



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Engenharia Elétrica
Laboratório de Engenharia Biomédica



Minicurso de C# e Lógica de Programação



Andrei Nakagawa, MSc

12/09/2016

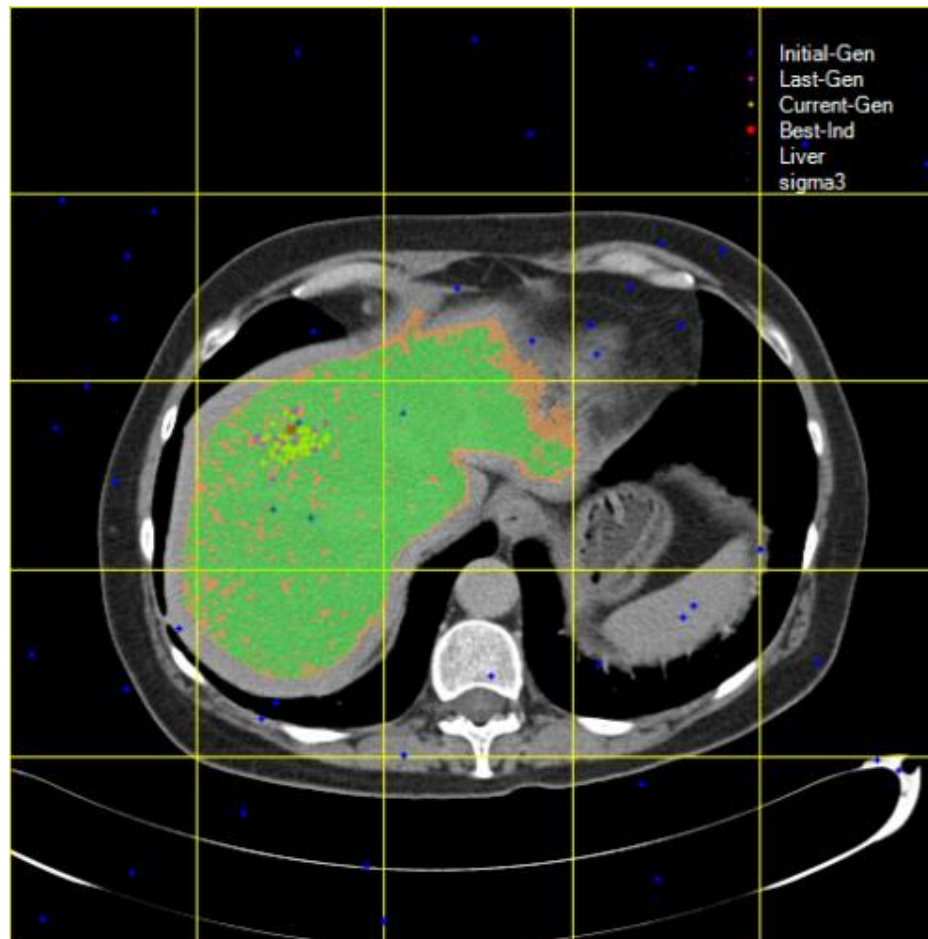
Programação na Eng. Biomédica

- Por que não gostar de programação?



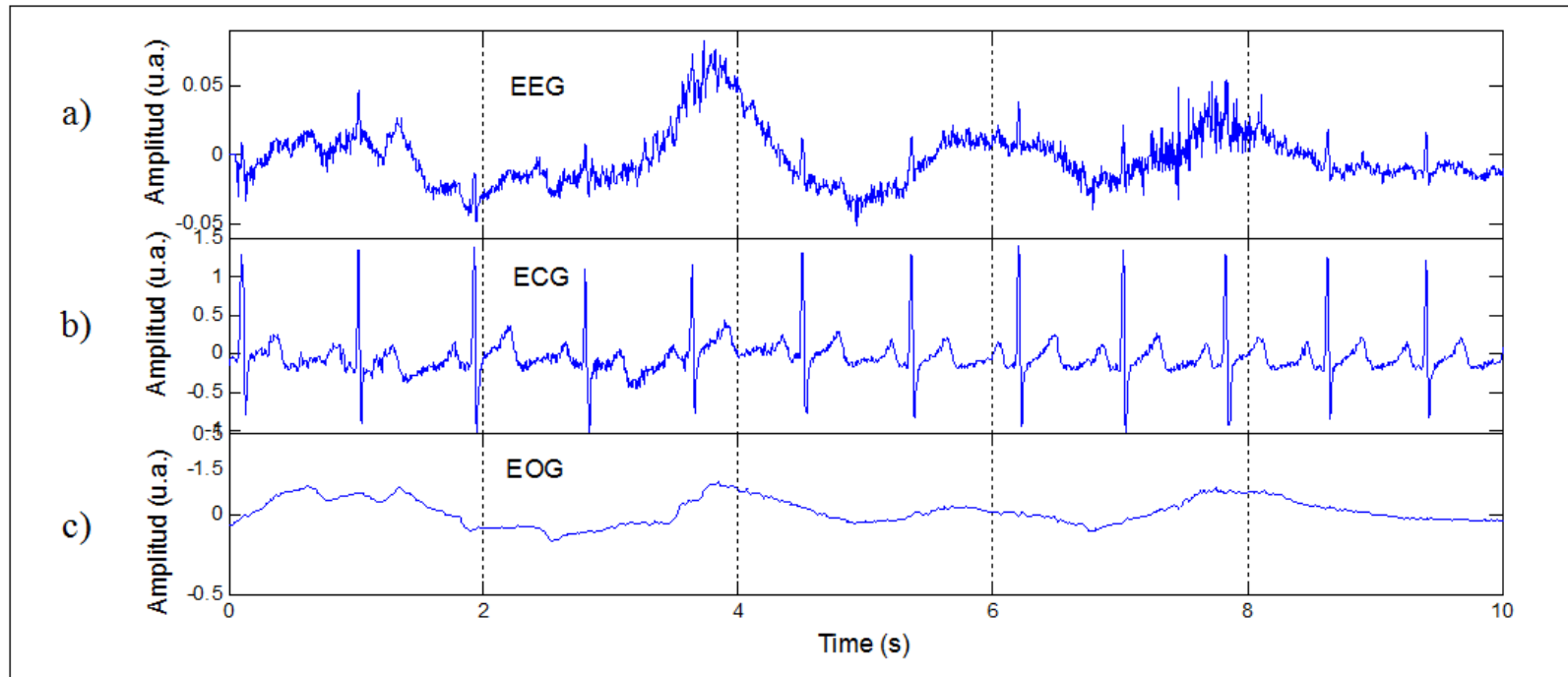
Programação na Eng. Biomédica

- Processamento de imagens médicas



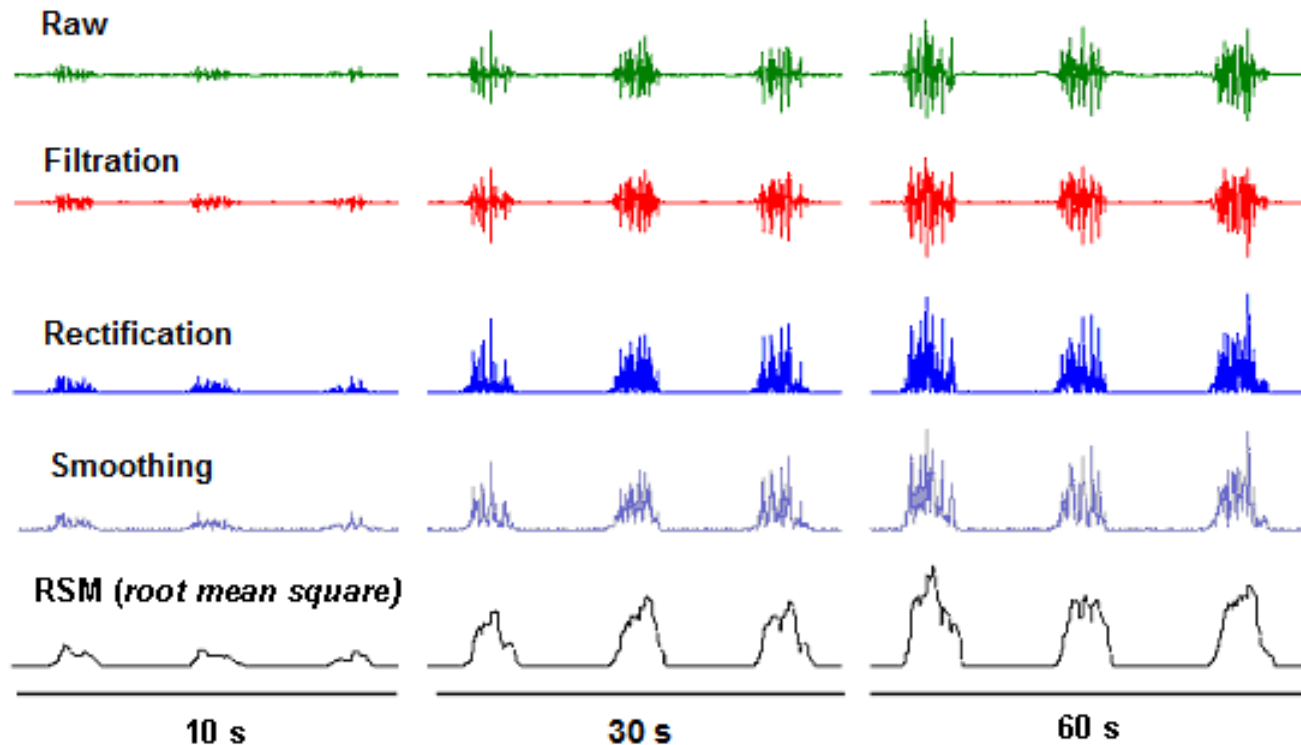
Programação na Eng. Biomédica

- Processamento de sinais biomédicos



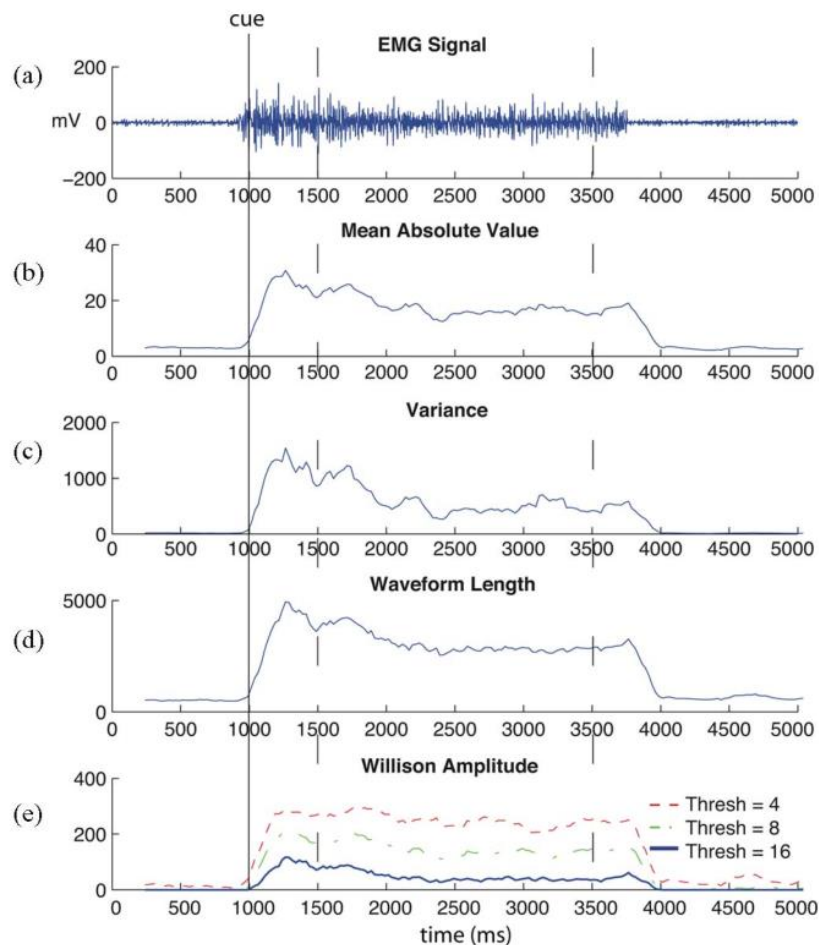
Programação na Eng. Biomédica

- Processamento de sinais biomédicos



Programação na Eng. Biomédica

- Processamento de sinais biomédicos



Programação na Eng. Biomédica

- Neuroengenharia e Engenharia de Reabilitação

SCIENTIFIC REPORTS

OPEN

Long-Term Training with a Brain-Machine Interface-Based Gait Protocol Induces Partial Neurological Recovery in Paraplegic Patients

