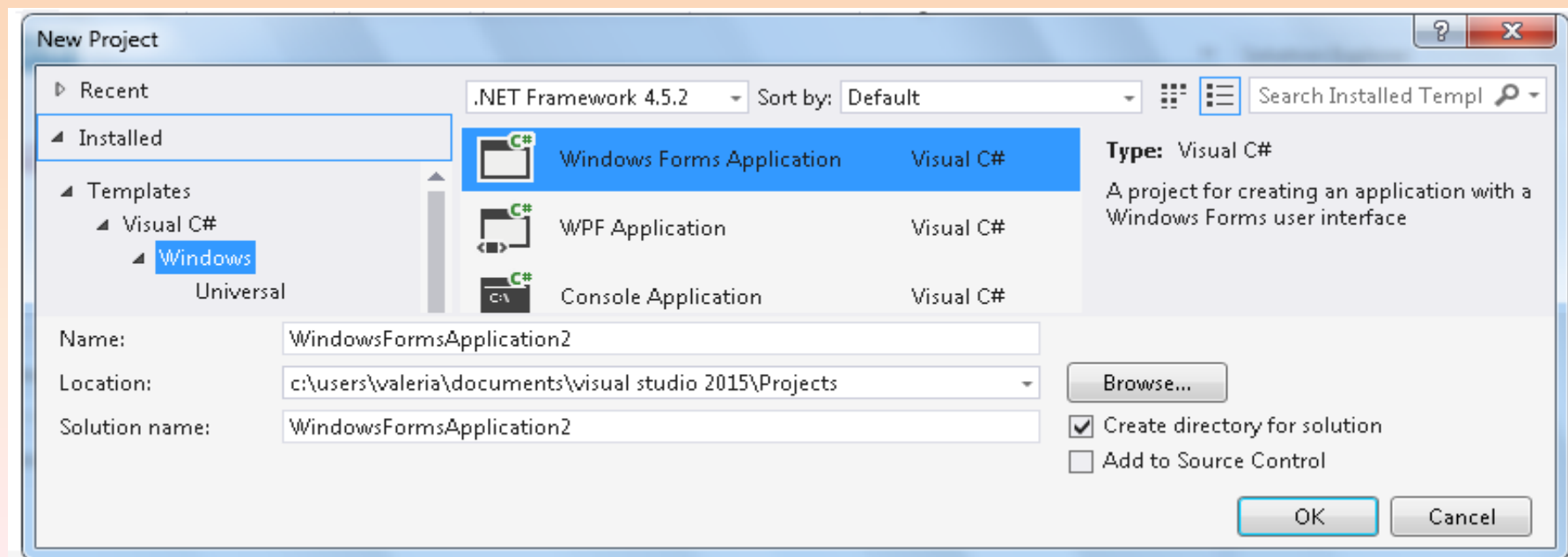
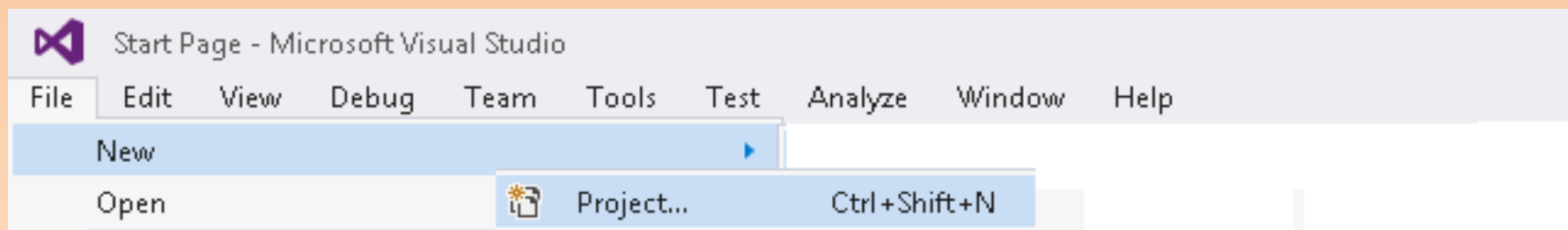


Programação Orientado a Objetos II

Prof. Rogério B. de Andrade

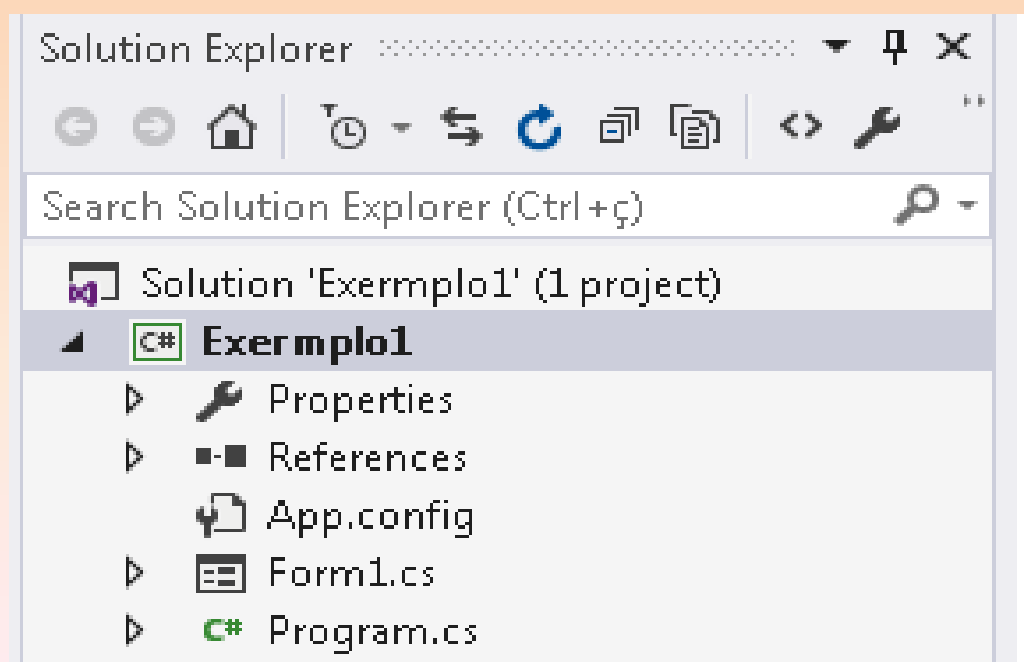
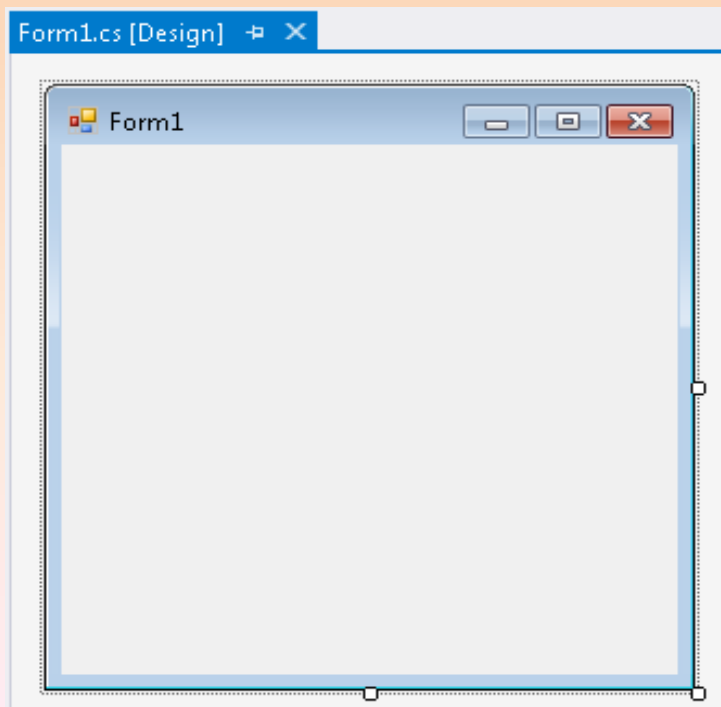
Bacharel em Ciência da Computação, Especialização em Educação, Licenciatura em Informática

