

Componentes de Seleção – C#

Componentes de Seleção

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

Operadores Relacionais e Lógicos

Os operadores relacionais do C# realizam *comparações* entre variáveis. São eles:

Operador	Ação
>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de
&&	And
	Or

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

Problema na Conversão de Tipos

A conversão de tipos numéricos possui dois problemas que acontecem em tempo de execução, são eles:

- Conversão de valores NULOS para valores numéricos (quando o usuário deixa um TextBox vazio)
- Conversão de valores alfanuméricos (letras) para valores numéricos (quando o usuário preenche o TextBox com letras)

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

EXEMPLO 1

**FAÇA UM PROGRAMA QUE LEIA
SOMENTE LETRAS NO TEXTBOX DO
NOME E SOMENTE NÚMEROS NO
TEXTBOX DO SALÁRIO E CALCULE O
NOVO SALÁRIO (40% DE REAJUSTE)**

Profa. Ana Paula Citro Fugarra Rodrigues

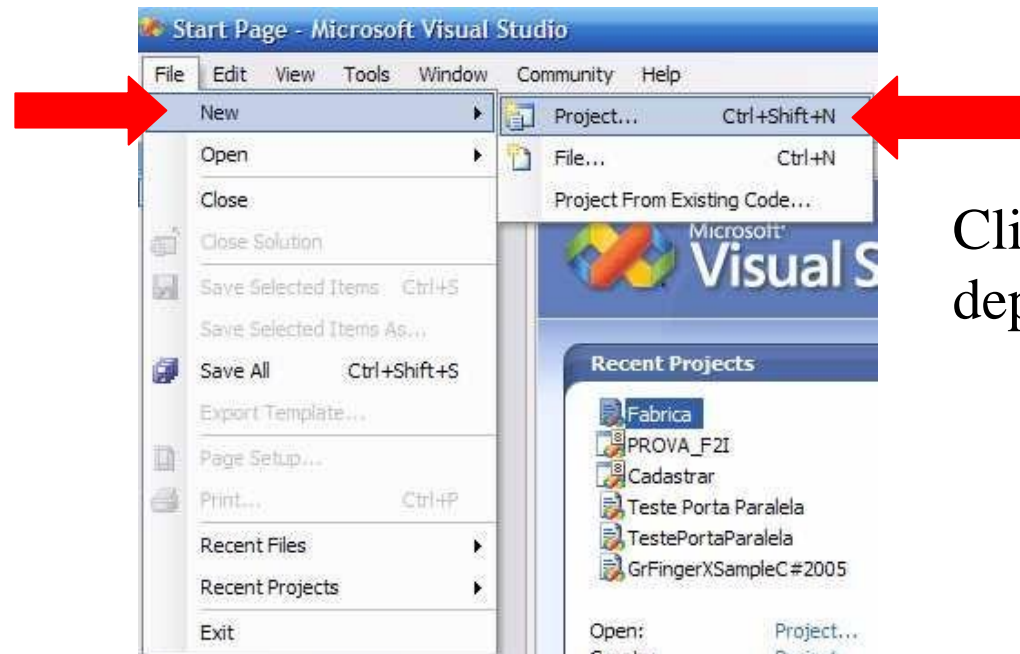
Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