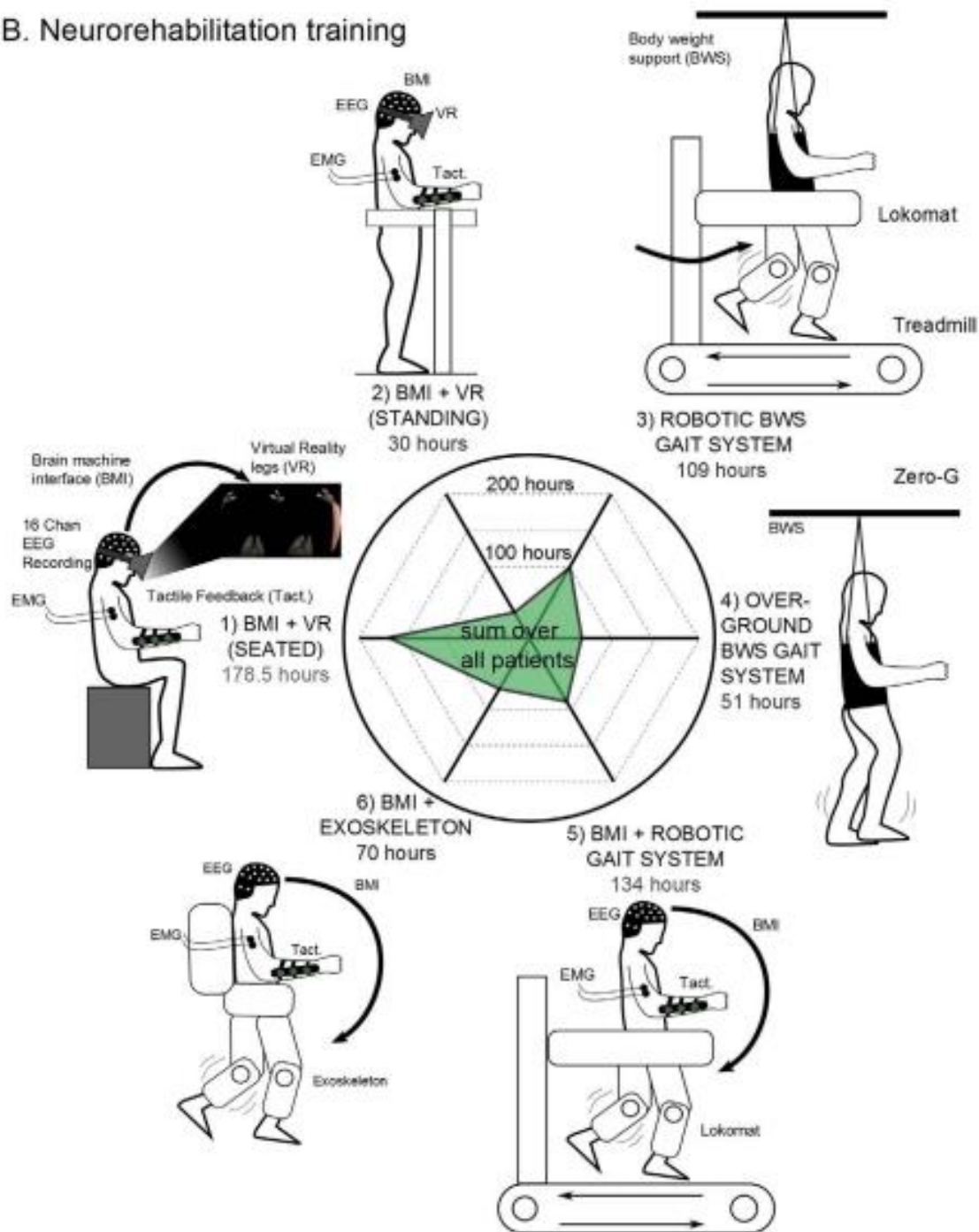
Received: 21 April 2016

Accepted: 04 July 2016

Published: 11 August 2016

Ana R. C. Donati^{1,2}, Solaiman Shokur¹, Edgard Morya^{3,4}, Debora S. F. Campos^{1,2}, Renan C. Moiolli^{3,4}, Claudia M. Gitti^{1,2}, Patricia B. Augusto^{1,2}, Sandra Tripodi^{1,2}, Cristhiane G. Pires^{1,2}, Gislaine A. Pereira^{1,2}, Fabricio L. Brasil^{3,4}, Simone Gallo⁵, Anthony A. Lin^{1,6}, Angelo K. Takigami¹, Maria A. Aratanha³, Sanjay Joshi⁷, Hannes Bleuler⁵, Gordon Cheng⁸, Alan Rudolph^{6,9} & Miguel A. L. Nicolelis^{1,3,6,10,11,12}

B. Neurorehabilitation training



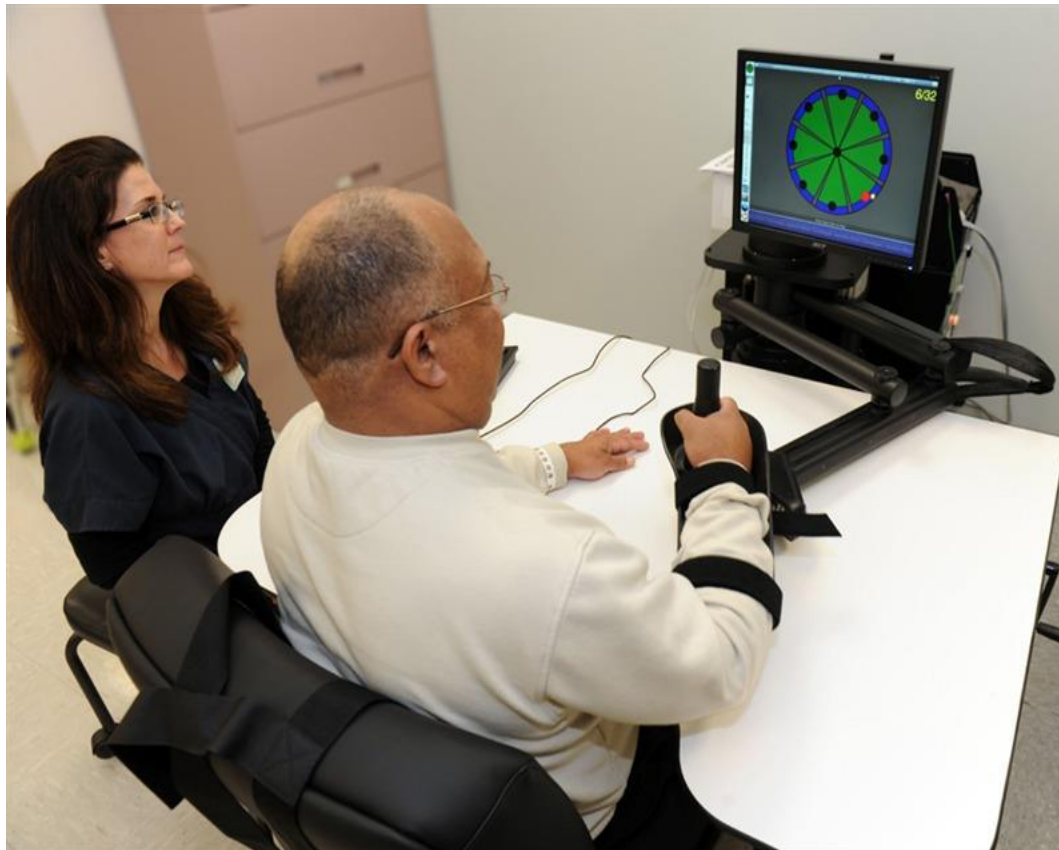
Programação na Eng. Biomédica

- Neuroengenharia e Engenharia de Reabilitação



Programação na Eng. Biomédica

- Engenharia de reabilitação



Programação na Eng. Biomédica

- Engenharia Clínica

Nova Ordem de Serviço para Cliente

☒ Cliente Cadastrado ☐ Novo Cliente


Cliente: 0020 CLIENTE DEMONSTRAÇÃO
Setor: 0020DOMI DOMICÍLIO

Equipamento:
N. Série: 0020DOMI0005 Patrimônio: TAG: 0020DOMI0005
Descrição: 475 NOTEBOOK (475)
Modelo: WRS - 315 Fabricante: ACER

Serviço:
Abertura: 14/04/2014 08:00 Data/Hora de parada: Entrada do Equipamento:
Oficina: TI INFORMÁTICA Tipo Manutenção: 1 CORRETIVA Prioridade: MÉDIA
Responsável: BARNEY BRANDÃO Defeito: BATERIA RUIM
Plano Manut.: <Nenhum> Reclamante: Nº Chamado:
Observação:

Histórico **Garantias / Contratos**
DE CONTRATO: MANUTENÇÃO DE CARRO DE ANESTESIA COM PEÇAS - DE: 01/10/2013 ATÉ 30/09/2014

Contrato de Cliente: 33 CLIENTE DEMONSTRAÇÃO - MANUTENÇÃO DE CARRO DE ANESTESIA COM PEÇAS (01/10/2013 à 30/09/2014)

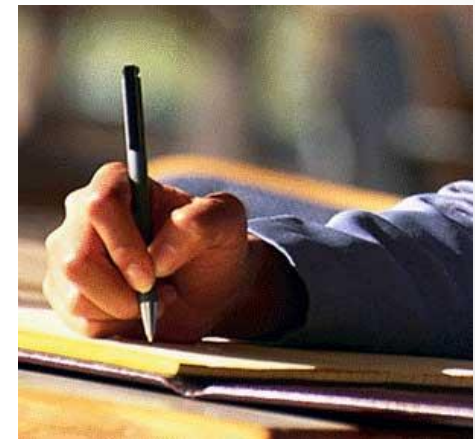
 Abrir OS

Introdução

- Linguagem Orientada a Objetos;
- Baseada no C++;
- Variedade de classes que facilitam a programação;
- Suporte para *asp.net*, *Silverlight*, *Web Services*, *SQL Server*.

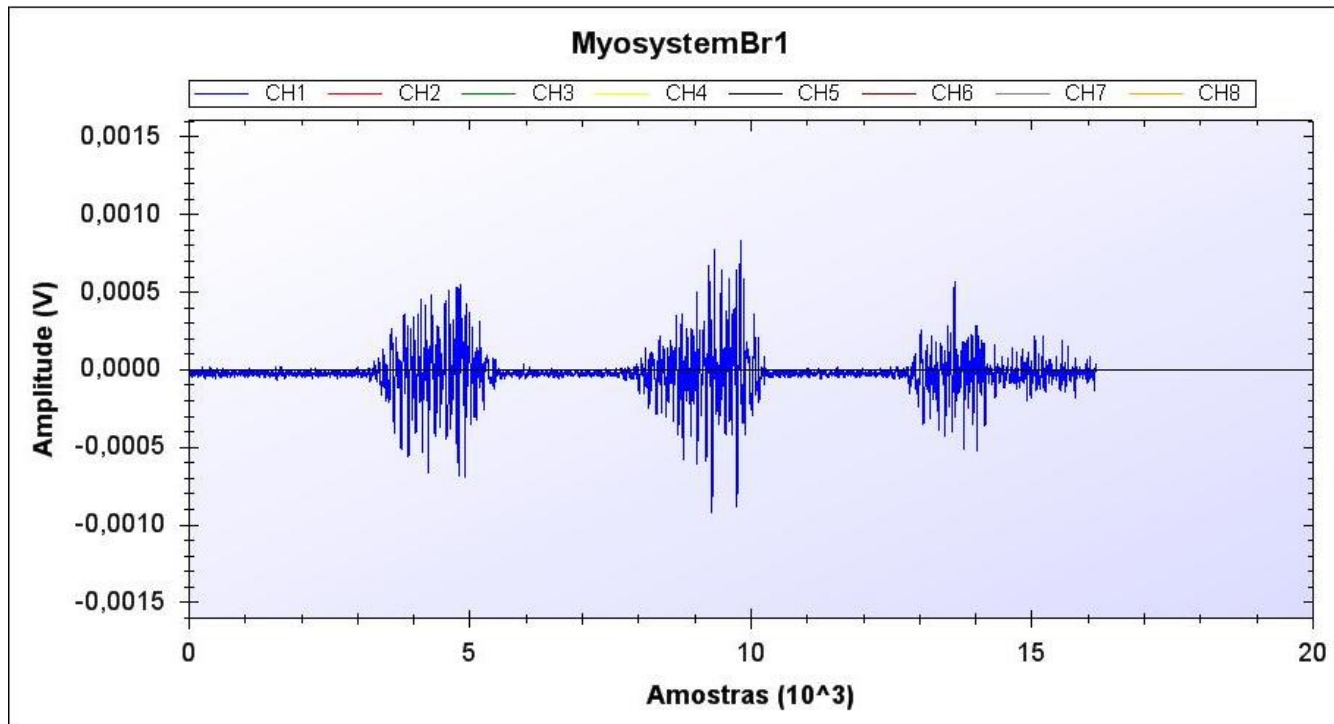
Sobre o minicurso

- Ensino baseado no desenvolvimento de um projeto de *software*
- Aplicação de conceitos teóricos
- **Boas práticas de programação**



Sobre o projeto

- Desenvolveremos juntos um programa para visualização e processamento de sinais EMG



Cronograma

- Aula 01
 - Windows Forms;
 - Construção de interfaces gráficas (GUI)
 - Controles
 - Propriedades e Eventos
- Aula 02
 - Programação orientada a objetos (POO)
 - Modificadores de acesso
 - Criando classes e utilizando objetos



Cronograma

- Aula 03
 - Manipulação de arquivos (leitura e escrita)
 - Manipulação de gráficos (MS Chart)
- Aulas 04 e 05
 - Princípios básicos de sinais biomédicos
 - Desenvolvimento de funções
 - Desenvolvimento do projeto
 - Simulação de eventos em tempo real





UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Engenharia Elétrica
Laboratório de Engenharia Biomédica



