

Alterações Apostila de C# MOD I

Sumário

AULA 01.....	3
Download Visual Studio, Aula Interativa	3
Instalação Visual Studio, Aula Interativa.....	3
Abrindo o Visual Studio 2022, Páginas 6 a 9	5
Componentes do Visual Studio, Páginas 9 a 15	7
AULA 02.....	11
Formulário Bem-Vindo, Interatividade	11
Criando o primeiro projeto, Páginas 17 a 29	11
AULA 03.....	18
Formulário Celular, Interatividade	18
Nomenclatura de variáveis, Página 33.....	19
Declarando variável, Página 34.....	19
Passo a Passo, Páginas 35 a 41:	19
AULA 04.....	23
Formulário Calculadora, Aula interativa	23
Páginas 44 a 48	24
Passo a Passo, Páginas 49 a 53:	25
Utilizando Instruções IF... Else, Páginas 53 a 55	27
Passo a Passo, Páginas 55 e 56:	27
AULA 05.....	28
Formulário Comparando Idade, Interatividade	28
Operadores Lógicos, Páginas 59 a 63.....	30
Instruções IF Encadeado, Páginas 65 a 69	32
AULA 06.....	35
Formulário Categoria, Interatividade.....	35
Instruções IF em Cascata, Páginas 72 a 82	36
AULA 07.....	42
Formulário Pacote de Viagem, Interatividade	42
Utilizando Instruções Switch para Tomada de Decisões, Páginas 89 a 96	44
AULA 08.....	47
Revisão das aulas anteriores, Páginas 98 a 100.....	47

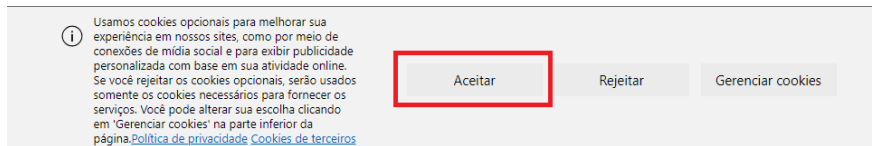
Icaro Daflon
C# MOD I

AULA 09.....	49
Formulário Pares e Ímpares, Interatividade	49
Estrutura de Repetição (While), Páginas 105 a 110.....	50
Estrutura de Repetição (For), Páginas 114 a 118.....	53
AULA 10.....	55
Formulário Somar Intervalo, Interatividade	55
Estrutura de Repetição (Do), Páginas 122 a 128.....	57
AULA 11.....	59
Formulário Somar Intervalo, Interatividade	59
Tratamento de Exceções, Páginas 133 a 139.....	61
Linhas de Comentários, Páginas 140 a 141.....	64
AULA 12.....	65
Revisão das aulas anteriores, Páginas 144 e 145.....	65
AULA 13.....	67
Conceitos Básicos de Programação Orientada a Objetos, Páginas 149 a 152.....	67
AULA 14.....	67
Formulário Categoria Utilizando Métodos, Interatividade.....	67
Criando Métodos com C#, Páginas 159 a 162.....	68
Chamando Métodos, Páginas 162 e 163	70
AULA 15.....	70
Formulário Cadastrar Livro, Interatividade.....	70
Trabalhando com o Conceito de Classes, Páginas 166 a 176	74
AULA 16.....	79
Formulário Caixa Eletrônico, Interatividade	79
Aplicando o Conceito de Herança, Páginas 178 a 187	84
AULA 17.....	87
Formulário Polimorfismo, Iteratividade.....	87
Interface, Páginas 190 a 199	89

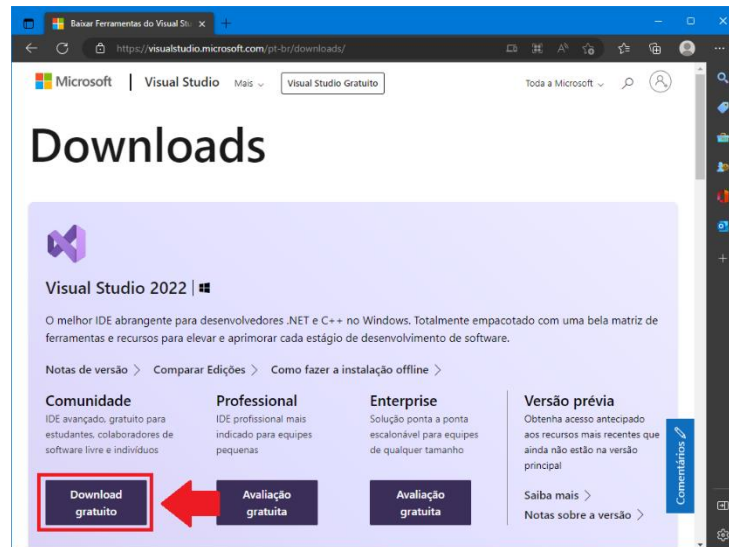
Icaro Daflon
C# MOD I
AULA 01

Download Visual Studio, Aula Interativa

- Primeiramente iremos realizar o download da IDE Visual Studio, para isso acesse o site <https://visualstudio.microsoft.com/pt-br/downloads/>
- Caso apareça a janela referente aos cookies do site, clique em aceitar:



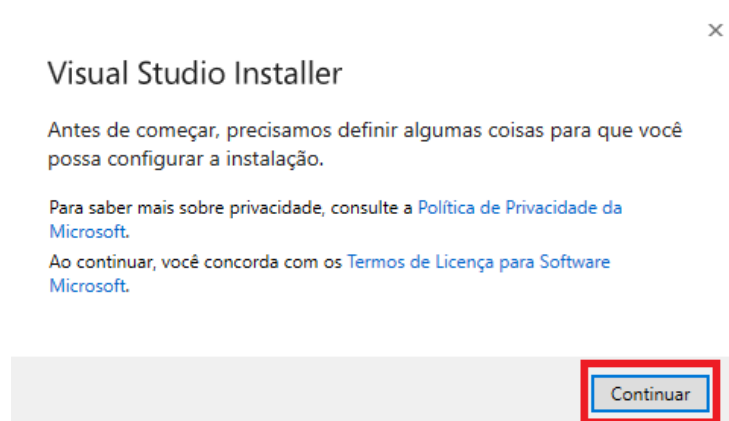
- Prosiga clicando no botão “Download Gratuito” da opção “Comunidade” do Visual Studio, Neste curso estaremos utilizando o Visual Studio 2022:



- Agora é só esperar o Download finalizar.

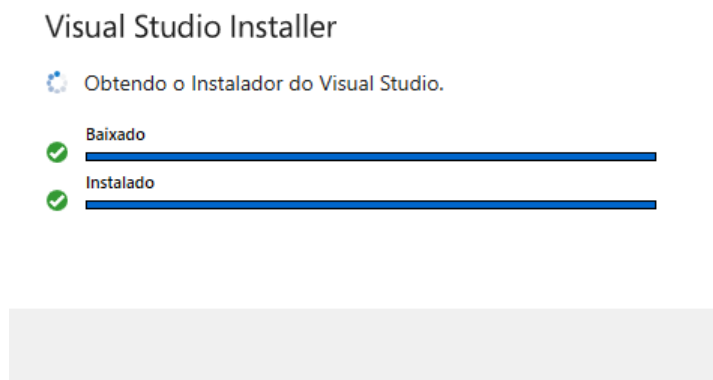
Instalação Visual Studio, Aula Interativa

- Abra o arquivo “VisualStudioSetup” baixado anteriormente.
- Na janela que irá aparecer apenas clique em Continuar:

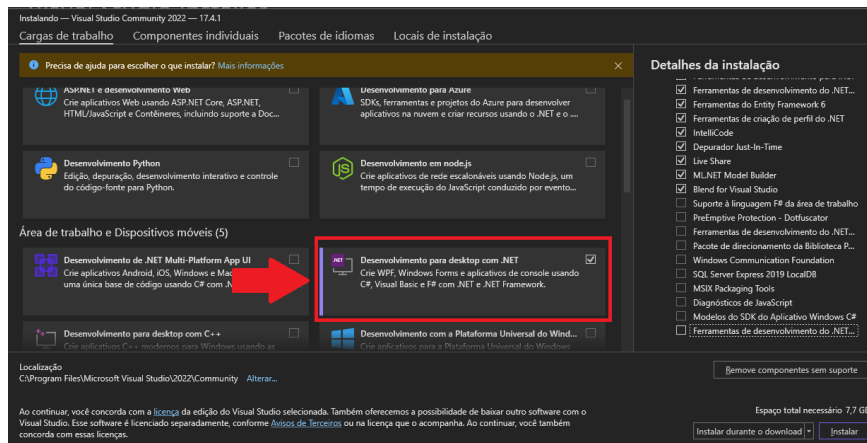


Icaro Daflon
C# MOD I

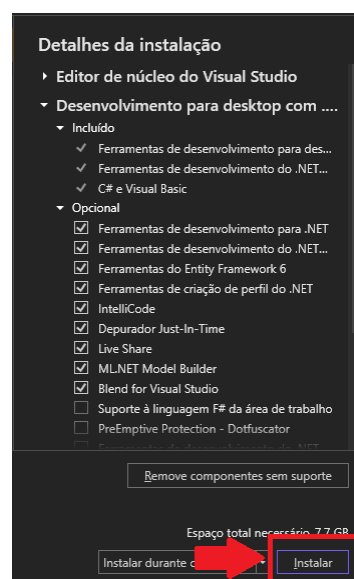
- Aguarde a instalação do Instalador do Visual Studio finalizar:



- Com a instalação finalizada irá aparecer na tela o Instalador do Visual Studio com diversas opções de componentes para possibilitar e auxiliar nas criações dos mais diversos softwares, sintá-se livre para explorar as opções e instalar as que você tiver interesse ou irá utilizar futuramente. Durante o curso utilizaremos apenas o framework .NET para Desktop, portanto marque a opção abaixo:



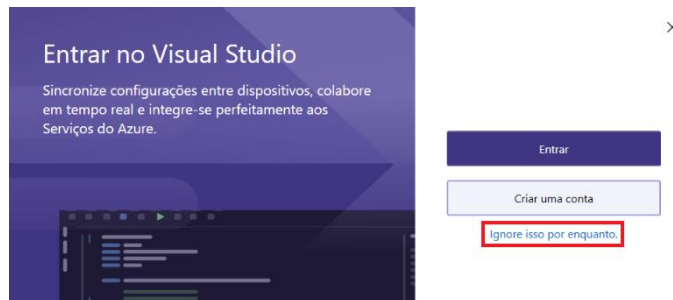
- Ao selecionar um componente, no canto direito do instalador, temos ainda a opção de instalar ferramentas adicionais caso necessário, no caso do curso não iremos precisar de nenhuma ferramenta além das padrões do componente portanto mantenha os opcionais marcados por padrão e então clique em “Instalar”:



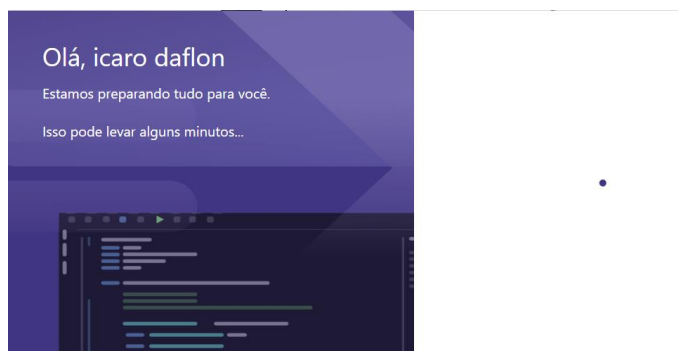
- Agora é só aguardar a instalação finalizar.

Abrindo o Visual Studio 2022, Páginas 6 a 9

- Para abrir o Visual Studio 2022, em seu computador, pesquise no menu Iniciar do Windows o programa “Visual Studio 2022”.
- Ao abrir o programa, caso seja sua primeira vez abrindo o programa irá aparecer uma tela para fazer o Login ou Cadastrar-se em uma conta da Microsoft (e-mails: @outlook.com ou @hotmail.com). É recomendado estar sempre conectado a sua conta para armazenamento de configurações que preferir, porém caso não deseje entrar em nenhuma conta apenas clique em “Ignore isso por enquanto”:

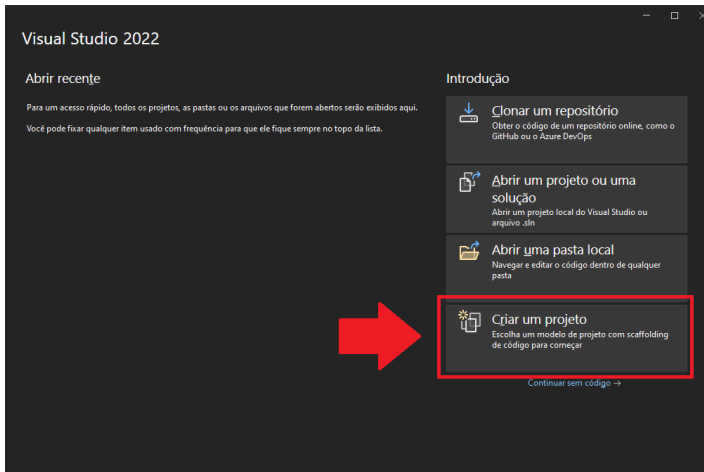


- Caso faça o Login o Visual Studio irá carregar suas configurações anteriores e preparar o ambiente para começarmos:



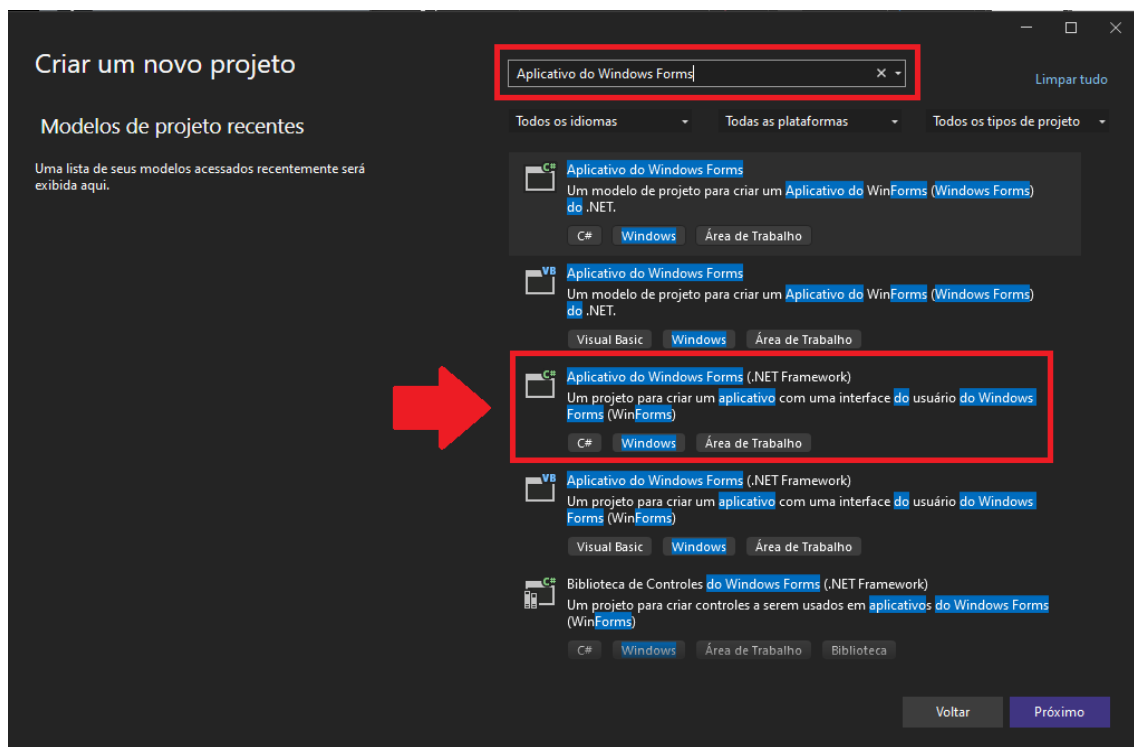
- Após tudo finalizado e com Visual Studio aberto na tela inicial temos 5 opções:
 - Clonar um repositório, ou seja, baixar um arquivo de projeto do GitHub ou outra ferramenta de repositório online.
 - Abrir um projeto ou solução salvas no computador
 - Abrir uma pasta para navegar pelos arquivos de código e alterá-los manualmente
 - Criar um novo projeto
 - E por fim, continuar sem código, ou seja, abrir o Visual Studio sem nenhum arquivo.
- No nosso caso iremos criar um novo projeto portanto clique na opção “Criar Projeto” ou caso o Visual Studio já esteja aberto clicar no menu “File” e após em “Novo Projeto”.

Icaro Daflon C# MOD I



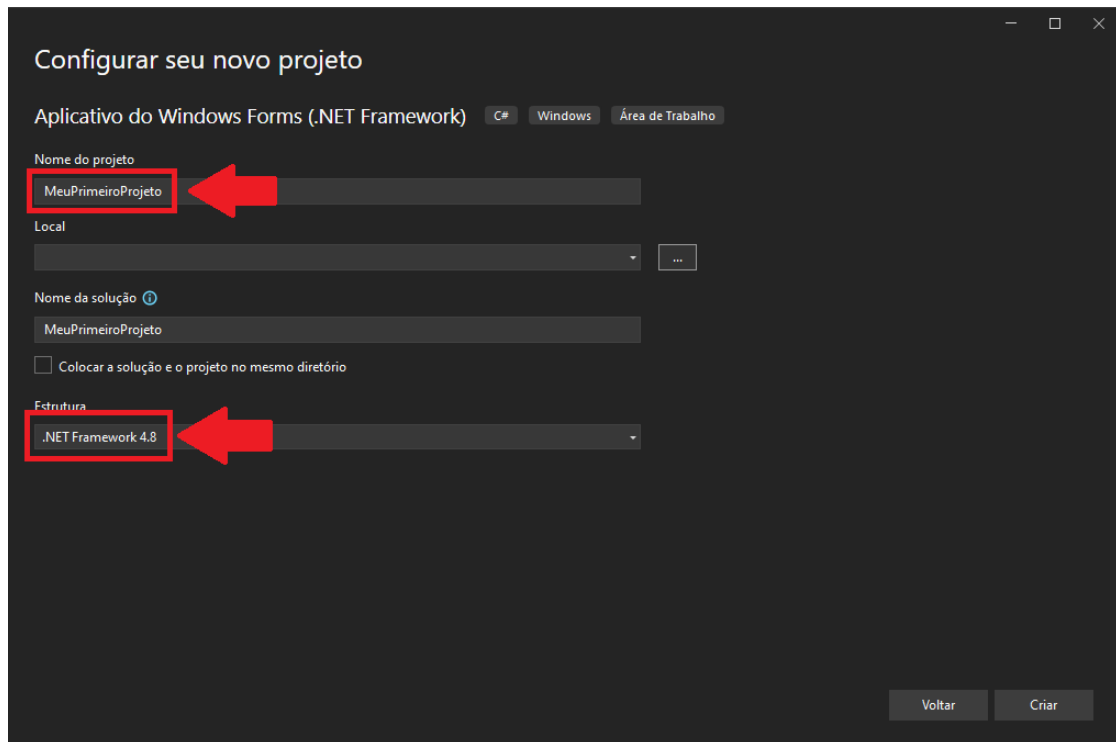
- Na janela de criação de um Novo Projeto temos, primeiramente, vários modelos de criação para os mais diversos Softwares. O que iremos aprender nestas aulas será a criação de aplicativos de Formulários, portanto utilizando a barra de pesquisa na parte superior localize e selecione a opção:

“Aplicativo do Windows Forms (.NET Framework)”

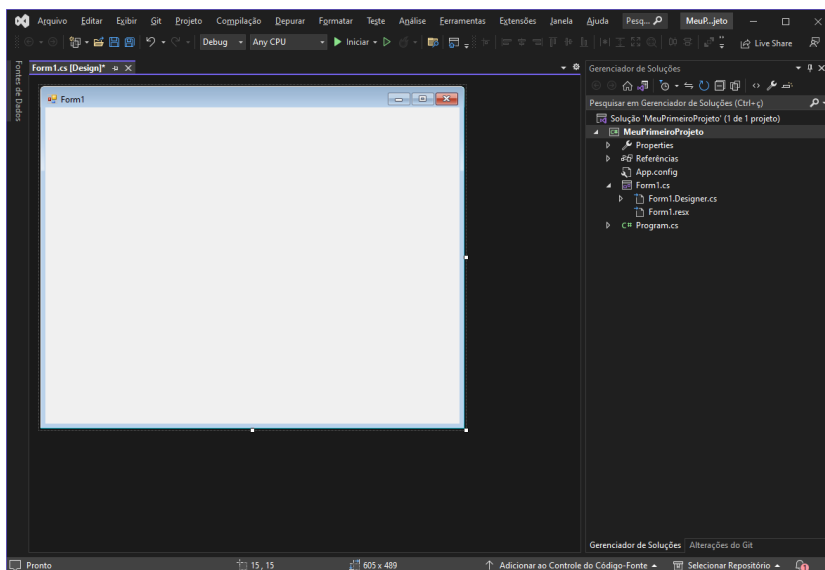


- É muito importante que se atente na criação do projeto que seja a opção (.NET Framework) e também que possua a marcação do C# nele. Após selecionar corretamente, clique em “Próximo”.
- Na janela seguinte iremos configurar o nome do projeto, o local onde será salvo e a versão do Framework utilizado. Você irá configurar o local de acordo com onde deseja salvar seu projeto, sendo recomendo ter um local específico para gravação dos projetos da aula para melhor organização e estudo. As demais especificações devem estar como abaixo e após clique em “Criar”:

Icaro Daflon C# MOD I

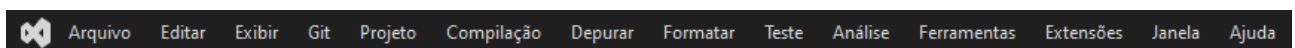


- Após clicar em criar o projeto deve estar aberto normalmente como abaixo:



Componentes do Visual Studio, Páginas 9 a 15

- As informações desta aula permanecem a mesma portanto irei apenas atualizar os prints de cada item.
- Barra de menu:



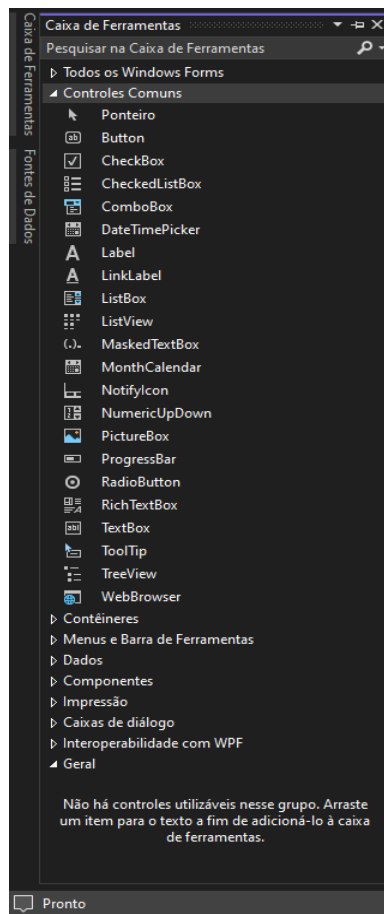
- Barra de Ferramentas:



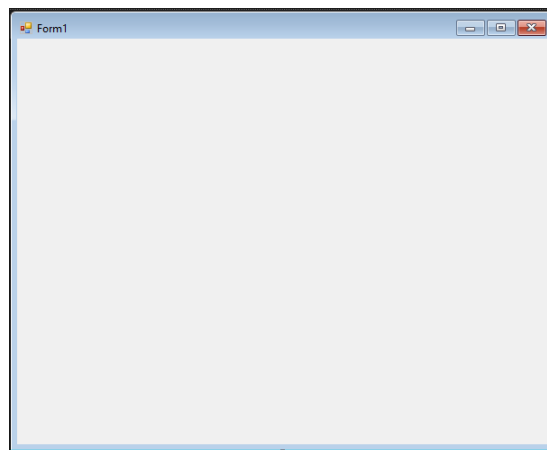
Icaro Daflon

C# MOD I

- Caixa de Ferramentas (Caso a caixa de ferramentas não esteja sendo exibida basta clicar no menu “Exibir” e em seguida em “Caixa de Ferramentas” para habilitá-la no canto esquerdo da tela):

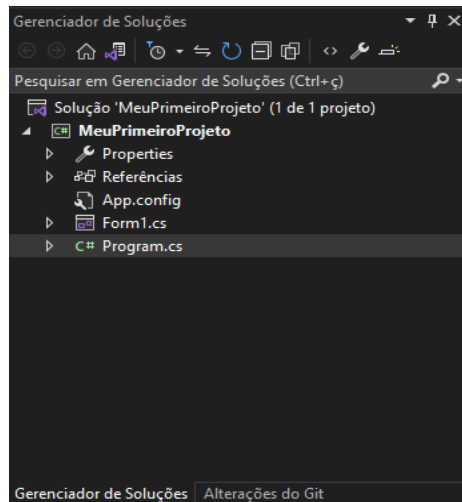


- Formulário Form1:

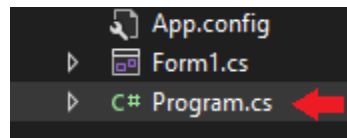


Icaro Daflon C# MOD I

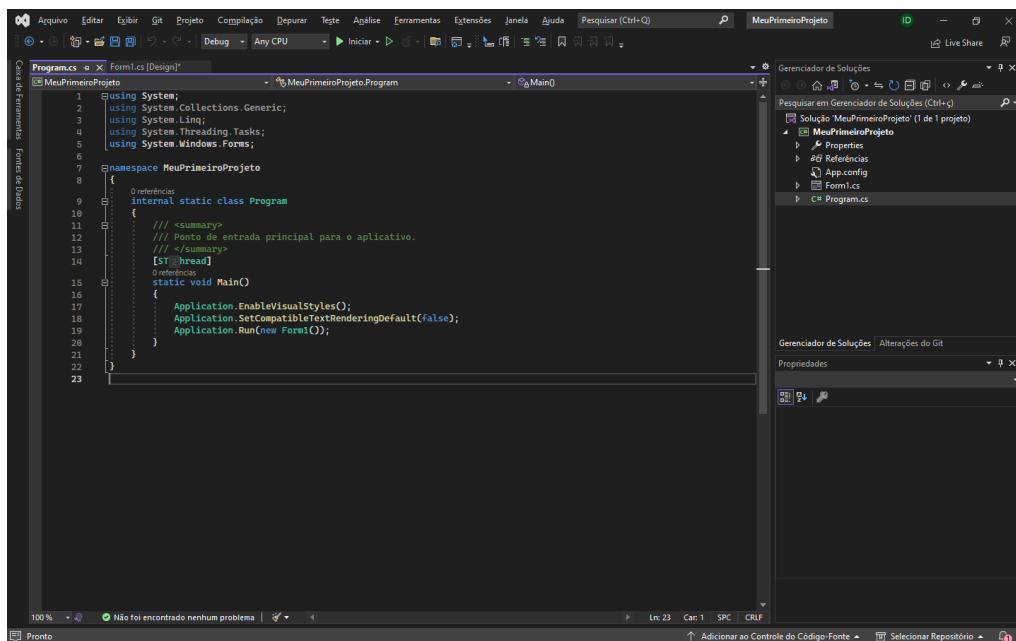
- Gerenciador de Soluções:



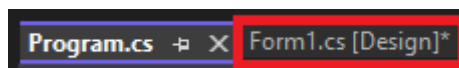
- Arquivo Programa.cs:



- Code and Text Editor:

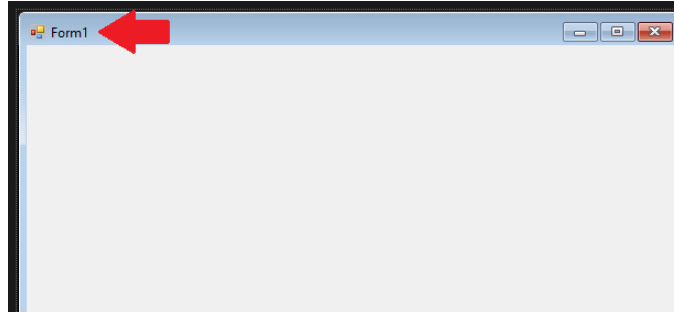


- Clicar na aba Form1.cs [Design]:

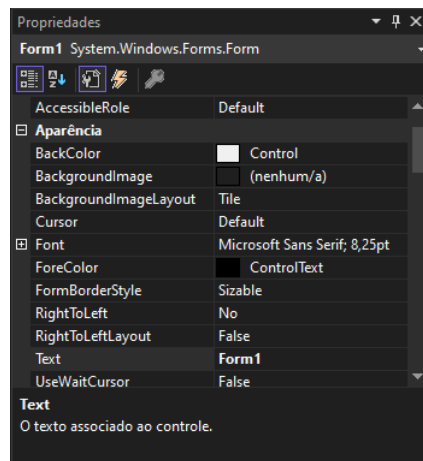


Icaro Daflon C# MOD I

- Duplo clique sobre Form1:



- Propriedades:



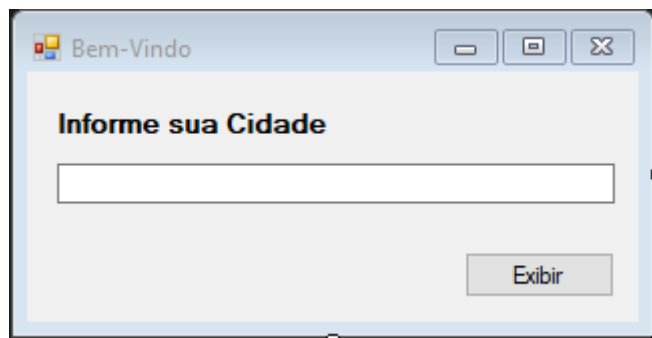
- Barra de Estado:



AULA 02

Formulário Bem-Vindo, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_02 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmBem-Vindo e altere a propriedade Text do mesmo para Bem-Vindo.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label
 - Propriedade Text: Informe sua Cidade
 - TextBox
 - Propriedade (Name): txtCidade
 - Button
 - Propriedade Text: Exibir
 - Propriedade (Name): btnExibir
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Exibir” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

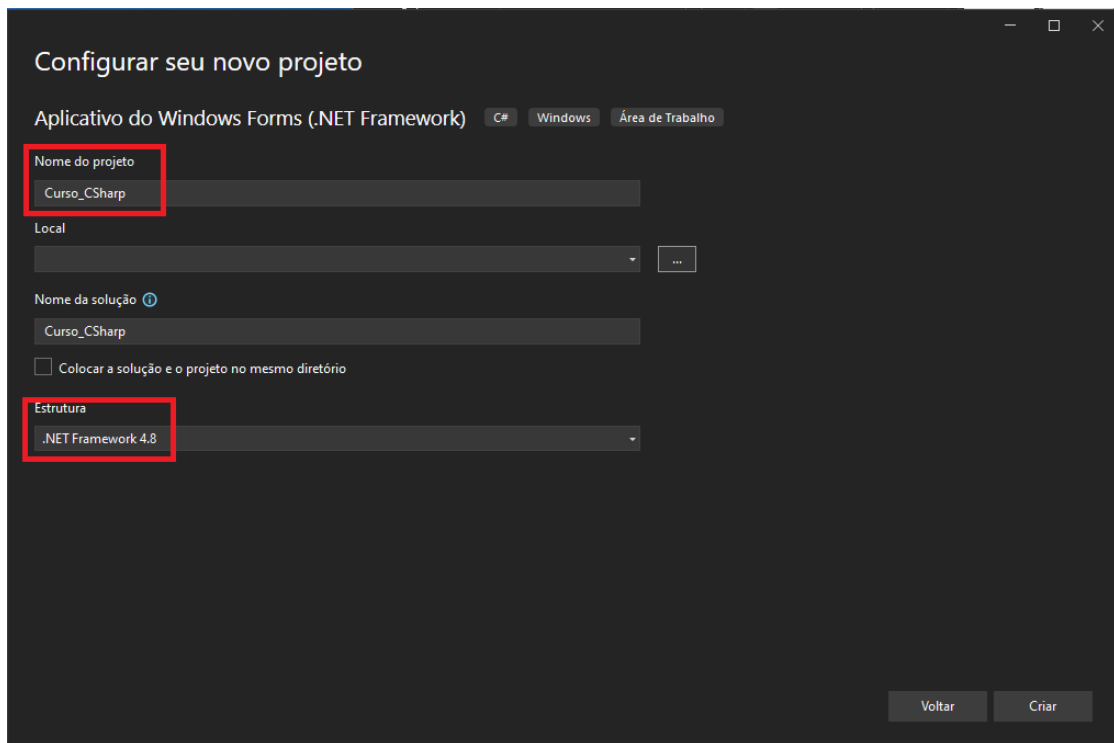
```
string cidade = txtCidade.Text;  
MessageBox.Show("Seja Bem-Vindo à " + cidade);
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Criando o primeiro projeto, Páginas 17 a 29

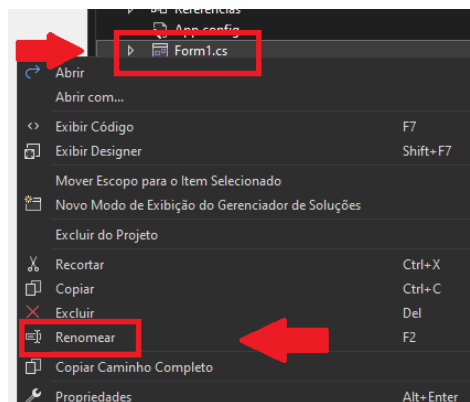
- Abra o Visual Studio 2022 e crie um novo projeto Aplicativo do Windows Forms (.NET Framework) como mostrado na aula anterior.

- Nomeie a aplicação para Curso_CSharp como abaixo (Lembre-se de selecionar o seu Local de Gravação no campo “Local” e de verificar o campo estrutura se está configurado com o “.NET Framework 4.8”):



Nós iremos trabalhar com o Form1.cs, portanto, o renomeie para frmSaudacao.

- Para isso clique com o botão direito do mouse sobre o Form1.cs no Gerenciador de Soluções a direita da tela e clique na opção Renomear:

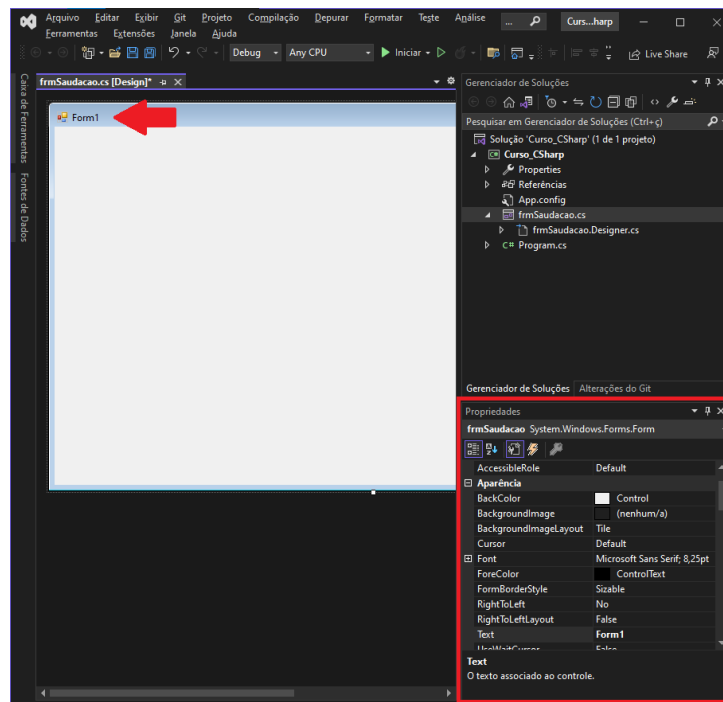


- Digite frmSaudacao e aperte a tecla ENTER.
- Na mensagem que surge aperte Sim, para que o Visual Studio renomeie também todas as referências ao formulário que renomeamos.
- Após renomear o formulário, caso o mesmo não esteja aberto no Code and Text Editor, dê um duplo clique sobre o formulário no Gerenciador de Soluções para abri-lo.

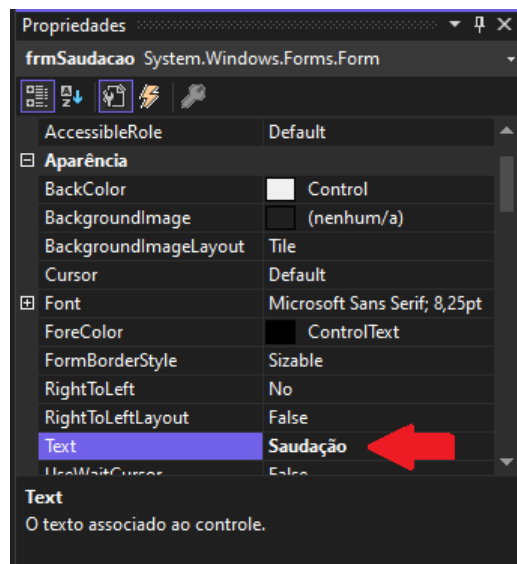
Icaro Daflon

C# MOD I

- Com o formulário aberto no Code and Text Editor dê um clique sobre ele para seleccioná-lo e abrir suas propriedades abaixo do gerenciador de Soluções:

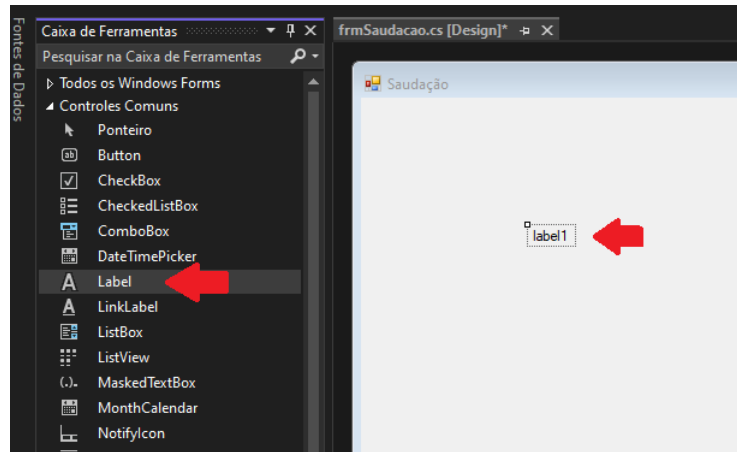


- Na janela de Propriedades, localize a propriedade Text. Essa propriedade se refere ao texto que aparece no topo do formulário. Por padrão vem escrito Form1, mas iremos alterá-lo, portanto, selecione a propriedade Text e escreva “Saudação” e após tecla ENTER. Note que o texto Saudação passa a ser exibido no topo do formulário:

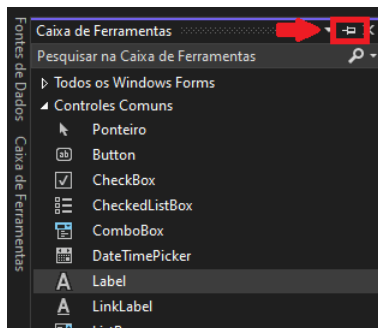


- Clique na guia “Caixa de Ferramentas” a esquerda do Formulário.
- Clique no controle “Label” e, em seguida, clique na parte visível do formulário. Um controle Label será adicionado ao formulário e a Caixa de Ferramentas será ocultada:

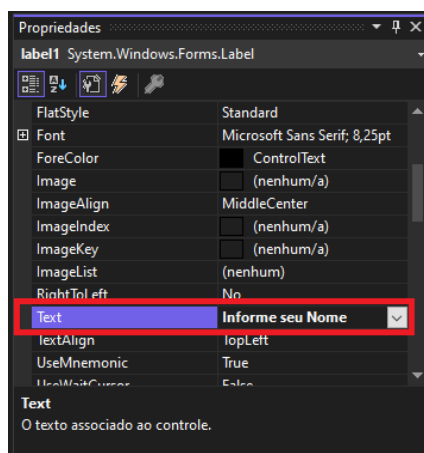
Icaro Daflon C# MOD I



- Se quiser que a Caixa de Ferramentas permaneça visível e não oculte nenhuma parte do formulário, clique no botão “Ocultar Automaticamente”, que se parece com um pino e fica à direita da barra de títulos da Caixa de Ferramentas. Caso clique sobre ela novamente, fará com que ela volte a ocultar-se:

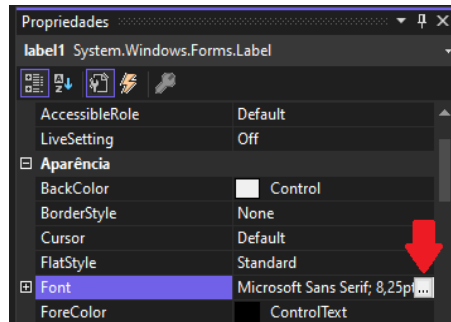


- Você pode clicar e arrastar os controles que adicionou a um formulário para reposicioná-los
- Mova o controle Label, posicionando-o próximo ao canto superior esquerdo do formulário.
- Com o controle Label selecionado, a janela Propriedades agora possui as propriedades do mesmo, podendo assim alterar a fonte, tamanho da letra, cor, e muitas outras características referentes ao controle label. Isso poderá ser feito em todos os controles inseridos na tela, cada um com suas propriedades específicas.
- Na janela Propriedades, encontre a propriedade Text e altere para “Informe seu Nome”

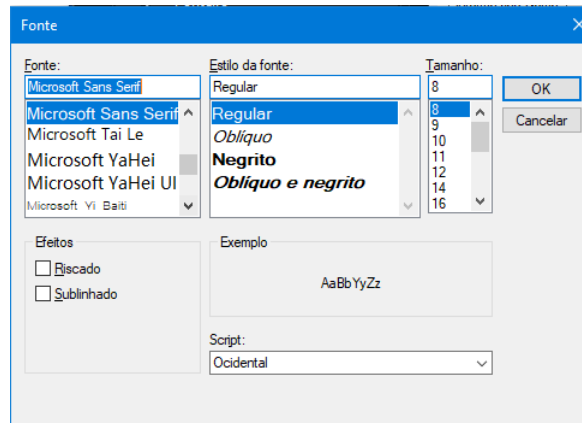


- Utilize a barra de rolagem da janela Propriedades e procure pela propriedade “Fonte”
- Dê um clique sobre a propriedade Fonte e note que agora aparece um botão com reticências. Clique sobre esse botão:

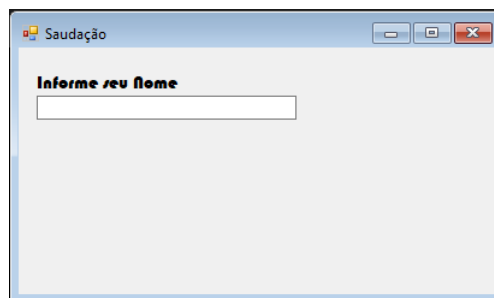
Icaro Daflon C# MOD I



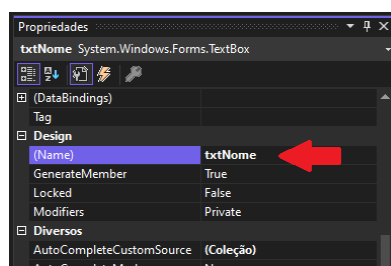
- Ao clicar sobre o botão, é apresentada uma tela para a configuração da fonte do texto que se encontra na Label.



- Altere a fonte de acordo com sua preferência e clique no botão OK.
- Na caixa de ferramentas, clique em “TextBox” e, em seguida, clique no formulário. Agora um controle de caixa de texto foi adicionado ao seu formulário.
- Mova o TextBox para posicioná-lo logo abaixo do controle Label, inserido anteriormente:

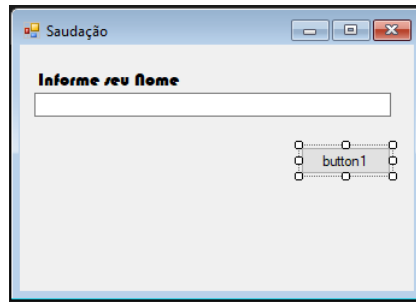


- Com o TextBox selecionado, na janela Propriedades altere o valor da propriedade (Name), localizada no grupo “Design” para “txtNome”, assim como abaixo:

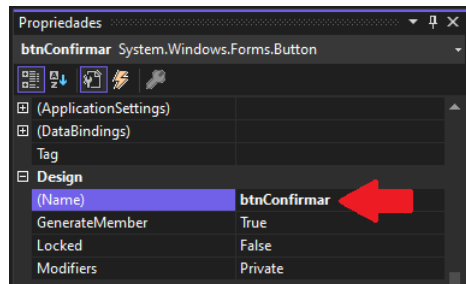


- Novamente na Caixa de Ferramentas, clique em Button e, em seguida, clique no formulário.
- Arraste o controle Button para logo abaixo do TextBox, no formulário:

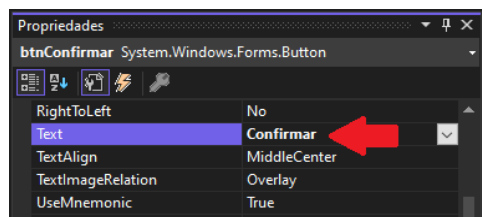
Icaro Daflon C# MOD I



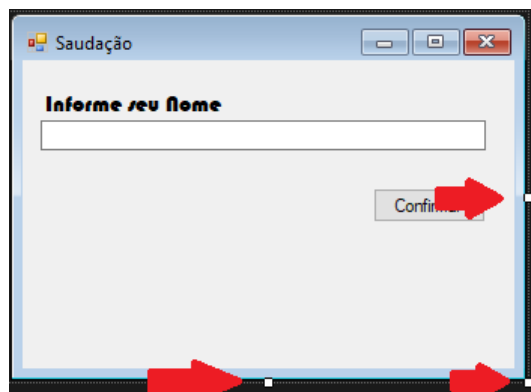
- Utilizando a janela Propriedades, altere a propriedade (Name) do Button para btnConfirmar.



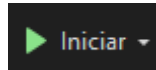
- Altere também a propriedade Text para “Confirmar” e aperte ENTER. Observe que a descrição contida no Button mudou para “Confirmar”.



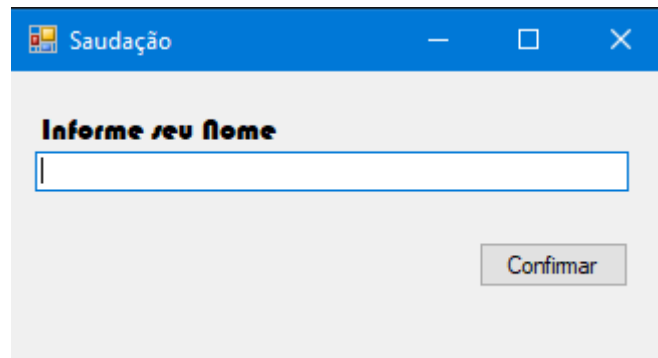
- Clique no formulário e note que algumas alças de redimensionamento (pequenos quadrados) aparecem no contorno:



- Quando quiser redimensionar o formulário, basta clicar sobre uma dessas alças e com o botão do mouse pressionado arrastá-las como desejar.
- Pronto o layout de nossa tela já está pronto, para visualizá-lo vamos clicar sobre o botão “Start” na barra de ferramentas:



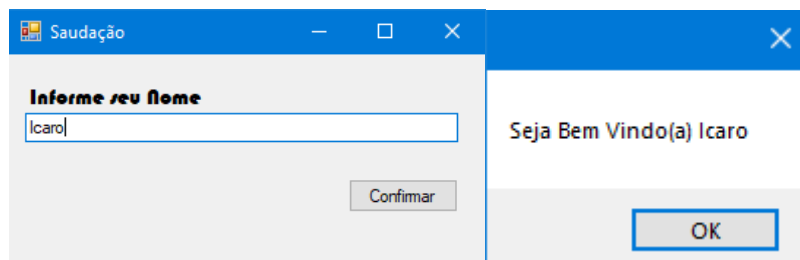
- Outra forma de executarmos nosso programa é utilizando da tecla de atalho F5.
- O aplicativo deve ser executado, exibindo nosso formulário. Você pode digitar seu nome no TextBox e clicar no Button Confirmar, porém, nada acontecerá ainda, pois ainda iremos adicionar as linhas de código para processar o evento de click.



- Clique no botão fechar do Formulário que está sendo executado para que retorne ao ambiente de desenvolvimento.
- Com a execução do programa finalizado. Dê um duplo clique sobre o Button “Confirmar”:
- Ao dar um duplo clique sobre o Button o código do formulário será aberto no Code and Text Editor, ou seja, a classe referente ao formulário será exibida com um novo método chamado btnConfirmar_Click. Este é um método gerado automaticamente pelo Visual Studio que será chamado sempre que o usuário clicar sobre o Button “Confirmar”.
- Deixe o método btnConfirmar_Click da seguinte maneira:

```
private void btnConfirmar_Click(object sender, EventArgs e)
{
    MessageBox.Show("Seja Bem Vindo(a) " + txtNome.Text);
}
```

- Pressione F5 para executar o projeto.
- Quando o formulário aparecer, digite seu nome na caixa de texto e, em seguida, clique no botão Confirmar. Uma caixa de mensagem de boas-vindas aparecerá com seu nome.



- Clique em OK na caixa de mensagem e ela será fechada.
- Aperte CTRL + S para salvar o projeto e, em seguida, feche o Visual Studio.

AULA 03

Formulário Celular, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_03 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCelular e altere a propriedade Text do mesmo para Celular.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Propriedade Text: Marca
 - Label2
 - Propriedade Text: Preço
 - Label3
 - Propriedade Text: Ano de Lançamento
 - TextBox1
 - Propriedade (Name): txtMarca
 - TextBox2
 - Propriedade (Name): txtPreco
 - TextBox3
 - Propriedade (Name): txtLancamento
 - Button
 - Propriedade Text: Mostrar
 - Propriedade (Name): btnMostrar
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Mostrar” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
string marca = txtMarca.Text;
int preco = int.Parse(txtPreco.Text);
int lancamento = int.Parse(txtLancamento.Text);

MessageBox.Show("O celular é da marca " + marca +
    ", custa R$ " + preco +
    " e foi fabricado em " + lancamento);
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Nesta aula a parte teórica está toda de acordo e não há nenhuma mudança além dos erros a serem comentados abaixo:

Nomenclatura de variáveis, Página 33

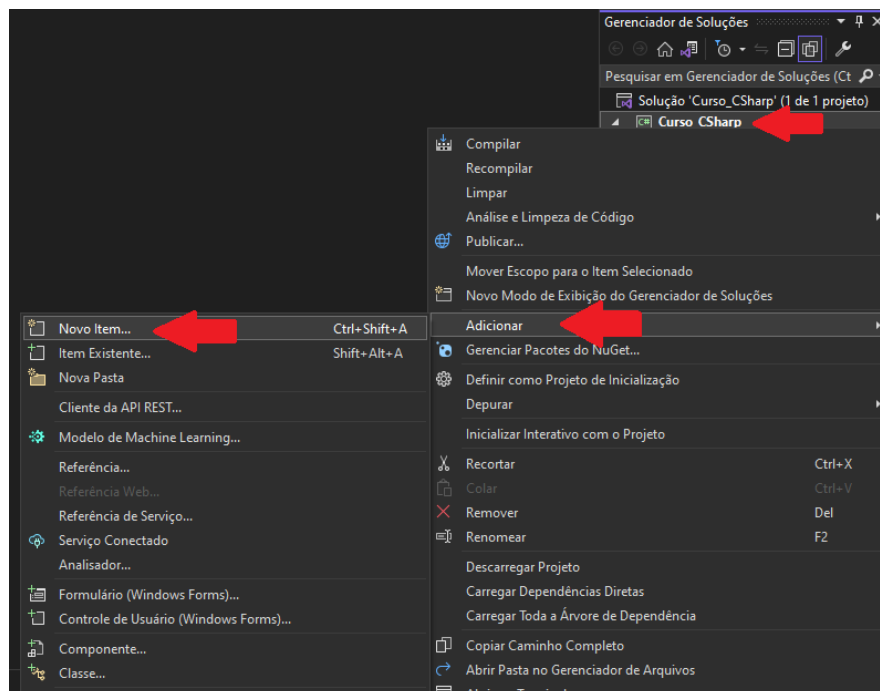
- No segundo tópico onde diz sobre o primeiro caractere do nome de uma variável ele cita que “deve começar sempre com uma letra maiúscula”, porém, mesmo que não cause nenhum tipo de erro no programa é convencional que variáveis comecem por letra minúscula.

Declarando variável, Página 34

- Alterar os exemplos “int Idade;” e “Idade = 25;” colocando-os com a inicial minúscula para adequar-se ao item acima.

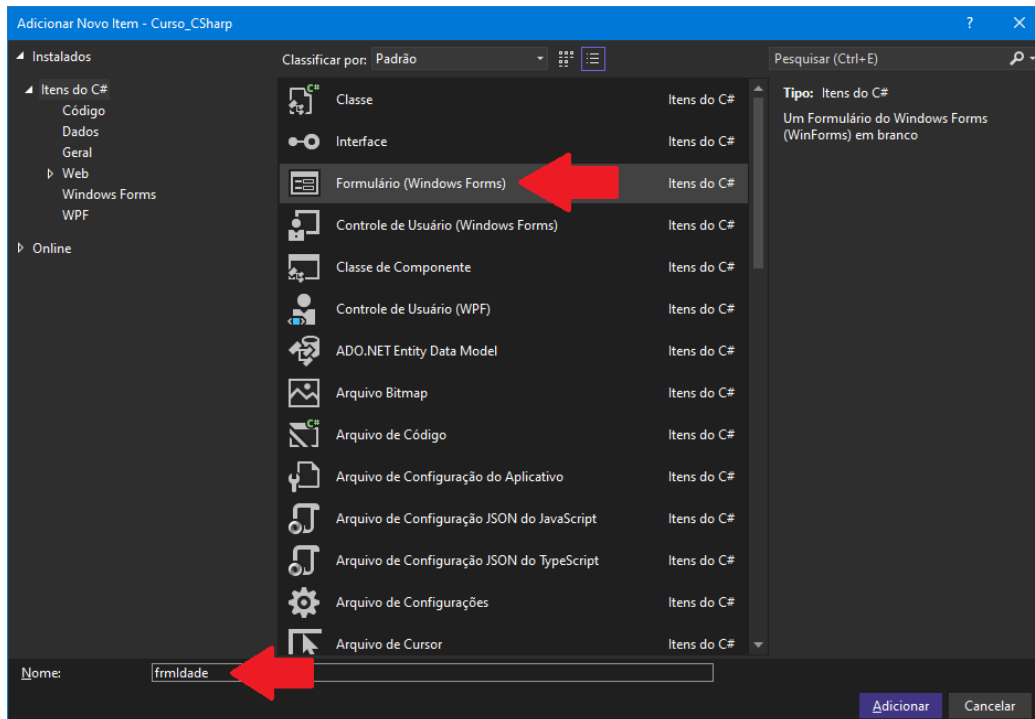
Passo a Passo, Páginas 35 a 41:

- Peça a seu instrutor a pasta “Aula 02”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 02”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, crie um novo formulário, clicando com o botão direito do mouse no projeto “Curso_CSharp”, seguindo para “Adicionar > Novo Item...”



Icaro Daflon C# MOD I

- Uma tela será aberta para que você escolha o tipo de item que seja adicionar ao seu projeto. Escolha “Formulário (Windows Forms)” e nomei-o de “frmIdade”.

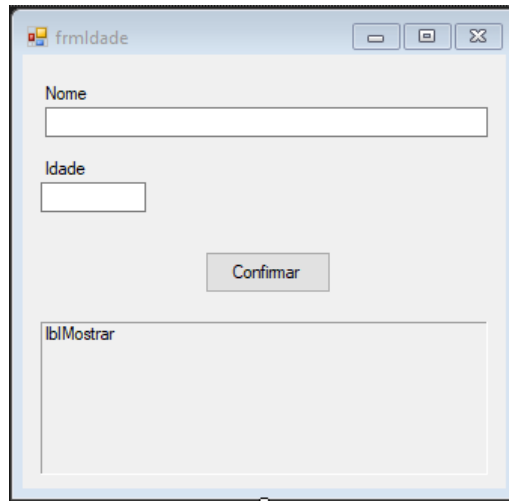


- Com o formulário aberto, adicione um controle Label e na propriedade Text defina como “Nome”
- Adicione outro controle Label, com a propriedade Text “Idade”
- Adicione um Button com a propriedade Name “btnConfirmar” e a propriedade Text “Confirmar”
- Adicione um controle TextBox com a propriedade Name “txtNome” e outro TextBox com a propriedade Name “txtIdade”
- Por último, adicione outro controle Label e escreva lblMostrar em sua propriedade Name e em sua propriedade Text

O controle Label, possui uma propriedade que se chama AutoSize. Essa propriedade, quando possui o valor True, o tamanho do controle é direcionado de acordo com o valor que possuir, mas se estiver como False, é o desenvolvedor que definirá o tamanho do controle

- Altere o valor do controle AutoSize do Label “lblMostrar” para False e aumente o tamanho do controle na tela, pois essa Label será utilizada para mostrar uma mensagem.
- Altere também a propriedade BorderStyle da Label “lblMostrar” para Fixed3D e repare que a borda da Label mudou de forma.
- Organize os itens inseridos acima para que fiquem como abaixo (Caso precise redimensione o tamanho do formulário para ficar do tamanho adequado):

Icaro Daflon C# MOD I



- Dê um duplo clique no botão “Confirmar” para ir para o código do botão, criando o event click.
- No event click do botão, crie uma variável do tipo string, para receber o nome:

```
string strNome;
```

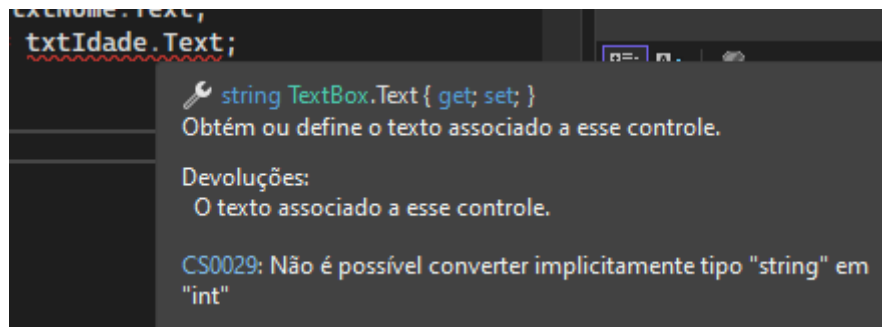
- Nesse mesmo evento, crie outra variável do tipo int, que receberá o valor da idade:

```
int intIdade;
```

- Adicione o valor dos campos txtNome e txtIdade nas variáveis strNome e intIdade, respectivamente, como abaixo:

```
strNome = txtNome.Text;  
intIdade = txtIdade.Text;
```

- Como podemos ver ao escrevermos o código acima, podemos notar que o Visual Studio indica um erro (Sublinhado Vermelho) onde atribuímos a variável intIdade. Passe o mouse em cima do erro e veja que abre uma janela descrevendo o erro:



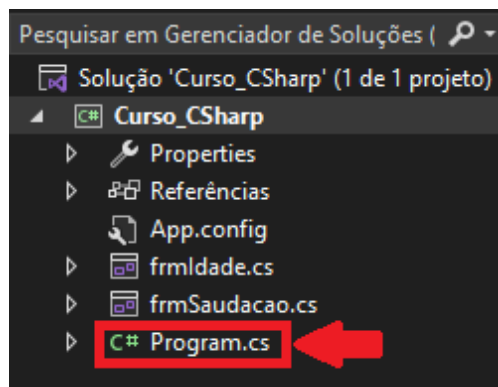
- Como podemos ver, o erro ocorreu, pois, foi atribuído a variável intIdade, que é do tipo int. um valor string. Por padrão todos os valores que são informados em um controle TextBox, são do tipo string. Portanto para corrigirmos esse problema é necessário converter o valor do TextBox de string para int. Para isso, existe uma classe, do próprio Visual Studio, responsável pela conversão de valores string.
- Altere a linha que está com erro pela seguinte instrução:

```
intIdade = int.Parse(txtIdade.Text);
```

- Após atribuir o valor das TextBox nas variáveis, é preciso recuperar esse valor e mostra-lo no controle label
- Adicione, abaixo do comando de atribuição da variável `intIdade`, a seguinte linha de código:

```
lblMostrar.Text = "O(A) " + strNome + " tem " + intIdade + " anos.";
```

- Nesse caso o sinal de “+” está sendo usado para concatenar (Juntar o final da string a esquerda com o início da string à direita do sinal) os valores strings. Esse sinal é um dos valores aritméticos utilizados no Visual Studio e que veremos na próxima aula.
- Com o código do botão completo, agora devemos compilar esse formulário, porém, se executarmos como está o formulário a ser aberto será o `frmSaudacao`. O formulário a ser aberto é definido pela classe `Program.cs`
- Dê um duplo clique na classe `Program.cs`:



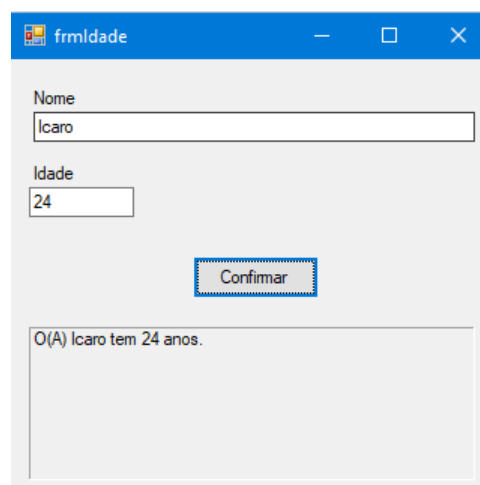
- Com o código do arquivo `Program.cs` aberto no Code and Text Editor, Localize a seguinte linha:

```
Application.Run(new frmSaudacao());
```

- Altere a linha acima alterando o nome do formulário `frmSaudacao`, para o formulário `frmIdade`:

```
Application.Run(new frmIdade());
```

- Execute o programa apertando a tecla F5, digite um nome no campo Nome e uma idade no campo Idade e. após clicar em Confirmar, note a alteração do controle `lblMostrar` com o nome e idade digitados:



- Com isso finalizamos esta aula, salve o projeto pressionando `Ctrl + S` e em seguida feche o Visual Studio.

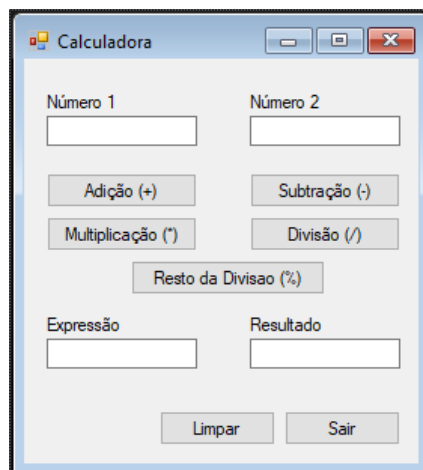
AULA 04

Formulário Calculadora, Aula interativa

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_04 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCalculadora e altere a propriedade Text do mesmo para Calculadora.
- Na janela que irá aparecer clique em “Sim”.
- Insira 4 controles Label, as propriedades Text de cada serão respectivamente: “Número 1”, “Número 2”, “Expressão” e “Resultado”.
- Insira agora 4 controles TextBox e alteres suas propriedades Name para as seguintes: “txtNum1”, “txtNum2”, “txtExpressao” e “txtResultado”.
- Nas TextBox “txtExpressao” e “txtResultado” altere também a propriedade “Enable” para False para que não seja possível que o usuário escreva ou edite o conteúdo dela visto que elas serão usadas apenas para mostrar o Resultado.
- Crie 7 botões com as seguintes configurações:

Propriedade Name	Propriedade Text
btnAdicao	Adição (+)
btnSubtracao	Subtração (-)
btnMultiplicacao	Multiplicação (*)
btnDivisao	Divisão (/)
btnRestoDivisao	Resto da Divisao (%)
btnLimpar	Limpar
btnSair	Sair

- Organize os componentes inseridos acima da seguinte forma:



- Não se esqueça de que os componentes TextBox devem estar abaixo de seus componentes Label respectivos, exemplo: O TextBox “txtNum1” estará logo abaixo da Label “Número 1” e em diante.
- Agora iremos configurar os event click para cada Button inserido.
- Portanto dê um duplo clique sobre o botão “Adição (+)” e insira o código abaixo:

```
int num1 = int.Parse(txtNum1.Text);  
int num2 = int.Parse(txtNum2.Text);  
int resultado = num1 + num2;  
  
txtExpressao.Text = num1 + " + " + num2;  
txtResultado.Text = resultado.ToString();
```

- Note que nas 3 primeiras linhas estamos declarando 3 variáveis. As 2 primeiras para armazenar os valores das caixas de texto digitadas e a 3ª para calcular o resultado.
- Após a declaração das variáveis as 2 linhas subsequentes servem para exibir o resultado que esperamos nas TextBox's “txtExpressao” e “txtResultado”. Note que para a TextBox “txtExpressao” utilizamos o operador “+” para concatenar uma string junto ao valor das variáveis num1 e num2, ao fazermos isso o conteúdo delas é automaticamente convertido para uma string. Já para a TextBox “txtResultado” por não haver nenhuma string a ser concatenada devemos utilizar do método “.ToString()” para converter o valor da mesma para string e podermos atribuí-la a TextBox que criamos.
- Agora crie os demais botões aritméticos seguindo o padrão acima, e fazendo as alterações necessárias para adequá-las a cada operação.
- Para o botão “Resto da Divisão (%)” iremos incluir uma condição para caso o resultado da operação seja igual a 0 apareça uma MessageBox informando o usuário. Para isso após a ultima linha do botão inclua o seguinte código:

```
if(resultado == 0)  
{  
    ...  
    MessageBox.Show("O resto da divisão é Zero!");  
}
```

- Após concluir os botões para realizar os cálculos aritméticos iremos criar o botão Limpar, portanto dê um duplo clique sobre ele e digite o código abaixo:

```
txtNum1.Text = null;  
txtNum2.Text = null;  
txtResultado.Text = null;  
txtExpressao.Text = null;
```

- Note que para limpar os campos apenas alteramos o valor deles para null fazendo com que fiquem vazios.
- Por fim dê um duplo clique no botão Sair e digite o código abaixo:

```
Close();
```

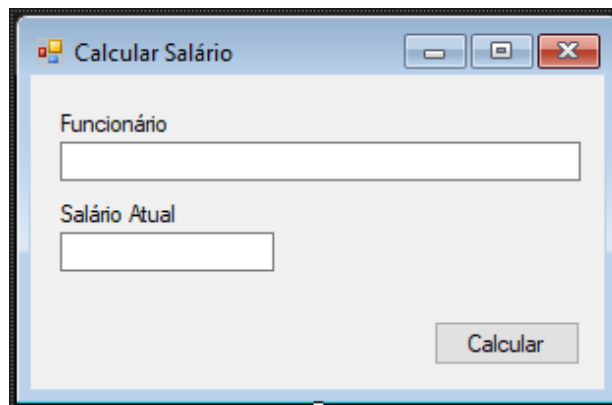
- Abra o formulário e teste as funções incluídas e ao final, salve o projeto e feche o Visual Studio.

Páginas 44 a 48

- Conteúdo teórico de acordo, não há alterações

Passo a Passo, Páginas 49 a 53:

- Peça a seu instrutor a pasta “Aula 03”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 03”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, insira um novo formulário com o nome de frmCalculoSalario.
- Altere a propriedade Text do formulário para Calcular Salário.
- Insira 2 controles Label com as seguintes configurações:
 - Label 1
 - Text: Funcionário
 - Label 2
 - Text: Salário Atual
- Insira 2 controles TextBox com as seguintes configurações:
 - TextBox1
 - Name: txtNomeFuncionario
 - TextBox2
 - txtSalarioAtual
- Insira um controle Button com as seguintes configurações:
 - Button1
 - Name: btnCalcular
 - Text: Calcular
- Organize os componentes inseridos acima da seguinte forma:



- Dê um duplo clique no botão Calcular e no código do event click digite o código abaixo:

Icaro Daflon
C# MOD I

```
string nomeFuncionario;  
int salarioAtual;  
int valorAumento;  
int totalSalario;  
  
nomeFuncionario = txtNomeFuncionario.Text;  
salarioAtual = int.Parse(txtSalarioAtual.Text);  
  
if(salarioAtual < 1000)  
{  
    valorAumento = salarioAtual * 5 / 100;  
    totalSalario = salarioAtual + valorAumento;  
    MessageBox.Show(nomeFuncionario + " você terá " +  
        valorAumento + " reais de aumento. Seu salário total será " +  
        totalSalario + " reais", "Total Salário");  
}  
txtNomeFuncionario.Focus();
```

Obs.: Digite todo o código do MessageBox na mesma linha, facilitando a identificação do sistema e o funcionamento do código.

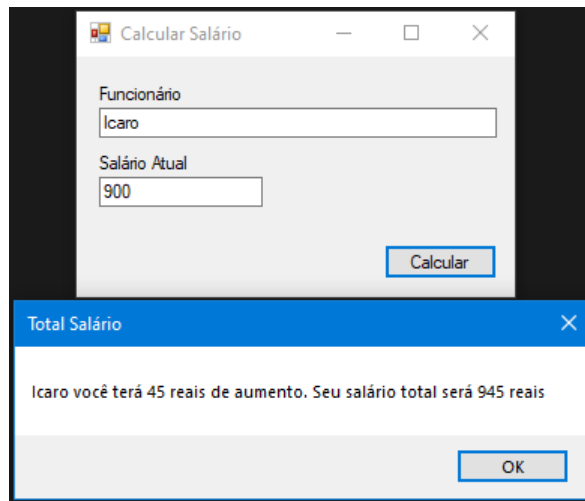
- A explicação das linhas de código permanece a mesma apenas alterando algumas partes do código, irei colocar abaixo os prints de cada parte do código atualizado:

```
string nomeFuncionario;  
  
int salarioAtual;  
  
int valorAumento;  
  
int totalSalario;  
  
nomeFuncionario = txtNomeFuncionario.Text;  
salarioAtual = int.Parse(txtSalarioAtual.Text);  
  
if(salarioAtual < 1000)  
{  
  
    valorAumento = salarioAtual * 5 / 100;  
  
    totalSalario = salarioAtual + valorAumento;  
    totalSalario = salarioAtual + valorAumento;  
  
    MessageBox.Show(nomeFuncionario + " você terá " +  
        valorAumento + " reais de aumento. Seu salário total será " +  
        totalSalario + " reais", "Total Salário");  
  
}  
  
txtNomeFuncionario.Focus();
```

- Altere a classe Program.cs para que o formulário a ser aberto seja o frmCalculoSalario ficando da seguinte maneira o código:

```
Application.Run(new frmCalculoSalario());
```

- Salve essa classe e execute o programa, teste o formulário Calcular Salário. Digite o nome do funcionário e o seu salário, que deve ser em formato de número inteiro. Coloque um salário abaixo de 1000 reais e clique em calcular. Note que uma mensagem com o valor do aumento e o valor total do salário irá aparecer. Coloque também um salário acima de 1000 e perceba que nada acontece, pois, a execução do if não será realizada.



- Após testar feche a execução do sistema.

Utilizando Instruções IF... Else, Páginas 53 a 55

- Conteúdo teórico de acordo, não há alterações

Passo a Passo, Páginas 55 e 56:

- Abra o formulário Calcular Salário e no código do event click do botão “Calcular” acrescente as seguintes linhas de código, logo abaixo do comando if, após o encerramento da MessageBox, logo abaixo do }.

```
else
{
    MessageBox.Show(nomeFuncionario +
        " você não terá aumento de salário.",
        "Aviso");
}
```

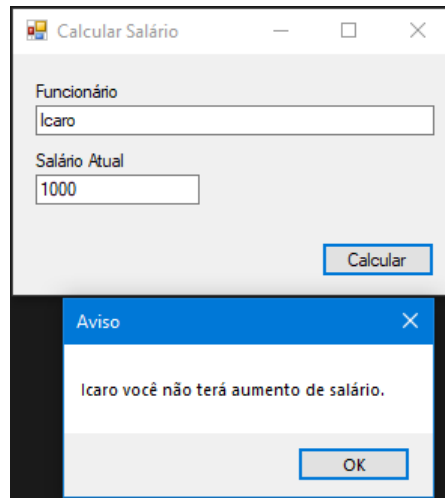
- O código total ficará da seguinte maneira:

```
string nomeFuncionario;
int salarioAtual;
int valorAumento;
int totalSalario;

nomeFuncionario = txtNomeFuncionario.Text;
salarioAtual = int.Parse(txtSalarioAtual.Text);

if(salarioAtual < 1000)
{
    valorAumento = salarioAtual * 5 / 100;
    totalSalario = salarioAtual + valorAumento;
    MessageBox.Show(nomeFuncionario + " você terá " +
        valorAumento + " reais de aumento. Seu salário total será " +
        totalSalario + " reais", "Total Salário");
}
else
{
    MessageBox.Show(nomeFuncionario +
        " você não terá aumento de salário.",
        "Aviso");
}
txtNomeFuncionario.Focus();
```

- Com a adição do comando else quando o salário do funcionário for maior que 1000 reais, ou seja, a condição contida no if for falsa, o código contido no else será executado. Nesse caso, será apresentada ao usuário uma tela com um aviso de que não terá aumento.
- Salve o formulário e execute o projeto, testando agora com um valor acima de 1000 reais.



- Faça mais alguns testes e, ao final, feche o Visual Studio.

AULA 05

Formulário Comparando Idade, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_05 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmComparandoIdade e altere a propriedade Text do mesmo para ComparandoIdade.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Propriedade Text: Nome Pessoa 1
 - Label2
 - Propriedade Text: Nome Pessoa 2
 - Label3
 - Propriedade Text: Idade Pessoa 1
 - Label4
 - Propriedade Text: Idade Pessoa 2

- Label5
 - Propriedade Text: *Apague o conteúdo*
 - Propriedade (Name): lblMostrar
 - Propriedade AutoSize: False
 - Propriedade BorderStyle: FixedSingle
 - TextBox1
 - Propriedade (Name): txtNomePessoa1
 - TextBox2
 - Propriedade (Name): txtNomePessoa2
 - TextBox3
 - Propriedade (Name): txtIdadePessoa1
 - TextBox3
 - Propriedade (Name): txtIdadePessoa2
 - Button
 - Propriedade Text: Verificar
 - Propriedade (Name): btnVerificar
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Verificar” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

Icaro Daflon
C# MOD I

```
string nomePessoa1 = txtNomePessoa1.Text;
string nomePessoa2 = txtNomePessoa2.Text;
int idadePessoa1 = int.Parse(txtIdadePessoa1.Text);
int idadePessoa2 = int.Parse(txtIdadePessoa2.Text);

if (idadePessoa1 == idadePessoa2 * 10 & idadePessoa2 != 0)
    lblMostrar.Text = nomePessoa1 + " é 10x mais velho que " + nomePessoa2;
else if (idadePessoa1 == idadePessoa2 * 2 & idadePessoa2 != 0)
    lblMostrar.Text = nomePessoa1 + " tem o dobro da idade de " + nomePessoa2;
else if (idadePessoa1 == idadePessoa2)
    lblMostrar.Text = "As duas idades são iguais!";
else
    lblMostrar.Text = nomePessoa1 + " tem " + idadePessoa1 +
        " anos e " + nomePessoa2 + " tem " + idadePessoa2 + "anos";
```

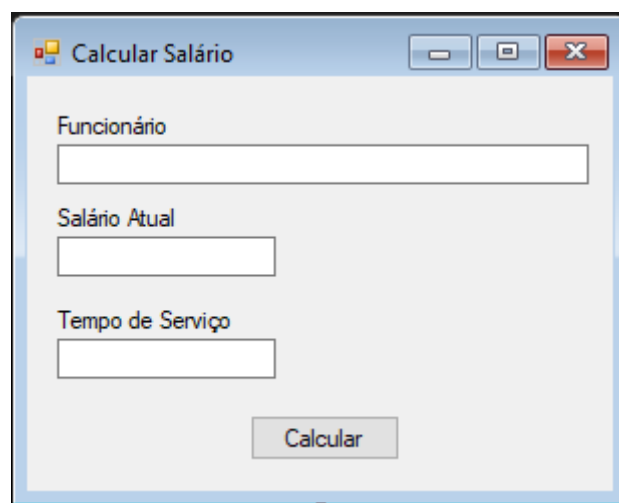
- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Operadores Lógicos, Páginas 59 a 63

Obs.: Alterar os operadores && (AND) e || (OR) para apenas 1 símbolo ficando assim: & (AND) e | (OR).

- Peça a seu instrutor a pasta “Aula 04”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 04”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, abra o formulário frmCalculoSalario dando um duplo clique sobre ele no gerenciador de soluções.
- Adicione os componentes:
 - Label1
 - Propriedade Text: Tempo de Serviço
 - TextBox1
 - Propriedade (Name): txtTempoDeServico

- A tela ficará da seguinte maneira:



- Em seguida, dê um duplo clique no botão Calcular.

- Com o código aberto crie uma nova variável do tipo int, chamada tempoDeServico, para receber o valor do tempo de serviço do funcionário:

```
int tempoDeServico;
```

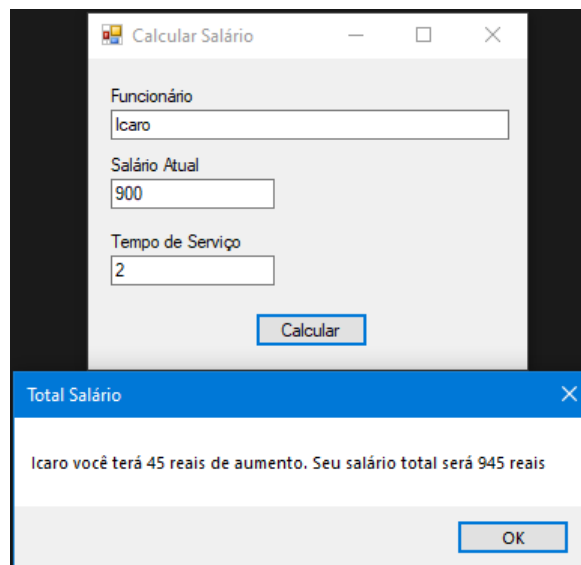
- Adicione o valor do campo txtTempoDeServico à variável tempoDeServico, inserindo o código abaixo:

```
tempoDeServico = int.Parse(txtTempoDeServico.Text);
```

- Agora altere a condição if para que fique da seguinte maneira:

```
if(salarioAtual < 1000 & tempoDeServico > 1)
{
    valorAumento = salarioAtual * 5 / 100;
    totalSalario = salarioAtual + valorAumento;
    MessageBox.Show(nomeFuncionario + " você terá " +
        valorAumento + " reais de aumento. Seu salário total será " +
        totalSalario + " reais", "Total Salário");
}
else
{
    MessageBox.Show(nomeFuncionario +
        " você não terá aumento de salário.",
        "Aviso");
}
txtNomeFuncionario.Focus();
```

- Salve o projeto e execute-o
- Teste as alterações, na caixa Tempo de Serviço digite apenas número (1, 2, 3...)
- Agora, informe um salário abaixo de 1000 reais e um tempo de serviço acima de 1 ano.



- Informe um salário acima de 1000 reais e um tempo de serviço acima de 1 ano:

Icaro Daflon C# MOD I

The screenshot shows a Windows application titled 'Calcular Salário'. It has three text input fields: 'Funcionário' with the value 'Icaro', 'Salário Atual' with the value '1500', and 'Tempo de Serviço' with the value '2'. Below these fields is a 'Calcular' button. An 'Aviso' (Warning) message box is displayed in front of the application, containing the text 'Icaro você não terá aumento de salário.' and an 'OK' button.

- Por último, informe um salário abaixo de 1000 reais e um tempo de serviço de 1 ano:

The screenshot shows the same 'Calcular Salário' application. The 'Salário Atual' field now contains the value '900' and the 'Tempo de Serviço' field contains the value '1'. The 'Calcular' button is still present. The 'Aviso' message box is again displayed with the same text: 'Icaro você não terá aumento de salário.' and an 'OK' button.

- Perceba que o código dentro do if só será executado se as duas condições forem verdadeiras, ou seja, se o funcionário tiver o salário menor que 1000 reais e trabalhar, na empresa, há mais de 1 ano.
- Agora altere o operador & (AND) pelo operador | (OR) na condição do if e teste o funcionamento do mesmo.

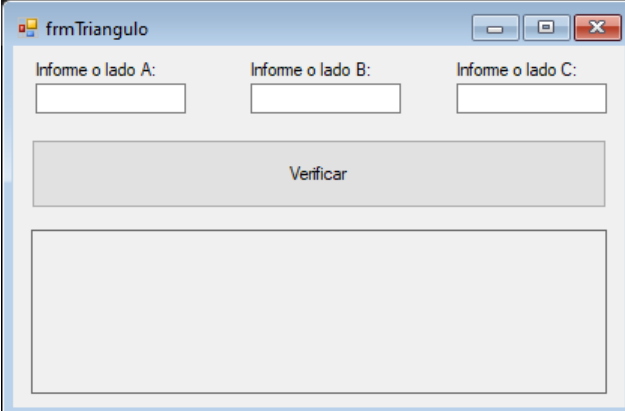
```
if(salarioAtual < 1000 | tempoDeServico > 1)
```

- Após testar, feche a janela do programa

Instruções IF Encadeado, Páginas 65 a 69

- Com o projeto ainda aberto, insira um novo formulário com o nome de frmTriangulo
- Insira os seguintes controles ao formulário e configure-os da seguinte maneira:
 - Label1
 - Text: Informe o lado A:
 - Label2

- Text: Informe o lado B:
 - Label3
 - Text: Informe o lado C:
 - Label4
 - Name: lblMostrar
 - Autosize: False
 - BorderStyle: FixedSingle
 - Text: *Apague o conteúdo*
 - TextBox1
 - Name: txtLadoA
 - TextBox2
 - Name: txtLadoB
 - TextBox3
 - Name: txtLadoC
 - Button1
 - Name: btnVerificar
 - Text: Verificar
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com a parte gráfica pronta, agora iremos inserir o código. Para isso dê um duplo clique sobre o botão “Verificar” e adicione o seguinte código dentro do event click:

Icaro Daflon
C# MOD I

