

Linguagem de Programação Orientada a Objetos

**Ambiente de Programação
Microsoft .Net
C# (C sharp)**

Prof. Laércio Silva

Antes do Ambiente

- Nada funciona sem uma “**Lógica de Programação**” bem definida;
- Deve-se estabelecer uma “**Sequência Lógica**” do projeto a ser construído;
- Construir através de “**Instruções**” todos os itens de sistemas organizados e orientados no processo;
- Utilização dos parâmetros para o desenvolvimento de software no ambiente **Visual Studio.Net**.

Microsoft. Net

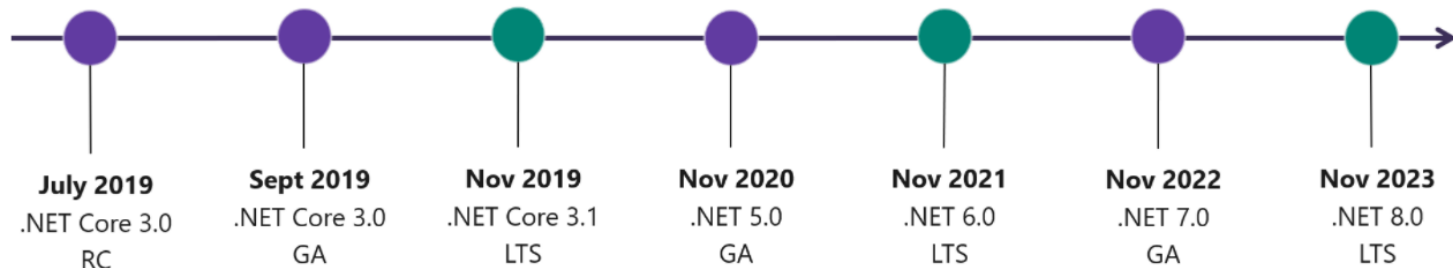
- Microsoft. Net é uma tecnologia que proporciona várias ferramentas para construção de aplicações cliente / servidor e aplicações web distribuídas de forma otimizada e organizada.
- As aplicações criadas em uma plataforma .Net podem ser utilizadas por um grande número de clientes, tais como: PC, Palm PC, Celulares, Smartphones, tablets, Games, Ultimate TV e etc.

Histórico do Microsoft. Net

- Primeira versão em fevereiro de 2002, segunda em março de 2003 e depois ao longo do período foram lançados as versões até a mais recente 2023.
- Propiciar a integração das várias ferramentas de desenvolvimento, Visual C#, Visual Basic, Visual J#, Visual C++ e outras ferramentas.
- Site Microsoft para desenvolvedores – MSDN –
<http://msdn.microsoft.com/pt-br/default>
- Linha do tempo net core.
<https://devblogs.microsoft.com/dotnet/introducing-net-5/>

Histórico do Microsoft. Net

.NET Schedule

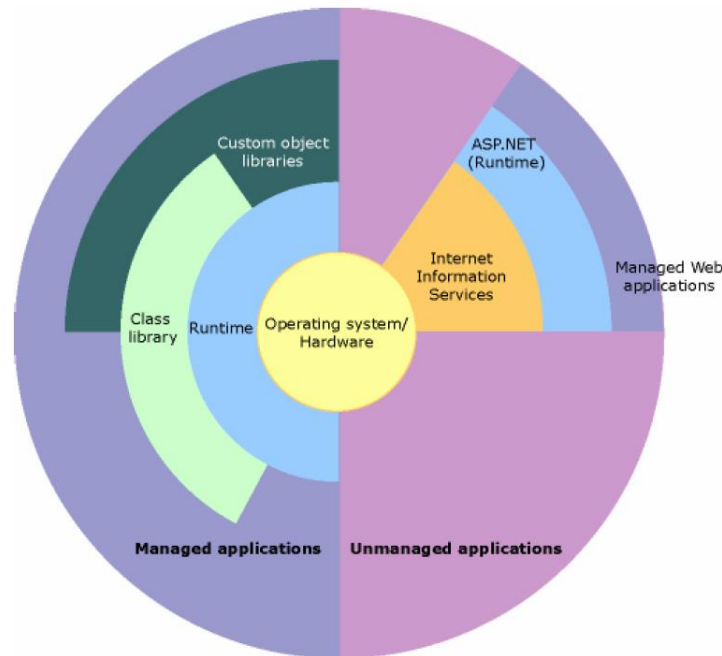


- .NET Core 3.0 release in September
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

Plataforma .Net Framework

Consiste em três etapas:

- **CLR (Common Language Runtime)** – Responsável pela interface entre o código e o sistema operacional;
- **Classes do Framework** - Todas as linguagens que usam a tecnologia .NET usam as mesmas classes;
- **ASP.NET** - Fornece o acesso direto a toda a linguagem **VB.NET**(Visual Basic) e/ou **C#** a partir de uma plataforma de "scriptação".



Common Language RunTime(CLR)

Nessa nova tecnologia da Microsoft, o compilador não gera código nativo, ou seja, ele gera um código intermediário para qualquer **SO**. O código passa a ser compilado para ser rodado dentro do Interpretador **CLR**. Com isto, o desenvolvedor poderá escolher uma entre as várias linguagens de programação que trabalham com essa tecnologia.



