





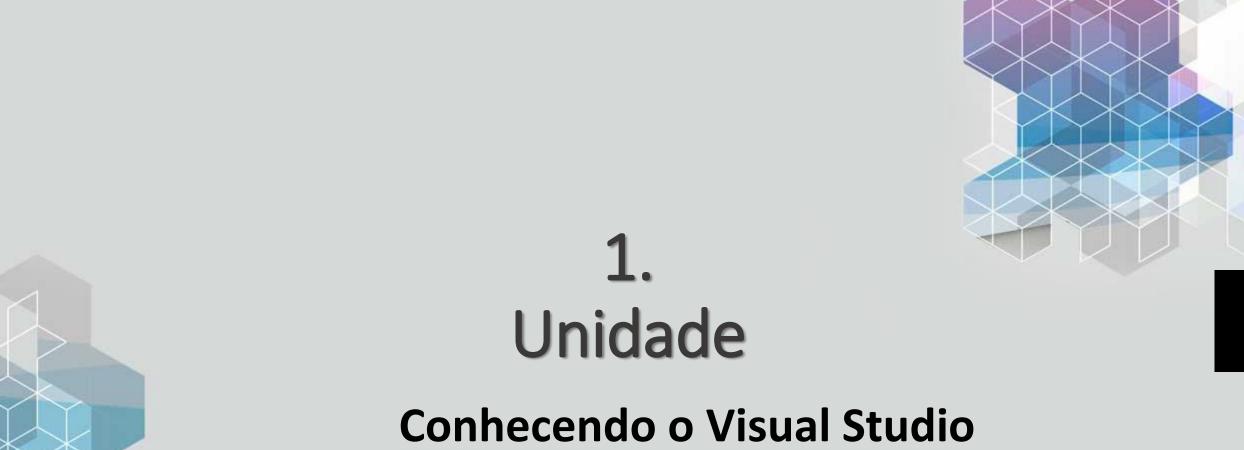
Técnicas de Programação II

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Aula 04

Prof. Claudio Benossi

Conhecendo o Visual Studio Exemplos em VB



VB.Net com Visual Studio

Objetivo da aula hoje:

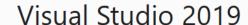
Trabalhar com os conceitos de Orientação a Objetos

VB.Net com Visual Studio

Objetivo da aula hoje:

Corrigir a Atividade;

Criar um cadastro com banco de dados;



Abrir recente

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

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

Introdução

<u>+</u>

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

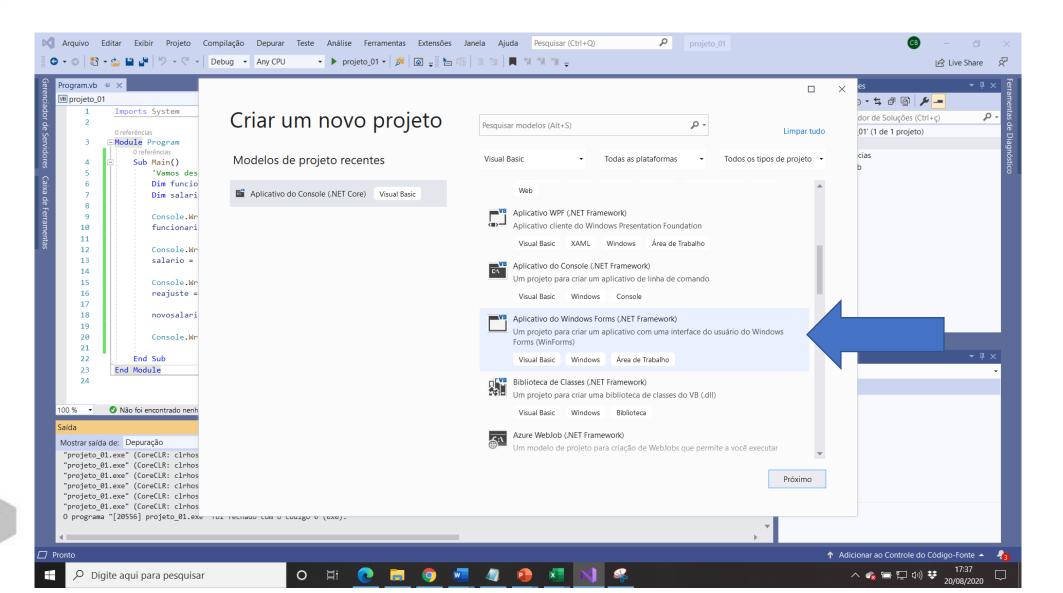
Navegar e editar o código dentro de qualquer pasta

