

# Visual Basic .NET

Text Boxes, Rich Text Boxes, Labels and Link Labels

Professor: Danilo Giacobbo

Página pessoal: [www.danilogiacobo.eti.br](http://www.danilogiacobo.eti.br)

E-mail: [danilogiacobo@gmail.com](mailto:danilogiacobo@gmail.com)

# Objetivos da aula

- ✓ Trabalhar com o controle **Text Box**
- ✓ Trabalhar com o controle **Rich Text Box**
- ✓ Trabalhar com o controle **Label**
- ✓ Trabalhar com o controle **Link Label**



# Introdução

Object

MarshalByRefObject

Component

Control

TextBoxBase

**RichTextBox**

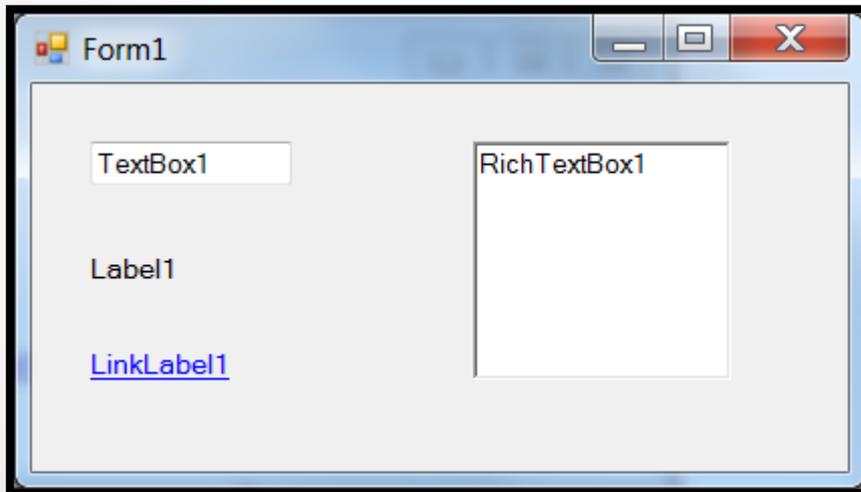


Figura 1 - Um componente text box, um rich text box, um label e um link label.

# Text Boxes

- São controles com design de caixa para digitar texto.
- Eles podem ter múltiplas linhas, barras de rolagem, modo apenas de leitura e outros atributos interessantes.
- A classe **TextBox** é derivada da classe **TextBoxBase** que é baseada na classe **Control**.

Object

MarshalByRefObject

Component

Control

TextBoxBase

**TextBox**

# Text Boxes

As propriedades mais utilizadas de um **TextBox** são:

- ✓ Text
- ✓ MultiLine
- ✓ MaxLength
- ✓ PasswordChar
- ✓ ReadOnly

Exemplo:

```
1 Public Class Form2
2     Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.Text = "Olá Turma de Desenvolvimento Visual!"
4     End Sub
5 End Class
```

**Dica:** Você pode validar os caracteres digitados em um campo do tipo **TextBox** usando o evento **KeyDown**.

# Rich Text Boxes

- São controles com design de caixa para digitar, mostrar e manipular texto.
- Eles fazem tudo que um **Text Box** faz com a adição de poder mostrar fontes, cores e links.
- Você pode usar os efeitos de itálico, negrito e sublinhado e também alterar o tipo e o tamanho da fonte. Inclusive escrever e ler em disco documentos com a extensão **.rtf**.
- A classe **TextBox** é derivada da classe **TextBoxBase** que é baseada na classe **Control**.