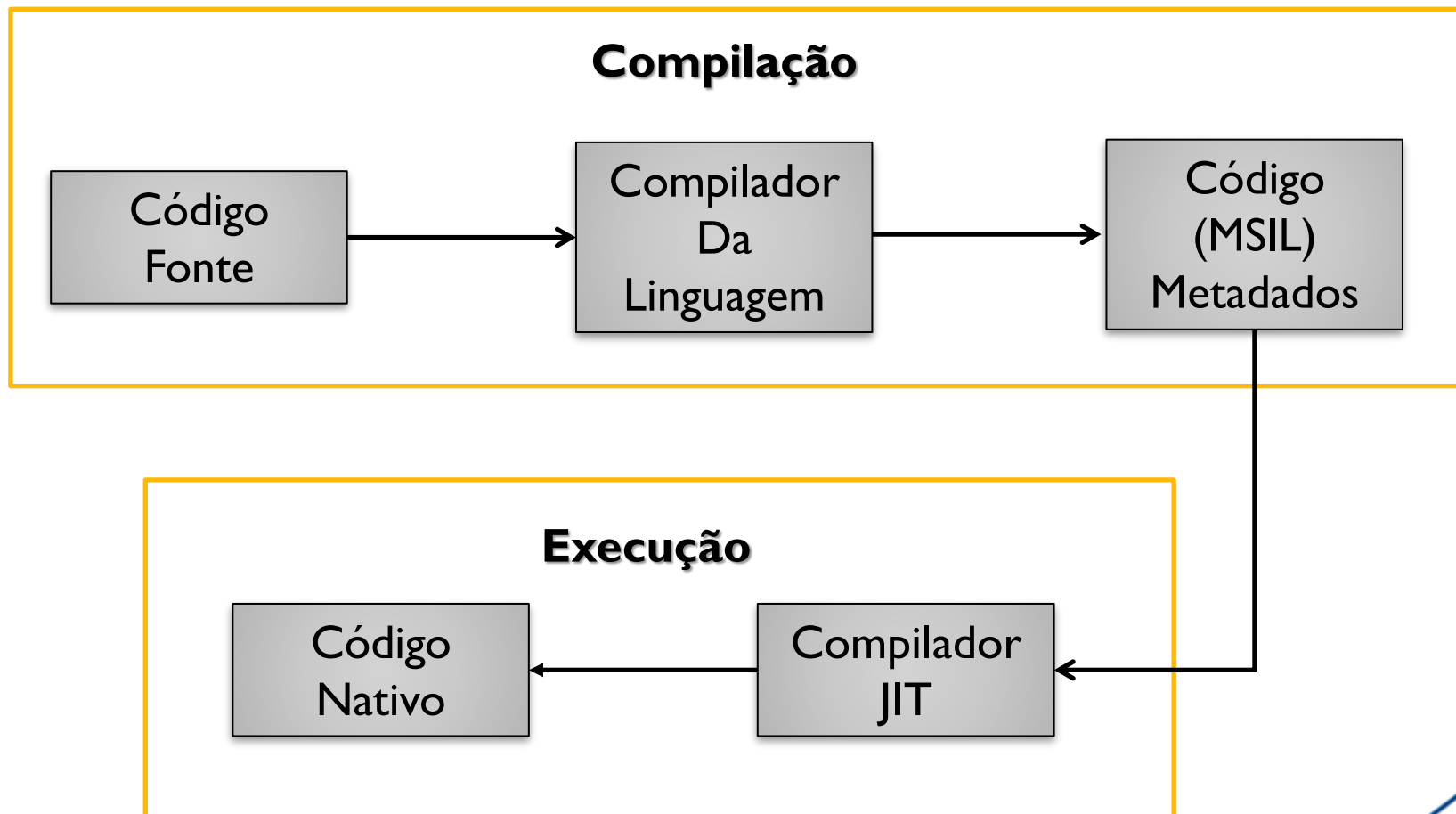
Compilação

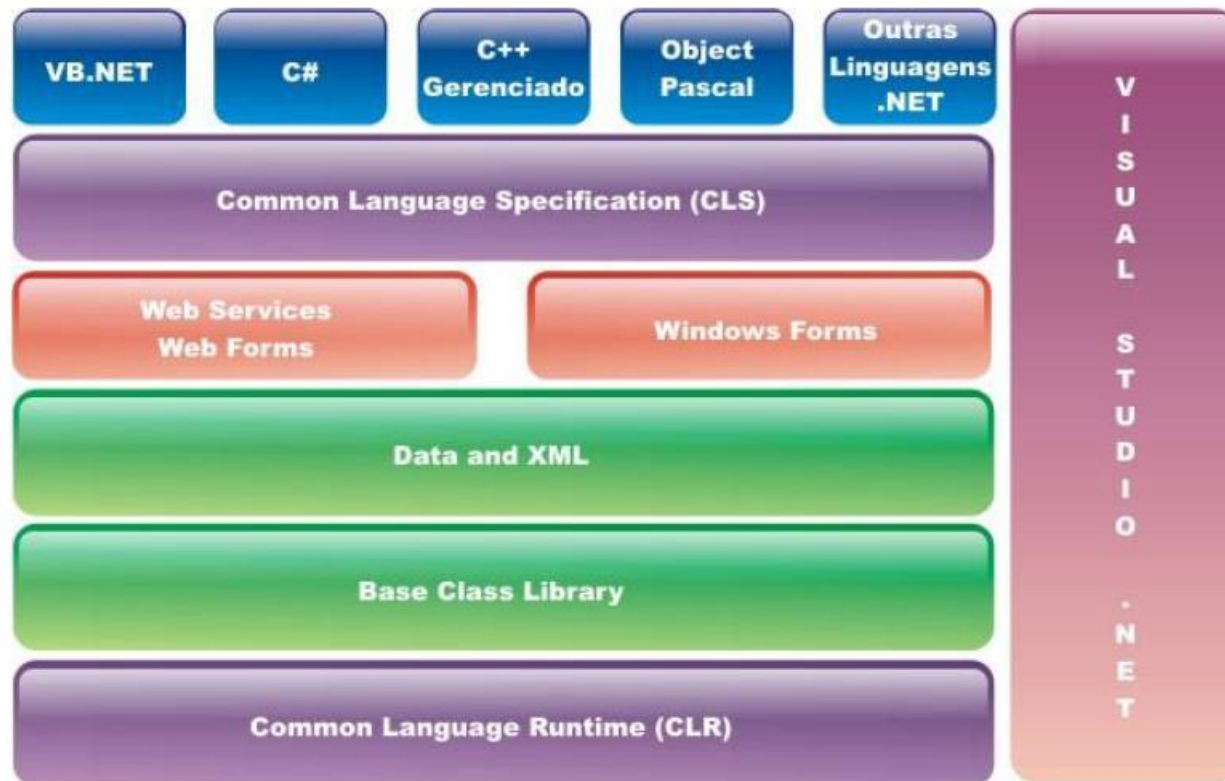
Qualquer linguagem que usa a tecnologia **.NET**, realiza o mesmo esquema de compilação, ou seja, ao compilar um código, gera-se um arquivo compilado para uma linguagem intermediária – **MSIL** (**Micro**Soft **I**ntermediate **L**anguage). Esse arquivo gerado é chamado de Assenbly, podendo ter duas extensões: **EXE** ou **DLL**.

Quando o arquivo é executado, o **JIT** (Just-In-Time) converte este programa em código de máquina para ser rodado sobre o **SO** em que o **CLR** está rodando.

Desta forma, o **MSIL** gera o mesmo arquivo binário para qualquer plataforma que tiver o **CLR**, que por sua vez converte esse arquivo para código de máquina compatível ao **SO** corrente.

Compilação e Execução





Daí se dá a idéia de **Portabilidade**, que consiste em procedida a compilação, o arquivo pode ser executado em várias plataformas de **SO** (Ex: Windows, Linux, MacOS, etc).

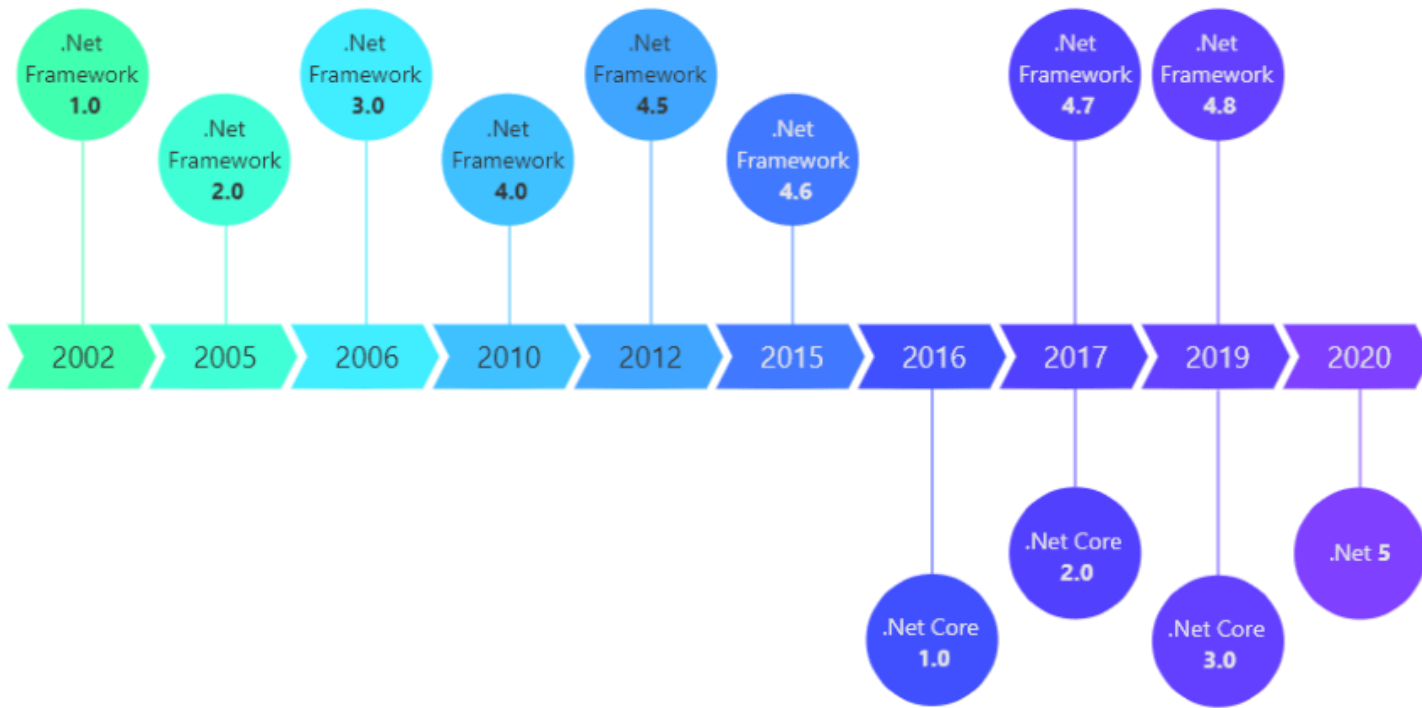
Linguagem C#

- O C# (pronuncia-se "C sharp") é uma linguagem de programação criada para o desenvolvimento de uma variedade de aplicações que executam sobre o .NET Framework.
- C# é uma linguagem simples, poderosa, com tipagem segura e orientada a objetos.
- As várias inovações no C# permitem o desenvolvimento rápido de aplicações, mantendo a expressividade e a elegância das linguagens C-Style.

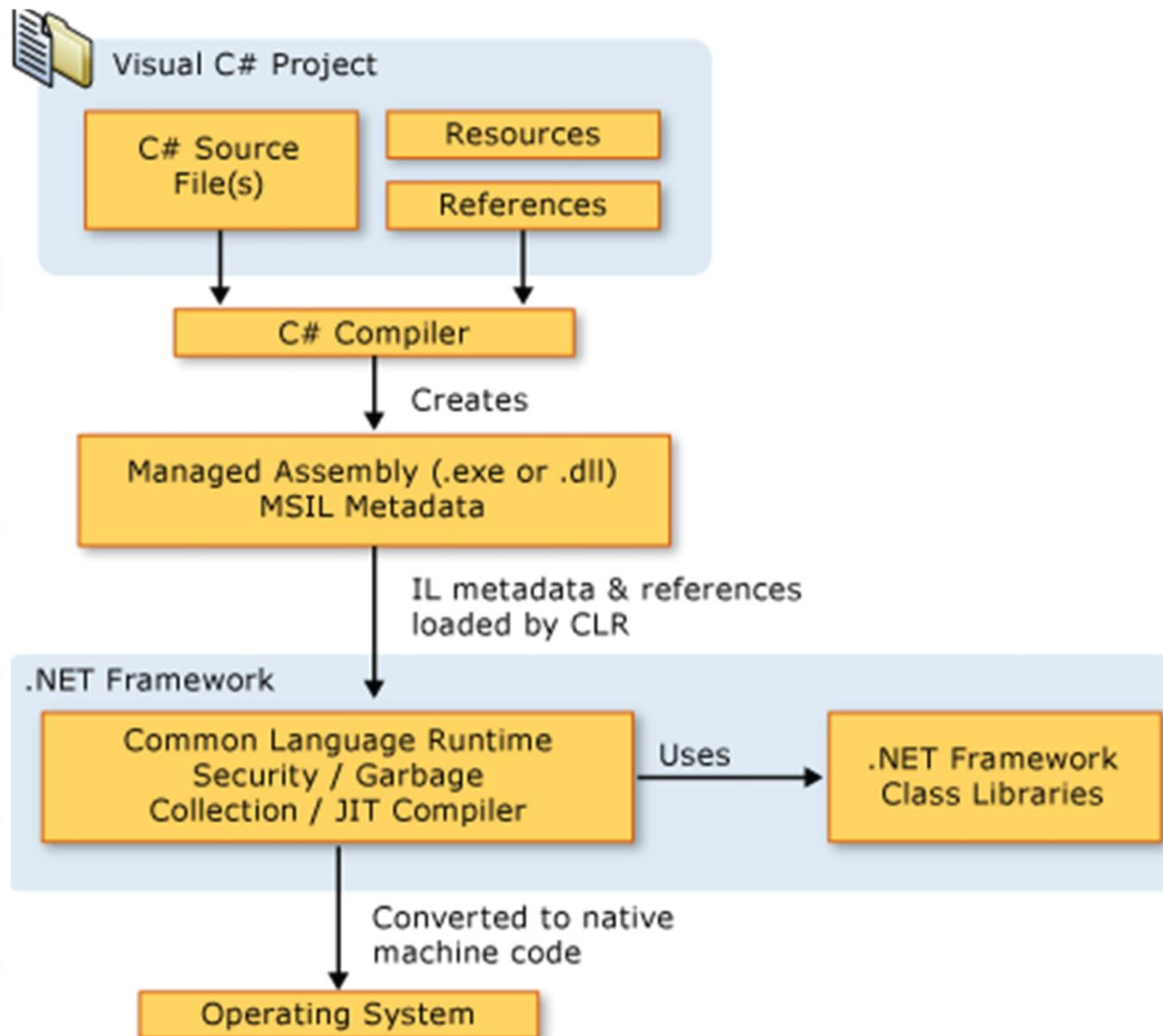
Linguagem C#

- Visual C# é uma implementação da linguagem C# pela Microsoft. Visual Studio suporta Visual C# com um editor de código completo, compilador, modelos de projetos, designers, assistentes de código, um depurador poderoso e fácil de usar e outras ferramentas.
- A biblioteca de classes do .NET Framework fornece acesso a vários serviços do sistema operacional e outras classes úteis e bem estruturadas que aceleram significativamente o ciclo de desenvolvimento.