```
double A, B, C;  
A = double.Parse(txtLadoA.Text);  
B = double.Parse(txtLadoB.Text);  
C = double.Parse(txtLadoC.Text);  
  
if (A < B + C & B < A + C & C < A + B)  
{  
    if (A == B & B == C)  
        lblMostrar.Text = "Triângulo Equilátero";  
    else if (A == B | A == C | B == C)  
        lblMostrar.Text = "Triângulo Isósceles";  
    else  
        lblMostrar.Text = "Triângulo Escaleno";  
}  
else  
    lblMostrar.Text = "Valores não formam um triângulo!";
```

- Significado das linhas alterar as imagens de referência:

```
double A, B, C;  
  
A = double.Parse(txtLadoA.Text);  
B = double.Parse(txtLadoB.Text);  
C = double.Parse(txtLadoC.Text);  
  
if (A < B + C & B < A + C & C < A + B)
```

- A seguinte linha há alterações na explicação também:

```
if (A == B & B == C)
```

- Verifica se os lados são iguais.
 - $A == B$: Verifica se o valor da variável A é igual a variável B
 - $B == C$: Verifica se o valor da variável B é igual a variável C
 - Devido ao operador & (AND) não há a necessidade de testar se $A == C$ já que para o if ser verdadeiro as condições $A == B$ e $B == C$ devem ser verdadeiras ao mesmo tempo, automaticamente A é igual a C.

- Continuação dos prints:

```
lblMostrar.Text = "Triângulo Equilátero";  
  
else if (A == B | A == C | B == C)  
    lblMostrar.Text = "Triângulo Isósceles";  
  
else  
    lblMostrar.Text = "Triângulo Escaleno";  
  
lblMostrar.Text = "Valores não formam um triângulo!";
```

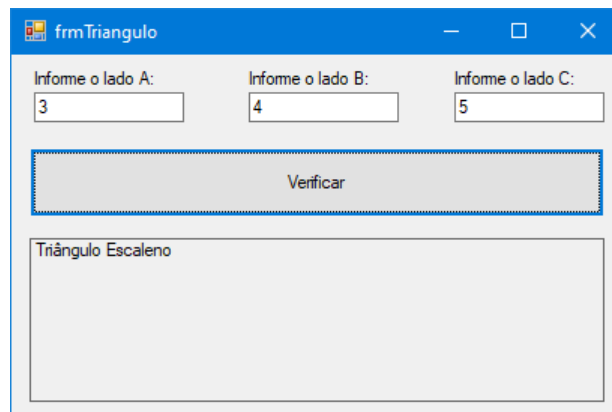
- Altere a classe Program.cs para que o formulário a ser executado seja o frmTriangulo. Alterando a linha onde se encontra o código responsável para que fique da seguinte maneira:

```
Application.Run(new frmTriangulo());
```

- Salve as alterações e execute o programa.

Icaro Daflon
C# MOD I

- Faça o teste com o formulário Triângulo. Coloque números diferentes nos TextBox's e clique no botão verificar para ver o resultado.



The screenshot shows a Windows Forms application window titled "frmTriangulo". It has a standard Windows interface with minimize, maximize, and close buttons. The form contains three text boxes for input, each preceded by a label: "Informe o lado A:" with the value "3", "Informe o lado B:" with the value "4", and "Informe o lado C:" with the value "5". Below these inputs is a button labeled "Verificar". At the bottom of the form, there is a label that reads "Triângulo Escaleno".

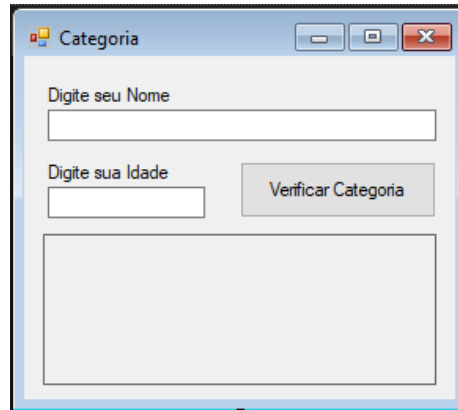
- Após finalizar os testes, finalize a aplicação e, após, feche o Visual Studio.

AULA 06

Formulário Categoria, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_06 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCategoria e altere a propriedade Text do mesmo para Categoria.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Propriedade Text: Digite seu Nome
 - Label2
 - Propriedade Text: Digite sua Idade
 - Label3
 - Propriedade Text: *Apague o conteúdo*
 - Propriedade (Name): lblMostrar
 - Propriedade AutoSize: False
 - Propriedade BorderStyle: FixedSingle
 - TextBox1
 - Propriedade (Name): txtNome

- TextBox2
 - Propriedade (Name): txtIdade
 - Button
 - Propriedade Text: Verificar Categoria
 - Propriedade (Name): btnVerificar
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Verificar” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
string nome = txtNome.Text;
int idade = int.Parse(txtIdade.Text);

if (idade <= 10)
    lblMostrar.Text = nome + "você é da categoria Pré-Mirim!";
else if (idade <= 15)
    lblMostrar.Text = nome + "você é da categoria Mirim!";
else if (idade <= 17)
    lblMostrar.Text = nome + "você é da categoria Infantil!";
else if (idade <= 21)
    lblMostrar.Text = nome + "você é da categoria Juvenil!";
else
    lblMostrar.Text = nome + "você é da categoria Adulto!";
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

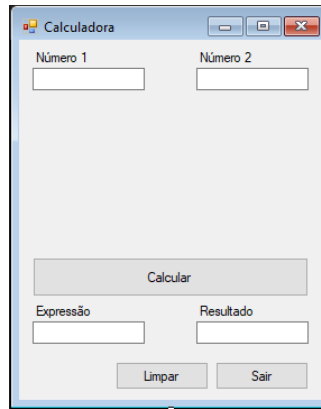
Instruções IF em Cascata, Páginas 72 a 82

- Peça a seu instrutor a pasta “Aula 05”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 05”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, insira um novo formulário com o nome de frmCalculadora.
- Altere a propriedade Text do formulário para Calculadora.
- Insira os seguintes controles ao formulário e configure-os da seguinte maneira:

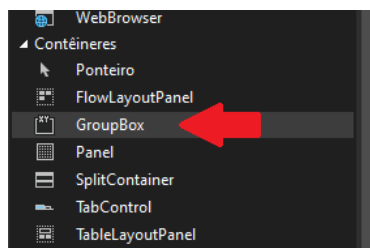
- Label 1
 - Text: Número 1
- Label 2
 - Text: Número 2
- Label 3
 - Text: Expressão
- Label 4
 - Text: Resultado
- TextBox1
 - Name: txtNumero1
- TextBox2
 - Name: txtNumero2
- TextBox3
 - Name: txtExpressao
 - Enable: False
- TextBox4
 - Name: txtResultado
 - Enable: False
- Button1
 - Name: btnCalcular
 - Text: Calcular
- Button2
 - Name: btnLimpar
 - Text: Limpar
- Button3
 - Name: btnSair
 - Text: Sair

➤ Organize os componentes inseridos acima da seguinte forma:

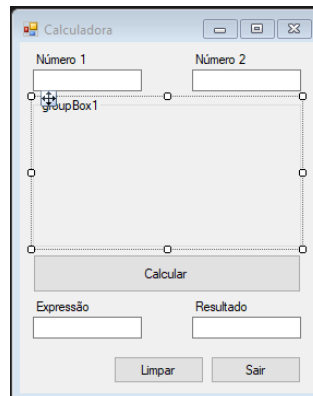
Icaro Daflon C# MOD I



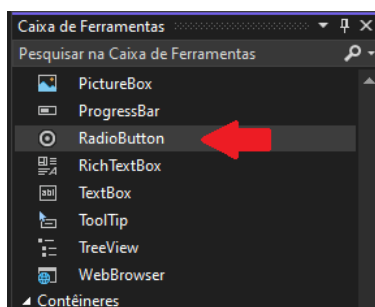
- Agora, iremos adicionar alguns controles ainda não vistos.
- Na caixa de ferramentas, dentro da categoria “Contêineres”, encontre controle “GroupBox” e insira-o no formulário. Este controle serve como um container de outros controles, formando um grupo de controles.



- Deixe o GroupBox da seguinte maneira:



- Com o controle GroupBox selecionado, altere a propriedade Text dele para “Operação”.
- Agora, dentro do controle GroupBox, iremos inserir novos controles chamados RadioButton, que é muito utilizado quando o formulário possui opções de escolha.
- Insira 5 RadioButton's, dentro da GroupBox, pois cada um deles representará uma operação da calculadora.



Icaro Daflon
C# MOD I

- Configure os controles RadioButtons da seguinte maneira:

Propriedade Name	Propriedade Text
optAdicao	Adição (+)
optSubtracao	Subtração (-)
optMultiplicacao	Multiplicação (*)
optDivisao	Divisão (/)
optRestoDivisao	Resto da Divisao (%)

- O formulário deve estar da seguinte maneira:

- Com a parte gráfica pronta, agora vamos criar as ações.
- Dê um duplo clique no botão Calcular e no código do event click digite o código abaixo:

```
double resultado = 0;

if (optAdicao.Checked)
{
    txtExpressao.Text = txtNumero1.Text + " + " + txtNumero2.Text;
    resultado = double.Parse(txtNumero1.Text) + double.Parse(txtNumero2.Text);
}
else if (optSubtracao.Checked)
{
    txtExpressao.Text = txtNumero1.Text + " - " + txtNumero2.Text;
    resultado = double.Parse(txtNumero1.Text) - double.Parse(txtNumero2.Text);
}
else if (optMultiplicacao.Checked)
{
    txtExpressao.Text = txtNumero1.Text + " * " + txtNumero2.Text;
    resultado = double.Parse(txtNumero1.Text) * double.Parse(txtNumero2.Text);
}
else if (optDivisao.Checked)
{
    txtExpressao.Text = txtNumero1.Text + " / " + txtNumero2.Text;
    resultado = double.Parse(txtNumero1.Text) / double.Parse(txtNumero2.Text);
}
else if (optRestoDivisao.Checked)
{
    txtExpressao.Text = txtNumero1.Text + " % " + txtNumero2.Text;
    resultado = double.Parse(txtNumero1.Text) % double.Parse(txtNumero2.Text);
}
txtResultado.Text = resultado.ToString();
```

Icaro Daflon
C# MOD I

- Veja o significado de cada linha do código do botão Calcular:

```
double resultado = 0;
```

- Cria uma variável resultado do tipo double, já atribuindo-lhe o valor 0

```
if (optAdicao.Checked)
```

- Verifica se o optionButton de adição foi selecionado pelo usuário

```
txtExpressao.Text = txtNumero1.Text + " + " + txtNumero2.Text;
```

- Atribui ao controle txtExpressao os valores dos controles txtNumero1 e txtNumero2, concatenando com o sinal de + (mais)

```
resultado = double.Parse(txtNumero1.Text) + double.Parse(txtNumero2.Text);
```

- Atribui à variável resultado da soma dos valores dos controles txtNumero1 e txtNumero2. Detalhe: antes de somar, os valores dos controles são convertidos para double usando o método double.Parse(string);

```
else if (optSubtracao.Checked)
```

- Verifica se o optionButton de subtração foi selecionado pelo usuário

```
txtExpressao.Text = txtNumero1.Text + " - " + txtNumero2.Text;
```

- Atribui ao controle txtExpressao os valores dos controles txtNumero1 e txtNumero2, concatenando com o sinal de - (menos)

```
resultado = double.Parse(txtNumero1.Text) - double.Parse(txtNumero2.Text);
```

- Atribui à variável resultado da subtração dos valores dos controles txtNumero1 e txtNumero2. Detalhe: antes de somar, os valores dos controles são convertidos para double usando o método double.Parse(string);

```
else if (optMultiplicacao.Checked)
```

- Verifica se o optionButton de multiplicação foi selecionado pelo usuário

```
txtExpressao.Text = txtNumero1.Text + " * " + txtNumero2.Text;
```

- Atribui ao controle txtExpressao os valores dos controles txtNumero1 e txtNumero2, concatenando com o sinal de * (multiplicação)

```
resultado = double.Parse(txtNumero1.Text) * double.Parse(txtNumero2.Text);
```

- Atribui à variável resultado da multiplicação dos valores dos controles txtNumero1 e txtNumero2. Detalhe: antes de somar, os valores dos controles são convertidos para double usando o método double.Parse(string);

```
else if (optDivisao.Checked)
```

- Verifica se o optionButton de divisão foi selecionado pelo usuário

```
txtExpressao.Text = txtNumero1.Text + " / " + txtNumero2.Text;
```

- Atribui ao controle txtExpressao os valores dos controles txtNumero1 e txtNumero2, concatenando com o sinal de / (divisão)

```
resultado = double.Parse(txtNumero1.Text) / double.Parse(txtNumero2.Text);
```


- Atribui à variável resultado da divisão dos valores dos controles txtNumero1 e txtNumero2. Detalhe: antes de somar, os valores dos controles são convertidos para double usando o método double.Parse(string);

```
else if (optRestoDivisao.Checked)
```

- Verifica se o optionButton de Resto da Divisão foi selecionado pelo usuário

```
txtExpressao.Text = txtNumero1.Text + " % " + txtNumero2.Text;
```

- Atribui ao controle txtExpressao os valores dos controles txtNumero1 e txtNumero2, concatenando com o sinal de % (mod)

```
resultado = double.Parse(txtNumero1.Text) % double.Parse(txtNumero2.Text);
```

- Atribui à variável resultado do resto da divisão dos valores dos controles txtNumero1 e txtNumero2. Detalhe: antes de somar, os valores dos controles são convertidos para double usando o método double.Parse(string);

```
txtResultado.Text = resultado.ToString();
```

- Atribui o valor da variável resultado ao controle txtResultado. Detalhe: antes de atribuir, o valor da variável é convertido para string usando o método objeto.ToString();

➤ Volte ao design do formulário e dê um duplo clique no botão Limpar

➤ Adicione o seguinte código ao Event Click:

```
txtExpressao.Text = null;  
txtResultado.Text = null;  
txtNumero1.Text = null;  
txtNumero2.Text = null;  
txtNumero1.Focus();  
optAdicao.Checked = false;  
optSubtracao.Checked = false;  
optMultiplicacao.Checked = false;  
optDivisao.Checked = false;  
optRestoDivisao.Checked = false;
```

➤ Volte ao design do formulário, e agora dê um duplo clique no botão Sair.

➤ Adicione o seguinte código ao Event Click:

```
this.Close();
```

➤ Altere a classe Program.cs para abrir o formulário frmCalculadora. A linha deve ficar da seguinte maneira

```
Application.Run(new frmCalculadora());
```

➤ Salve a solução e execute o programa

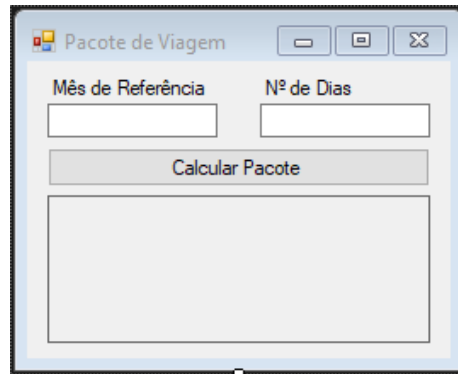
➤ Teste as funcionalidade e resultados ao finalizar, feche o Visual Studio.

AULA 07

Formulário Pacote de Viagem, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_07 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmPacoteDeViagem e altere a propriedade Text do mesmo para Pacote de Viagem.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Propriedade Text: Mês de Referência
 - Label2
 - Propriedade Text: N° de Dias
 - Label3
 - Propriedade Text: *Apague o conteúdo*
 - Propriedade (Name): lblMostrar
 - Propriedade AutoSize: False
 - Propriedade BorderStyle: FixedSingle
 - TextBox1
 - Propriedade (Name): txtMes
 - TextBox2
 - Propriedade (Name): txtDias
 - Button
 - Propriedade Text: Calcular Pacote
 - Propriedade (Name): btnCalcular
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

Icaro Daflon
C# MOD I



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Calcular Pacote” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
int mes = int.Parse(txtMes.Text);
int dias = int.Parse(txtDias.Text);

switch (mes)
{
    case 1:
        lblMostrar.Text = "Em Janeiro o valor de sua estadia é de " +
            (dias * 500) + "reais.";
        break;
    case 2:
        lblMostrar.Text = "Em Fevereiro o valor de sua estadia é de " +
            (dias * 500) + "reais.";
        break;
    case 3:
        lblMostrar.Text = "Em Março o valor de sua estadia é de " +
            (dias * 500) + "reais.";
        break;
    case 4:
        lblMostrar.Text = "Em Abril o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 5:
        lblMostrar.Text = "Em Maio o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 6:
        lblMostrar.Text = "Em Junho o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 7:
        lblMostrar.Text = "Em Julho o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 8:
        lblMostrar.Text = "Em Agosto o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 9:
        lblMostrar.Text = "Em Setembro o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 10:
        lblMostrar.Text = "Em Outubro o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 11:
        lblMostrar.Text = "Em Novembro o valor de sua estadia é de " +
            (dias * 300) + "reais.";
        break;
    case 12:
        lblMostrar.Text = "Em Dezembro o valor de sua estadia é de " +
            (dias * 500) + "reais.";
        break;
    default:
        lblMostrar.Text = "O número referente ao Mês não existe!";
        break;
}
```

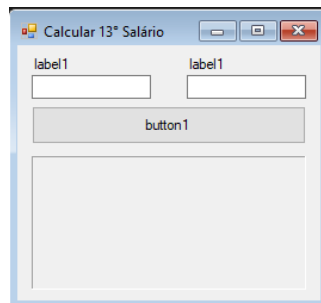
- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Utilizando Instruções Switch para Tomada de Decisões, Páginas 89 a 96

- Peça a seu instrutor a pasta “Aula 06”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 06”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, insira um novo formulário com o nome de frmCalcularI3Salario.
- Altere a propriedade Text do formulário para Calcular I3º Salário.

