



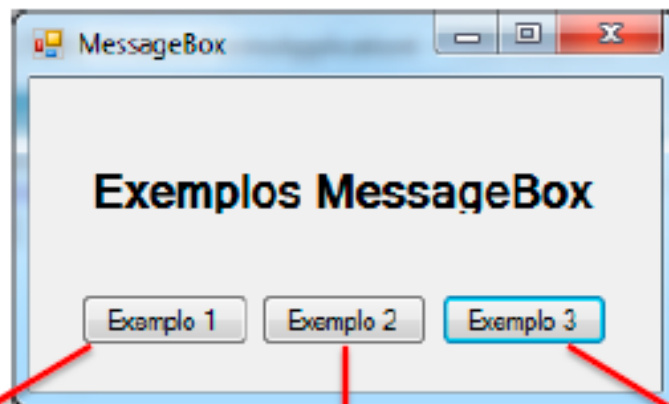
Habilitação Profissional Técnica de Nível Médio de **TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

DESENVOLVIMENTO DE SISTEMAS

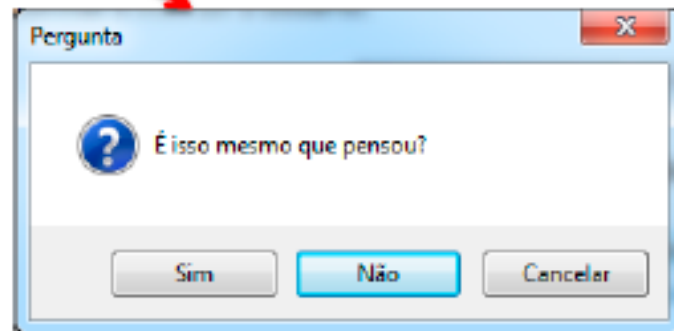
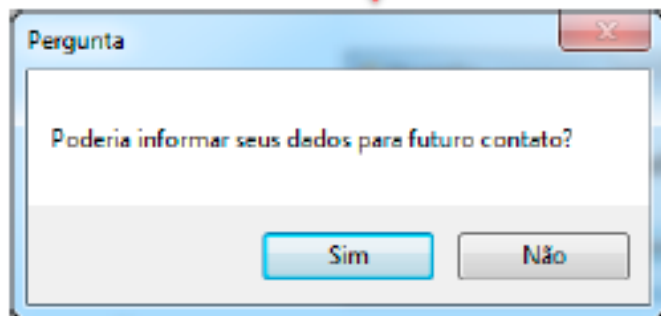
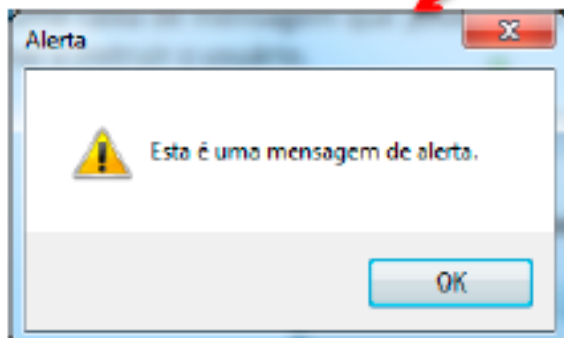
MessageBox

Exibe uma caixa de mensagem que pode conter texto, botões e símbolos que possam informar e instruir o usuário.

Crie um formulário mensagem contendo esta interface gráfica.



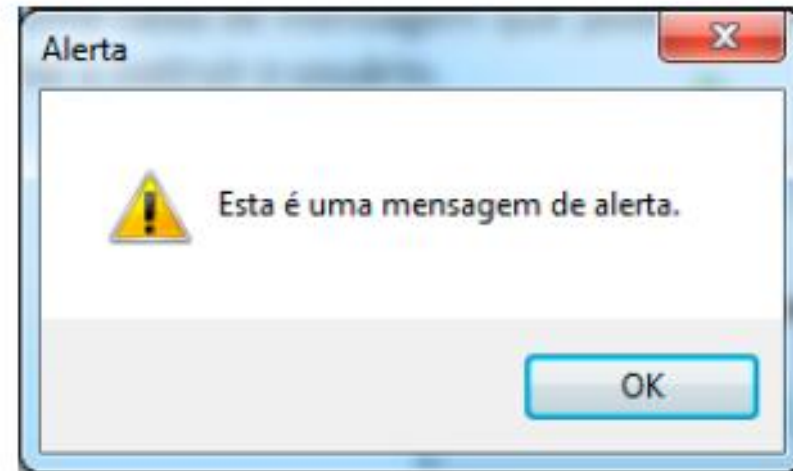
Quando clicar em cada um dos botões, uma caixa de mensagem será exibida.