1. **Passo:** Criar um projeto

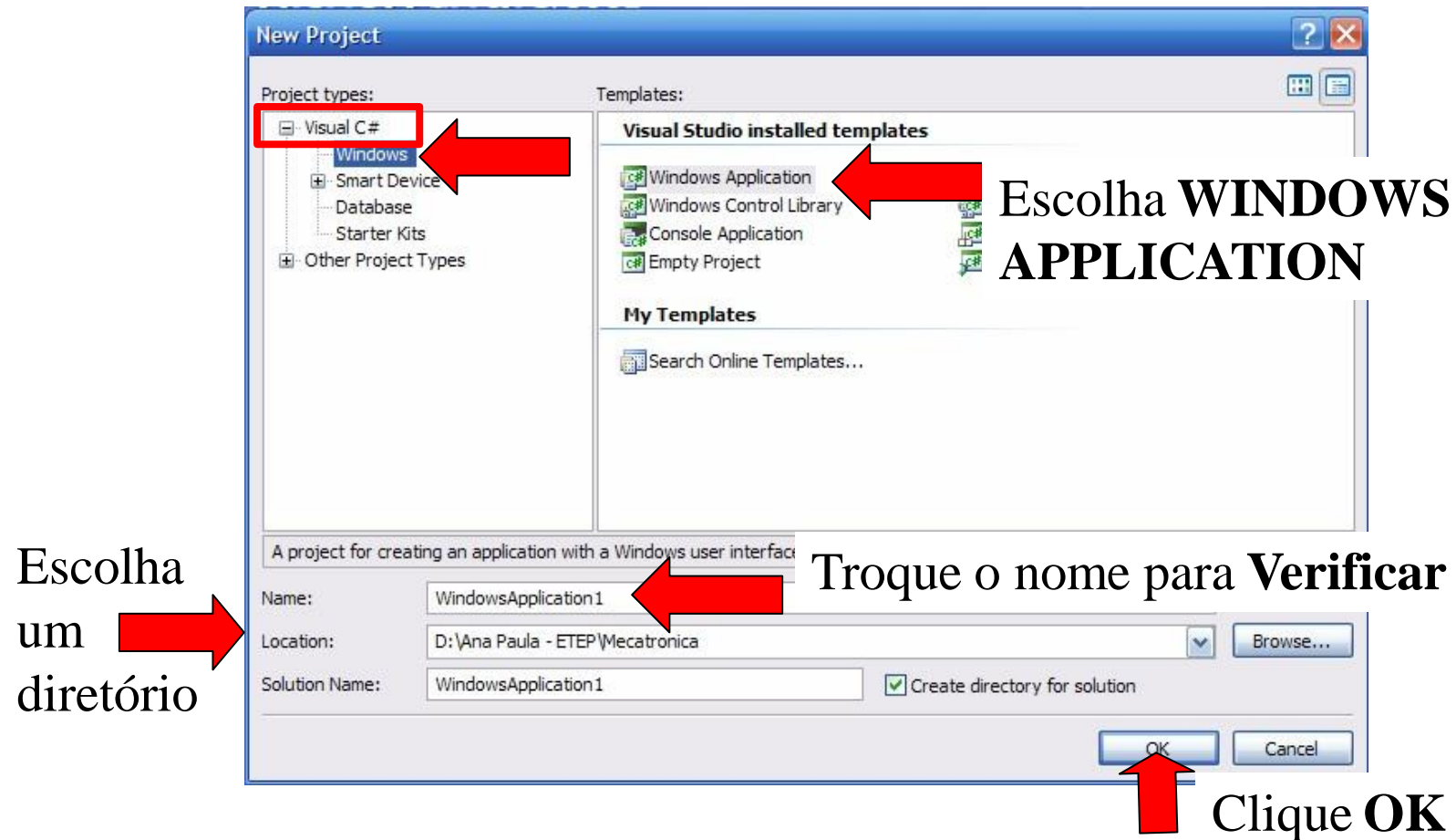


Clique no Menu File



Clique no SubItem New e depois clique em Project

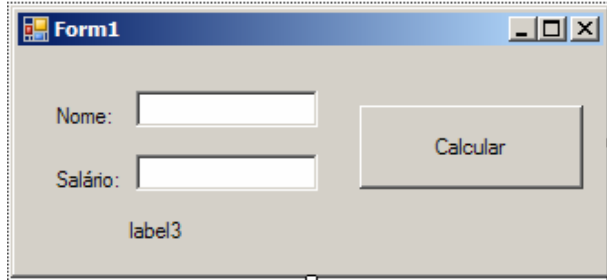
Componentes de Seleção – C#



Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#



Adicione:

3 Labels

2 TextBoxs

1 Button

Altere as propriedades NAME e TEXT para:

Label1

Text: Nome

Name: lblNome

Label2

Text: Salário

Name: lblSalario

Label3

Text: “vazio”

