



Componentes Windows Form

Programação III

Prof. Edson Mota, PhD, MSc, PMP

Existe uma infinidade de
Componentes

Componentes

- Estes componentes vão desde recursos para descrição de campos até a realização de seleções e listagens de grandes volumes de dados.
- Em geral, em uma aplicação desktop, lançamos mão de uma série de recursos de componentes, a fim de maximizarmos a experiência do usuário.
- Além disso, podemos criar novos componentes mais adequados à nossa necessidade.



Alguns Componentes

- Em geral, os componentes possuem dois recursos principais que podemos utilizar, são eles:
- As Propriedades (Características)
- Os Métodos (ou Eventos)



Windows Form

The image shows a Windows Forms application in Visual Studio. On the left is a window titled 'Form1' with a light blue background. On the right is the 'Propriedades' (Properties) window for 'Form1' of type 'System.Windows.Forms.Form'. A red line points from the 'Propriedades' label in a box to the Properties window.

Propriedades

Form1 System.Windows.Forms.Form

- Acessibilidade**
 - AccessibleDescription
 - AccessibleName
 - AccessibleRole: Default
- Aparência**
 - BackColor**: 192; 192; 255
 - BackgroundImage: (none)
 - BackgroundImageLayout: Tile
 - Cursor: Default
- Font**
 - Font: Segoe UI; 9pt
 - ForeColor: ControlText
 - FormBorderStyle: Sizable
 - RightToLeft: No
 - RightToLeftLayout: False
 - Text: **Form1**
 - UseWaitCursor: False
- Comportamento**
 - AllowDrop: False
 - AutoValidate: EnablePreventFocusChange
 - ContextMenuStrip: (none)
 - DoubleBuffered: False
 - Enabled: True
 - ImeMode: NoControl
- Dados**
 - (DataBindings): (ControlBindings)

Windows Form

Form1.cs [Projeto] [ícone] [X]

Form1

Propriedades

Form1 System.Windows.Forms.Form

QueryContinueDrag

Chave

KeyDown

KeyPress

KeyUp

PreviewKeyDown

Comportamento

ChangeUICues

ControlAdded

ControlRemoved

FormClosed

FormClosing

HelpButtonClicked

HelpRequested

ImeModeChanged

InputLanguageChanged

InputLanguageChanging

Load

Form1_Load_1

QueryAccessibilityHelp

Shown

StyleChanged

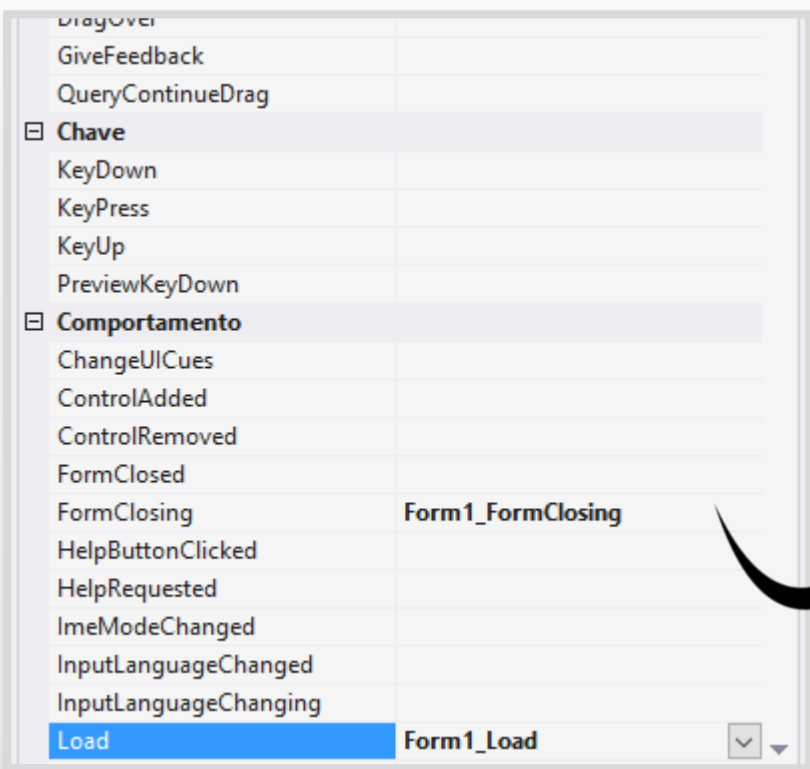
SystemColorsChanged

Dados

Posicionando o cursor em um evento + ENTER, você será levado à programação deste evento.