Desenvolvimento UI com WinForms

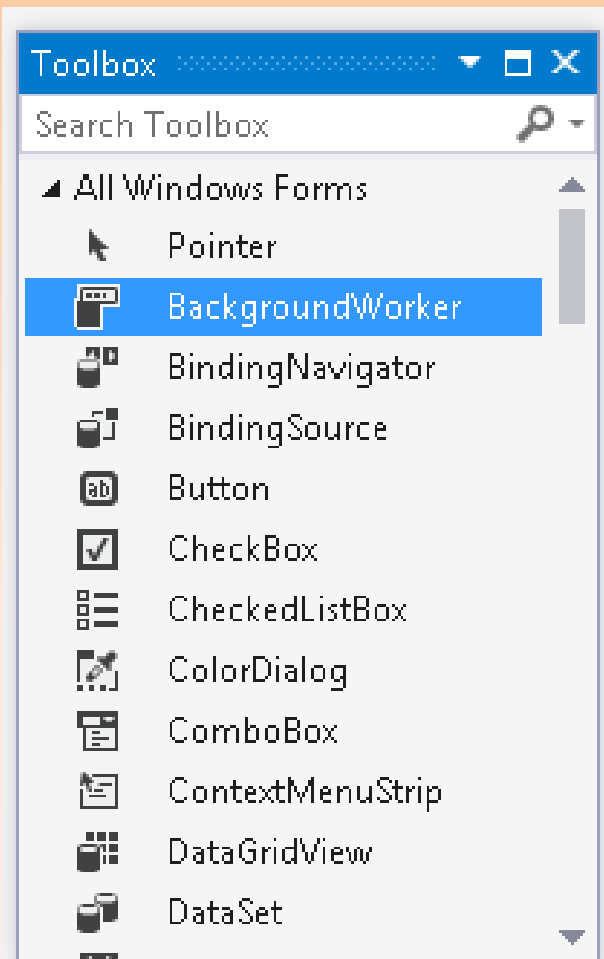
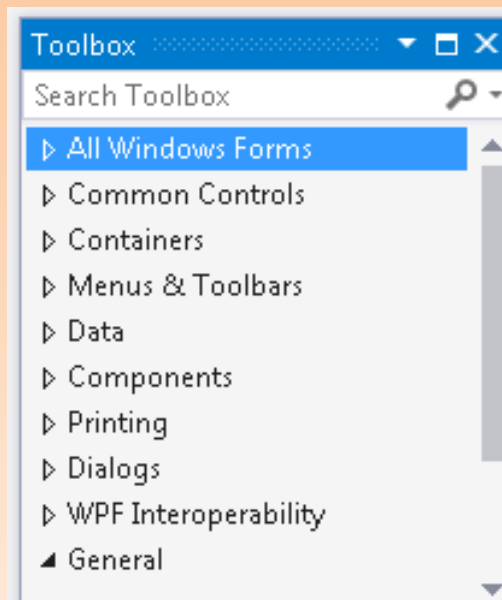
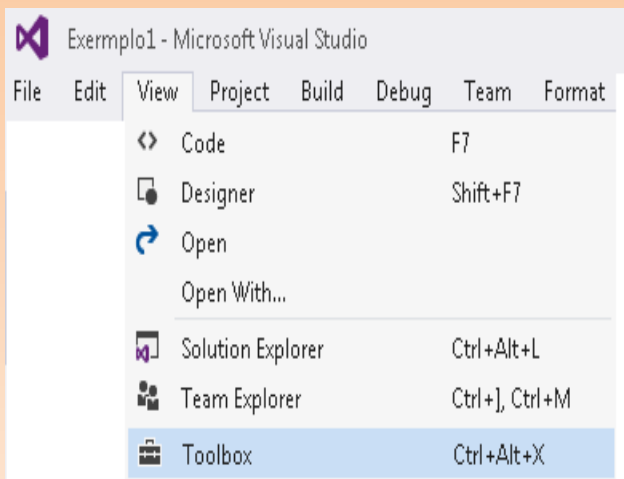


