

Técnicas de Programação II



**Curso Superior de Tecnologia em Desenvolvimento de
Software Multiplataforma**

Aula 03

Prof. Claudio Benossi



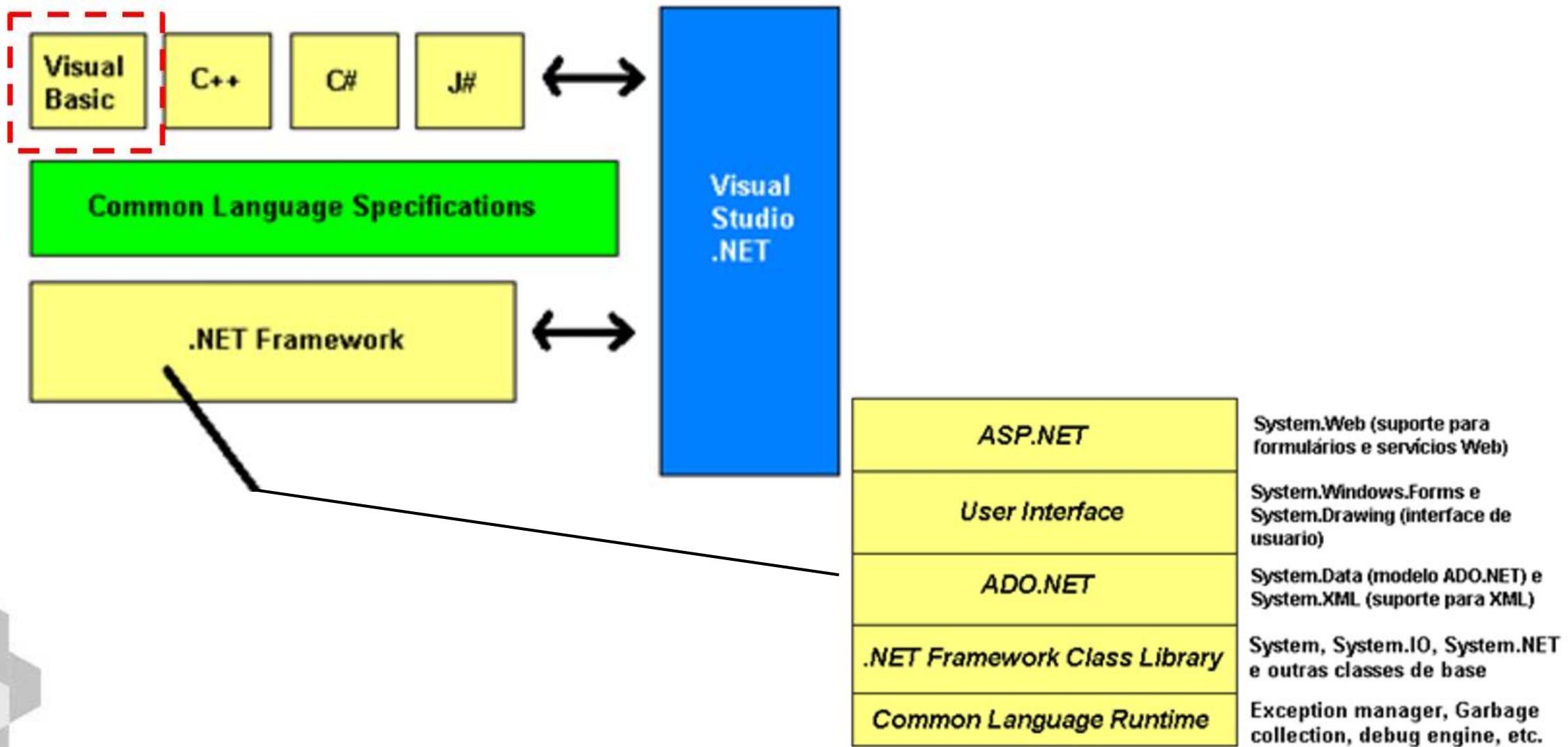
1.

Unidade

**Conhecendo o Visual Studio
Exemplos em VB**



VB.Net com Visual Studio



VB.Net com Visual Studio

Objetivo da aula hoje:

Trabalhar com os conceitos de Orientação a Objetos

Java x VB.Net

- Estrutura de decisão

Java

```
if (condição)
{
    // instruções
}
else
{
    // instruções
}
```

VB.Net

```
if condição then
    ' instruções
else
    ' instruções
end if
```

Java x VB.Net

- Estrutura de decisão

Java

```
switch (valor) {  
    case condição 1:  
        // instruções  
        break;  
    case condição 2:  
        // instruções  
        break;  
    ...  
    default;  
        // instruções  
    }  
}
```

VB.Net

```
select case valor  
case condição 1  
    ' instruções  
case condição 2  
    ' instruções  
...  
case else  
    ' instruções  
end select
```

Java x VB.Net

- Estrutura de repetição

Java

```
while (condição) {  
    // instruções  
}
```

VB.Net

```
while condição  
    ' instruções  
end while
```

Java x VB.Net

- Estrutura de repetição

Java

```
do {  
    // instruções  
} while (condição);
```

VB.Net

```
do  
    ' instruções  
loop while condição
```

Java x VB.Net

- Estrutura de repetição

Java

```
for (inicialização;condição/incremento){  
    // instruções  
}
```

VB.Net

```
for inicialização to valorFinal <step repetição> (opcional)  
    ' instruções  
Next
```

Java x VB.Net

- Procedimentos / Funções / Métodos

Java

```
modificador void nome(parâmetros) {  
    // instruções  
}
```

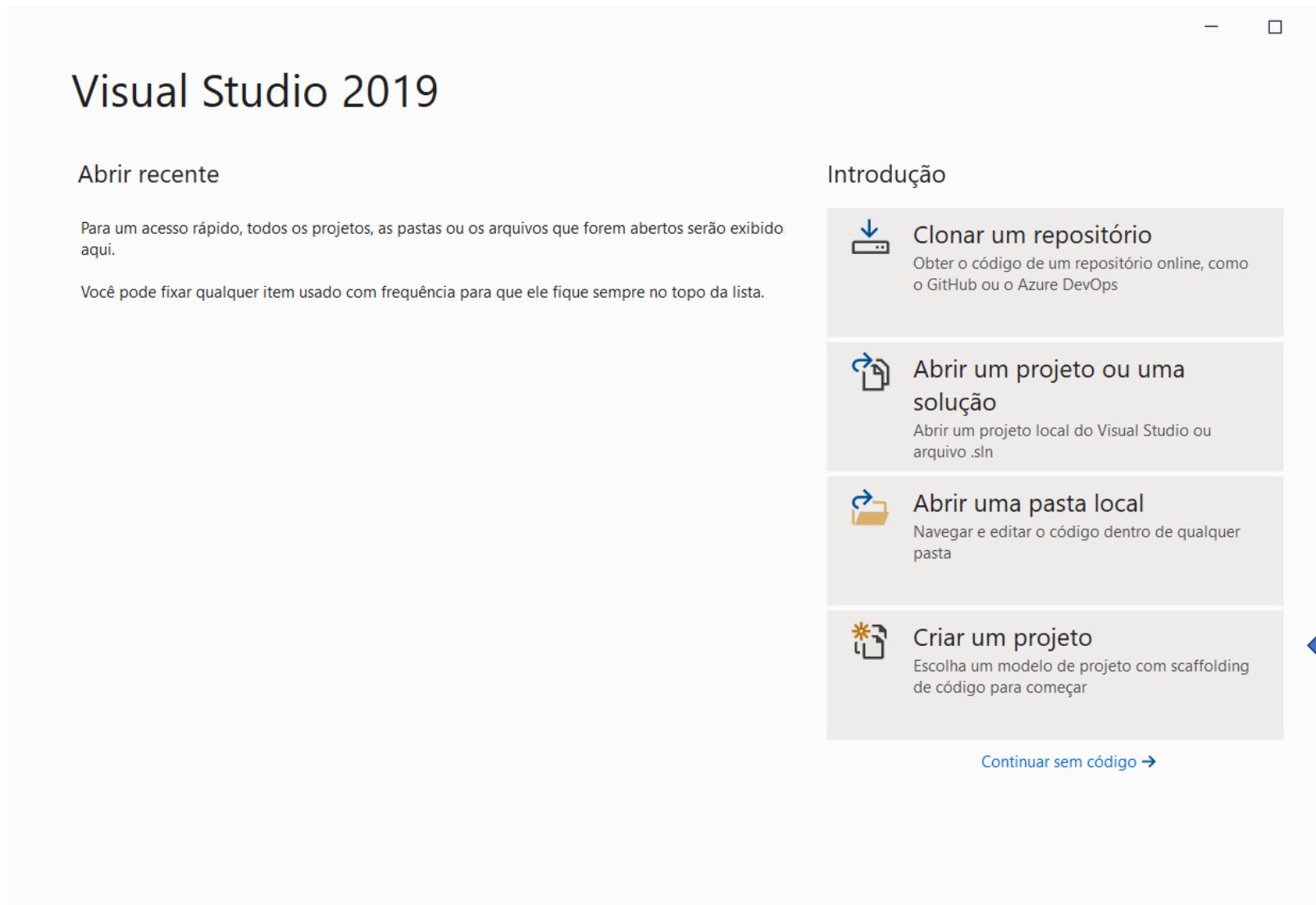
```
modificador tipo nome(parâmetros) {  
    // instruções  
    return <valor>;  
}
```

VB.Net

```
sub nome (parâmetros)  
    ' instruções  
end sub
```

```
function nome (parâmetros)  
    ' instruções  
    return <valor>  
end function
```

Primeiro projeto VB.NET



Visual Studio 2019

Abrir recente

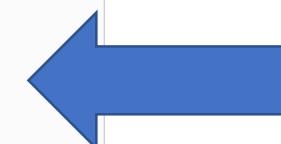
Para um acesso rápido, todos os projetos, as pastas ou os arquivos que forem abertos serão exibidos aqui.

Você pode fixar qualquer item usado com frequência para que ele fique sempre no topo da lista.

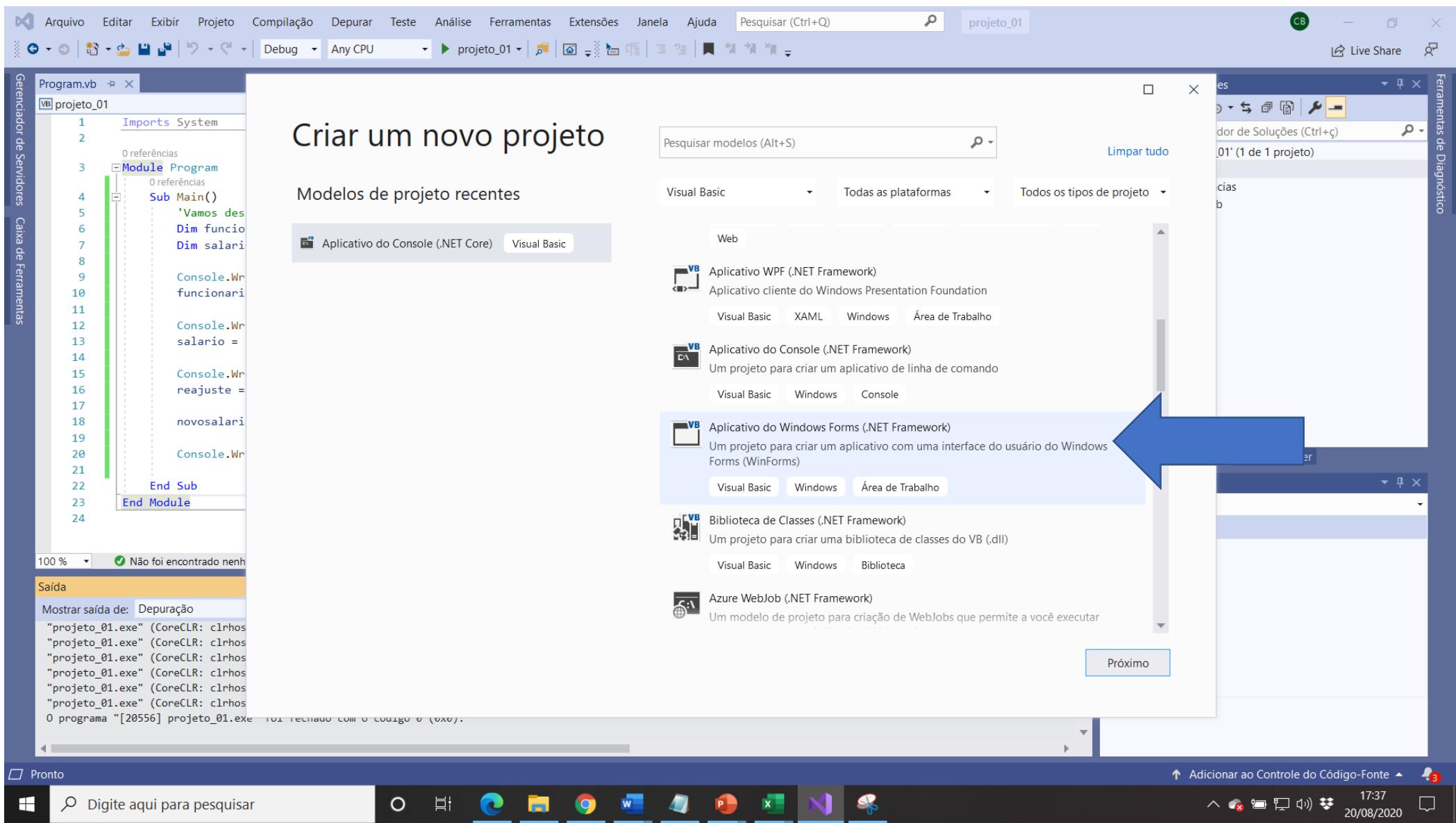
Introdução

-  Clonar um repositório
Obter o código de um repositório online, como o GitHub ou o Azure DevOps
-  Abrir um projeto ou uma solução
Abrir um projeto local do Visual Studio ou arquivo .sln
-  Abrir uma pasta local
Navegar e editar o código dentro de qualquer pasta
-  Criar um projeto
Escolha um modelo de projeto com scaffolding de código para começar