Minicurso de C#

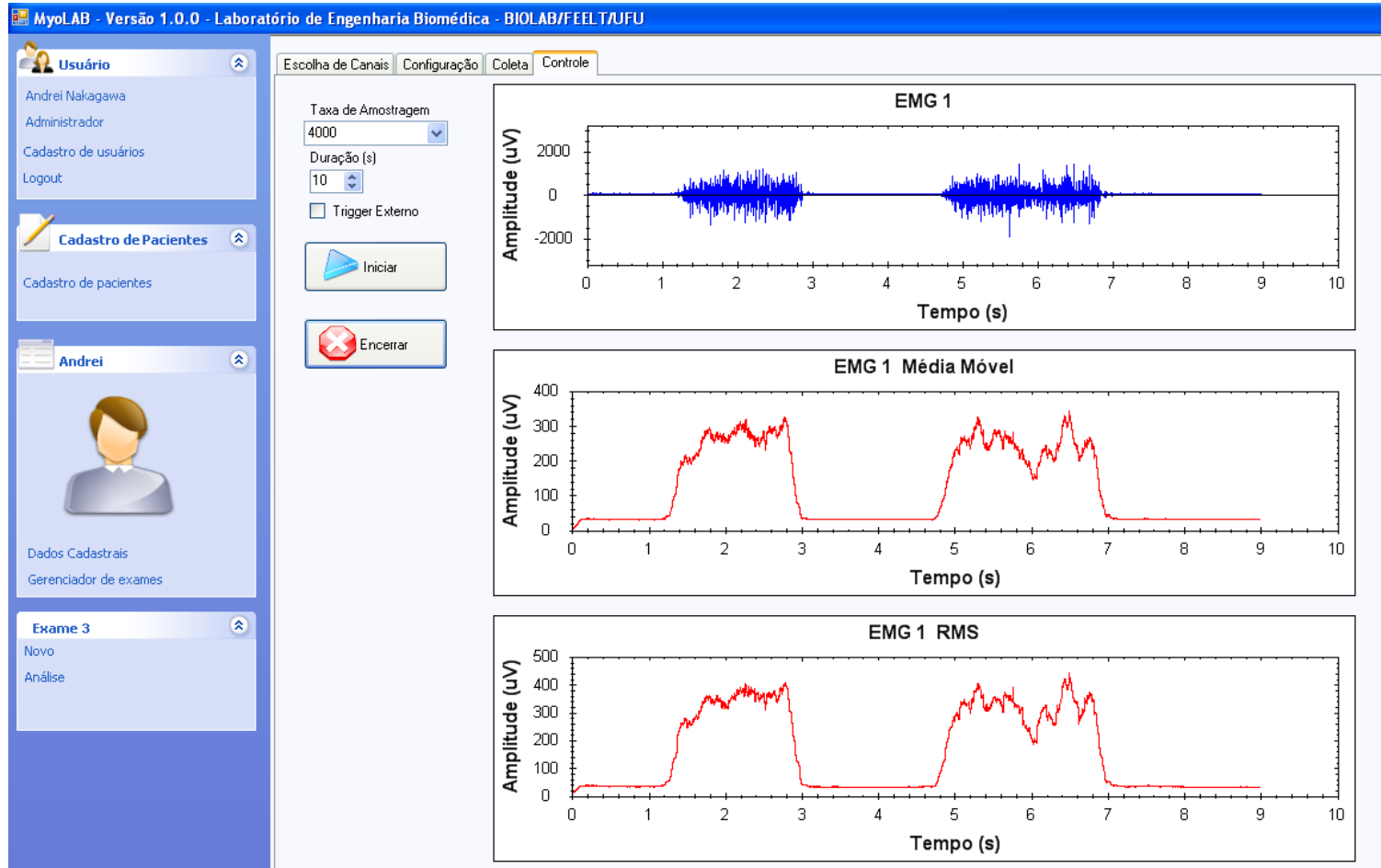
Aula 01



Andrei Nakagawa, MSc

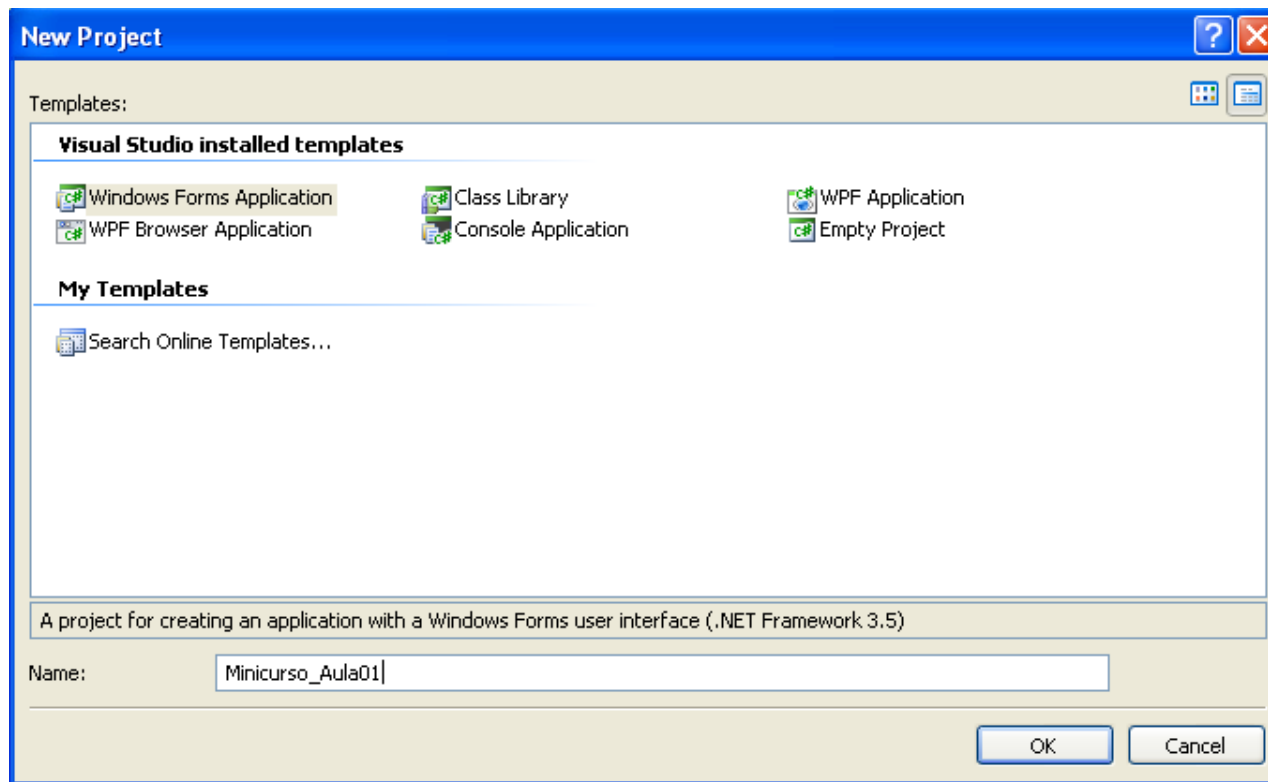
12/09/2016

Windows Forms em C#



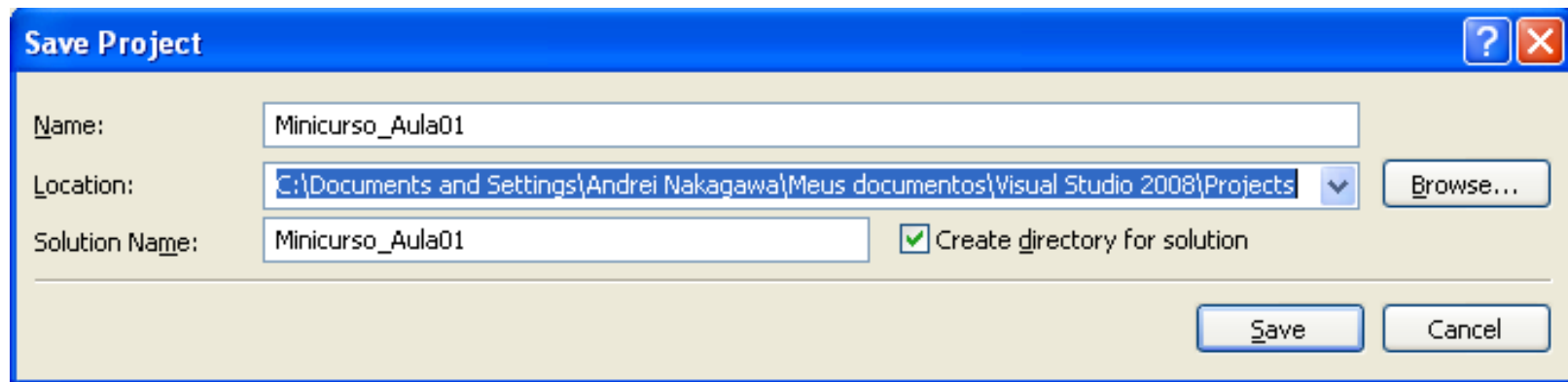
Criando um novo projeto

- File -> New Project

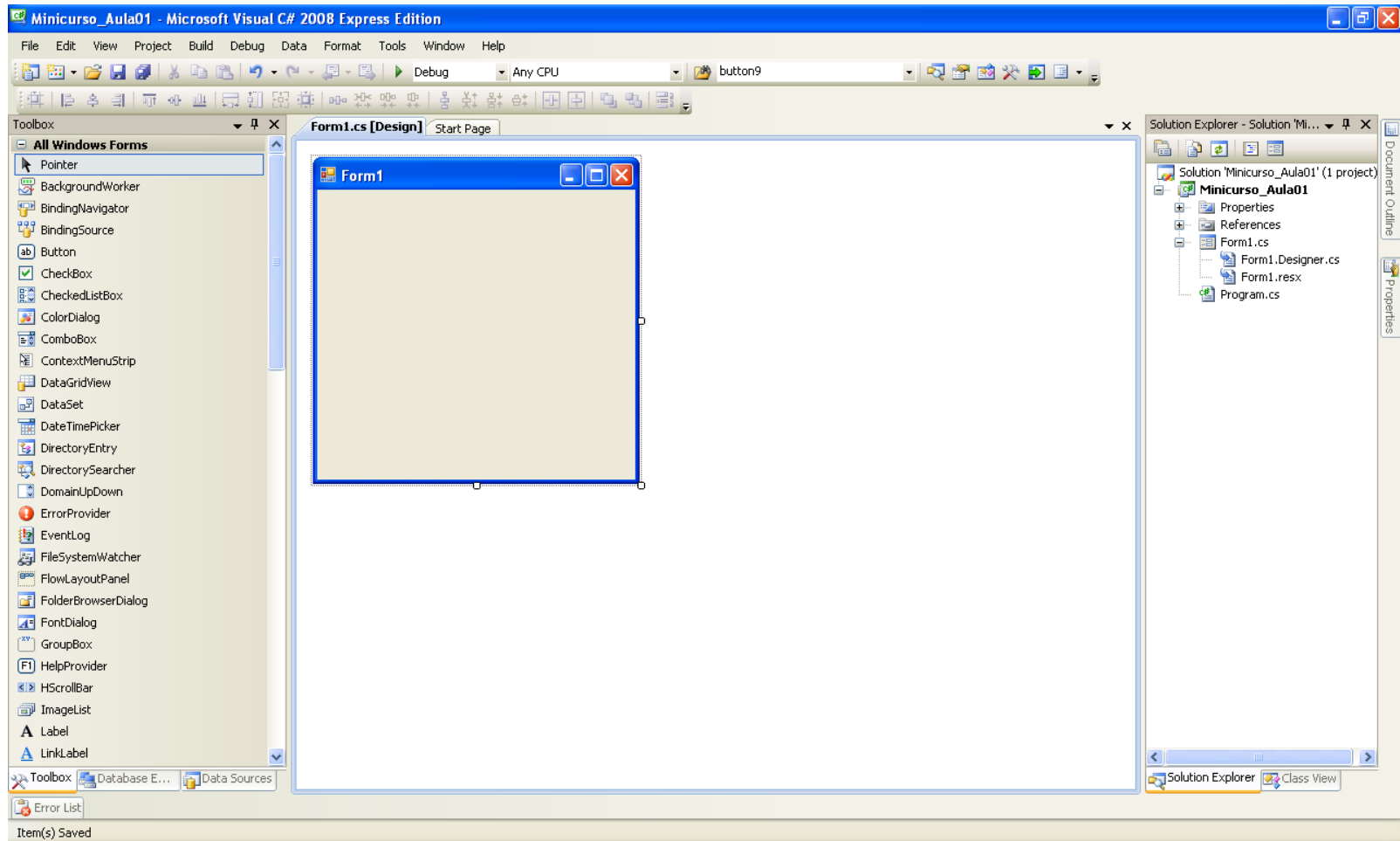


Salvando o Projeto

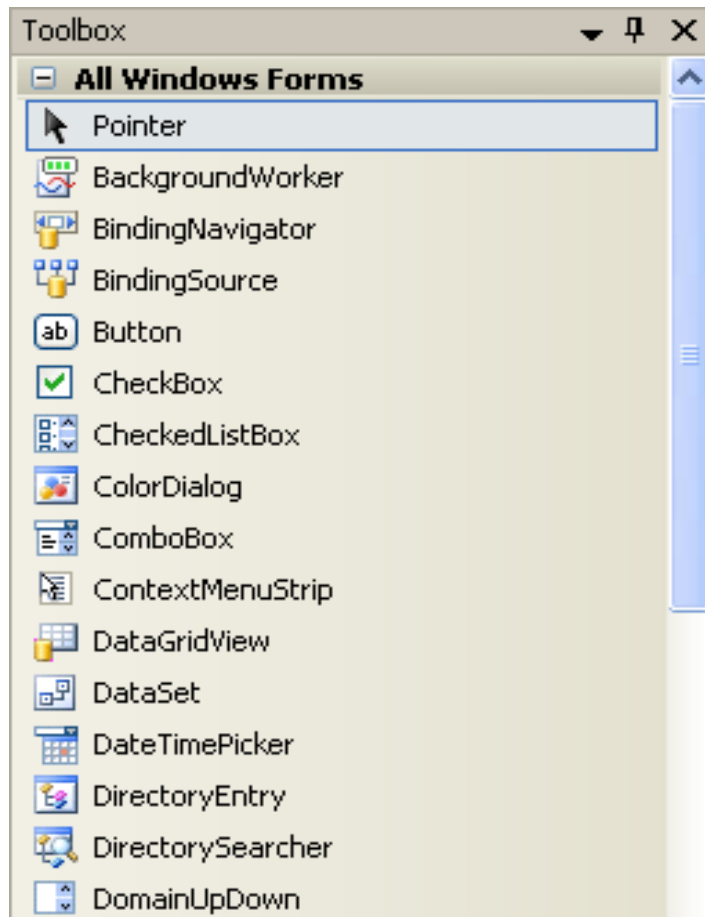
- File -> Save All



Ambiente de desenvolvimento

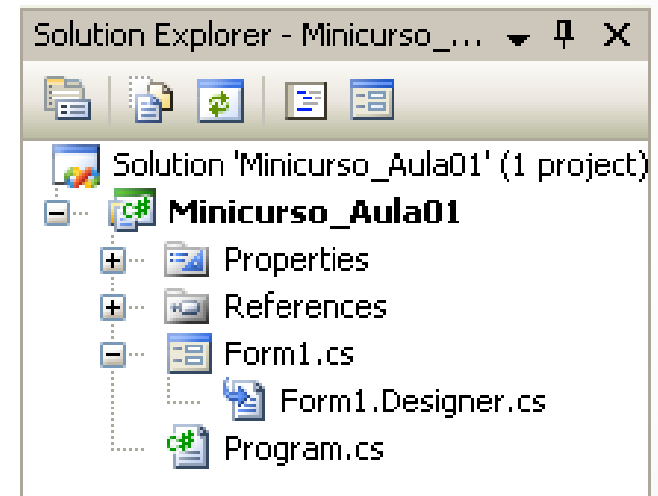
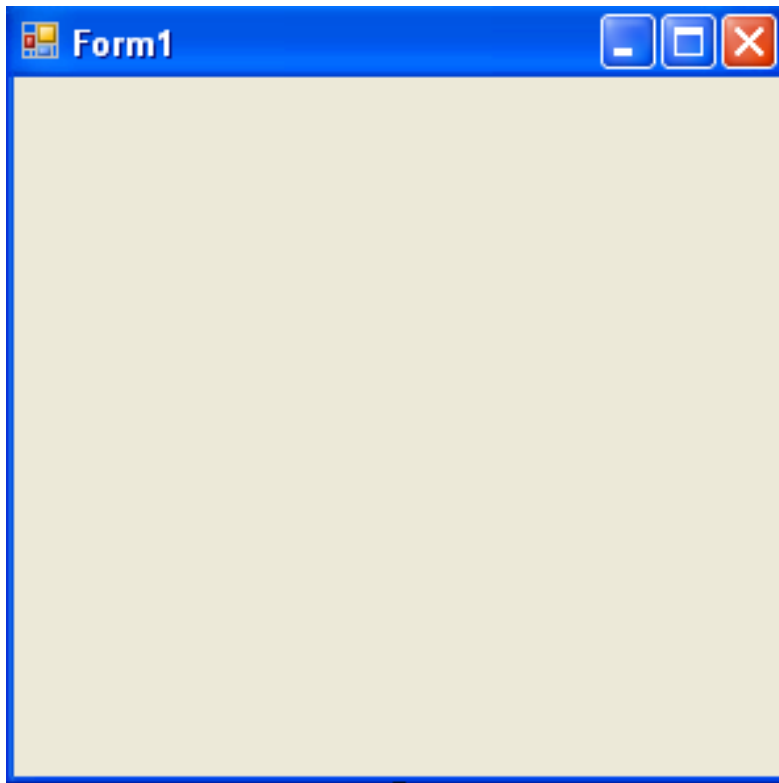


Construindo GUI's



- Em C# com Windows Forms:
 - Estático: *drag and drop (Toolbox)*
 - Dinâmico: Codificação

Formulários

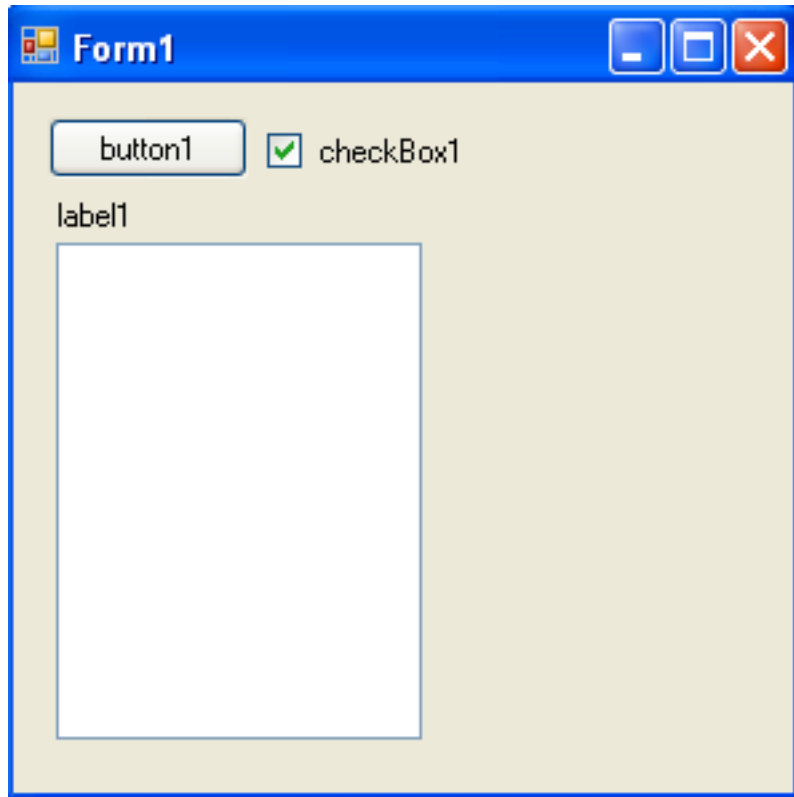


Formulários - Propriedades



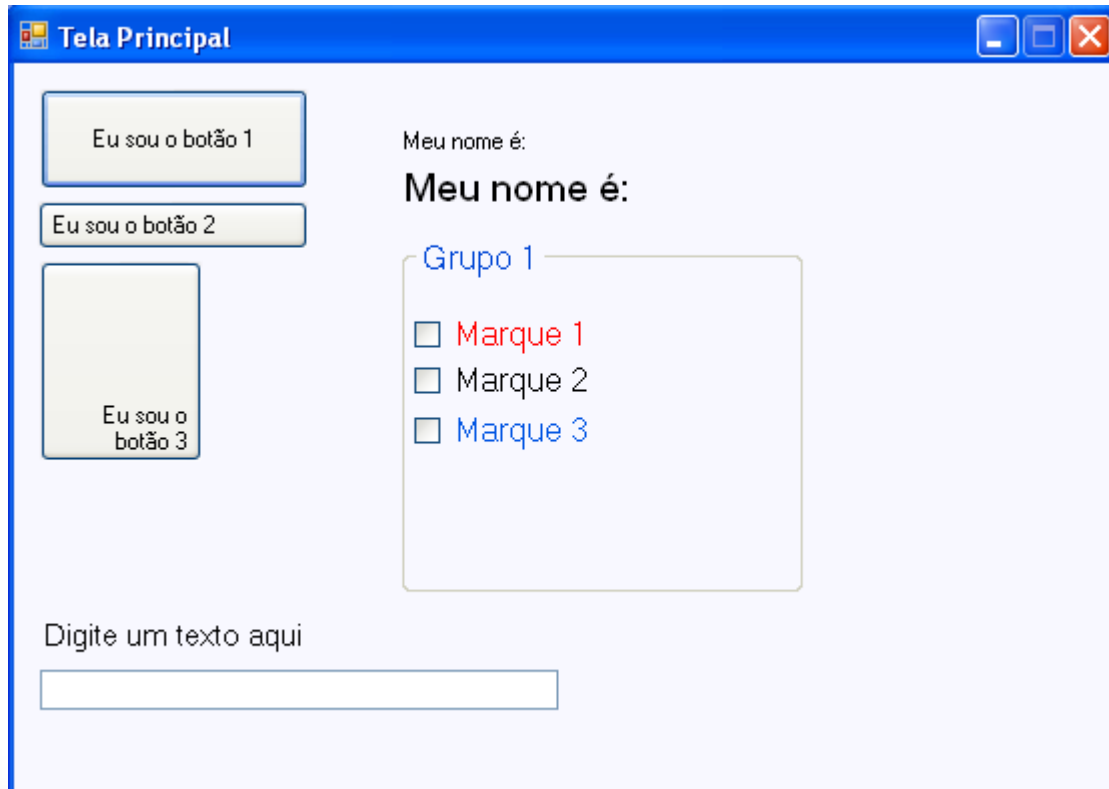
Properties	
