

Visual Basic .NET

List Boxes, Checked List Boxes, Combo Boxes e Picture Boxes

Professor: Danilo Giacobbo

Página pessoal: www.danilogiacobbo.eti.br

E-mail: danilogiacobbo@gmail.com

Objetivos da aula

- ✓ Trabalhar com o controle **List Box**
- ✓ Trabalhar com o controle **Checked List Box**
- ✓ Trabalhar com o controle **Combo Box**
- ✓ Trabalhar com o controle **Picture Box**



Introdução

List Boxes

- Apresenta uma lista de itens
- A pessoa seleciona o item desejado
- Podem conter barras de rolagem

Checked List Boxes

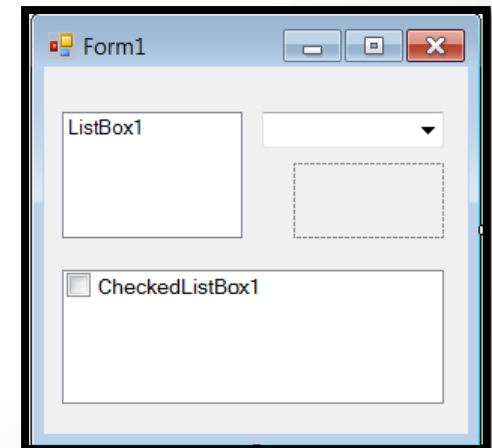
- Derivado do controle List Box
- Possui um check box para cada item

Combo Boxes

- Combinação de List Box com Text Box
- A pessoa pode selecionar o item ou digitar o mesmo

Picture Boxes

- Apresenta uma imagem



List Boxes

- Apresentam uma lista de itens da qual o usuário pode escolher um ou mais.
- Se a lista ficar grande a barra de rolagem aparecerá automaticamente.
- Os elementos mais importantes do controle List Box são:

Propriedades

- MultiColumn
- ScrollAlwaysVisible
- SelectedIndex
- SelectedItem
- SelectionMode
- SelectedItems
- SelectedIndices

Métodos

- SetSelected
- BeginUpdate
- EndUpdate
- FindString
- FindStringExact

Outros

- Coleção Items (Count, Add, Insert, Clear, Remove, AddRange)

Checked List Boxes

- Derivam do controle List Box e suportam um checkbox para cada item.
- Os elementos mais importantes do controle Checked List Box são:

Propriedades

- Items
- CheckedItems
- CheckedIndices

Métodos

- GetItemChecked
- SetItemChecked
- GetItemCheckState
- SetItemCheckState

Eventos

- ItemCheck

Combo Boxes

- Usado para mostrar dados de forma “drop-down”.
- A pessoa pode selecionar o item ou informar o mesmo digitando na caixa.
- Os elementos mais importantes do controle Combo Box são:

Propriedades

- Text
- SelectedIndex
- SelectedItem
- DropDownStyle
- Coleção Items (Count, Add, Insert, Clear, AddRange, Remove)

Métodos

- BeginUpdate e EndUpdate
- FindString e FindStringExact

Picture Boxes

- Usada para mostrar imagens do tipo bitmap, ícones, JPEG, GIF e outros.
- Os elementos mais importantes do controle Picture Box são:

Propriedades

- Image
- SizeMode
- ClientSize
- BorderStyle

Eventos

- Click
- MouseDown

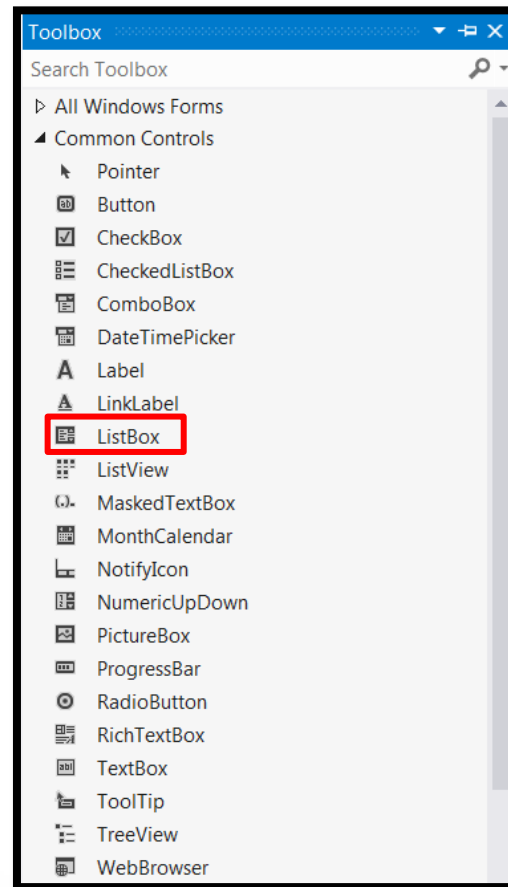
Enumerações

- PictureBoxSizeMode
 - Normal
 - StretchImage
 - AutoSize
 - CenterImage

Usando a classe **ListBox**

□ A hierarquia de classes do controle **ListBox** é a seguinte:

System
 Windows
 Forms
 ListBox

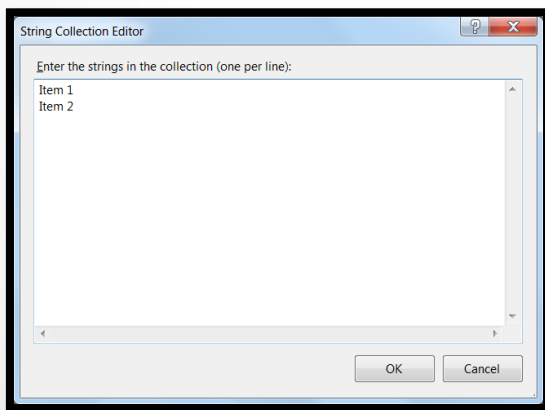


Adicionando itens a um List Box

- Você pode adicionar itens a um List Box tanto em modo de design quanto em tempo de execução.
- O primeiro item da lista possui índice igual a 0.
- Em tempo de execução você pode usar a propriedade **Items** e o método **Add** para adicionar um item na lista (ver exemplo no próximo slide).
- Use a propriedade **Items.Count** para obter o número de itens de uma lista.
- Você pode acessar um item da lista usando a propriedade **Items**.