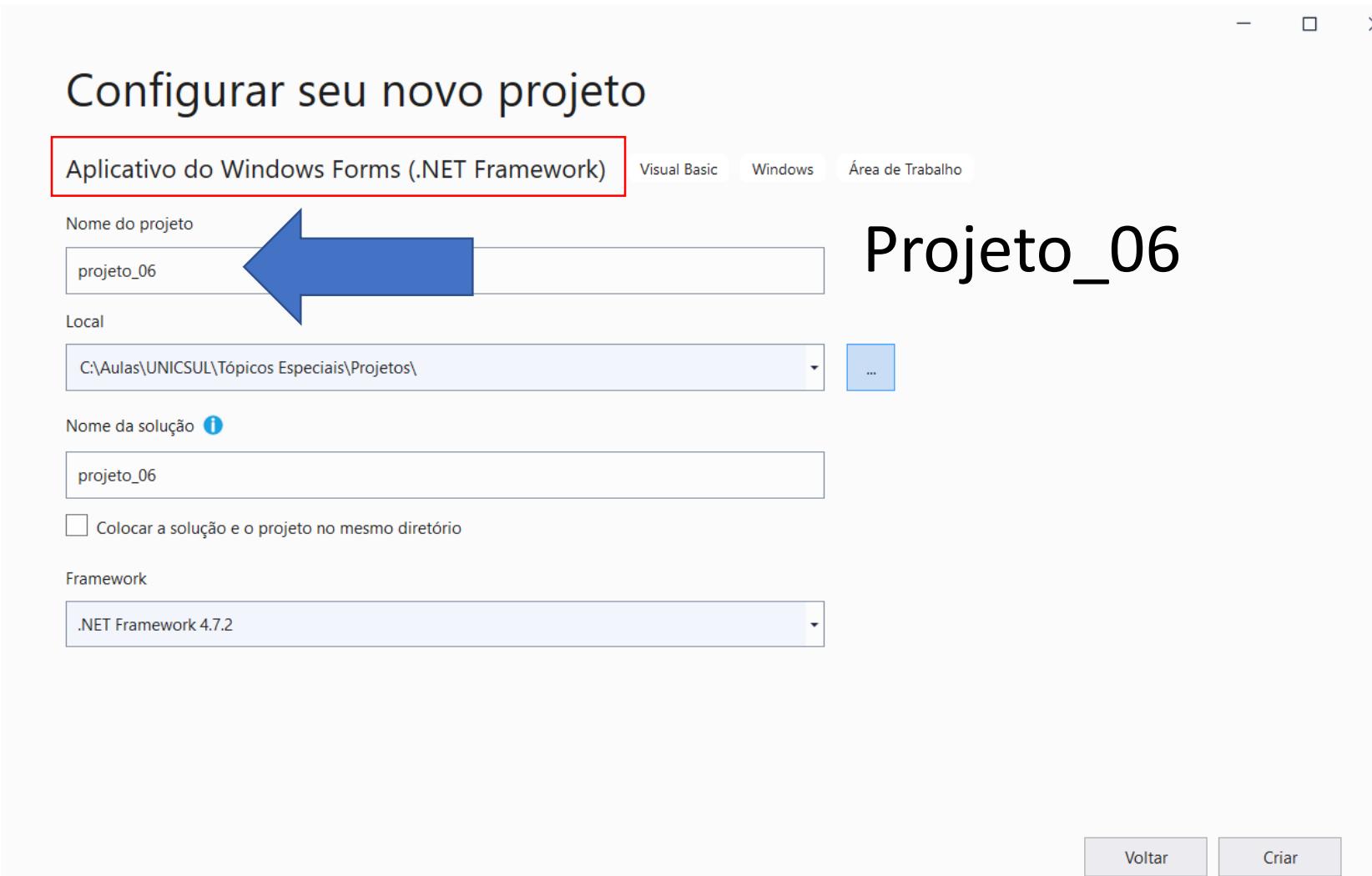
Continuar sem código →



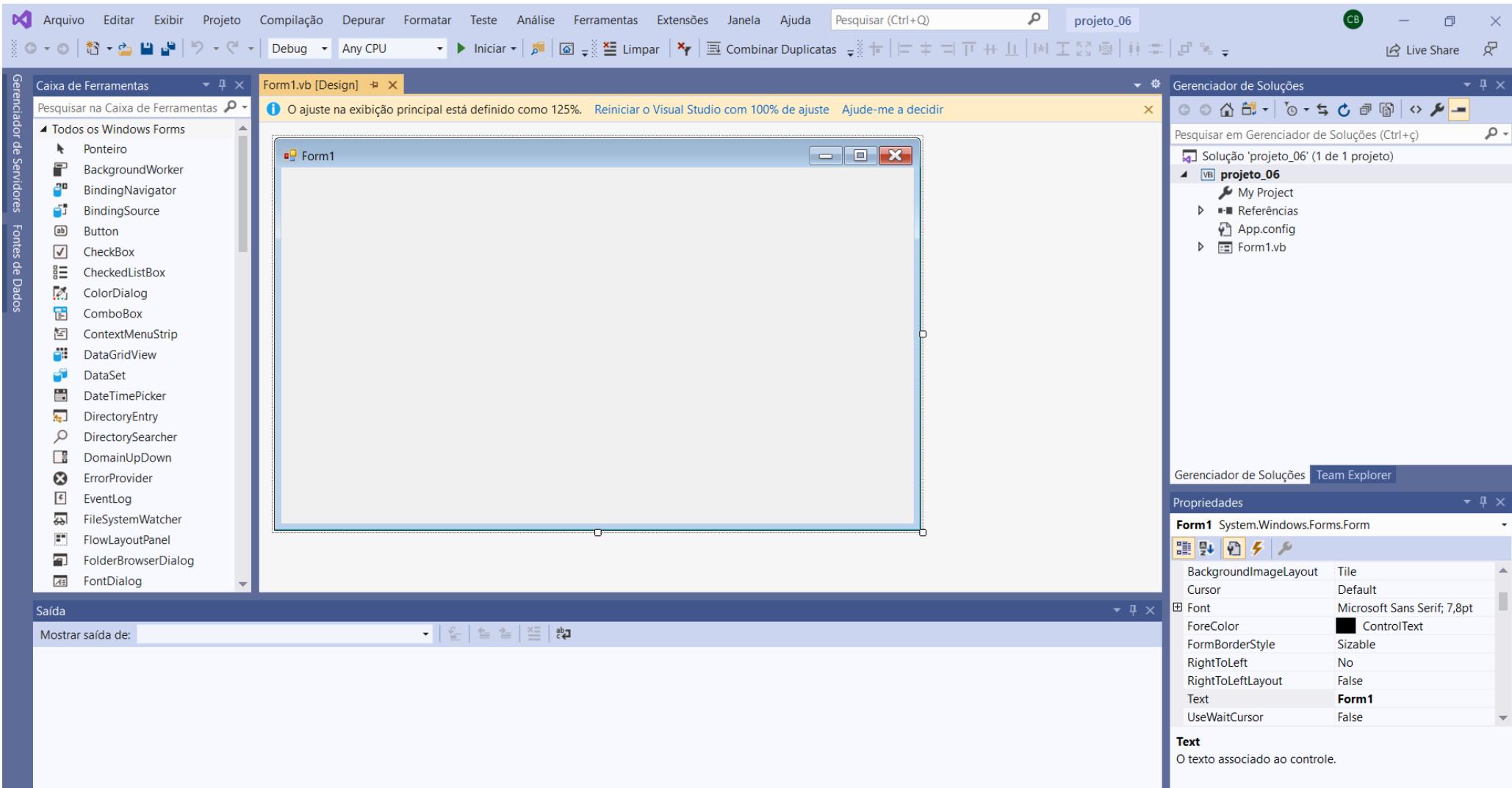
Projeto com interface gráfica VB.NET



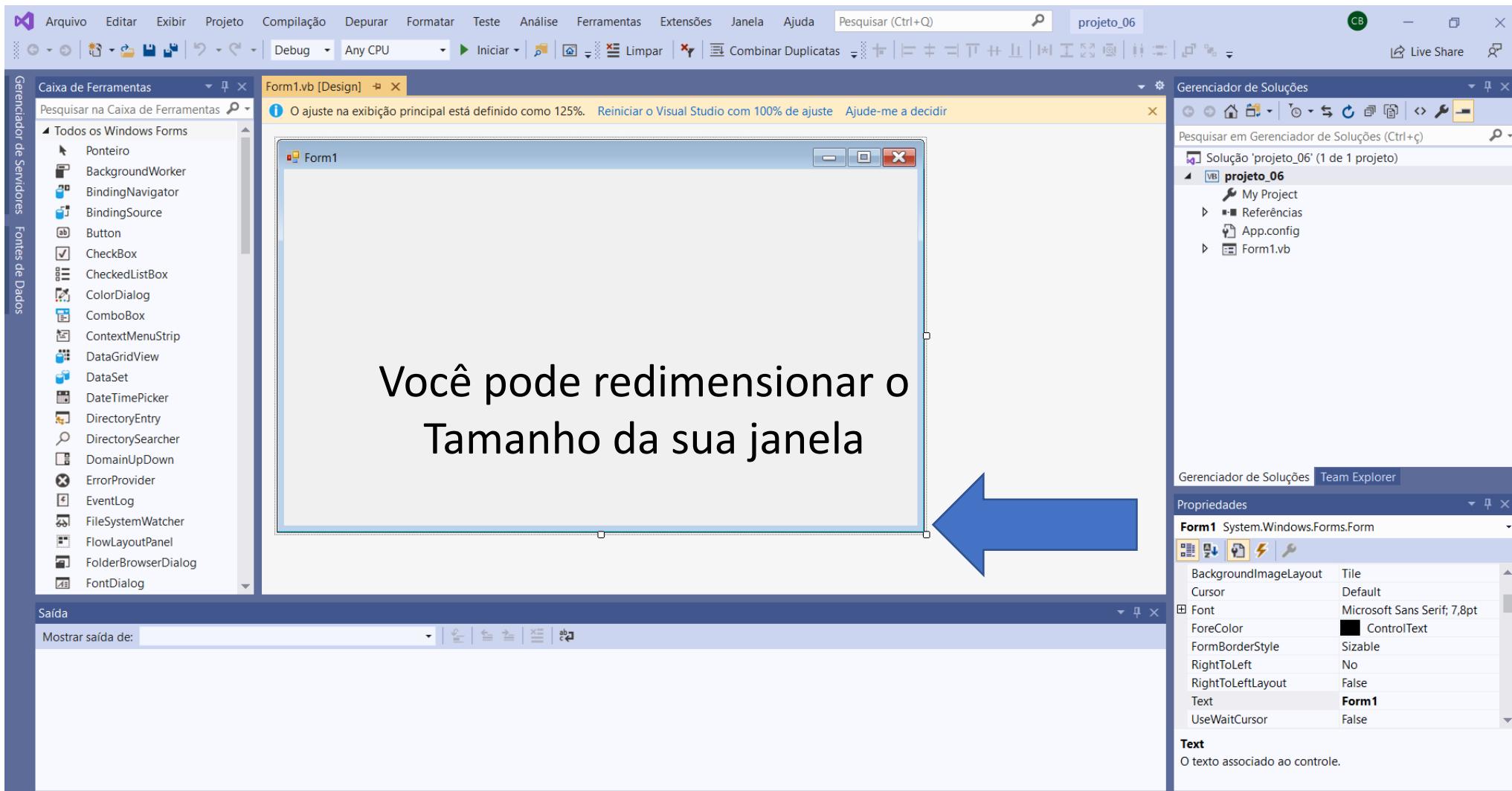
Projeto com interface gráfica VB.NET



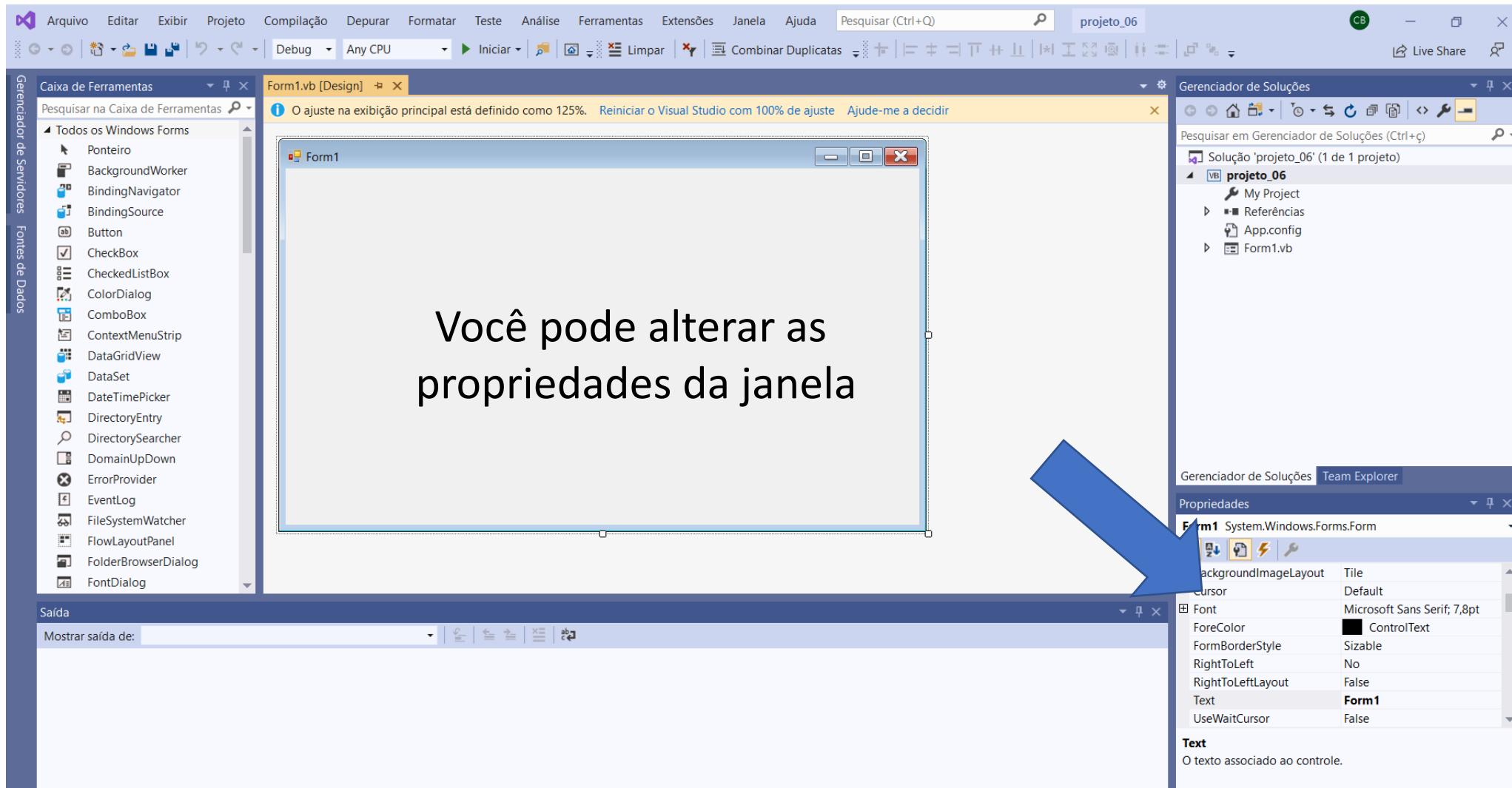
Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET

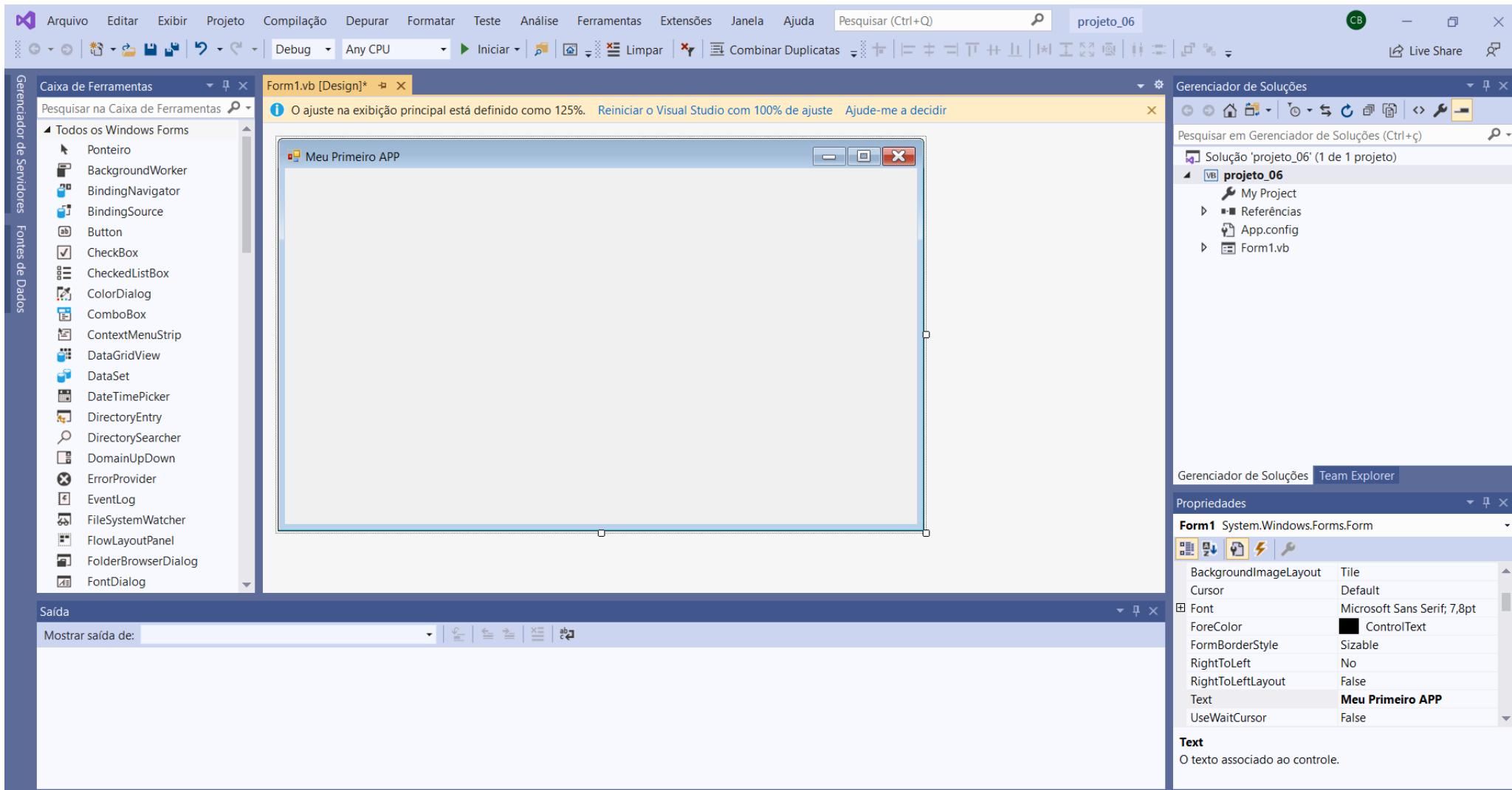


Projeto com interface gráfica VB.NET

Vamos alterar a propriedade Text da janela:

Meu Primeiro APP

Projeto com interface gráfica VB.NET

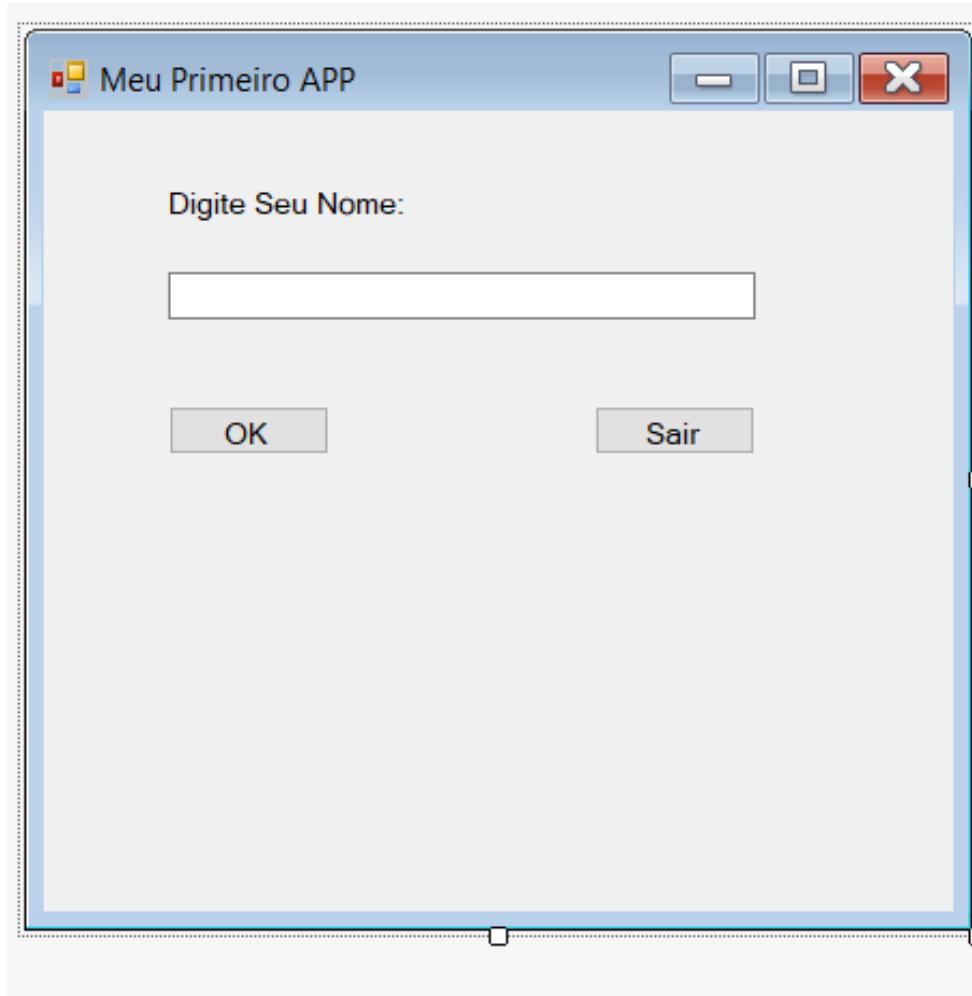


Projeto com interface gráfica VB.NET

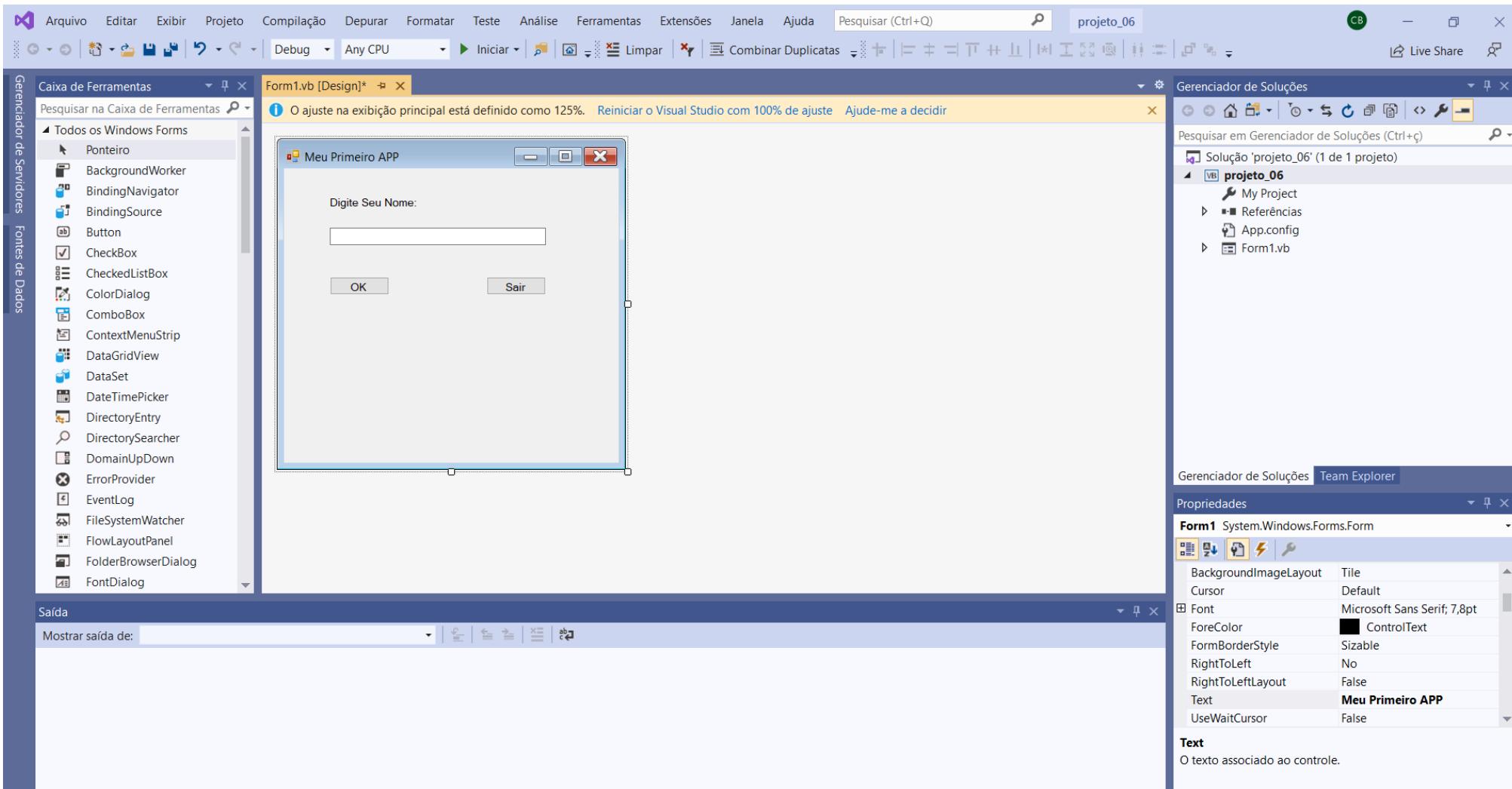
Agora vamos montar um layout simples com:

- 01 Label
- 01 TextBox
- 02 Buttons

Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET

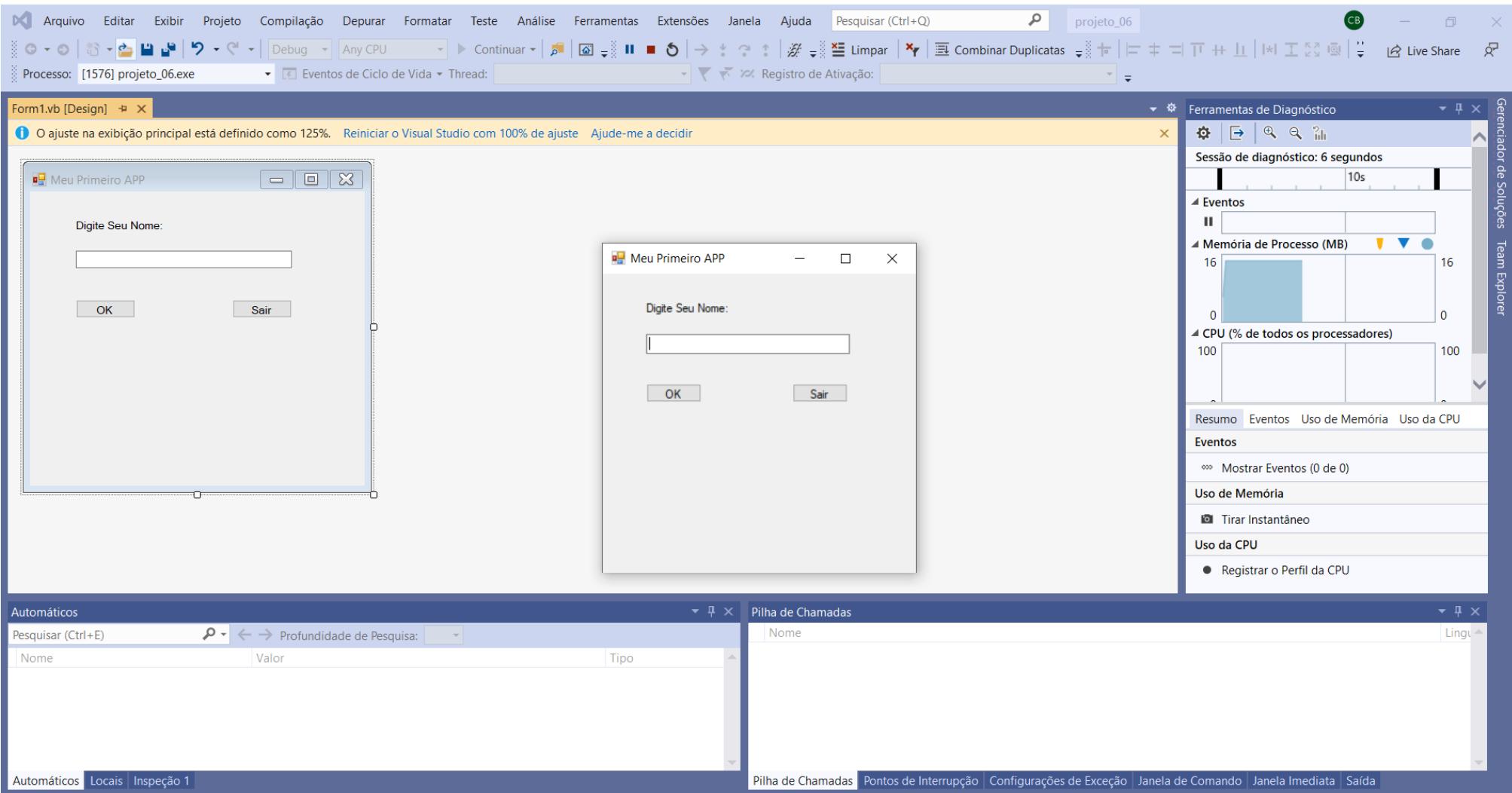


Projeto com interface gráfica VB.NET

Para que possamos centralizar a janela durante a execução vamos alterar a propriedade da janela:

StarPosition = CenterScreen

Projeto com interface gráfica VB.NET

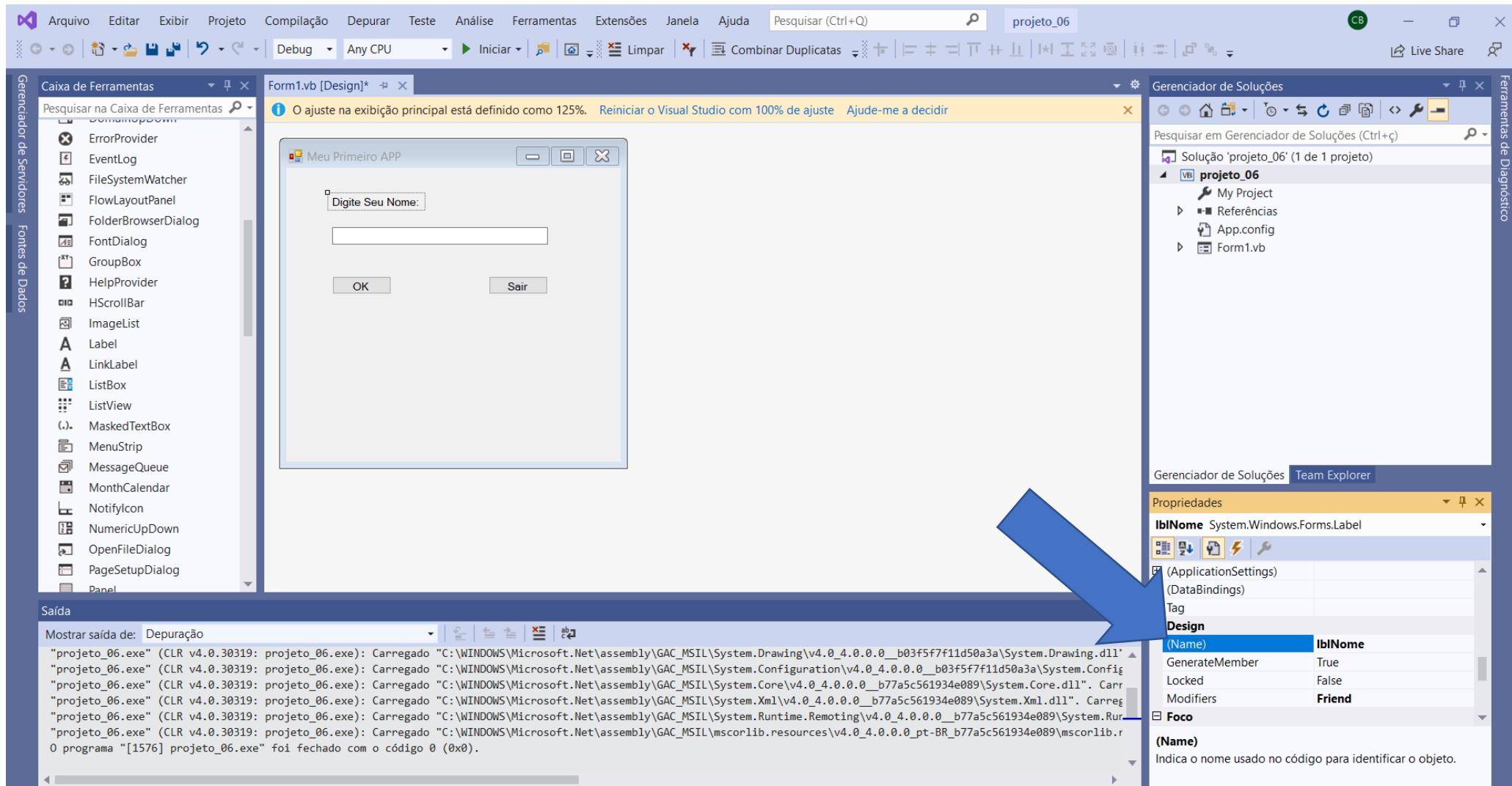


Projeto com interface gráfica VB.NET

Agora vamos renomear nossos objetos:

- Objeto Label vamos mudar o name para: **lblNome**

Projeto com interface gráfica VB.NET

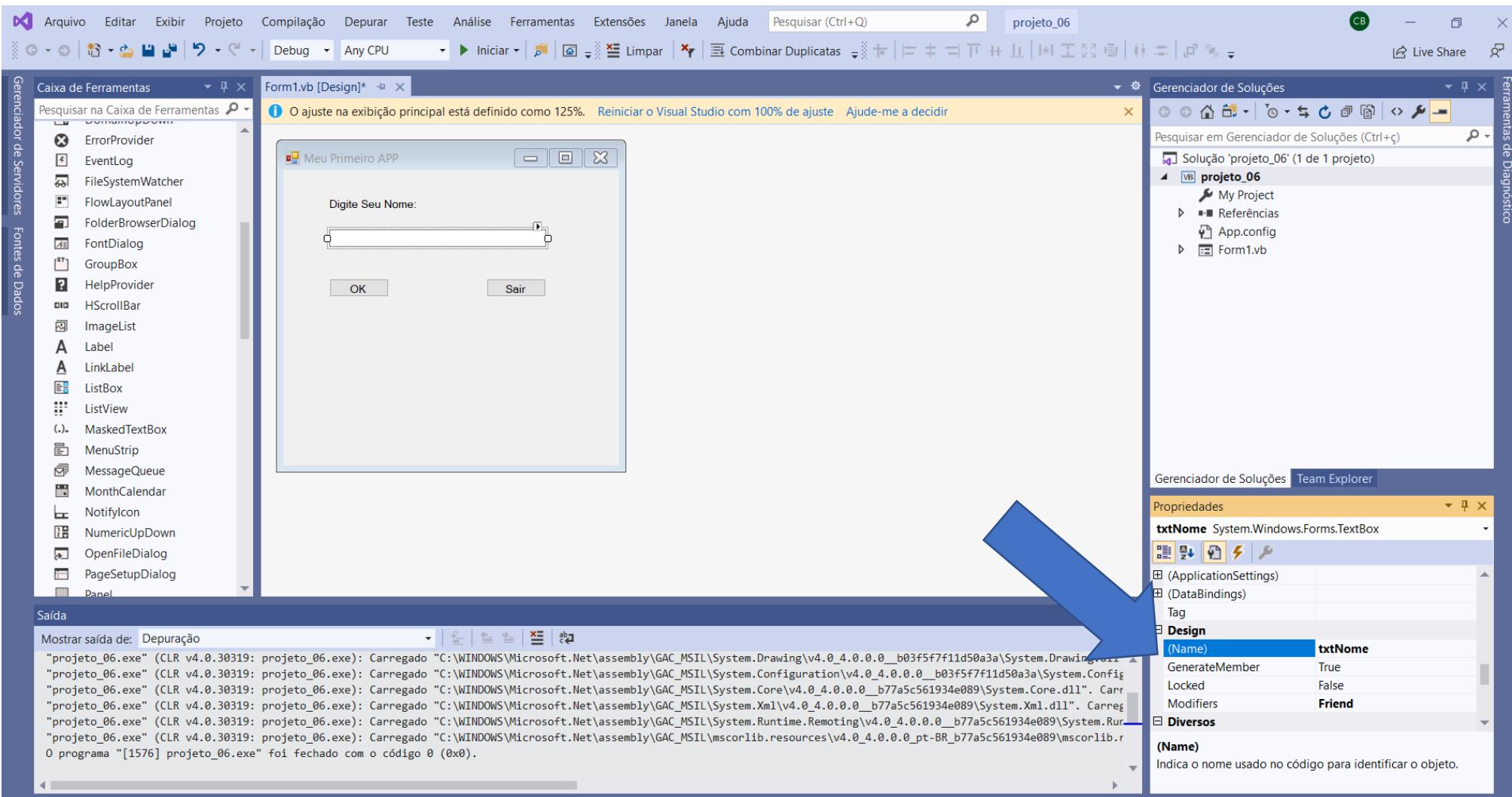


Projeto com interface gráfica VB.NET

Agora vamos renomear nossos objetos:

- Objeto textBox vamos mudar o name para:
txtNome

Projeto com interface gráfica VB.NET



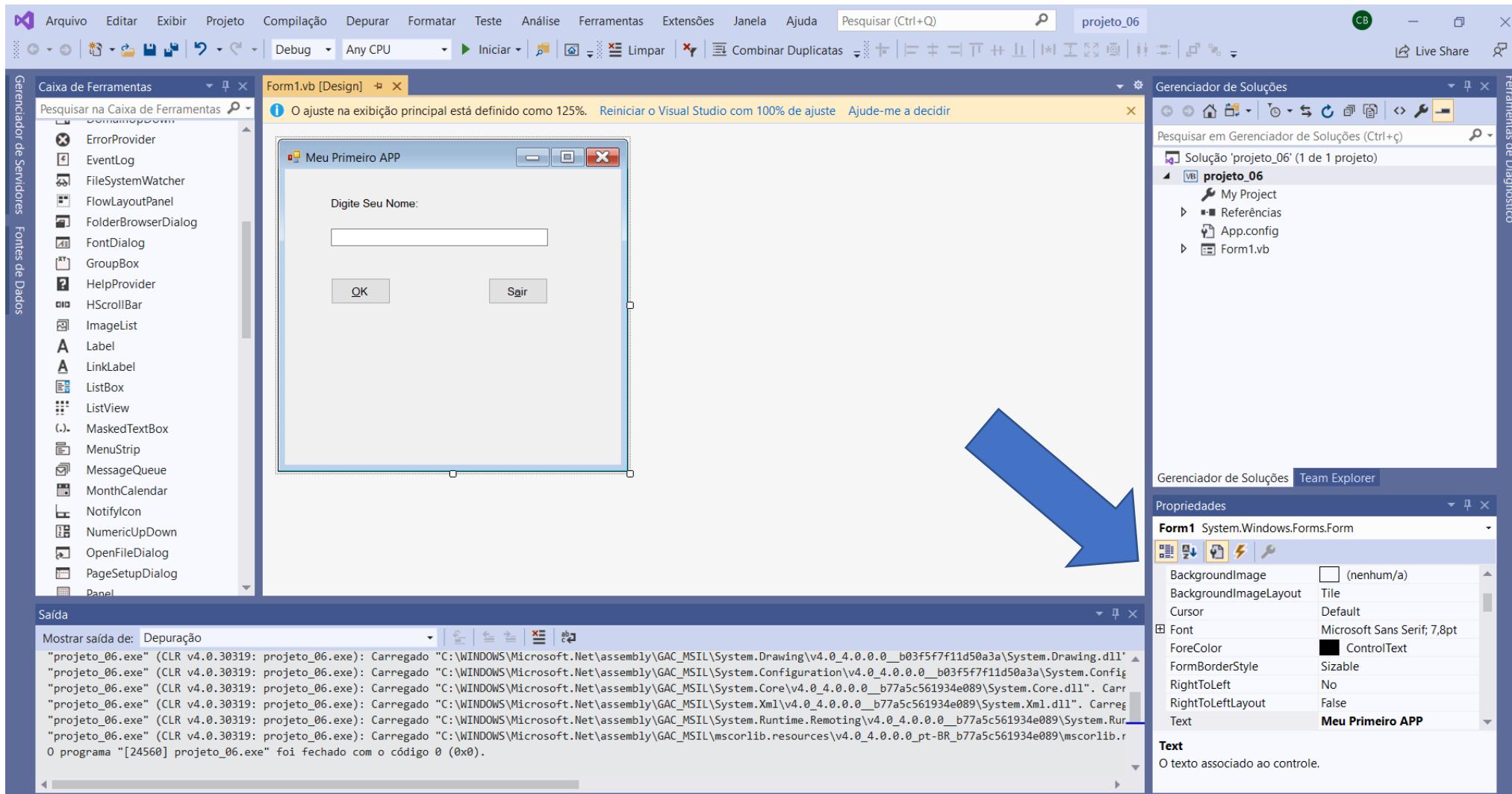
Projeto com interface gráfica VB.NET

Agora vamos renomear nossos objetos:

- Objetos Button vamos mudar o name para: **btnOK** e **btnSair**
- Vamos alterar também o text para: **&OK** e **S&air**

Dessa forma estamos criando atalhos para os botões.

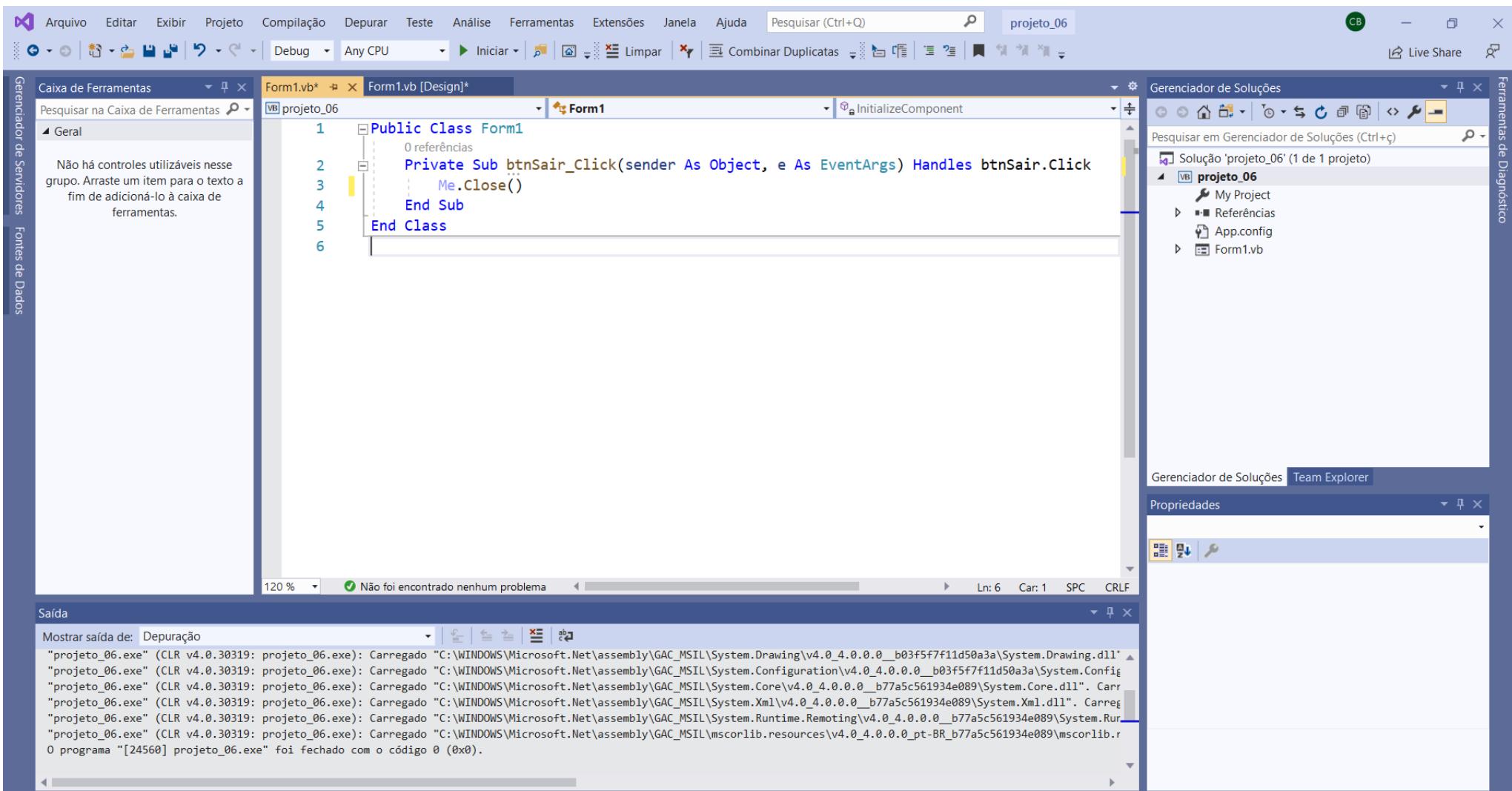
Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET

Vamos inserir o código no botão sair, para isso clique sobre o botão sair:

Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET

Vamos inserir o código no botão OK, para isso clique sobre o botão OK:

Projeto com interface gráfica VB.NET



Form1.vb* X Form1.vb [Design]*

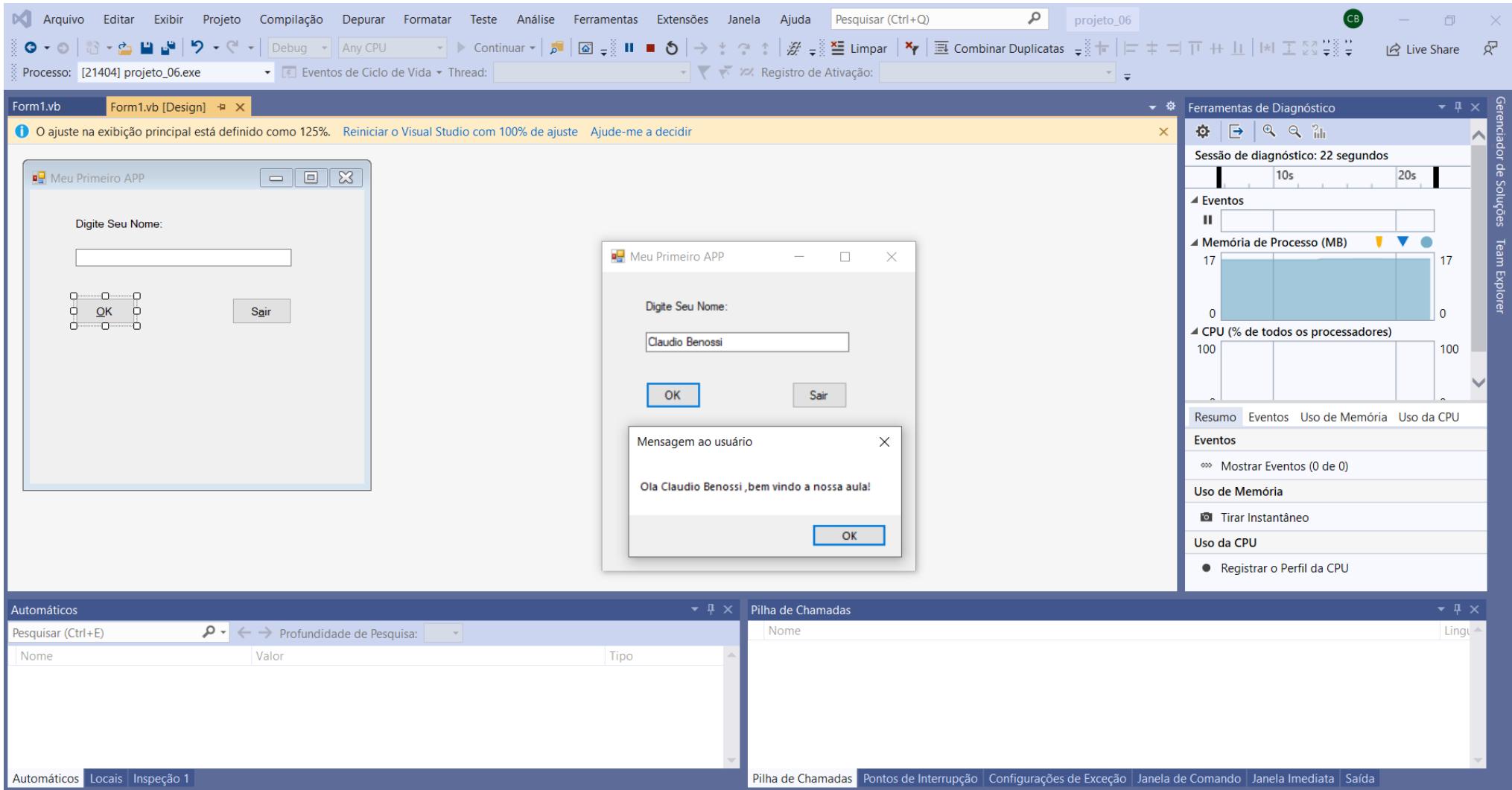
VB projeto_06 Form1 InitializeComponent

```
1 referência
1  Public Class Form1
2      0 referências
3          2 - Private Sub btnSair_Click(sender As Object, e As EventArgs) Handles btnSair.Click
4              Me.Close()
5          End Sub
6
7          0 referências
8              2 - Private Sub btnOK_Click(sender As Object, e As EventArgs) Handles btnOK.Click
9                  'revisão da aula anterior, vamos exibir uma janela de mensagem
10                 MessageBox.Show("Olá " & txtNome.Text & ", bem vindo à nossa aula!", "Mensagem ao usuário")
11             End Sub
12         End Class
```

Projeto com interface gráfica VB.NET

Agora é só testar!!!