Eventos

Programando um Evento



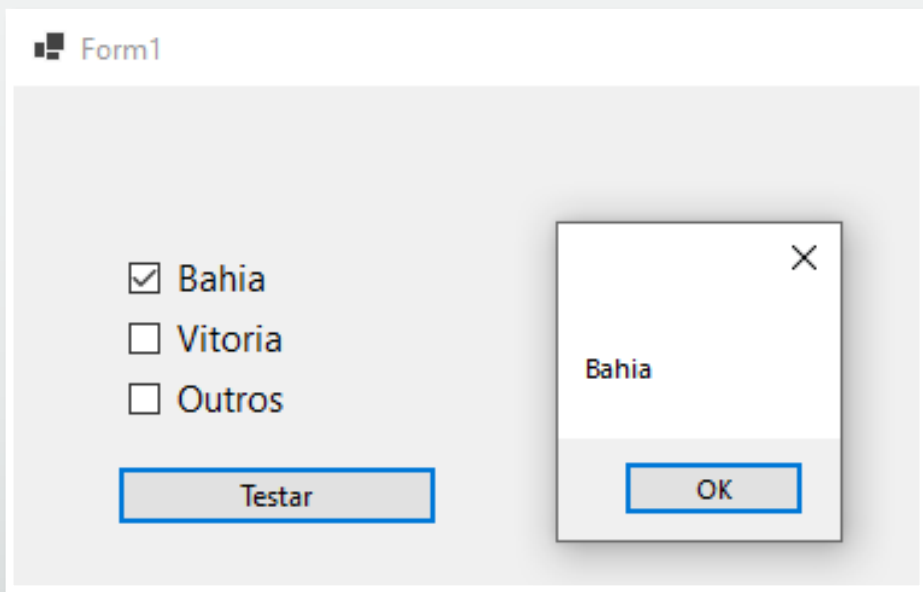
```
0 referências
private void Form1_Load_1(object sender, EventArgs e)
{
    ...
}

1 referência
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    MessageBox.Show("Encerrando Formulário!");
}
```

Alguns Componentes - CheckBox

Permite realizar a seleção de múltiplos elementos;

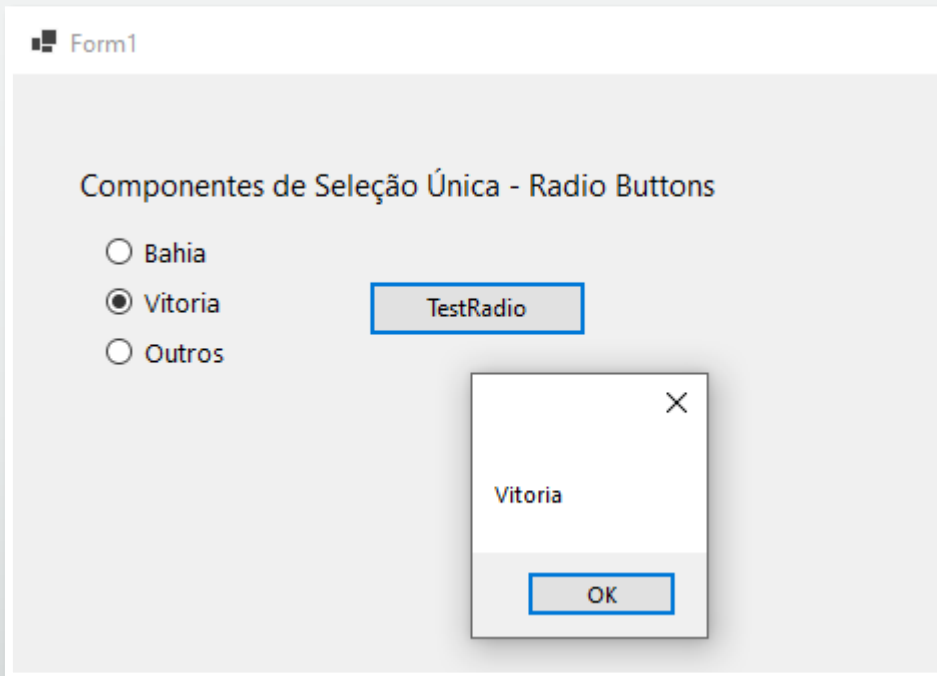
Seu código, em geral, requer a realização de um teste para validar quais das opções foram marcadas.



```
1 referência
private void button1_Click(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        MessageBox.Show(checkBox1.Text);
    }
}
```