Desenvolvimento UI com WinForms

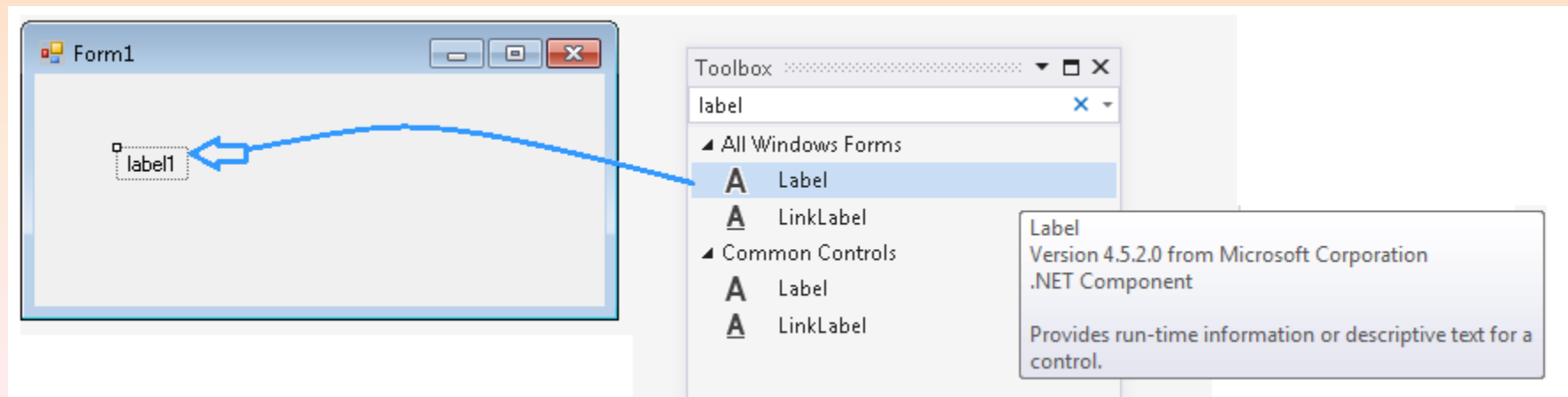
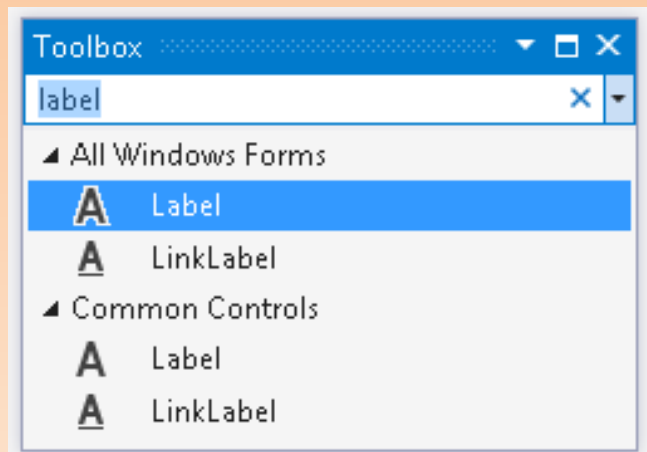
Name:	Exermplo1
Location:	c:\users\valeria\documents\visual studio 2015\Projects
Solution name:	Exermplo1



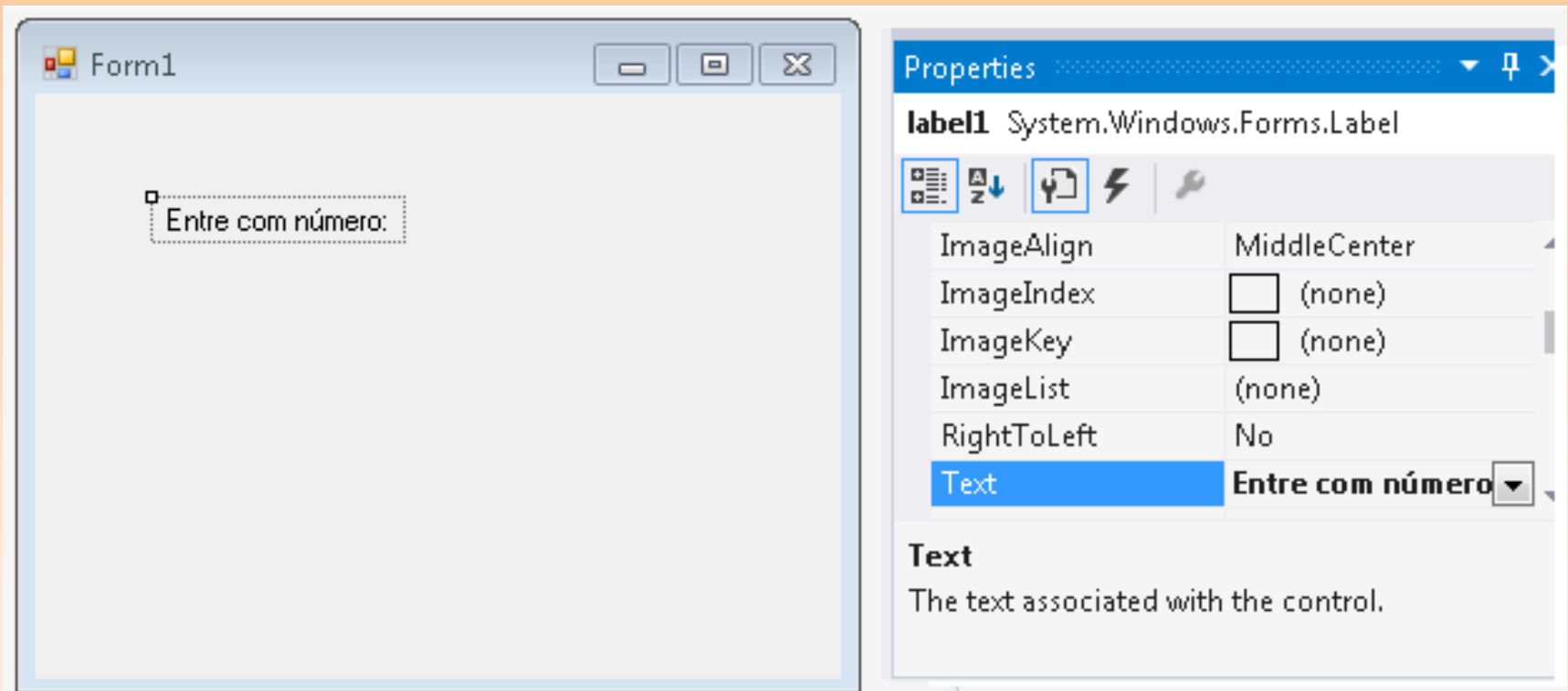
Desenvolvimento UI com WinForms



Desenvolvimento UI com WinForms



Desenvolvimento UI com WinForms



The screenshot displays a WinForms application window titled "Form1". Inside the form, there is a label with the text "Entre com número:". To the right of the form, the "Properties" window is open, showing the properties for the selected control, "label1" (System.Windows.Forms.Label). The "Text" property is highlighted, and its value is "Entre com número".

Properties

label1 System.Windows.Forms.Label

ImageAlign: MiddleCenter

ImageIndex: ☐ (none)

ImageKey: ☐ (none)

ImageList: (none)

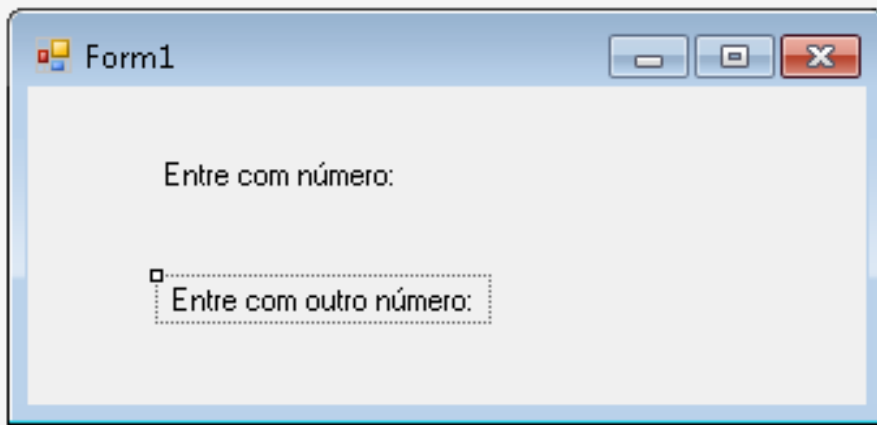
RightToLeft: No

Text Entre com número

Text

The text associated with the control.