Form1 System.Windows.Forms.Form	
⊕ (ApplicationSettings)	
⊕ (DataBindings)	
(Name)	Form1
AcceptButton	(none)
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
AutoScaleMode	Font
AutoScroll	False
⊕ AutoScrollMargin	0; 0
⊕ AutoScrollMinSize	0; 0
AutoSize	False
AutoSizeMode	GrowOnly
AutoValidate	EnablePreventFo
BackColor	Control

Controles



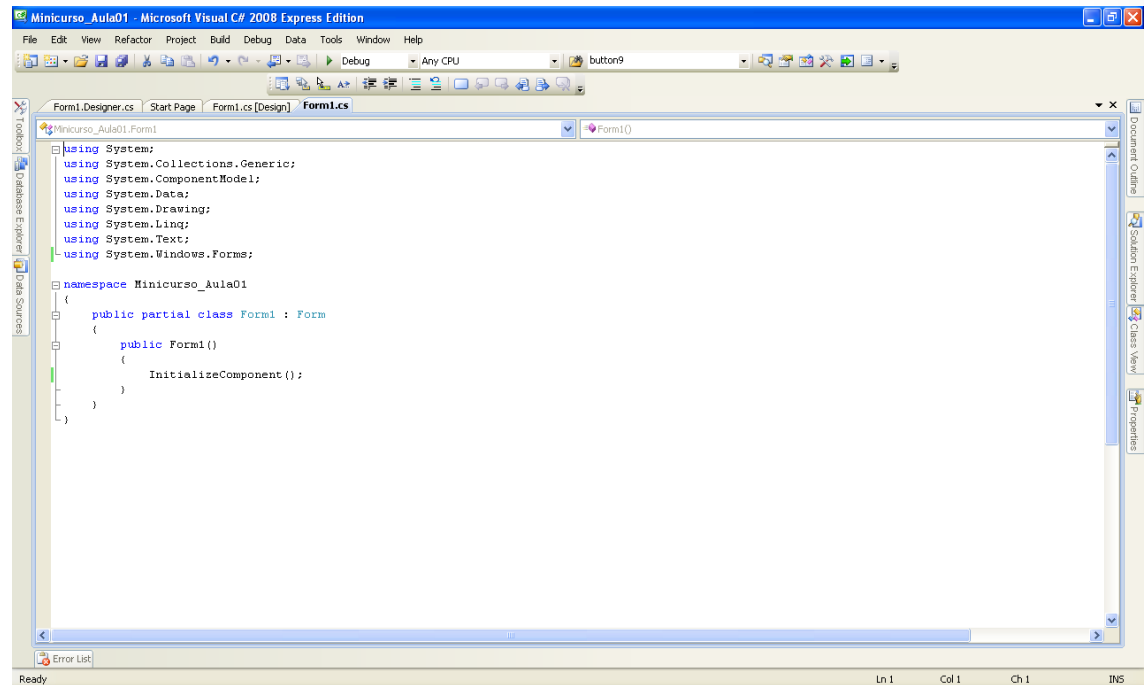
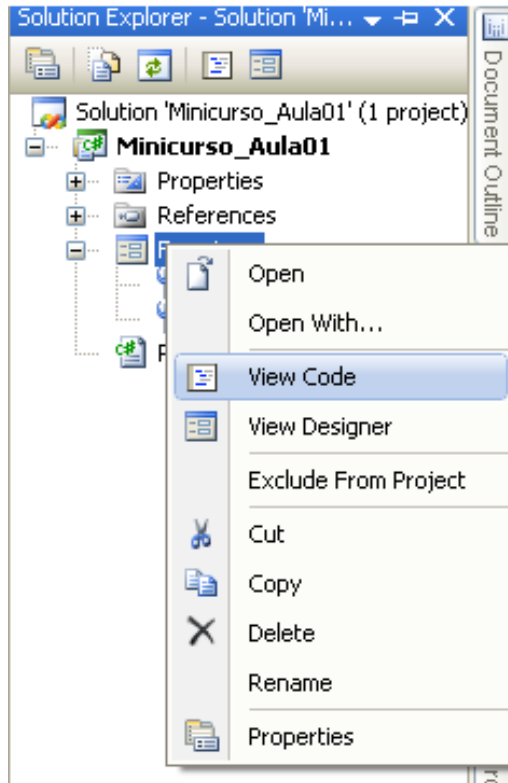
- O *Toolbox* oferece diversos controles para construção de interfaces gráficas;
- *Button, textbox, checkbox, label.*

Playground



- Brinquem um pouco com os controles;
- Criatividade!
- Nesta tela:
 - *Button*
 - *Label*
 - *GroupBox*
 - *TextBox*
 - *CheckBox*

Codificação



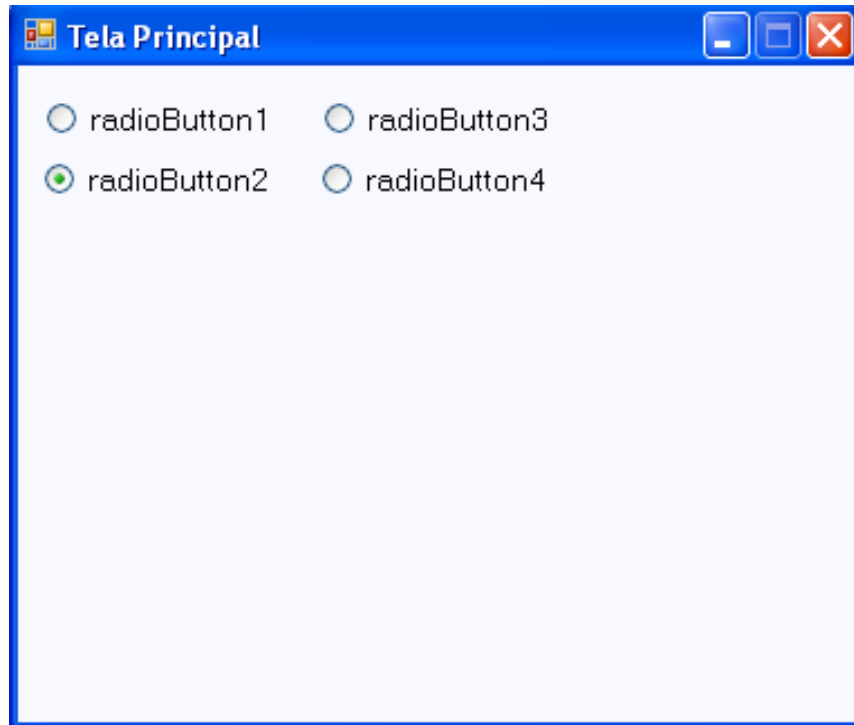
O que acontece?

