

Prof. Roberto de Castro

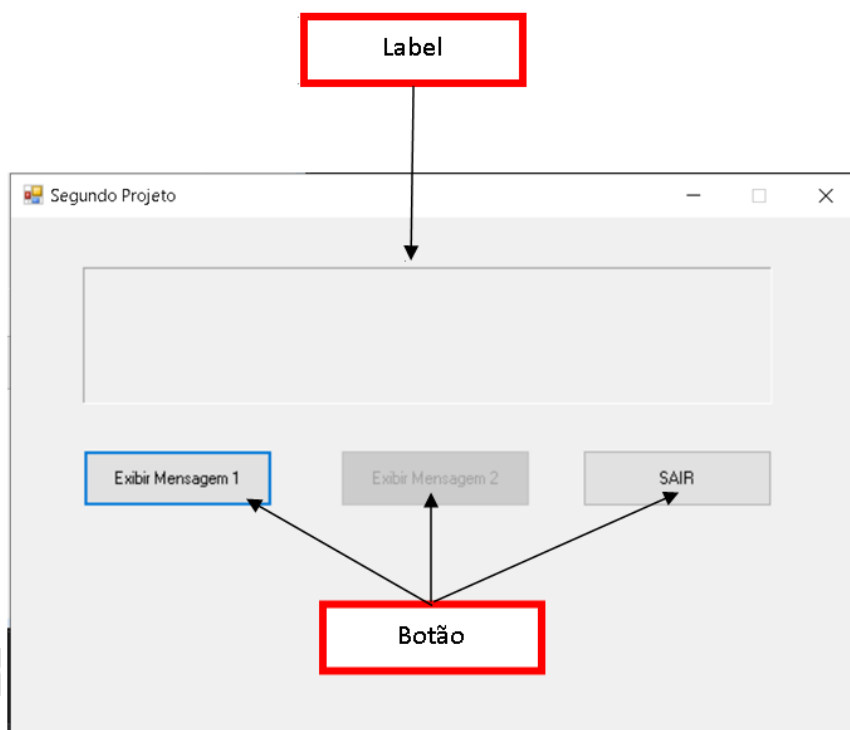
Escola Técnica Estadual “São Paulo” - ETESP Linguagem C# - Visual Studio 2019

Segundo Projeto

Para o desenvolvimento deste segundo projeto seguiremos a mesma metodologia já estudada anteriormente. São basicamente 4 (quatro etapas):

- Primeira Etapa → Desenho da interface gráfica;
- Segunda Etapa → Configuração das propriedades e;
- Terceira Etapa → Programação dos “eventos”.
- Quarta etapa → Testes

Seguindo a metodologia acima, nosso segundo projeto terá como aparência final, o modelo abaixo:



O projeto tem: um formulário; um label e três botões.

Vamos abrir o Visual Studio, criar um “Projeto C# → Windows → Área de Trabalho”: Aplicativo do Windows Forms - NET Framework.

O nome do projeto será “Exe002”.