Exemplo: `strText = ListBox1.Items(5)`.

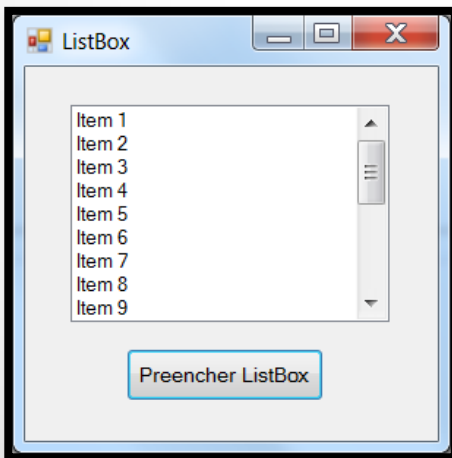
- Em modo de design basta clicar na propriedade **Items**. Uma janela será aberta e você poderá inserir os valores.



Adicionando itens a um List Box

Exemplo 1 (Usando o método Add):

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         ListBox1.BeginUpdate()
4         Dim intContador As Integer
5         For intContador = 1 To 20
6             ListBox1.Items.Add("Item " & intContador.ToString())
7         Next intContador
8         ListBox1.EndUpdate()
9     End Sub
10 End Class
```



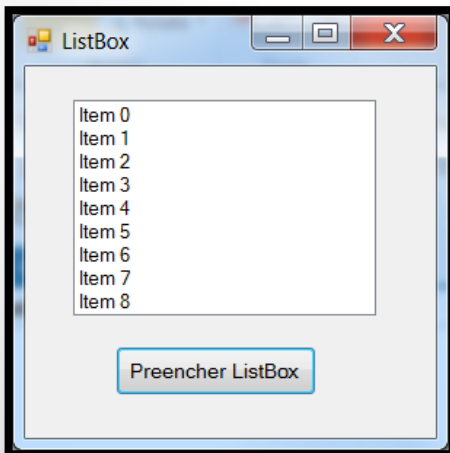
Teste isso:

Tente criar o mesmo exemplo usando o método **Insert**.

Adicionando itens a um List Box

Exemplo 2 (Usando o método AddRange):

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Dim arrDados(8) As String
4         Dim intContador As Integer
5
6         For intContador = 0 To UBound(arrDados)
7             arrDados(intContador) = "Item " & intContador
8         Next intContador
9
10        ListBox1.Items.AddRange(arrDados)
11    End Sub
12 End Class
```



Obtendo um item da lista pelo índice

- Cada item da lista possui um índice (o primeiro é 0).
- Quando a pessoa seleciona um item da lista o evento **SelectedIndexChanged** é chamado.
- Você pode obter o índice do item por meio da propriedade **SelectedIndex**.
- Tente usar também o método **IndexOf** para recuperar o índice de um objeto da lista e a propriedade **Text**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Dim intContador As Integer
4
5         For intContador = 0 To 10
6             ListBox1.Items.Insert(intContador, "Item " & intContador + 1)
7         Next intContador
8     End Sub
9
10    Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) _
11        Handles ListBox1.SelectedIndexChanged
12        TextBox1.Text = "Você selecionou o item " & ListBox1.SelectedIndex + 1
13        TextBox2.Text = "Você selecionou o item de índice " & ListBox1.Items.IndexOf(ListBox1.SelectedItem)
14    End Sub
15 End Class
```

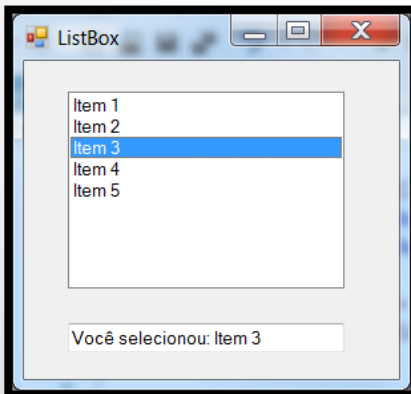
Eventos do List Box

SelectedIndexChanged

- É o evento padrão.
- Ocorre quando a pessoa muda a seleção de um item na lista.

Exemplo:

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles _  
    ListBox1.SelectedIndexChanged  
    TextBox1.Text = "Você selecionou: " & ListBox1.SelectedItem.ToString  
End Sub
```



Eventos do List Box

Click e DoubleClick

- O evento **Click** é usado para, por exemplo, mostrar algo a mais de acordo com o item selecionado (uma fonte, uma cor, etc...)
- O evento **DoubleClick** pode ser usado para ativar um programa exibido em uma lista de programas disponíveis no computador. Tome cuidado porque o evento DoubleClick sempre ativa o evento Click também.

Exemplo:

```
1 Public Class Form1
2     Private Sub ListBox1_Click(sender As Object, e As EventArgs) Handles ListBox1.Click
3         TextBox1.Text = "evento Click: " & ListBox1.SelectedIndex
4     End Sub
5
6     Private Sub ListBox1_DoubleClick(sender As Object, e As EventArgs) Handles ListBox1.DoubleClick
7         TextBox2.Text = "evento DoubleClick: " & ListBox1.SelectedIndex
8     End Sub
9 End Class
```

Removendo itens de um List Box

- Você pode usar o método **RemoveAt** para remover itens de um ListBox.
 - Exemplo: `ListBox1.Items.RemoveAt(5)`
- Para remover o item selecionado.
 - Exemplo: `ListBox1.Items.RemoveAt(ListBox1.SelectedIndex)`
- Para remover um determinado objeto use o método **Remove**.
 - Exemplo: `ListBox1.Items.Remove(ListBox1.SelectedItem)`
- Você pode remover o objeto usando o nome dele também.
 - Exemplo: `ListBox1.Items.Remove("Item 1")`

Importante:

- Removendo um item da lista altera os índices de todos os demais itens.