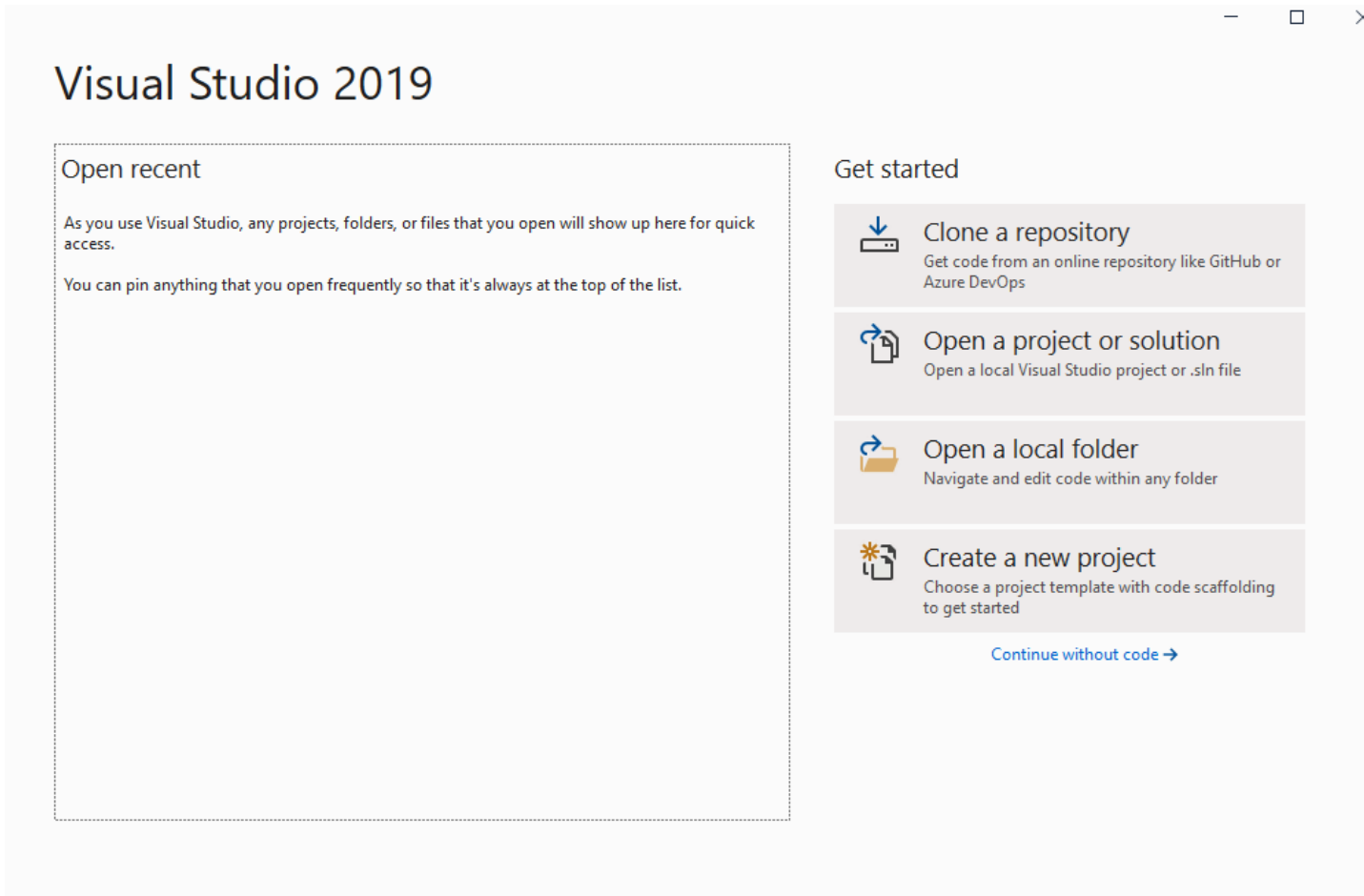
C # tem evoluído ao longo do tempo para incluir muitas novidades



Arquitetura da plataforma do .NET Framework no C#



Janela Inicial Visual Studio



Janela Criar novo projeto

Create a new project


Recent project templates

A list of your recently accessed templates will be displayed here.


All languages

All platforms


All project types

 **Console Application**
A project for creating a command-line application that can run on .NET Core on Windows, Linux and macOS


C# Linux macOS Windows Console

 **Console Application**
A project for creating a command-line application that can run on .NET Core on Windows, Linux and macOS


Visual Basic Linux macOS Windows Console

 **ASP.NET Core Web App**
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

C# Linux macOS Windows Cloud Service Web

 **Blazor WebAssembly App**
A project template for creating a Blazor app that runs on WebAssembly and is optionally hosted by an ASP.NET Core app. This template can be used for web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web

 **Class library**
A project for creating a class library that targets .NET Standard or .NET Core

Back

Next

Janela Criar novo projeto

Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S)

All languages

All platforms

All project types



A project that contains NUnit tests that can run on .NET Core on windows, LINUX and MacOS.

Visual Basic

Linux

macOS

Windows

Desktop

Test

Web



Windows Forms App (.NET Framework)

A project for creating an application with a Windows Forms (WinForms) user interface

C#

Windows

Desktop



Windows Forms App

A project template for creating a .NET Windows Forms (WinForms) App.

C#

Windows

Desktop



Windows Forms App

A project template for creating a .NET Windows Forms (WinForms) App.

Visual Basic

Windows

Desktop



WPF Application

A project for creating a .NET Core WPF Application

C#

Windows

Desktop



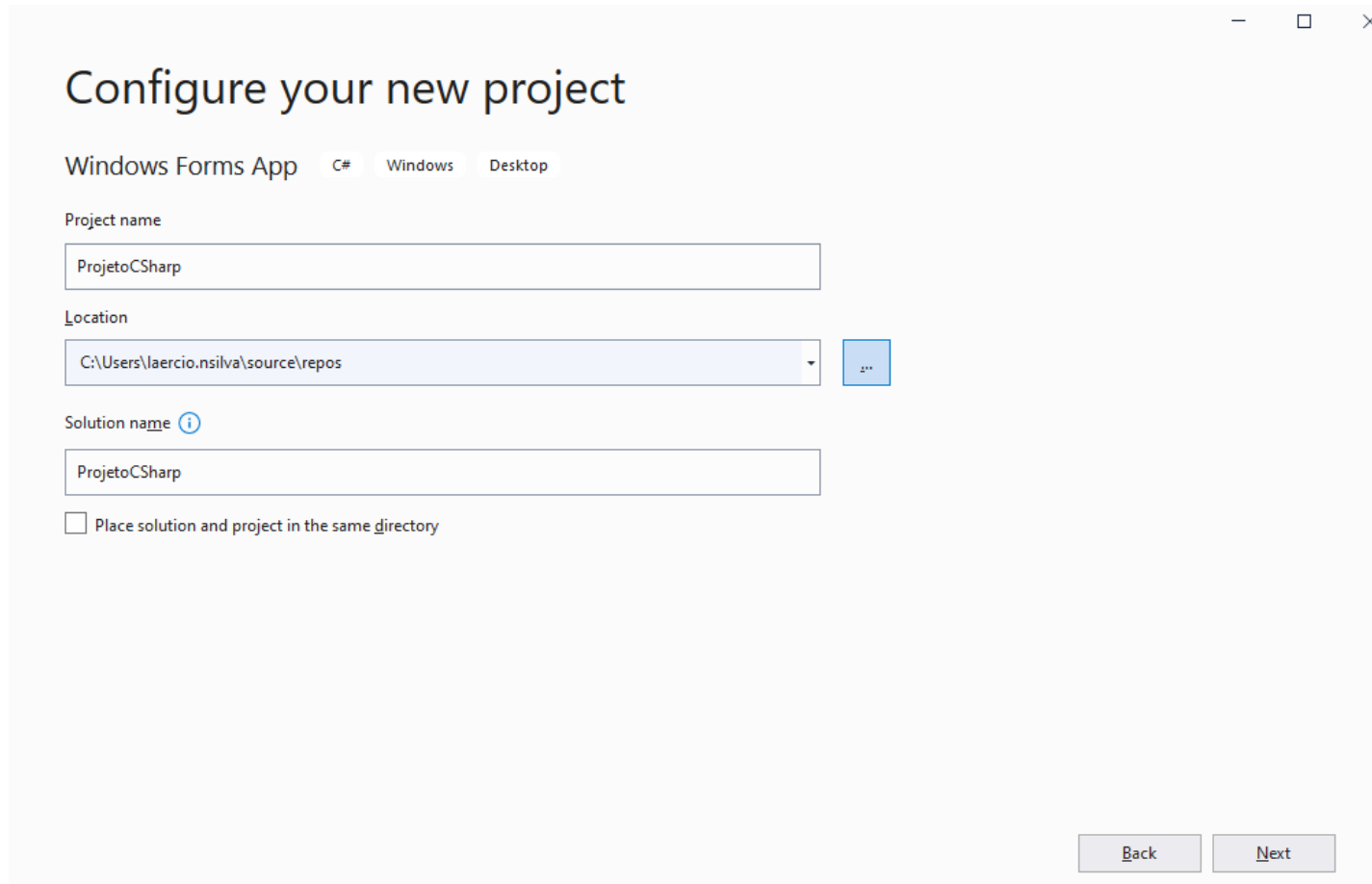
WPF Application

A project for creating a .NET Core WPF Application

Back

Next

Configurando seu projeto

A screenshot of the 'Configure your new project' dialog box in Visual Studio. The dialog has a title bar with standard Windows window controls. The main content area is titled 'Configure your new project'. Below the title, there are tabs for 'Windows Forms App', 'C#', 'Windows', and 'Desktop'. The 'C#' tab is selected. The 'Project name' field contains 'ProjetoCSharp'. The 'Location' field is a dropdown menu showing 'C:\Users\laercio.nsilva\source\repos' with a blue button to the right. The 'Solution name' field, which has an information icon, contains 'ProjetoCSharp'. At the bottom, there is an unchecked checkbox labeled 'Place solution and project in the same directory'. At the bottom right, there are 'Back' and 'Next' buttons.