#region Windows Form Designer generated code

```
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.button1 = new System.Windows.Forms.Button();
    this.button2 = new System.Windows.Forms.Button();
    this.button3 = new System.Windows.Forms.Button();
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.checkBox1 = new System.Windows.Forms.CheckBox();
    this.checkBox2 = new System.Windows.Forms.CheckBox();
    this.checkBox3 = new System.Windows.Forms.CheckBox();
    this.groupBox1 = new System.Windows.Forms.GroupBox();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.label3 = new System.Windows.Forms.Label();
    this.groupBox1.SuspendLayout();
    this.SuspendLayout();
}
```

```
//
// botão1
//
this.botão1.Location = new System.Drawing.Point(13, 13);
this.botão1.Name = "botão1";
this.botão1.Size = new System.Drawing.Size(134, 50);
this.botão1.TabIndex = 0;
this.botão1.Text = "Eu sou o botão 1";
this.botão1.UseVisualStyleBackColor = true;
//
```

RadioButton



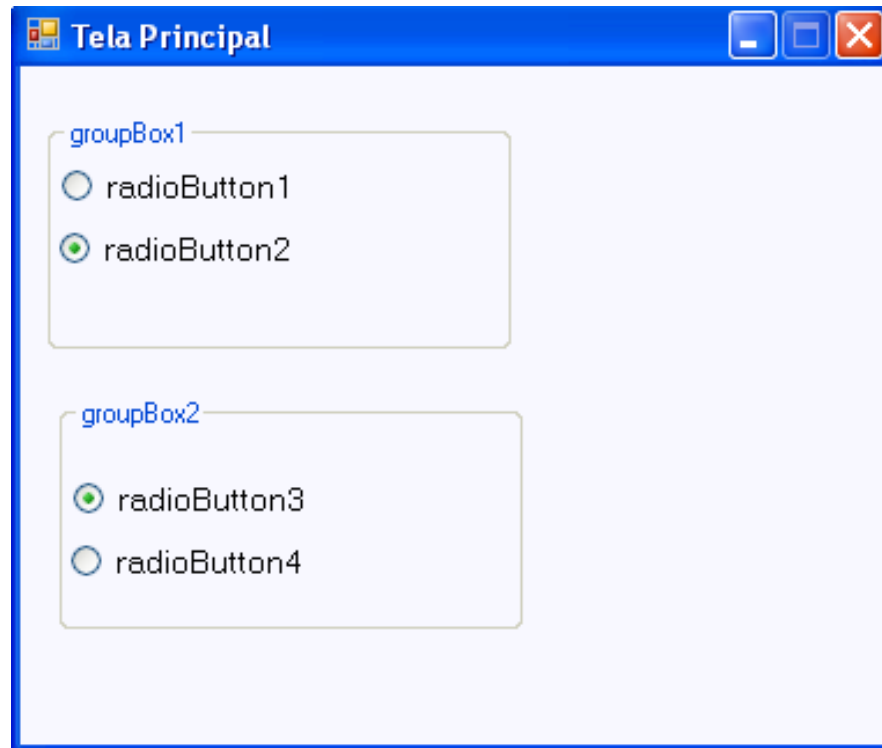
Properties

radioButton1 System.Windows.Forms.RadioButton

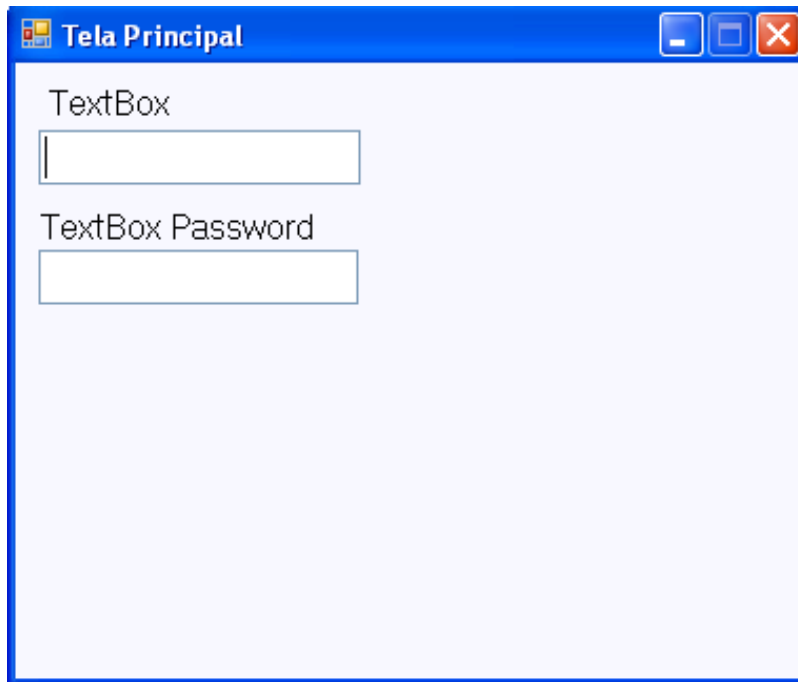
(ApplicationSettings)	
(DataBindings)	
(Name)	radioButton1
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
Appearance	Normal
AutoCheck	True
AutoEllipsis	False
AutoSize	True
BackColor	<input type="checkbox"/> GhostWhite
BackgroundImage	<input type="checkbox"/> (none)
BackgroundImageLayout	Tile
CausesValidation	True
CheckAlign	MiddleLeft
Checked	False
ContextMenuStrip	(none)
Cursor	Default
Dock	None
Enabled	True
FlatAppearance	
FlatStyle	Standard
Font	Microsoft Sans Serif; 10pt
ForeColor	<input type="checkbox"/> ControlText
GenerateMember	True
Image	<input type="checkbox"/> (none)
ImageAlign	MiddleCenter

Checked
Indicates whether the radio button is checked or not.

RadioButton e GroupBox



TextBox



Properties

textBox1 System.Windows.Forms.TextBox

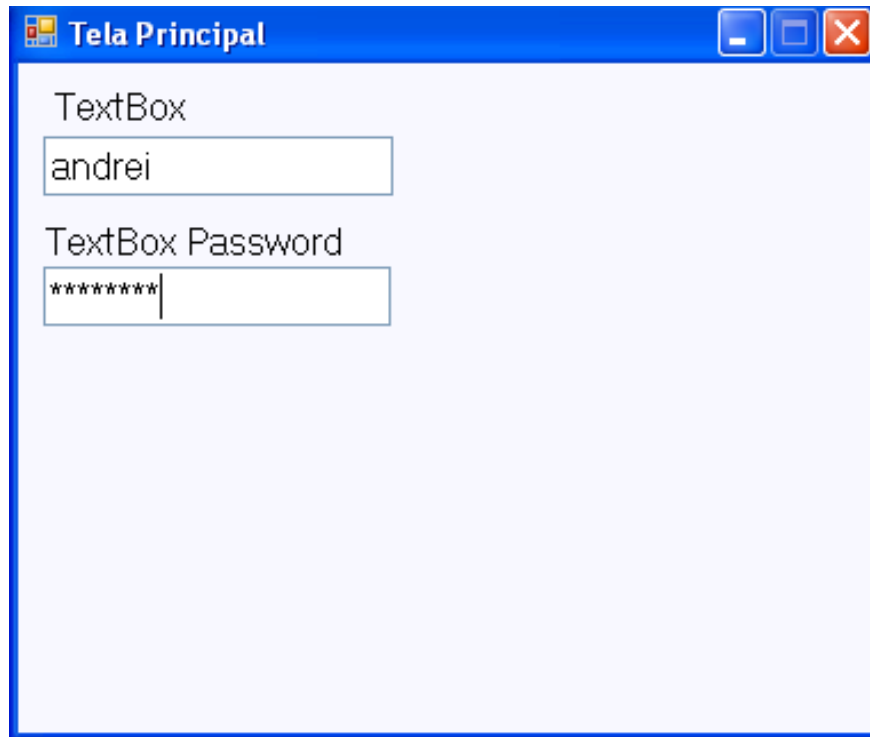
(ApplicationSettings)

(DataBindings)

(Name)	textBox1
AcceptsReturn	False
AcceptsTab	False
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoCompleteCustomSource	(Collection)
AutoCompleteMode	None
AutoCompleteSource	None
BackColor	Window
BorderStyle	Fixed3D
CausesValidation	True
CharacterCasing	Normal
ContextMenuStrip	(none)
Cursor	IBeam
Dock	None
Enabled	True
Font	Microsoft Sans Serif; 8,25pt
ForeColor	WindowText
GenerateMember	True
HideSelection	True
ImeMode	NoControl
Lines	String[] Array
Location	12; 23
Locked	False

TabIndex
Determines the index in the TAB order that this control will occupy.

TextBox



Properties

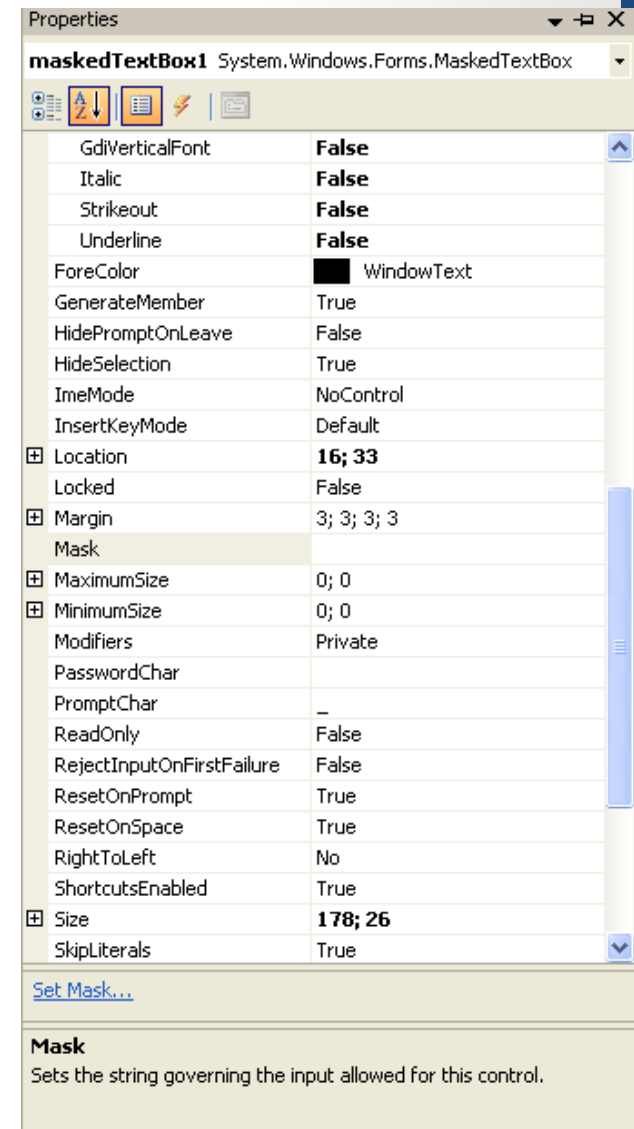
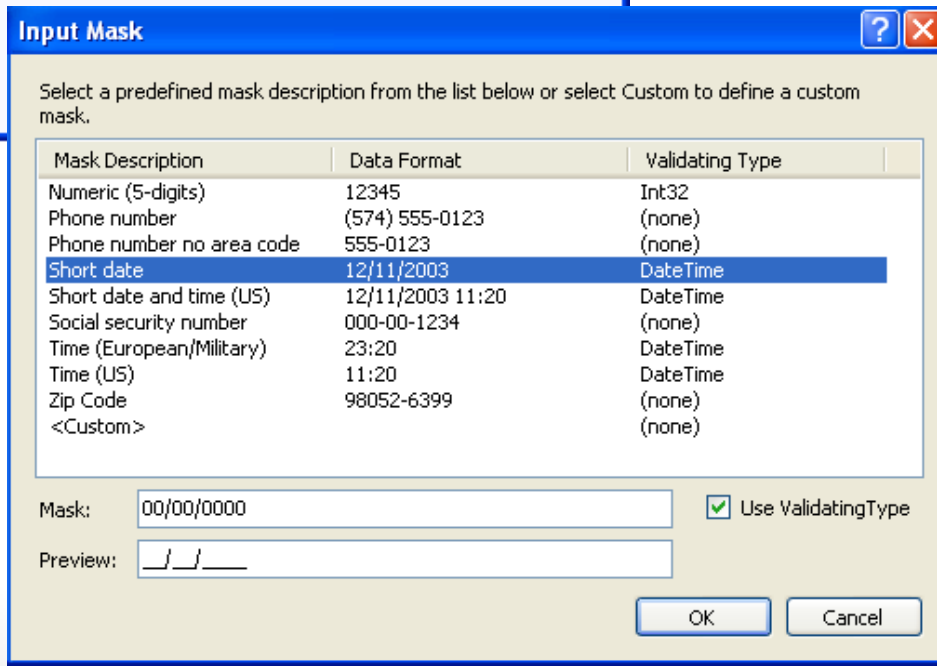
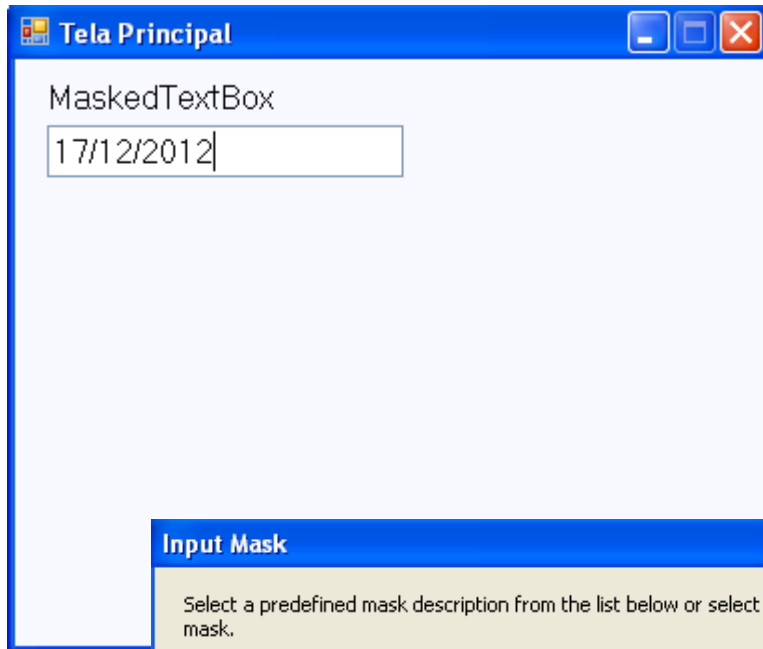
textBox2 System.Windows.Forms.TextBox