Object

MarshalByRefObject

Component

Control

TextBoxBase

**RichTextBox**

**Curiosidade:** Este é o componente usado para criar um editor de texto como o Word.

# Rich Text Boxes

As propriedades mais utilizadas de um **RichTextBox** são:

- ✓ Text
- ✓ LoadFile e SaveFile
- ✓ DetectUrls

**Dica:** Tente usar o evento **LinkClicked** e os métodos **Undo**, **Redo** e **CanRedo**.

Exemplo:

```
1 Public Class Form3
2     Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         RichTextBox1.Text = "Olá Turma de Desenvolvimento Visual"
4         RichTextBox1.SelectAll()
5         RichTextBox1.SelectionColor = Color.White
6         RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
7         RichTextBox1.SelectionBackColor = Color.Black
8         RichTextBox1.DeselectAll()
9     End Sub
10 End Class
```

# Labels

- São controles usados apenas para mostrar textos (etiquetas) na aplicação.
- Ideal para mostrar informações para o usuário.
- Não podem ser editados pela pessoa.
- Não podem receber o foco.
- A classe **Label** é baseada na classe **Control**.

Object

MarshalByRefObject

Component

Control

**Label**

# Labels

As propriedades mais utilizadas de um **Label** são:

- ✓ Text
- ✓ TextAlign
- ✓ UseMnemonic
- ✓ Image e ImageList

Exemplo:

```
1 Public Class Form4
2     Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         Label1.Text = "&Olá! Eu sou um Label."
4         Label1.TextAlign = ContentAlignment.BottomCenter
5         Label1.UseMnemonic = True
6     End Sub
7 End Class
```

# Link Labels

- São controles usados para fazer links para outros objetos e páginas Web.
- Funcionam de forma semelhante a de um label normal.
- A classe **LinkLabel** é baseada na classe **Label**.

Object

MarshalByRefObject

Component

Control

Label

**LinkLabel**

# Link Labels

As propriedades mais utilizadas de um **LinkLabel** são:

- ✓ Text
- ✓ LinkColor
- ✓ VisitedLinkColor
- ✓ ActiveLinkColor
- ✓ LinkArea

**Dica:** O evento mais importante desta classe se chama **LinkClicked**.

Exemplo:

```
1 Public Class Form5
2     Private Sub Form5_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         LinkLabel1.Text = "www.danilogiacobo.eti.br"
4         LinkLabel1.LinkColor = Color.Chocolate
5         LinkLabel1.VisitedLinkColor = Color.DarkOliveGreen
6         LinkLabel1.ActiveLinkColor = Color.BlanchedAlmond
7     End Sub
8 End Class
```

# Configurando o ponteiro do mouse

Para cada controle em sua aplicação (que derive da classe **Control**) você pode alterar o **ponteiro do mouse**. Para isso use a propriedade **Cursor** do controle desejado e escolha o modelo de cursor que você preferir.

Exemplo:

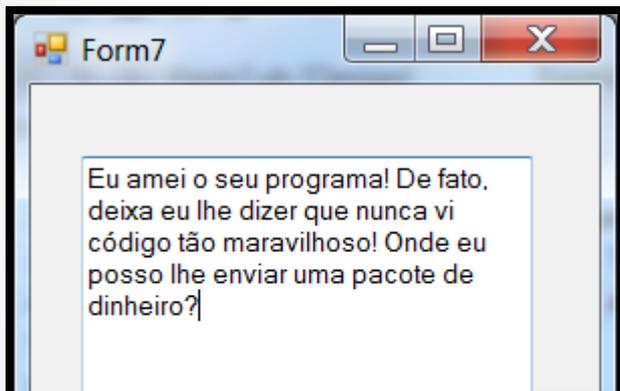
```
1 Public Class Form6
2     Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         Me.Cursor = Cursors.WaitCursor
4         LinkLabel1.Cursor = Cursors.Hand
5         TextBox1.Cursor = Cursors.Help
6     End Sub
7 End Class
```

**Dica:** Em tarefas que demoram para serem processadas, um cursor em forma de ampulheta pode ser útil para a pessoa saber que ela deve esperar o processo ser terminado antes dela continuar mexendo no aplicativo.