Configure your new project

Windows Forms App C# Windows Desktop

Project name

ProjetoCSharp

Location

C:\Users\laercio.nsilva\source\repos

Solution name ⓘ

ProjetoCSharp

☐ Place solution and project in the same directory

Back Next



Configurando o net Framework

Additional information

Windows Forms App C# Windows Desktop

Target Framework [i](#)

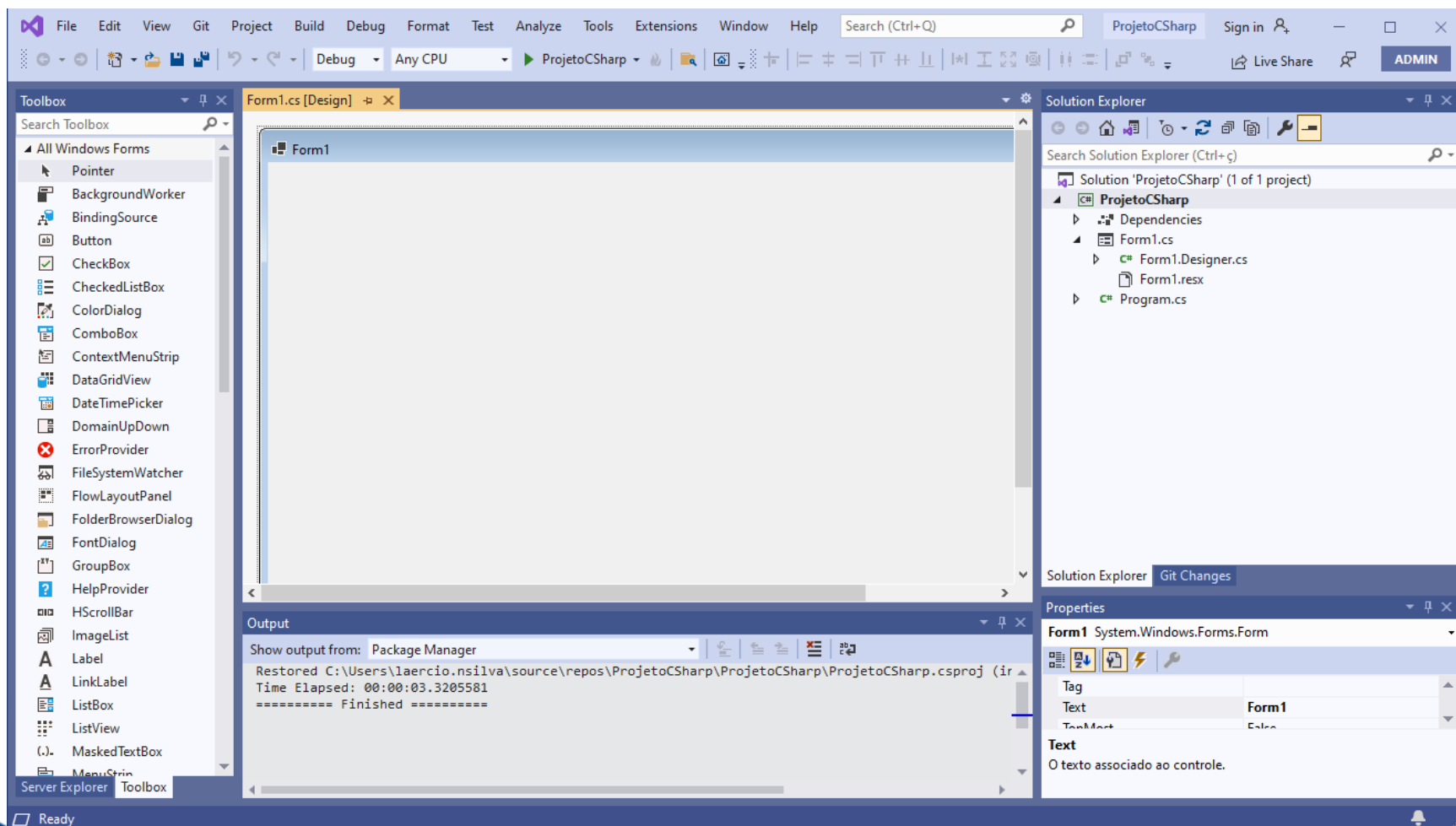
.NET Core 3.1 (Long-term support)

Back

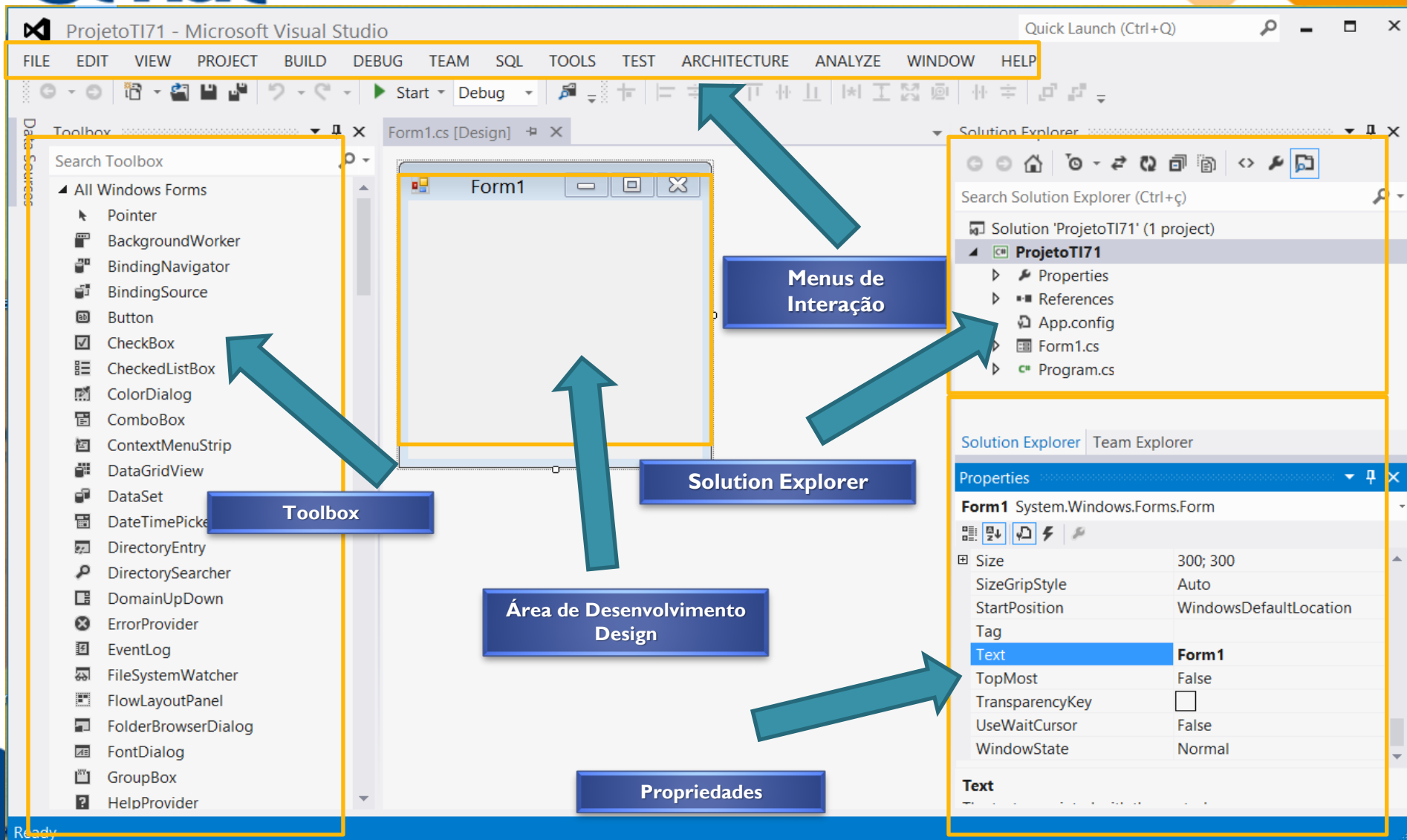
Create



Ambiente de Desenvolvimento do C#



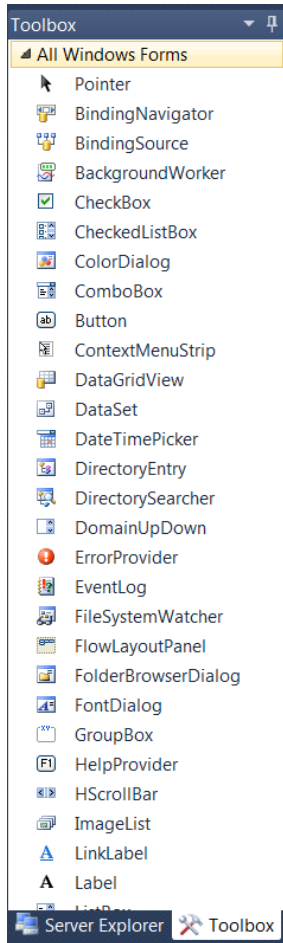
Janela de Desenvolvimento



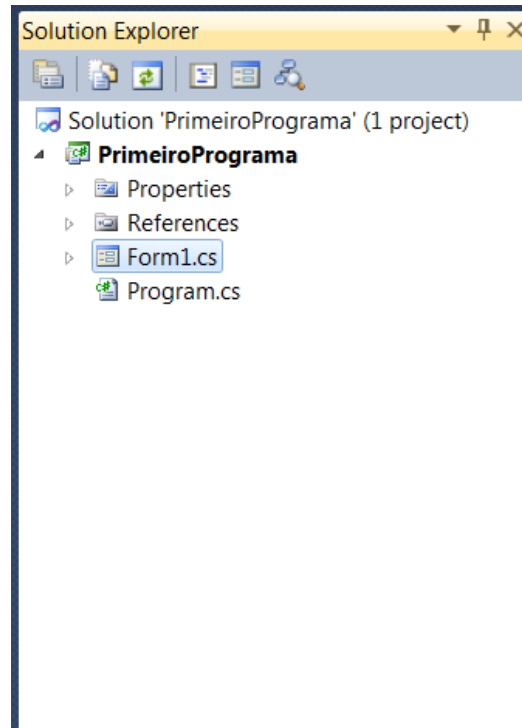
The image shows the Microsoft Visual Studio IDE interface for a project named 'ProjetoTI71'. The interface is annotated with five blue boxes and teal arrows pointing to specific components:

- Menus de Interação**: Points to the menu bar at the top of the window, which includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, TOOLS, TEST, ARCHITECTURE, ANALYZE, WINDOW, and HELP.
- Solution Explorer**: Points to the Solution Explorer pane on the right side of the IDE, which displays the project structure for 'ProjetoTI71' (1 project), including Properties, References, App.config, Form1.cs, and Program.cs.
- Área de Desenvolvimento Design**: Points to the central design area where 'Form1' is being developed.
- Propriedades**: Points to the Properties window at the bottom right, which shows the properties for the selected 'Form1' (System.Windows.Forms.Form), including Size, SizeGripStyle, StartPosition, Tag, Text, TopMost, TransparencyKey, UseWaitCursor, and WindowState.
- Toolbox**: Points to the Toolbox on the left side of the IDE, which contains a list of Windows Forms controls such as Pointer, BackgroundWorker, BindingNavigator, BindingSource, Button, CheckBox, CheckedListBox, ColorDialog, ComboBox, ContextMenuStrip, DataGridView, DataSet, DateTimePicker, DirectoryEntry, DirectorySearcher, DomainUpDown, ErrorProvider, EventLog, FileSystemWatcher, FlowLayoutPanel, FolderBrowserDialog, FontDialog, GroupBox, and HelpProvider.