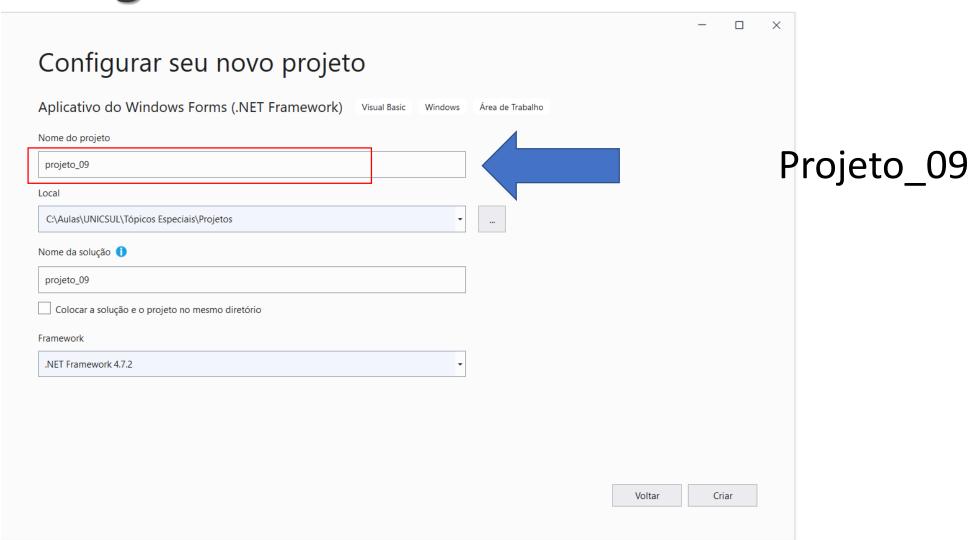


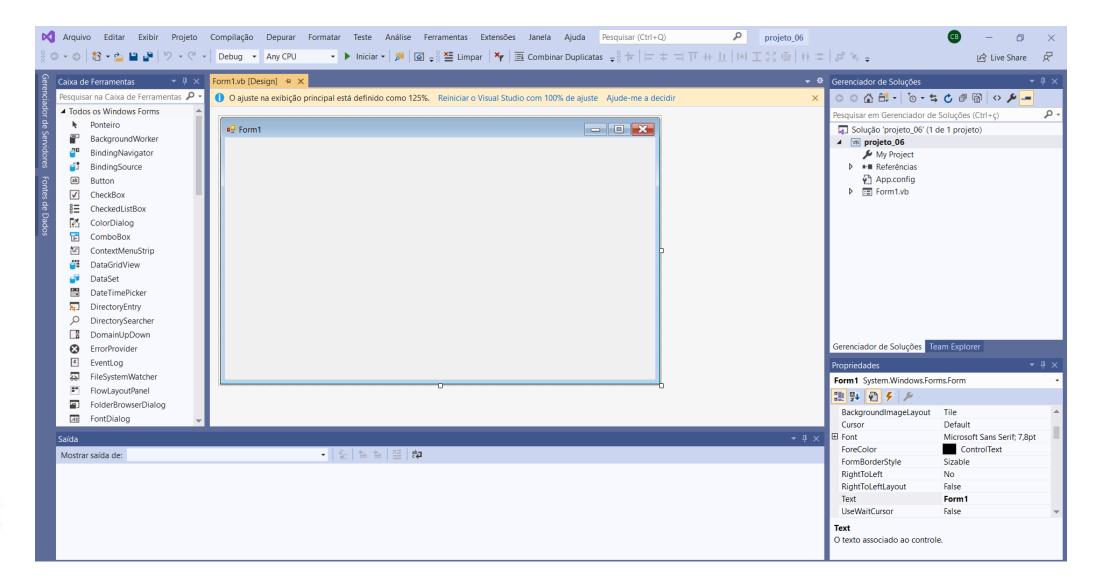
Criar um projeto

Escolha um modelo de projeto com scaffolding de código para começar

Continuar sem código →

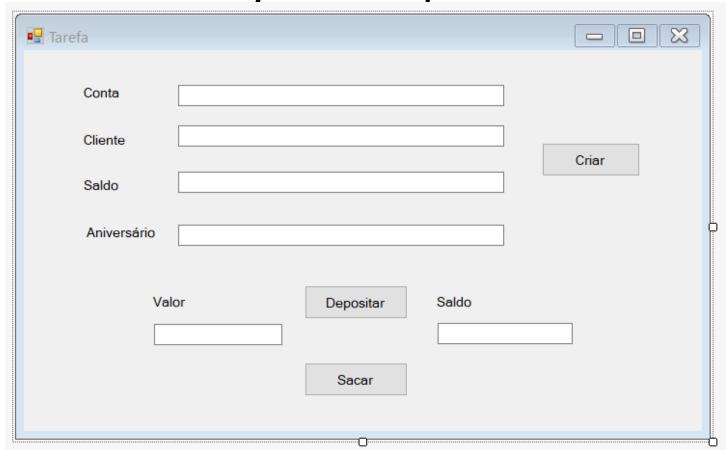


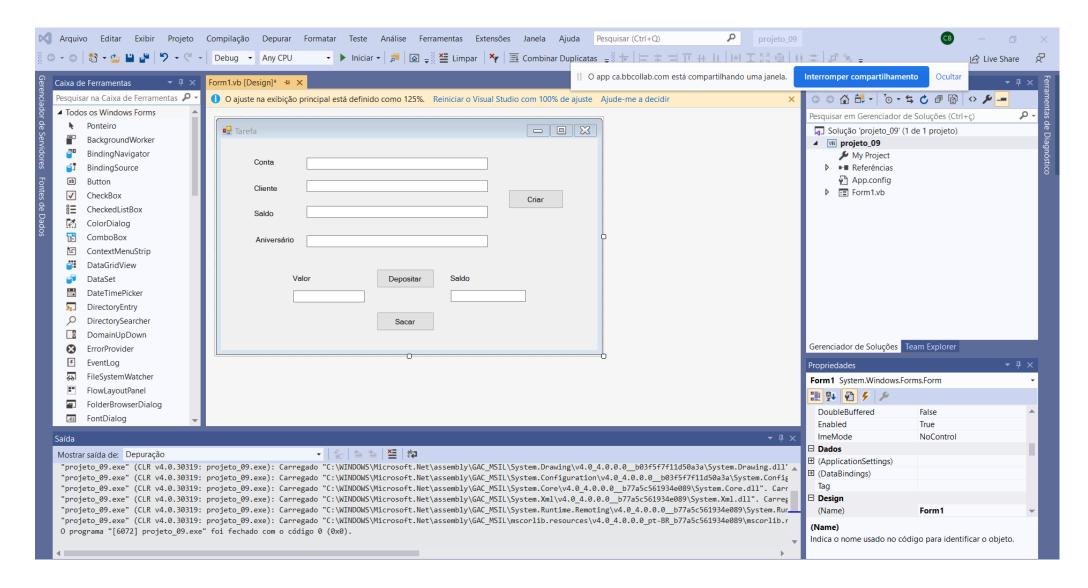




Agora vamos montar um layout simples com:

- 06 Label
- 06 TextBox
- •03 Buttons





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

StarPosition = CenterScreen



Agora vamos renomear nossos objetos:

- txbConta
- txbCliente
- txbSaldo
- txbAniversario
- txbValor
- txbSaldoFinal

- btnCriar
- btnDepositar
- btnSacar

Agora vamos renomear nossos objetos:

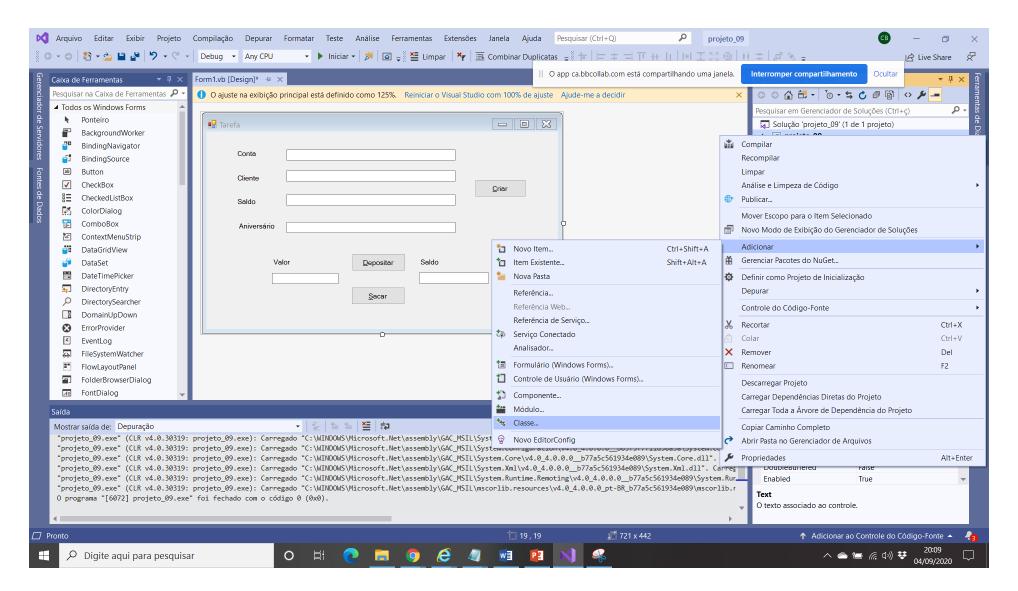
Vamos alterar também o text para: &Criar, &Depositar e
 &Sacar

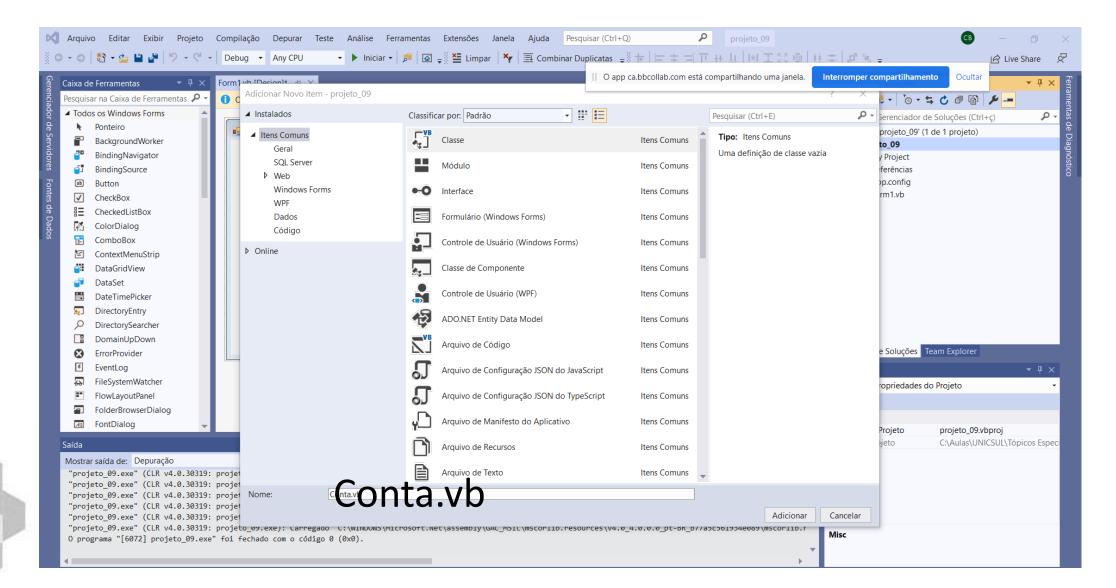
Dessa forma estamos criando atalhos para os botões.