Desenvolvimento UI com WinForms

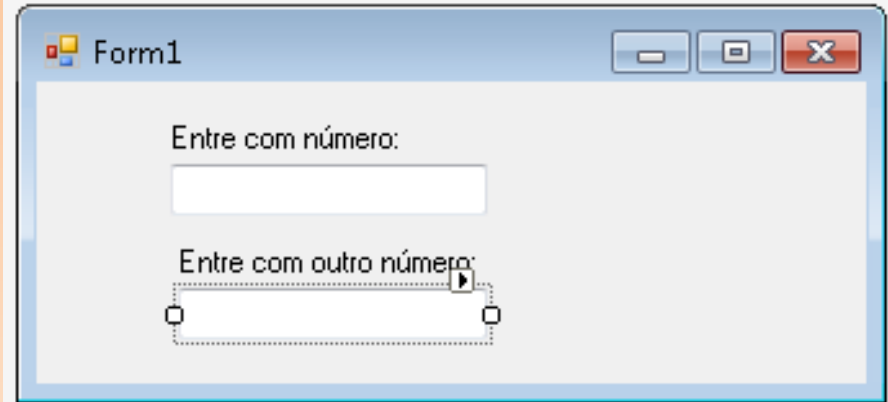


Form1

Entre com número:

Entre com outro número:

This screenshot shows a Windows Forms application window titled 'Form1'. It contains two labels: 'Entre com número:' and 'Entre com outro número:'. The second label is currently selected, indicated by a dashed border and a small square handle in the top-left corner.

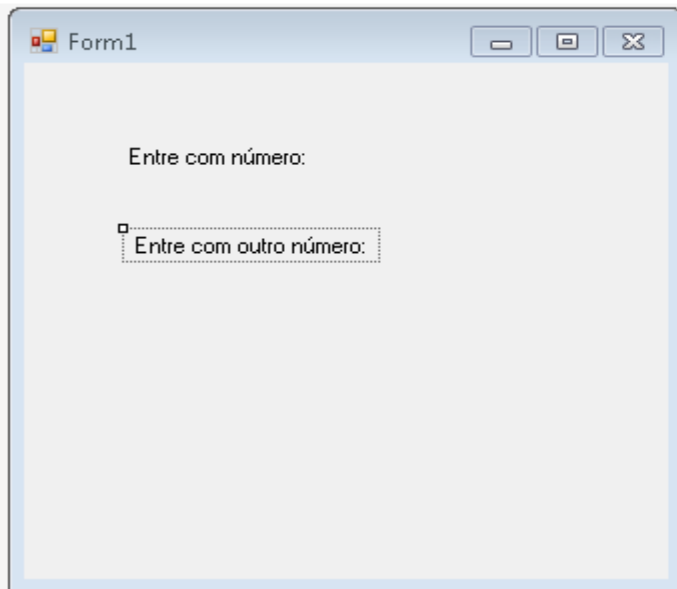


Form1

Entre com número:

Entre com outro número:

This screenshot shows the same 'Form1' window, but now with two text boxes added. The first text box is positioned below the first label, and the second text box is below the second label. The second text box is currently selected, indicated by a dashed border and a small square handle in the bottom-right corner.

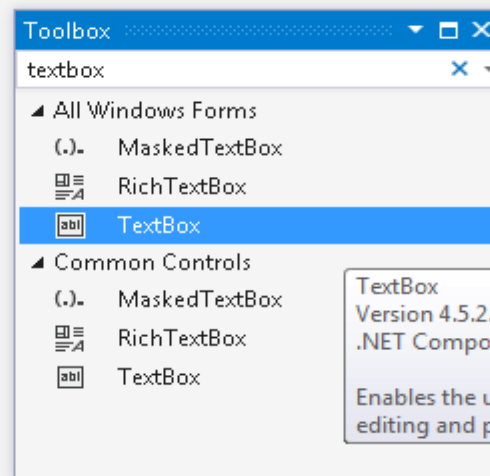


Form1

Entre com número:

Entre com outro número:

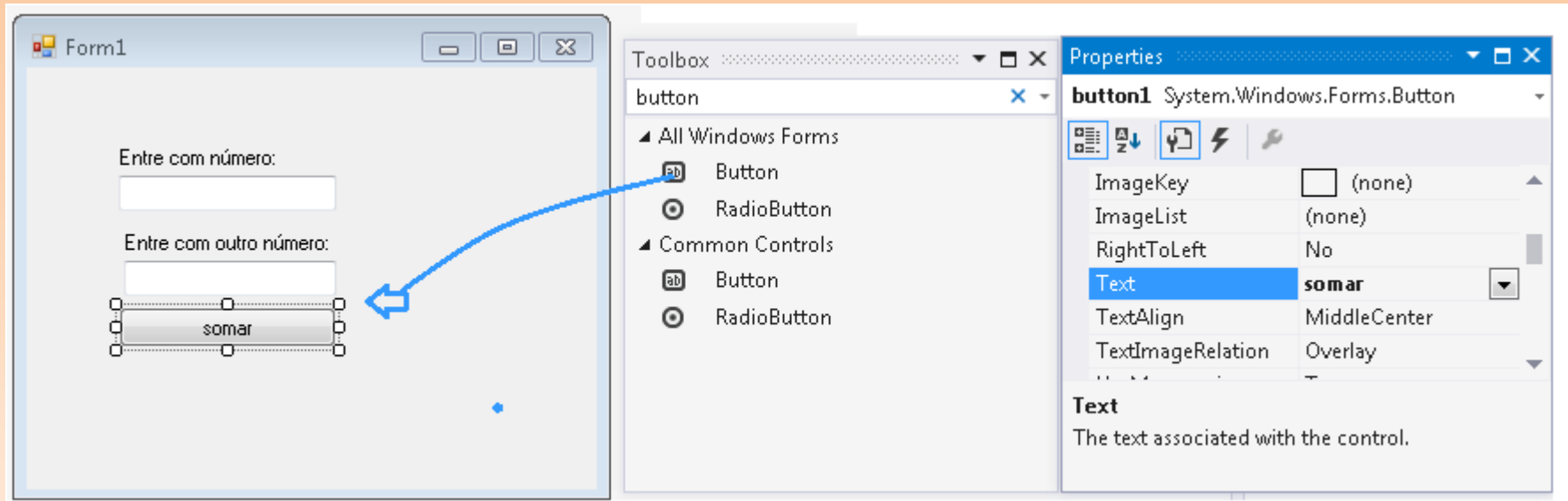
This screenshot shows the 'Form1' window with the two text boxes. The second text box is still selected, indicated by a dashed border and a small square handle in the bottom-right corner.