GenerateMember	True
HideSelection	True
ImeMode	NoControl
Lines	String[] Array
Location	11; 89
Locked	False
Margin	3; 3; 3; 3
MaximumSize	0; 0
MaxLength	32767
MinimumSize	0; 0
Modifiers	Private
Multiline	False
PasswordChar	*
ReadOnly	False
RightToLeft	No
ScrollBars	None
ShortcutsEnabled	True
Size	152; 26
TabIndex	1
TabStop	True
Tag	
Text	
TextAlign	Left
UseSystemPasswordCh	False
UseWaitCursor	False
Visible	True
WordWrap	True

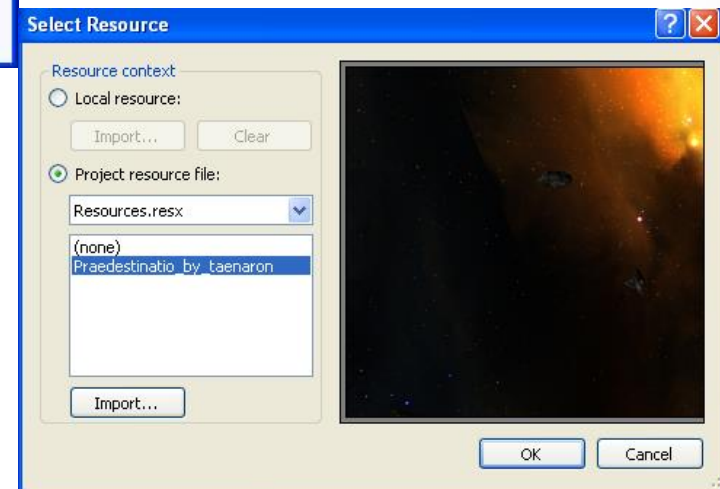
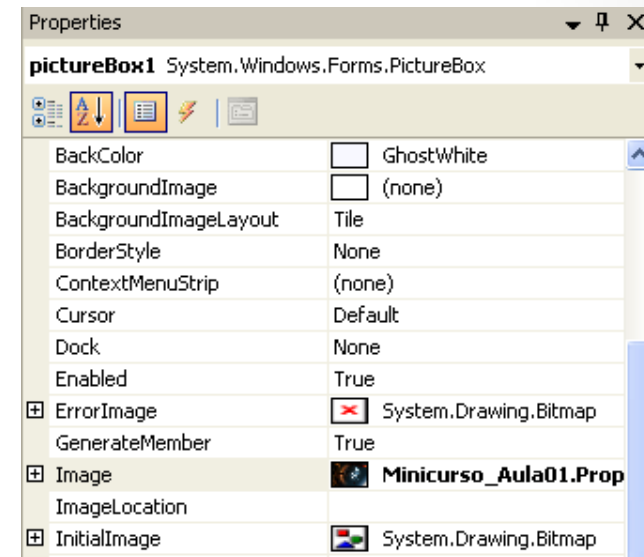
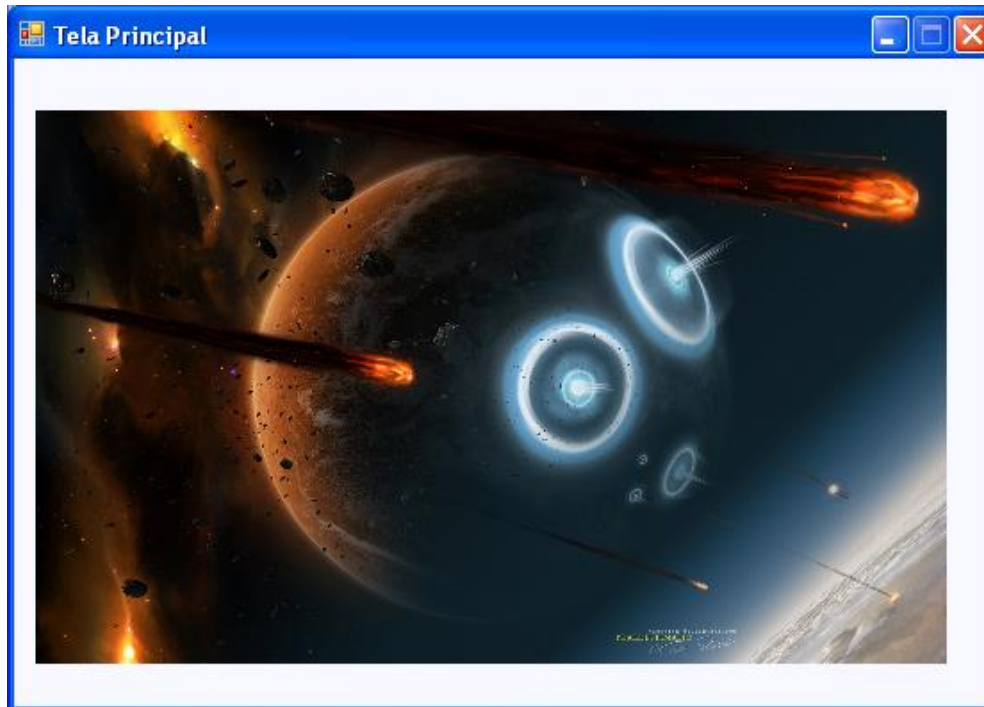
PasswordChar

Indicates the character to display for password input for single-line edit controls.

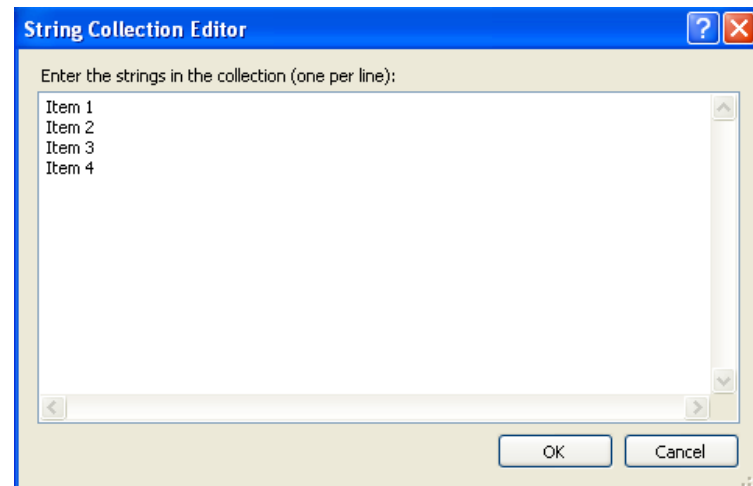
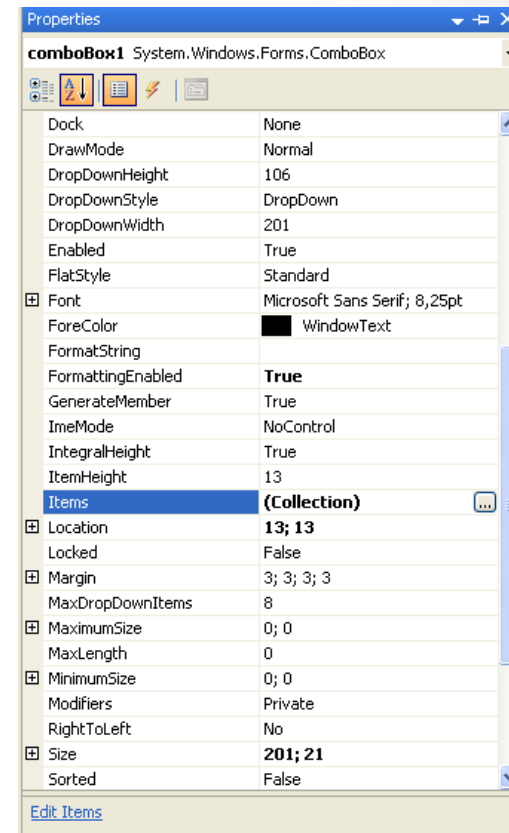
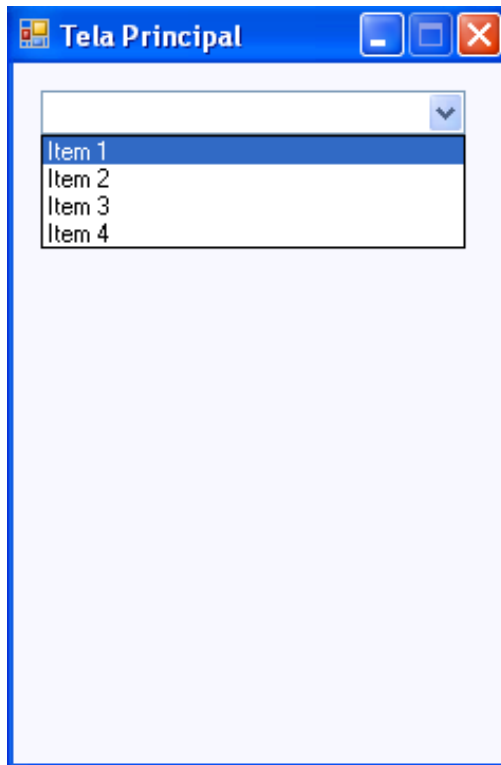
MaskedTextBox