Projeto com interface gráfica VB.NET



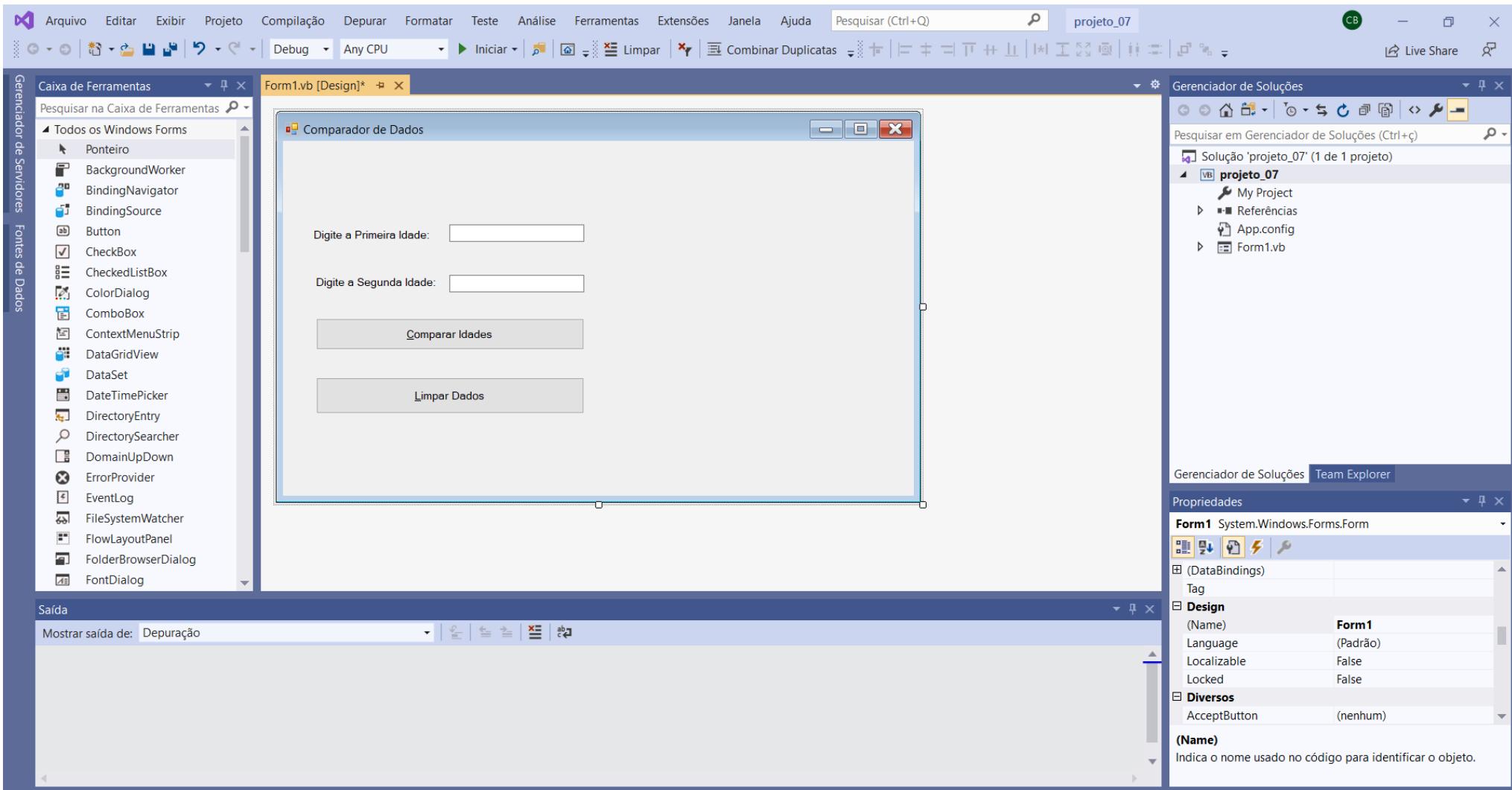
VB.NET

Agora vamos criar um novo projeto como projeto_07.

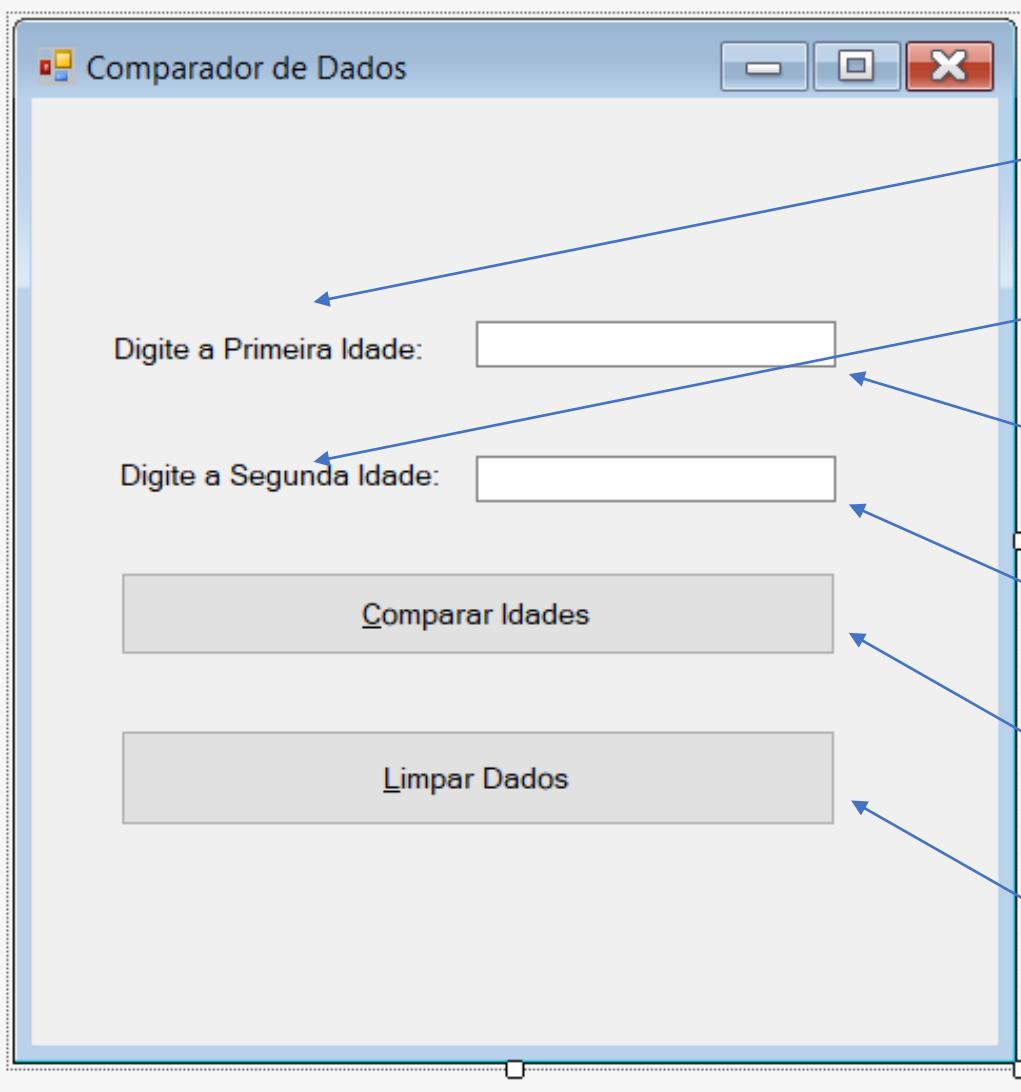
36

Objetivo: comparar dados

Projeto com interface gráfica VB.NET



Projeto com interface gráfica VB.NET



lblIdade1

lblIdade2

tbtIdade1

tbtIdade2

btnComparar

btbLimpar

VB.NET

Agora vamos inserir o código no botão Comparar Idade:

Projeto com interface gráfica VB.NET

```
1 referência
1  -Public Class Form1
2      0 referências
3          2  Private Sub btnComparar_Click(sender As Object, e As EventArgs) Handles btnComparar.Click
4              ' criando as variáveis que serão utilizadas no processo
5              Dim idade1, idade2 As Integer
6              ' inserindo valores nas variáveis, recuperando dos objetos TextBox
7              ' convertendo o conteúdo do textBox que é uma string para números inteiros com a função CInt
8              idade1 = CInt(tbtIdade1.Text)
9              idade2 = CInt(tbtIdade2.Text)
10
11
12      End Sub
13  End Class
14
```

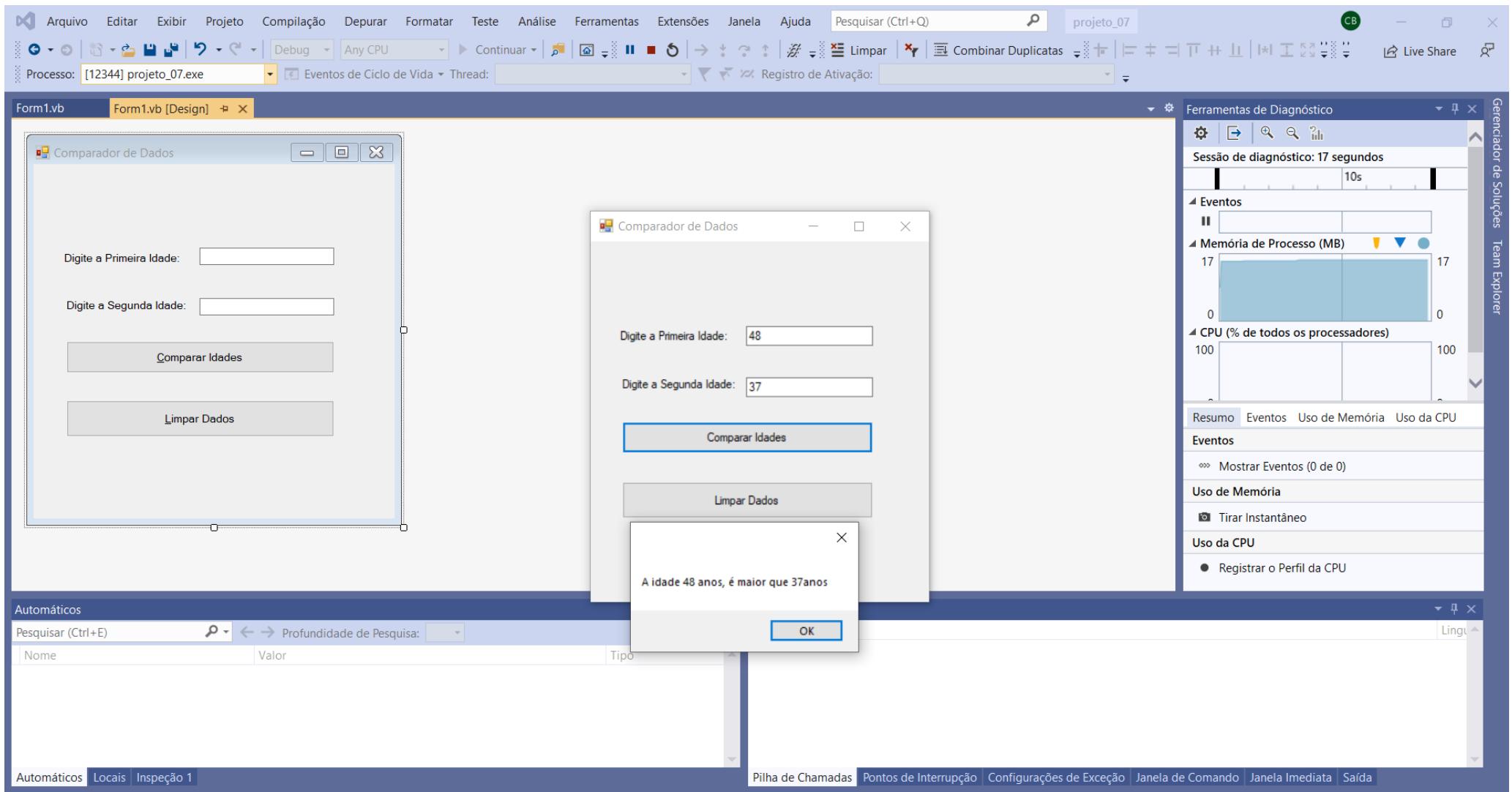
Projeto com interface gráfica VB.NET

```
1 referência
1  Public Class Form1
2      0 referências
3          Private Sub btnComparar_Click(sender As Object, e As EventArgs) Handles btnComparar.Click
4              ' criando as variáveis que serão utilizadas no processo
5              Dim idade1, idade2 As Integer
6              ' inserindo valores nas variáveis, recuperando dos objetos TextBox
7              ' convertendo o conteúdo do textBox que é uma string para números inteiros com a função Cint
8              idade1 = CInt(tbtIdade1.Text)
9              idade2 = CInt(tbtIdade2.Text)
10
11             ' vamos comparar os valores das variáveis usando a estrutura de decisão
12             If idade1 > idade2 Then
13                 MessageBox.Show("A idade " & idade1 & " anos, é maior que " & idade2 & "anos")
14             ElseIf idade2 > idade1 Then
15                 MessageBox.Show("A idade " & idade2 & " anos, é maior que " & idade1 & "anos")
16             Else
17                 MessageBox.Show("A idade " & idade1 & " anos, é igual a idade " & idade2 & "anos")
18             End If
19
20         End Sub
21     End Class
```

VB.NET

Podemos testar:

Projeto com interface gráfica VB.NET



VB.NET

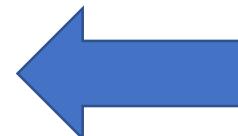
Agora vamos inserir o código no botão limpar dados:

Projeto com interface gráfica VB.NET

```
20  
21 0 referências  
22 Private Sub btnLimpar_Click(sender As Object, e As EventArgs) Handles btnLimpar.Click  
23     tbtIdade1.Clear()  
24     tbtIdade2.Clear()  
25 End Sub  
26 End Class  
27
```

VB.NET

```
9  
10   ' vamos comparar os valores das variáveis usando a estrutura de decisão  
11   If idade1 > idade2 Then  
12     MessageBox.Show("A idade " & idade1 & " anos, é maior que " & idade2 & "anos")  
13   ElseIf idade2 > idade1 Then  
14     MessageBox.Show("A idade " & idade2 & " anos, é maior que " & idade1 & "anos")  
15   Else  
16     MessageBox.Show("A idade " & idade1 & " anos, é igual a idade " & idade2 & "anos")  
17   End If  
18  
19   tbtIdade1.Clear()  
20   tbtIdade2.Clear()  
21  
22 End Sub
```



```
23  
24  
25  
26  
27  
28  
29
```

```
0 referências  
24   Private Sub btnLimpar_Click(sender As Object, e As EventArgs) Handles btnLimpar.Click  
25     tbtIdade1.Clear()  
26     tbtIdade2.Clear()  
27   End Sub  
28 End Class
```



```
10      ' vamos comparar os valores das variáveis usando a estrutura de decisão
11      If idade1 > idade2 Then
12          MessageBox.Show("A idade " & idade1 & " anos, é maior que " & idade2 & "anos")
13      ElseIf idade2 > idade1 Then
14          MessageBox.Show("A idade " & idade2 & " anos, é maior que " & idade1 & "anos")
15      Else
16          MessageBox.Show("A idade " & idade1 & " anos, é igual a idade " & idade2 & "anos")
17      End If
18
19      LimparCampos() ←
20
21  End Sub
22
23  0 referências
24  Private Sub btnLimpar_Click(sender As Object, e As EventArgs) Handles btnLimpar.Click
25      LimparCampos() ←
26  End Sub
27
28  2 referências
29  Private Sub LimparCampos()
30      tbtIdade1.Clear()
31      tbtIdade2.Clear()
32  End Sub
33
34  End Class
```



VB.NET

Agora vamos criar um novo projeto, focando a Orientação a Objetos.

