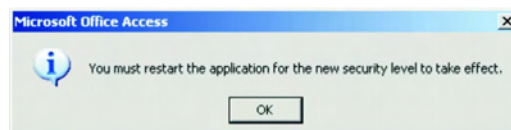


Figura 11 – Mensagem para reiniciar a aplicação

Ao mudar o nível de segurança para Médio ou Alto, o Access poderá mostrar um novo alerta incentivando você a instalar o Microsoft Jet 4.0 Service Pack 8.

## CONCLUSÃO

Usando o subformulário, temos mais liberdade para a construção dos itens da ListBar, pois podemos utilizar ícones e posicioná-los aonde for desejado, porém, usando a caixa de listagem, temos uma manutenção mais simples e performance melhor. De qualquer forma, o desenvolvimento é facilitado quando usamos os objetos nomeados como a estrutura de uma matriz (como cmd1, cmd2, ... e item1, item2, ...). Aplicativos podem ser facilmente dimensionados nesse ambiente com uma boa interface e, o melhor, sem custo para a empresa e o desenvolvedor.

Aqui usei uma série de tipos de cadastros, mas pode-se usar esse recurso para outras situações semelhantes, como relatórios, interfaces e menus de controle.

Espero que esta matéria ilumine o caminho de diversos projetos semelhantes para serem aprimorados com as técnicas aqui apresentadas. Fiquem à vontade para mandar idéias, sugestões e críticas.

Bom estudo a todos! ■

\* **Paulo Sergio Sarraino** (pss@sarraino.com.br) é analista de sistemas e sócio da Sarraino Systems. Possui graduação de Tecnologia em Processamento de Dados pela FASP (1997) e pós-graduação em Análise de Sistemas c/ Ênfase em Gerência de Projetos pela FIAP (2000). Atua como consultor em Análise e Desenvolvimento de Sistemas em ambiente MS Office e Internet (especialista em Access, VBA e ASP) há mais de 10 anos.