*** Lembre-se de informar o local em que seu projeto será gravado. ***

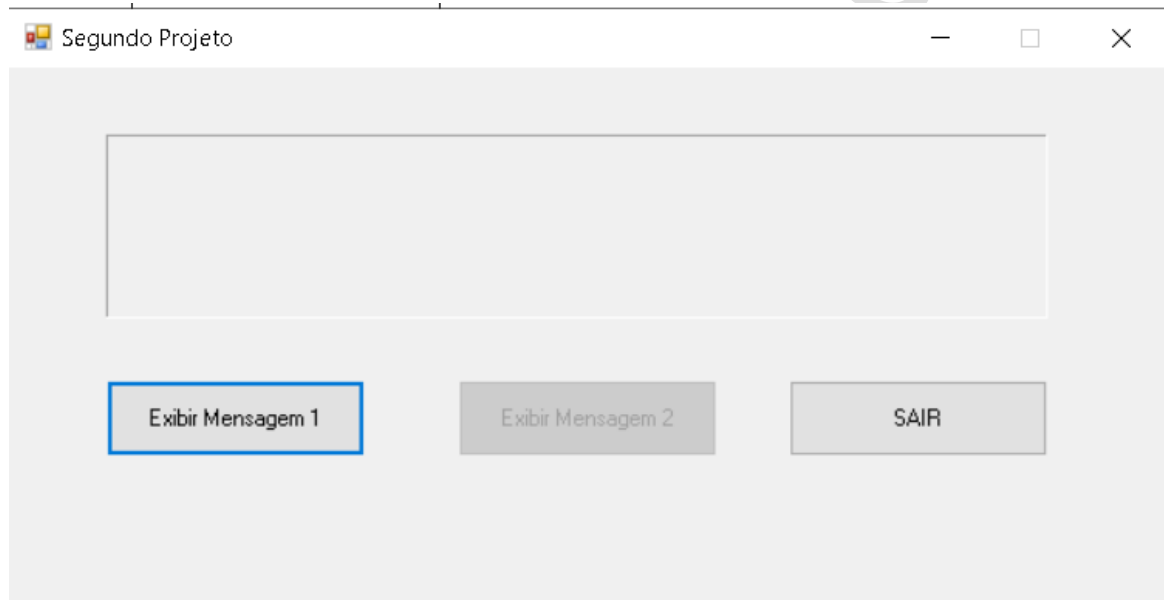
QUADRO DE PROPRIEDADES

CONTROLE	PROPRIEDADE	CONTEÚDO
Form	Name	FrmExe002
	MaximizeBox	False
	StartPosition	CenterScreen
	Text	Segundo Projeto

Prof. Roberto de Castro

Label	Name	LblMensagem
	AutoSize	False
	BorderStyle	Fixed3D
	Font	16
	Text	(deixar a propriedade "text" sem nenhum texto)
Button 1	Name	BtnMensagem1
	Text	Exibir Mensagem 1
Button 2	Name	BtnMensagem2
	Enabled	False (esta opção deixará o botão "desabilitado" inicialmente)
	Text	Exibir Mensagem 2
Button 3	Name	BtnSair
	Text	Sair

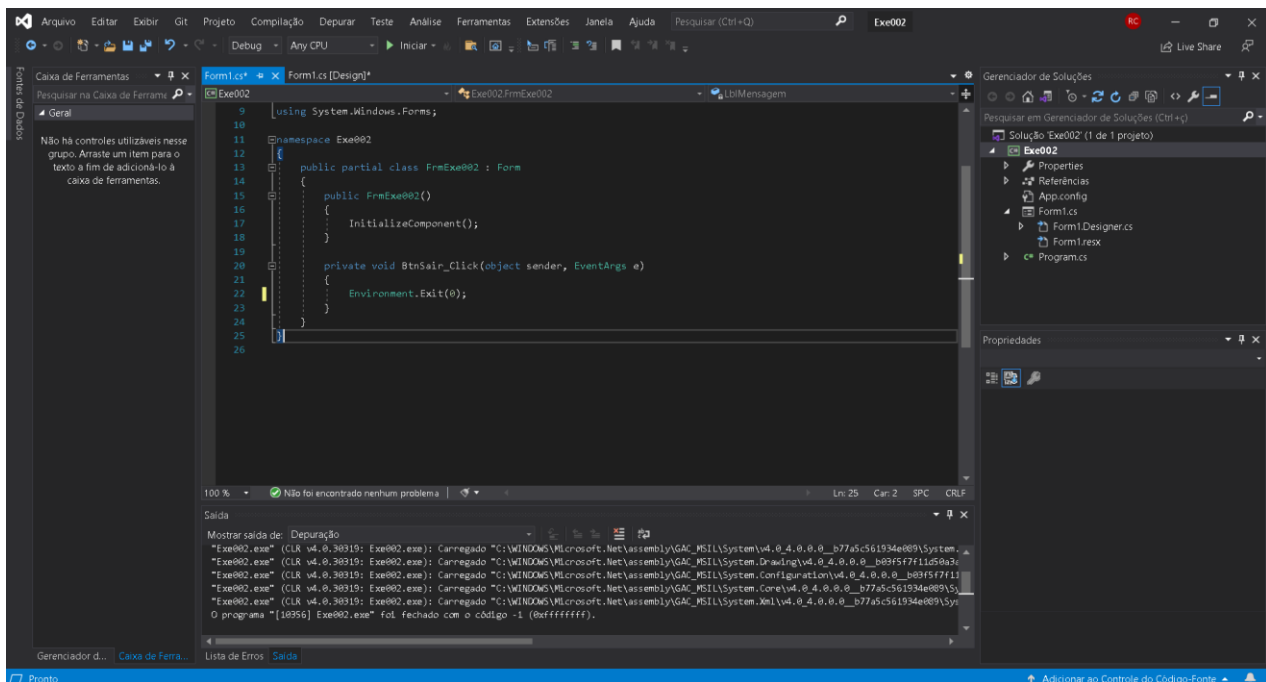
Execute o projeto para "visualizar" o resultado atual.. deve estar "parecido" com a imagem abaixo:



Vamos programar os "eventos". Podemos iniciar pela programação do "botão sair". Dê dois cliques sobre ele para entrarmos no modo de "programação":

Vamos digitar a instrução: `Application.Exit();`

Você deverá estar visualizando:



Se desejar, poderá testar o programa para verificar o que irá acontecer quando “clique” no botão Sair.

Vamos retornar ao modo de “Design” (clique no menu, opção “Exibir → Design”).

Vamos programar o botão “Mensagem 1”.

Este botão, em tempo de execução, quando for “clicado”, deverá executar três instruções:

1. A primeira instrução é para “exibir” no label “LblMensagem” o texto abaixo:
“Se você não faz o que gosta... Então clique no botão Mensagem 2...”.
2. A segunda instrução é para “habilitar” o botão “Mensagem 2”, pois está “travado”.
3. A terceira instrução irá “desabilitar” o botão “Mensagem 1”.

Pois então, vamos programar o botão “Mensagem 1” para que isto seja possível....

Dois “cliques” no botão “Mensagem 1” e vamos digitar as instruções abaixo, que DEVERÃO ser escritas entre as linhas do cabeçalho “private void....”, conforme modelo abaixo:

```
private void BtnMensagem1_Click(object sender, EventArgs e)
{
    LblMensagem.Text = "Se você não faz o que gosta.... Clique no botão Mensagem 2...";
    BtnMensagem2.Enabled = true;
    BtnMensagem1.Enabled = false;
}
```

ATENÇÃO:

1. O texto que está em vermelho não deverá ser digitado, pois o próprio Visual Studio inseriu este “cabeçalho” no momento em que clicamos duas vezes no botão.
2. Muito cuidado na digitação para não gerar erros de “sintaxe” (escrita).
 - a. Observe as “aspas” na primeira linha de digitação e
 - b. O ponto e vírgula (;) no final das linhas.