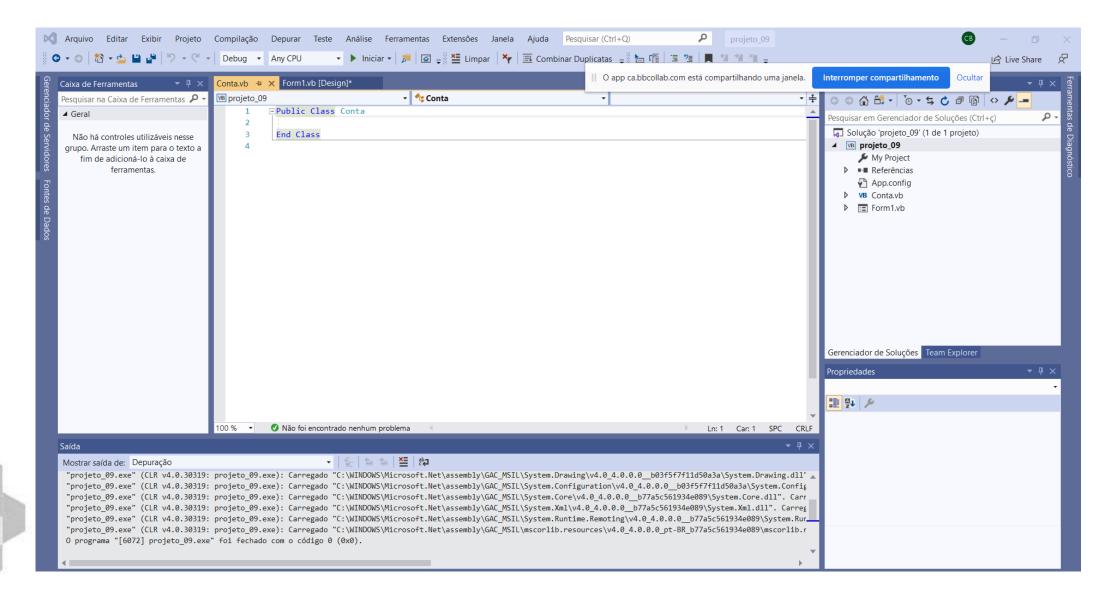
Descrição da atividade:

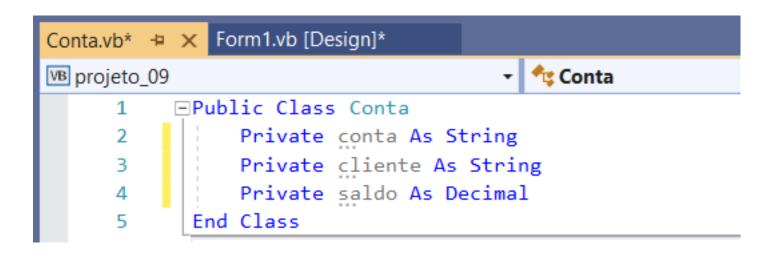
Criar uma classe em VB.NET chamada Conta que possui atributos privados para nome do cliente, número da conta e saldo.

Criar métodos para as propriedades e métodos para sacar e depositar na conta.



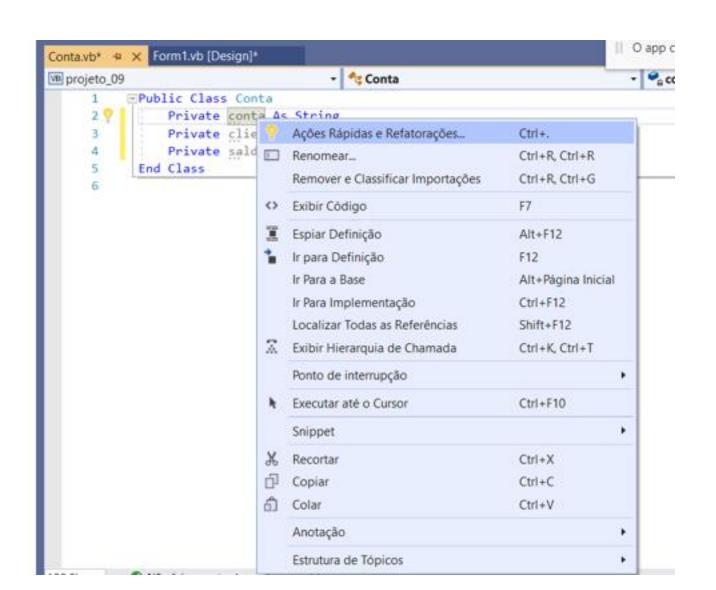




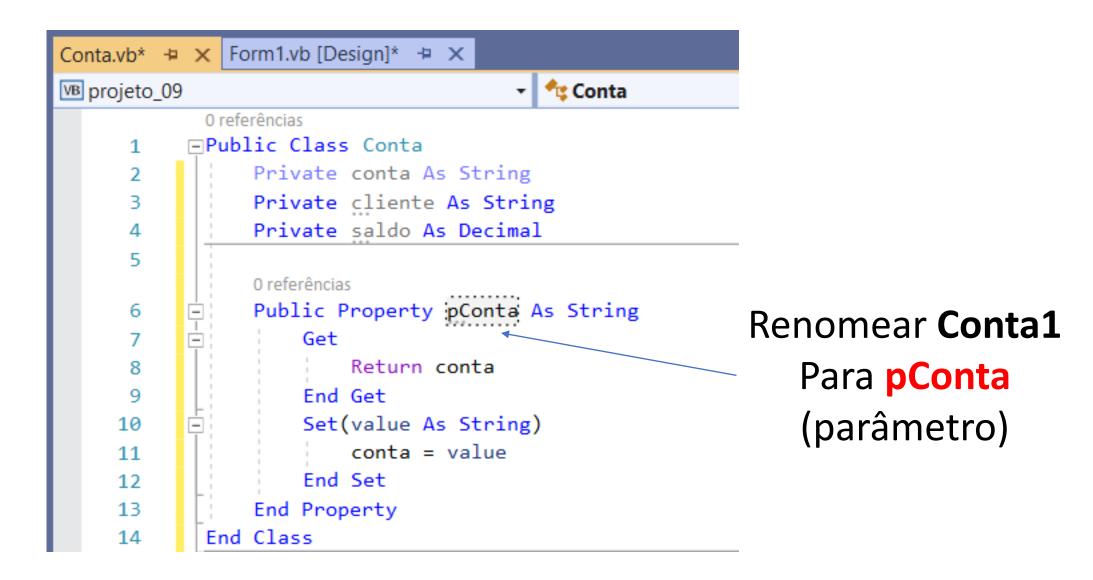


Vamos criar métodos de acesso aos dados Private através do encapsulamento.

"Regras para acesso aos dados"