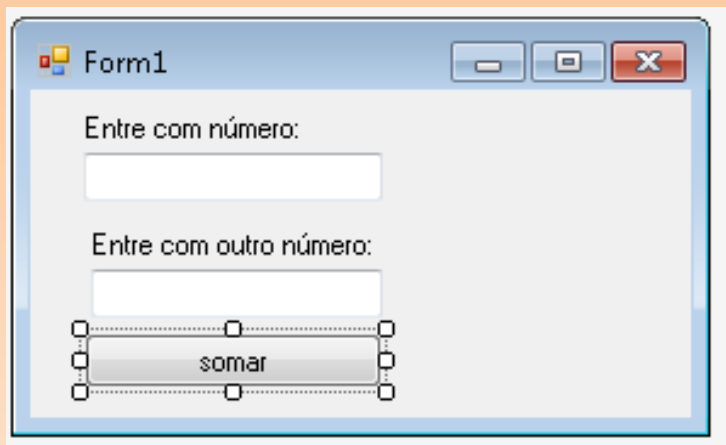
TextBox
Version 4.5.2.0 from Microsoft Corporation
.NET Component

Enables the user to enter text, and provides multiline editing and password character masking.

Desenvolvimento UI com WinForms



Desenvolvimento UI com WinForms



```
20 private void button1_Click(object sender, EventArgs e)
21 {
22
23 }
```

```
private void button1_Click(object sender, EventArgs e)
{
    double n1=0.0, n2 = 0.0;
}
```

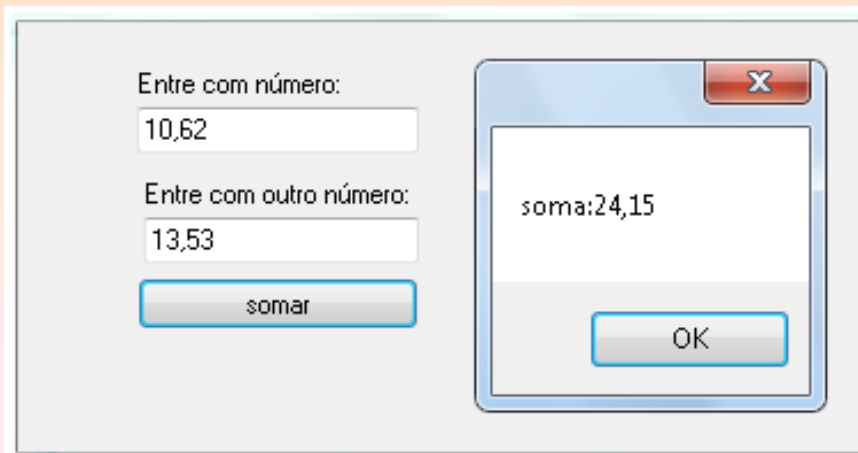
(local variable) `double n1`

The variable 'n1' is assigned but its value is never used

[Show potential fixes \(Ctrl+.\)](#)

Desenvolvimento UI com WinForms

```
20 private void button1_Click(object sender, EventArgs e)
21 {
22     double n1=0.0, n2 = 0.0, s=0.0;//declara variáveis
23     // converte texto contindo em textBox1 para double
24     n1 = Convert.ToDouble(textBox1.Text);
25     // converte texto contindo em textBox2 para double
26     n2 = Convert.ToDouble(textBox2.Text);
27     // calcula a soma
28     s = n1 + n2;
29     // mostra resultado
30     MessageBox.Show("soma:" + s);
31 }
```