Prof. Roberto de Castro

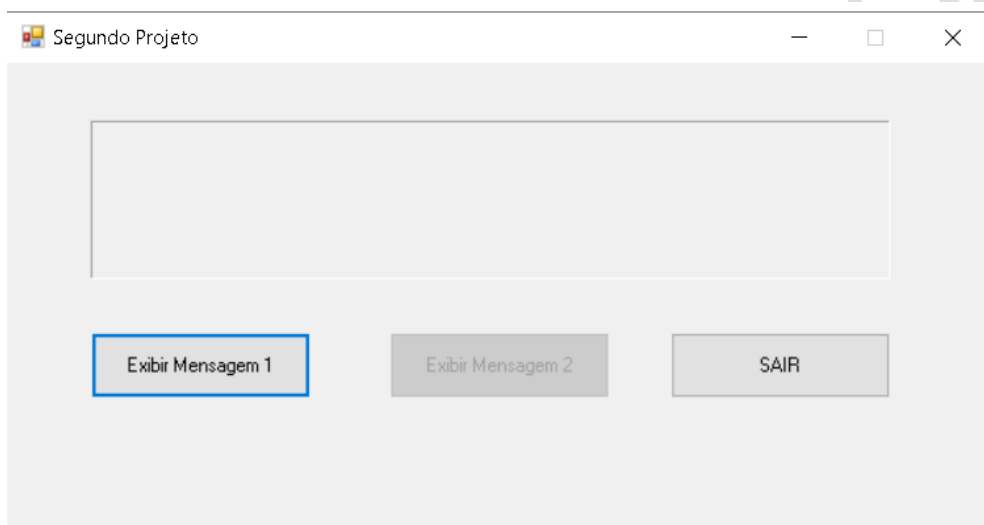
3. A primeira instrução está atribuindo para a propriedade "Text" do label "LblMensagem" a mensagem: Se você não faz o que gosta... Clique no botão Mensagem 2...
4. A segunda instrução "destrava" (habilita) o botão Mensagem 2, que inicialmente estava travado (desabilitado).
5. A terceira instrução "trava" (desabilita) o botão Mensagem 1.

O projeto NÃO está pronto, mas testá-lo neste momento será importante para entender o seu funcionamento.

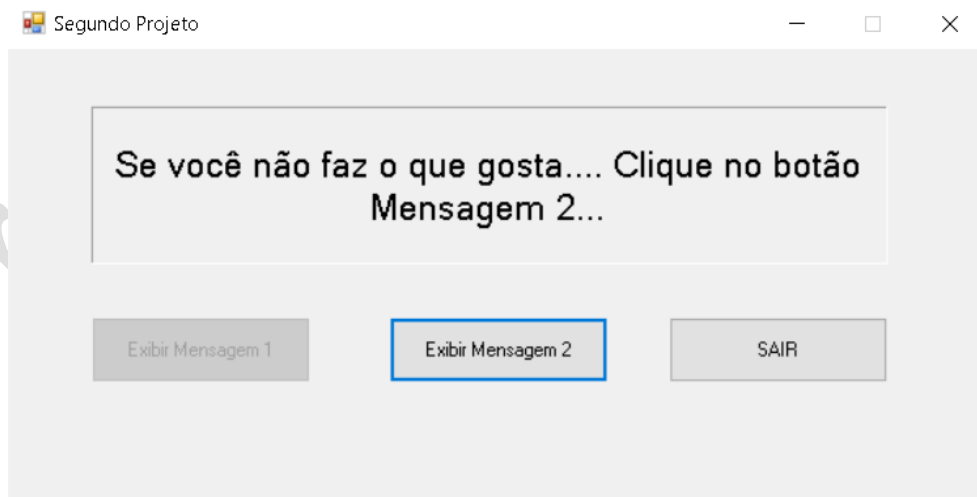
LEMBRE-SE: O botão "mensagem 2" está sem nenhuma funcionalidade no momento. Apenas os "botões": Mensagem 1 e Sair possuem alguma funcionalidade.

Quando executar o projeto e clicar no botão "Mensagem 1", o resultado deverá ser:

Tela inicial:



Ao clicar no botão "Mensagem 1":



Faça uma pausa. Compare os resultados... Analise o que fizemos até o momento. Estude as linhas de programação que inserimos nos botões. Isto será muito importante para podermos prosseguir, ok?

Finalize a execução do seu projeto (poderá clicar no botão Sair).

Prof. Roberto de Castro

Vamos agora programar o botão “Exibir Mensagem 2”.