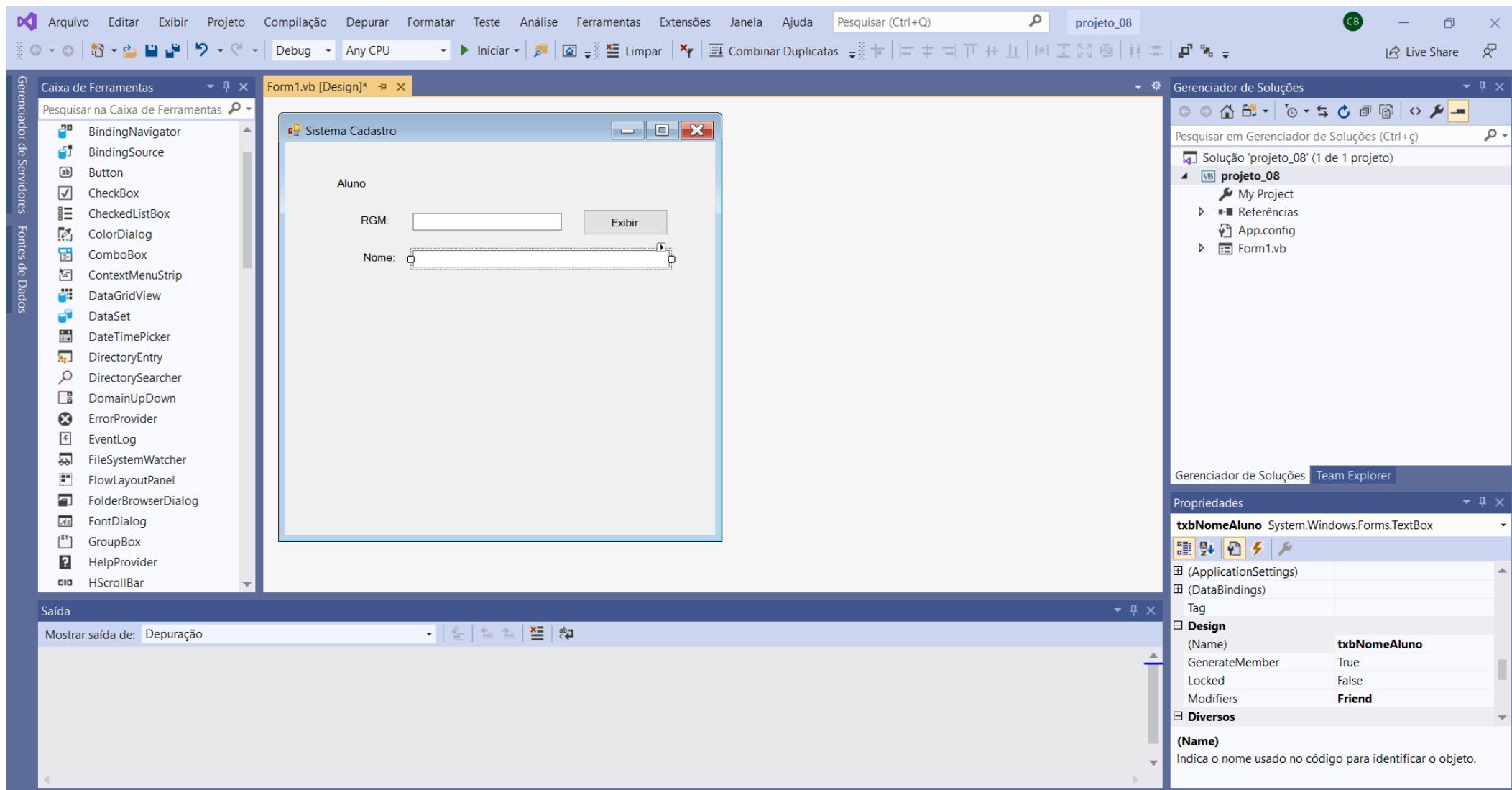
418
418

Projeto_08 – Sistema Cadastro

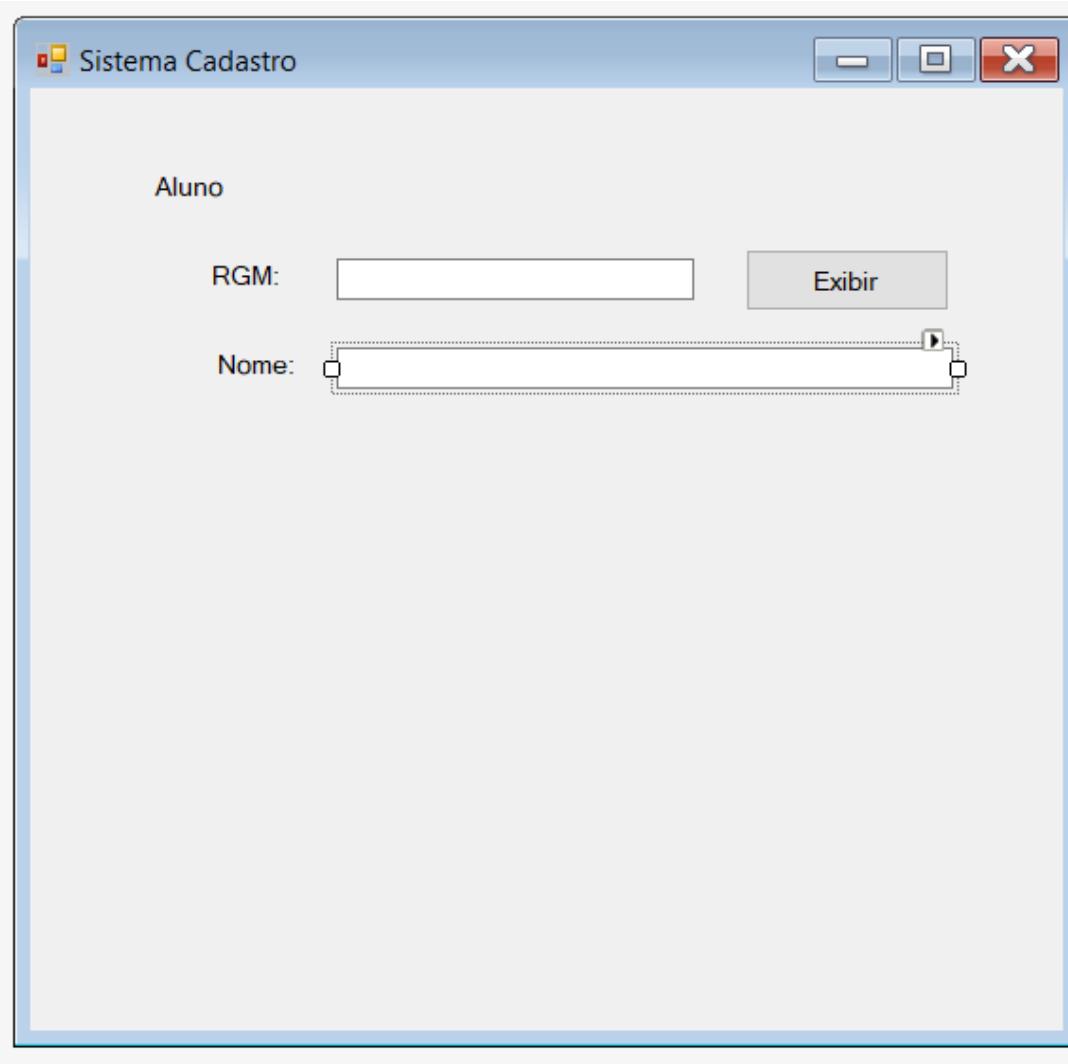
Vamos Criar Classes -

Pessoa
Aluno
Professor

VB.NET



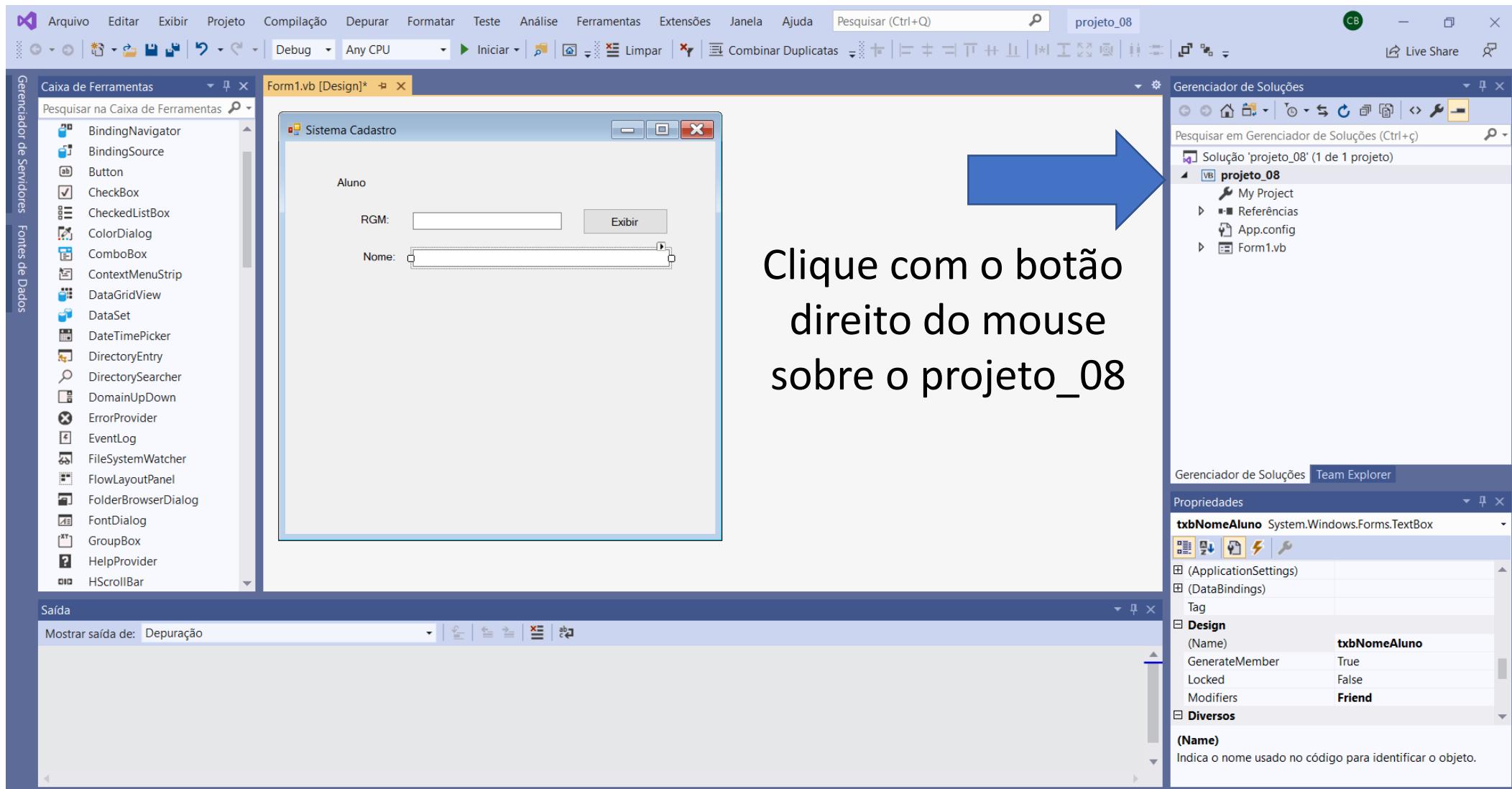
VB.NET



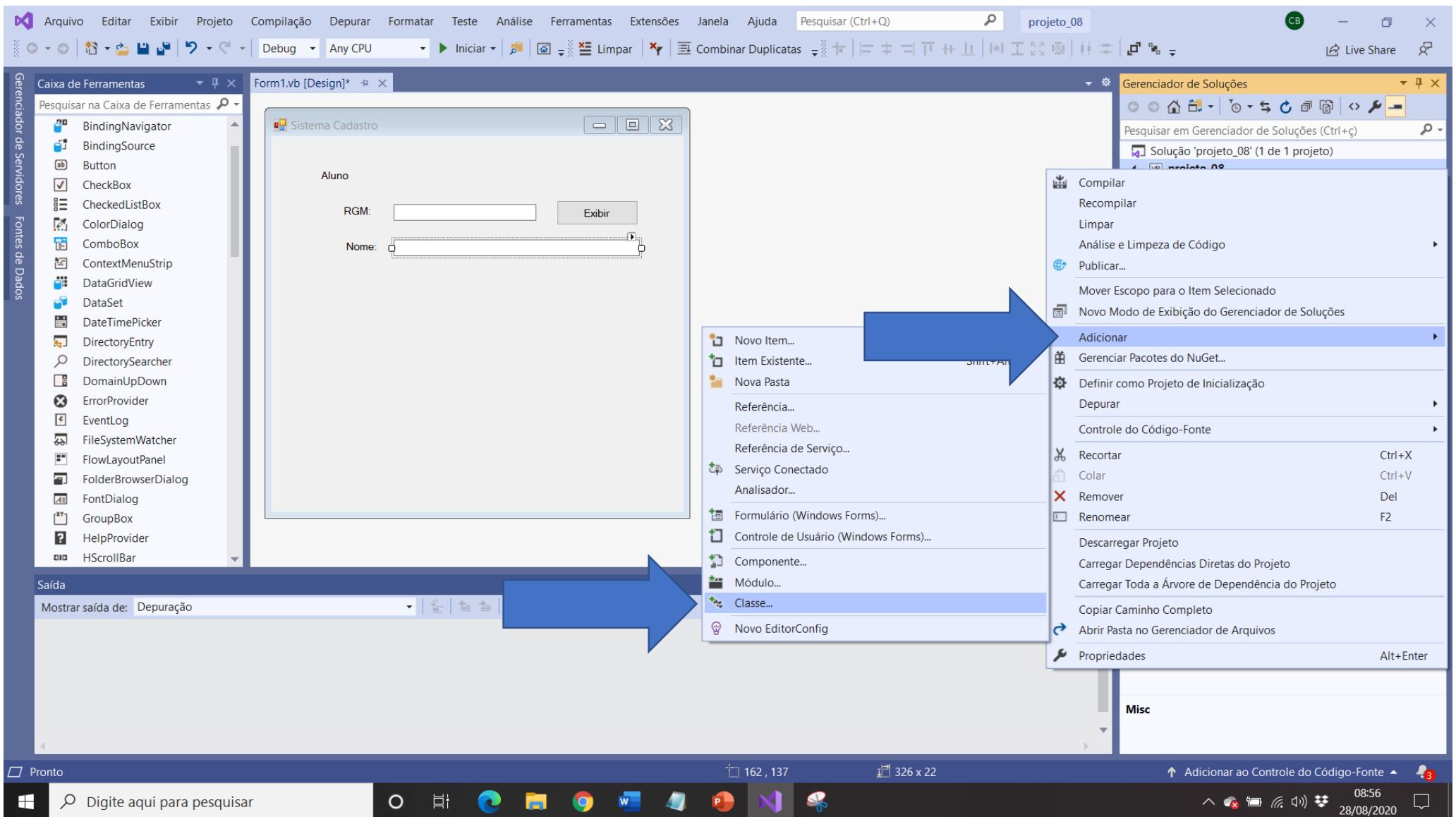
VB.NET

Vamos empregar o conceito de herança entre classes.

