

# SISTEMAS DISTRIBUÍDOS

C# - Windows Forms (validação)  
Visual Studio

PROFESSOR HILDEBERTO MELO



# Apresentação

- Técnico em Desenvolvimento de Sistemas - Ibratec, Recife-PE
- Bacharel em Sistemas de Informação – FIR, Recife-PE
- Especialista em Docência no Ensino Superior – Faculdade Maurício de Nassau, Recife-PE
- Mestre em Ciência da Computação – UFPE/CIN, Recife-PE
- *Currículo Lattes* <http://lattes.cnpq.br/0759508594425296>)
- *Homepage* <https://sites.google.com/site/hildebertomelo/>



# Disciplinas Lecionadas

---

- Desenvolvimento de Aplicações Desktop
- Programação Orientada a Objetos
- Estrutura de Dados
- Tecnologia da Informação & Sociedade
- Sistemas Operacionais
- Sistemas Distribuídos
- Introdução a Informática
- Lógica de Programação
- Informática Aplicada a Saúde
- Banco de Dados
- Projeto de Banco de Dados
- Análise de Projetos Orientado a Objetos
- Programação Cliente Servidor
- Linguagens de Programação: C, C#, Pascal, PHP, ASP, Delphi, Java, JavaScript
- Programação WEB



# Roteiro

---

- Validação de campos em formulário
- Exercícios



# Identar Código Fonte

---

- Ctrl+E, D --> Formata documento
- 
- Ctrl+E. F --> Formata seleção.



# Dica

---

- No método do clique de um botão, utilizar o comando **return** para encerrar o método.

```
if (textBoxNome.Text == "")  
{  
    MessageBox.Show("Informar o nome do funcionário", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);  
    return;  
}
```



# Dica

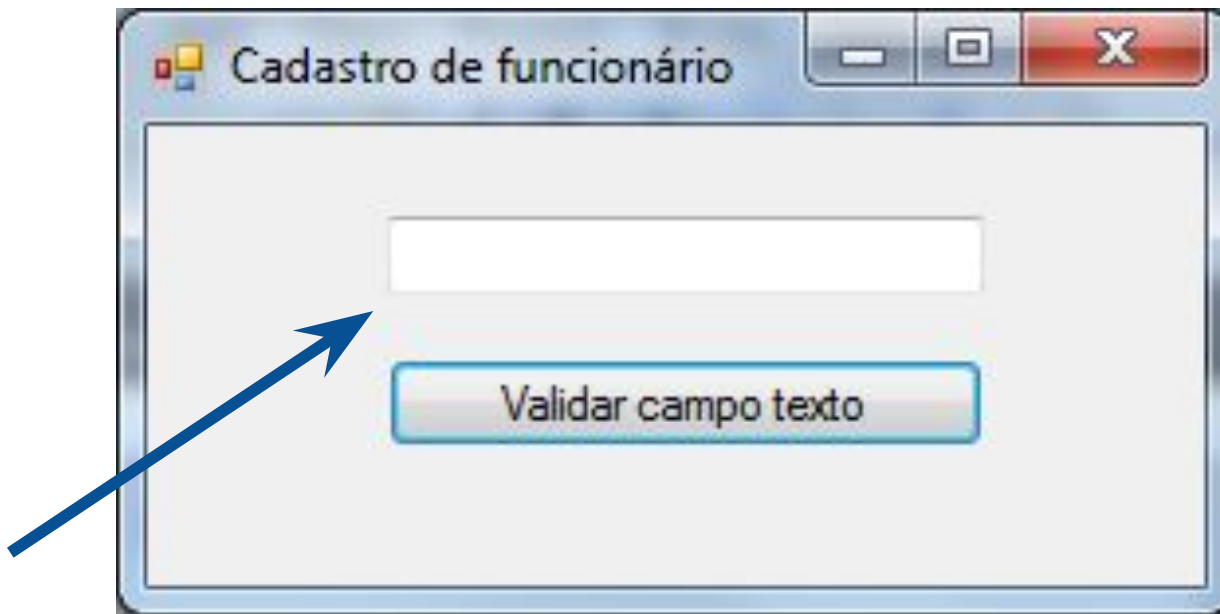
---

- Utilizar o comando Focus para enviar o foco a um determinado componente.

```
if (textBoxNome.Text == "")  
{  
    MessageBox.Show("Informar o nome do funcionário","Atenção",MessageBoxButtons.OK,MessageBoxIcon.Asterisk);  
    textBoxNome.Focus();  
    return;  
}
```