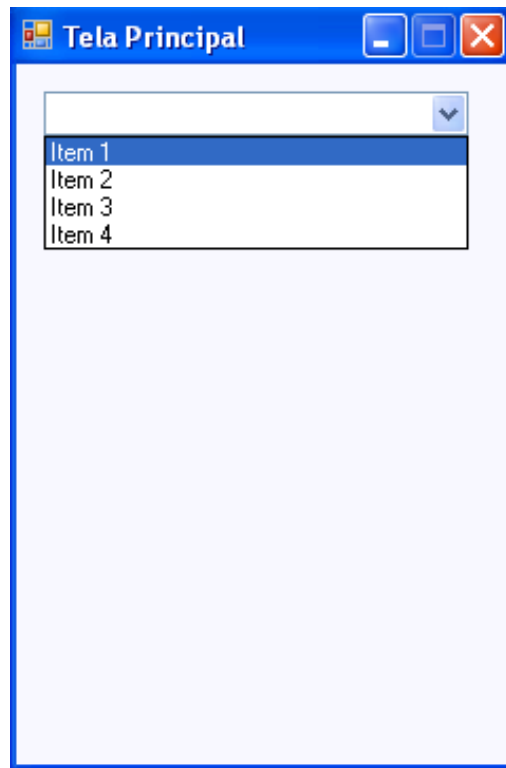
PictureBox



ComboBox



ComboBox



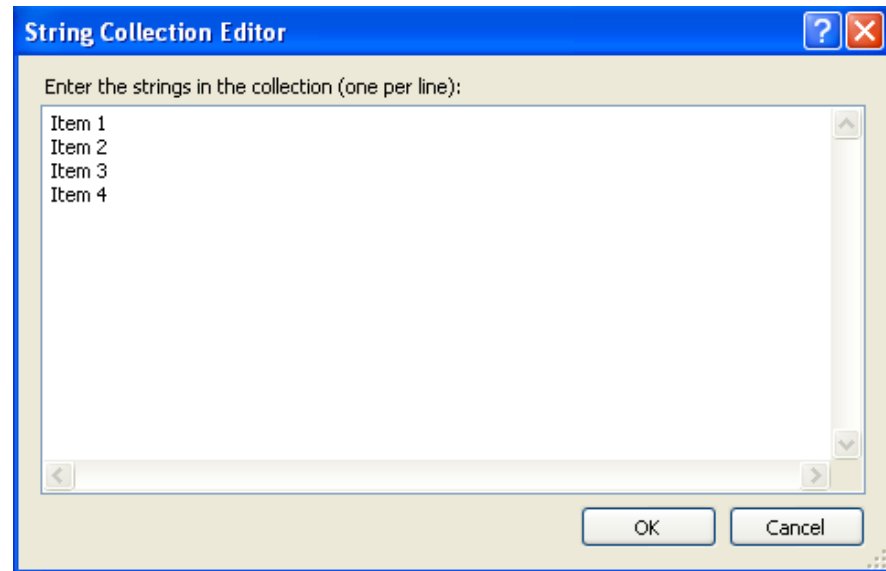
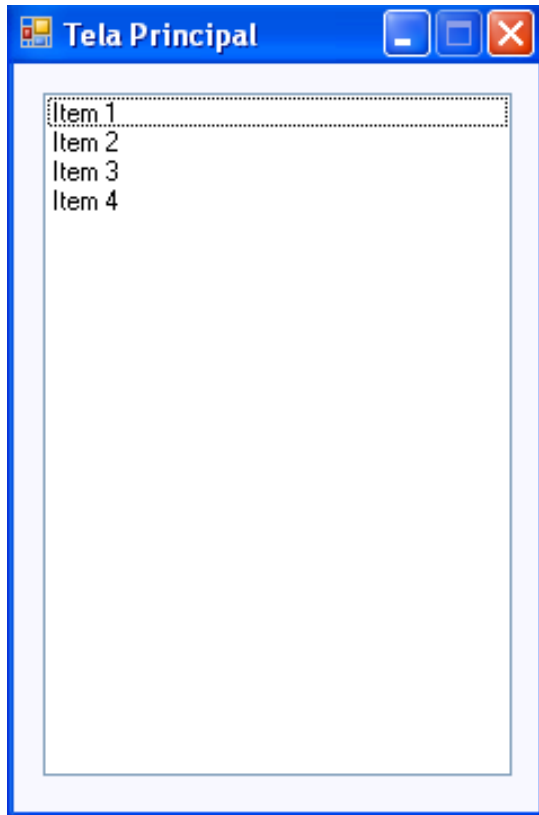
```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            comboBox1.Items.Add("Item 1");
            comboBox1.Items.Add("Item 2");
            comboBox1.Items.Add("Item 3");
            comboBox1.Items.Add("Item 4");
        }
    }
}
```

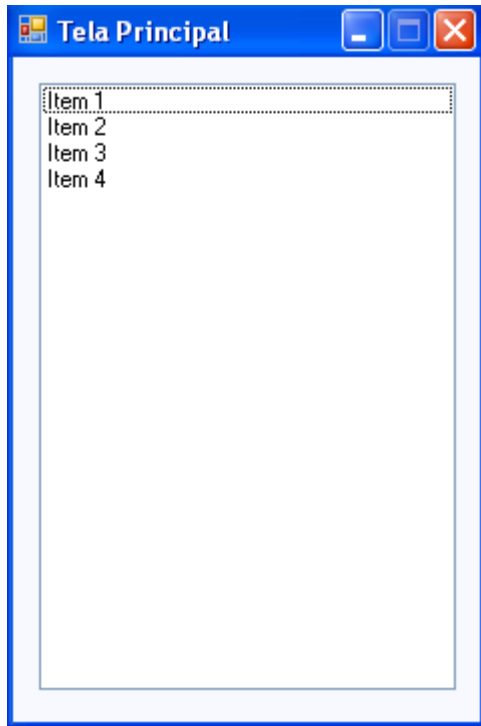
```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            for (int i = 1; i < 5; i++)
            {
                comboBox1.Items.Add("Item " + i);
            }
        }
    }
}
```

ListBox



ListBox



```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

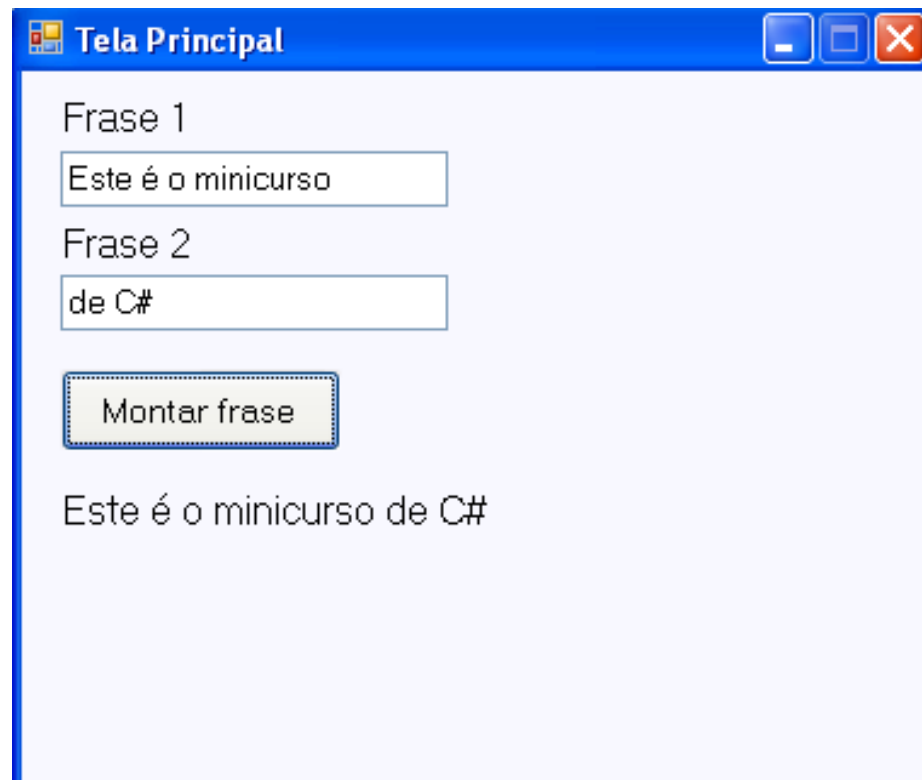
        private void Form1_Load(object sender, EventArgs e)
        {
            listBox1.Items.Add("Item 1");
            listBox1.Items.Add("Item 2");
            listBox1.Items.Add("Item 3");
            listBox1.Items.Add("Item 4");
        }
    }
}

namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            for (int i = 1; i < 5; i++)
            {
                listBox1.Items.Add("Item " + i);
            }
        }
    }
}
```

Exercício 1

- Montem um programa que juntem uma frase a partir de palavras escritas em textbox diferentes



Exercício 1

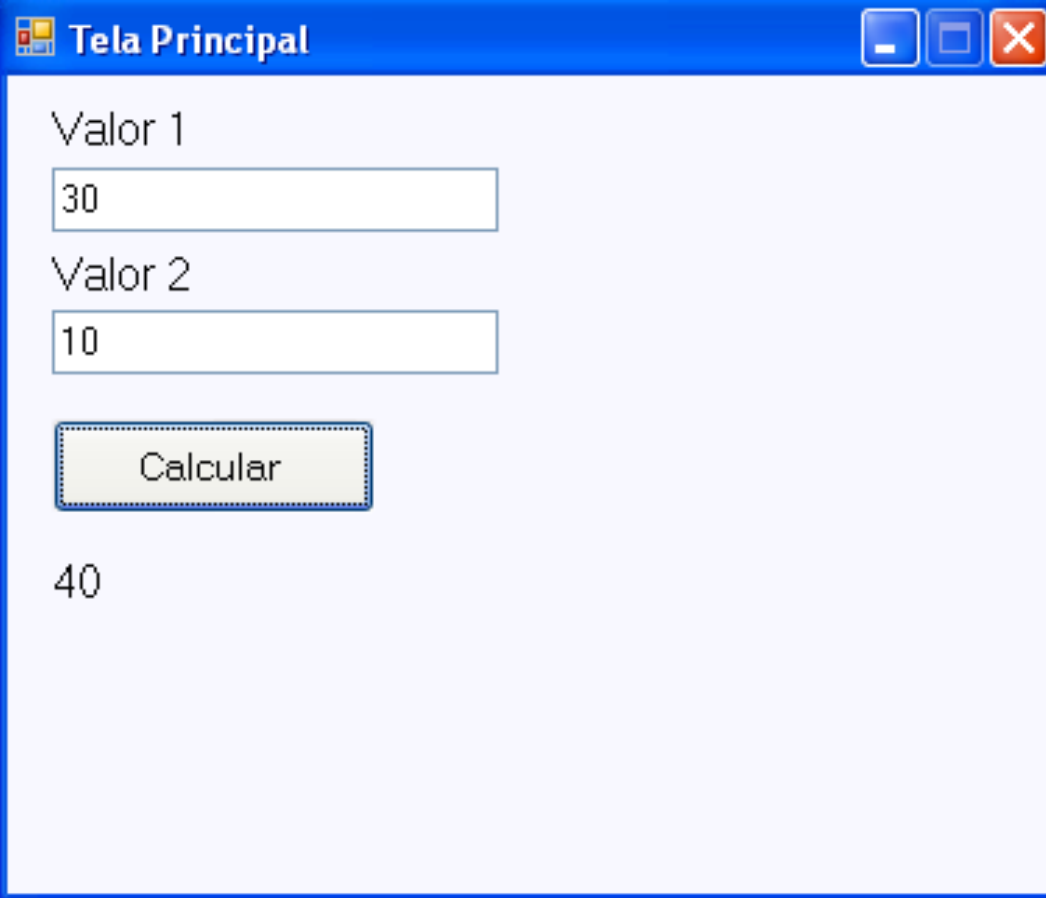
- RESPOSTA!

```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label3.Text = textBox1.Text + " " + textBox2.Text;
        }
    }
}
```


Exercício 2

- Montem um programa que calcule a soma de dois valores



The image shows a screenshot of a Windows application window titled "Tela Principal". The window has a blue title bar with standard minimize, maximize, and close buttons. Inside the window, there are two input fields. The first is labeled "Valor 1" and contains the number "30". The second is labeled "Valor 2" and contains the number "10". Below these fields is a button labeled "Calcular". At the bottom of the window, the result "40" is displayed.

Exercício 2

- RESPOSTA!

```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

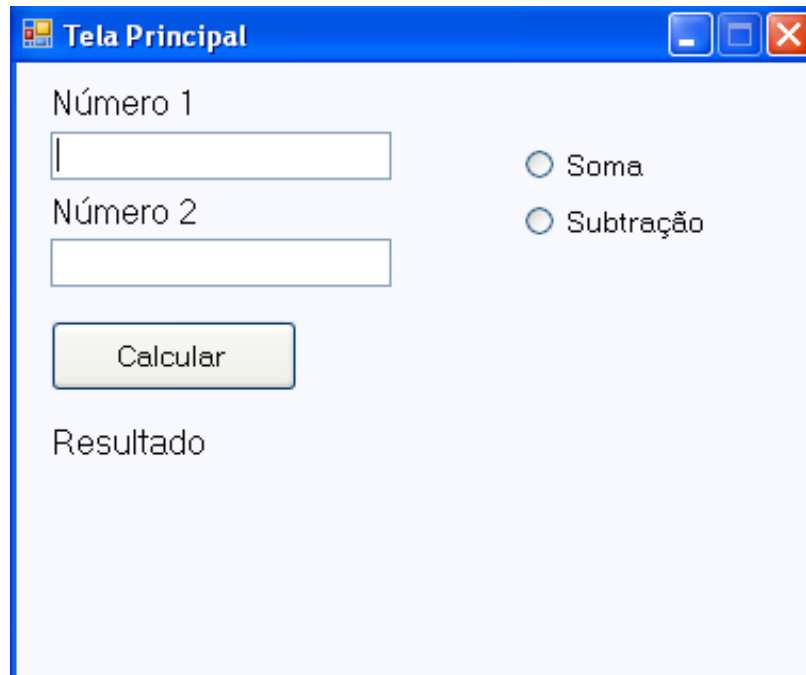
        private void button1_Click(object sender, EventArgs e)
        {
            double soma = 0;

            soma = Convert.ToDouble(textBox1.Text) + Convert.ToDouble(textBox2.Text);

            label3.Text = soma.ToString();
        }
    }
}
```

Exercício 3

- Calculadora
 - Somar e subtrair;
 - Usuário escolhe operação através de RadioButton;
 - Os valores são colocados em dois TextBox;



The screenshot shows a Windows application window titled "Tela Principal". Inside the window, there are two text input fields labeled "Número 1" and "Número 2". To the right of these fields are two radio buttons: "Soma" (selected) and "Subtração". Below the input fields is a button labeled "Calcular". At the bottom of the window is a label "Resultado".