Name: lblResp

TextBox1

Name: txtNome

TextBox2

Name: txtSalario

Button1

Text: Calcular

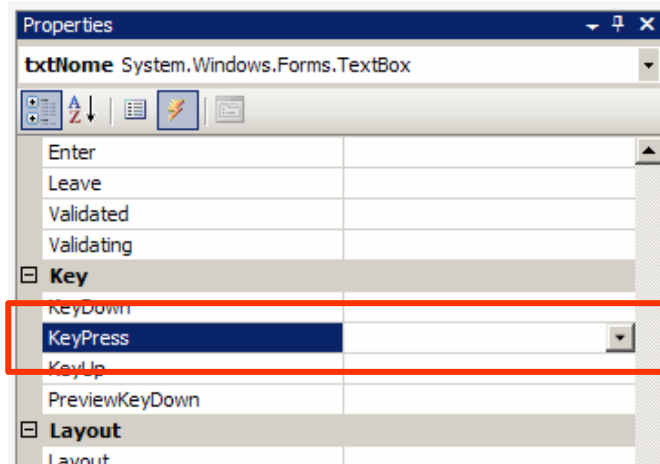
Name: btnCalcular

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

Para travarmos as letras ou os números de um TextBox precisamos usar o evento **KEYPRESS**. Este evento é responsável pela captura de cada tecla pressionada no TextBox.



Selecione o TextBox do Nome (txtNome), acione o evento **KeyPress** e digite o código:


Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

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

        private void txtNome_KeyPress(object sender, KeyPressEventArgs e)
        {
            if ((e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= 'A' && e.KeyChar <= 'Z'))
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
            }
        }
    }
}
```



Este código é responsável pela verificação letra a letra (digitada) no TextBox. Caso esta letra esteja entre as letras minúsculas/maiúsculas ele libera a exibição da mesma (**e.Handled=false**), caso contrário ele apaga a letra digitada (**e.Handled=true**).

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

Observe que o código digitado anteriormente libera somente as letras (minúsculas/maiúsculas), conclusão não conseguimos usar o Backspace para apagar um texto digitado errado. Para corrigir o erro usaremos a tabela ASCII, neste caso o código do Backspace é o número 8.

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

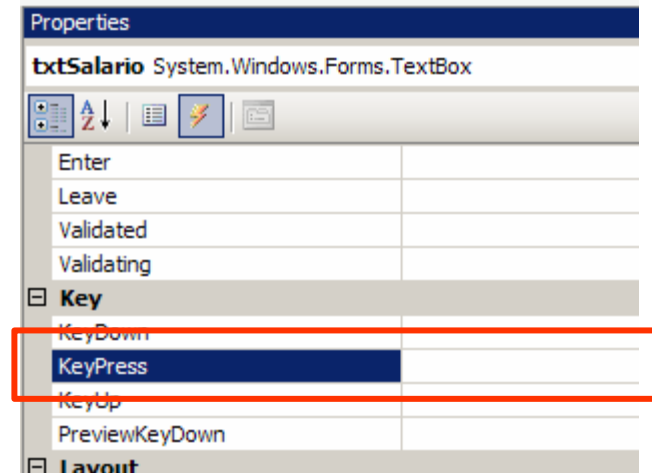
        private void txtNome_KeyPress(object sender, KeyPressEventArgs e)
        {
            if ((e.KeyChar>='a' && e.KeyChar<='z') || (e.KeyChar>='A' && e.KeyChar<='Z') || e.KeyChar==8)
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
            }
        }
    }
}
```

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

Agora usaremos os mesmos passos para o TextBox do Salário (txtSalario).



Selecione o TextBox do Salário (txtSalario), acione o evento KeyPress e digite o código:

Componentes de Seleção – C#

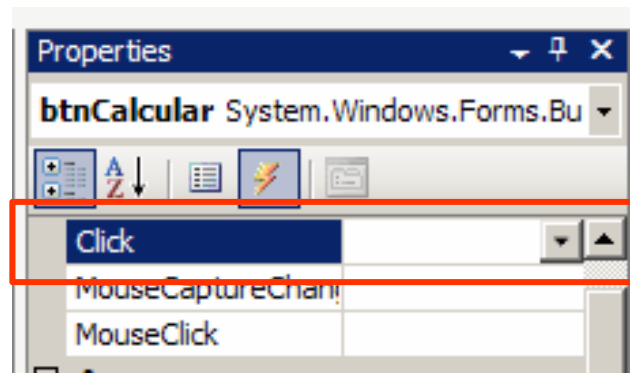
```
namespace Verificar
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void txtNome_KeyPress(object sender, KeyPressEventArgs e)
        {
            if ((e.KeyChar>='a' && e.KeyChar<='z') || (e.KeyChar>='A' && e.KeyChar<='Z') || e.KeyChar==8)
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
            }
        }

        private void txtSalario_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (e.KeyChar >= '0' && e.KeyChar <= '9' || e.KeyChar == 8)
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
            }
        }
    }
}
```

Componentes de Seleção – C#

Agora vamos efetuar o cálculo do novo salário, lembrando que talvez o usuário pode deixar de preencher o textbox e clicar em Calcular.



Acione o evento **CLICK** do btnCalcular e escreva o código:

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double sal, novo_sal;

    if (txtSalario.Text == "")
    {
        MessageBox.Show("Favor preencher seu salário!!!");
    }
    else
    {
        sal = Convert.ToDouble(txtSalario.Text);

        novo_sal = sal * 1.40;