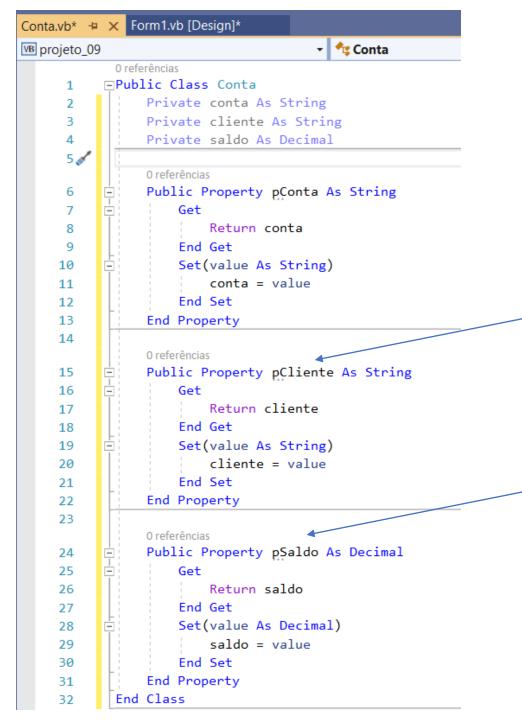






Repetir o procedimento para: Private Cliente e Private Saldo



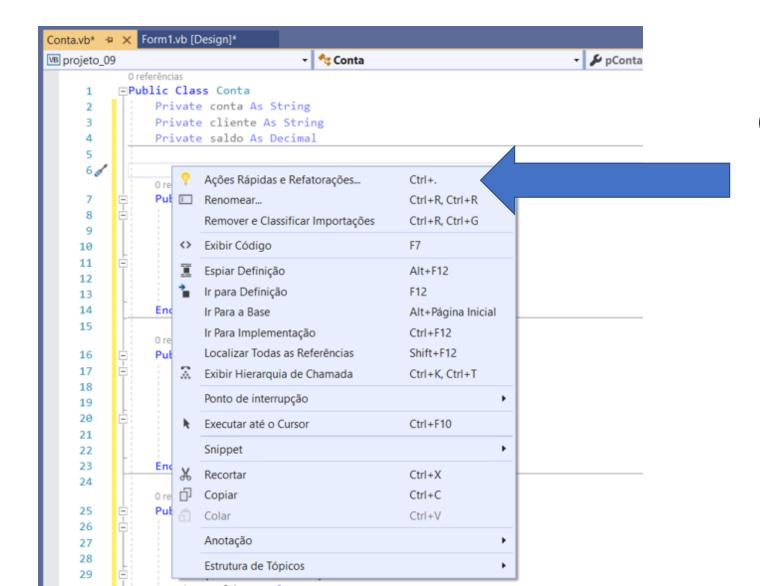


Renomear Cliente1 Para pCliente

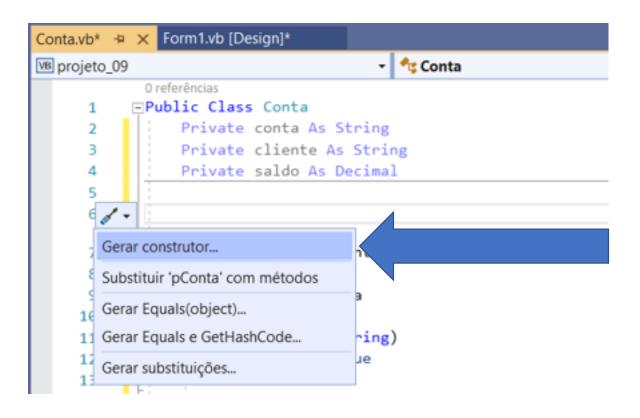
Renomear **Saldo1**Para **pSaldo**

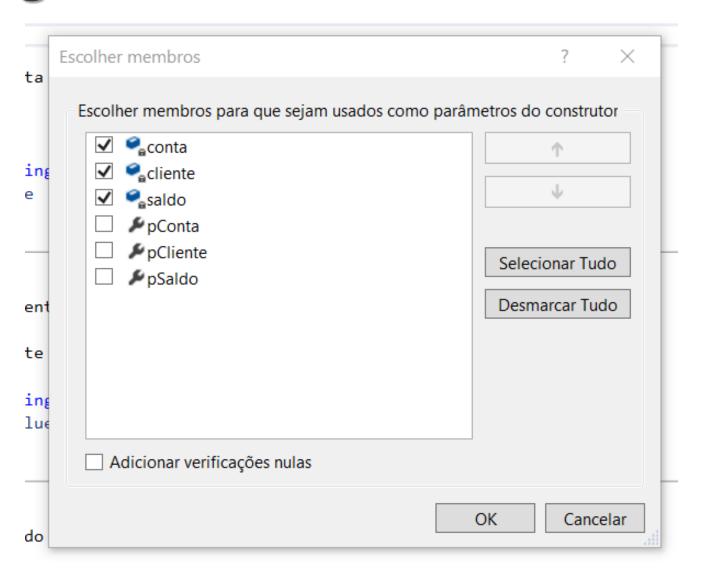
Vamos criar o método construtor dentro da classe Conta





Clique com o botão direito do mouse







Descrição ativid

número da con

Criar uma para depois

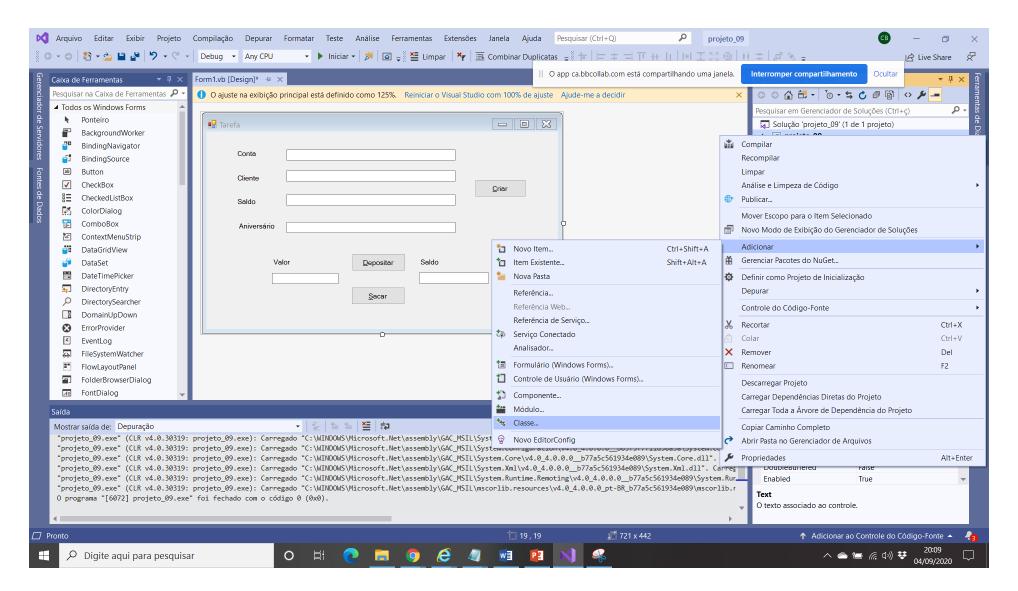
Vamos deixar para depois

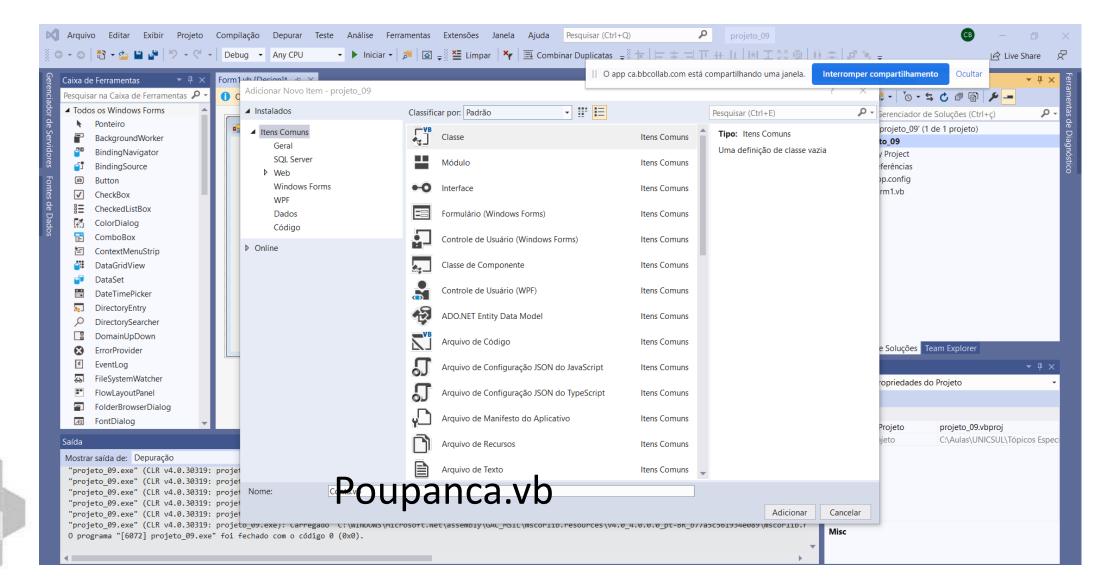
da Conta que la cliente,

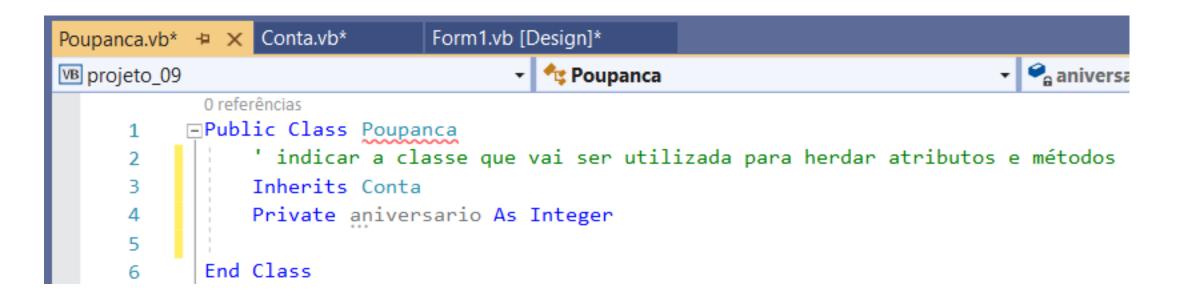
Criar métodos para propi dades e métodos para sacar e depositar na conta.