Exercício 3

Tela Principal

Valor 1
30

Valor 2
10

Operações

☒ Soma

☐ Subtração

Calcular

$30 + 10 = 40$

Tela Principal

Valor 1
30

Valor 2
10

Operações

☐ Soma

☒ Subtração

Calcular

$30 - 10 = 20$

Exercício 3

```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked)
            {
                double soma = Convert.ToDouble(textBox1.Text) + Convert.ToDouble(textBox2.Text);
                label3.Text = textBox1.Text + " + " + textBox2.Text + " = " + soma.ToString();
            }
            else if (radioButton2.Checked)
            {
                double subtracao = Convert.ToDouble(textBox1.Text) - Convert.ToDouble(textBox2.Text);
                label3.Text = textBox1.Text + " - " + textBox2.Text + " = " + subtracao.ToString();
            }
            else
            {
                MessageBox.Show("Selecione uma operação!", "ERRO", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

Exercício 3

```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

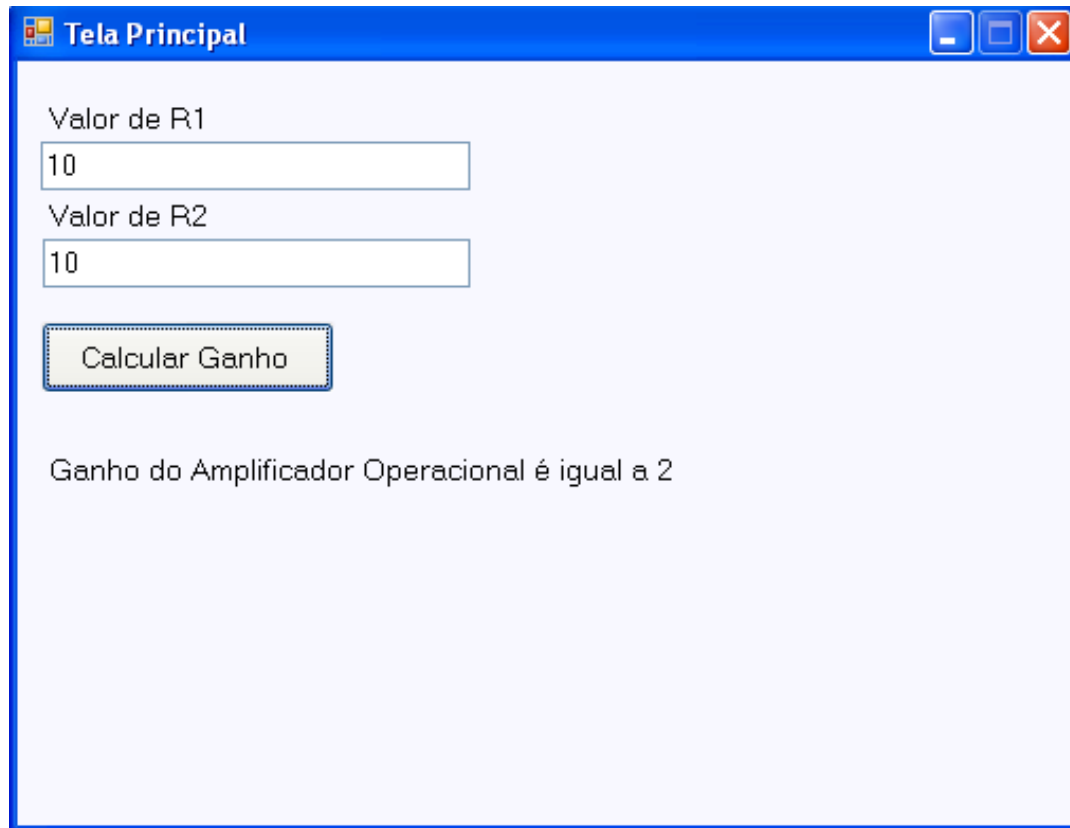
        private void btnCalcular_Click(object sender, EventArgs e)
        {
            if (rbSoma.Checked)
            {
                double soma = Convert.ToDouble(tbValor1.Text) + Convert.ToDouble(tbValor2.Text);
                labelResultado.Text = tbValor1.Text + " + " + tbValor2.Text + " = " + soma.ToString();
            }
            else if (rbSubtracao.Checked)
            {
                double subtracao = Convert.ToDouble(tbValor1.Text) - Convert.ToDouble(tbValor2.Text);
                labelResultado.Text = tbValor1.Text + " - " + tbValor2.Text + " = " + subtracao.ToString();
            }
            else
            {
                MessageBox.Show("Selecione uma operação!", "ERRO", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

Exercício 3

```
//Evento ativado quando o botão "Calcular" é pressionado
private void btnCalcular_Click(object sender, EventArgs e)
{
    //Se a operação escolhida é a soma
    if (rbSoma.Checked)
    {
        //SOMA
        //Variável local
        //Converte o texto dos textBox em valores double e calcula a soma
        double soma = Convert.ToDouble(tbValor1.Text) + Convert.ToDouble(tbValor2.Text);
        //Escreve o resultado no labelResultado
        labelResultado.Text = tbValor1.Text + " + " + tbValor2.Text + " = " + soma.ToString();
    }
    //Se a operação escolhida é subtração
    else if (rbSubtracao.Checked)
    {
        //SUBTRAÇÃO
        //Variável local
        //Converte o texto dos textBox em valores double e calcula a subtração
        double subtracao = Convert.ToDouble(tbValor1.Text) - Convert.ToDouble(tbValor2.Text);
        //Escreve o resultado no labelResultado
        labelResultado.Text = tbValor1.Text + " - " + tbValor2.Text + " = " + subtracao.ToString();
    }
    //Se nenhuma operação for selecionada
    else
    {
        //Mostra uma mensagem de erro na tela
        MessageBox.Show("Selecione uma operação!", "ERRO", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Exercício 4

- Monte um programa que receba o valor de dois resistores que configuram o ganho de um amplificador não – inversor.



Tela Principal

Valor de R1
10

Valor de R2
10

Calcular Ganho

Ganho do Amplificador Operacional é igual a 2

Exercício 4

- RESPOSTA!

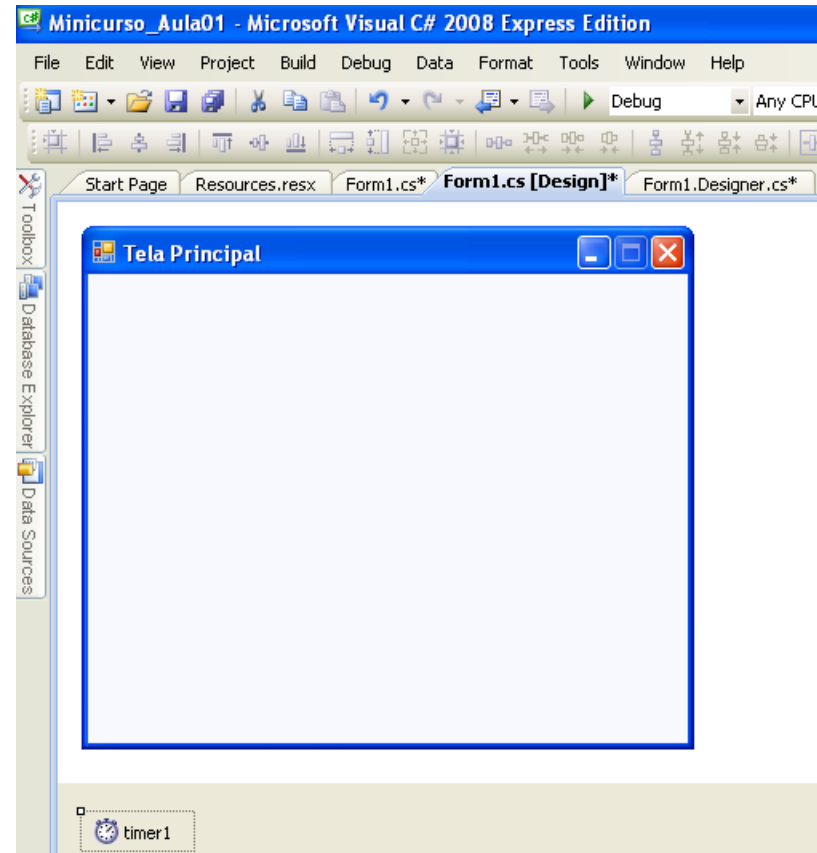
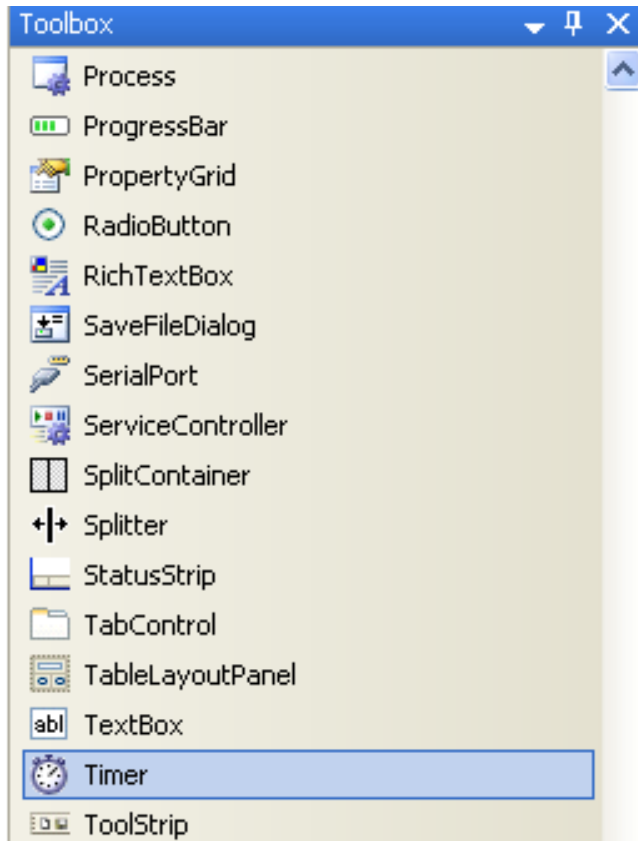
```
namespace Minicurso_Aula01
{
    public partial class TelaPrincipal : Form
    {
        public TelaPrincipal()
        {
            InitializeComponent();
        }

        private void btnCalcularGanho_Click(object sender, EventArgs e)
        {
            //Inicialização da variável local que receberá o valor do ganho
            double ganhoAmplificadorOperacional = 0;

            //O ganho de um AO Não Inversor é dado por:
            //Vout = (R2/R1) + 1
            ganhoAmplificadorOperacional = (Convert.ToDouble(tbR2.Text) / Convert.ToDouble(tbR1.Text)) + 1;

            //Escreve o resultado na tela
            labelResultado.Text = "Ganho do Amplificador Operacional é igual a " + ganhoAmplificadorOperacional.ToString();
        }
    }
}
```

Timer



Timer – Propriedades e Eventos

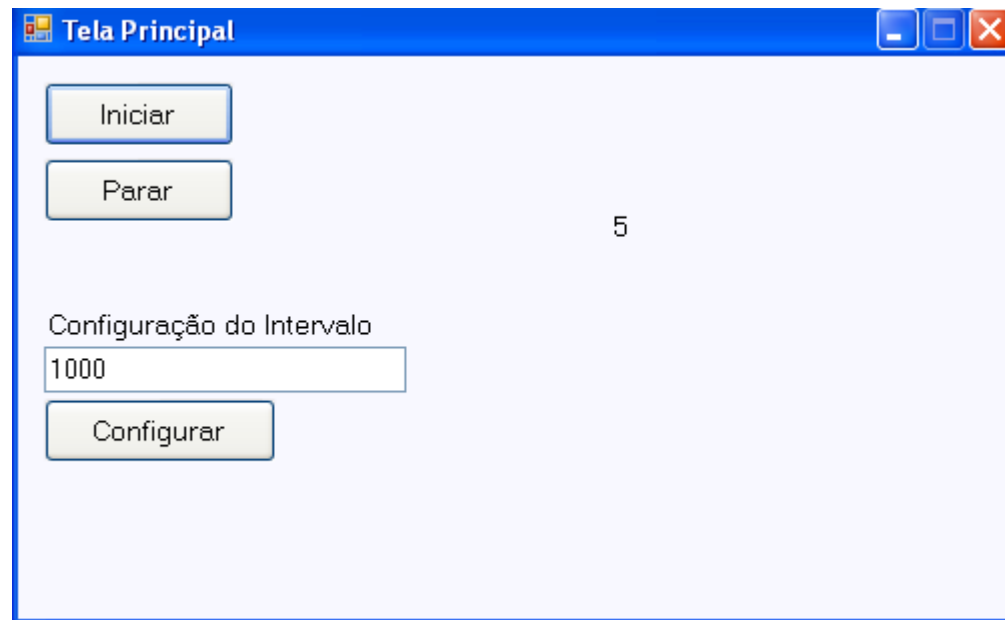
Properties	
timer1 System.Windows.Forms.Timer	
+ (ApplicationSettings)	
(Name)	timer1
Enabled	False
GenerateMember	True
Interval	100
Modifiers	Private
Tag	

```
namespace Minicurso_Aula01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
        }
    }
}
```

Exercício 5

- Montem um programa que permita começar e parar o timer, bem como configurar o seu intervalo.
- A cada processo do timer, uma variável deve ser incrementada e apresentada na tela.



Exercício 5

```
public partial class TelaPrincipal : Form
{
    //Variável global
    //Contador para o timer
    int contadorTimer = 0;

    public TelaPrincipal()
    {
        InitializeComponent();
    }

    private void btnConfigIntervalo_Click(object sender, EventArgs e)
    {
        timer1.Interval = Convert.ToInt32(tbIntervalo.Text); //Configuração do intervalo
    }

    private void btnIniciar_Click(object sender, EventArgs e)
    {
        timer1.Start(); //Inicia o timer
    }

    private void btnParar_Click(object sender, EventArgs e)
    {
        timer1.Stop(); //Para o timer
        contadorTimer = 0; //Zera o contador
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        contadorTimer = contadorTimer + 1; //Incrementa o contador
        labelTimer.Text = contadorTimer.ToString(); //Escreve o contador na tela
    }
}
```

Exercício 6

- Faça um cronômetro