# Criando um Text Box de Múltiplas Linhas

Para textos grandes que precisam ser visualizados de forma fácil e simples pelo usuário em um único **TextBox**, é necessário alterar o valor da propriedade **MultiLine** para **True** com quebra automática de texto (**Word Wrap**).

Veja um exemplo:



```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.WordWrap = True
4         TextBox1.Text = "Eu amei o seu programa! De fato, deixa eu lhe dizer " &
5             "que nunca vi código tão maravilhoso! Onde eu " &
6             "posso lhe enviar uma pacote de dinheiro?"
7     End Sub
8 End Class
```

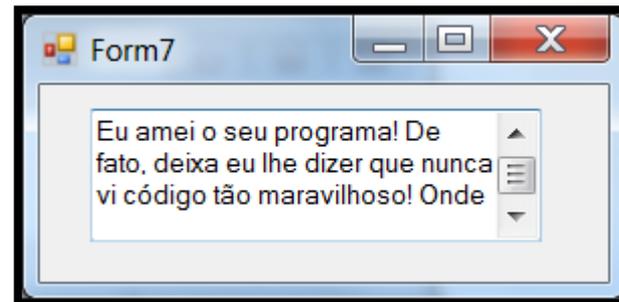
# Adicionando Barras de Rolagem

Para adicionar barras de rolagem em um controle **TextBox** com múltiplas linhas é preciso usar a propriedade **ScrollBars**.

**Importante:** A propriedade **MultiLine** deve estar configurada como **True**.

A propriedade **ScrollBars** tem as seguintes configurações:

- 0: None** (nenhuma barra de rolagem)
- 1: Horizontal**
- 2: Vertical**
- 3: Both** (duas barras de rolagem)



```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.Multiline = True
4         TextBox1.ScrollBars = ScrollBars.Vertical
5     End Sub
6 End Class
```

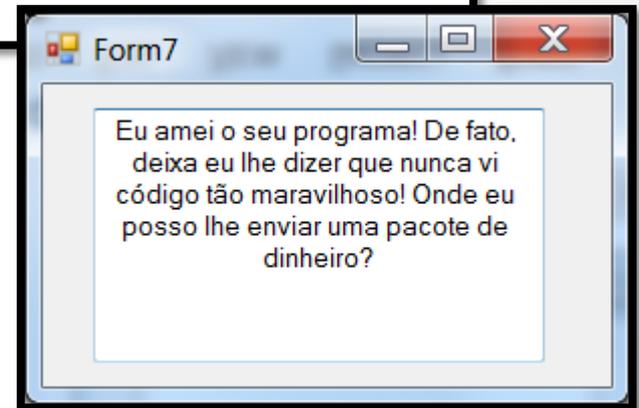
# Alinhando o texto de um Text Box

Você pode usar a propriedade **TextAlign** para ajustar o alinhamento do texto dentro de um componente **TextBox**.