Vamos criar a classe Poupanca.vb

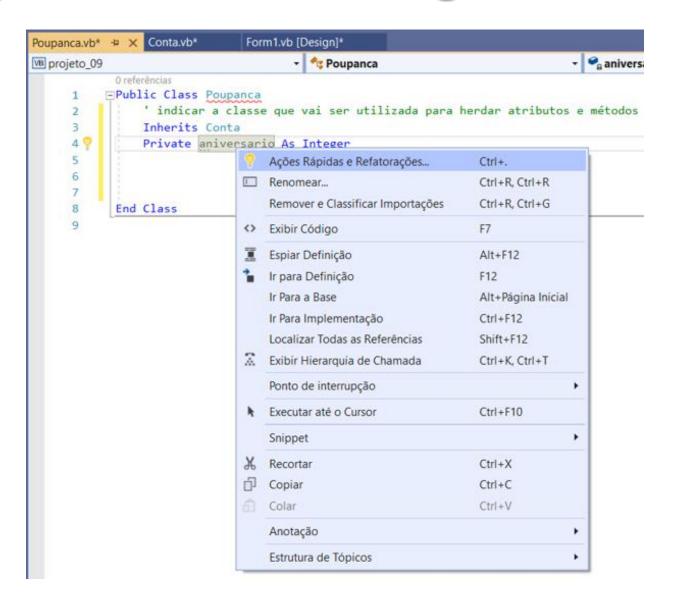


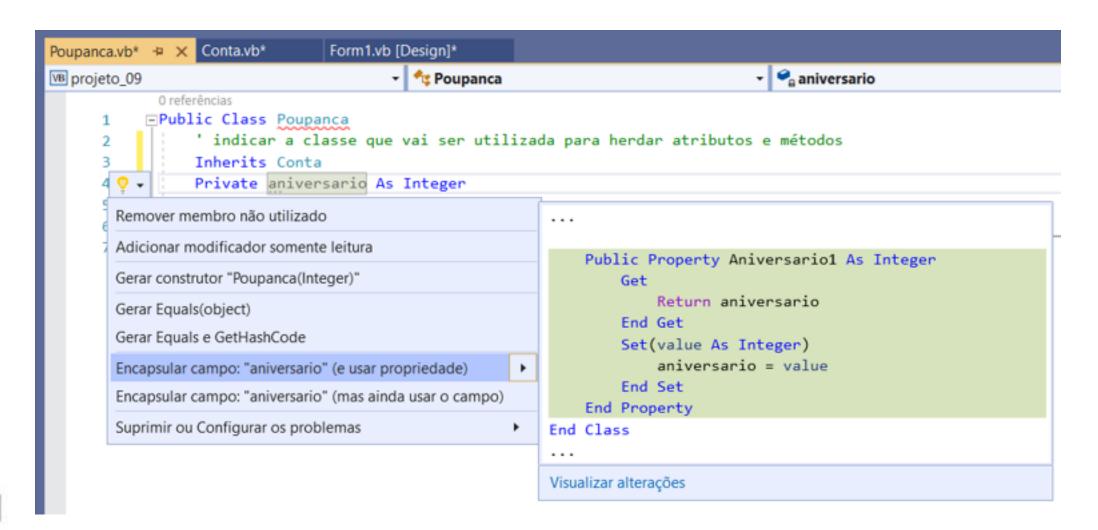


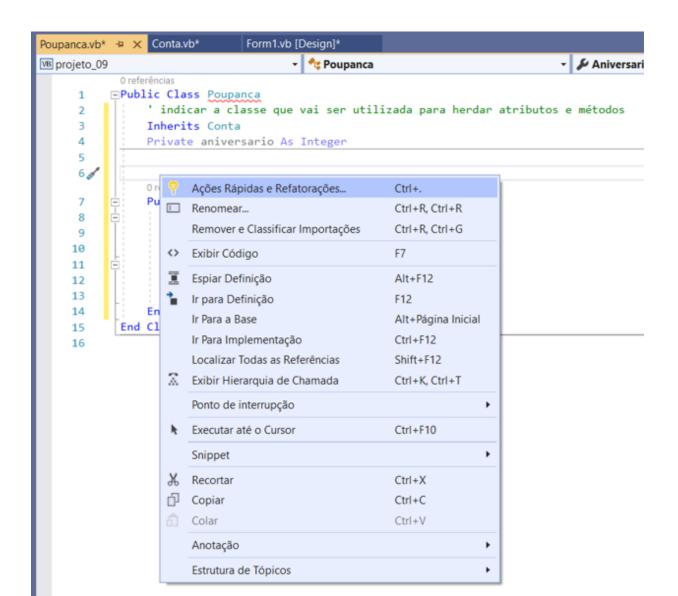


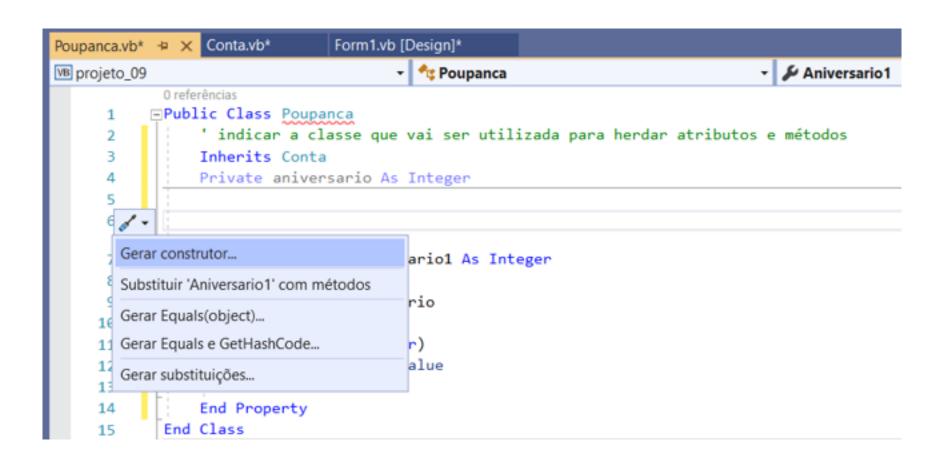


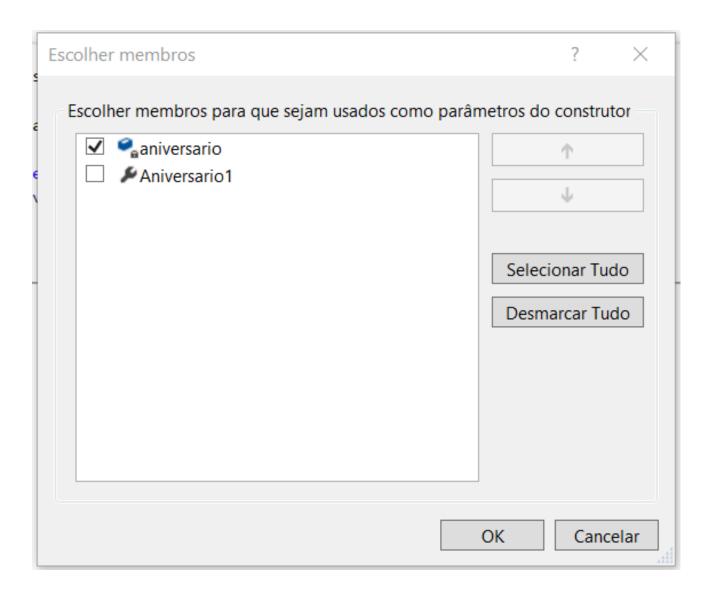
Projeto com interface gráfica VB.NET













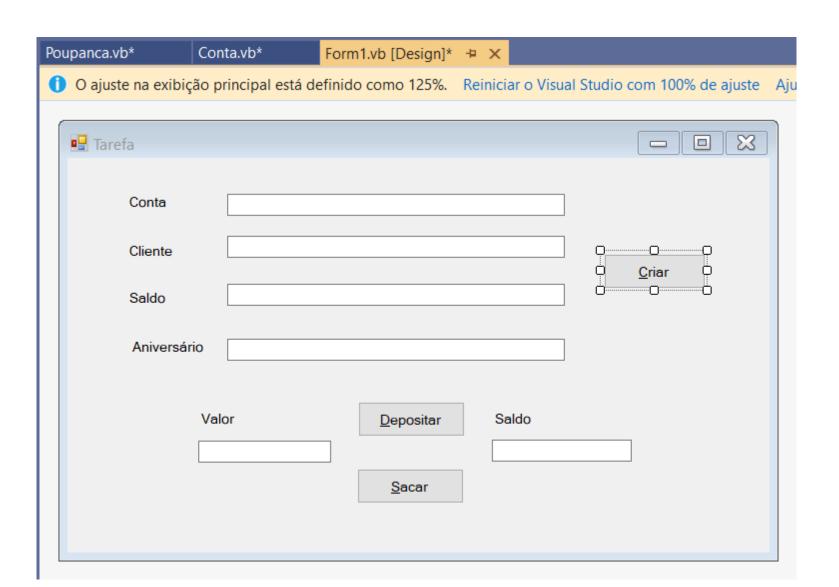
Como a classe Poupanca.vb herda atributos da classe Conta.vb, precisamos tratar isso ...



```
Form1.vb [Design]*
                                      🕶 🄩 Poupanca
                                                                               ▼ Maiversario1
VB projeto_09
            1 referência
          ■Public Class Poupanca
                ' indicar a classe que vai ser utilizada para herdar atributos e métodos
                Inherits Conta
     4
               Private aniversario As Integer
               0 referências
                Public Sub New (conta As String, cliente As String, saldo As Decimal, aniversario As Integer)
     6
                    MyBase.New(conta, cliente, saldo)
                    Me.aniversario = aniversario
                End Sub
     9
    10
                0 referências
                Public Property Aniversario1 As Integer
    11
    12
                    Get
                        Return aniversario
    13
                    End Get
    14
    15
                    Set(value As Integer)
    16
                        aniversario = value
                    End Set
    17
                End Property
    18
            End Class
    19
```