MessageBox

Quando clicar em Exemplo 1, estará sendo acionado o seguinte evento:

```
private void btnEx1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Esta é uma mensagem de alerta.", "Alerta",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
```



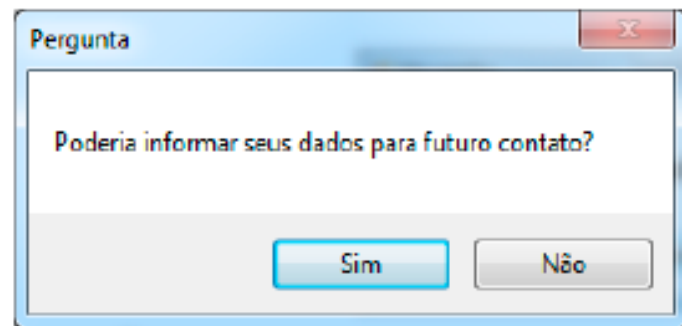
MessageBox

Quando clicar em Exemplo 2, estará sendo acionado o seguinte evento:

```
private void btnEx2_Click(object sender, EventArgs e)
{
    string message = "Poderia informar seus dados para futuro contato?";
    string caption = "Pergunta";
    MessageBoxButtons buttons = MessageBoxButtons.YesNo;
    DialogResult result;

    result = MessageBox.Show(message, caption, buttons);

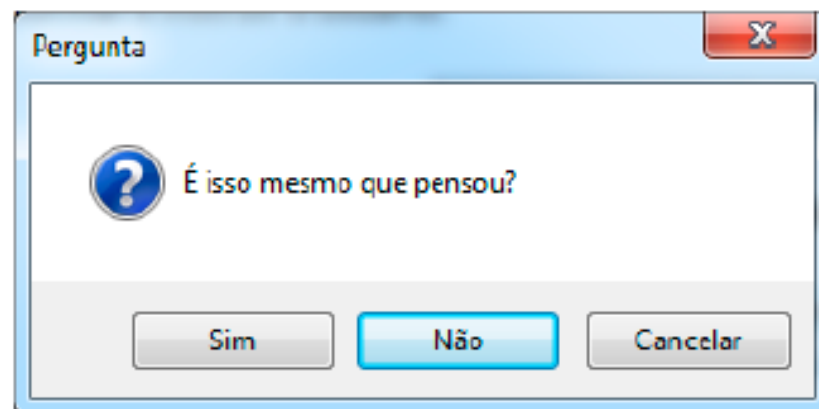
    if (result == System.Windows.Forms.DialogResult.Yes)
    {
        MessageBox.Show("OK Obrigado! Em breve entrarei  
em contato!", "Resposta SIM", MessageBoxButtons.OK);
    }
    else
    {
        MessageBox.Show("OK Sem problemas! ", "Resposta NÃO", MessageBoxButtons.OK);
    }
}
```



MessageBox

Quando clicar em Exemplo 3, estará sendo acionado o seguinte evento:

```
private void btnEx3_Click(object sender, EventArgs e)
{
    DialogResult resultado = MessageBox.Show("É isso mesmo que pensou?",
        "Pergunta",
        MessageBoxButtons.YesNoCancel,
        MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button2);
}
```



MessageBox

Métodos	Descrição
Show(String)	Exibe uma caixa de mensagem com o texto especificado.
Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator, Object)	Exibe uma caixa de mensagem com o texto especificado, legenda, botões, ícone, botão padrão, opções e botão de Ajuda, usando o arquivo de Ajuda especificado, HelpNavigator o tópico da Ajuda.
Show(String, String)	Exibe uma caixa de mensagem com o texto especificado e a legenda.
Show(String, String, MessageBoxButtons)	Exibe uma caixa de mensagem com botões, legenda e texto especificado.
Show(IWin32Window, String, String)	Exibe uma caixa de mensagem na frente do objeto especificado e com o texto especificado e a legenda.

<http://msdn.microsoft.com/pt-br/library/system.windows.forms.messagebox.show%28v=vs.90%29.aspx>

MessageBoxButtons

Membros	Descrição
OK	A caixa de mensagem contém um botão OK.
OKCancel	A caixa de mensagem contém botões OK e Cancelar.
AbortRetryIgnore	A caixa de mensagem contém botões Abortar, repetir e ignorar.
YesNoCancel	A caixa de mensagem conterá Sim, não e Cancelar botões.
YesNo	A caixa de mensagem contém botões Sim e não.
RetryCancel	A caixa de mensagem contém botões de Repetir e Cancelar.

<http://msdn.microsoft.com/pt-br/library/system.windows.forms.messageboxbuttons%28v=vs.90%29.aspx>

MessageBoxIcon

Membros	Descrição
Question	Símbolo que consiste em um ponto de interrogação em um círculo.
Exclamation	Símbolo que consiste em um ponto de exclamação dentro de um triângulo com um plano de fundo amarelo.
Stop	Símbolo que consiste em X branco em um círculo com um plano de fundo vermelho.
Error	Símbolo que consiste em X branco em um círculo com um plano de fundo vermelho.
Warning	Símbolo que consiste em um ponto de exclamação dentro de um triângulo com um plano de fundo amarelo.
Information	Símbolo consiste em uma letra em minúsculas i em um círculo.

<http://msdn.microsoft.com/pt-br/library/system.windows.forms.messageboxicon%28v=vs.90%29.aspx>

MessageBoxDefaultButton

Membros	Descrição
Button1	O primeiro botão na caixa de mensagem é o botão padrão.
Button2	O segundo botão na caixa de mensagem é o botão padrão.
Button3	O terceiro botão na caixa de mensagem é o botão padrão.