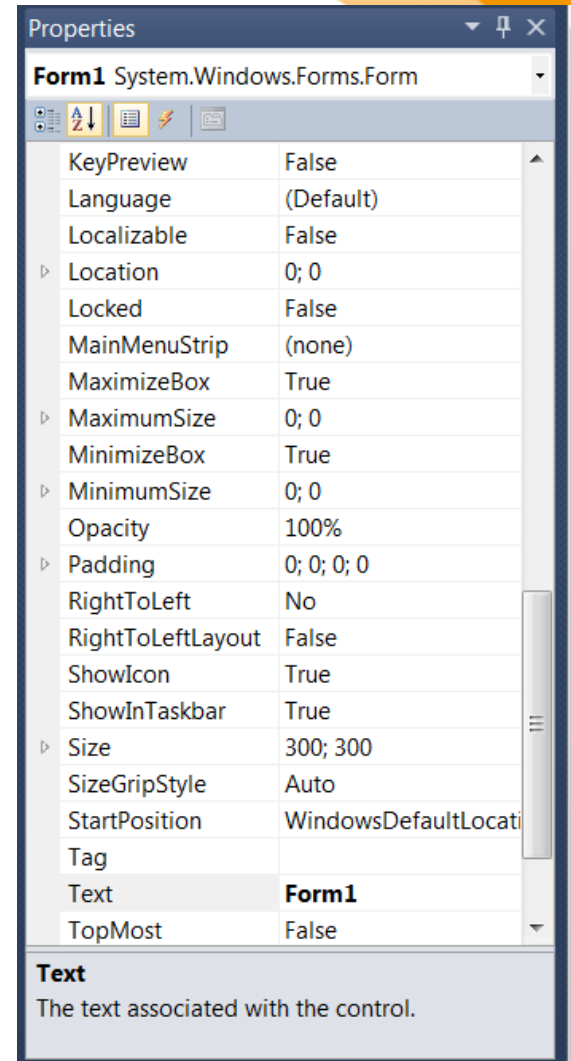
Ferramentas de Desenvolvimento



**Ferramentas
Toolbox**



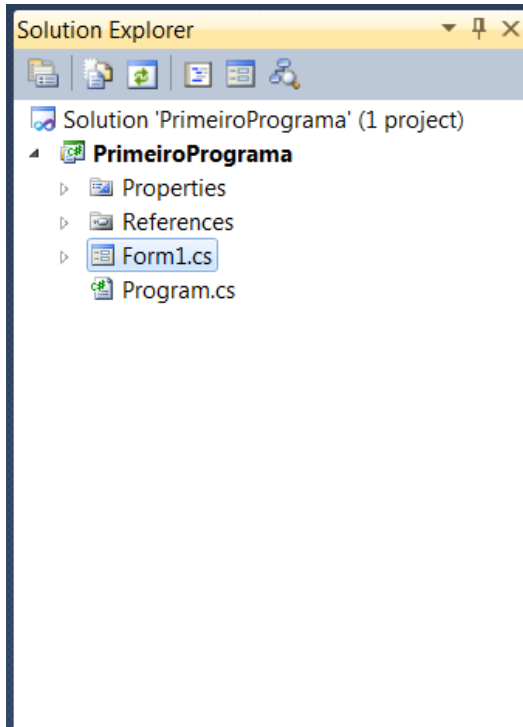
**Ferramentas
Solution
Explorer**



**Ferramentas
Propriedades**

Exercício 1

- Experimentar a navegação no ambiente:
 - Coloque a barra de toolbox do lado direito da tela flutuando;
 - Coloque a barra de propriedades juntamente com o “Form1.cs [Designer]”;
 - Coloque a barra de Solution Explorer na parte inferior da janela de edição do visual c#;
 - Inserir a janela de código editor ao lado do “Form1.cs [Designer]”;
 - Deixe todas a janelas flutuando sobre a área de trabalho do visual c#;
 - Reorganize sua área de trabalho com as barras de acordo com o início da aula.



O Solution Explorer é dividido da seguinte forma:

- Projeto;
- Classes;
- Propriedades;
- Referencias;
- “Designer” - Form.cs ;
- Código - “Program.cs”.



- **Error** – erros em linha de código;
- **Warnings** – aviso de sistema;
- **Messages** - mensagens diversas de ocorrências de sistemas

The screenshot shows the 'Error List' window in Visual Studio. At the top, there are three summary boxes: '1 Error' (red X icon), '0 Warnings' (yellow triangle icon), and '0 Messages' (blue i icon). Below these is a table with columns: Description, File, Line, Column, and Project. The first row contains an error message: 'Only assignment, call, increment, decrement, and new object expressions can be used as a statement'. This error is associated with the file 'Form1.cs' at line 22, column 13, under the project 'PrimeiroPrograma'.

	Description	File	Line	Column	Project
✖ 1	Only assignment, call, increment, decrement, and new object expressions can be used as a statement	Form1.cs	22	13	PrimeiroPrograma

Tecclas de Atalho

Tecclas que auxiliam no processo de codificação do sistema:

- **CTRL+K+D** – organiza o código;
- **CTRL+A** – Seleciona o código
- **CTRL+ F5 ou F5**– executar a parte selecionada do editor de consultas ou todo o editor de consultas se nada estiver selecionado;
- **CTRL+SHIFT+B** - constrói o código e verifica possíveis erros antes de executar;

Estrutura de Programa C#

Bibliotecas do net Framework

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Windows.Forms;
```

Projeto

```
namespace PrimeiroPrograma  
{
```

Classe

```
    static class Program  
    {
```

```
        /// <summary>  
        /// The main entry point for the application.  
        /// </summary>
```