Quando o projeto estiver em execução, este “botão” deverá ter as seguintes funcionalidades, quando for clicado:

1. A primeira instrução é para “exibir” no label “LblMensagem” o texto:
“Aprenda a gostar do que faz... Será melhor para você!! Volte agora para a Mensagem 1!!”.
2. A segunda instrução é para “habilitar” o botão “Mensagem 1”, pois está “travado”.
3. A terceira instrução irá “desabilitar” o botão “Mensagem 2”.

```
private void BtnMensagem2_Click(object sender, EventArgs e)
{
    LblMensagem.Text = "Aprenda a gostar do que faz... Será melhor para você!! Volte agora
                        para a Mensagem 1!!";
    BtnMensagem1.Enabled = true;
    BtnMensagem2.Enabled = false;
}
```

Execute o projeto novamente... Perceba a “alternância” dos botões. Estude as instruções que foram digitadas. Estude as linhas de programação.

Lembrando que: A propriedade “text” de um controle (componente visual) faz referência ao texto que será exibido neste controle. Os botões alteraram o conteúdo do Label em tempo de execução. O mesmo ocorreu com a propriedade “enabled” dos botões. Inicialmente o botão “mensagem 2” estava desabilitado (travado) e através da programação conseguimos alterar o seu estado para “habilitado”.

Neste momento não precisa se preocupar em “decorar” o que está escrito. O mais importante é aprender a interpretar o que o comando está fazendo.

Salve todo o projeto e feche a solução. Vamos nos preparar para o terceiro projeto!!

4.4 Terceiro Projeto

Já caminhamos bastante na construção de programas.

Conhecemos dois “controles” e suas finalidades: Label e Botão (além do próprio formulário).

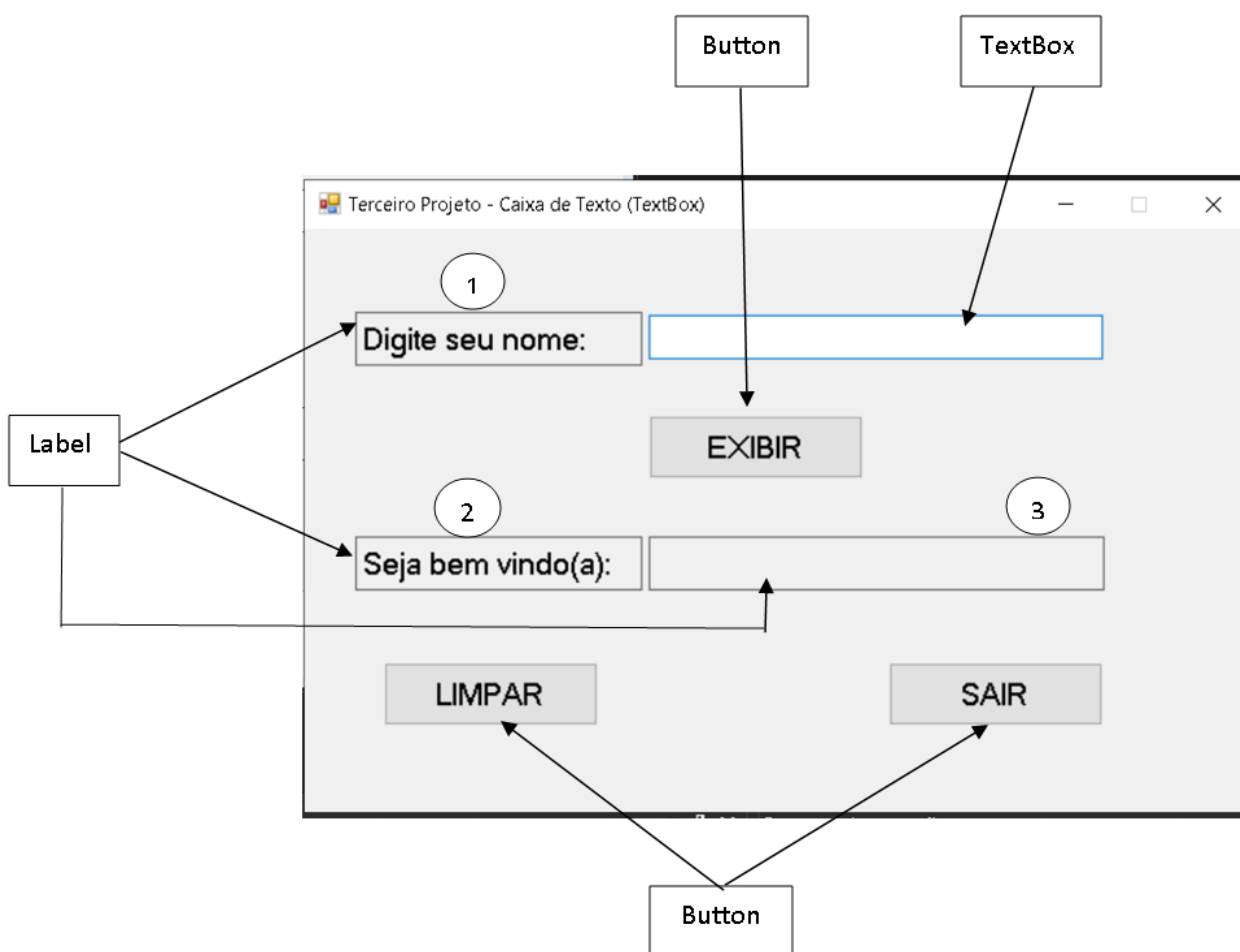
Estudamos também algumas propriedades (características visuais dos controles): Text; Enabled, Font, Name...