RadioButton

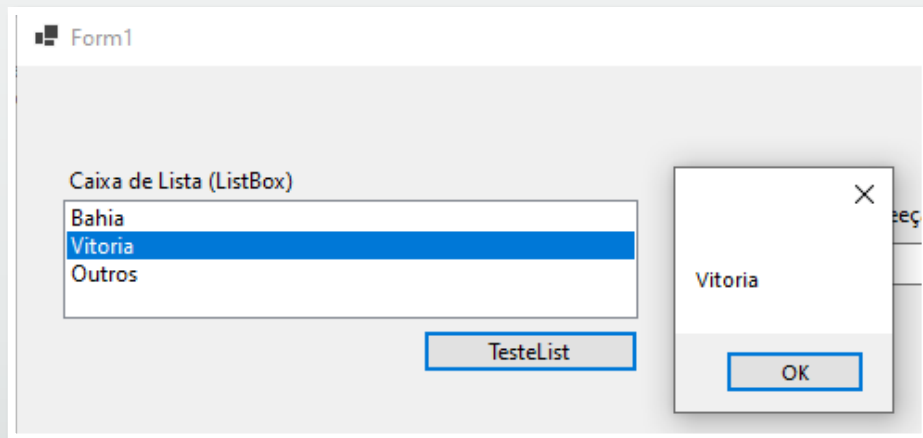
- Este componente permite seleção única de um item, garantindo que o usuário não poderá selecionar mais de uma opção.



```
1 referência
private void button4_Click(object sender, EventArgs e)
{
    if (radioButton2.Checked)
    {
        MessageBox.Show(radioButton2.Text);
    }
}
```

ListBox

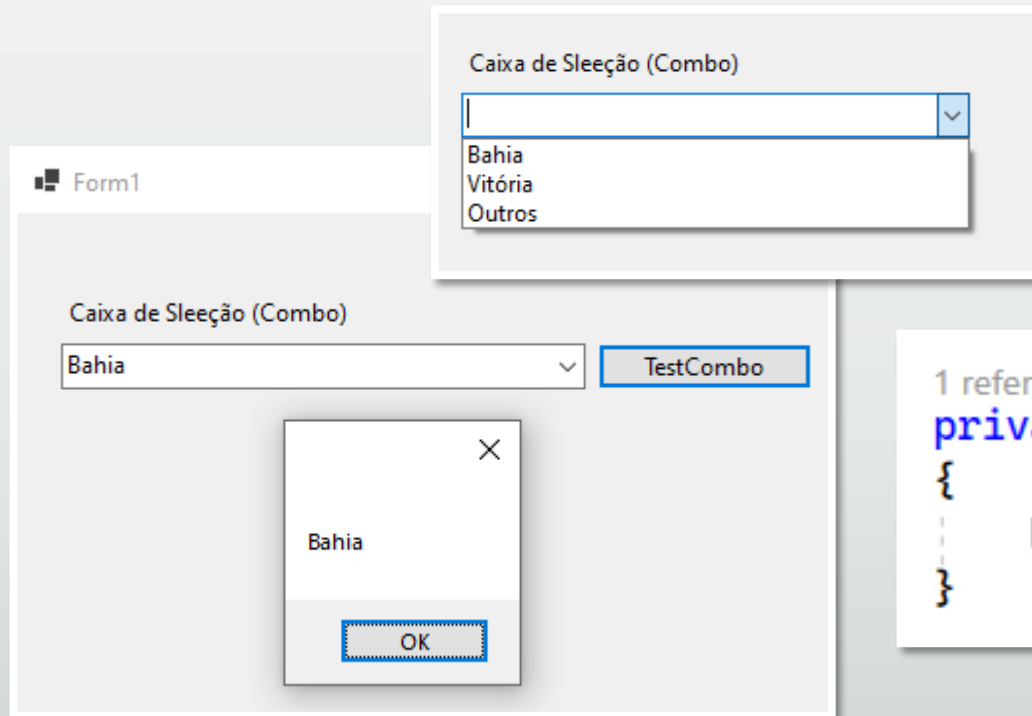
- Permite realizar a listagem de itens e interação com a lista por meio de métodos disponíveis para inclusão, alteração ou remoção de itens



```
1 referência
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show(listBox1.SelectedItem.ToString());
}
```