```
        [STAThread]  
        static void Main()  
        {
```

Métodos

```
            Application.EnableVisualStyles();  
            Application.SetCompatibleTextRenderingDefault(false);  
            Application.Run(new Form1());
```

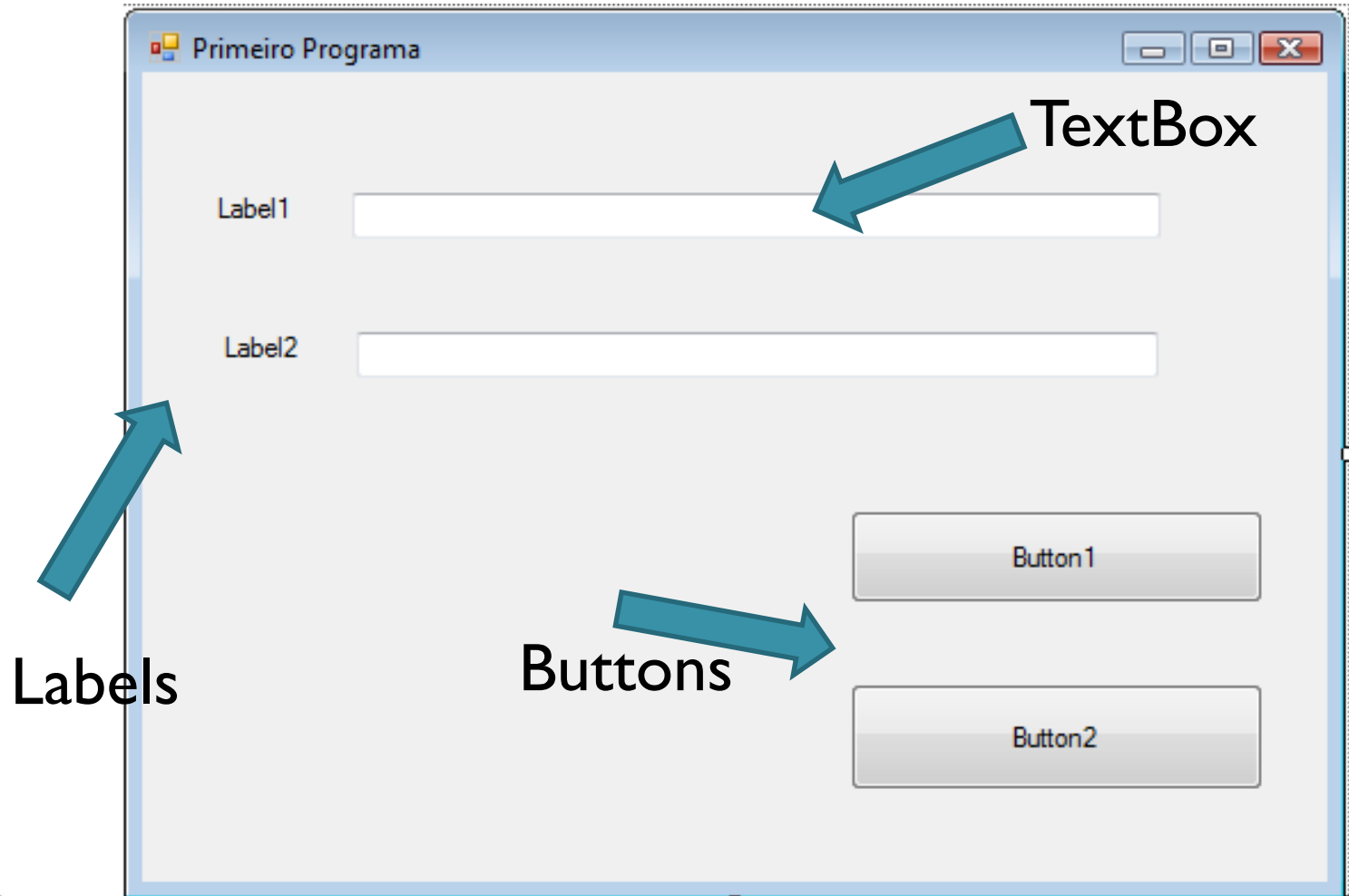
```
        }  
    }  
}
```

- TEXTBOX;
- BUTTONS;
- LABELS;
- LISTBOX;
- Dentre outros
- Auxiliam na criação de aplicações no Visual c#.

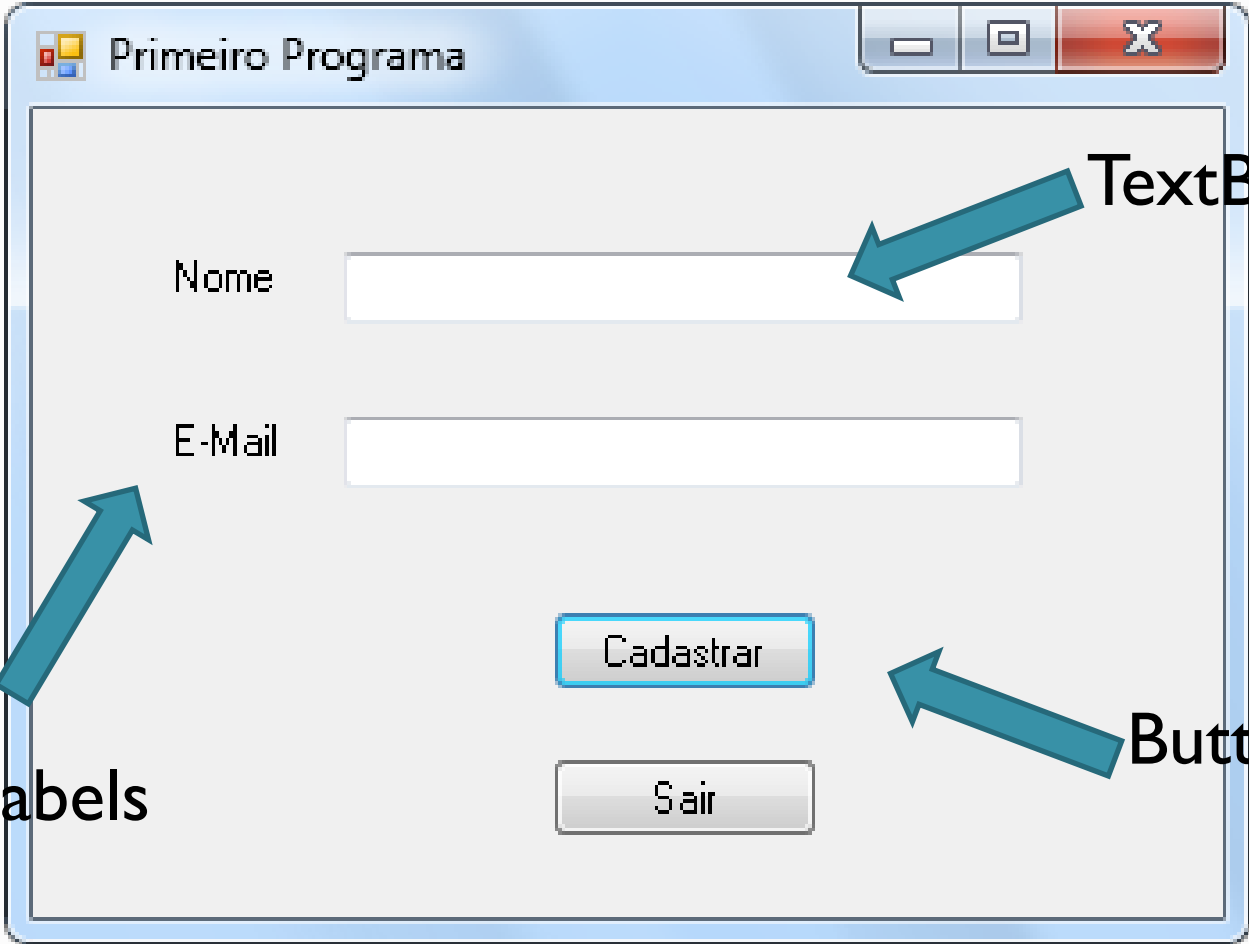
Algumas Nomenclaturas para Nomes dos Objetos

Objetos (componentes)	Prefixo	Exemplo
Form	frm	frmFileOpen
UserControl	usc	uscLogin
TextBox	txt	txtGetText
Label	lbl	lblTitle
ComboBox	cbo	cboCity
Image	img	imgIcon
PictureBox	pic	picHeader
Grid	grd	grdPrices
GridView	grv	grvClients
Radio Button	rbt	rbtRequerid
LinkLabel	lkl	lklSite
CheckBox	chk	chkSex
ListBox	lst	lstPrices
Button	btn	btnClientSave
TreeView	trv	trvMenu
DropDownList	Ddl	ddlVendedores

Exemplo de Alguns Controles



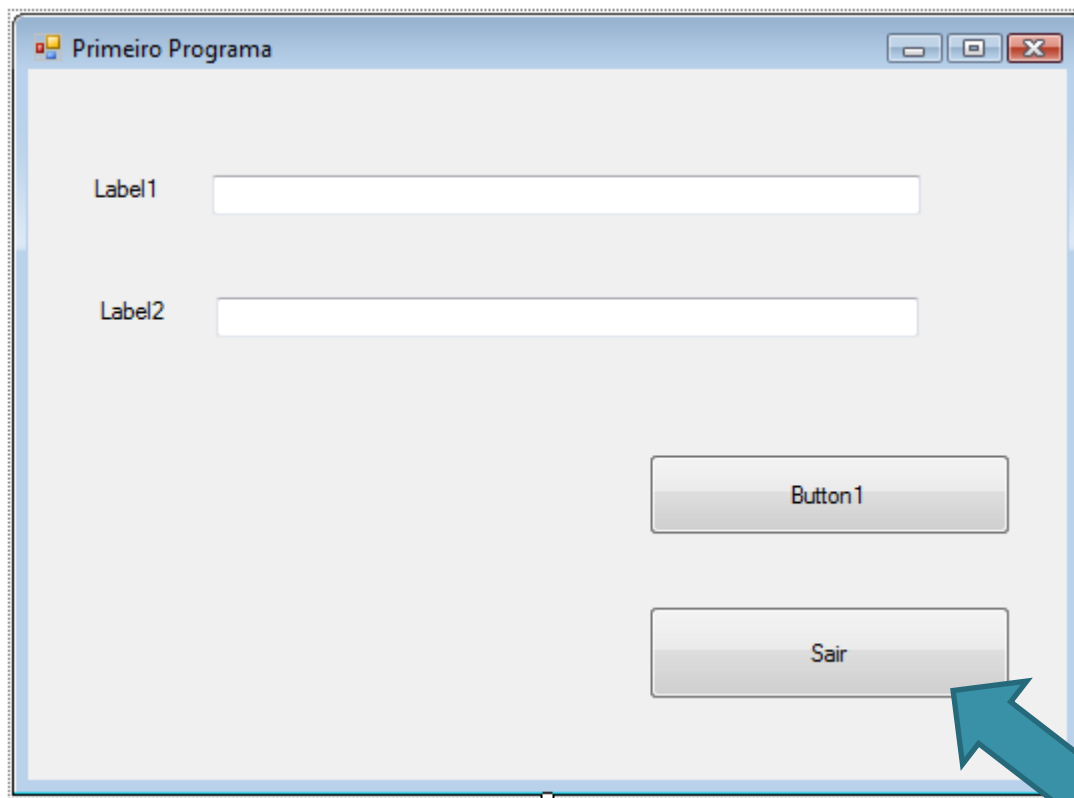
Exemplo de Alguns Controles



The image shows a screenshot of a Windows application window titled "Primeiro Programa". The window contains a registration form with the following elements:

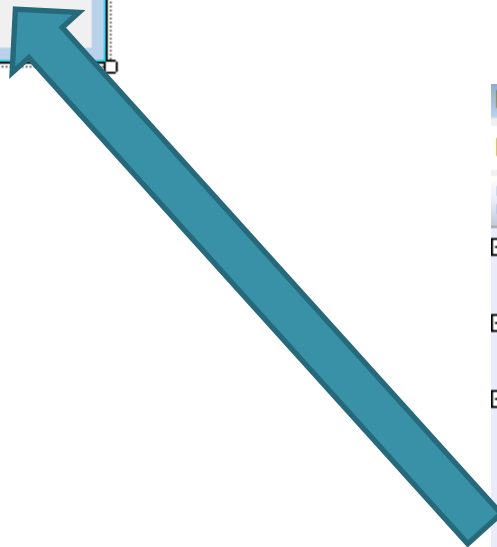