Entre com número:
10,62

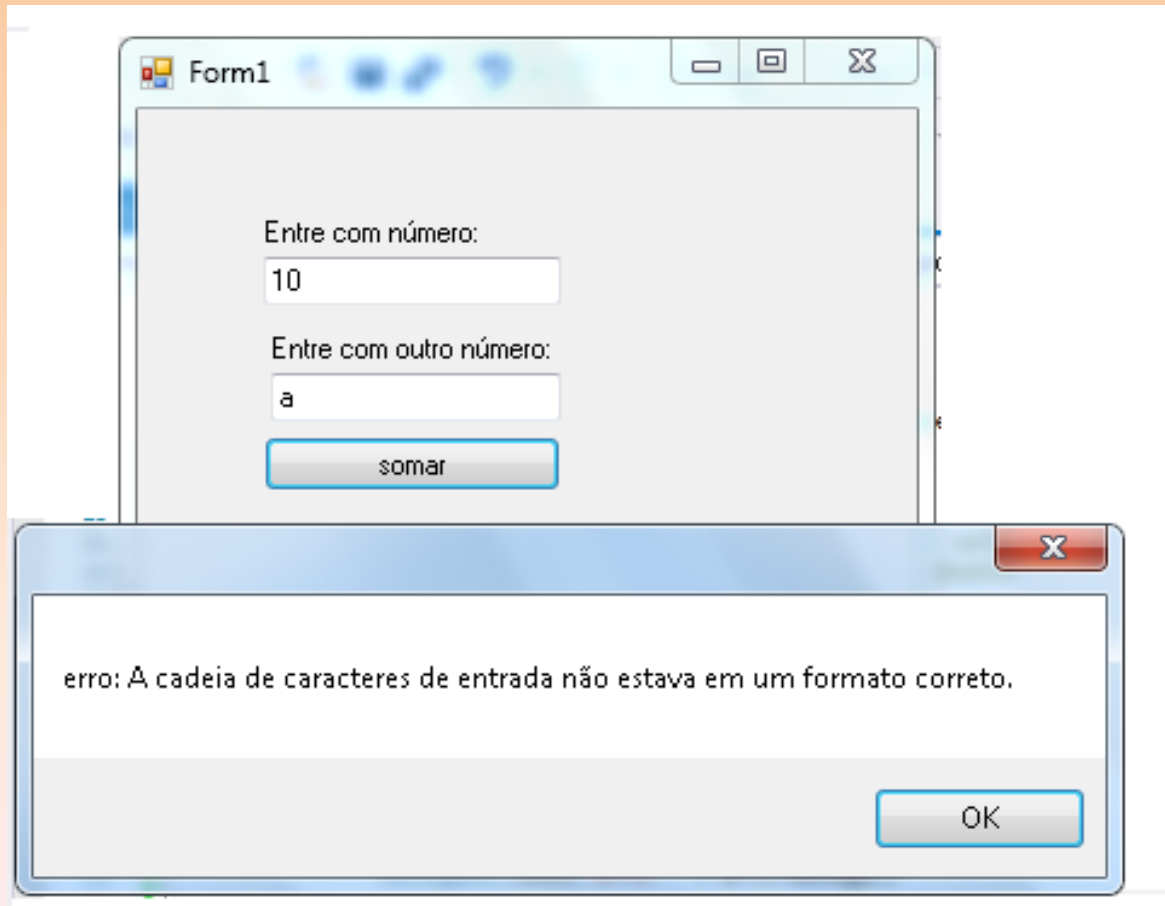
Entre com outro número:
13,53

somar

soma:24,15

OK

Desenvolvimento UI com WinForms



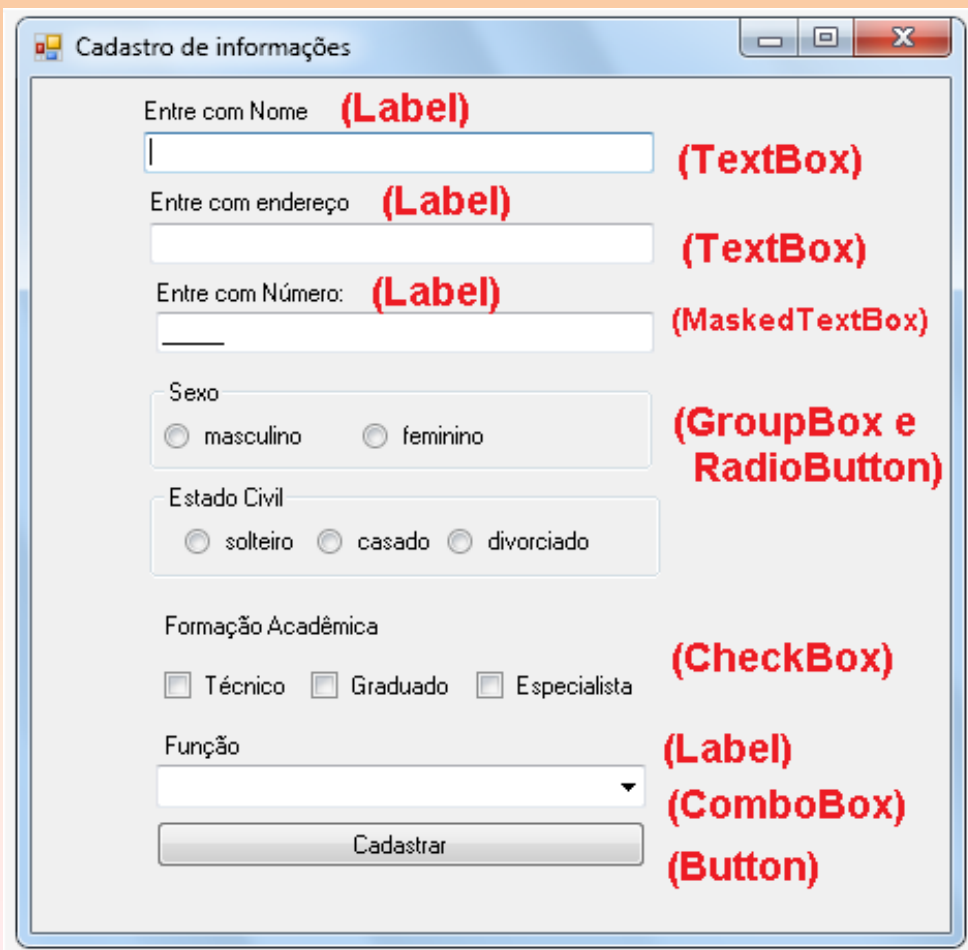
Uso de Try, Catch e tratamento de erro (Exception)

```
20 private void button1_Click(object sender, EventArgs e)
21 {
22     try //tratamento de erro
23     {
24         double n1 = 0.0, n2 = 0.0, s = 0.0; //declara variáveis
25         // converte texto contido em textBox1 para double
26         n1 = Convert.ToDouble(textBox1.Text);
27         // converte texto contido em textBox2 para double
28         n2 = Convert.ToDouble(textBox2.Text);
29         // calcula a soma
30         s = n1 + n2;
31         // mostra resultado
32         MessageBox.Show("soma:" + s);
33     } catch (FormatException erro) {
34         // captura e mostra erro
35         MessageBox.Show("erro: " + erro.Message);
36     }
37 }
38 }
```

Substitui . (ponto) por , (virgula)

```
20 private void button1_Click(object sender, EventArgs e)
21 {
22     try //tratamento de erro
23     {
24         double n1 = 0.0, n2 = 0.0, s = 0.0; //declara variáveis
25         // substitui caracter . por ,
26         textBox1.Text = textBox1.Text.Replace(".", ",");
27         // converte texto contido em textBox1 para double
28         n1 = Convert.ToDouble(textBox1.Text);
29         // substitui caracter . por ,
30         textBox2.Text = textBox2.Text.Replace(".", ",");
31         // converte texto contido em textBox2 para double
32         n2 = Convert.ToDouble(textBox2.Text);
33         // calcula a soma
34         s = n1 + n2;
35         // mostra resultado
36         MessageBox.Show("soma:" + s);
37     } catch (FormatException erro) {
38         // captura e mostra erro de conversão de formato
39         MessageBox.Show("erro: " + erro.Message);
40     }
41 }
42 }
43 }
```

Uso e aplicação Label, TextBox, Radio, CheckBox, ComboBox, Button



Cadastro de informações

Entre com Nome **(Label)**
 (TextBox)

Entre com endereço **(Label)**
 (TextBox)

Entre com Número: **(Label)**
 (MaskedTextBox)


Sexo
☐ masculino ☐ feminino **(GroupBox e RadioButton)**

Estado Civil
☐ solteiro ☐ casado ☐ divorciado

Formação Acadêmica
☐ Técnico ☐ Graduado ☐ Especialista **(CheckBox)**

Função **(Label)**
 (ComboBox)

(Button)



- A** **Label**
Mostrar mensagens ou resultado de processamento
- abl** **TextBox**
Solicita entrada de dados
- ☒ **CheckBox**
Permite seleção múltiplas de opções
- ☐ **RadioButton**
Permite seleção exclusiva de opções
- Button**
Executa uma ação
- ComboBox**
Mostra e permite selecionar opções

ComboBox e evento Form1()


O evento Form1() é primeiro a ser executado quando executamos a aplicação. Neste evento após o método InitializeComponent() apagamos todos os itens do comboBox1 por meio do método cClear() e em seguida adicionamos novos itens com o uso do método Add. Quando o formulário for visualizado o usuário já terá a lista do comboBox1 preenchida. Atenção após a execução do método InitializeComponent() todos os componentes do formulários já foram criados.