Vamos inserir os códigos no botão Criar



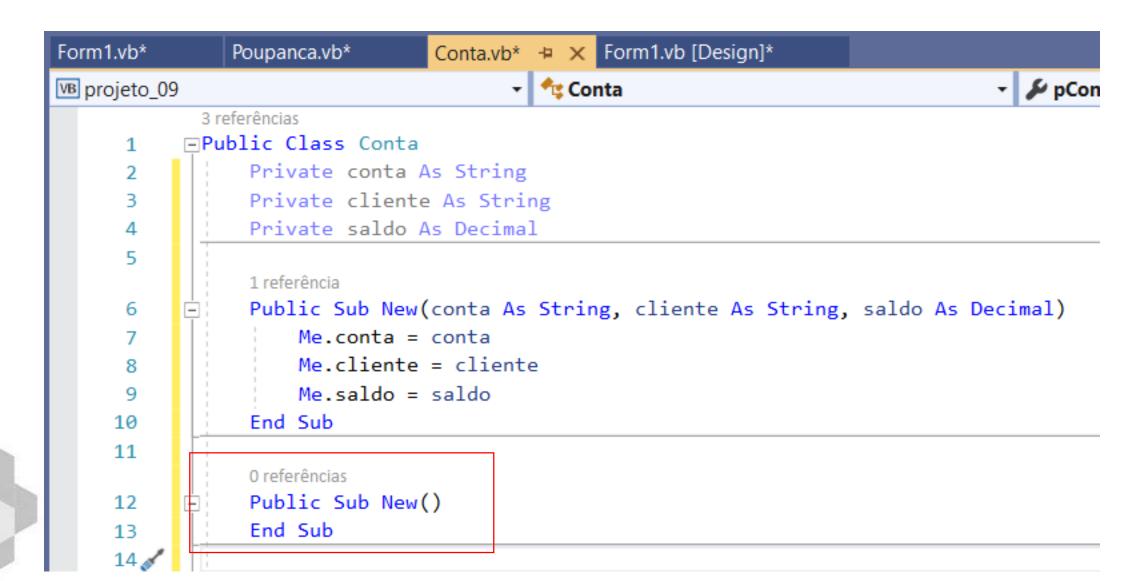


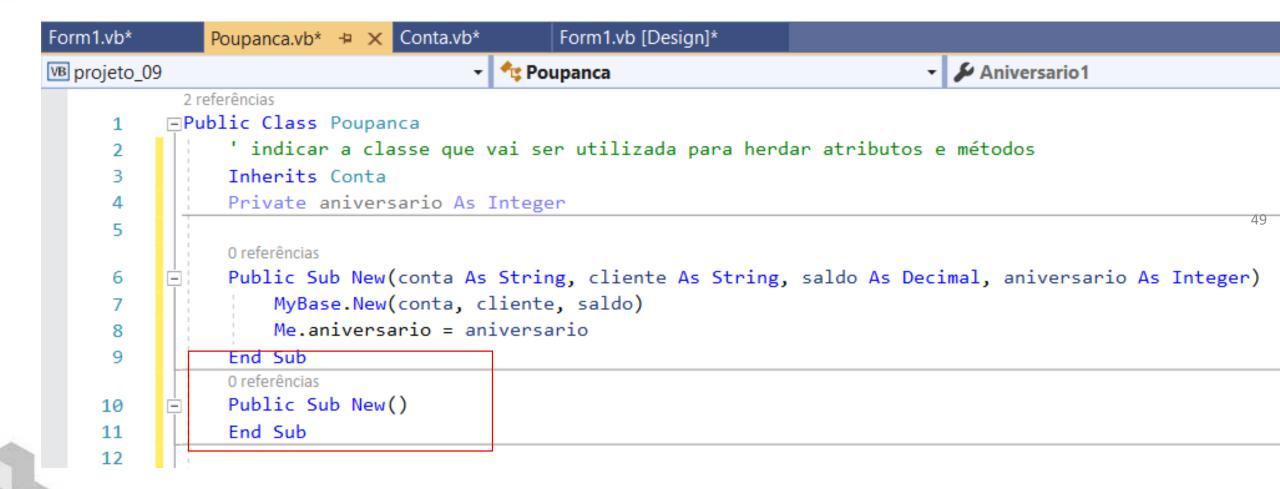
Porém para garantir acesso a todos os dados no momento que utilizarmos os botões, precisamos fazer alguns ajustes...

Poupanca.vb*	Conta.vb*	Form1.vb [Design]*	₽ X			
O ajuste na exibiç	ão principal está	definido como 125%.	Reiniciar o Vis	ual Studio	com 100% de ajuste	Aj
■ Tarefa						
Conta						
Cliente					<u>C</u> riar	
Saldo Aniversár						
Aniversar	10					
	<u>D</u> epositar	Saldo				
		<u>S</u> acar				

Vamos criar um método construtor vazio dentro das duas classes..

Isso porque estamos usando os objetos locais... (dentro do Private)

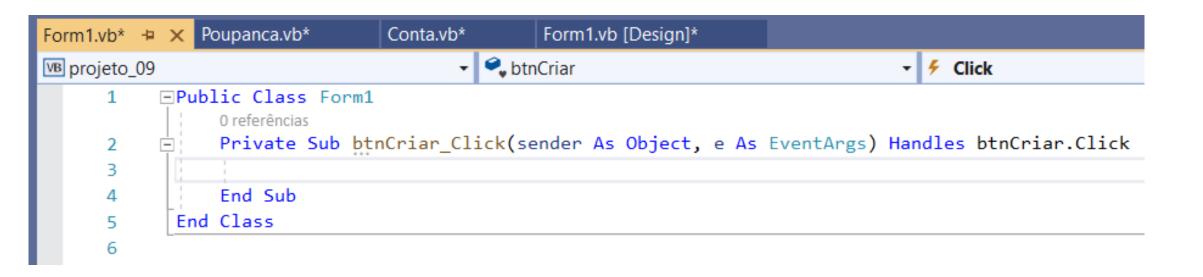




Isso garante acesso aos objetos, porém pode ser feito de outras formas ...

Porém são mais demorados ...

Vamos dar um duplo clique no botão criar.



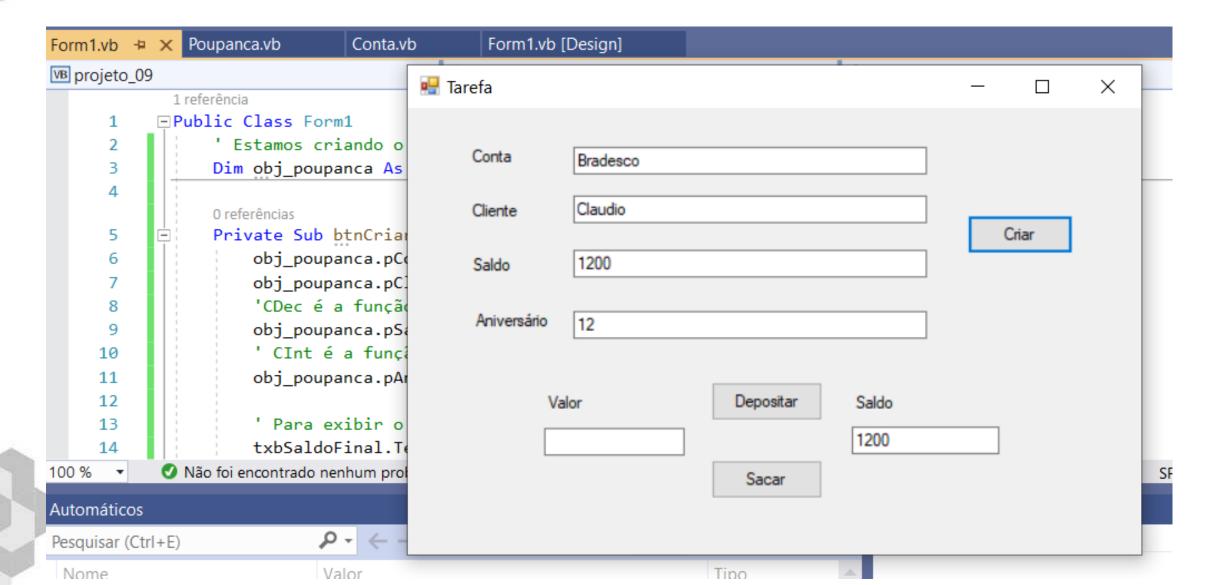


Agora vamos criar o objeto e atribuir valores ...



```
Form1.vb* ₽ X Poupanca.vb*
                                Conta.vb*
                                               Form1.vb [Design]*
                                       ▼ ¶ Form1
                                                                                 ▼ Cobj poupanca
VB projeto_09
            1 referência
           □Public Class Form1
                 ' Estamos criando o objeto como público ou seja global, todos tem acesso...
                Dim obj poupanca As New Poupanca
                O referências
                Private Sub btnCriar_Click(sender As Object, e As EventArgs) Handles btnCriar.Click
                     obj poupanca.pConta = txbConta.Text
                     obj poupanca.pCliente = txbCliente.Text
                     'CDec é a função para converter texto em número decimal
                     obj_poupanca.pSaldo = CDec(txbSaldo.Text)
                     ' CInt é a função para converter tento em número inteiro
    10
    11
                     obj poupanca.pAniversario = CInt(txbAniversario.Text)
    12
                     ' Para exibir o saldo final
    13
    14
                     txbSaldoFinal.Text = obj poupanca.pSaldo
                End Sub
    15
            End Class
     16
```

Vamos testar ...