ComboBox

- A caixa de listagem permite a apresentação dos itens em formato de lista suspensa, essa estratégia melhora o aproveitamento do espaço em tela ajudando a acomodar melhor os demais componentes.



```
1 referência
private void button3_Click(object sender, EventArgs e)
{
    MessageBox.Show(comboBox1.Text);
}
```

GridView

O GridView é utilizado para apresentar grandes volumes de dados, em geral, carrega-se um dataset contendo os dados a partir de uma única propriedade, embora este componente ofereça muitos recursos de customização.

Se estivermos utilizando uma arquitetura baseada em objeto, por exemplo nossa abordagem multicamada, o processo torna-se muito simples.



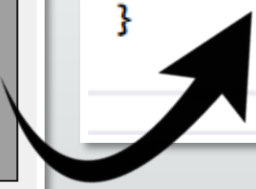
The screenshot shows a Windows application window. At the top left, there is a button labeled 'Carrega Lista'. Below it is a GridView control displaying a table with the following data:

	Id	Nome	Telefone	Email
▶	11	Marcos	r3245434	marcos@teste....

The 'Id' column has a blue selection bar over the value '11'. The first cell of the first row contains a right-pointing triangle icon.

1 referência

```
private void button4_Click(object sender, EventArgs e)
{
    List<Cliente> list = ClienteRepository.GetAll();
    dataGridView1.DataSource = list;
}
```



Você já ouviu falar em
UserControl?



UserControl

- Algumas vezes, precisamos desenvolver como unificar funcionalidades de certos componentes criando uma nova funcionalidade que poderá ser replicada em toda a aplicação.
- UserControl é um tipo de controle **personalizado** que você pode criar e usar em seu aplicativo.
- Um UserControl pode ser usado em um formulário do Windows Forms como qualquer outro controle.
- Algumas vantagens de sua utilização são:
 - Reutilização de código
 - Capacidade de personalização
 - Facilidade de manutenção