        lblResp.Text = "O novo salário é R$ " + novo_sal;
    }
}
```

Verifica se o txtSalário é vazio exibindo uma mensagem

Caso contrário efetua a conversão e os cálculos

Conclusão:

Para corrigirmos o erro conversão quando um textbox está vazio devemos fazer um if antes da conversão numérica, isto é, a conversão só pode acontecer caso o textbox tenha sido preenchido.

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

EXEMPLO 2

**FAÇA UM PROGRAMA LEIA O NOME, A
TURMA E O CURSO DE UM ALUNO
EXIBINDO TODAS AS INFORMAÇÕES
EM UM TEXTBOX**

Profa. Ana Paula Citro Fugarra Rodrigues

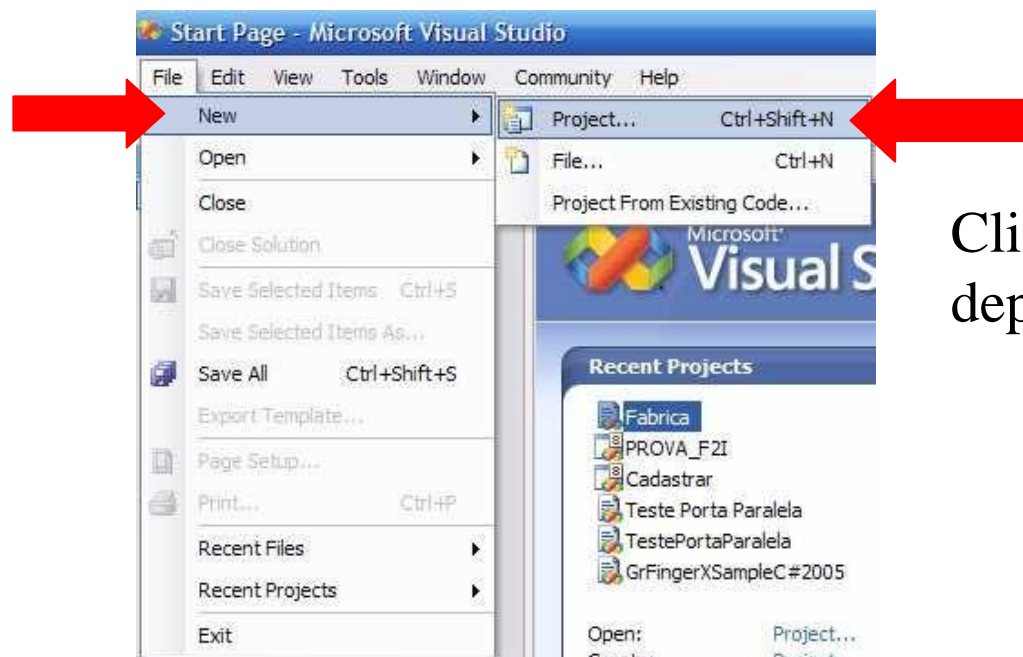
Prof. Ronaldo Vaqueli de Paula

Componentes de Seleção – C#

1. **Passo:** Criar um projeto

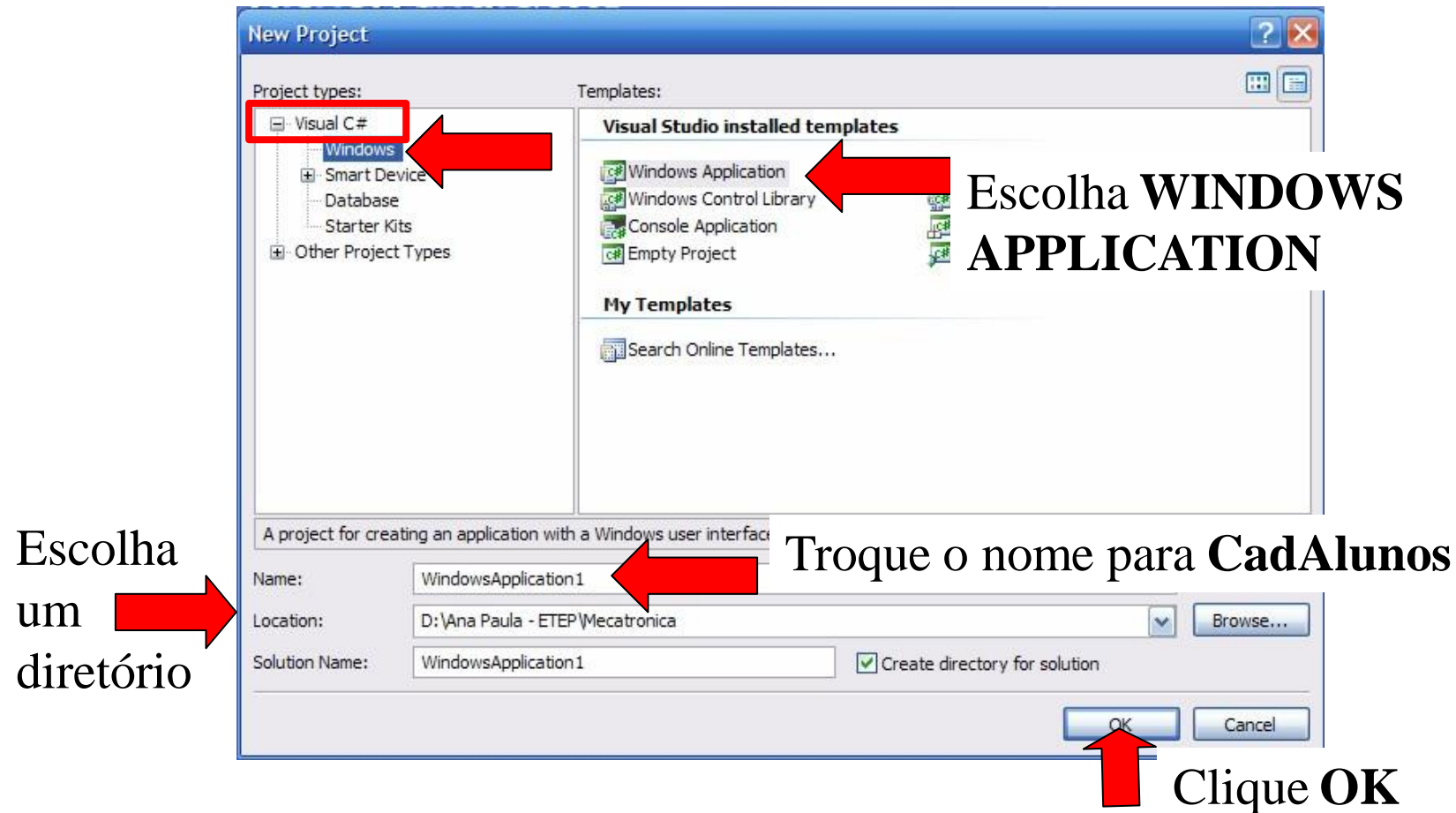


Clique no Menu File

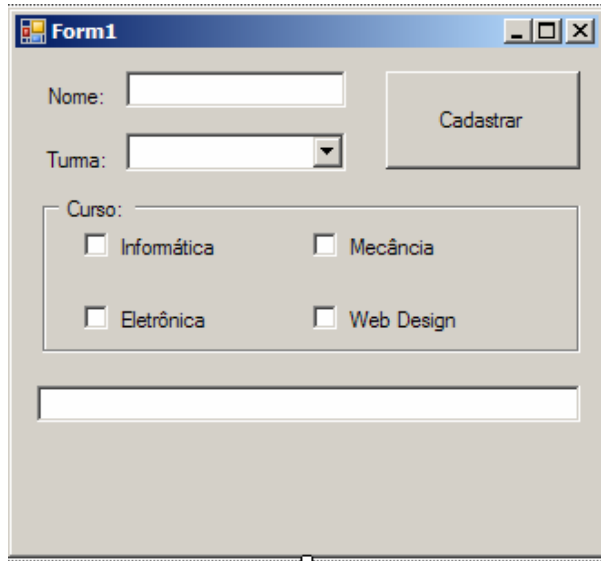


Clique no SubItem New e depois clique em Project

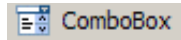
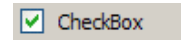
Componentes de Seleção – C#



Componentes de Seleção – C#



- Adicione:
- 2 Labels
- 2 TextBoxs
- 1 Button
- 4 CheckButtons
- 1 GroupBox
- 1 ComboBox