Agora vamos conhecer um novo controle (componente visual) que permitirá uma “interação” com o usuário. O controle “TextBox” (caixa de texto).

Novamente, para o desenvolvimento deste terceiro projeto seguiremos a mesma metodologia já estudada anteriormente. São basicamente 4 (quatro etapas):

- Primeira Etapa → Desenho da interface gráfica;
- Segunda Etapa → Configuração das propriedades;
- Terceira Etapa → Programação dos “eventos”.
- Quarta etapa → Testes

Seguindo a metodologia acima, nosso segundo projeto terá como “design” final, o modelo abaixo:



QUADRO DE PROPRIEDADES

CONTROLE	PROPRIEDADE	CONTEÚDO
Form	Name	FrmExe003
	MaximizeBox	False
	StartPosition	CenterScreen
	Text	Terceiro Projeto – Caixa de Texto (TextBox)
Label 1	Name	Label1 (iremos manter o nome original)
	AutoSize	False
	BorderStyle	Fixed3D
	Font	16
	Text	Digite seu nome:
TextBox	Name	TxtNome
	TabIndex	0
	Text	(deixar em branco – sem nenhum texto digitado)
Button 1	Name	BtnExibir
	TabIndex	1

Prof. Roberto de Castro

	Text	&EXIBIR
Label 2	Name	Label2 (iremos manter o nome original)
	AutoSize	False
	BorderStyle	Fixed3D
	Font	16
	Text	Seja bem vindo(a):
Label 3	Name	LblResposta
	AutoSize	False
	BorderStyle	Fixed3D
	Font	16
	Text	(deixar em branco, sem nenhum texto digitado)
Button 2	Name	BtnLimpar
	Enabled	False
	TabIndex	2
	Text	&LIMPAR
Button 3	Name	BtnSair
	TabIndex	3
	Text	&Sair