# Campos Texto





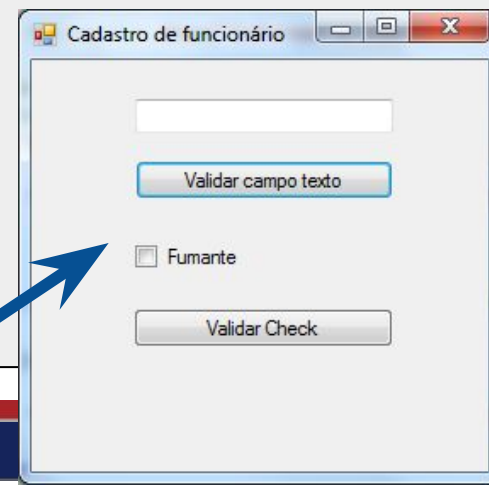


```
private void button1_Click(object sender, EventArgs e)
{
    //verificando se o usuário informou o nome
    //primeira forma
    if (textBoxNome.Text == "")
    {
        MessageBox.Show("Informar o nome do funcionário", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        textBoxNome.Focus();
        return;
    }
    //verificando se o usuário informou o nome
    //segunda forma
    if (textBoxNome.Text.Length < 4)
    {
        MessageBox.Show("O nome do funcionário deverá ter mais de 4 caracteres", "Atenção", MessageBoxButtons.OK,
        MessageBoxIcon.Asterisk);
        textBoxNome.Focus();
        return;
    }
    //verificando se o usuário informou o nome
    //terceira forma
    if (textBoxNome.Text.Trim() == "")
    {
        MessageBox.Show("Informar o nome do funcionário", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        textBoxNome.Focus();
        return;
    }
    //verificando se o usuário informou o nome
```



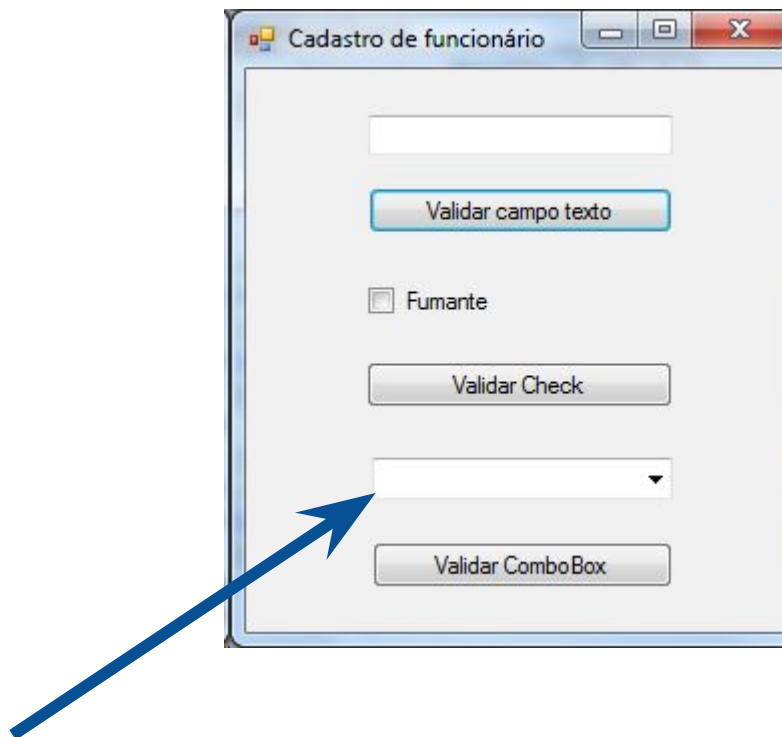
# Campos CheckBox

```
private void button2_Click(object sender, EventArgs e)
{
    if (checkBoxFumante.Checked == true)
    {
        MessageBox.Show("Você é fumante", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
    else
    {
        MessageBox.Show("Você não é fumante", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
}
```