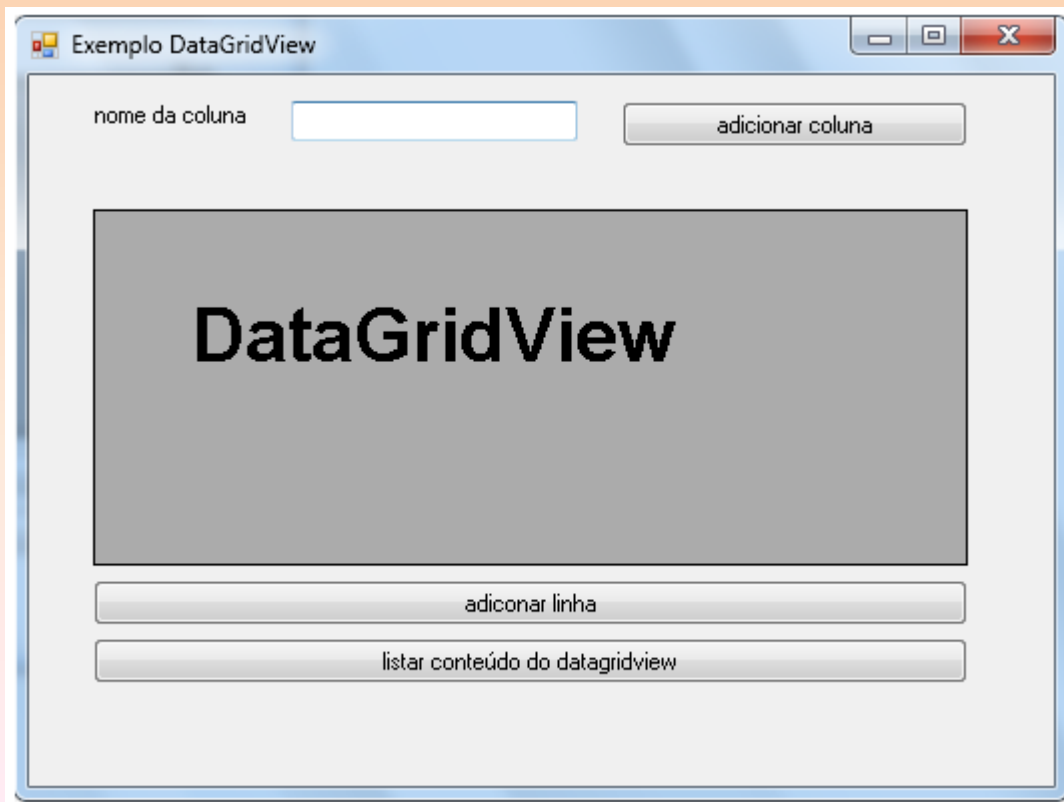
```
15 public Form1()  
16 {  
17     InitializeComponent();  
18     // preenche lista de opções do comboBox1  
19     comboBox1.Items.Clear();// apaga todos itens  
20     // adicionar itens  
21     comboBox1.Items.Add("Analista de Sistemas");  
22     comboBox1.Items.Add("Gerente de Projetos");  
23     comboBox1.Items.Add("Progrmdador");
```

ComboBox e evento Form1() -> Exemplo prático

```
// evento click do Button1
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Conteúdo de um textBox: " + textBox1.Text); // mostra texto do textbox
    MessageBox.Show("Conteúdo de um label: " + label1.Text); // mostra texto label
    if (radioButton1.Checked) // verifica se radioButton1 foi selecionado
        MessageBox.Show("Radio selecionado: " + radioButton1.Text); // obtém texto
    else
        MessageBox.Show("Radio selecionado: " + radioButton2.Text);
    string selecionados = ""; // concatena seleção do checkbox
    if (checkBox1.Checked) selecionados += "(" + checkBox1.Text + " ";
    if (checkBox2.Checked) selecionados += "(" + checkBox2.Text + " ";
    if (checkBox3.Checked) selecionados += "(" + checkBox3.Text + " ";
    // mostra selecionados
    MessageBox.Show("CheckBox selecionado: " + selecionados);
    // verifica se um item do combobox foi selecionado
    if (comboBox1.SelectedIndex == -1)
        MessageBox.Show("Selecionar item do combobox"); // solicita selecionar
    else
    {
        // mostra índice selecionado e texto do item selecionado
        MessageBox.Show("índice selecionado: " + comboBox1.SelectedIndex);
        MessageBox.Show("texto do índice selecionado: " + comboBox1.Items[comboBox1.SelectedIndex]);
    }
    // mostra texto indicado no combobox1
    MessageBox.Show("texto informado no combobox: " + comboBox1.Text);
}
```


DataGridView

 **DataGridView** É um componente que permite visualizar informações no formato matriz, onde as colunas representam campos específicos e as linhas o conteúdo destes campos.



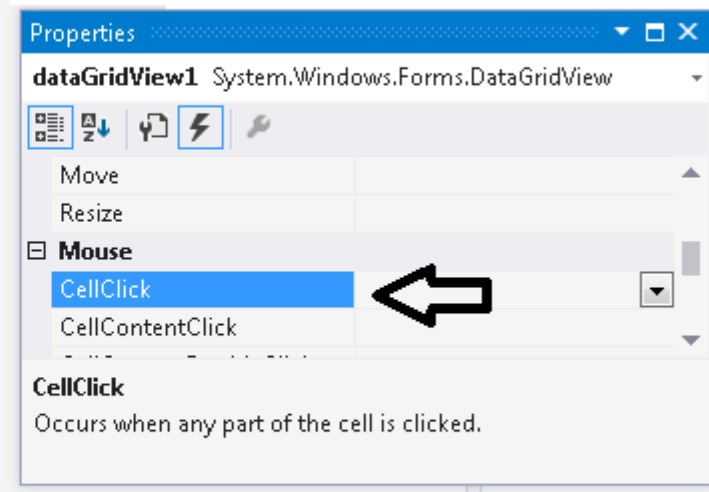
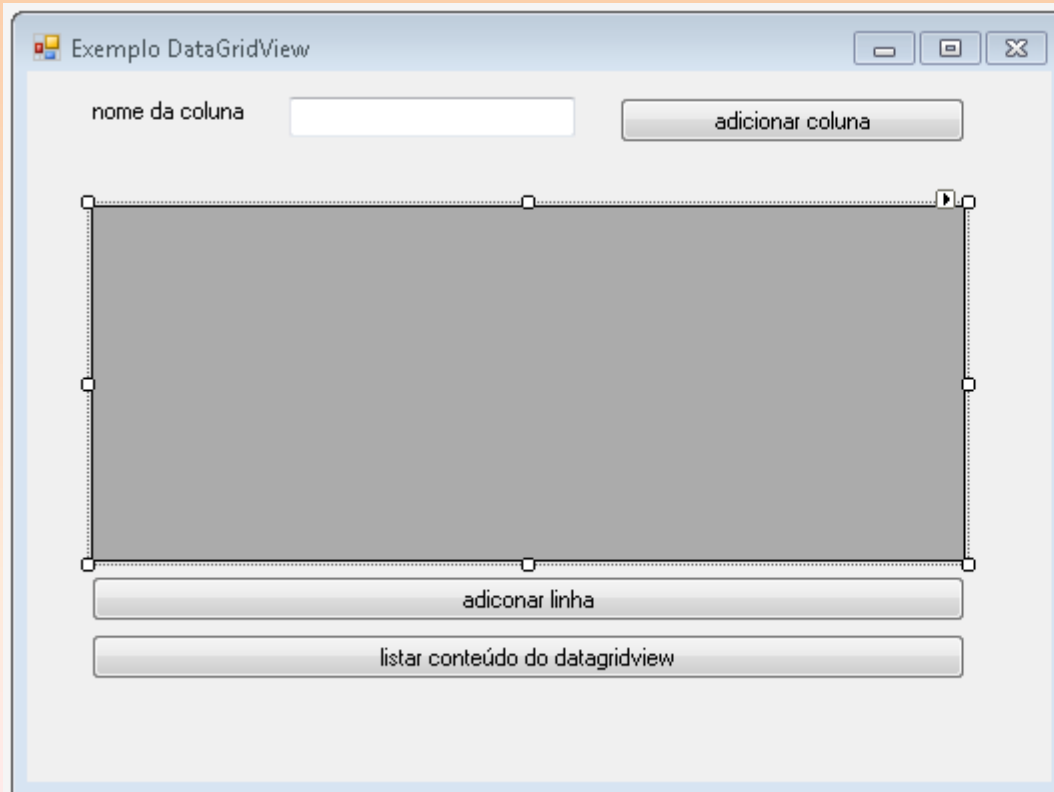
DataGridView