- Altere as propriedades Text dos Objetos para que seu Form fique igual ao Form acima. Depois altere as propriedades Name para:

Label1: lblNome

Label2: lblTurma

TextBox1: txtNome

TextBox2: txtInformacoes

GroupBox1: grpbxCurso

ComboBox1: cmbxTurma

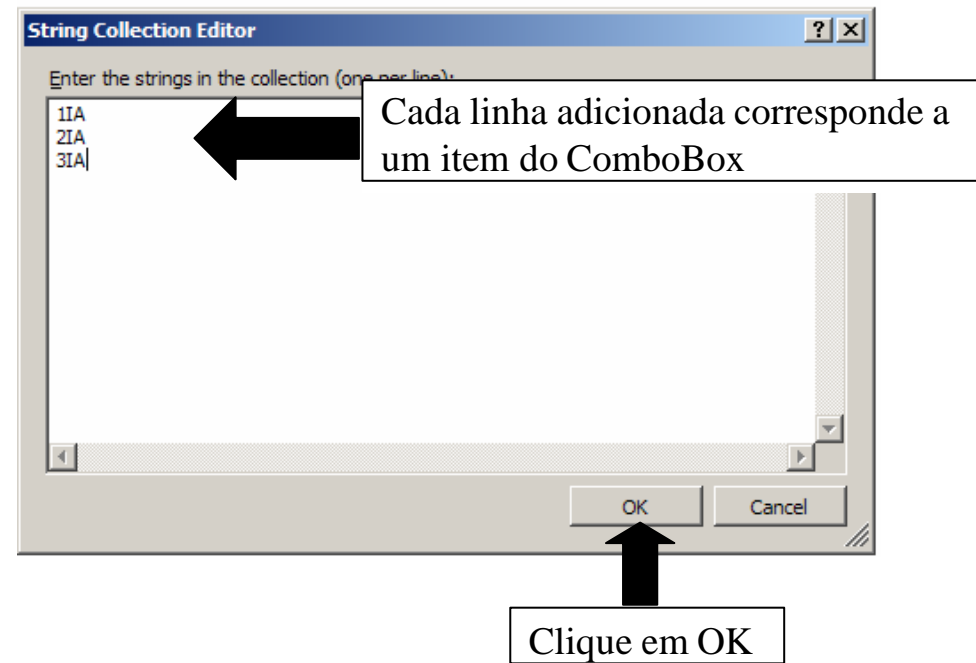
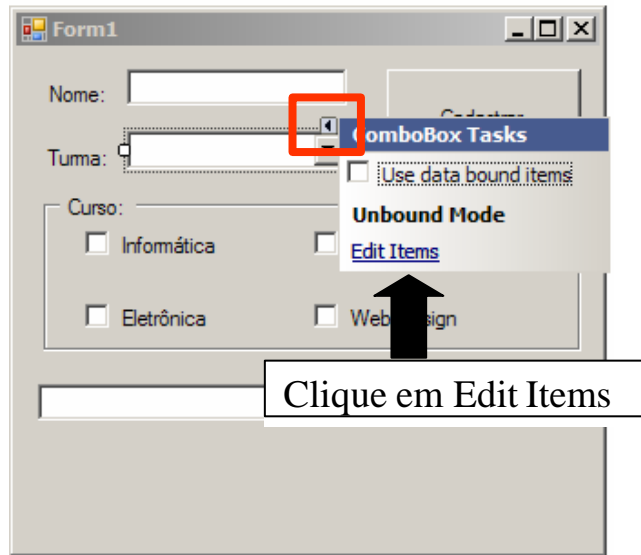
CheckBox1: chkbxInfo

CheckBox2: chkbxEle

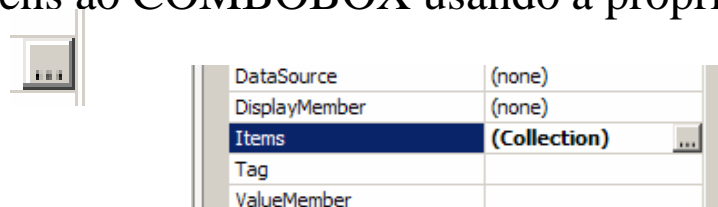
CheckBox3: chkbxMec

CheckBox4: chkbxWeb

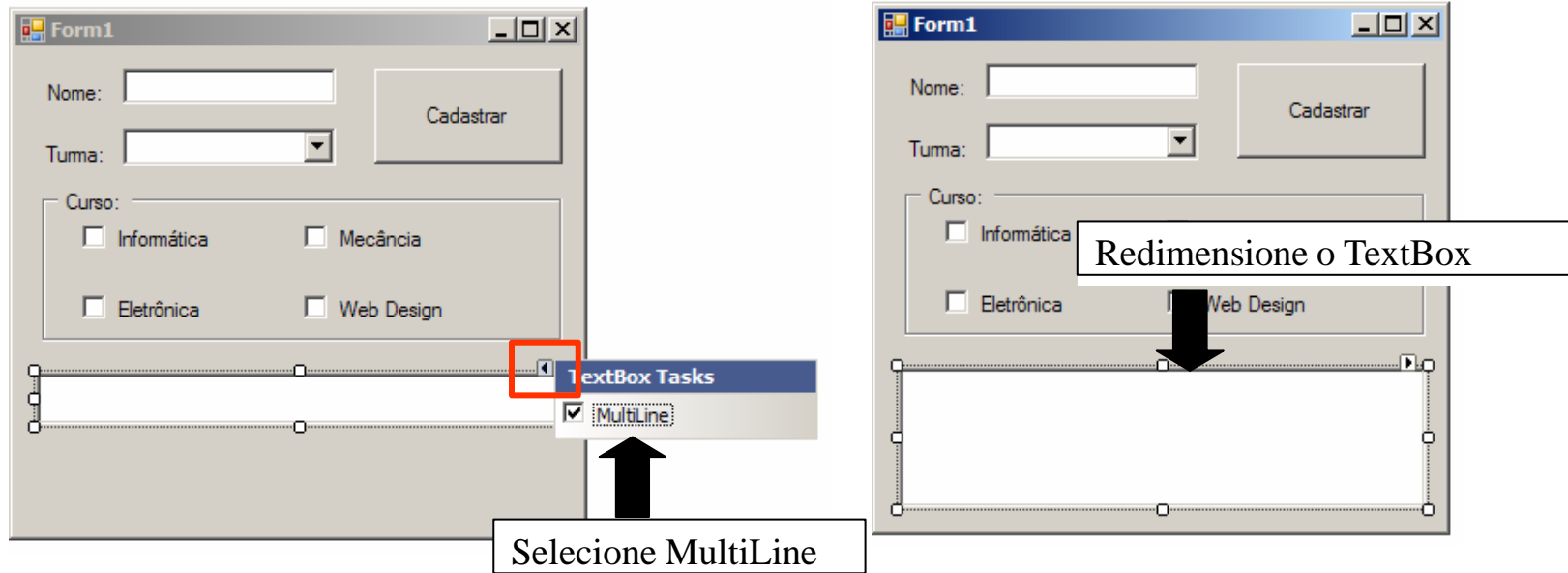
Componentes de Seleção – C#



Também podemos adicionar itens ao COMBOBOX usando a propriedade ITEMS, da janela Properties, clicando em



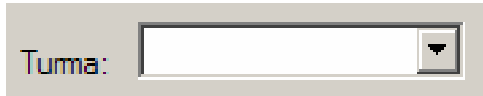
Componentes de Seleção – C#



Também podemos selecionar Multilines de TextBox usando a propriedade Multilines, na janela Properties, alterando de **FALSE** para **TRUE**.

ImeMode	NoControl
MaxLength	32767
Multiline	True
PasswordChar	

Componentes de Seleção – C#



O ComboBox usa a propriedade TEXT para armazenar o item selecionado pelo usuário como se fosse um TEXTBOX.



O CheckButton usa a propriedade CHECKED para armazenar se o item foi selecionado pelo usuário. Quando a propriedade CHECKED é **FALSE** significa item não selecionado e quando ela é **TRUE** significa item selecionado

Usando estes conceitos vamos codificar o programa. Acione o evento CLICK do botão Cadastrar e digite o código:

Componentes de Seleção – C#

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

        private void btnCadastrar_Click(object sender, EventArgs e)
        {
            String resp;

            resp = "Nome: " + txtNome.Text + "\r\n";
            resp = resp + "Turma: " + cmbxTurma.Text + "\r\n";

            if (chkbxInfo.Checked == true)
            {
                resp = resp + "Curso: Informática \r\n";
            }
            if (chkbxEle.Checked == true)
            {
                resp = resp + "Curso: Eletrônica \r\n";
            }
            if (chkbxMec.Checked == true)
            {
                resp = resp + "Curso: Mecânica \r\n";
            }
            if (chkbxWeb.Checked == true)
            {
                resp = resp + "Curso: Web Design \r\n";
            }

            txtInformacoes.Text = resp;
        }
    }
}
```