# Campos ComboBox





```
private void button3_Click(object sender, EventArgs e)
{
    //mostrando o index selecionado
    MessageBox.Show("Index selecionado: " + comboBoxEstadoCivil.SelectedIndex, "Atenção", MessageBoxButtons.OK,
    MessageBoxIcon.Asterisk);
    //verificando se escolheu um item
    if (comboBoxEstadoCivil.SelectedIndex <= -1)
    {
        MessageBox.Show("Informar o estado civil", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        comboBoxEstadoCivil.Focus();
        return;
    }
    //mostrando o texto do index selecionado
    if (comboBoxEstadoCivil.SelectedIndex >= 0 )
    {
        int index = comboBoxEstadoCivil.SelectedIndex;
        String texto = comboBoxEstadoCivil.Items[index].ToString();

        MessageBox.Show(texto, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
}
```



# Campo ListBox

The screenshot shows a Windows application window titled "Cadastro de funcionário". The window contains several form controls and validation buttons:

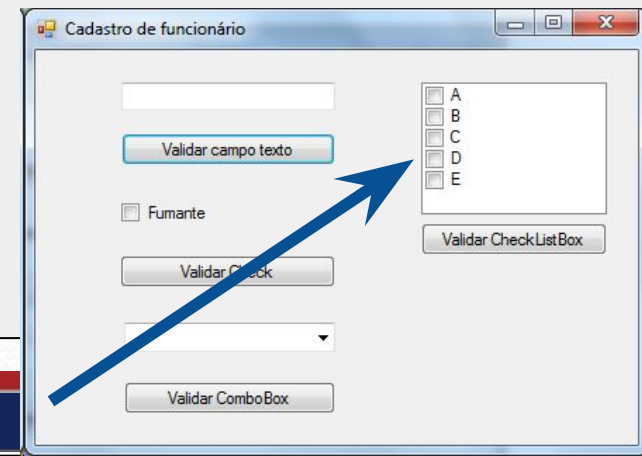
- A text input field at the top left.
- A button labeled "Validar campo texto" below the text input.
- A checkbox labeled "Fumante" below the "Validar campo texto" button.
- A button labeled "Validar Check" below the "Fumante" checkbox.
- A dropdown menu (ComboBox) below the "Validar Check" button.
- A button labeled "Validar ComboBox" below the dropdown menu.
- A list box on the right side containing five items: A, B, C, D, and E, each with a checkbox.
- A button labeled "Validar CheckListBox" below the list box.
- A list box on the right side containing three items: Santa Cruz, Náutico, and Sport (infelizmente).
- A button labeled "Validar ListBox" below the second list box.

A blue arrow points from the bottom left towards the "Validar ListBox" button.



# Campo CheckedListBox

```
private void button4_Click(object sender, EventArgs e)
{
    //verificando se escolheu algum item
    if (checkedListBox1.CheckedItems.Count <= 0)
    {
        MessageBox.Show("Escolher pelo menos um item", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        checkedListBox1.Focus();
        return;
    }
    //listando os itens selecionados
    for (int i = 0; i < checkedListBox1.CheckedItems.Count; i++)
    {
        MessageBox.Show(checkedListBox1.CheckedItems[i].ToString(), "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
}
```





# Campo ListBox

```
private void button5_Click(object sender, EventArgs e)
{
    //mostrando o index selecionado
    MessageBox.Show("Index selecionado: " + listBoxTimes.SelectedIndex, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    //verificando se escolheu um item
    if (listBoxTimes.SelectedIndex <= -1)
    {
        MessageBox.Show("Informar o seu time", "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        listBoxTimes.Focus();
        return;
    }
    //mostrando o texto do index selecionado
    if (listBoxTimes.SelectedIndex >= 0)
    {
        int index = listBoxTimes.SelectedIndex;
        String texto = listBoxTimes.Items[index].ToString();

        MessageBox.Show(texto, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
}
```