- Labels:** Two labels are present on the left side of the form: "Nome" and "E-Mail". A blue arrow points from the word "Labels" to the "Nome" label.
- TextBox:** Two text input boxes are located to the right of the labels. A blue arrow points from the word "TextBox" to the top text box.
- Buttons:** Two buttons are located at the bottom of the form: "Cadastrar" and "Sair". A blue arrow points from the word "Buttons" to the "Cadastrar" button.

Edição da Propriedade



Propriedade
Text - Botão Sair

Edição da Propriedade



Properties

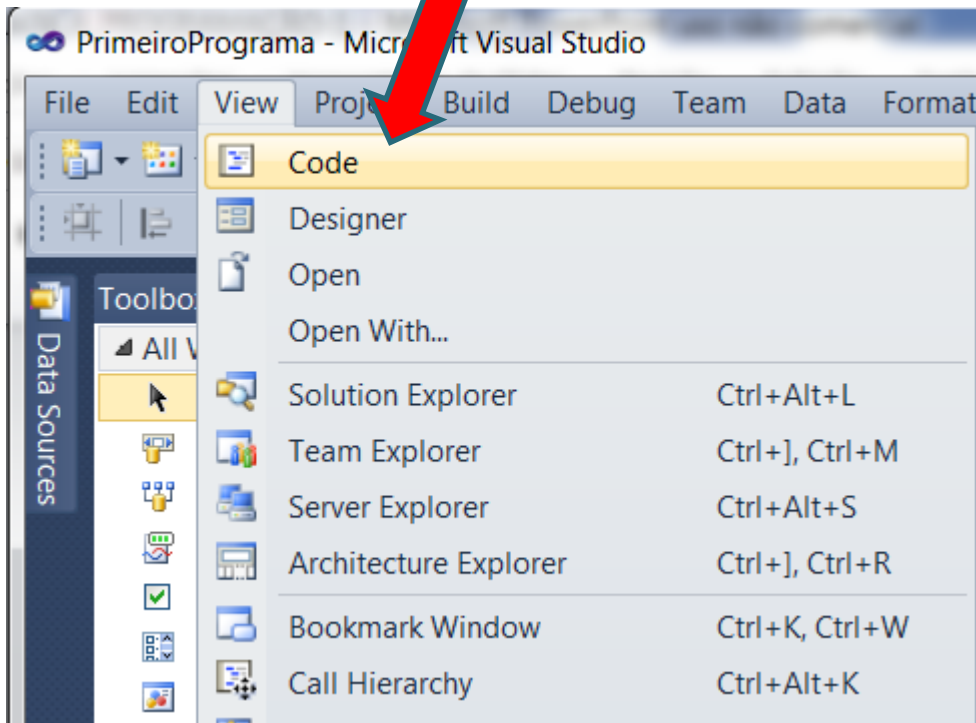
Button1 System.Windows.Forms.Button

MinimumSize	0; 0
Modifiers	Friend
Padding	0; 0; 0; 0
RightToLeft	No
Size	92; 35
TabIndex	0
TabStop	True
Tag	
Text	Sair
TextAlign	MiddleCenter

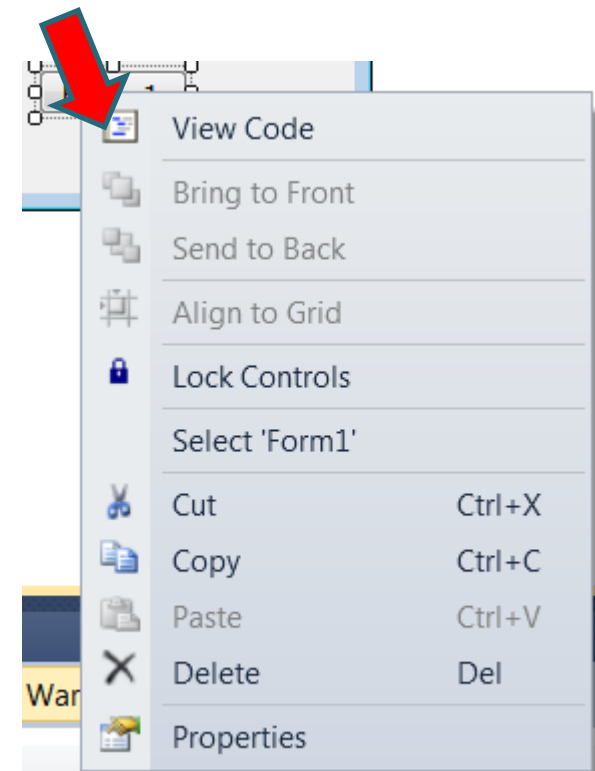
Text
The text associated with the control.

Edição de Códigos

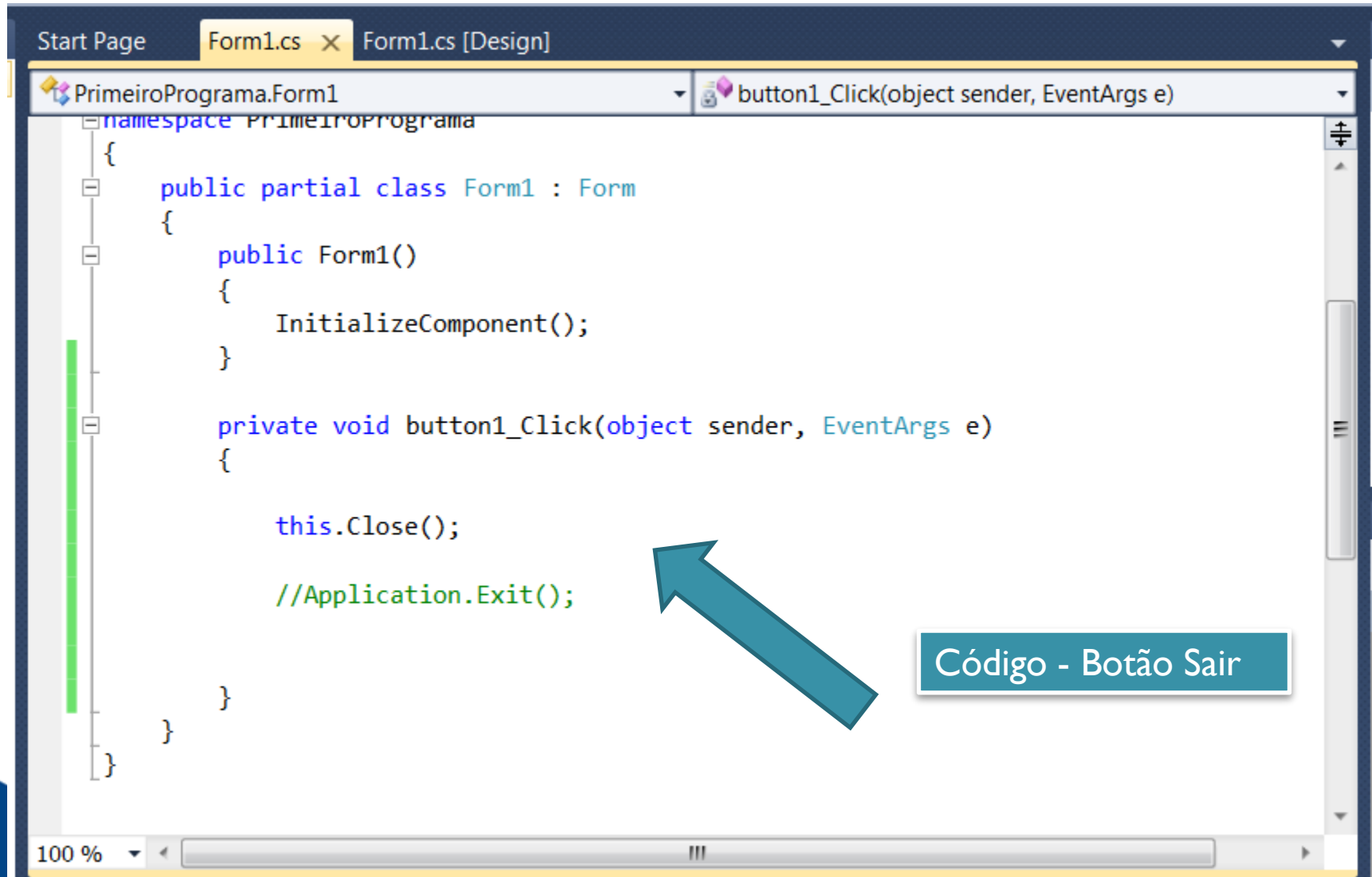
- Menu suspenso - View – Code



- Botão direito do mouse



Edição de Códigos



The screenshot shows the Visual Studio IDE with the 'Form1.cs' file open in the 'Design' view. The code editor displays the following C# code:

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

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();

            //Application.Exit();
        }
    }
}
```