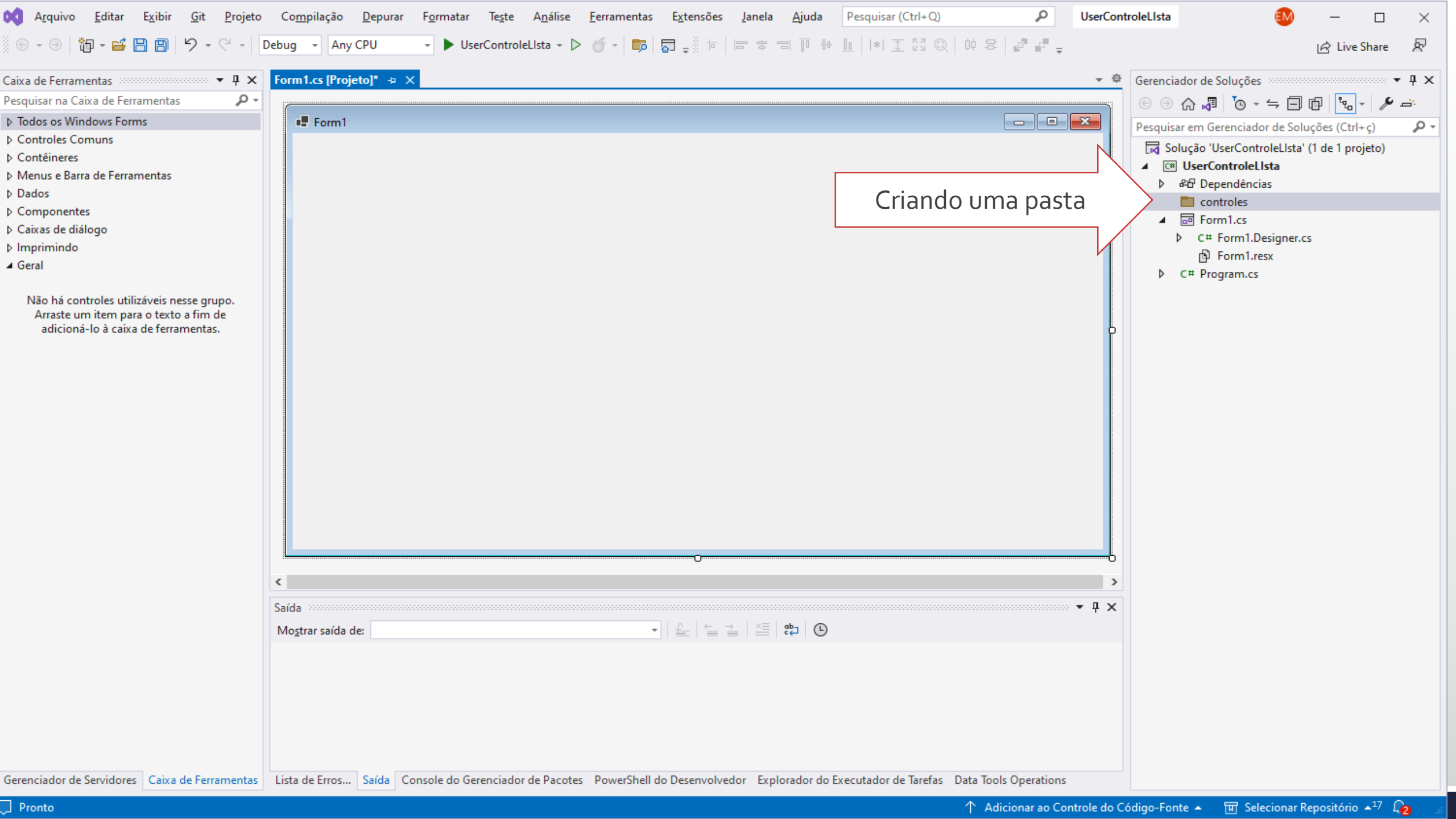


Cenário

- Imagine que precisamos implementar um controle que será utilizado em várias telas da sua aplicação;
- Esse controle consiste em uma caixa de texto e uma lista onde é possível adicionar estes itens;
- Podemos resolver isso de duas formas:
 1. Implementar em todas as telas
 2. Implementar um UserControl e usar em todas as telas



Vamos criar um
UserController?



Criando uma pasta

Arquivo

Editar

Exibir

Git

Projeto

Compilação

Depurar

Formatar

Teste

Análise

Ferramentas

Extensões

Janela

Ajuda

Pesquisar (Ctrl+Q)

UserControlLista

Debug

Any CPU

UserControlLista

Caixa de Ferramentas

Pesquisar na Caixa de Ferramentas

Todos os Windows Forms

Controles Comuns

Contêineres

Menus e Barra de Ferramentas

Dados

Componentes

Caixas de diálogo

Imprimindo

Geral

Não há controles utilizáveis nesse grupo.

Arraste um item para o texto a fim de adicioná-lo à caixa de ferramentas.

Form1.cs [Projeto]*

Adicionar Novo Item - UserControlLista

Instalados

Itens do C#

Código

Dados

Geral

Web

Windows Forms

SQL Server

Online

Classificar por: Padrão

	Classe	Itens do C#
	Interface	Itens do C#
	Formulário (Windows Forms)	Itens do C#
	Controle de Usuário (Windows Forms)	Itens do C#
	Classe de Componente	Itens do C#
	ADO.NET Entity Data Model	Itens do C#
	Arquivo Bitmap	Itens do C#
	Arquivo de Código	Itens do C#
	Arquivo de Configuração do Aplicativo	Itens do C#
	Arquivo de Configuração JSON do JavaScript	Itens do C#
	Arquivo de Configuração JSON do TypeScript	Itens do C#
	Arquivo de Configurações	Itens do C#
	Arquivo de Cursor	Itens do C#
	Arquivo de Ícone	Itens do C#

Pesquisar (Ctrl+E)

Tipo: Itens do C#

Um controle do Windows Forms (WinForms) reutilizável

Nome:

UserControl1.cs

Adicionar

Cancelar

Gerenciador de Soluções

ar em Gerenciador de Soluções (Ctrl+ç)

Solução 'UserControlLista' (1 de 1 projeto)

UserControlLista

Dependências

controles

Form1.cs

C# Form1.Designer.cs

Form1.resx

C# Program.cs

Gerenciador de Servidores

Caixa de Ferramentas

Lista de Erros...