Dica:

- Para **remover todos os itens de uma lista** use o método **Items.Clear**.

Desafio:

- Tente remover um item selecionado usando a tecla DEL.

Removendo itens de um List Box

Exemplo:

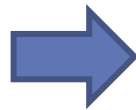
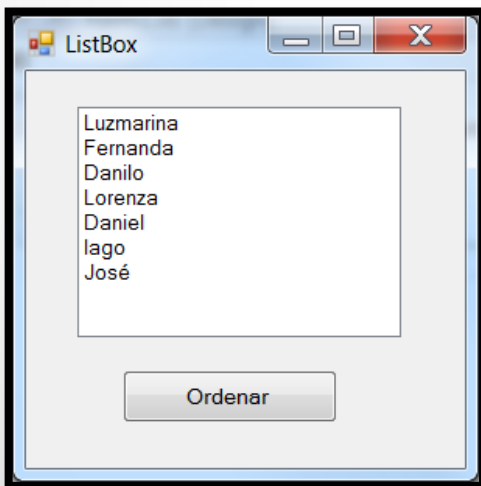
```
1 Public Class Form1
2     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
3         If ListBox1.SelectedIndex >= 0 Then
4             'ListBox1.Items.RemoveAt(ListBox1.SelectedIndex)
5             ListBox1.Items.Remove(ListBox1.SelectedItem)
6         Else
7             MsgBox("Selecione um item da lista para excluir!", MsgBoxStyle.Information)
8         End If
9     End Sub
10
11 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
12     If Trim(TextBox1.Text) <> "" Then
13         ListBox1.Items.RemoveAt(CInt(TextBox1.Text))
14     End If
15 End Sub
16
17 Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
18     If Trim(TextBox2.Text) <> "" Then
19         ListBox1.Items.Remove(TextBox2.Text)
20     End If
21 End Sub
22 End Class
```


Ordenando um List Box

Você pode ordenar os itens de uma lista de modo alfabético usando a propriedade **Sorted**. Alterando o valor dela para **True** a sua lista ficará ordenada. Tome cuidado porque os índices serão reorganizados.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         ListBox1.Sorted = True
4     End Sub
5 End Class
```



Determinando quantos itens há no ListBox

Para saber a quantidade de itens de uma lista você pode usar a propriedade **Items.Count**.

Ela é útil para o caso de precisar percorrer todos os elementos de uma lista por meio de um laço de repetição.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         MsgBox("A ListBox possui " & ListBox1.Items.Count & " itens.", _
4             MsgBoxStyle.Information, "Quantidade de Itens")
5     End Sub
6 End Class
```

ListBox com múltiplos itens selecionados

- Se a propriedade **SelectionMode** estiver configurada para **MultiSimple** ou **MultiExtended** a pessoa consegue selecionar vários itens de uma lista em vez de apenas um por vez.
- Para este tipo de situação você deve usar as propriedades **SelectedItems** e **SelectedIndices** para obter a lista de itens que foram selecionados.
- Para deixar alguns itens configurados você pode usar o método **SetSelected**.

Exemplo:

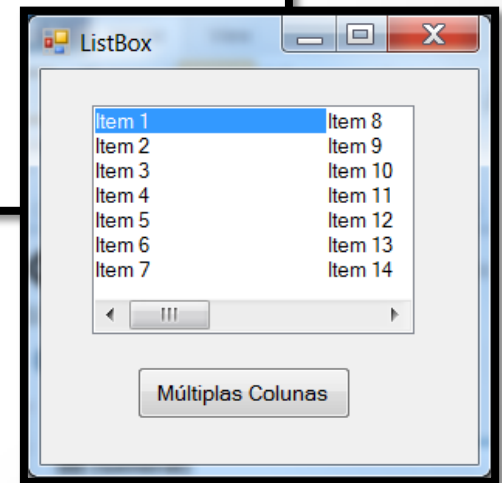
```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         TextBox1.Text = ""
4
5         For Each Item In ListBox1.SelectedItems
6             TextBox1.Text &= " " & Item.ToString
7         Next
8
9         TextBox2.Text = ""
10        For Each Index In ListBox1.SelectedIndices
11            TextBox2.Text &= " " & ListBox1.Items(CInt(Index)).ToString
12        Next
13    End Sub
14 End Class
```

List Boxes com múltiplas colunas

Para deixar a sua lista com múltiplas colunas, isto é, com rolagem horizontal, você deve usar alterar a propriedade **MultiColumn** para **True**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         ListBox1.MultiColumn = True
4
5         ListBox1.BeginUpdate()
6         Dim intIndice As Integer
7         For intIndice = 1 To 20
8             ListBox1.Items.Add("Item " & intIndice.ToString())
9         Next intIndice
10        ListBox1.EndUpdate()
11    End Sub
12 End Class
```

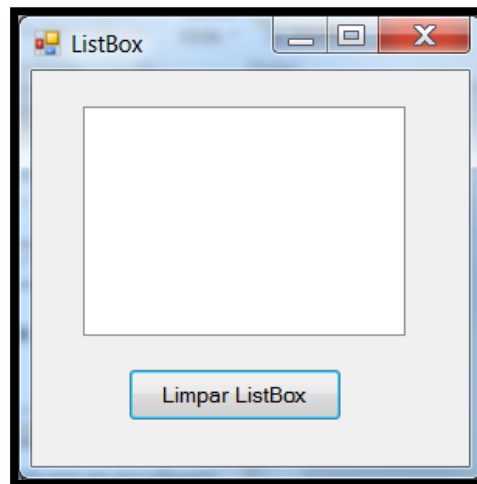
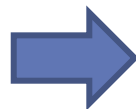
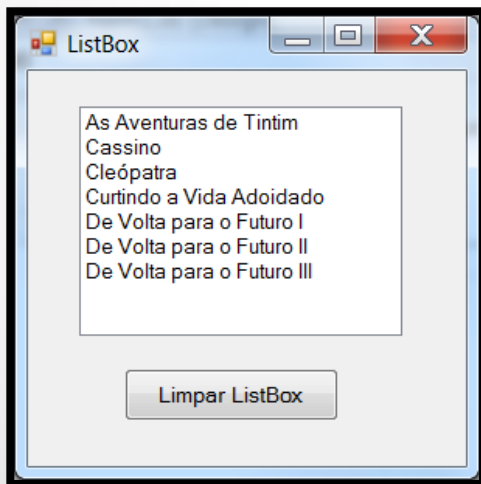


Limpendo um List Box

Para apagar todos os itens de um List Box você pode usar o método **Items.Clear**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         ListBox1.Items.Clear()
4     End Sub
5 End Class
```