Você pode alinhar o texto à direita, esquerda ou de forma centralizada. O alinhamento justificado infelizmente não existe.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.TextAlign = HorizontalAlignment.Center
4     End Sub
5 End Class
```



# Text Box em modo de leitura

- Usando a propriedade **ReadOnly**
  - Ajustando o valor dessa propriedade para **True** significará que o usuário não poderá digitar um texto.
  - Em tempo de execução o programa poderá escrever sem problema nenhum.
- Desabilitando um Text Box
  - Você pode desabilitar um text box ajustando a propriedade **Enabled** para **False**.
  - A diferença é que o componente ficará com uma cor cinza indicando que ele está inacessível.
- Usando Labels em vez de Text Boxes
  - Outra alternativa é usar o próprio componente Label para mostrar textos que são apenas de leitura.

**Faça:** Crie um projeto visual e teste essas três possibilidades.

# Selecionando e substituindo texto

Para trabalhar com parte do texto de um **TextBox**, você seleciona o texto desejado usando 3 propriedades:

- **SelectionLength**
- **SelectionStart**
- **SelectedText**

Exemplo:

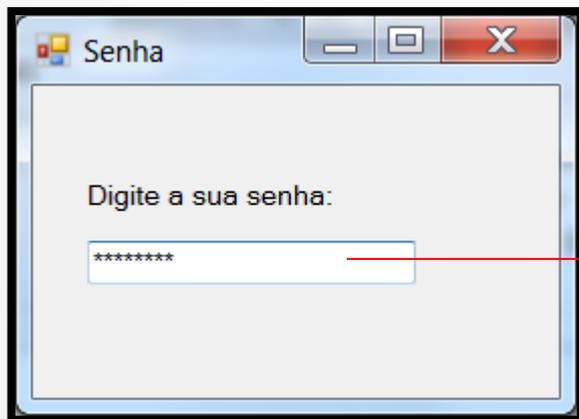
```
1 Public Class Form8
2     Private Sub Form8_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.SelectionStart = 0
4         TextBox1.SelectionLength = Len(TextBox1.Text)
5         TextBox1.SelectedText = "Olá Prof. Danilo Giacobbo"
6     End Sub
7 End Class
```

Faça: Teste a propriedade HideSelection.

# Criando um controle do tipo Password

Para transformar o seu componente **TextBox** em um **Password Box** você precisa alterar a propriedade **PasswordChar** e atribuir algum caractere para a mesma (geralmente é usado "\*").

Exemplo:



Não é possível copiar e colar o texto de um campo Password.

```
1 Public Class Form9
2     Private Sub Form9_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         TextBox1.PasswordChar = "*"
4         TextBox1.MaxLength = 8
5     End Sub
6 End Class
```

# Controlando a entrada de dados

Teste os dois exemplos e veja a diferença:

```
1 Public Class Form10
2     Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox1.KeyPress
3         If (e.KeyChar < "0" Or e.KeyChar > "9") Then
4             MsgBox("Por favor digite apenas números!")
5         End If
6     End Sub
7 End Class
```

```
1 Public Class Form10
2     Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox1.KeyPress
3         If (e.KeyChar < "0" Or e.KeyChar > "9") Then
4             e.Handled = True
5         End If
6     End Sub
7 End Class
```

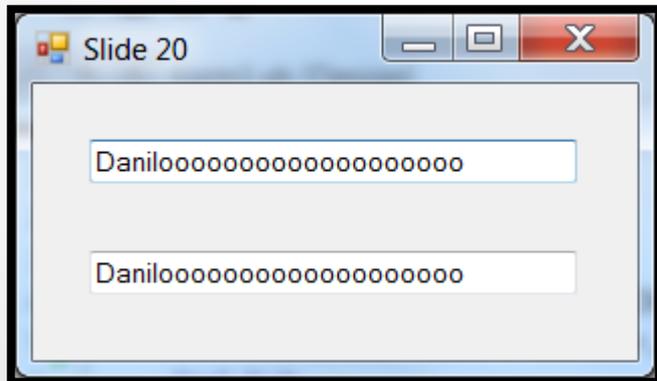
E as teclas de controle, tais como: **BackSpace** e **Del**? Como funcionam?

# Controlando a entrada de dados

Usando o evento **TextChanged** (que ocorre quando a pessoa faz uma mudança no Text Box) você consegue controlar também o que foi digitado.

Exemplo:

```
1 Public Class Form10
2     Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
3         TextBox2.Text = sender.Text
4     End Sub
5 End Class
```



# Criando um Text Box via código

Uma outra possibilidade de uso de um controle do tipo Text Box (e de outros controles também) é criar o mesmo a partir do código (em tempo de execução). Ele usa o construtor da classe para instanciar um novo objeto da classe TextBox e colocar o mesmo no form (a instrução **Me** se refere ao formulário corrente).

Exemplo:

```
1 Public Class Form10
2     Private Sub Form10_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         Dim MeuTextBox As New TextBox()
4         MeuTextBox.Size = New Size(150, 20)
5         MeuTextBox.Location = New Point(80, 20)
6         MeuTextBox.Text = "Text Box criado via código"
7         Me.Controls.Add(MeuTextBox)
8     End Sub
9 End Class
```

# Acessando o texto de um Rich Text Box

Para acessar o texto contido em um Rich Text Box, você pode usar duas propriedades: **Text** (plain text) e **Rtf** (rich text format).

Exemplo:

```
1 Public Class Form11
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         RichTextBox2.Text = RichTextBox1.Text
4         RichTextBox3.Rtf = RichTextBox1.Rtf
5     End Sub
6 End Class
```

# Formatando o texto do Rich Text Box

Para formatar o texto de um campo rich text box em *itálico*, **negrito**, sublinhado e ~~tachado~~) é necessário usar a propriedade **SelectionFont**.

Exemplo:

```
1 Public Class Form11
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         RichTextBox1.SelectionStart = RichTextBox1.Find("itálico")
4         Dim FonteItalica As New Font(RichTextBox1.Font, FontStyle.Italic)
5         RichTextBox1.SelectionFont = FonteItalica
6
7         RichTextBox1.SelectionStart = RichTextBox1.Find("negrito")
8         Dim FonteNegrito As New Font(RichTextBox1.Font, FontStyle.Bold)
9         RichTextBox1.SelectionFont = FonteNegrito
10
11        RichTextBox1.SelectionStart = RichTextBox1.Find("sublinhado")
12        Dim FonteSublinhado As New Font(RichTextBox1.Font, FontStyle.Underline)
13        RichTextBox1.SelectionFont = FonteSublinhado
14
15        RichTextBox1.SelectionStart = RichTextBox1.Find("tachado")
16        Dim FonteTachado As New Font(RichTextBox1.Font, FontStyle.Strikeout)
17        RichTextBox1.SelectionFont = FonteTachado
18
19        RichTextBox2.Text = RichTextBox1.Text
20        RichTextBox3.Text = RichTextBox1.Rtf
21    End Sub
22 End Class
```

# Abrir parágrafo no texto do Rich Text Box

Para controlar o espaçamento dos parágrafos e demais elementos do mesmo use as seguintes propriedades:

- ❖ **SelectionIndent**
- ❖ **SelectionHangingIndent**
- ❖ **SelectionRightIndent**

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         RichTextBox1.SelectAll()
4         RichTextBox1.SelectionIndent = 20
5         RichTextBox1.SelectionHangingIndent = -25
6         RichTextBox1.SelectionRightIndent = 10
7         RichTextBox1.DeselectAll()
8     End Sub
9 End Class
```

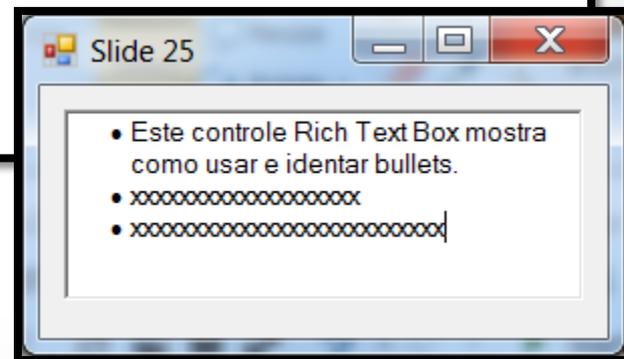
# Adicionando Bullets no Rich Text Box

Para adicionar os famosos bullets em seu controle de rich text box use as seguintes propriedades:

- ❖ **BulletIndent**
- ❖ **SelectionBullet**

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         RichTextBox1.Text = "Este controle Rich Text Box mostra como usar e identar bullets."
4         RichTextBox1.SelectAll()
5         RichTextBox1.SelectionIndent = 20
6         RichTextBox1.BulletIndent = 10
7         RichTextBox1.SelectionBullet = True
8         RichTextBox1.DeselectAll()
9     End Sub
10 End Class
```



# Adicionando Cores no Rich Text Box

Para mudar a cor do texto do seu controle de rich text box use a propriedade **SelectionColor** e a enumeração **Colors**.

Exemplo:

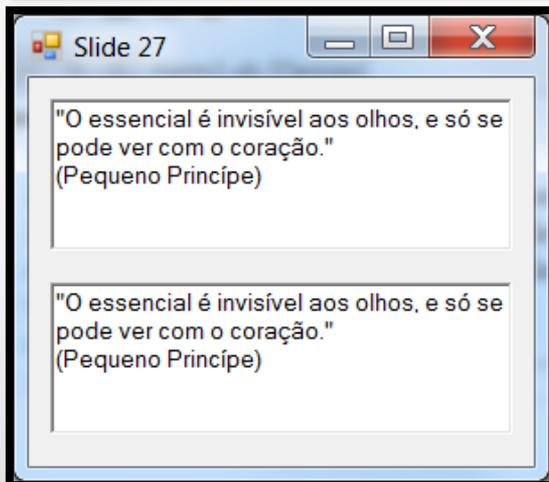
```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         RichTextBox1.Text = "Este controle Rich Text Box suporta várias cores de fonte. " & _
4             "Exemplo: vermelho, verde e azul."
5         RichTextBox1.SelectionStart = RichTextBox1.Find("vermelho")
6         RichTextBox1.SelectionColor = Color.Red
7
8         RichTextBox1.SelectionStart = RichTextBox1.Find("verde")
9         RichTextBox1.SelectionColor = Color.Green
10
11        RichTextBox1.SelectionStart = RichTextBox1.Find("azul")
12        RichTextBox1.SelectionColor = Color.Blue
13    End Sub
14 End Class
```

# Salvando e Carregando arquivos RTF

Você pode usar o método **SaveFile** para salvar o texto de um Rich Text Box no disco e o método **LoadFile** para ler ele de volta.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         RichTextBox1.SaveFile("C:/texto.rtf")
4         RichTextBox2.LoadFile("C:/texto.rtf")
5     End Sub
6 End Class
```



# Alinhando o texto do Rich Text Box

Você pode ajustar o alinhamento do texto de um rich text box parágrafo por parágrafo usando a propriedade **SelectionAlignment**. Ela possui os seguintes valores:

- **HorizontalAlignment.Left**
- **HorizontalAlignment.Right**
- **HorizontalAlignment.Center**

Exemplo:

```
1 Public Class Form11
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
4         RichTextBox2.SelectionAlignment = HorizontalAlignment.Left
5         RichTextBox3.SelectionAlignment = HorizontalAlignment.Right
6     End Sub
7 End Class
```

# Criando um Rich Text Box via código

Uma outra possibilidade de uso de um controle do tipo Rich Text Box (e de outros controles também) é criar o mesmo a partir do código. Ele usa o construtor da classe para instanciar um novo objeto da classe RichTextBox e coloca o mesmo no form (a instrução **Me** se refere ao formulário padrão).

Exemplo:

```
1 Public Class Form11
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         Dim MeuRichTextBox As New RichTextBox()
4         MeuRichTextBox.Size = New Size(150, 100)
5         MeuRichTextBox.Location = New Point(70, 20)
6         MeuRichTextBox.Text = "Rich Text Box criado via código"
7         Me.Controls.Add(MeuRichTextBox)
8     End Sub
9 End Class
```

# Usando Labels

- **Label** é um controle gráfico que apresenta apenas texto sem a possibilidade de edição.
- Em muitos casos a aparência de um **Label** é melhor do que a de um **Text Box**.
- Para exibir o resultado de um cálculo ou uma mensagem informativa no form a melhor opção é usar Labels.