Insira os seguintes controles ao formulário e configure-os da seguinte maneira:

- Label 1
 - Text: Salário
 - Label 2
 - Text: Mês Referencial
 - Label3
 - Propriedade Text: *Apague o conteúdo*
 - Propriedade (Name): lblMostrar
 - Propriedade AutoSize: False
 - Propriedade BorderStyle: Fixed3D
 - TextBox1
 - Name: txtSalario
 - TextBox2
 - Name: txtMes
 - Button1
 - Name: btnCalcular
 - Text: Calcular
- Organize os componentes inseridos acima da seguinte forma:



- Com a parte gráfica pronta, agora vamos criar as ações.
- Dê um duplo clique no botão Calcular e no código do event click digite o código abaixo:

Icaro Daflon

C# MOD I

```
double valorMensal, resultado = 0;
string mesSelecioneado = string.Empty;

valorMensal = double.Parse(txtSalario.Text) / 12;

switch (int.Parse(txtMes.Text))
{
    case 1:
        mesSelecioneado = "Janeiro";
        resultado = valorMensal * 1;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 2:
        mesSelecioneado = "Fevereiro";
        resultado = valorMensal * 2;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 3:
        mesSelecioneado = "Março";
        resultado = valorMensal * 3;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 4:
        mesSelecioneado = "Abril";
        resultado = valorMensal * 4;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 5:
        mesSelecioneado = "Maio";
        resultado = valorMensal * 5;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 6:
        mesSelecioneado = "Junho";
        resultado = valorMensal * 6;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 7:
        mesSelecioneado = "Julho";
        resultado = valorMensal * 7;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 8:
        mesSelecioneado = "Agosto";
        resultado = valorMensal * 8;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 9:
        mesSelecioneado = "Setembro";
        resultado = valorMensal * 9;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 10:
        mesSelecioneado = "Outubro";
        resultado = valorMensal * 10;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 11:
        mesSelecioneado = "Novembro";
        resultado = valorMensal * 11;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    case 12:
        mesSelecioneado = "Dezembro";
        resultado = valorMensal * 12;
        lblMostrar.Text = "O valor do 13º Salário no mês de " +
            mesSelecioneado + " é igual a " + resultado + " reais";
        break;
    default:
        lblMostrar.Text = "Mês inválido!";
        break;
}
```

➤ Explicação do código:

```
double valorMensal, resultado = 0;
```

- Cria 2 variáveis valorMensal e resultado, do tipo double, atribuindo a variável resultado o valor 0

Obs.: Veja que é possível criar diversas variáveis, que sejam do mesmo tipo, em uma única linha de código, informando o tipo e o nome das variáveis separadas por vírgula.

```
string mesSelecioneado = string.Empty;
```

- Cria uma variável mesSelecioneado do tipo string para receber o mês selecionado pelo usuário. Atribui um valor vazio ("") para a variável. Valor este diferente de nulo.

```
valorMensal = double.Parse(txtSalario.Text) / 12;
```

- Atribui a variável valorMensal, o valor digitado no controle salário, convertendo-o para double e dividindo-o por 12.

```
switch (int.Parse(txtMes.Text))
```

- Verifica qual o mês selecionado. Converte o valor para int, pois serão utilizados, nos rótulos das cases, apenas valores inteiros entre 1 e 12

```
case 1:
```

- Identifica que o valor digitado foi o valor 1.

```
mesSelecioneado = "Janeiro";
```

- Atribui à variável mesSelecioneado, o valor "Janeiro", informando que o mês que o usuário selecionou foi Janeiro;

```
resultado = valorMensal * 1;
```

- Atribui à variável resultado, o valor mensal do 13º multiplicado pelo número do mês, neste caso 1.

```
lblMostrar.Text = "O valor do 13º Salário no mês de " +  
mesSelecioneado + " é igual a " + resultado + " reais";
```

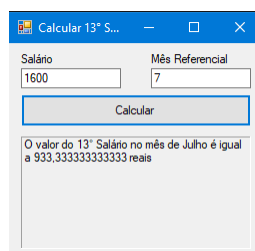
- Atribui ao controle lblMostrar o texto "O valor do 13º salário no mês de", concatenando com o mês selecionado pelo usuário, o texto "é igual a", concatenando com o valor do 13º salário calculado e o texto "reais".

```
break;
```

- Finaliza a execução do case.
- Os próximos Cases, a ideia será a mesma, não necessitando assim, de uma explicação mais detalhada sobre cada linha.
- Altere a classe Program.cs para que o formulário a ser aberto seja o frmCalcular13Salario. A linha onde se encontra o código responsável deve ficar da seguinte maneira:

```
Application.Run(new frmCalcular13Salario());
```

- Salve a solução e execute o programa



- Teste as funcionalidade e resultados ao finalizar, feche o Visual Studio

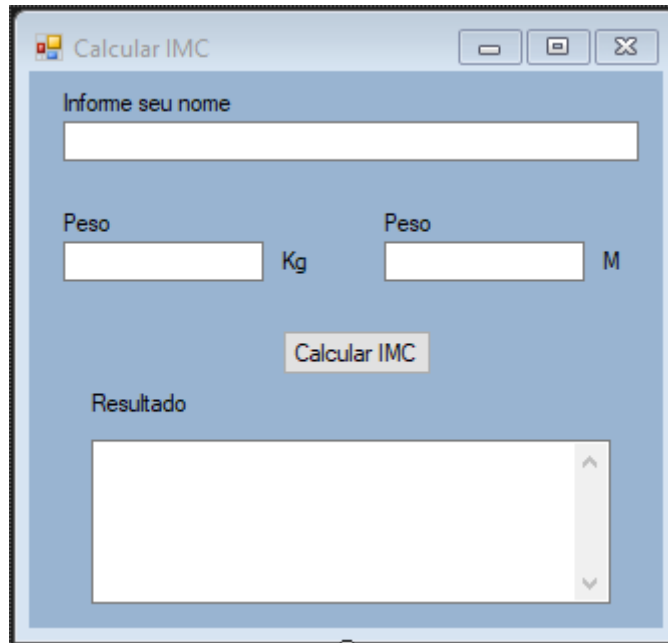
AULA 08

Revisão das aulas anteriores, Páginas 98 a 100

Passo a passo para fins de gabarito, pois deve ser feito inteiramente pelo aluno

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_08 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCalcularIMC e altere a propriedade Text do mesmo para Pacote de Viagem.
- Altere a propriedade BackColor do Form para “ActiveCaption”
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Text: Informe seu nome
 - Label2
 - Text: Peso
 - Label3
 - Text: Altura
 - Label4
 - Text: Kg
 - Label5
 - Text: M
 - Label6
 - Text: Resultado
 - TextBox1
 - Name: txtNome
 - TextBox2
 - Name: txtPeso
 - TextBox3
 - Name: txtAltura
 - TextBox4
 - Name: txtResultado
 - Multiline: True

- Scrollbars: Vertical
 - Button
 - Propriedade Text: Calcular IMC
 - Propriedade (Name): btnCalcularIMC
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Calcular Pacote” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
string nome = txtNome.Text;
double peso = double.Parse(txtPeso.Text);
double altura = double.Parse(txtAltura.Text);

double IMC = peso / altura;

if (IMC <= 18)
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nVocê está abaixo do peso ideal";
else if (IMC <= 24)
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nParabéns! você está em seu peso normal";
else if (IMC <= 29)
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nVocê está acima de seu peso (sobrepeso)";
else if (IMC <= 34)
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nObesidade grau I";
else if (IMC <= 39)
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nObesidade grau II";
else
    txtResultado.Text = nome + "seu imc é:\n" +
        IMC + "\nObesidade grau III";
```

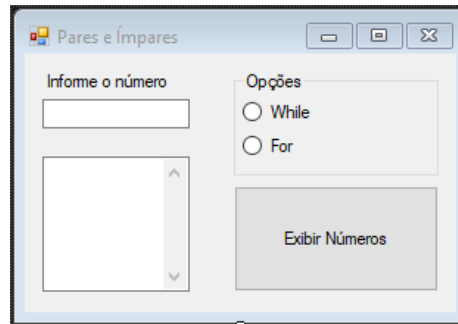
- O aluno também é livre para alterar fontes, e cores dos controles bem como os formatos desde que o código funcione desta mesma forma

AULA 09

Formulário Pares e Ímpares, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_09 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmParesImpares e altere a propriedade Text do mesmo para Pares e Ímpares.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Text: Informe o número
 - TextBox1
 - Name: txtNumero
 - TextBox2
 - Name: txtResultado
 - Scrollbars: Vertical
 - Multiline: True
 - Button
 - Text: Exibir Números
 - Name: btnExibir
 - GroupBox
 - Name: groupBoxEstrutura
 - Text: Opções
 - RadioButton1
 - Text: While
 - Name: optWhile
 - RadioButton2
 - Text: For
 - Name: optFor
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

Icaro Daflon C# MOD I



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Exibir Números” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
int numero = int.Parse(txtNumero.Text);
int contador = 0;

if(optWhile.Checked)
{
    txtResultado.Text = "Números Ímpares:";
    while (contador <= numero)
    {
        if (contador % 2 == 1)
            txtResultado.Text += "\r\n" + contador.ToString();
        contador++;
    }
}
else if (optFor.Checked)
{
    txtResultado.Text = "Números Pares:";
    for (contador = 0; contador <= numero; contador++)
    {
        if (contador % 2 == 0)
            txtResultado.Text += "\r\n" + contador;
    }
}
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Estrutura de Repetição (While), Páginas 105 a 110

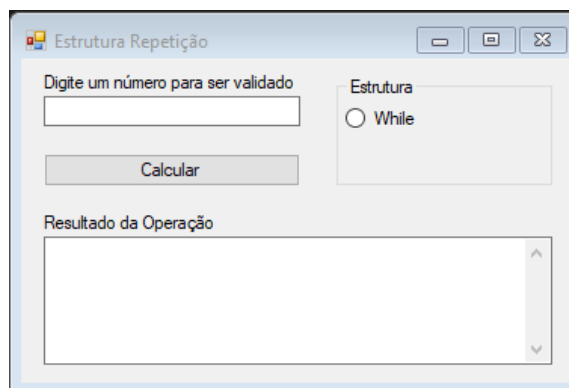
- Peça a seu instrutor a pasta “Aula 07”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 07”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, insira um novo formulário com o nome de frmEstruturaRepeticao.
- Altere a propriedade Text do formulário para Estrutura Repetição.

Insira os seguintes controles ao formulário e configure-os da seguinte maneira:

- Label 1
 - Text: Digite um número para ser validade
- Label 2
 - Text: Resultado da Operação
- TextBox1
 - Name: txtNumero

- TextBox2
 - Name: txtResultado
 - ScrollBars: Vertical
 - Multiline: True
- Button1
 - Name: btnCalcular
 - Text: Calcular
- GroupBox
 - Name: groupBoxEstrutura
 - Text: Opções
- RadioButton1
 - Text: While
 - Name: optWhile

➤ Organize os componentes inseridos acima da seguinte forma:



- Com a parte gráfica pronta, agora vamos criar as ações.
- Dê um duplo clique no botão Calcular e no código do event click digite o código abaixo:

```
long numero = long.Parse(txtNumero.Text);
int contaDivisoes = 0;

if (optWhile.Checked)
{
    txtResultado.Text = "While:\r\n";

    long contador = 1;

    while (contador <= numero / 2)
    {
        txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";

        if (numero % contador == 0) {
            txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
            contaDivisoes++;
        }

        contador++;
    }

    txtResultado.Text += "Divisível por " + numero + "\r\n";

    if (contaDivisoes > 1)
        txtResultado.Text += "\r\n0 número não é primo!";
    else
        txtResultado.Text += "\r\n0 número é primo, " +
            "pois é divisível apenas por 1 e por ele mesmo!";
}
```

Icaro Daflon
C# MOD I

➤ Explicação do código:

```
long numero = long.Parse(txtNumero.Text);
```

- Cria a variável número, do tipo long, já atribuindo o valor que o usuário informou no controle txtNumero

```
int contaDivisoes = 0;
```

- Cria a variável contaDivisoes, do tipo int, já atribuindo-lhe o valor 0 (zero).

```
if (optWhile.Checked)
```

- Verifica se o valor option optWhile foi selecionado pelo usuário.

```
txtResultado.Text = "While:\r\n";
```

- Atribui, ao controle txtResultado, o texto “While:” e um comando de quebra de linha

```
long contador = 1;
```

- Cria a variável contador, do tipo long, já atribuindo-lhe o valor 1.

```
while (contador <= numero / 2)
```

- Define o critério de parada do laço. Nesse caso o critério de parada é: Enquanto contador for menor ou igual a metade do valor numero que foi digitado pelo usuário no controle txtNumero.

```
txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";
```

- Atribui ao controle txtResultado o texto “Realizar a divisão por ”, concatenando com o valor da variável que é utilizada no critério de parada do loop. (Note que esta atribuição é feita utilizando o sinal de + [mais] antes do sinal de igual, isso se dá para que atribuição feita concatene com o valor atual do txtResultado, atualizando o texto sem apagar o conteúdo anterior.)

```
if (numero % contador == 0)
```

- Verifica se o número digitado pelo usuário é divisível pelo contador, que é usado como critério de parada.

```
txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “ → Divisível por ” concatenando ao valor da variável contador e adiciona o comando de quebra de linha

```
contaDivisoes++;
```

- Incrementa I ao contador de divisões exatas.

```
contador++;
```

- Incrementa I ao contador, que servirá de critério de parada para o loop’

```
txtResultado.Text += "Divisível por " + numero + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “Divisível por ” mais o número digitado pelo usuário e adiciona uma quebra de linha

```
if (contaDivisoes > 1)
```

- Verifica se o valor da variável contaDivisões é maior que 1, ou seja, foi realizada mais de uma divisão exata com o número digitado pelo usuário.

Icaro Daflon
C# MOD I

```
txtResultado.Text += "\r\nO número não é primo!";
```

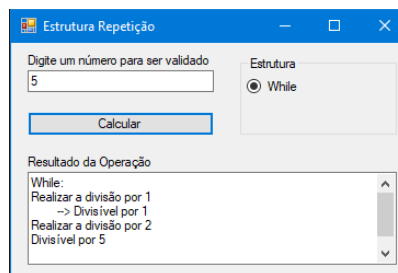
- Concatena ao valor do controle txtResultado o texto “O número não é primo!”, informando que o número que o usuário informou não é primo

```
txtResultado.Text += "\r\nO número é primo, " +  
"pois é divisível apenas por 1 e por ele mesmo!";
```

- Concatena ao valor do controle txtResultado o texto "**\r\nO número é primo, pois é divisível apenas por 1 e por ele mesmo!**", informando que o número que o usuário digitou é primo.
- Altere a classe Program.cs para que o formulário a ser aberto seja o frmEstruturaRepeticao. A linha onde se encontra o código responsável deve ficar da seguinte maneira:

```
Application.Run(new frmEstruturaRepeticao());
```

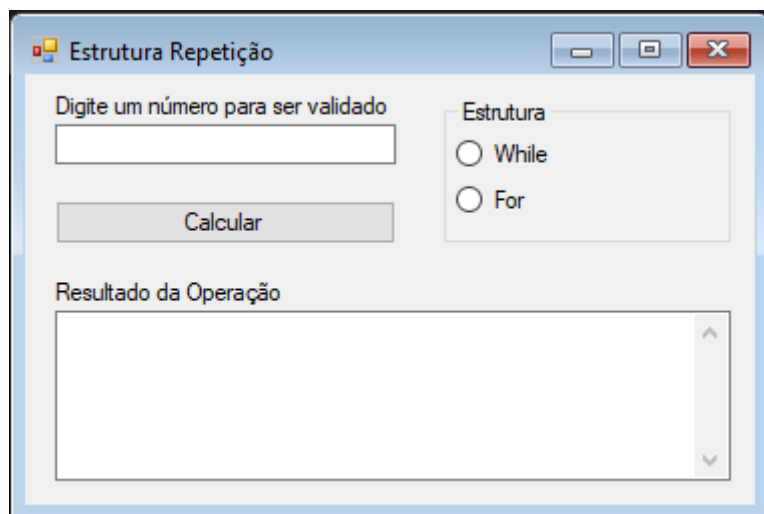
- Salve a solução e execute o programa



- Teste as funcionalidade e resultados ao finalizar, feche o Programa.

Estrutura de Repetição (For), Páginas 114 a 118

- Com o formulário frmEstruturaRepeticao.cs já aberto, insira o controle abaixo:
 - RadioButton
 - Name: optFor
 - Text: For
- Posicione o RadioButton da seguinte forma:



- Agora, altere o código do formulário. Para isso dê um duplo clique no botão Calcular e escreva, logo abaixo da chave que fecha o comando if (optWhile.Checked), escreva o código abaixo:

```
else if (optFor.Checked)
{
    txtResultado.Text = "For: \r\n";

    for (long contador = 1; contador <= numero / 2; contador++)
    {
        txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";

        if (numero % contador == 0)
        {
            txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
            contaDivisoes++;
        }
    }

    txtResultado.Text += "Divisível por " + numero + "\r\n";

    if (contaDivisoes > 1)
        txtResultado.Text += "\r\n0 número não é primo!";
    else
        txtResultado.Text += "\r\n0 número é primo, " +
            "pois é divisível apenas por 1 e por ele mesmo!";
}
```

- Explicação do código:

```
else if (optFor.Checked)
```

- Verifica se o valor option optFor foi selecionado pelo usuário.

```
txtResultado.Text = "For: \r\n";
```

- Atribui, ao controle txtResultado, o texto “For:” e um comando de quebra de linha

```
for (long contador = 1; contador <= numero / 2; contador++)
```

- Inicia a variável contador, que será usada no critério de parada do laço, com o valor 1. Define o critério de parada do laço, neste caso: Para contador menor ou igual a metade do valor numero e por fim incrementa 1 a variável contador.

```
txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";
```

- Atribui ao controle txtResultado o texto “Realizar a divisão por ”, concatenando com o valor da variável que é utilizada no critério de parada do loop. (Note que esta atribuição é feita utilizando o sinal de + [mais] antes do sinal de igual, isso se dá para que atribuição feita concatene com o valor atual do txtResultado, atualizando o texto sem apagar o conteúdo anterior.)

```
if (numero % contador == 0)
```

- Verifica se o número digitado pelo usuário é divisível pelo contador, que é usado como critério de parada.

```
txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “ → Divisível por ” concatenando ao valor da variável contador e adiciona o comando de quebra de linha

```
contaDivisoes++;
```

- Incrementa 1 ao contador de divisões exatas.

```
txtResultado.Text += "Divisível por " + numero + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “Divisível por ” mais o número digitado pelo usuário e adiciona uma quebra de linha

```
if (contaDivisoes > 1)
```

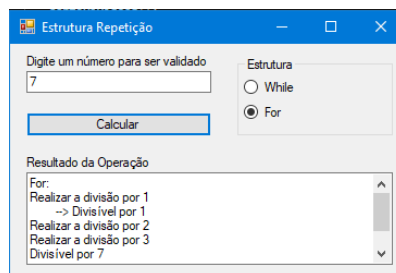
- Verifica se o valor da variável contaDivisões é maior que 1, ou seja, foi realizada mais de uma divisão exata com o número digitado pelo usuário.

```
txtResultado.Text += "\r\n0 número não é primo!";
```

- Concatena ao valor do controle txtResultado o texto “O número não é primo!”, informando que o número que o usuário informou não é primo

```
txtResultado.Text += "\r\n0 número é primo, " +  
"pois é divisível apenas por 1 e por ele mesmo!";
```

- Concatena ao valor do controle txtResultado o texto "**\r\n0 número é primo, pois é divisível apenas por 1 e por ele mesmo!**", informando que o número que o usuário digitou é primo.
- Salve a solução e execute o programa



- Teste as funcionalidade e resultados ao finalizar, feche o Visual Studio.

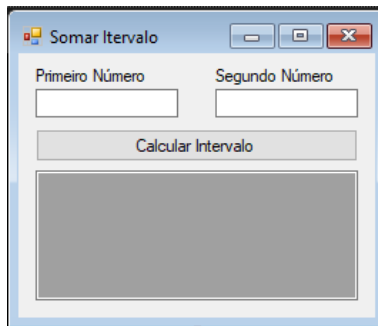
AULA 10

Formulário Somar Intervalo, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_10 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmSomarIntervalo e altere a propriedade Text do mesmo para Somar Intervalo.
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
- Label1
 - Text: Primeiro Número
 - Label2
 - Text: Segundo Número
 - TextBox1
 - Name: txtPrimeiroNumero

- TextBox2
 - Name: txtSegundoNumero
- TextBox3
 - Name: txtResultado
 - BackColor: ControlDark
 - Multiline: True
- Button
 - Text: Calcular Intervalo
 - Name: btnCalcular

- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Calcular Intervalo” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
long numero1 = int.Parse(txtPrimeiroNumero.Text);
long numero2 = int.Parse(txtSegundoNumero.Text);
string resultado = string.Empty;
long soma = 0;

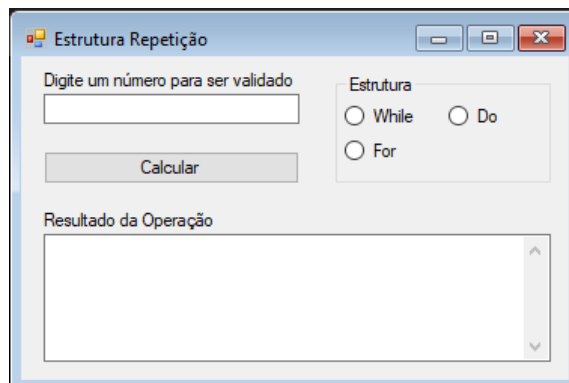
do
{
    soma += numero1;
    resultado += numero1 + " ";
    numero1++;
} while (numero1 <= numero2);

resultado += "= " + soma;
txtResultado.Text = resultado;
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Estrutura de Repetição (Do), Páginas 122 a 128

- Peça a seu instrutor a pasta “Aula 09”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 09”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, abra o formulário com o nome de frmEstruturaRepeticao.cs.
- Com o formulário frmEstruturaRepeticao.cs aberto, insira o controle abaixo:
 - RadioButton
 - Name: optDo
 - Text: Do
- Organize os componentes inseridos acima da seguinte forma:



- Com a parte gráfica pronta, agora vamos criar as ações.
- Dê um duplo clique no botão Calcular e, ao final das chaves que fecham a instrução else if (optFor.Checked), digite o código abaixo:

```
else if (optDo.Checked)
{
    txtResultado.Text = "Do:\r\n";

    long contador = 1;

    do
    {
        txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";

        if (numero % contador == 0)
        {
            txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
            contaDivisoes++;
        }
        contador++;
    } while (contador <= numero / 2);

    txtResultado.Text += "Divisível por " + numero + "\r\n";

    if (contaDivisoes > 1)
        txtResultado.Text += "\r\n0 número não é primo!";
    else
        txtResultado.Text += "\r\n0 número é primo, " +
            "pois é divisível apenas por 1 e por ele mesmo!";
}
```

- Explicação do código:

```
else if (optDo.Checked)
```

- Verifica se o valor option optDo foi selecionado pelo usuário.

```
txtResultado.Text = "Do:\r\n";
```

- Atribui, ao controle txtResultado, o texto “Do:” e um comando de quebra de linha

```
long contador = 1;
```

- Cria a variável contador, do tipo long, já atribuindo-lhe o valor 1.

```
do
```

- Indica o início do laço

```
txtResultado.Text += "Realizar a divisão por " + contador + "\r\n";
```

- Atribui ao controle txtResultado o texto “Realizar a divisão por ”, concatenando com o valor da variável que é utilizada no critério de parada do loop. (Note que esta atribuição é feita utilizando o sinal de + [mais] antes do sinal de igual, isso se dá para que atribuição feita concatene com o valor atual do txtResultado, atualizando o texto sem apagar o conteúdo anterior.)