Usando a classe `CheckedListBox`

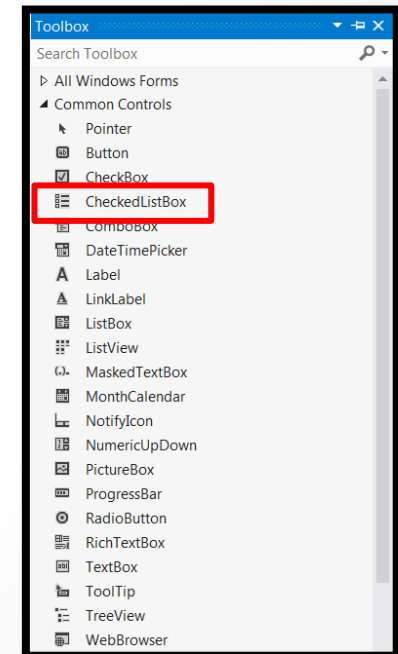
- ❑ Controle que suporta um **check box** para cada item da lista.
- ❑ Uma propriedade que é digna de nota é a **CheckOnClick**. Quando ela é configurada para **True** você pode marcar ou desmarcar um item com apenas um clique do mouse; caso contrário é necessário clicar duas vezes no mesmo.
- ❑ A hierarquia de classes do controle **CheckedListBox** é a seguinte:

System

Windows

Forms

CheckedListBox

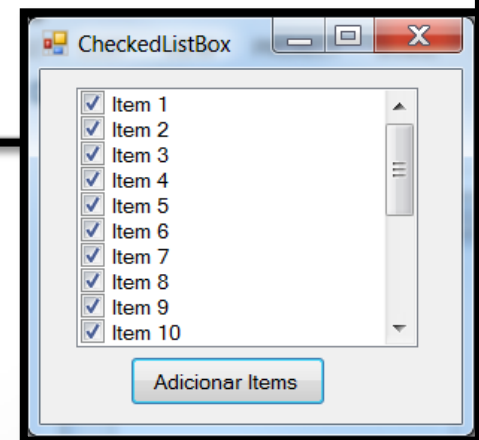


Adicionando itens a um CheckedListBox

- Para adicionar um item na lista você usa o método **Items.Add**.
- Ao adicionar um item você pode configurá-lo para marcado/desmarcado.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
3         CheckedListBox2.CheckOnClick = True
4         CheckedListBox2.BeginUpdate()
5         Dim intIndice As Integer
6         For intIndice = 1 To 20
7             CheckedListBox2.Items.Add("Item " & intIndice.ToString(), True)
8         Next intIndice
9         CheckedListBox2.EndUpdate()
10    End Sub
11 End Class
```

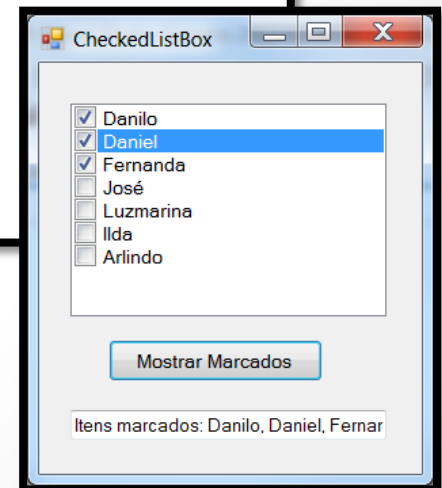


Checked List Box - Itens marcados

- O método **GetItemChecked** retorna **True** se um item foi marcado.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Dim intIndice As Integer
4         Dim strTexto As String
5         strTexto = "Itens marcados: "
6
7         For intIndice = 0 To (CheckedListBox1.Items.Count - 1)
8             If CheckedListBox1.GetItemChecked(intIndice) = True Then
9                 strTexto &= CheckedListBox1.Items(intIndice).ToString & ", "
10            End If
11        Next intIndice
12
13        TextBox1.Text = strTexto
14    End Sub
15 End Class
```