Saída

Console do Gerenciador de Pacotes

PowerShell do Desenvolvedor

Explorador do Executador de Tarefas

Data Tools Operations

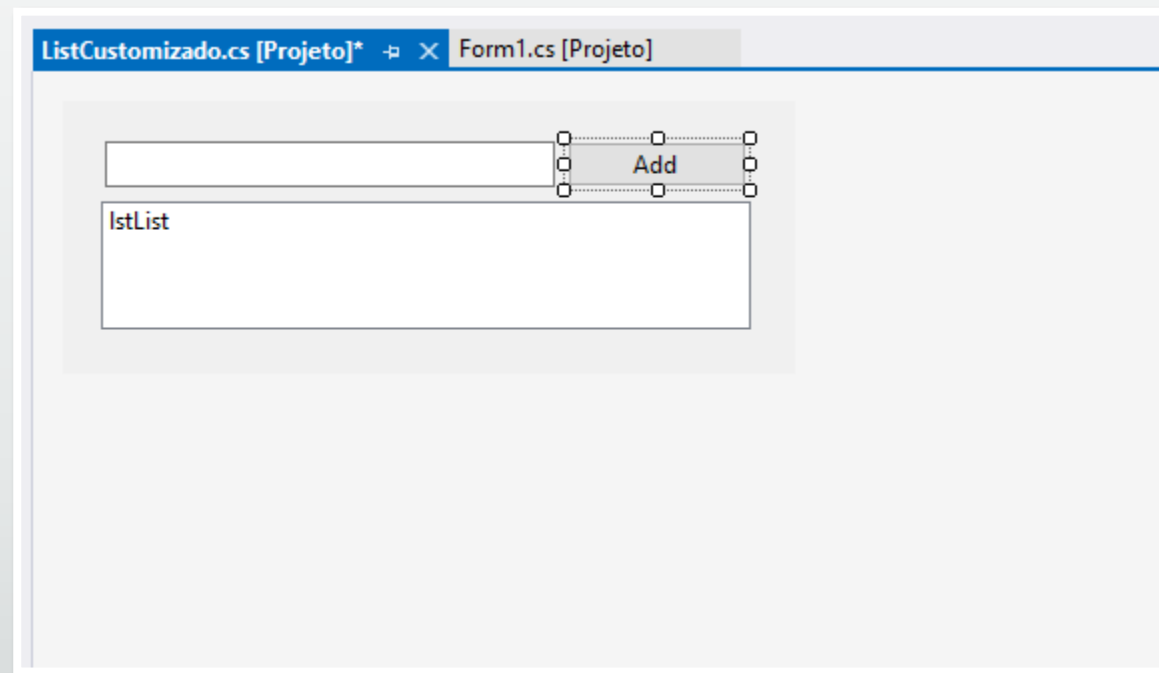
Pronto

Adicionar ao Controle do Código-Fonte

Selecionar Repositório

Construindo o Layout

- Nesse layout, quando o usuário escrever algo e clicar no botão add, o item será lançado na lista