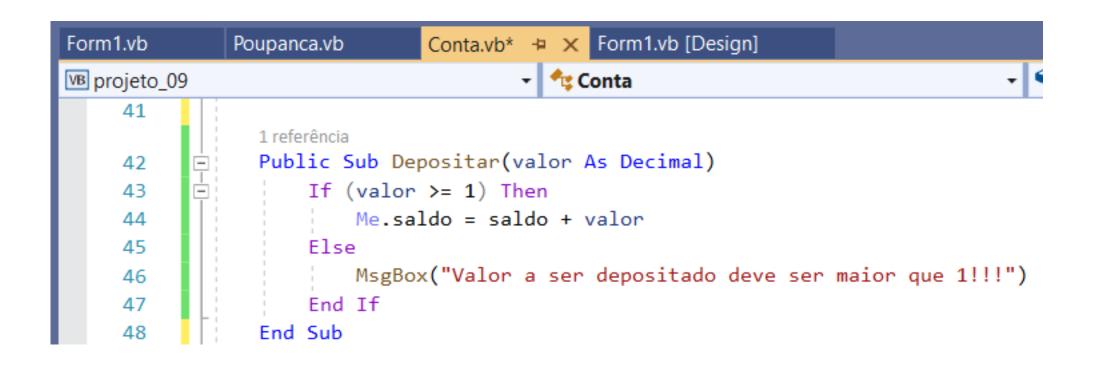
Criar uma classe em VB.NET chamada **Conta** que possui atributos privados para nome do cliente, número da conta e saldo.

Criar métodos para as propriedades e métodos para sacar e depositar na conta.

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

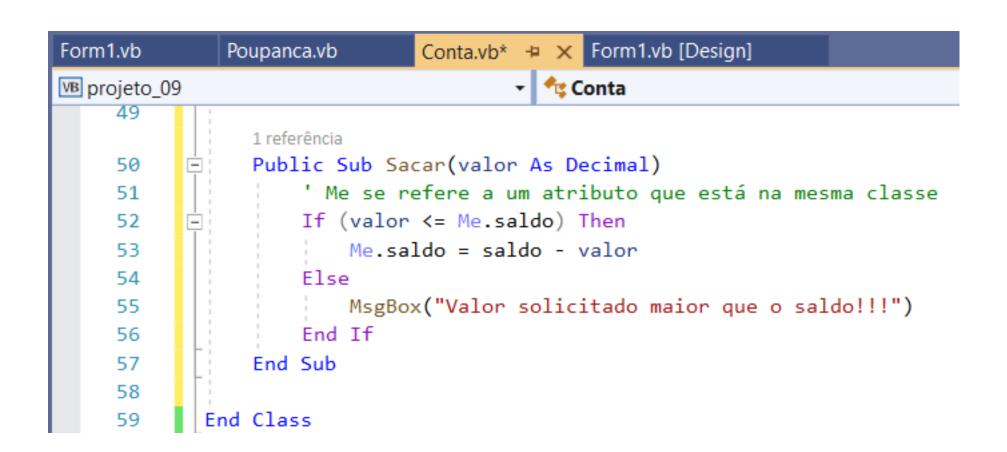
Vamos criar um método para Depositar na classe Conta.vb





Agora vamos criar um método para Sacar na classe Conta.vb





Agora precisamos inserir os códigos nos botões Depositar e Sacar



633

Conferindo o Resultado:



Vamos elaborar um sistema para calcular o IMC, onde o IMC é realizado com o seguinte calculo:

IMC = peso / (altura*altura)



Efetuar a classificação utilizando a tabela abaixo:

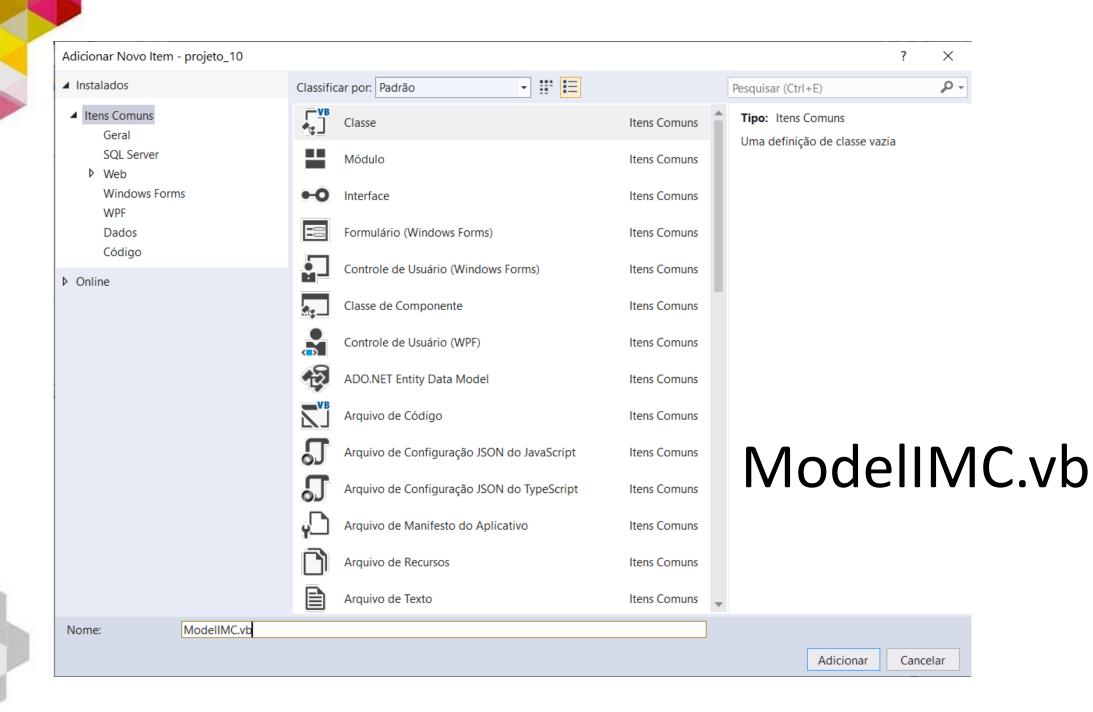
IMC	Classificação		
< 18.5	Desnutrido		
18.5 – 24.9	Adequado		
25.0 – 29.9	Pré-obeso		
30.0 – 34.9	Obesidade grau I		
35.0 – 39.9	Obesidade grau II		

Exemplo de tela:



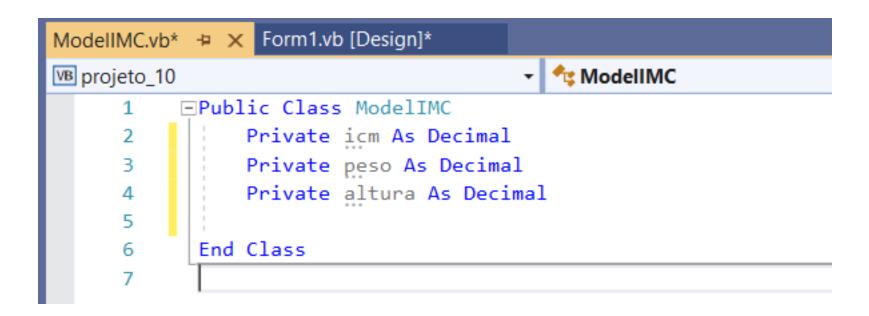
Vamos criar um novo projeto e criar uma classe ModelIMC





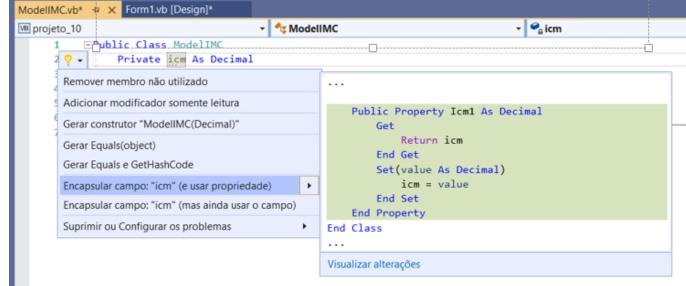
Vamos criar os atributos...