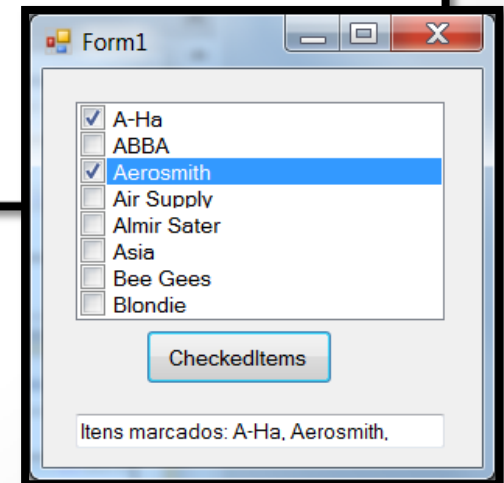


Checked List Box - Itens marcados

- Você pode usar também a propriedade **CheckedItems** o que torna o processo muito mais simples.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Dim strTexto, strData As String
4         strTexto = "Itens marcados: "
5
6         For Each strData In CheckedListBox1.CheckedItems
7             strTexto &= strData & ", "
8         Next
9
10        TextBox1.Text = strTexto
11    End Sub
12 End Class
```

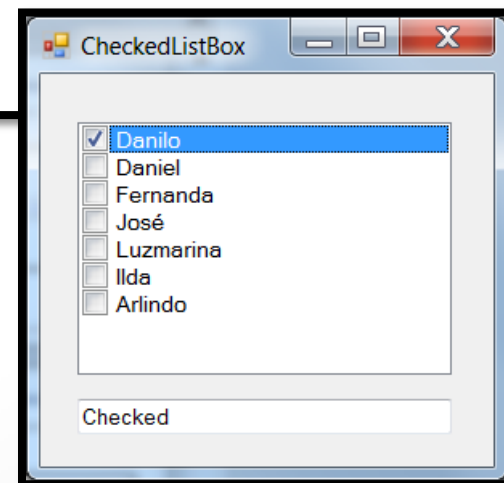


Checked List Box - Itens marcados

- Você pode usar também a propriedade **CheckedIndices**. Ela retorna uma coleção contendo os índices dos itens que estão marcados.
- A propriedade **GetItemCheckState** retorna o estado do item: *marcado*, *desmarcado* ou *indeterminado*.

Exemplo:

```
1 Public Class Form1
2     Private Sub CheckedListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles _
3         CheckedListBox1.SelectedIndexChanged
4         Dim estado As CheckState
5         estado = CheckedListBox1.GetItemCheckState(CheckedListBox1.SelectedIndex)
6         TextBox1.Text = estado.ToString()
7     End Sub
8 End Class
```

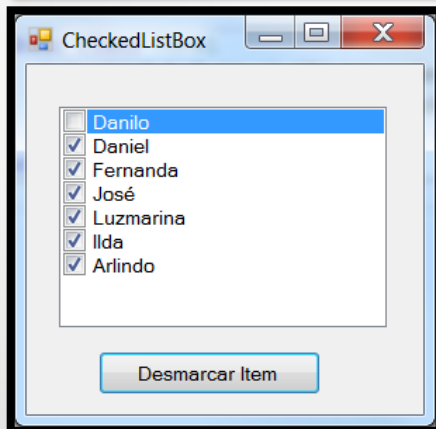


Marcando/Desmarcando um Item

- Você pode usar o método **SetItemChecked** para marcar ou desmarcar itens de uma lista passando o valor **True** ou **False** respectivamente.
- Para trabalhar com checkboxes com 3 estados é necessário usar o método **SetItemCheckState**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         CheckedListBox1.SetItemChecked(0, False)
4     End Sub
5 End Class
```

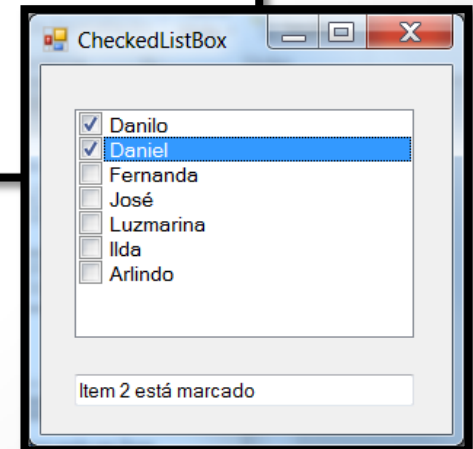


Checked List Box - Eventos

- O principal evento dessa classe é chamado de **ItemChecked**.
- Ele ocorre quando a pessoa marca/desmarca um item.
- Os membros **Index** e **NewValue** contém os valores necessários para que você consiga trabalhar com o item selecionado.

Exemplo:

```
1 Public Class Form1
2     Private Sub CheckedListBox1_ItemCheck(sender As Object, e As ItemCheckEventArgs) Handles _
3         CheckedListBox1.ItemCheck
4         Select Case e.NewValue
5             Case CheckState.Checked
6                 TextBox1.Text = "Item " & e.Index + 1 & " está marcado"
7             Case CheckState.Unchecked
8                 TextBox1.Text = "Item " & e.Index + 1 & " não está marcado"
9         End Select
10    End Sub
11 End Class
```



Usando a classe ComboBox

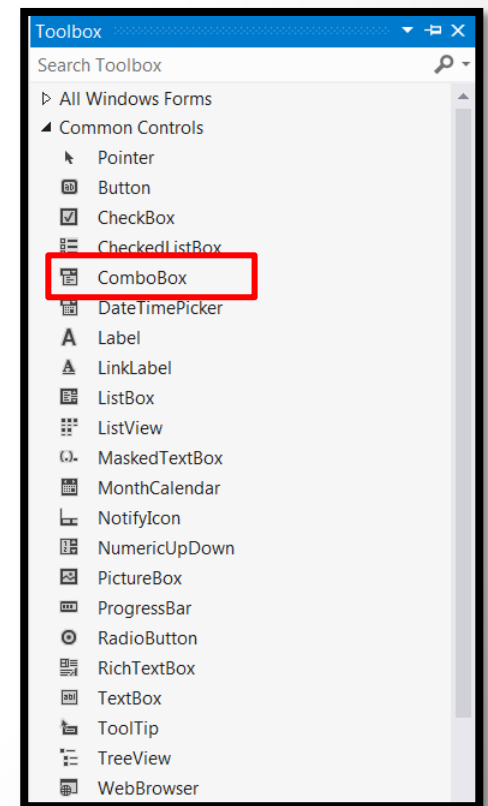
- ❑ Controle que combina um text box com um list box.
- ❑ A hierarquia de classes do controle **ComboBox** é a seguinte:

System

Windows

Forms

ComboBox



Simple, Drop-down e Drop-down List

- Há **três tipos** de Combo Boxes.
- A propriedade **DropDownStyle** define o estilo do mesmo.
- As configurações do mesmo são as seguintes:

DropDown

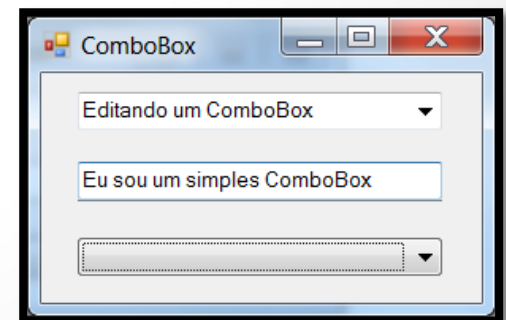
- Configuração padrão.
- Inclui uma lista do tipo drop-down e um text box.
- A pessoa pode selecionar um item da lista ou digitar algo no text box.