Método Adicionar

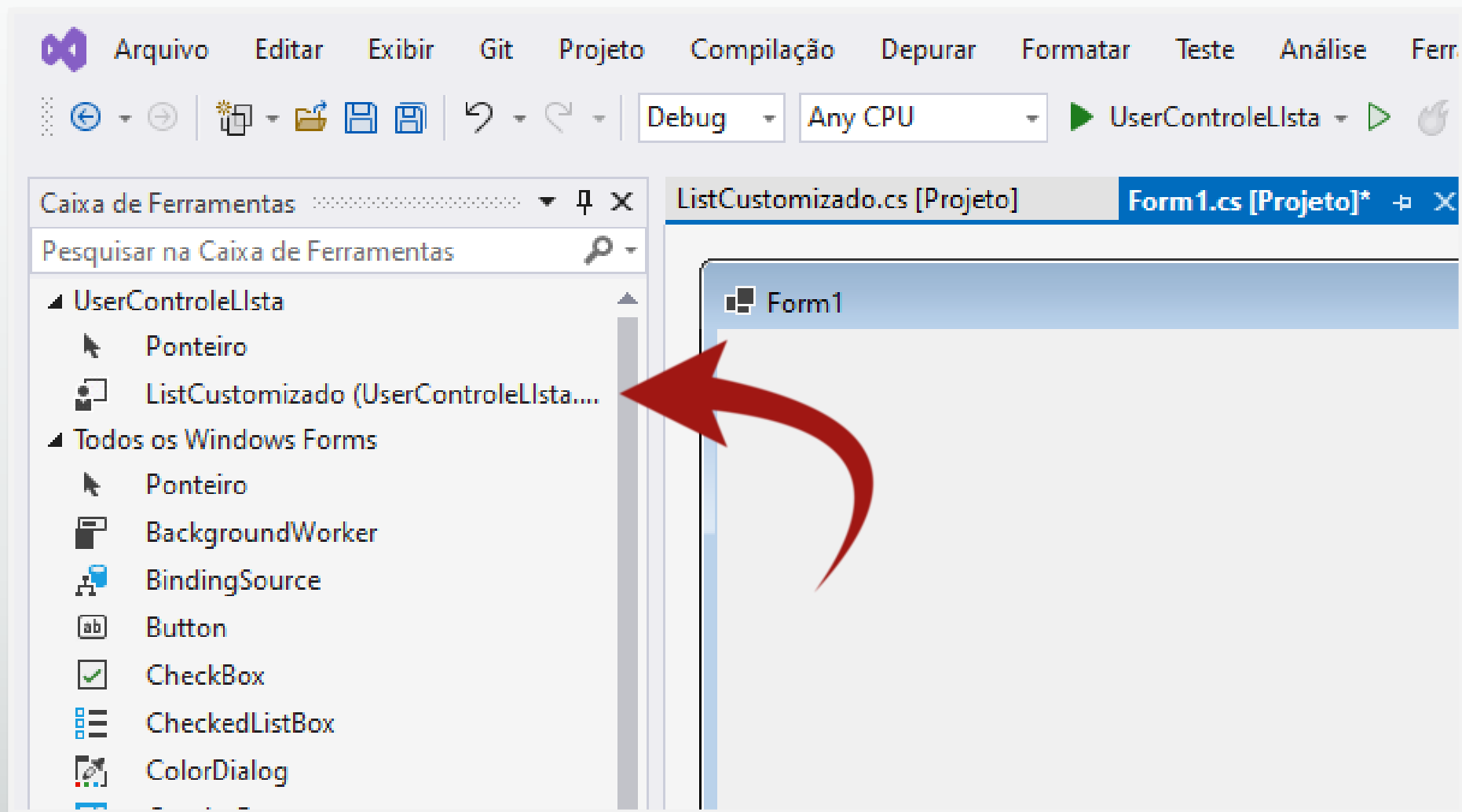
- No evento click do button, podemos programar a funcionalidade que permitirá o cadastramento do conteúdo da caixa de texto, como um item da lista

```
ListCustomizado.cs [Projeto] Form1.cs [Projeto]
C# UserControlLista UserControlLista.controles.ListCustomizado txtText

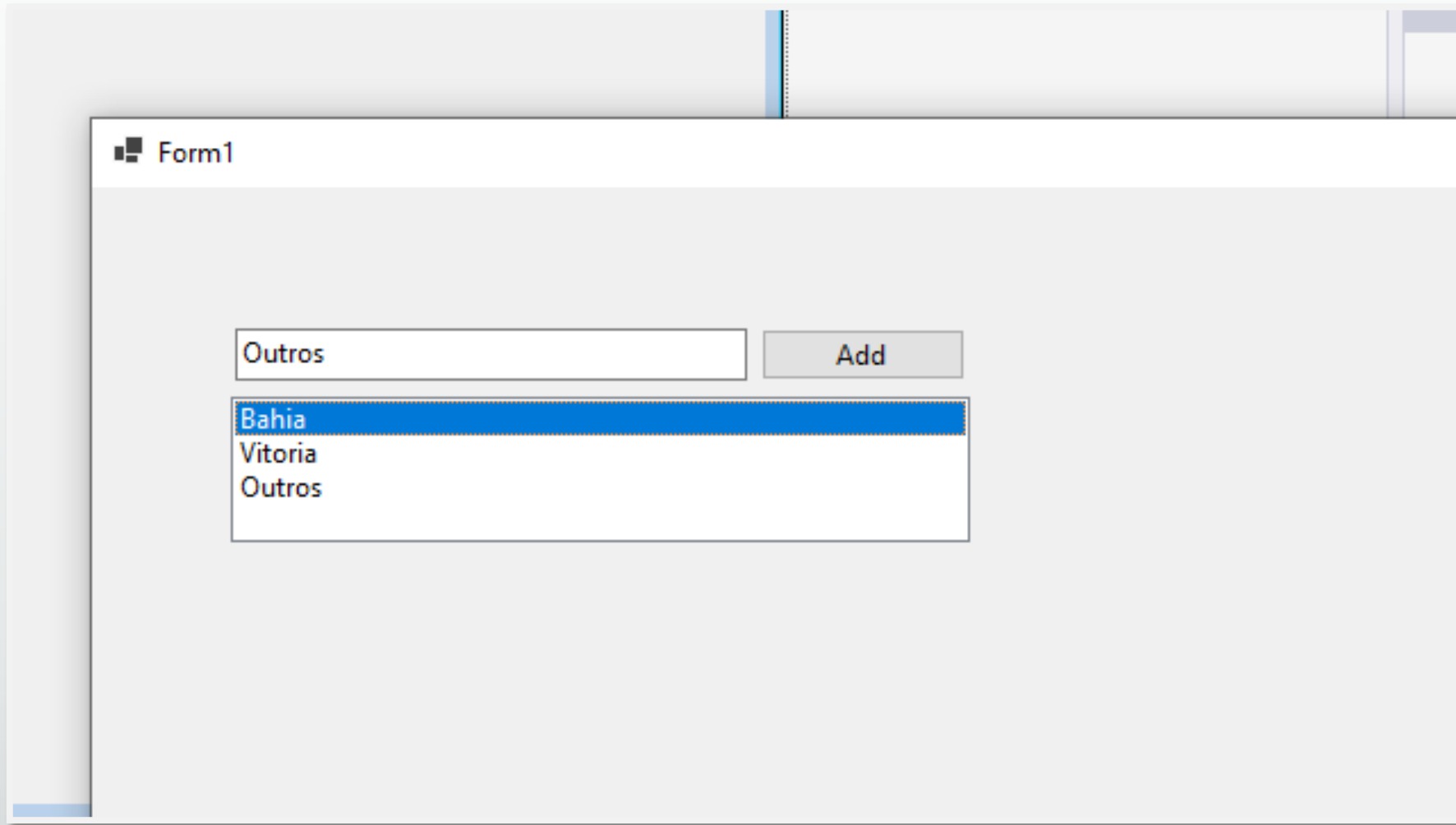
11 namespace UserControlLista.controles
12 {
13     2 referências
14     public partial class ListCustomizado : UserControl
15     {
16         0 referências
17         public ListCustomizado()
18         {
19             InitializeComponent();
20         }
21     }
22     1 referência
23     private void btAdd_Click(object sender, EventArgs e)
24     {
25         lstList.Items.Add(txtText.Text);
26     }
27 }
```

Onde está o nosso novo controle?

- Pronto, agora basta arrastar o nosso novo controle e usar em todos os formulários onde ele for necessário.



Usando o Novo Controle



The image shows a screenshot of a Windows application window titled "Form1". Inside the window, there is a text input field containing the text "Outros". To the right of the input field is a button labeled "Add". Below the input field is a list box containing three items: "Bahia", "Vitoria", and "Outros". The item "Bahia" is currently selected and highlighted with a blue background.

Revisão

- Elabore um UserControl no qual seja possível selecionar um item e mostrar imagem correspondente.
- A imagem pode ser inserida no componente com o seguinte trecho de código:

```
Image minhaImagem = Image.FromFile("D:\\IMAGEM.JPG");  
pictureBox1.Image= minhaImagem;
```

Bons Estudos!