VB.NET



VB.NET



Adicionar Novo Item - projeto_08

? X

Instalados

Itens Comuns

- Geral
- SQL Server
- Web
 - Windows Forms
 - WPF
 - Dados
 - Código

Online

Classificar por:		Padrão	?	X
			Pesquisar (Ctrl+E)	?
	Classe	Itens Comuns	Tipo: Itens Comuns Uma definição de classe vazia	
	Módulo	Itens Comuns		
	Interface	Itens Comuns		
	Formulário (Windows Forms)	Itens Comuns		
	Controle de Usuário (Windows Forms)	Itens Comuns		
	Classe de Componente	Itens Comuns		
	Controle de Usuário (WPF)	Itens Comuns		
	ADO.NET Entity Data Model	Itens Comuns		
	Arquivo de Código	Itens Comuns		
	Arquivo de Configuração JSON do JavaScript	Itens Comuns		
	Arquivo de Configuração JSON do TypeScript	Itens Comuns		
	Arquivo de Manifesto do Aplicativo	Itens Comuns		
	Arquivo de Recursos	Itens Comuns		
	Arquivo de Texto	Itens Comuns		

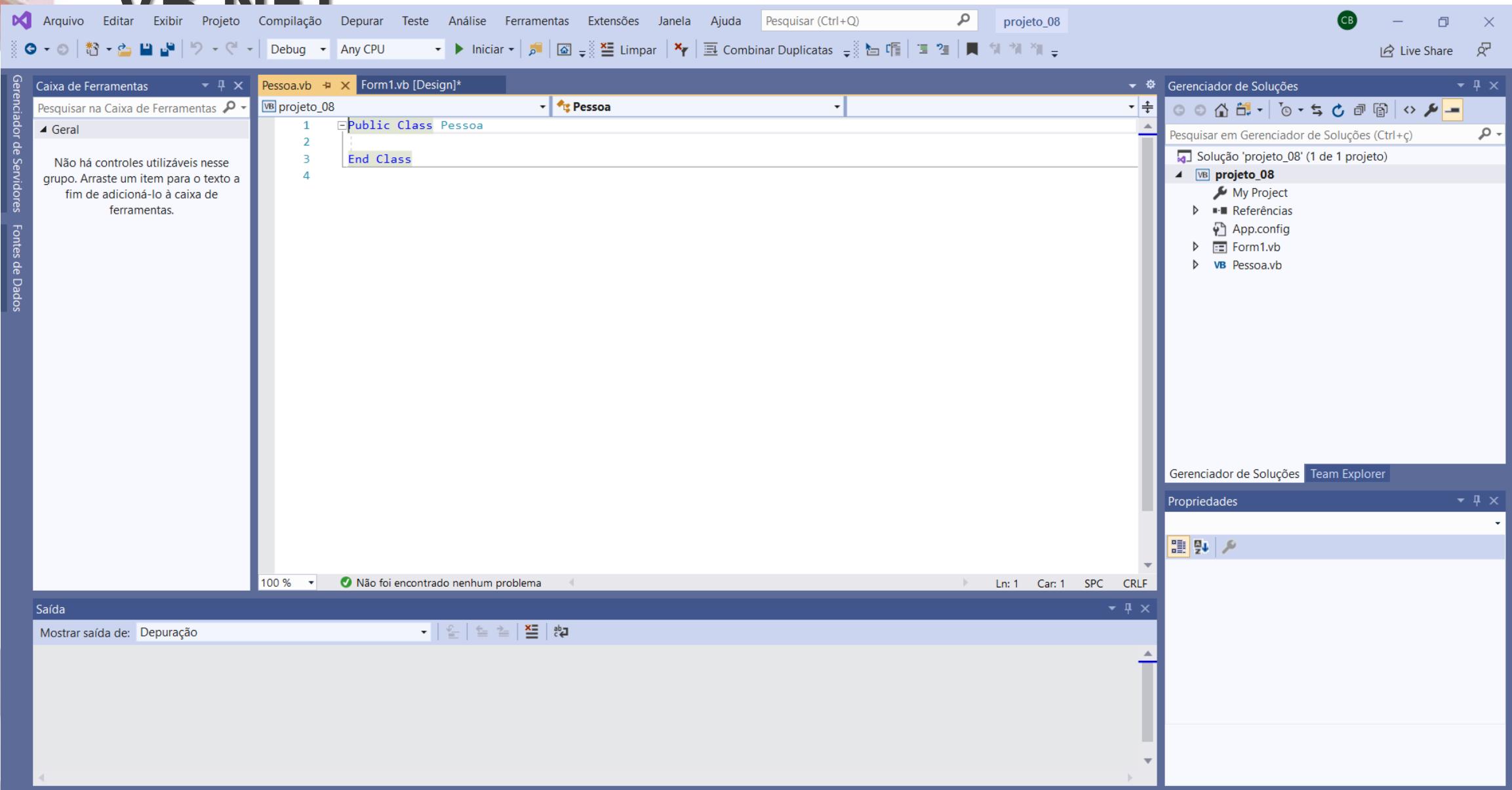
Nome:

Pessoavb

Adicionar

Cancelar

VB.NET



VB.NET

Repita o processo para criar a classe Aluno.



Arquivo Editar Exibir Projeto Compilação Depurar Teste Análise Ferramentas Extensões Janela Ajuda Pesquisar (Ctrl+Q) projeto_08

Debug Any CPU Iniciar Limpar Combinar Duplicatas Live Share

Caixa de Ferramentas Aluno.vb Pessoa.vb Form1.vb [Design]*

VB projeto_08 Aluno

```
1 Public Class Aluno
2 
3 End Class
4
```

Geral

Não há controles utilizáveis nesse grupo. Arraste um item para o texto a fim de adicioná-lo à caixa de ferramentas.

Gerenciador de Soluções

Solução 'projeto_08' (1 de 1 projeto)

- My Project
- Referências
- Aluno.vb
- App.config
- Form1.vb
- Pessoa.vb

Gerenciador de Soluções Team Explorer

Propriedades

100 % Não foi encontrado nenhum problema

Ln: 1 Car: 1 SPC CRLF

Saída

Mostrar saída de: Depuração

VB.NET

Agora vamos trabalhar com a classe Pessoa.

Aluno.vb

Pessoa.vb* ✎ X Form1.vb [Design]*

VB projeto_08

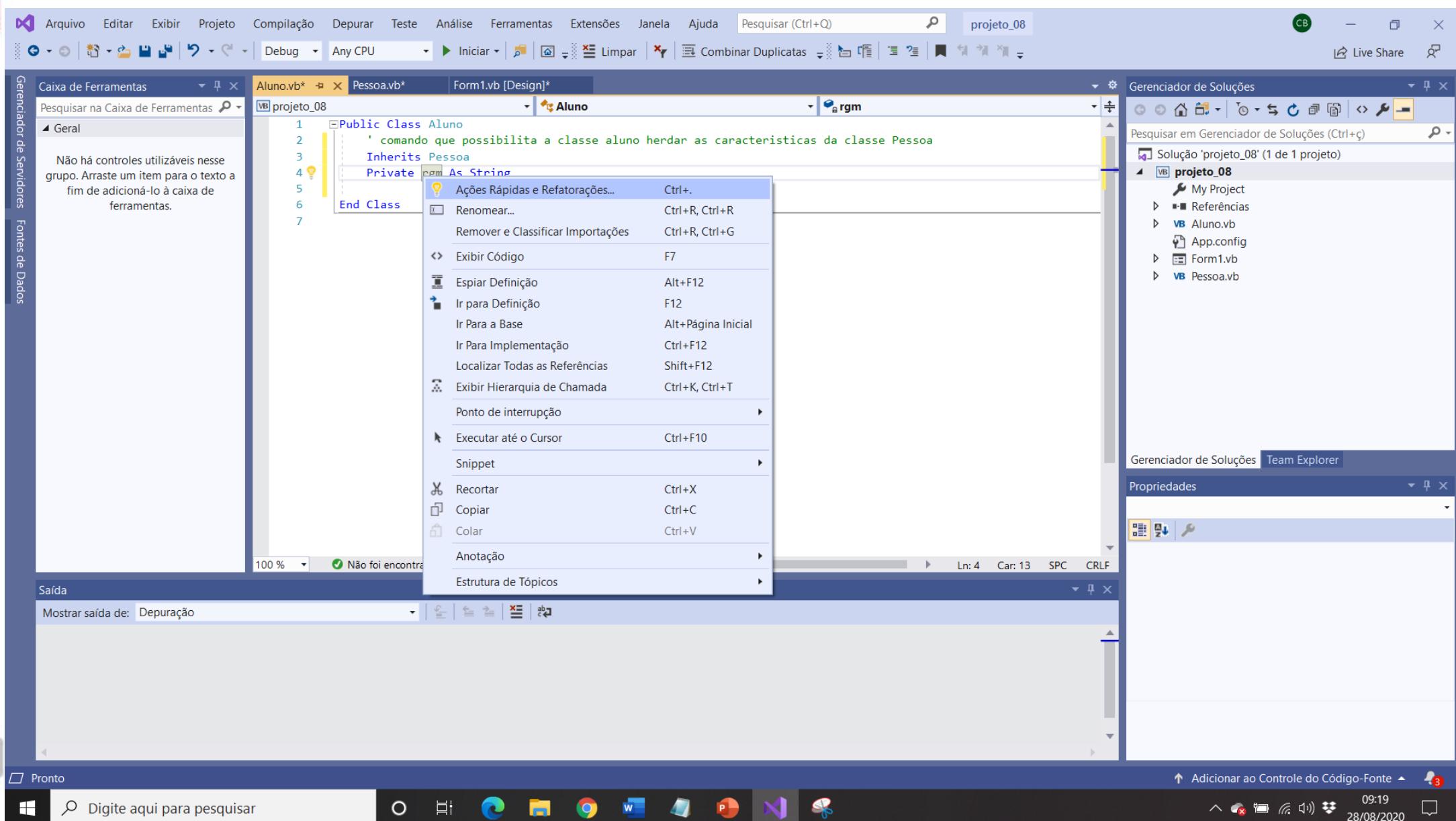
Pessoa

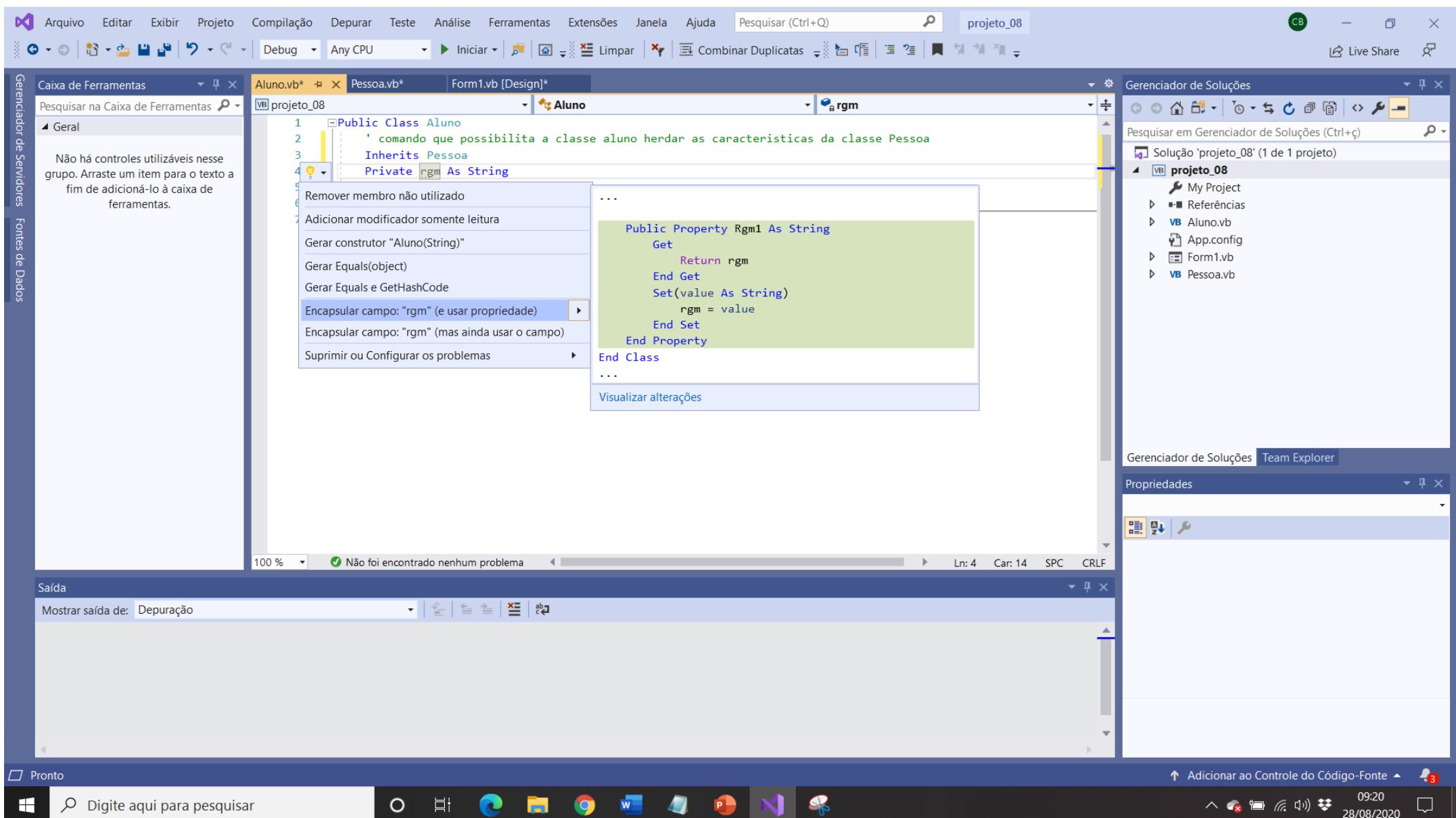
PNome

```
1  Public Class Pessoa
2      ' Private para que ele não possa ser acessado por uma classe que não seja dele
3      Private nome As String
4
5      0 referências
6      Public Property PNome As String
7          Get
8              Return nome
9          End Get
10         Set(value As String)
11             nome = value
12         End Set
13     End Property
14
--
```

VB.NET

Agora vamos trabalhar com a classe Aluno.







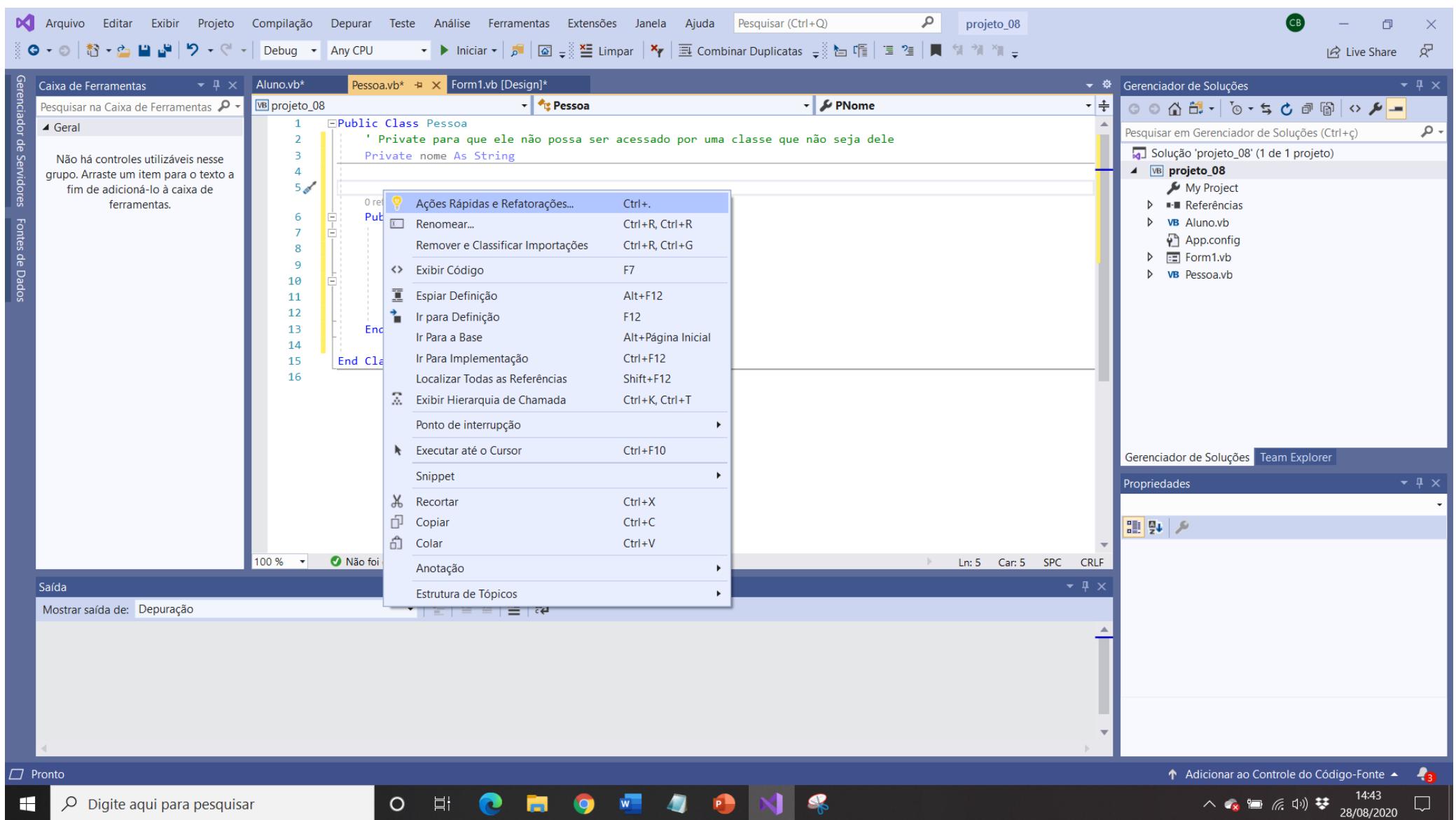
Aluno.vb* X Pessoa.vb* Form1.vb [Design]*

VB projeto_08 Aluno rgm

```
1  Public Class Aluno
2      ' comando que possibilita a classe aluno herdar as caracteristicas da classe Pessoa
3      Inherits Pessoa
4      Private rgm As String
5
6      'referências
7      Public Property pRgm As String
8          Get
9              Return rgm
10         End Get
11         Set(value As String)
12             rgm = value
13         End Set
14     End Property
15 End Class
```

VB.NET

Vamos criar o método construtor da classe Pessoa



A screenshot of the Microsoft Visual Studio 2019 IDE interface. The main window shows the code editor with the following code:

```
1  Public Class Pessoa
2      ' Private para que ele não possa ser acessado por uma classe que não seja dele
3      Private nome As String
4
5      ' PNome
6      ' End Property
7
8      ' End Class
```

The cursor is positioned at the end of the first line of the class definition. A context menu is open, listing several options:

- Gerar construtor...
- Substituir 'PNome' com métodos
- Gerar Equals(object)...
- Gerar Equals e GetHashCode...
- Gerar substituições...