Simple

- Inclui uma lista e um text box.
- A pessoa pode selecionar um item da lista ou digitar algo no text box.
- Ajuste a propriedade **Height** para mostrar mais itens.

DropDownList

- Inclui a seleção apenas pela lista drop-down.
- A pessoa pode selecionar apenas um item da lista.

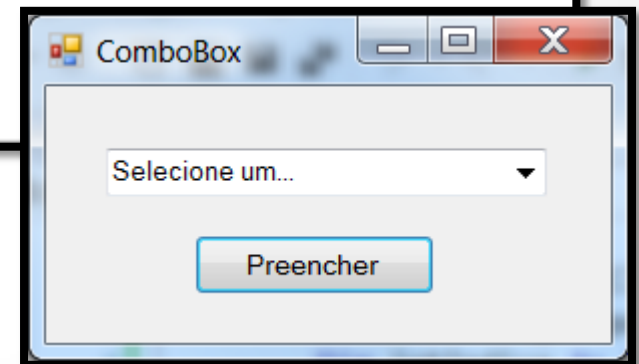


Adicionando itens a um Combo Box

- Para adicionar um texto use a propriedade **Text**.
- Para adicionar itens na lista use a propriedade **Items**.
- Você pode usar os métodos **Insert**, **Add** e **AddRange** para adicionar itens.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         ComboBox1.BeginUpdate()
4         Dim intIndice As Integer
5         For intIndice = 0 To 20
6             ComboBox1.Items.Add("Item " + intIndice.ToString())
7         Next
8         ComboBox1.Text = "Selecione um..."
9         ComboBox1.EndUpdate()
10    End Sub
11 End Class
```



Combo Box - Eventos

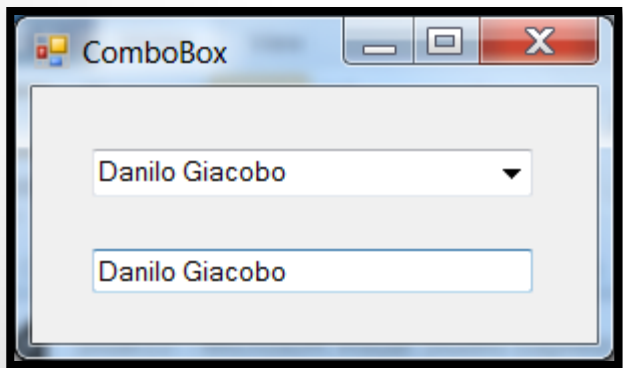
- O evento da parte textual é chamado de **TextChanged**.
- Você não pode fazer seleção múltipla em um combo box.
- Para a seleção de itens da lista você pode usar os eventos:
 - **SelectedIndexChanged**
 - **Click**
 - **DoubleClick**

Combo Box - Evento TextChanged

- Quando a pessoa muda o texto de um combo box o evento **TextChanged** ocorre.
- Você pode usar a propriedade **Text** para recuperar o texto digitado.

Exemplo:

```
1 Public Class Form1
2     Private Sub ComboBox1_TextChanged(sender As Object, e As EventArgs) Handles _
3         ComboBox1.TextChanged
4         TextBox1.Text = ComboBox1.Text
5     End Sub
6 End Class
```

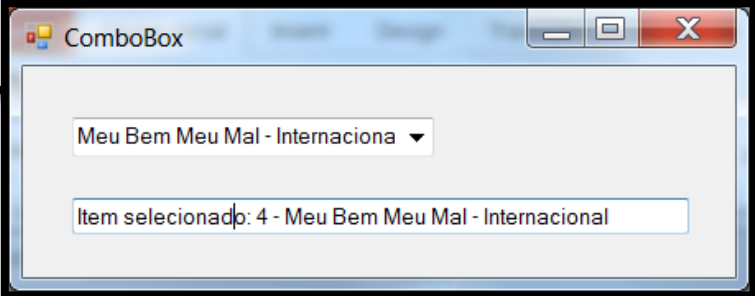


Combo Box - Evento SelectedIndexChanged

- Quando a pessoa muda a seleção de um item de um combo box o evento **SelectedIndexChanged** ocorre.
- Você pode usar as propriedades **SelectedIndex** e **SelectedItem** para recuperar o item selecionado.

Exemplo:

```
1 Public Class Form1
2     Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles _
3         ComboBox1.SelectedIndexChanged
4         Dim intIndiceSelecioneado As Integer
5         intIndiceSelecioneado = ComboBox1.SelectedIndex
6         Dim objItemSelecioneado As Object
7         objItemSelecioneado = ComboBox1.SelectedItem
8         TextBox1.Text = "Item selecionado: " & intIndiceSelecioneado.ToString & _
9             " - " & objItemSelecioneado.ToString
10    End Sub
11 End Class
```

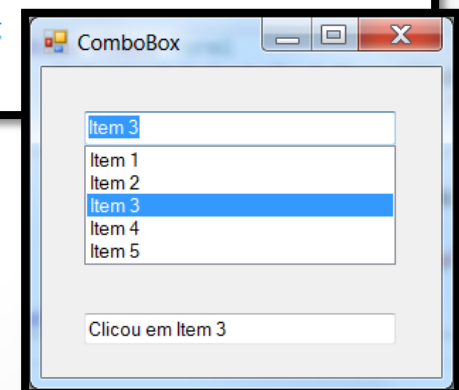


Combo Box - Evento Click e DoubleClick

- Quando a pessoa faz uma seleção na lista com o mouse o evento **Click** é disparado.
- Você pode usar as propriedades **SelectedIndex** e **SelectedItem** para recuperar o item que foi clicado.
- O evento **DoubleClick** funciona apenas em combo boxes do tipo **Simple**.

Exemplo:

```
1 Public Class Form1
2     Private Sub ComboBox1_MouseClick(sender As Object, e As MouseEventArgs) Handles ComboBox1.MouseClick
3         TextBox1.Text = "Clicou em " & ComboBox1.SelectedItem.ToString
4     End Sub
5
6     Private Sub ComboBox1_MouseDoubleClick(sender As Object, e As MouseEventArgs) Handles _
7         ComboBox1.MouseDoubleClick
8         TextBox1.Text = "Clique duplo em " & ComboBox1.SelectedItem.ToString
9     End Sub
10 End Class
```

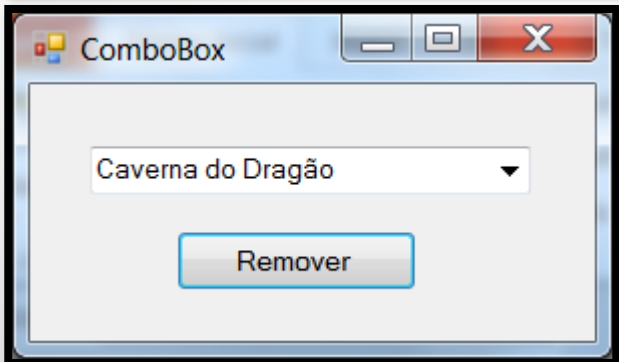


Combo Box - Removendo um item

- Para remover um item você usa os métodos **Remove** e **RemoveAt**.
- É importante salientar que remover um item da lista altera os índices dos demais elementos da lista.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Try
4             ComboBox1.Items.RemoveAt(1)
5         Catch ex As Exception
6             MsgBox("0 índice de número 1 da lista não existe.", MsgBoxStyle.Exclamation, "Erro")
7         End Try
8     End Sub
9 End Class
```

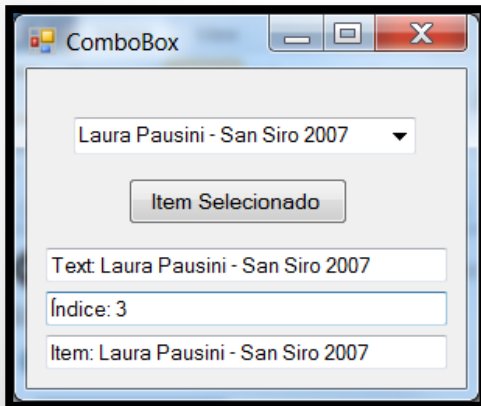


Combo Box - Obtendo a seleção atual

- Você pode usar as propriedades **Text**, **SelectedIndex** e **SelectedItem** para recuperar o item que foi selecionado.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         TextBox1.Text = "Text: " & ComboBox1.Text
4         TextBox2.Text = "Índice: " & ComboBox1.SelectedIndex.ToString
5         TextBox3.Text = "Item: " & ComboBox1.SelectedItem.ToString
6     End Sub
7 End Class
```

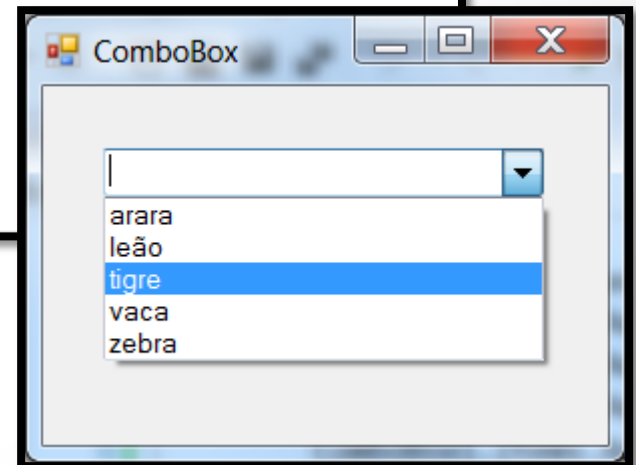


Ordenando um Combo Box

- Para ordenar alfabeticamente os itens de um combo box é necessário configurar para **True** a propriedade **Sorted**.
- É importante salientar que ordenar os itens pode alterar os índices dos demais elementos da lista.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         ComboBox1.Items.Add("zebra")
4         ComboBox1.Items.Add("tigre")
5         ComboBox1.Items.Add("leão")
6         ComboBox1.Items.Add("arara")
7         ComboBox1.Items.Add("vaca")
8         ComboBox1.Sorted = True
9     End Sub
10 End Class
```