Abaixo código que permite criar colunas, linhas e listar conteúdo de cada célula do datagridview

```
20 private void button1_Click(object sender, EventArgs e)
21 {
22     //adicionar uma coluna ao datagridview
23     dataGridView1.Columns.Add(textBox1.Text, textBox1.Text);
24 }
25
26 private void button2_Click(object sender, EventArgs e)
27 {
28     //adicionar uma linha ao datagridview
29     dataGridView1.Rows.Add();
30 }
31
32 private void button3_Click(object sender, EventArgs e)
33 {
34     // percorre as linhas
35     for (int l=0;l<dataGridView1.RowCount;l++)
36         // percorre as colunas
37         for (int c = 0; c < dataGridView1.Columns.Count; c++)
38         {
39             // contém informação associada?
40             if (dataGridView1.Rows[l].Cells[c].Value!=null)
41                 // mostra informação
42                 MessageBox.Show(dataGridView1.Rows[l].Cells[c].Value.ToString());
43         }
44 }
```

DataGridView: implementando evento CellClick

Na janela de propriedades no ícone “raio” (eventos) dê um duplo clique no evento CellClick



DataGridView: implementando evento CellClick

```
46 private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
47 {
48     int l = 0, c = 0;
49     l = e.RowIndex; // obtém linha selecionada
50     c = e.ColumnIndex; // obtém coluna selecionada
51     //mostrar valor da célula (encontro da linha e coluna selecionada)
52     // contém informação associada?
53     if (dataGridView1.Rows[l].Cells[c].Value != null)
54         // mostra informação
55         MessageBox.Show(dataGridView1.Rows[l].Cells[c].Value.ToString());
56 }
```

C# Linguagem de Programação

Atividades

- 1) Com base nos exemplos vistos nesta aula implemente os seguintes programas em C# utilizando Window Form:
 - a) Dado o nome de uma disciplina, o total de aulas dadas até o momento e as faltas do aluno, calcular a frequência do aluno. Posterior ao cálculo envie uma mensagem informando se o aluno está abaixo da frequência esperada (inferior a 75%) ou não.
 - b) Dado um número inteiro qualquer implementar um programa que calcule a respectiva tabuada.

C# Linguagem de Programação

Atividades

c) Dado o layout ao lado:

Implemente um programa Em C# Windows Form que Analisa a seleção de candidatos a uma vaga na área de Ti.

Regra para seleção:

- Conhecer Linux e Android;
- Conhecer C# ou Java;
- Conhecer MySql e Sql Server

Mostrar uma mensagem “você foi selecionado para a vaga na área de TI”, quando o candidato for selecionado.

Nome

Curso em que está matriculado?

☐ Ciência da Computação ☐ Eng. da Computação

Sistema Operacional que tem conhecimento:

☐ Windows ☐ Linux ☐ Android

Linguagens que tem conhecimento:

☐ C# ☐ PHP ☐ Java

Banco de Dados que tem conhecimento:

☐ MySql ☐ Sql Server ☐ PostGree

C# Linguagem de Programação

Atividades

d) Implemente um programa em C# que solicite ao usuário o nome de um aluno e a respectiva nota de avaliação (de 0.0 até 10.0). As informações solicitadas deverão ser armazenadas em um objeto DataGridView. Um botão deverá percorrer o conteúdo armazenado na DataGridView e filtrar os alunos que tenham nota de avaliação inferior a 6.0. Os alunos com nota inferior a 6.0 deverão ser copiados para um outro DataGridView.

NOME NOTA

ARMAZENAR INFORMAÇÕES

NOME DO ALUNO	NOTA DE AVALIAÇÃO
RICARDO	10.0
MARIA	5.0
JOSÉ	4.0
JOÃO	8.0

FILTRAR NOTAS ABAIXO DE 6.0

NOTA DE AVALIAÇÃO	
MARIA	5.0
JOSÉ	4.0

C# Linguagem de Programação

Atividades

Selecionar
Produtos

1 - produto A R\$ 100,00
2- produto B R\$ 50,00
3- produto C R\$ 30,00

Selecionar Cliente

1 - Cliente X
2- Cliente Y
3- Cliente Z

Quantidade

Adicionar

Produtos Selecionados

idProduto	Nome do Produto	Preço do Produto	Quantidade	Total
3	Produto C	30,00	2	60,00
1	Produto A	100,00	1	100,00

Cliente Selecionado: 3 - Cliente Z

Finalizar

Pesquisar e permitir remover um produto já inserido no pedido

e) O layout ao lado mostra uma versão básica de um carrinho de compras. Implemente o layout ao lado utilizando os componentes vistos durante a aula. Para armazenar temporariamente utilize componente `DataGridView`. Adicione também um controle básico de estoque. Todas vez que um produto for adicionado, subtrair a quantidade desejada pelo que está no estoque (se possível).

C# Linguagem de Programação

Referências:



- John Shapr, Microsoft Visual C# 2013 passo a passo, Ed. Bookman