Componentes de Seleção – C#

Analisando o código:

```
String resp;
```

A variável resp foi criada para armazenar o conjunto de informações que serão exibidas no txtInformacoes.

```
resp = "Nome: " + txtNome.Text + "\r\n";
```

A variável resp está recebendo (armazenando) a texto fixo “Nome: ” mais o nome digitado pelo usuário que ficou armazenado no txtNome.Text mais o comando “\r\n” que tem a função de pular de linha.

```
resp = resp + "Turma: " + cmbxTurma.Text + "\r\n";
```

Como o programa deve armazenar todas as informações e exibir no txtInformacoes. Para que isto ocorra perfeitamente precisamos ir acumulando os valores na variável resp, caso contrário as informações seriam sobrescritas.

Para acumular valores usamos o código resp = resp mais texto fixo “Turma: ” mais o item selecionado no cmbxTurma que ficou armazenado no cmbxTurma.Text mais o comando “\r\n”.

Componentes de Seleção – C#

```
if (chkbxInfo.Checked == true)
{
    resp = resp + "Curso: Informática \r\n";
}
```

Para verificarmos qual o curso escolhido pelo usuário, usamos o comando if e a propriedade CHECKED, caso esta propriedade seja verdadeira (**TRUE**) significa que o item foi selecionado, isto é, curso escolhido.

Tendo o curso escolhido acumulamos novamente os valores na variável resp e adicionamos o texto fixo “Curso: Informática \r\n”.

Usamos a mesma análise para os outros 3 if's.

```
txtInformacoes.Text = resp;
```

Por último armazenamos no txtInformacoes.Text a variável resp, como a variável resp é String e a propriedade TEXT também não é necessária a conversão.

Profa. Ana Paula Citro Fugarra Rodrigues

Prof. Ronaldo Vaqueli de Paula