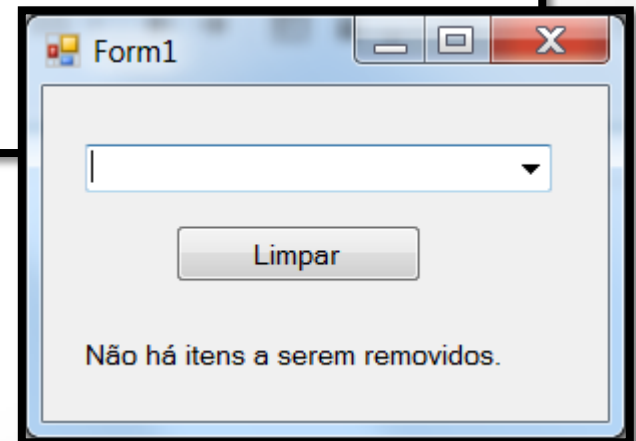


Limpendo um Combo Box

- Limpar um combo box significa apagar todos os itens do mesmo.
- Você pode usar o método **Items.Clear** para realizar esta tarefa.
- **Cuidado:** Não há o método **Unclear**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         If ComboBox1.Items.Count < 1 Then
4             Label1.Text = "Não há itens a serem removidos."
5         Else
6             Label1.Text = "Todos os itens foram removidos."
7             ComboBox1.Items.Clear()
8         End If
9     End Sub
10 End Class
```

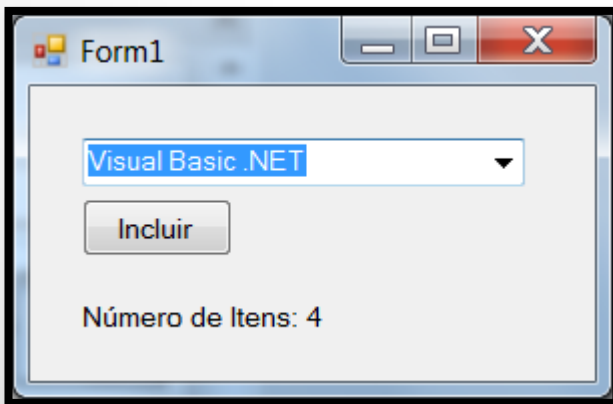


Obtendo o número de itens do Combo Box

- Você pode obter o número de itens de um combo box usando a propriedade **Items.Count**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         If Trim(ComboBox1.Text) <> "" Then
4             ComboBox1.Items.Add(ComboBox1.Text)
5             Label1.Text = "Número de Itens: " & ComboBox1.Items.Count
6         End If
7     End Sub
8 End Class
```



Armazenando objetos em um List/Combo Box

Atenção!

Este exemplo é longo e deve ser visto no próprio Visual Studio .NET.



Usando a classe PictureBox

- ❑ Controle usado para mostrar imagens.
- ❑ A hierarquia de classes do controle **PictureBox** é a seguinte:

System

Windows

Forms

PictureBox

As **propriedades** e os **eventos** dignos de nota de um objeto **Picture Box** são os seguintes:

- **BorderStyle**
- **Image**
- **Resize**
- **SizeModeChanged**

Configurando uma imagem

- ❑ Para adicionar uma imagem a um controle **PictureBox** você deve alterar a propriedade **Image**.
- ❑ Ela aceita um objeto do tipo **Image** que você pode criar usando o método **FromFile**.
- ❑ Os tipos suportados são: .bmp, .ico, .wmf, .jpg, .gif e outros tipos.
- ❑ Em modo de design basta procurar a propriedade **Image** e clicar nos três pontinhos.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         PictureBox1.Image = Image.FromFile("C:\Users\Danilo Giacobbo\Pictures\coke.jpg")
4     End Sub
5 End Class
```

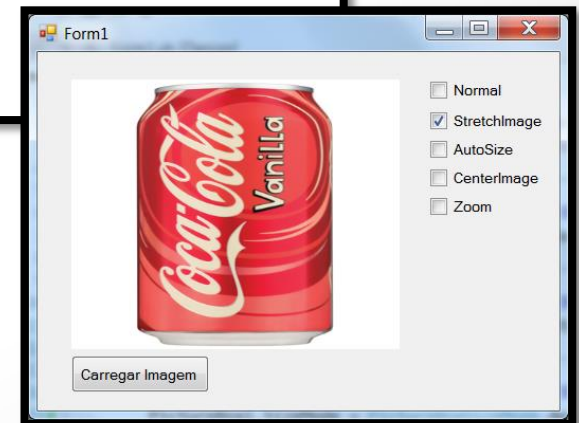


Ajustando o tamanho da imagem

- ❑ Para que uma imagem se ajuste corretamente ao espaço disponível do **PictureBox** é necessário usar a propriedade **SizeMode**.
- ❑ Os possíveis valores dessa enumeração são:
 - ❖ *Normal*
 - ❖ *StretchImage*
 - ❖ *AutoSize*
 - ❖ *CenterImage*
 - ❖ *Zoom*

Exemplo:

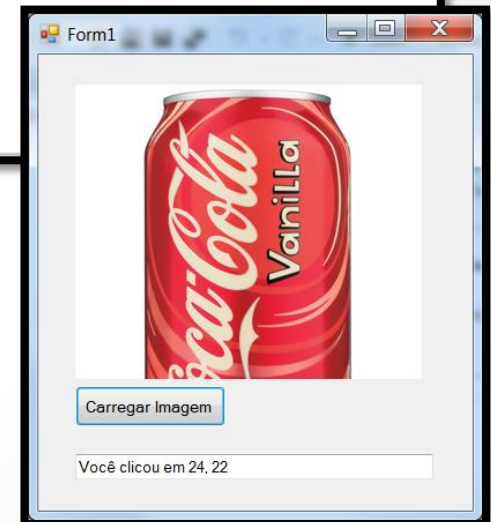
```
Private Sub CheckBox2_CheckedChanged(sender As Object, e As EventArgs) Handles _  
    CheckBox2.CheckedChanged  
    PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage  
End Sub
```



Picture Box - Eventos

- ❑ A maioria dos eventos dessa classe são baseados no mouse.
- ❑ O exemplo abaixo mostra as coordenadas de onde a pessoa clicou na imagem.

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         PictureBox1.Image = Image.FromFile("C:\Users\Danilo Giacobbo\Pictures\coke.jpg")
4     End Sub
5
6     Private Sub PictureBox1_MouseDown(sender As Object, e As MouseEventArgs) Handles _
7         PictureBox1.MouseDown
8         TextBox1.Text = "Você clicou em " & e.X & ", " & e.Y
9     End Sub
10 End Class
```



Referências Bibliográficas

- HOLZNER, Steven. **Visual basic.NET: black book**. Arizona: Coriolis Group Books, 2002. xxxviii, 1144 p ISBN 1-57610-835-X.