## Dicas:

Para deixar um **Label** parecido com um **Text Box** e vice-versa faça as seguintes mudanças:

### Text Box com cara de Label

- o Altere a propriedade **BackColor** para o valor **Control**;
- o Altere a propriedade **ReadOnly** para o valor **True**; e
- o Altere a propriedade **BorderStyle** para o valor **None**;

### Label com cara de Text Box

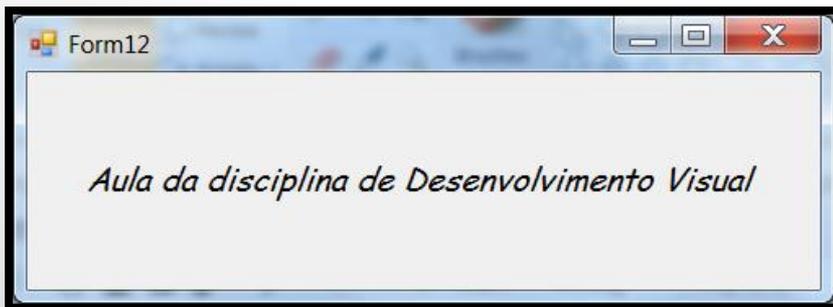
- o Altere a propriedade **BackColor** para o valor **Window**; e
- o Altere a propriedade **BorderStyle** para o valor **Fixed3D**.

# Labels - Formatando Texto

- Para que o texto do Label se ajuste corretamente ao tamanho do mesmo é necessário usar a propriedade **AutoSize** com valor **True**.
- A propriedade **Font** permite que você altere a formatação do texto do Label assim como foi visto anteriormente com o controle Text Box.

Exemplo:

```
1 Public Class Form12
2     Private Sub Form12_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         Label1.Text = "Aula da disciplina de Desenvolvimento Visual"
4         Label1.AutoSize = True
5         Label1.Font = New Font("Comic Sans MS", 12, FontStyle.Italic)
6     End Sub
7 End Class
```



# Labels - Alinhando Texto

Para alinhar o texto contido em um **Label** é necessário usar a propriedade **TextAlign** em modo de design ou tempo de execução. Ela possui os valores oriundos da enumeração **ContentAlignment**:

- BottomCenter
- BottomLeft
- BottomRight
- MiddleCenter
- MiddleLeft
- MiddleRight
- TopCenter
- TopLeft
- TopRight

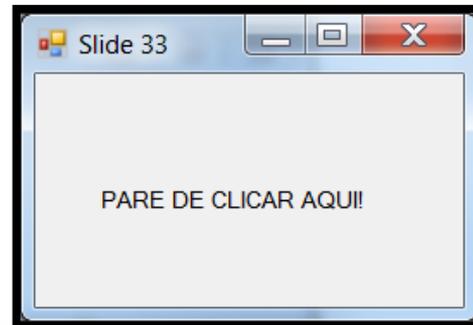
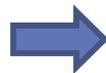
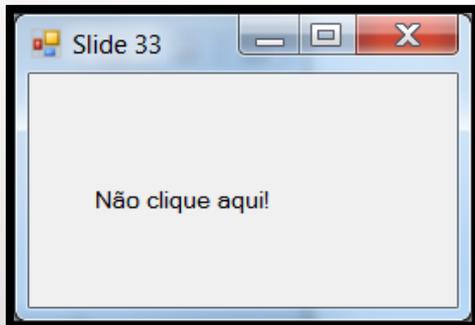
**Faça:** Crie um **Label** e teste essa propriedade em modo de design.

# Labels - Manipulando Eventos

- Labels possuem eventos tais como **Click** e **DoubleClick**.
- Eles não possuem eventos relacionados com o teclado.