<http://msdn.microsoft.com/pt-br/library/system.windows.forms.messageboxdefaultbutton%28v=vs.90%29.aspx>

DialogResult

Especifica identificadores para indicar o valor retornado de uma caixa de diálogo.

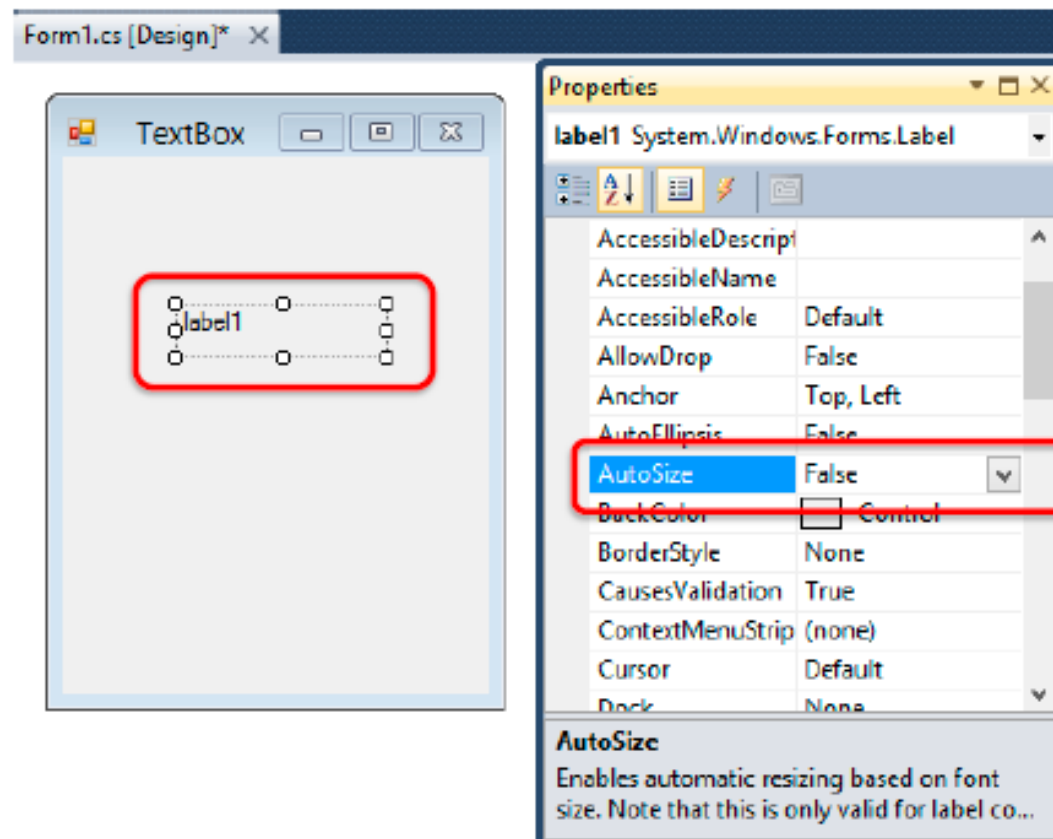
Membros	Descrição
None	Nothing é retornado pelo formulário da caixa de diálogo.
OK	O valor retornado de caixa de diálogo é OK
Cancel	O valor retornado de caixa de diálogo é Cancel
Abort	O valor retornado de caixa de diálogo é Abort
Retry	O valor retornado de caixa de diálogo é Retry
Ignore	O valor retornado de caixa de diálogo é Ignore

<http://msdn.microsoft.com/pt-br/library/system.windows.forms.dialogresult%28v=vs.90%29.aspx>

Label

A Label

O Label é usado para adicionar texto como forma de adicionar etiquetas para identificar a funcionalidade de outros controles. Arraste um controle de rótulo da caixa de ferramentas para o formulário. Por padrão, ele terá um texto inicial **label1**.



Label

Propriedade	Descrição
AutoSize	Se for verdade, o tamanho das fronteiras do rótulo no designer será redimensionada automaticamente dependendo do texto dentro dele.
BorderStyle	Especifica o tipo de borda ao redor do rótulo.
Font	Usado para alterar as propriedades de fonte do texto dentro do controle de rótulo.
Text	O texto da etiqueta.
TextAlign	O alinhamento do texto dentro do controle <i>Label</i>

[http://msdn.microsoft.com/pt-br/library/system.windows.forms.label\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.windows.forms.label(v=vs.110).aspx)

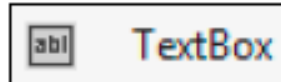
Label

Nestes exemplos estão sendo utilizadas as propriedades:

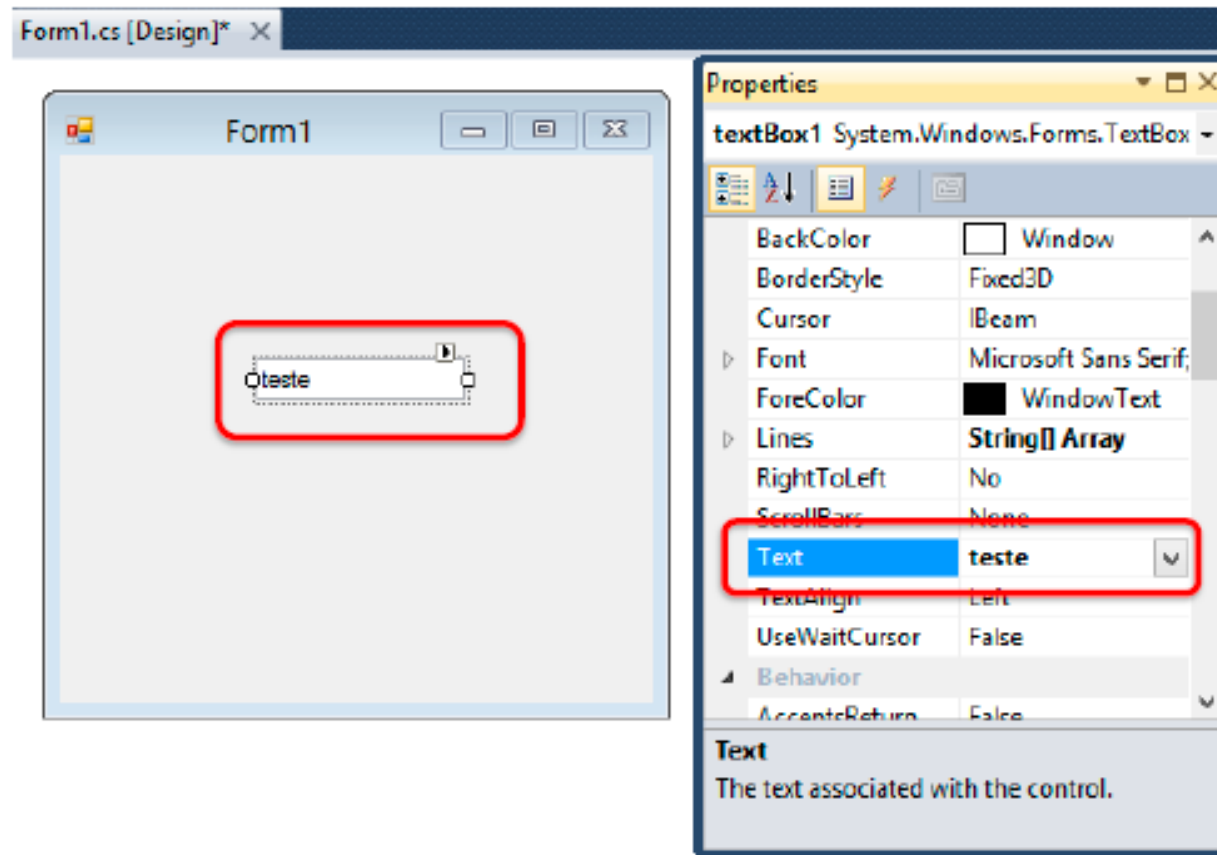
- AutoSize
- BorderLayout
- BackColor
- Font
- ForeColor
- Size
- TextAlign



TextBox



É um componente básico de entrada de dados em formulário. Basta o usuário digitar os dados de entrada dentro da caixa de texto. O texto digitado pode ser acessado usando a propriedade **Text**.



TextBox

Propriedade	Descrição
AcceptsReturn	Usado com várias linhas. Diz se a tecla de retorno está incluída na entrada. O retorno será convertido em uma sequência de escape \n.
Enabled	Selecionado como <i>false</i> para deixar a caixa de texto somente como leitura.
Font	As propriedades de fonte que serão usadas pela caixa de texto.
Lines	As linhas de texto em uma caixa de texto de várias linhas.
Multiline	Defina como <i>true</i> para permitir várias linhas em uma caixa de texto.
Text	Texto dentro da caixa de texto.
PasswordChar	Aceita um caractere que será usado para mascarar cada caractere digitado pelo usuário.
ReadOnly	Determina se o texto pode ser editado.
Visible	Determina se a caixa de texto é visível dentro do formulário.
WordWrap	Usado com várias linhas. Defina como <i>true</i> para permitir a quebra automática de palavras.

[http://msdn.microsoft.com/pt-br/library/system.windows.forms.textbox\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.windows.forms.textbox(v=vs.110).aspx)

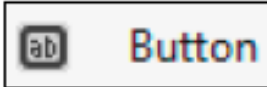
TextBox

Neste exemplo foram alteradas as propriedades:

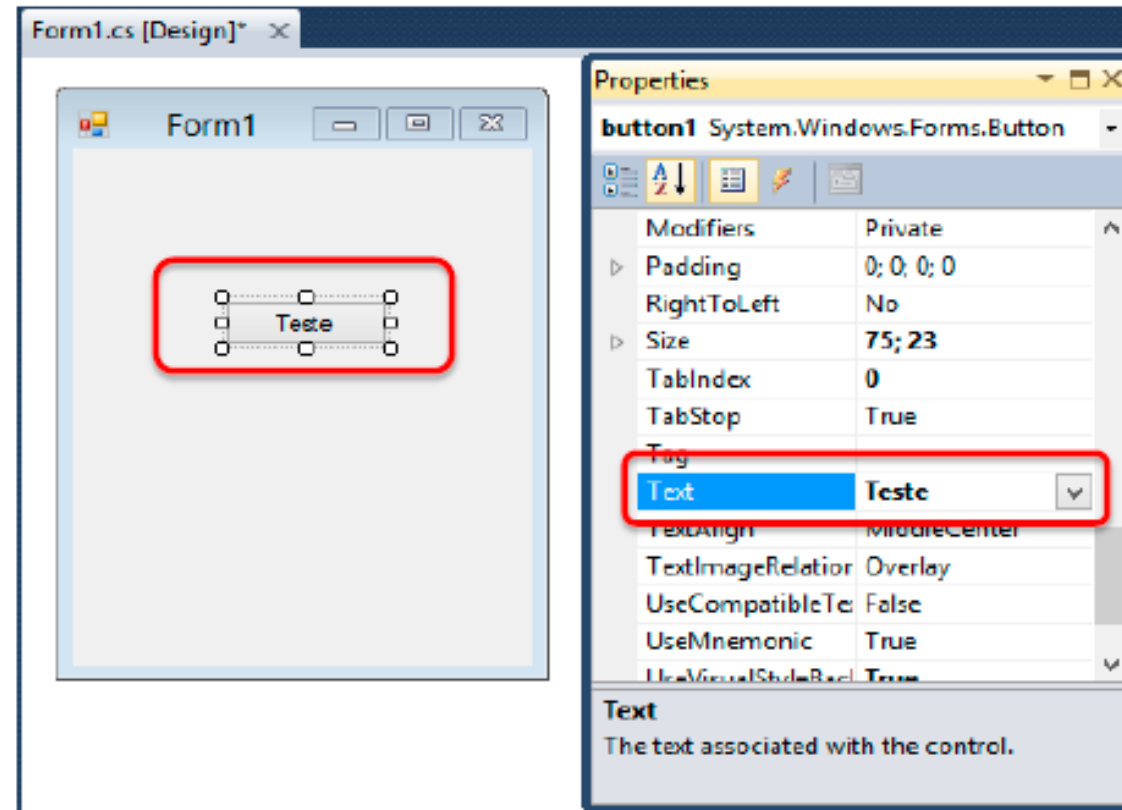
- BackColor
- Enable
- ForeColor
- PasswordChar
- Text
- WordWrap



Button



O Button é geralmente usado para executar comandos quando é clicado. Quando um botão é clicado, você determina os códigos que serão usados. São normalmente utilizados para confirmar ou cancelar uma ação, para executar ações diferentes.



Button

Propriedade	Descrição
AutoEllipsis	Especifica se é necessário acrescentar pontos (...) quando o texto é muito longo e não pode caber dentro do botão.
AutoSize	Especifica se o botão irá redimensionar automaticamente para caber seu conteúdo.
FlatStyle	Obtém ou define aparência plana ao controle botão.
Enabled	Se definido como <i>false</i> , o botão não pode ser clicado ou receber foco.
Image	Uma imagem opcional que você pode colocar dentro do componente.
ImageAlign	O alinhamento da imagem no botão.
Text	O texto dentro do botão.
Visible	Diz se o botão está visível ou não.

[http://msdn.microsoft.com/pt-br/library/system.windows.forms.button_properties\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.windows.forms.button_properties(v=vs.110).aspx)

Button

Evento	Descrição
Click	Ocorre quando você clica no botão.
Leave	Ocorre quando o ponteiro do mouse sai do controle.
LocationChanged	Ocorre quando a localização do botão é alterado.
MouseDown	Ocorre quando o ponteiro do mouse está sobre o controle e um botão do mouse é pressionado.
MouseEnter	Ocorre quando o ponteiro do mouse entra no controle.
MouseHover	Ocorre quando o ponteiro do mouse é posicionado sobre o controle.
MouseUp	Ocorre quando o ponteiro do mouse está sobre o controle e um botão do mouse seja liberado.

[http://msdn.microsoft.com/pt-br/library/system.windows.forms.button_events\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.windows.forms.button_events(v=vs.110).aspx)

Button

Para criar um evento quando o botão for clicado, basta dar um duplo clique no botão, que na sequência aparecerá uma área para codificação.

No caso dê um duplo clique no botão *Bem vindo!*

Aparecerá a seguinte tela:

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnBemVindo_Click(object sender, EventArgs e)
        {
        }
    }
}
```

É nesta área que
iremos codificar a
ação do botão
Bem Vindo!

Button

Para criar um evento quando o botão for clicado, basta dar um duplo clique no botão, que na sequencia aparecerá uma área para codificação.

No caso dê um duplo clique no botão *Bem vindo!*

Aparecerá a seguinte tela:

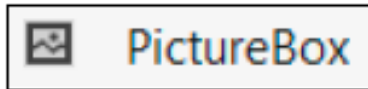
```
private void btnBemVindo_Click(object sender, EventArgs e)
{
    MessageBox.Show("Seja bem vindo ao mundo da programação!");
}
```

Este é o nome atributo na propriedade **(name)** do botão

Este é o comando que será executado quando o botão for clicado

Agora você poderá criar uma ação para o botão *Até mais!*

PictureBox



A função da **picturebox** é permitir a exibição de uma imagem qualquer.

O evento default do componente é o **Click**.

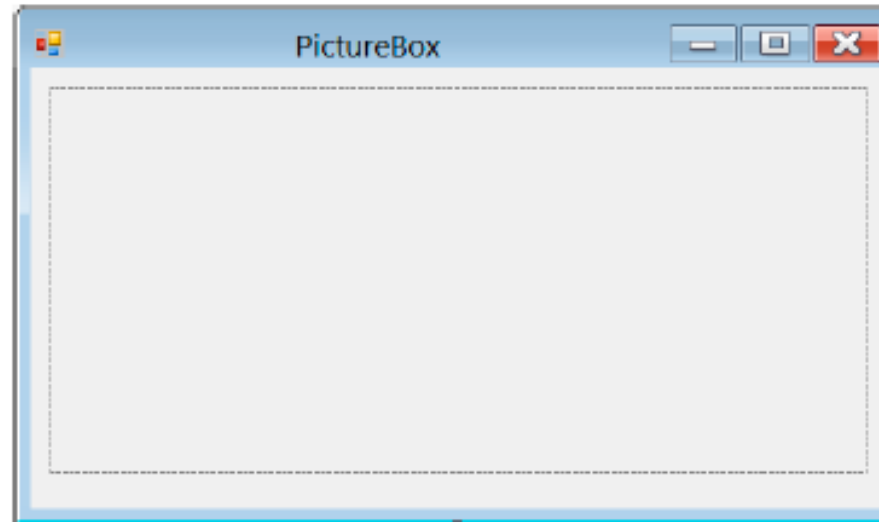
```
private void pcbFoto_Click(object sender, EventArgs e)
{
}
}
```

As propriedades que iremos modificar no exemplo são:

(name) → Nome do componente

BorderStyle → Estilo da borda

- **None (Sem borda)**
- **FixedSingle (Borda Simples)**
- **Fixed3D (Borda em 3D)**




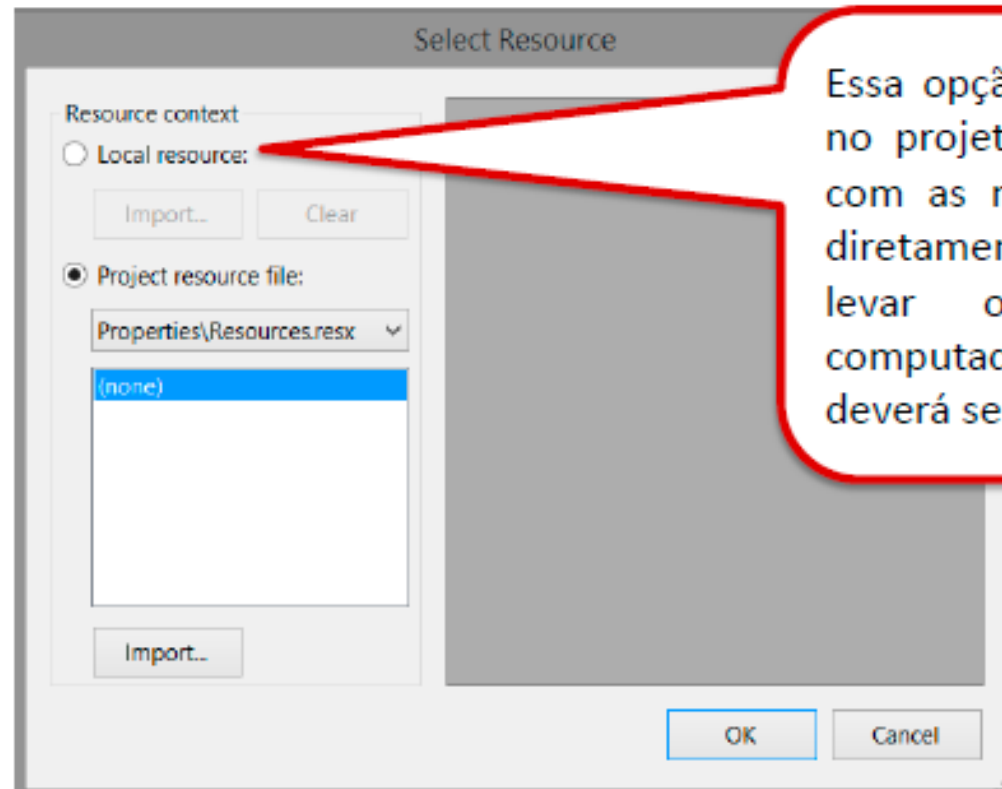
PictureBox

Propriedade **Image** → Indica qual a imagem que será carregada no componente.



Para inserir uma imagem dentro componente devemos seguir os seguintes passos:

1. Clicar no botão  da propriedade **Image**, em seguida abrirá a tela abaixo:



Essa opção permite inserir imagens no projeto, mas cria-se uma pasta com as mesmas sem incorpora-las diretamente ao projeto. Ou seja, ao levar o projeto em outro computador, a pasta de imagens deverá ser copiada manualmente.

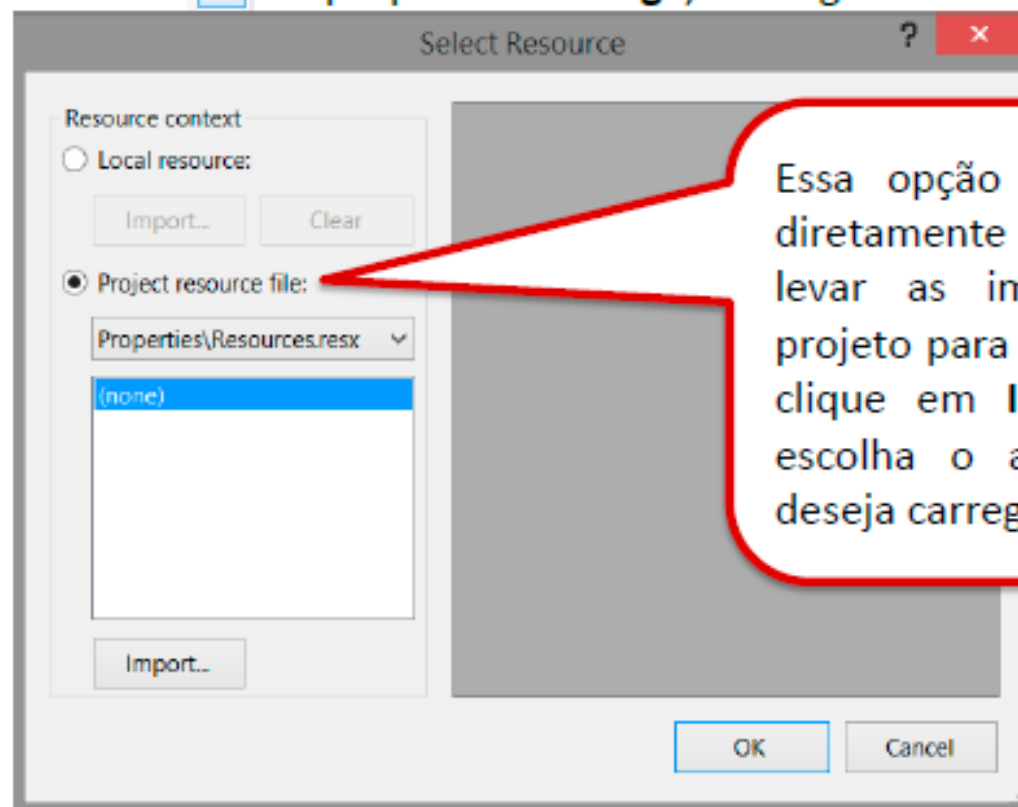
PictureBox

Propriedade **Image** → Indica qual a imagem que será carregada no componente.



Para inserir uma imagem dentro componente devemos seguir os seguintes passos:

1. Clicar no botão [...] da propriedade **Image**, em seguida abrirá a tela abaixo:



Essa opção permite inserir imagens diretamente no projeto, permitindo levar as imagens juntamente com projeto para outro computador. Depois clique em **Import...** E na sequência escolha o arquivo de imagem que deseja carregar. E depois clique em **OK**.

PictureBox

SizeMode ➔ Como a imagem será exibida no componente.

- **Normal** - Não altera o tamanho da imagem e nem o tamanho do componente, se a imagem for muito grande, só aparecerá uma parte dela.
- **StretchImage** - Ajustar a imagem de acordo com o tamanho do componente
- **AutoSize** - Ajustar o componente de acordo com o tamanho da imagem
- **CenterImage** - Centralizar a imagem dentro do componente
- **Zoom** - Aplicar um ZOOM na imagem dentro do componente

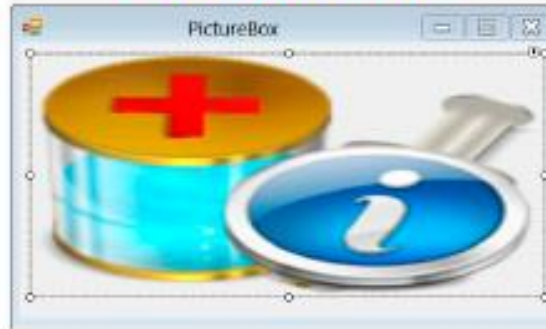
PictureBox

Para alterar a exibição da imagem dentro do componente, utilizaremos a propriedade **SizeMode**.

NORMAL



STRECHIMAGE



AUTOSIZE



CENTERIMAGE



ZOOM

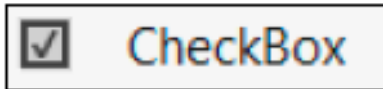


PictureBox

Propriedade	Descrição
BackgroundImage	Obtém ou define a imagem de plano de fundo exibida no controle.
BorderStyle	Indica o estilo de borda para o controle.
Image	Obtém ou define a imagem que é exibida por PictureBox.
Width	Obtém ou define a largura do controle.
Height	Obtém ou define a altura do controle.
ResizeRedraw	Obtém ou define um valor indicando se o controle é redesenha quando redimensionado.

[http://msdn.microsoft.com/pt-br/library/system.windows.forms.picturebox\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.windows.forms.picturebox(v=vs.110).aspx)

CheckBox



A função do **checkbox** é permitir selecionar mais de uma opção ao mesmo tempo. A propriedade que faz o controle, se está selecionado ou não é a **Checked**.

Exemplo:

Checked = false (Opção não selecionada)



☐ checkBox1

Checked = true (Opção selecionada)

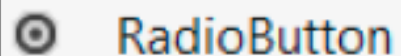


☒ checkBox1

O evento default para o checkbox quando clicamos sobre ele é o **CheckedChanged**, ou seja, quando clicamos estamos alterando a propriedade **checked** do componente para **true** ou **false**.

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}
}
```