A blue arrow points from a text box labeled 'Código - Botão Sair' to the `button1_Click` method in the code.

100 %

Tipos de Comentários de Códigos

Comentário de bloco de linhas.

```
/*using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;*/

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

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();

            //Application.Exit(); } } }
```

O compilador ignora as linhas comentadas.
Utilizado para documentar o sistema.

Comentário de linha.



Exercício 2

Encontre o erro no código abaixo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace PrimeiroPrograma
{
    public partial classe Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            este.Close();

            Aplicativo.Sair();
        }
    }
}
```

Tela Programa

A screenshot of a Windows application window titled "Primeiro Programa". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is titled "Dados Pessoais" in bold. It contains four input fields: "Nome" (a single-line text box), "Cargo" (a single-line text box), "Telefone" (a single-line text box), and "Departamento" (a dropdown menu). At the bottom of the window, there are two buttons: "Cadastrar" (with an underline under the 'C') and "Sair" (with an underline under the 'S').

Primeiro Programa

Dados Pessoais

Nome

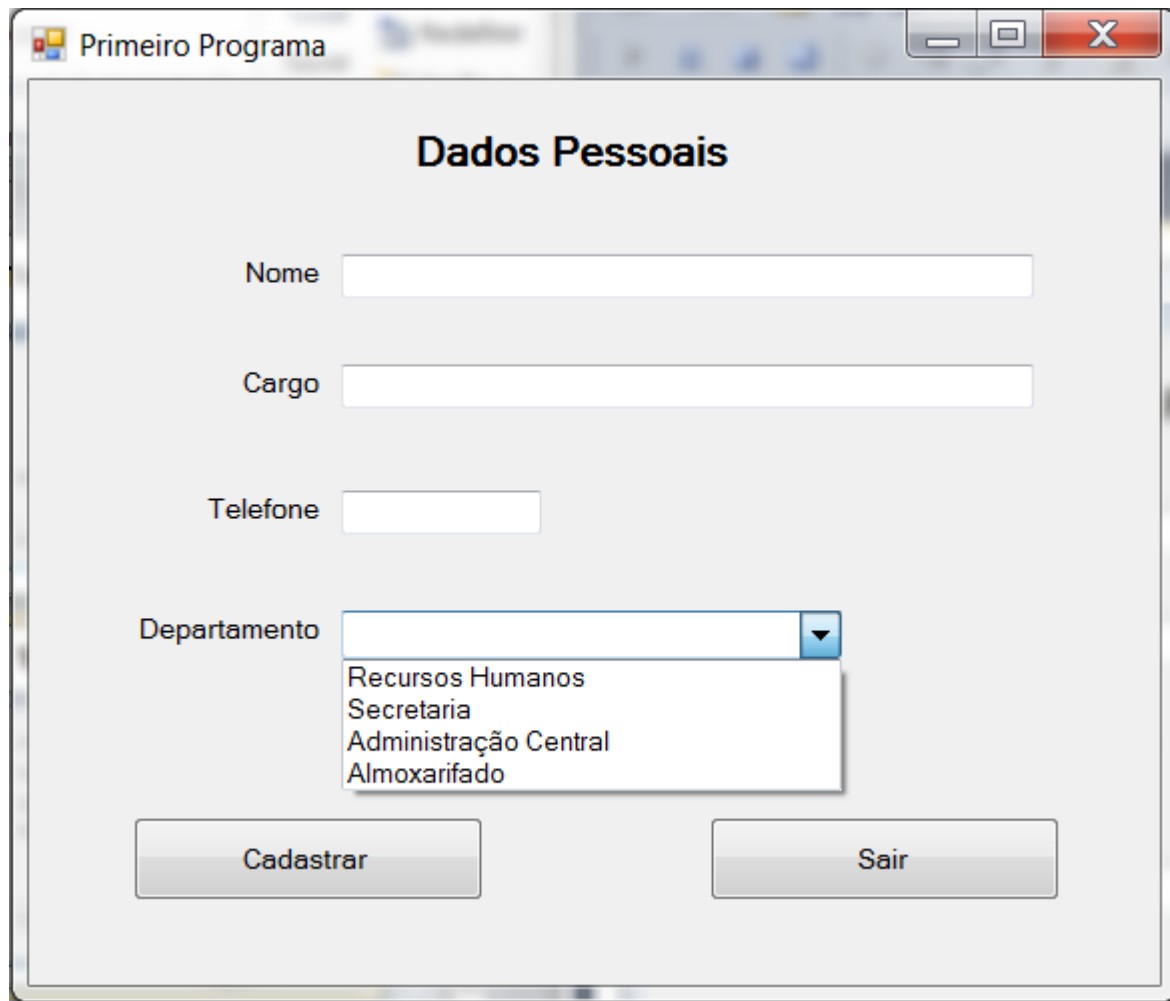
Cargo

Telefone

Departamento

Cadastrar Sair

Tela Programa

A screenshot of a Windows application window titled "Primeiro Programa". The window contains a form titled "Dados Pessoais" with the following fields: "Nome" (text input), "Cargo" (text input), "Telefone" (text input), and "Departamento" (dropdown menu). The dropdown menu is open, showing a list of departments: "Recursos Humanos", "Secretaria", "Administração Central", and "Almoxarifado". At the bottom of the form are two buttons: "Cadastrar" and "Sair".

Primeiro Programa

Dados Pessoais

Nome

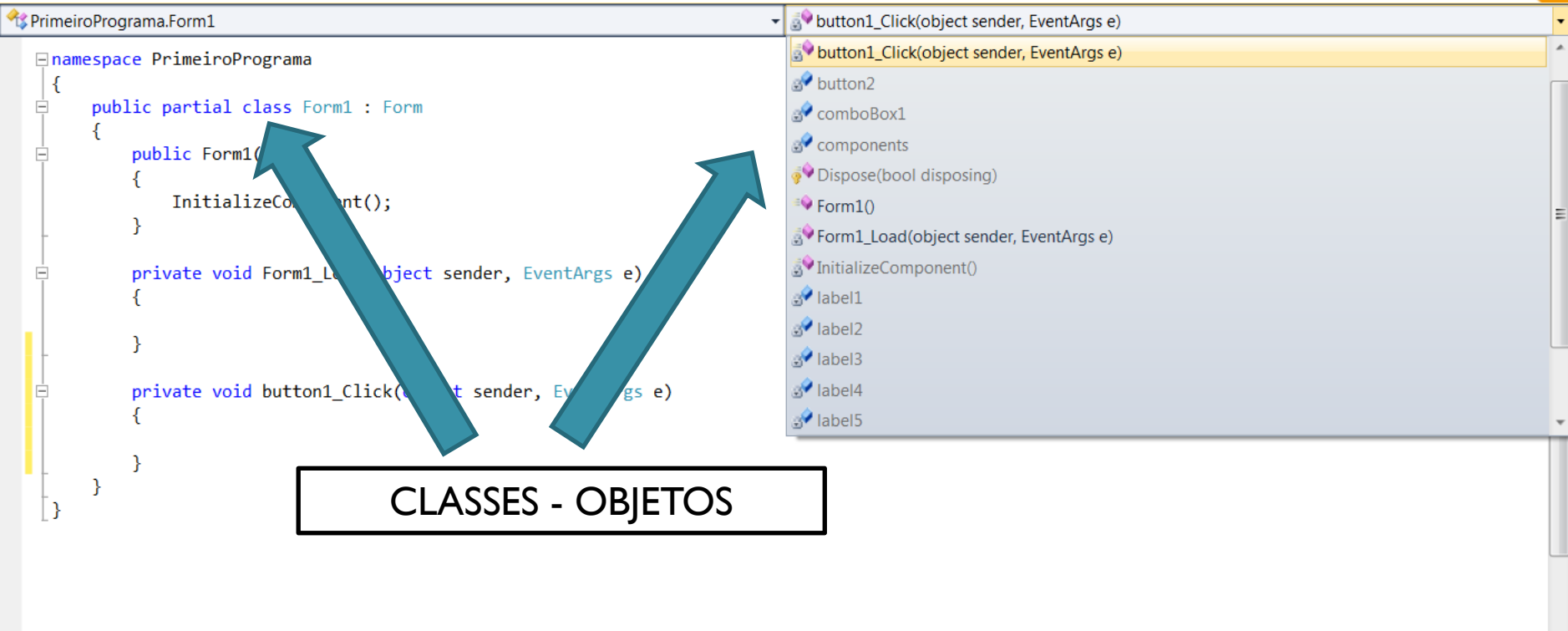
Cargo

Telefone

Departamento

- Recursos Humanos
- Secretaria
- Administração Central
- Almoxarifado

Classes - Objetos



The screenshot displays the Visual Studio IDE with the C# code for `PrimeiroPrograma.Form1` on the left and the Properties window on the right. Two blue arrows originate from a central box labeled "CLASSES - OBJETOS". One arrow points to the `Form1` class definition in the code, and the other points to the `button1_Click` event handler in the Properties window.

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

        private void Form1_Load(object sender, EventArgs e)
        {
        }

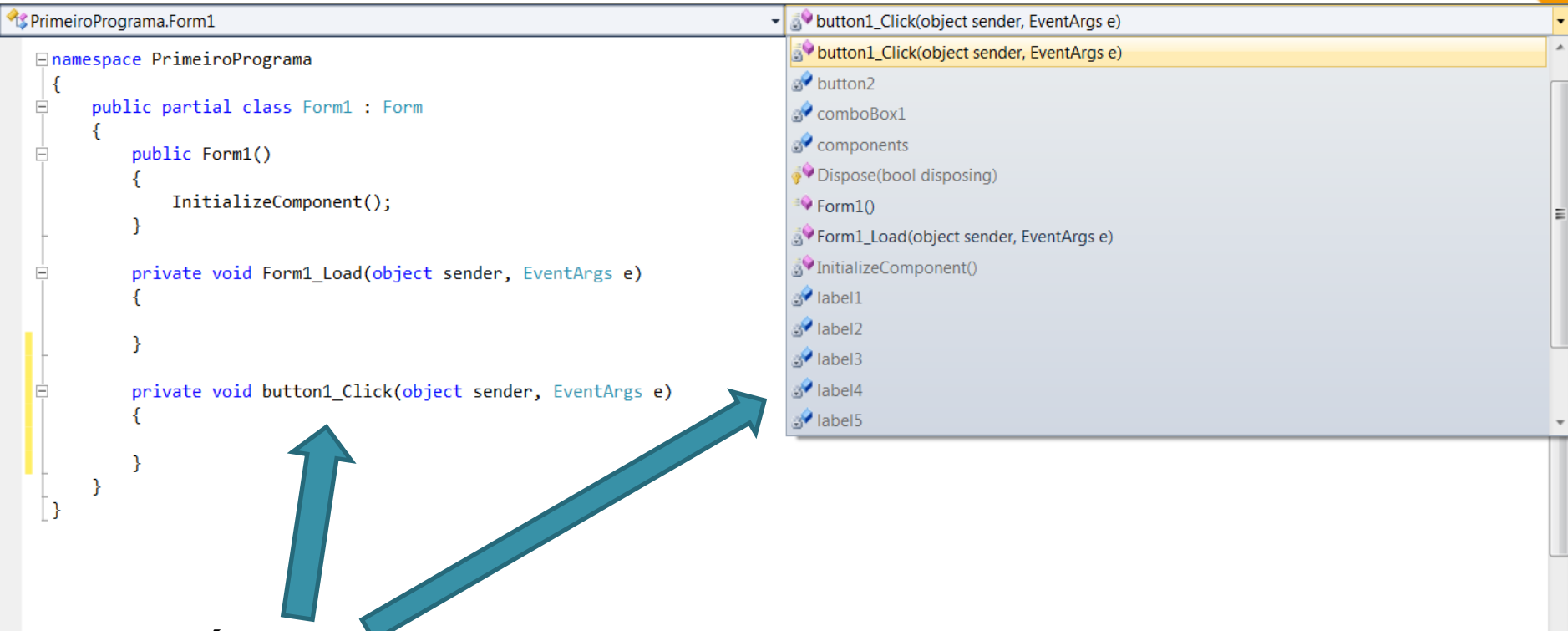
        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

The Properties window shows the following items:

- `button1_Click(object sender, EventArgs e)` (selected)
- `button1_Click(object sender, EventArgs e)`
- `button2`
- `comboBox1`
- `components`
- `Dispose(bool disposing)`
- `Form1()`
- `Form1_Load(object sender, EventArgs e)`
- `InitializeComponent()`
- `label1`
- `label2`
- `label3`
- `label4`
- `label5`

CLASSES - OBJETOS

Métodos



The image shows a screenshot of a Visual Studio IDE with two panes. The left pane displays the source code for `PrimeiroPrograma.Form1`. The right pane shows the Solution Explorer with the `button1_Click(object sender, EventArgs e)` event handler selected.

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

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

The Solution Explorer on the right lists the following items:

- `button1_Click(object sender, EventArgs e)` (selected)
- `button2`
- `comboBox1`
- `components`
- `Dispose(bool disposing)`
- `Form1()`
- `Form1_Load(object sender, EventArgs e)`
- `InitializeComponent()`
- `label1`
- `label2`
- `label3`
- `label4`
- `label5`

MÉTODOS



Programa - Eventos

The image shows a software interface with a blue background. On the left, a window titled "Primeiro Programa" contains a form titled "Dados Pessoais". The form has four input fields: "Nome", "Cargo", "Telefone", and "Departamento" (a dropdown menu). At the bottom of the form are two buttons: "Cadastrar" and "Sair".

On the right, a yellow box contains the text "Mensagem do Sistema para o Usuário." with a yellow arrow pointing down towards a smaller dialog box at the bottom center. This dialog box is titled "Sistema de Cadastro" and contains a question mark icon followed by the text "DESEJA SAIR DO SISTEMA ?". At the bottom of the dialog are three buttons: "Sim", "Não", and "Cancelar".

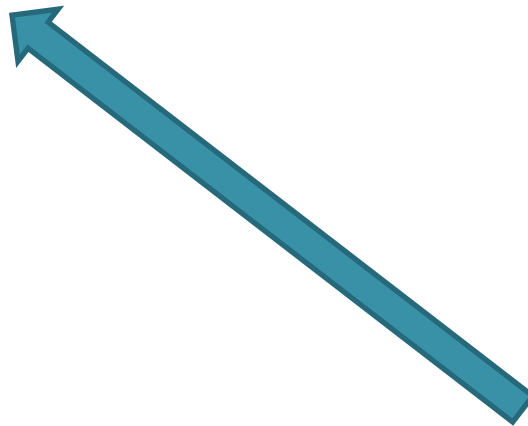


Programa – Código Eventos

```
private void button2_Click(object sender, EventArgs e)
{
    string mensagem = "DESEJA SAIR DO SISTEMA ?";
    string titulo = "Sistema de Cadastro";
    MessageBoxButtons buttons = MessageBoxButtons.YesNoCancel;
    DialogResult result;

    result = MessageBox.Show(mensagem, titulo, buttons, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);

    if (result == DialogResult.Yes)
    {
        this.Close();
    }
}
```



Código Botão – Sair
Com sistema de resposta...

Exercício 3

- 1) Na empresa onde trabalhamos, há tabelas com o quanto foi gasto em cada mês. Para fechar o balanço do primeiro trimestre, precisamos somar o gasto total. Sabendo que, em Janeiro, foram gastos 15000 reais, em Fevereiro, 23000 reais e em Março, 17000 reais, faça um programa que calcule e imprima o gasto total no trimestre.
- 2) Construa uma janela para o sistema.

Referências

- <https://docs.microsoft.com/pt-br/dotnet/csharp/whats-new/csharp-version-history>
- <https://docs.microsoft.com/pt-br/dotnet/fundamentals/implementations>
- <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/>
- <https://docs.microsoft.com/pt-br/dotnet/core/whats-new/dotnet-6>
- <https://docs.microsoft.com/pt-br/dotnet/core/introduction>



Muito Obrigado

Até a Próxima Aula

Prof. Laércio Silva
Email: Indsilva@hotmail.com