Exemplo:

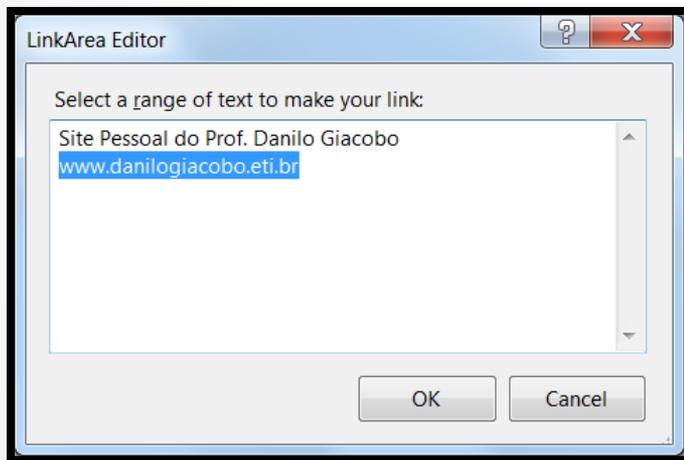
```
1 Public Class Form12
2     Private Sub Form12_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         Label1.Cursor = Cursors.Hand
4         Label1.Text = "Não clique aqui!"
5     End Sub
6
7     Private Sub Label1_DoubleClick(sender As Object, e As EventArgs) Handles Label1.DoubleClick
8         Label1.Text = "PARE DE CLICAR AQUI!"
9     End Sub
10 End Class
```



# Criando um LinkLabel

- Este controle possui propriedades e comportamentos similares ao do **Label** normal mas com suporte a **hyperlinks**.
- Para adicionar hyperlinks você usa a propriedade **LinkArea**.

Exemplo em modo de design:



- Você também pode usar as propriedades **LinkColor**, **VisitedLinkColor** e **ActiveLinkColor** para mudar a aparência de um hyperlink.

# Criando um LinkLabel

- Quando a pessoa clica no hyperlink o evento **LinkClicked** é gerado.
- No exemplo abaixo ele é direcionado para o site relacionado no hyperlink.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         LinkLabel1.Text = "http://www.danilogiacobo.eti.br"
4         LinkLabel1.LinkColor = Color.Black
5         LinkLabel1.VisitedLinkColor = Color.DarkRed
6         LinkLabel1.ActiveLinkColor = Color.YellowGreen
7     End Sub
8
9     Private Sub LinkLabel1_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles _
10        LinkLabel1.LinkClicked
11        LinkLabel1.LinkVisited = True
12        System.Diagnostics.Process.Start("http://www.danilogiacobo.eti.br")
13    End Sub
14 End Class
```

# Criando um LinkLabel em código

```
1 Public Class Form13
2     Private WithEvents LinkLabel1 As LinkLabel
3
4     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
5         LinkLabel1 = New LinkLabel()
6         LinkLabel1.AutoSize = True
7         LinkLabel1.Location = New Point(15, 15)
8         LinkLabel1.Size = New Size(135, 15)
9         LinkLabel1.Text = "Interessado em aprender? Clique aqui!"
10
11         LinkLabel1.Links.Add(15, 8, "informação")
12         LinkLabel1.Links.Add(32, 4, "www.danilogiacobo.eti.br")
13
14         AddHandler LinkLabel1.LinkClicked, AddressOf Me.LinkLabel1_LinkClicked
15         Me.Controls.Add(LinkLabel1)
16     End Sub
```

# Linkando para outro Form e para a Web

```
18 Private Sub LinkLabel1_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs)
19     LinkLabel1.Links(LinkLabel1.Links.IndexOf(e.Link)).Visited = True
20
21     If (e.Link.LinkData.ToString() = "informação") Then
22         Dim JanelaInformacao As New Form14()
23         JanelaInformacao.Show()
24     Else
25         System.Diagnostics.Process.Start(e.Link.LinkData.ToString())
26     End If
27 End Sub
28 End Class
```

# Referências Bibliográficas

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