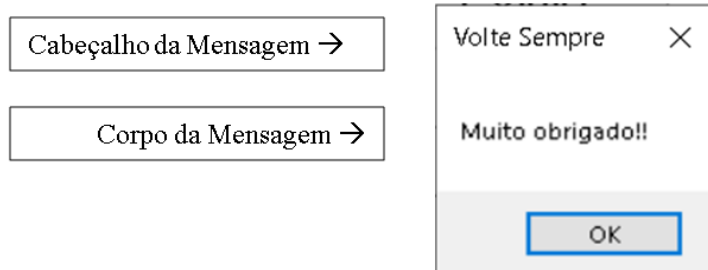
Importante: Temos neste quadro de propriedades, algumas “novidades”, entre elas:

1. Propriedade TABINDEX (na caixa de texto e nos “botões”) → é a programação das “paradas” da tecla “TAB” quando o projeto estiver em execução. A primeira parada do TAB é na caixa de texto, a segunda parada na tecla TAB é no botão EXIBIR e assim sucessivamente...
2. Para visualizar a ordem de parada da tecla TAB, clique no menu View (Exibir) → TabOrder
3. A inclusão do símbolo & na frente das propriedades TEXT dos botões. Este símbolo, quando inserido, indicará uma “tecla de atalho”. Por exemplo: o usuário poderá “clicar” no botão “EXIBIR” ou pressionar as teclas “alt E”.
4. Estas “novidades” representam um aprimoramento da nossa interface gráfica.

Vamos agora para a etapa de “programação”.

Vamos programar o botão “SAIR” (e incluir nele também uma novidade).

Em tempo de execução, quando o usuário clicar no botão SAIR (ou pressionar a tecla de atalho Alt S), o projeto exibirá uma caixa de mensagem (MessageBox) de despedida e somente depois será encerrado. A mensagem que será exibida segue abaixo:



Programação do botão SAIR:

```
private void BtnSair_Click(object sender, EventArgs e)
{
    MessageBox.Show("Muito obrigado!!", "Volte Sempre");
    Environment.Exit(0);
}
```

O comando "MessageBox.Show" exibirá uma caixa de texto (como visto logo acima). Está "dividido" em dois textos:

O primeiro "Muito obrigado", indica o "corpo da mensagem".

O segundo (separado por uma vírgula) "Volte Sempre", indica o "cabeçalho" da caixa de texto.

Execute o projeto e veja o resultado desta programação, até o momento...

Depois deste pequeno teste, para verificar o andamento do projeto, vamos programar o botão EXIBIR. Ele deverá:

1. Exibir no label "LblResposta" o nome que o usuário digitar;
2. Habilitar o botão limpar "BtnLimpar" e
3. Travar (desabilitar) o botão exibir "BtnExibir".

Pois então vamos inserir o código necessário para executar as tarefas descritas acima:

```
private void BtnExibir_Click(object sender, EventArgs e)
{
    LblResposta.Text = TxtNome.Text;
    BtnLimpar.Enabled = true;
    BtnExibir.Enabled = false;
}
```

Muita atenção na digitação dos comandos para evitar erros de escrita (sintaxe).

Observe o comando: LblResposta.Text = TxtNome.Text;