# Formulário

---

- Criar um formulário de cadastro com os seguintes campos:
  - Nome, endereço, número, complemento, bairro, cidade, uf, país, data de nascimento, cpf, rg, filiação.
  - Agrupar as informações por características
  - Todos os campos são obrigatórios, portanto validar os mesmos.
  - Exibir mensagem com as informações do formulário, cada campo deverá ser exibido em uma linha da mensagem.
  - Utilizar fontes e cores que facilitem o preenchimento das informações pelo usuário.
- Aplicar o que foi visto em sala de aula: funções e mensagens.
- O exercício é individual e deverá ser entregue na próxima aula.





# Formulário

---

- **Recomendações:**

- Utilizar o padrão de mensagens com ícones, títulos e mensagem ao usuário
- Renomear os componentes no formulário
- Identar o código fonte corretamente
- Organização e alinhamento dos componentes no formulário são muito importantes



# Exemplo

Validação de Campos

Nome:

Data de nascimento:

Endereço

Logradouro:

Número:

Complemento:

Bairro:

Cidade:

Pais:

Filiação

Pai:

Mãe:

Documentos

CPF:

RG:

Validar Agrupado

Validar Individual



# Mensagem Agrupada

Validação de Campos

Nome:

Data de nascimento:

Endereço:

Logradouro:

Número:

Complemento:

Bairro:

Cidade:

Pais:

Filiação:

Pai:

Mãe:

Documento:

CPF:

RG:

Validar Agrupado

Validar Individual

**Atenção**

Informar o nome da pessoa.  
Informar a data de nascimento no formato dd/mm/aaaa.  
Informar o logradouro.  
Informar o complemento.  
Informar o bairro.  
Informar a cidade.  
Informar o país.  
Informar o nome do pai.  
Informar o nome da mãe.  
Informar o RG.  
Informar o CPF.

OK



# Mensagem Individual

Validação de Campos

Nome:

Data de nascimento:

Endereço

Logradouro:

Número

Complemento:

Bairro:

Cidade:

Pais:

Filiação

Pai:

Mãe:

Documentos


CPF:

RG:

Validar Agrupado

Validar Individual

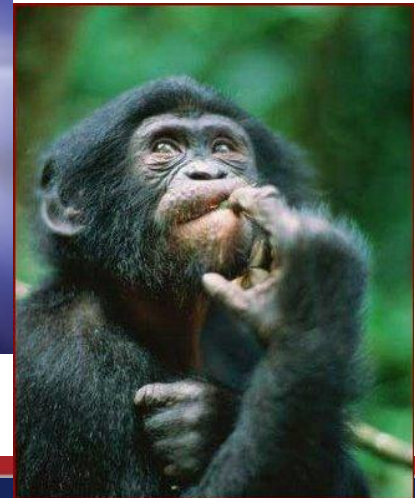
**Atenção**

 Informar o nome da pessoa.

OK



# Perguntas





# Referências

Bibliografia Básica (títulos, periódicos, etc.)						
Título/Periódico	Autor	Edição	Local	Editora	Ano	LT/BV¹
Microsoft Visual C# 2008: passo a passo	SHARP, John	1ª	Porto Alegre	Bookman	2008	LT
Desenvolvimento em camadas com C# .Net	CAMACHO JÚNIOR, Carlos Olavo de Azevedo	1ª	Florianópolis	Visual Books	2008	LT
C#: como programar	DEITEL, H. M.	1ª	São Paulo	Pearson	2003	LT
Bibliografia Complementar (títulos, periódicos, etc.)						
Título/Periódico	Autor	Edição	Local	Editora	Ano	LT/BV¹
Use a cabeça C#	STELLMAN, Andrew; GREENE, Jeniffer	2ª	Rio de Janeiro	Alta Books	2011	LT
Treinamento avançado em XML	FARIA, Rogério Amorim De	1ª	São Paulo	<u>Digerati</u>	2005	LT
Projeto de Banco de Dados com XML	GRAVES, Mark	1ª	São Paulo	Pearson	2003	BV