The solution explorer on the right shows the project structure:

- Solução 'projeto_08' (1 de 1 projeto)
 - My Project
 - Referências
 - VB Aluno.vb
 - App.config
 - Form1.vb
 - VB Pessoa.vb

The status bar at the bottom displays the message "Não foi encontrado nenhum problema" (No errors found).

Arquivo Editar Exibir Projeto Compilação Depurar Teste Análise Ferramentas Extensões Janela Ajuda Pesquisar (Ctrl+Q) projeto_08

Debug Any CPU Iniciar Limpar Combinar Duplicatas Live Share

Caixa de Ferramentas Pesquisar na Caixa de Ferramentas

Geral Não há controles utilizáveis nesse grupo. Arraste um item para o texto a fim de adicioná-lo à caixa de ferramentas.

Aluno.vb* Pessoa.vb* Form1.vb [Design]*

VB projeto_08 Pessoa PNome

```
1 Public Class Pessoa
2     ' Private para que ele não possa ser acessado por uma classe que não seja dele
3     Private nome As String
4
5     0 referências
6     Public Property PNome As String
7         Get
8             Return nome
9         End Get
10        Set(value As String)
11            nome = value
12        End Set
13    End Property
14
15 End Class
16
```

Escolher membros

Escolher membros para que sejam usados como parâmetros do construtor

nome
 PNome

Selecionar Tudo Desmarcar Tudo

Adicionar verificações nulas

OK Cancelar

100 % Não foi encontrado nenhum problema

Saída Mostrar saída de: Depuração

Pronto Adicionar ao Controle do Código-Fonte 3 14:46 28/08/2020

Digitando aqui para pesquisar

Gerenciador de Soluções

Pesquisar em Gerenciador de Soluções (Ctrl+C)

Solução 'projeto_08' (1 de 1 projeto)

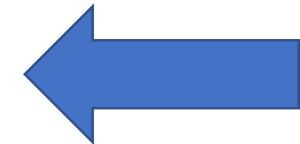
VB projeto_08

- My Project
- Referências
- VB Aluno.vb
- App.config
- Form1.vb
- Pessoa.vb

Gerenciador de Soluções Team Explorer

Propriedades

```
2 referências
1  Public Class Pessoa
2      ' Private para que ele não possa ser acessado por uma classe que não seja dele
3      Private nome As String
4
5      0 referências
6      Public Sub New(nome As String)
7          Me.nome = nome
8      End Sub
9
10     0 referências
11     Public Property PNome As String
12         Get
13             Return nome
14         End Get
15         Set(value As String)
16             nome = value
17         End Set
18     End Property
19
20 End Class
```



VB.NET

Vamos criar o método construtor da classe Aluno, repetir a operação anterior que foi feita na classe Pessoa.

A screenshot of the Microsoft Visual Studio 2019 IDE interface. The main window displays a code editor with the following code:

```
VB projeto_08
    Public Class Aluno
        ' comando que possibilita a classe aluno herdar as características da classe Pessoa
        Inherits Pessoa
        Private rgm As String
    End Class
```

The cursor is positioned on the word "Aluno". A context menu is open, listing various actions such as "Ações Rápidas e Refatorações...", "Renomear...", "Exibir Código", "Espiar Definição", "Ir para Definição", "Executar até o Cursor", and "Recortar".

The Visual Studio interface includes several toolbars and panes:

- Top Bar:** Arquivo, Editar, Exibir, Projeto, Compilação, Depurar, Teste, Análise, Ferramentas, Extensões, Janela, Ajuda, Pesquisar (Ctrl+Q), projeto_08, Live Share.
- Solution Explorer:** Gerenciador de Soluções, projeto_08, My Project, Referências, Aluno.vb, App.config, Form1.vb, Pessoa.vb.
- Properties Explorer:** Propriedades.
- Task List:** Mostrar saída de: Depuração.
- Bottom Status Bar:** Pronto, Adicionar ao Controle do Código-Fonte, 3, 14:50, 28/08/2020.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code editor with the file `Aluno.vb` open. The code defines a class `Aluno` that inherits from `Pessoa` and contains a private string member `rgm` and a public property `pRgm` with get and set methods. A tooltip for `pRgm` indicates it is a command that allows the `Aluno` class to inherit characteristics from the `Pessoa` class.

A modal dialog box titled "Escolher membros" (Select Members) is displayed, prompting the user to choose members for constructor parameters. It lists two members: `rgm` (checked) and `pRgm`. Buttons for "OK" and "Cancelar" (Cancel) are at the bottom, along with "Selecionar Tudo" (Select All) and "Desmarcar Tudo" (Uncheck All) options. A checkbox for "Adicionar verificações nulas" (Add null checks) is also present.

The Solution Explorer on the right shows the project structure with files `Aluno.vb`, `App.config`, `Form1.vb`, and `Pessoa.vb`.

At the bottom, the Taskbar shows various pinned icons, and the system tray displays the date and time as 28/08/2020 14:50.

```
Public Class Aluno
    ' comando que possibilita a classe aluno herdar as características da classe Pessoa
    Inherits Pessoa
    Private rgm As String

    Public Property pRgm As String
        Get
            Return rgm
        End Get
        Set(value As String)
            rgm = value
        End Set
    End Property
End Class
```

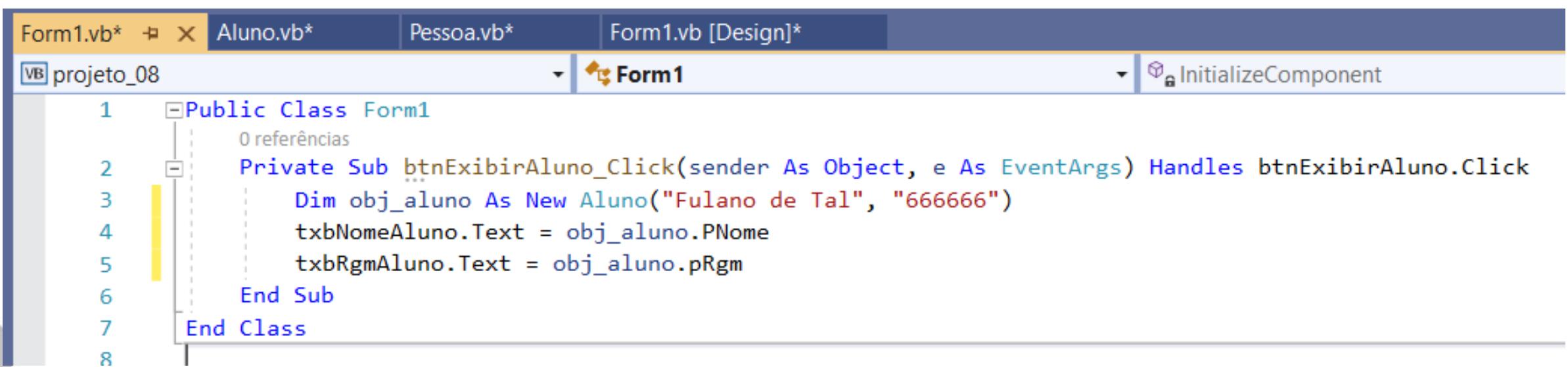
1 referência

```
1 Public Class Aluno
2     ' comando que possibilita a classe aluno herdar as características da classe Pessoa
3     Inherits Pessoa
4     Private rgm As String
5
6     0 referências
7     Public Sub New(nome As String, rgm As String)
8         ' como essa classe herda características da classe Pessoa preciso informar na linha de baixo
9         MyBase.New(nome)
10        ' MyBase informa que essa informação é da classe "PAI"
11        Me.rgm = rgm
12    End Sub
13
14    0 referências
15    Public Property pRgm As String
16        Get
17            Return rgm
18        End Get
19        Set(value As String)
20            rgm = value
21        End Set
22    End Property
23
24 End Class
```

VB.NET

Agora vamos inserir o código no botão Exibir:

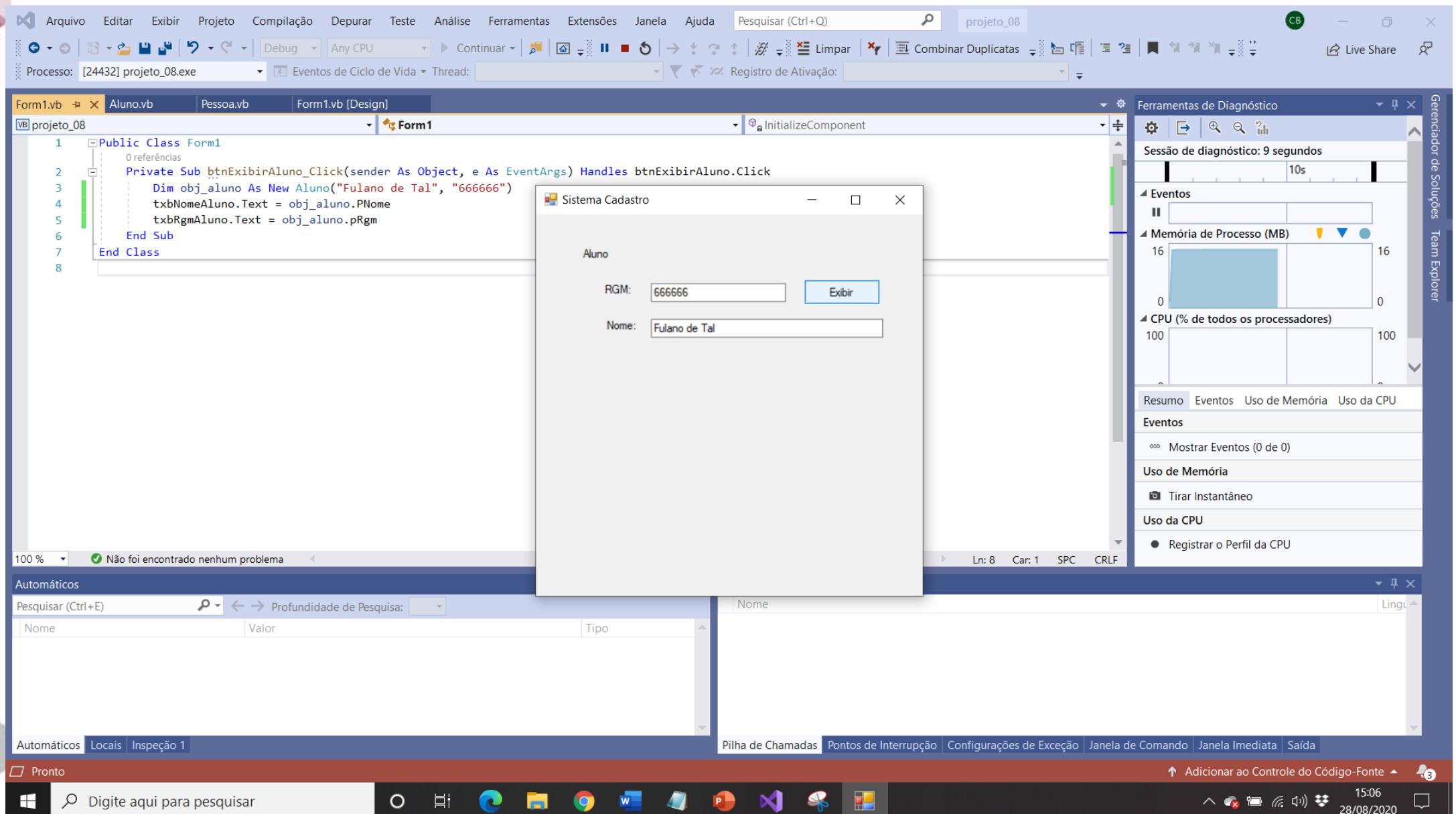
Agora vamos inserir o código no botão Exibir:



```
Form1.vb* ✘ X Aluno.vb* Pessoa.vb* Form1.vb [Design]* 7/4
VB projeto_08 Form1 InitializeComponent
1  Public Class Form1
2      | 0 referências
3      |  Private Sub btnExibirAluno_Click(sender As Object, e As EventArgs) Handles btnExibirAluno.Click
4          |      | Dim obj_aluno As New Aluno("Fulano de Tal", "666666")
5          |      | txbNomeAluno.Text = obj_aluno.PNome
6          |      | txbRgmAluno.Text = obj_aluno.pRgm
7      End Sub
8  End Class
```

VB.NET

Vamos testar:



VB.NET

Agora vamos repetir o processo criando a classe Professor:



Sistema Cadastro

Aluno

RGM:

Nome:

Professor

Título

Nome

Tempo para realizar o processo: 15 minutos

Conferindo o Resultado:



Atividade

Atividade 01

Valor: 0,50 Ponto

80

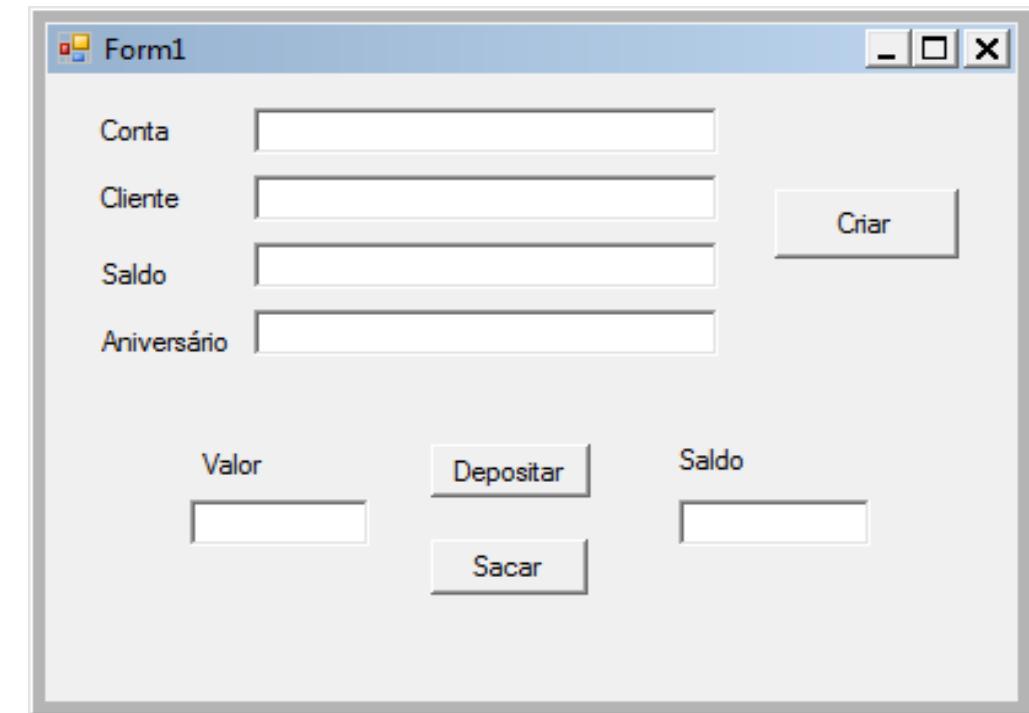
Entrega: 09/03/2022

Atividade

Criar uma classe em VB.NET chamada **Conta** que possui atributos privados para nome do cliente, número da conta e saldo. Criar métodos para as propriedades e métodos para sacar e depositar na conta.

Em seguida, criar uma classe chamada **Poupança** que herda de Conta com atributo de dia de aniversário.

Criar uma tela onde o usuário cadastra uma poupança e permita depositar e sacar da sua conta e mostrar o saldo. Veja figura ao lado:



Atividade

Dicas:

- Private saldo As Decimal → float
- Private conta As Integer → int
- Private nome As String → String



**"Nada na vida deve ser
temido, somente
compreendido.**

**Agora é hora de
compreender mais para
temer menos "**



Marié Curié





Obrigado!

Se precisar ...

Prof. Claudio Benossi

Claydoo.benossi@fatec.sp.gov.br