RadioButton



A função do **radiobutton** é permitir selecionar somente uma opção por vez. A propriedade que faz o controle, se está selecionado ou não é a **Checked**.

Exemplo:

Checked = false (Opção não selecionada)



☐ radioButton1

Checked = true (Opção selecionada)



☒ radioButton1

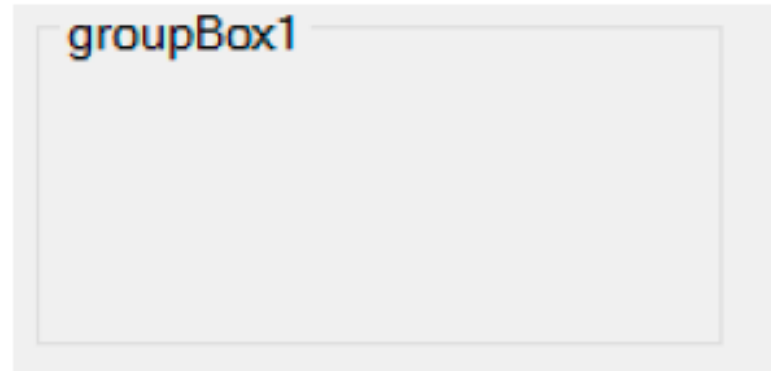
O evento default para o checkbox quando clicamos sobre ele é o **CheckedChanged**, ou seja, quando clicamos estamos alterando a propriedade **checked** do componente para **true** ou **false**.

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
}
}
```


GroupBox



A função do **groupbox** é permitir agrupar um ou mais componentes dentro de um formulário.



Para inserir um ou mais componentes dentro do groupbox, basta **criá-lo (arrastar e soltar)** dentro do mesmo.

O evento default para o groupbox é o **Enter**, ou seja, quando pressionamos a tecla enter do teclado sobre ele.

```
private void groupBox1_Enter(object sender, EventArgs e)
{
}
}
```

Exemplo

Criaremos uma aplicação que quando selecionamos uma das opções, a imagem exibirá conteúdos diferentes e vamos definir se queremos ou não bordas na imagem.



Exemplo – Código Fonte

```
private void rdoImagem1_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.Image = Properties.Resources.imgAlimentos;
}
```

```
private void rdoImagem2_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.Image = Properties.Resources.imgH
}
```

Buscar a imagem que foi importada para dentro do projeto.

```
private void rdoSemImagem_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.Image = null;
}
```

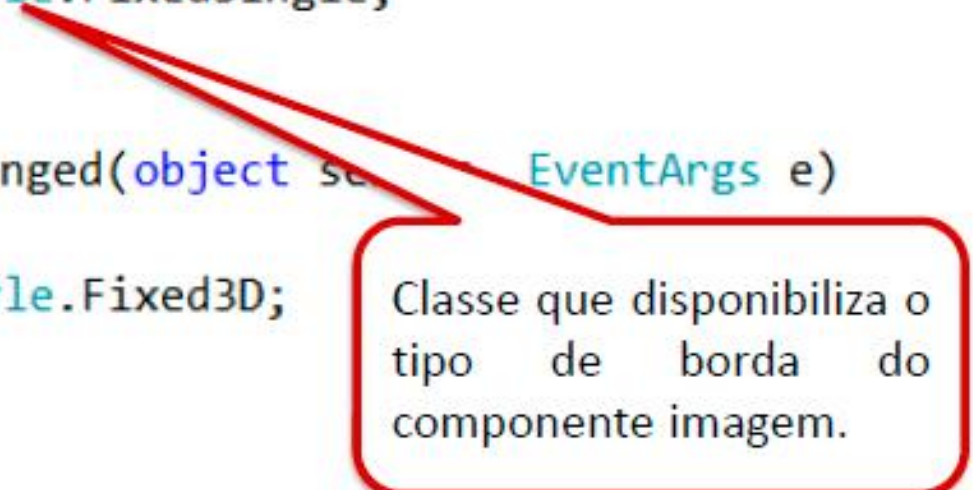
Limpar o componente de imagem.

Exemplo – Código Fonte

```
private void rdoSemBordas_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.BorderStyle = BorderStyle.None;
}
```

```
private void rdoBordasFixa_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.BorderStyle = BorderStyle.FixedSingle;
}
```

```
private void rdoBordas3D_CheckedChanged(object sender, EventArgs e)
{
    pcbFoto.BorderStyle = BorderStyle.Fixed3D;
}
```



Classe que disponibiliza o tipo de borda do componente imagem.

Exemplo – Código Fonte

```
private void chkVisivelInvisivel_CheckedChanged(object sender, EventArgs e)
{
    if (chkVisivelInvisivel.Checked)
    {
        pcbFoto.Visible = true;
    }
    else
    {
        pcbFoto.Visible = false;
    }
}

private void btnSair_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Através da propriedade **checked**, é validado se aparecerá ou não a borda na imagem.

Sair do sistema.