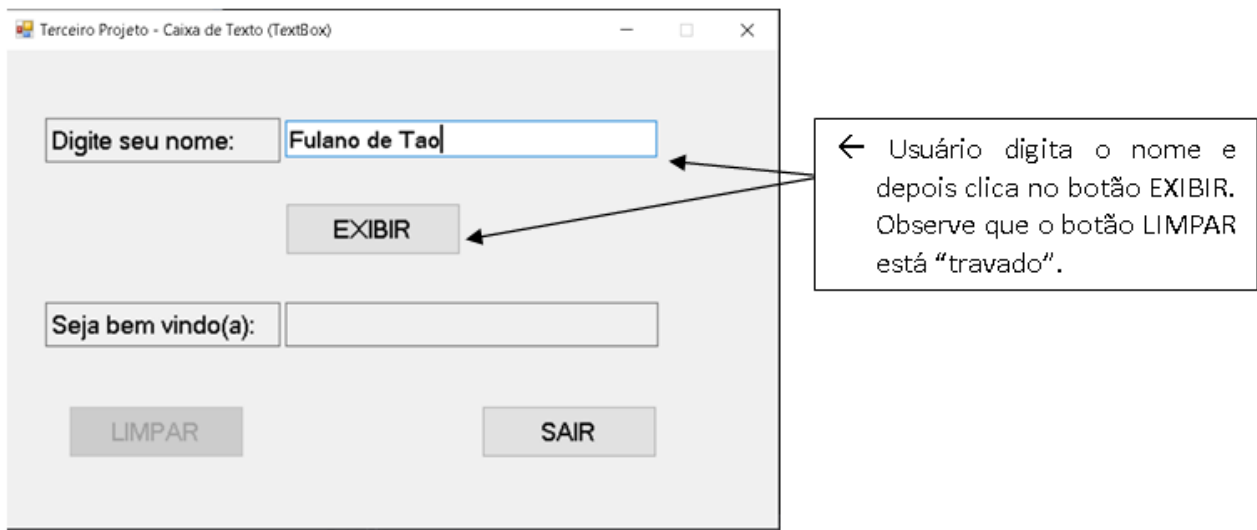
Está "instruindo" para atribuir o conteúdo do TxtNome.Text para o label "LblResposta".

Execute o projeto...

O formulário será exibido.

Digite o seu nome e clique no botão "Exibir".

Veja o resultado: O seu nome será exibido no "label". O botão "limpar" ficará "habilitado" e o botão "Exibir" ficará desabilitado.



Após clicar no botão “EXIBIR”, o resultado será:



Observe o resultado e compare com as instruções que você inseriu no botão “EXIBIR”.. Procure entender cada linha de programação. Não precisa decorar o comando, mas “entender”, ok?

Vamos agora programar o botão LIMPAR.

Ao ser “clicado” pelo usuário, o botão LIMPAR deverá:

1. Apagar o texto que está na caixa de Texto “TxtNome”;
2. Apagar o texto que está no label “LblResposta”;
3. Habilitar o botão “EXIBIR”;
4. Desabilitar o botão “LIMPAR” e
5. Voltar o “foco” (cursor) para a caixa de texto “TxtNome”.

Veja a programação como ficará:

```
private void BtnLimpar_Click(object sender, EventArgs e)
{
    TxtNome.Text = "";
    LblResposta.Text = "";
    BtnExibir.Enabled = true;
    BtnLimpar.Enabled = false;
    TxtNome.Focus();
}
```

Execute o projeto novamente. Digite o nome. Clique no botão “Exibir”. Veja o resultado e compare com a programação. Clique no botão “limpar” e veja o resultado.

Veja a programação completa, na listagem abaixo:

```
private void BtnSair_Click(object sender, EventArgs e)
{
    MessageBox.Show("Muito obrigado!!", "Volte Sempre");
    Environment.Exit(0);
}

private void BtnExibir_Click(object sender, EventArgs e)
{
    LblResposta.Text = TxtNome.Text;
    BtnLimpar.Enabled = true;
    BtnExibir.Enabled = false;
}

private void BtnLimpar_Click(object sender, EventArgs e)
{
    TxtNome.Text = "";
    LblResposta.Text = "";
    BtnExibir.Enabled = true;
    BtnLimpar.Enabled = false;
    TxtNome.Focus();
}
```

Desta forma finalizamos a programação deste projeto, porém, ele precisará ser “melhorado” (receber novas programações).

Veja o que ocorre ao executar o projeto e clicar no botão EXIBIR, sem digitar nenhum nome.

Nenhuma mensagem é exibida no label, o botão EXIBIR ficou travado e habilitou o botão LIMPAR. Precisaremos incluir, em outro momento, uma CONDIÇÃO ESPECIAL para “forçar” a digitação do nome, mas isto é um procedimento que será estudado um pouco mais adiante. Guarde este projeto pois retornaremos nele para esta “manutenção”.