Vamos selecionar os atributos e selecionar a opção ações rápidas e refatorações,

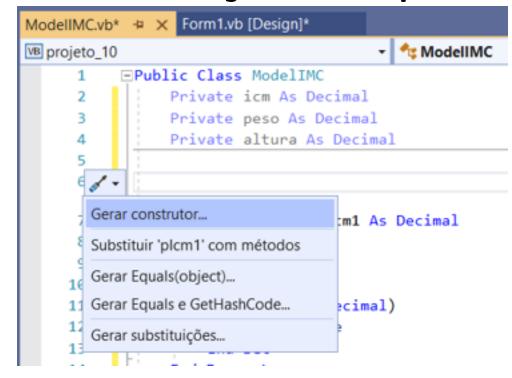
Encapsular campo ..

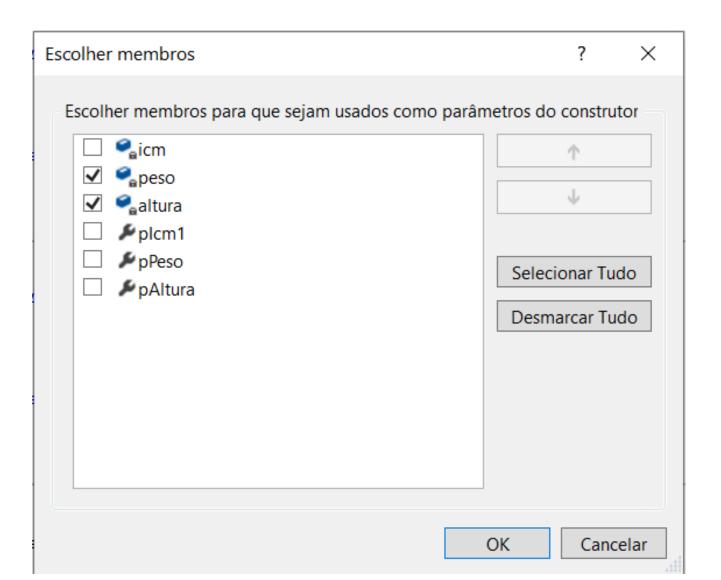


Adiciono o p para indicar que é um parâmentro

Vamos criar um método construtor através da opção ações rápidas e refatorações depois

Gerar Construtor





```
ModelIMC.vb* → × Form1.vb [Design]*
                                         ▼ ModelIMC
VB projeto_10
            1 referência
           □Public Class ModelIMC
                Private icm As Decimal
               Private peso As Decimal
                Private altura As Decimal
                0 referências
                Public Sub New(peso As Decimal, altura As Decimal)
                     Me.peso = peso
                     Me.altura = altura
                 End Sub
```

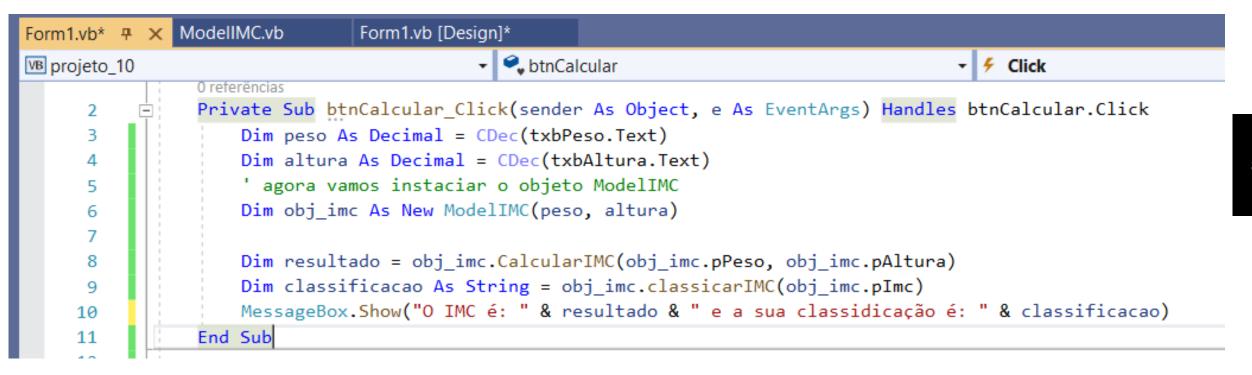
Vamos criar um método para exibir a classificação de acordo com o cálculo do IMC



```
ModelIMC.vb* → X Form1.vb [Design]*
                                         ▼ ModelIMC
VB projeto_10
     42
                0 referências
                Public Function classicarIMC(imc As Decimal)
     43
                     ' Vamos usar Function porque teremos um retorno
     44
                     Dim classificacao As String
     45
                     If (imc < 8.5) Then
     46
     47
                         classificação = "Desnutrido"
    48 😨
                     ElseIf (imc < 25) Then
                         classificacao = "Adequado"
     49
                     ElseIf (imc < 30) Then
     50
     51
                         classificacao = "Sobrepeso"
                     ElseIf (imc < 35) Then
     52
                         classificação = "Obsedidasde Nível I"
     53
                     Else
     54
                         classificação = "Obsedidasde Nível II"
    55
    56
                     End If
                     Return classificação
     57
                End Function
    58
     59
            End Class
     60
```

Vamos inserir o código no botão Calcular IMC no formulário





Vamos inserir o código no botão Sair

```
ModelIMC.vb*
                                                                                                                                                                                                                                                                                                                                                           Form1.vb [Design]*
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              ▼ 🗣 Initializ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         ▼ 1 transfer + Transf
VB projeto_10
                                                                                                                                                                               ena Sub
                                                                                                                                                                            0 referências
                                                                                                                                                                             Private Sub btnSair_Click(sender As Object, e As EventArgs) Handles btnSair.Click
                                                  10
                                                                                                                                                                                                                         Me.Close()
                                                  11
                                                                                                                                                                              End Sub
                                                  12
                                                                                                                                  End Class
                                                    13
```

Vamos inserir o código no botão Limpar

```
Form1.vb [Design]*
                                          → StnLimpar
                                                                                        → F Click

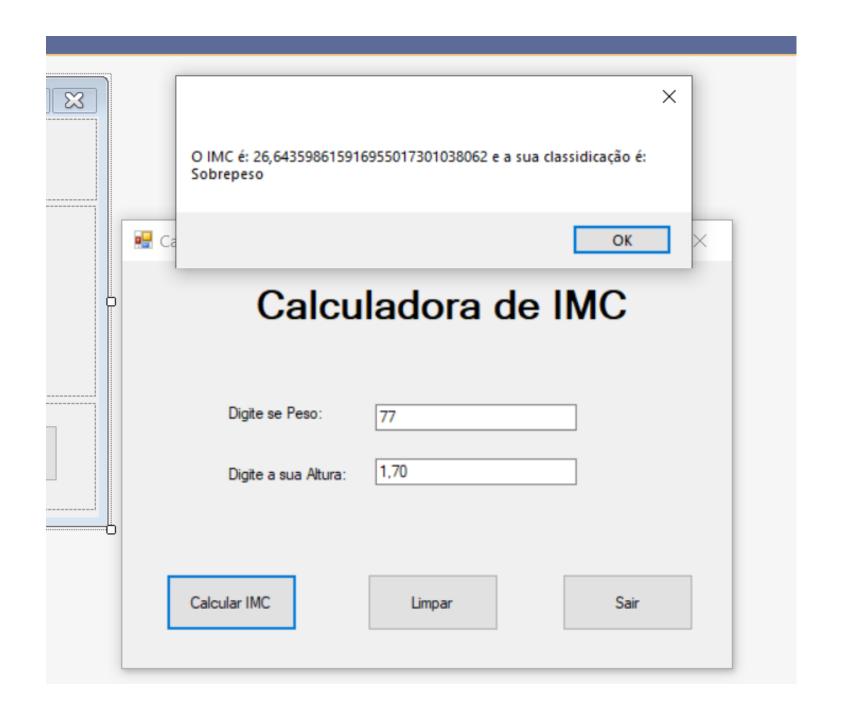
    projeto_10

                   Me.Close()
    TT
               End Sub
    12
    13
               O referências
               Private Sub btnLimpar_Click(sender As Object, e As EventArgs) Handles btnLimpar.Click
    14
                   txbPeso.Clear()
    15
                   txbAltura.Clear()
    16
               End Sub
    17
           End Class
    18
    19
```

Corrigindo a Atividade

Conferindo o Resultado:





"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

Claudio.benossi@fatec.sp.gov.br