```
if (numero % contador == 0)
```

- Verifica se o número digitado pelo usuário é divisível pelo contador, que é usado como critério de parada.

```
txtResultado.Text += "    --> Divisível por " + contador + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “ → Divisível por “ concatenando ao valor da variável contador e adiciona o comando de quebra de linha

```
contaDivisoes++;
```

- Incrementa 1 ao contador de divisões exatas.

```
contador++;
```

- Incrementa 1 ao contador, que servirá de critério de parada para o loop

```
} while (contador <= numero / 2) ;
```

- Define o critério de parada do laço. Nesse caso o critério de parada é: Enquanto contador for menor ou igual a metade do valor numero que foi digitado pelo usuário no controle txtNumero.

```
txtResultado.Text += "Divisível por " + numero + "\r\n";
```

- Concatena ao valor do controle txtResultado o texto “Divisível por ” mais o número digitado pelo usuário e adiciona uma quebra de linha

```
if (contaDivisoes > 1)
```

- Verifica se o valor da variável contaDivisões é maior que 1, ou seja, foi realizada mais de uma divisão exata com o número digitado pelo usuário.

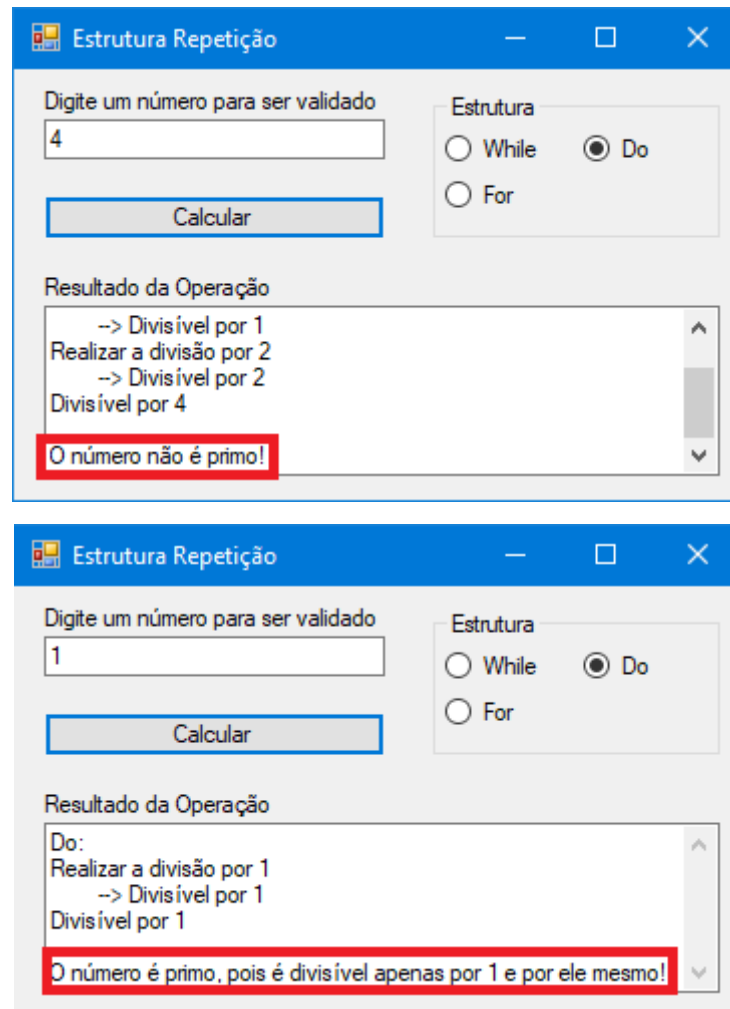
```
txtResultado.Text += "\r\n0 número não é primo!";
```

- Concatena ao valor do controle txtResultado o texto “O número não é primo!”, informando que o número que o usuário informou não é primo

```
txtResultado.Text += "\r\n0 número é primo, " +  
    "pois é divisível apenas por 1 e por ele mesmo!";
```

- Concatena ao valor do controle txtResultado o texto “\r\n0 número é primo, pois é divisível apenas por 1 e por ele mesmo!”, informando que o número que o usuário digitou é primo.

- Salve a solução e execute o programa



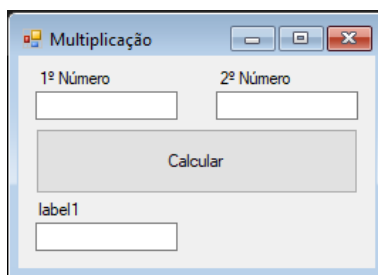
- Teste as funcionalidade e resultados ao finalizar, feche o Visual Studio.

AULA 11

Formulário Somar Intervalo, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto "Aplicativo do Windows Forms (.NET Framework)" de nome Aula_10 e estrutura .NET Framework 4.8
- Renomeie o formulário "Form1" para frmSomarIntervalo e altere a propriedade Text do mesmo para Somar Intervalo.
- Na janela que irá aparecer clique em "Sim".
- Insira os seguintes controles:
 - Label1
 - Text: 1º Número
 - Label2
 - Text: 2º Número

- Label3
 - Text: Resultado
 - TextBox1
 - Name: txtNumero1
 - TextBox2
 - Name: txtNumero2
 - TextBox3
 - Name: txtResultado
 - Button
 - Text: Calcular
 - Name: btnCalcular
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Calcular” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:

```
try
{
    int numero1 = int.Parse(txtNumero1.Text);
    int numero2 = int.Parse(txtNumero2.Text);
    int resultado = numero1 * numero2;

    txtResultado.Text = resultado.ToString();
}
catch(Exception ex)
{
    MessageBox.Show("Preencha as caixas corretamente!");
}
```

- A aula depois troca a última linha `MessageBox.Show` (“Preencha as caixas corretamente!”); para o seguinte:

```
MessageBox.Show(ex.Message);
```

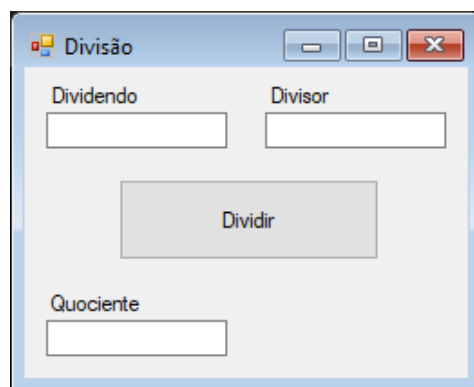
- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Tratamento de Exceções, Páginas 133 a 139

- Peça a seu instrutor a pasta “Aula 10”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 10”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, insira um novo formulário com o nome de frmDivisao.
- Altere a propriedade Text do formulário para Divisão.

Insira os seguintes controles ao formulário e configure-os da seguinte maneira:

- Label 1
 - Text: Dividendo
 - Label 2
 - Text: Divisor
 - Label 3
 - Text: Quociente
 - TextBox1
 - Name: txtDividendo
 - TextBox2
 - Name: txtDivisor
 - TextBox3
 - Name: txtQuociente
 - Button1
 - Name: btnDividir
 - Text: Dividir
- Organize os componentes inseridos acima da seguinte forma:



- Com a parte gráfica pronta, agora vamos criar as ações.
- Dê um duplo clique no botão Calcular e no código do event click digite o código abaixo:

Icaro Daflon
C# MOD I

```
int dividendo = int.Parse(txtDividendo.Text);  
int divisor = int.Parse(txtDivisor.Text);  
  
int quociente = dividendo / divisor;  
  
txtQuociente.Text = quociente.ToString();
```

- Explicação do código:

```
int dividendo = int.Parse(txtDividendo.Text);
```

- Cria a variável dividendo, do tipo double, já atribuindo o valor que o usuário informou no controle txtDividendo

```
int divisor = int.Parse(txtDivisor.Text);
```

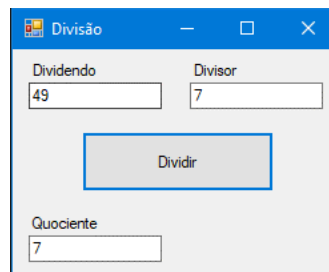
- Cria a variável divisor, do tipo double, já atribuindo o valor que o usuário informou no controle txtDivisor

```
int quociente = dividendo / divisor;
```

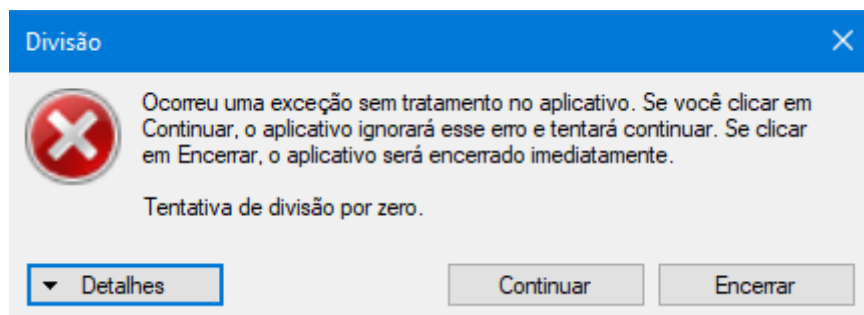
- Cria a variável quociente, do tipo double, já atribuindo o valor da divisão entre os valores das variáveis dividendo e divisor.

```
txtQuociente.Text = quociente.ToString();
```

- Atribui, ao controle txtResultado o valor da variável quociente convertendo-a para o tipo string.
- Altere a classe Program.cs para que o formulário a ser aberto seja o frmDivisao. Com ensinado anteriormente.
- Salve a solução e execute o programa, informe o dividendo e o divisor, e depois clique no botão Dividir



- Perceba que foi calculado certinho sem nenhum problema.
- Agora, experimente colocar no campo Divisor o valor 0 (Zero) e clique no botão Dividir



- Veja que surge uma janela informando que não existe divisão por zero, como você pode ver, o sistema aborta e gera o erro.
- Pare a execução do programa, clicando no botão "Encerrar"

Icaro Daflon
C# MOD I

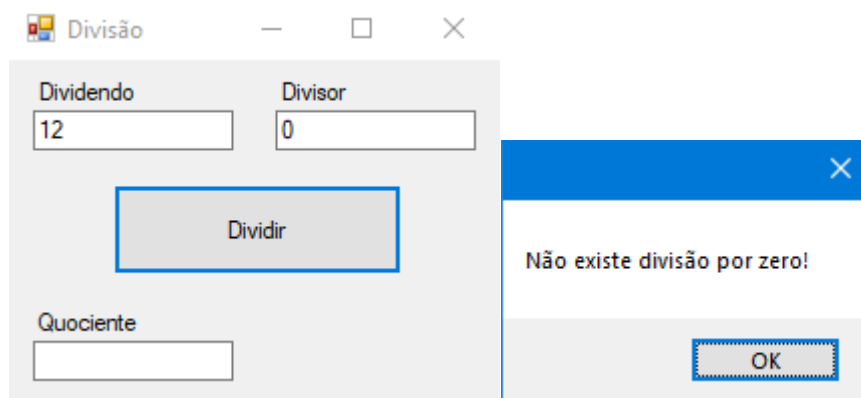
- Dê um duplo clique no botão Dividir e altere o código anterior. Note que são adicionadas as cláusulas try e catch da seguinte maneira:

```
try
{
    int dividendo = int.Parse(txtDividendo.Text);
    int divisor = int.Parse(txtDivisor.Text);

    int quociente = dividendo / divisor;

    txtQuociente.Text = quociente.ToString();
}
catch (DivideByZeroException)
{
    MessageBox.Show("Não existe divisão por zero!");
    txtDivisor.Focus();
}
```

- Perceba que todo o cálculo realizado na divisão foi colocado dentro da cláusula try. A cláusula catch recebe como parâmetro o nome do erro anterior, para que o sistema saiba qual erro deve ser tratado. Dentro dessa cláusula, é informado o que o sistema deve fazer, caso ocorra o erro. Neste caso, o sistema apenas irá abrir o controle MessageBox, informando ao usuário que ele não deve realizar a divisão por zero, e o foco do cursor enviado para o campo onde encontra o valor incorreto, no caso, o Divisor.
- Salve a solução e execute o programa, faça uma divisão por zero e veja o que acontece agora.

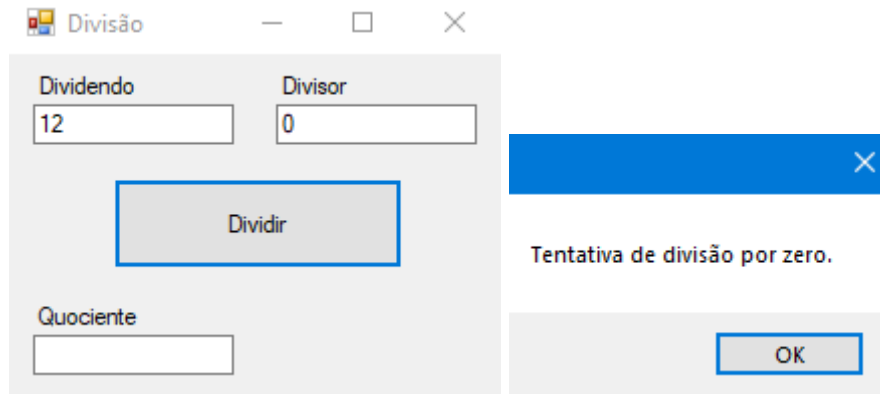


- Agora o sistema não aborta mais, apenas informa ao usuário de que não existe divisão por zero e dá-lhe chance de alterar o valor.
- Você pode também mostrar ao usuário o erro que o próprio Visual Studio mostraria.
- Altere a clausula catch para o seguinte:

```
catch (DivideByZeroException ex)
{
    MessageBox.Show(ex.Message);
    txtDivisor.Focus();
}
```

- Salve a solução e execute o programa, faça uma divisão por zero e veja o que acontece agora.

Icaro Daflon C# MOD I



- Observe que a mensagem muda, porém, é muito parecida com a que inserimos manualmente. Desse modo, você pode tratar qualquer exceção que possa ser provocada pelo uso incorreto do sistema.
- Teste as funcionalidade e resultados ao finalizar, feche o Programa.

Linhas de Comentários, Páginas 140 a 141

- Comente o código do formulário frmDividir e deixe-o da seguinte maneira:

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

/*
 * Programa para cálculo da divisão de valores.
 * O usuário deve informar um dividendo e um
 * divisor e clicar no botão Dividir.
 */

/*
 * Programa.....: Dividir
 * Autor(es).....: Icaro
 * Data.....: 20/11/2022
 * Versão.....: 1.0
 * Revisão.....: 27/11/2022
 */

namespace Curso_CSharp
{
    public partial class frmDivisao : Form
    {
        public frmDivisao()
        {
            InitializeComponent();
        }
    }
}
```

```
try
{
    // Cria as variáveis e atribui o valor dos controles a elas
    int dividendo = int.Parse(txtDividendo.Text);
    int divisor = int.Parse(txtDivisor.Text);

    // Realiza o cálculo da divisão
    int quociente = dividendo / divisor;

    //Mostra o resultado ao usuário
    txtQuociente.Text = quociente.ToString();
}
catch (DivideByZeroException ex)
{
    //Caso o usuário faça uma divisão por zero gera uma exceção
    MessageBox.Show(ex.Message);
    txtDivisor.Focus();
}
```


- Salve o seu projeto, e após, feche o Visual Studio

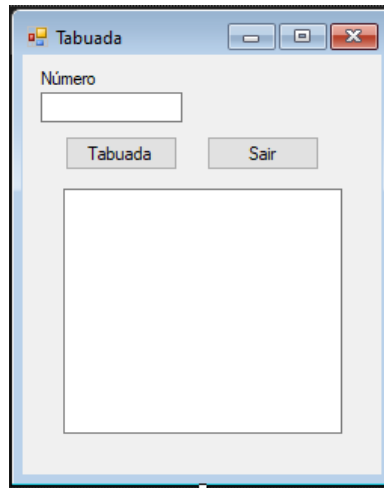
AULA 12

Revisão das aulas anteriores, Páginas 144 e 145

Passo a passo para fins de gabarito, pois deve ser feito inteiramente pelo aluno

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_12 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmTabuada e altere a propriedade Text do mesmo para Tabuada
- Na janela que irá aparecer clique em “Sim”.
- Insira os seguintes controles:
 - Label1
 - Text: Número
 - TextBox1
 - Name: txtNumero
 - TextBox2
 - Name: txtResultado
 - Multiline: True
 - Button1
 - Text: Tabuada
 - Name: btnTabuada
 - Button2
 - Text: Sair
 - Name: btnSair
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

Icaro Daflon C# MOD I



- Com todos os componentes dispostos, dê um duplo clique sobre o botão “Tabuada” para gerar o event click.
- Com o código aberto digite o seguinte código no método event click do botão:
 - Utilizando While:

```
try
{
    int numero = int.Parse(txtNumero.Text);
    int contador = 0;
    string resultado = string.Empty;

    while (contador <= 10)
    {
        resultado += contador + " X " + numero + " = " + (contador * numero) + "\r\n";
        contador++;
    }

    txtResultado.Text = resultado;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    txtNumero.Focus();
}
```

- Utilizando For:

```
try
{
    int numero = int.Parse(txtNumero.Text);
    string resultado = string.Empty;

    for (int contador = 0; contador <= numero; contador++)
    {
        resultado += contador + " X " + numero + " = " + (contador * numero) + "\r\n";
    }

    txtResultado.Text = resultado;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    txtNumero.Focus();
}
```

- Utilizando Do:

Icaro Daflon
C# MOD I

```
try
{
    int numero = int.Parse(txtNumero.Text);
    int contador = 0;
    string resultado = string.Empty;

    do
    {
        resultado += contador + " X " + numero + " = " + (contador * numero) + "\r\n";
        contador++;
    } while (contador <= 10);

    txtResultado.Text = resultado;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    txtNumero.Focus();
}
```

- Volte para a guia Design e dê um duplo clique sobre o botão Sair e escreva o código abaixo:

```
this.Close();
```

- O aluno também é livre para alterar fontes, e cores dos controles bem como os formatos desde que o código funcione desta mesma forma.

AULA 13

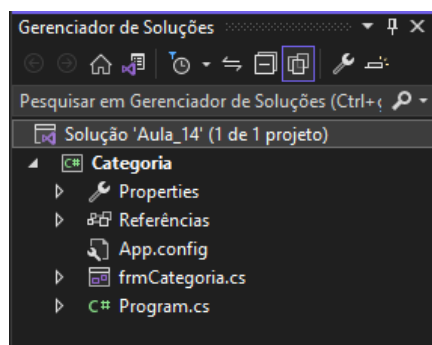
Conceitos Básicos de Programação Orientada a Objetos, Páginas 149 a 152

- Aula totalmente teórica e de acordo, portanto sem alterações.

AULA 14

Formulário Categoria Utilizando Métodos, Interatividade

- Copie a pasta de nome Aula_06 do seu local de gravação ou peça a seu instrutor a pasta e cole ela no seu local de gravação, a renomeando para Aula_14.
- Após renomear a pasta, abra ela e, em seguida, abra o arquivo Aula_06.sln
- Renomeie a Solução para Aula_14 ficando assim:



- Agora iremos modificar o código utilizando um método com parâmetros e retorno para podermos imprimir a mensagem na tela, para isso, dê um duplo clique sobre o botão Verificar Categoria na guia design do formulário e, com o código aberto crie o seguinte método dentro da classe:

```
private string VerificarCategoria (int idade)
{
    string resultado;
    if (idade <= 10)
        resultado = " você é da categoria Pré-Mirim!";
    else if (idade <= 15)
        resultado = " você é da categoria Mirim!";
    else if (idade <= 17)
        resultado = " você é da categoria Infantil!";
    else if (idade <= 21)
        resultado = " você é da categoria Juvenil!";
    else
        resultado = " você é da categoria Adulto!";

    return resultado;
}
```

- Em seguida, altere o código do event click para que fique da seguinte maneira:

```
private void btnVerificar_Click(object sender, EventArgs e)
{
    string nome = txtNome.Text;
    lblMostrar.Text = nome + VerificarCategoria(int.Parse(txtIdade.Text));
}
```

- O código completo da classe deve esta da seguinte forma:

```
public partial class frmCategoria : Form
{
    public frmCategoria()
    {
        InitializeComponent();
    }

    private string VerificarCategoria (int idade)
    {
        string resultado;
        if (idade <= 10)
            resultado = " você é da categoria Pré-Mirim!";
        else if (idade <= 15)
            resultado = " você é da categoria Mirim!";
        else if (idade <= 17)
            resultado = " você é da categoria Infantil!";
        else if (idade <= 21)
            resultado = " você é da categoria Juvenil!";
        else
            resultado = " você é da categoria Adulto!";

        return resultado;
    }

    private void btnVerificar_Click(object sender, EventArgs e)
    {
        string nome = txtNome.Text;
        lblMostrar.Text = nome + VerificarCategoria(int.Parse(txtIdade.Text));
    }
}
```

- Abra o formulário e teste as funções inseridas e ao final, salve o projeto e feche o Visual Studio.

Criando Métodos com C#, Páginas 159 a 162

- Peça a seu instrutor a pasta “Aula II”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula II”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, abra o código do formulário com o nome de frmCalculadora.cs.
- Crie o seguinte método dentro da classe (Exemplo sem parâmetro e sem retorno):

Icaro Daflon
C# MOD I

```
private void Calcular()
{
    double resultado = 0;

    if (optAdicao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " + " + txtNumero2.Text;
        resultado = double.Parse(txtNumero1.Text) + double.Parse(txtNumero2.Text);
    }
    else if (optSubtracao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " - " + txtNumero2.Text;
        resultado = double.Parse(txtNumero1.Text) - double.Parse(txtNumero2.Text);
    }
    else if (optMultiplicacao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " * " + txtNumero2.Text;
        resultado = double.Parse(txtNumero1.Text) * double.Parse(txtNumero2.Text);
    }
    else if (optDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " / " + txtNumero2.Text;
        resultado = double.Parse(txtNumero1.Text) / double.Parse(txtNumero2.Text);
    }
    else if (optRestoDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " % " + txtNumero2.Text;
        resultado = double.Parse(txtNumero1.Text) % double.Parse(txtNumero2.Text);
    }
    txtResultado.Text = resultado.ToString();
}
```

- Para chamar esse método, deixe o código do event click do botão calcular da seguinte maneira:

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    Calcular();
}
```

- A chamada de método será explicada melhor posteriormente.
- Agora modifique o código do método Calcular para que tenha passagem de parâmetro, deixando-o desta forma:

```
private void Calcular(double numero1, double numero2)
{
    double resultado = 0;

    if (optAdicao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " + " + txtNumero2.Text;
        resultado = numero1 + numero2;
    }
    else if (optSubtracao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " - " + txtNumero2.Text;
        resultado = numero1 - numero2;
    }
    else if (optMultiplicacao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " * " + txtNumero2.Text;
        resultado = numero1 * numero2;
    }
    else if (optDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " / " + txtNumero2.Text;
        resultado = numero1 / numero2;
    }
    else if (optRestoDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " % " + txtNumero2.Text;
        resultado = numero1 % numero2;
    }
    txtResultado.Text = resultado.ToString();
}
```

- E para chamar este método modifique o código do event click deixando-o desta forma:

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    Calcular(double.Parse(txtNumero1.Text), double.Parse(txtNumero2.Text));
}
```

- Agora, veja o mesmo método, só que agora está retornando um valor. Observe que a palavra void não é utilizada:

Icaro Daflon
C# MOD I

```
private double Calcular(double numero1, double numero2)
{
    double resultado = 0;

    if (optAdicao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " + " + txtNumero2.Text;
        resultado = numero1 + numero2;
    }
    else if (optSubtracao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " - " + txtNumero2.Text;
        resultado = numero1 - numero2;
    }
    else if (optMultiplicacao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " * " + txtNumero2.Text;
        resultado = numero1 * numero2;
    }
    else if (optDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " / " + txtNumero2.Text;
        resultado = numero1 / numero2;
    }
    else if (optRestoDivisao.Checked)
    {
        txtExpressao.Text = txtNumero1.Text + " % " + txtNumero2.Text;
        resultado = numero1 % numero2;
    }

    return resultado;
}
```

- Observe o retorno (return resultado;)
- Para chamar esse método, modifique o código do event click deixando-o da seguinte maneira:

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double total = Calcular(double.Parse(txtNumero1.Text),
        double.Parse(txtNumero2.Text));

    txtResultado.Text = total.ToString();
}
```

- Veja que foi necessário criar uma variável do mesmo tipo do método, no caso, double, para receber o valor de retorno do método Calcular.
- Salve o projeto e em seguida feche o Visual Studio.

Chamando Métodos, Páginas 162 e 163

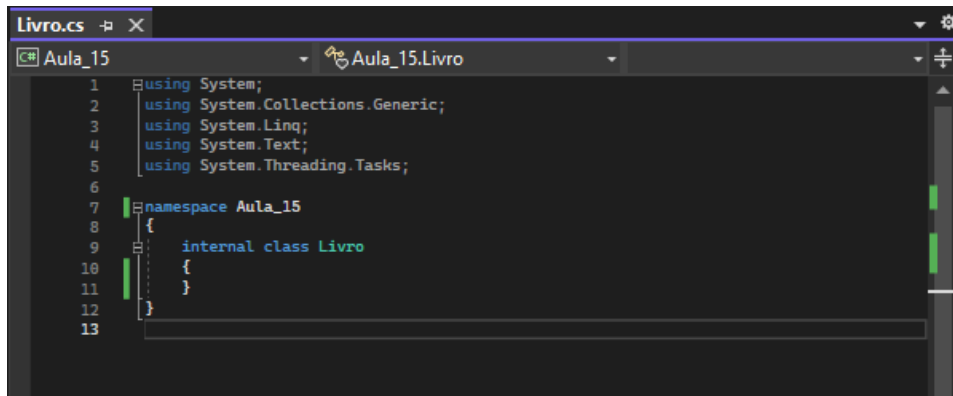
- Conteúdo teórico de acordo, portanto sem alterações.

AULA 15

Formulário Cadastrar Livro, Interatividade

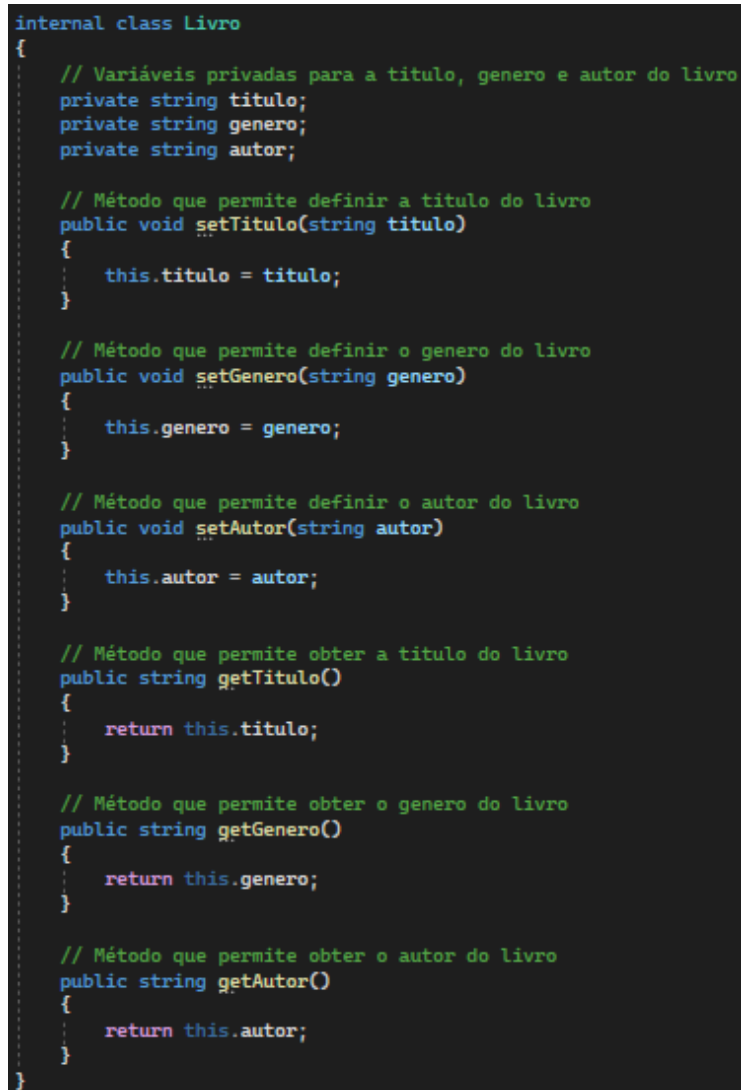
- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_15 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCadastrarLivro e altere a propriedade Text do mesmo para Cadastrar Livro.
- Na janela que irá aparecer clique em “Sim”.
- Com o projeto aberto, adicione uma nova Classe com o nome de Livro.cs

- O código inicial da classe será exibido:



```
Livro.cs
Aula_15
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Aula_15
8 {
9     internal class Livro
10    {
11    }
12 }
13
```

- Agora, coloque o código abaixo em sua classe Livro, deixando-a desta forma:



```
internal class Livro
{
    // Variáveis privadas para a titulo, genero e autor do livro
    private string titulo;
    private string genero;
    private string autor;

    // Método que permite definir a titulo do livro
    public void setTitulo(string titulo)
    {
        this.titulo = titulo;
    }

    // Método que permite definir o genero do livro
    public void setGenero(string genero)
    {
        this.genero = genero;
    }

    // Método que permite definir o autor do livro
    public void setAutor(string autor)
    {
        this.autor = autor;
    }

    // Método que permite obter a titulo do livro
    public string getTitulo()
    {
        return this.titulo;
    }

    // Método que permite obter o genero do livro
    public string getGenero()
    {
        return this.genero;
    }

    // Método que permite obter o autor do livro
    public string getAutor()
    {
        return this.autor;
    }
}
```

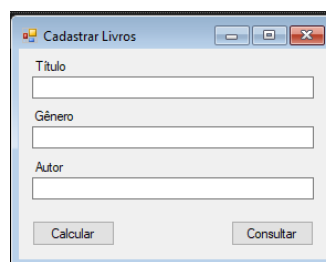
- O comentário de cada linha explica o que ela faz.
- Vamos ver, agora, a utilização da classe Livro que criamos
- Para isso crie um novo formulário com o nome frmCadastrarLivro e altere sua propriedade Text para Cadastrar Livro

Icaro Daflon
C# MOD I

➤ Com o formulário aberto insira os controles a seguir com as seguintes propriedades:

- Label1
 - Text: Título
- Label2
 - Text: Gênero
- Label3
 - Text: Autor
- TextBox1
 - Name: txtTitulo
- TextBox2
 - Name: txtGenero
- TextBox3
 - Name: txtAutor
- Button1
 - Name: btnCadastrar
 - Text: Cadastrar
- Button2
 - Name: btnConsultar
 - Text: Consultar

➤ Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



➤ Agora dê um duplo clique no botão Cadastrar e acima do event click escreva o código:

```
// Cria um objeto do tipo da classe livro  
Livro livro = new Livro();
```

➤ A classe ficará da seguinte forma:

Icaro Daflon
C# MOD I

```
public partial class frmCadastrarLivro : Form
{
    public frmCadastrarLivro()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe livro
    Livro livro = new Livro();

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

- Veja que estamos criando uma instância da classe Livro. Como estamos criando essa instância fora de qualquer método, ela poderá ser usada por qualquer método ou evento que for criado nessa classe.
- Agora, insira dentro do event click do botão btnCadastrar o seguinte código:

```
// Utiliza o método setTitulo da classe livro para armazenar a titulo
livro.setTitulo(txtTitulo.Text);

// Utiliza o método setGenero da classe livro para armazenar o genero
livro.setGenero(txtGenero.Text);

// Utiliza o método setAutor da classe livro para armazenar o autor
livro.setAutor(txtAutor.Text);

// Limpa os campos da tela
txtTitulo.Text = null;
txtGenero.Text = null;
txtAutor.Text = null;
```

- Agora volte para o design do formulário, dê um duplo clique no botão btnConsultar para criar o event click e, em seguida, escreva o código abaixo:

```
// Cria variáveis para receber os dados do veiculo
string titulo, genero, autor;

// Utiliza o método getTitulo da classe livro para obter a titulo
titulo = livro.getTitulo();

// Utiliza o método getGenero da classe livro para obter a genero
genero = livro.getGenero();

// Utiliza o método getAutor da classe livro para obter a autor
autor = livro.getAutor();

// Exibe uma MessageBox com os dados do veiculo cadastrado anteriormente
MessageBox.Show("Titulo do Livro: " + titulo +
    " - Genero: " + genero + " - Autor: " + autor, "Livro Cadastrado");
```

- O código completo deve estar da seguinte maneira:

Icaro Daflon
C# MOD I

```
public partial class frmCadastrarLivro : Form
{
    public frmCadastrarLivro()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe livro
    Livro livro = new Livro();

    private void btnCalcular_Click(object sender, EventArgs e)
    {
        // Utiliza o método setTitulo da classe livro para armazenar a titulo
        livro.setTitulo(txtTitulo.Text);

        // Utiliza o método setGenero da classe livro para armazenar o genero
        livro.setGenero(txtGenero.Text);

        // Utiliza o método setAutor da classe livro para armazenar o autor
        livro.setAutor(txtAutor.Text);

        // Limpa os campos da tela
        txtTitulo.Text = null;
        txtGenero.Text = null;
        txtAutor.Text = null;
    }

    private void btnConsultar_Click(object sender, EventArgs e)
    {
        // Cria variáveis para receber os dados do veículo
        string titulo, genero, autor;

        // Utiliza o método getTitulo da classe livro para obter a titulo
        titulo = livro.getTitulo();

        // Utiliza o método getGenero da classe livro para obter a genero
        genero = livro.getGenero();

        // Utiliza o método getAutor da classe livro para obter a autor
        autor = livro.getAutor();

        // Exibe uma MessageBox com os dados do veículo cadastrado anteriormente
        MessageBox.Show("Titulo do Livro: " + titulo +
            " - Genero: " + genero + " - Autor: " + autor, "Livro Cadastrado");
    }
}
```

- Salve o projeto e em seguida feche o Visual Studio.

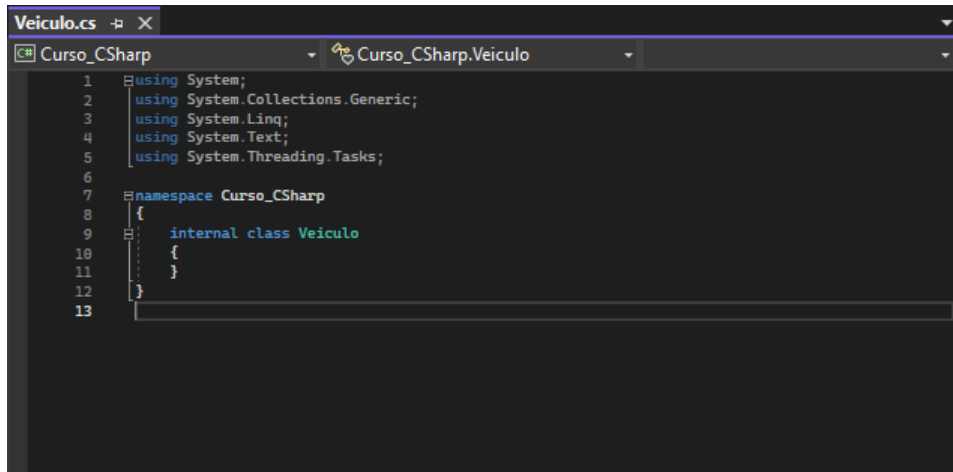
Trabalhando com o Conceito de Classes, Páginas 166 a 176

- Item 2 da explicação de como construir uma classe, traduzir para a ordem: Adicionar → Novo Item... → Classe
- O restante do conteúdo teórico está de acordo, portanto sem alterações.

Passo a Passo:

- Peça a seu instrutor a pasta “Aula I4”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula I4”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, adicione uma nova Classe com o nome de Veiculo.cs
- O código inicial da classe será exibido:

Icaro Daflon
C# MOD I



```
Veiculo.cs
C# Curso_CSharp
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Curso_CSharp
8 {
9     internal class Veiculo
10     {
11     }
12 }
13
```

- Agora, coloque o código abaixo em sua classe Veiculo, deixando-a desta forma:

```
internal class Veiculo
{
    // Variáveis privadas para a marca, modelo e ano do veículo
    private string marca;
    private string modelo;
    private string ano;

    // Método que permite definir a marca do veículo
    public void setMarca (string marca)
    {
        this.marca = marca;
    }

    // Método que permite definir o modelo do veículo
    public void setModelo(string modelo)
    {
        this.modelo = modelo;
    }

    // Método que permite definir o ano do veículo
    public void setAno(string ano)
    {
        this.ano = ano;
    }

    // Método que permite obter a marca do veículo
    public string getMarca()
    {
        return this.marca;
    }

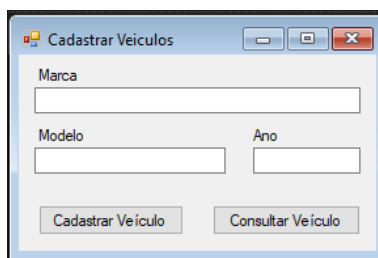
    // Método que permite obter o modelo do veículo
    public string getModelo()
    {
        return this.modelo;
    }

    // Método que permite obter o ano do veículo
    public string getAno()
    {
        return this.ano;
    }
}
```

- O comentário de cada linha explica o que ela faz.
- Vamos ver, agora, a utilização da classe Veiculo que criamos

Icaro Daflon
C# MOD I

- Para isso crie um novo formulário com o nome `frmCadastrarVeiculos` e altere sua propriedade `Text` para `Cadastrar Veículo`
- Com o formulário aberto insira os controles a seguir com as seguintes propriedades:
 - `Label1`
 - `Text: Marca`
 - `Label2`
 - `Text: Modelo`
 - `Label3`
 - `Text: Ano`
 - `TextBox1`
 - `Name: txtMarca`
 - `TextBox2`
 - `Name: txtModelo`
 - `TextBox3`
 - `Name: txtAno`
 - `Button1`
 - `Name: btnCadastrar`
 - `Text: Cadastrar Veículo`
 - `Button2`
 - `Name: btnConsultar`
 - `Text: Consultar Veículo`
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Agora dê um duplo clique no botão `Cadastrar Veículo` e acima do event click escreva o código:

```
// Cria um objeto do tipo da classe Veiculo  
Veiculo veiculo = new Veiculo();
```

- A classe ficará da seguinte forma:

Icaro Daflon
C# MOD I

```
namespace Curso_CSharp
{
    public partial class frmCadastrarVeiculos : Form
    {
        public frmCadastrarVeiculos()
        {
            InitializeComponent();

            // Cria um objeto do tipo da classe Veiculo
            Veiculo veiculo = new Veiculo();

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

- Veja que estamos criando uma instância da classe Veiculo. Como estamos criando essa instância fora de qualquer método, ela poderá ser usada por qualquer método ou evento que for criado nessa classe.
- Agora, insira dentro do event click do botão btnCadastrar o seguinte código:

```
// Utiliza o método setMarca da classe Veiculo para armazenar a marca
veiculo.setMarca(txtMarca.Text);

// Utiliza o método setModelo da classe Veiculo para armazenar a marca
veiculo.setModelo(txtModelo.Text);

// Utiliza o método setAno da classe Veiculo para armazenar a marca
veiculo.setAno(txtAno.Text);

// Limpa os campos da tela
txtMarca.Text = null;
txtModelo.Text = null;
txtAno.Text = null;
```

- Agora volte para o design do formulário, dê um duplo clique no botão Consultar Veículo para criar o event click e, em seguida, escreva o código abaixo:

```
// Cria variáveis para receber os dados do veiculo
string marca, modelo, ano;

// Utiliza o método getMarca da classe Veiculo para obter a marca
marca = veiculo.getMarca();

// Utiliza o método getModelo da classe Veiculo para obter a modelo
modelo = veiculo.getModelo();

// Utiliza o método getAno da classe Veiculo para obter a ano
ano = veiculo.getAno();

// Exibe uma MessageBox com os dados do veiculo cadastrado anteriormente
MessageBox.Show("Marca do Veiculo: " + marca +
    " - Modelo: " + modelo + " - Ano: " + ano, "Veículo Cadastrado");
```

- O código completo deve estar da seguinte maneira:

Icaro Daflon

C# MOD I

```
public partial class frmCadastrarVeiculos : Form
{
    public frmCadastrarVeiculos()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe Veiculo
    Veiculo veiculo = new Veiculo();

    private void btnCadastrar_Click(object sender, EventArgs e)
    {
        // Utiliza o método setMarca da classe Veiculo para armazenar a marca
        veiculo.setMarca(txtMarca.Text);

        // Utiliza o método setModelo da classe Veiculo para armazenar a marca
        veiculo.setModelo(txtModelo.Text);

        // Utiliza o método setAno da classe Veiculo para armazenar a marca
        veiculo.setAno(txtAno.Text);

        // Limpa os campos da tela
        txtMarca.Text = null;
        txtModelo.Text = null;
        txtAno.Text = null;
    }

    private void btnConsultar_Click(object sender, EventArgs e)
    {
        // Cria variáveis para receber os dados do veiculo
        string marca, modelo, ano;

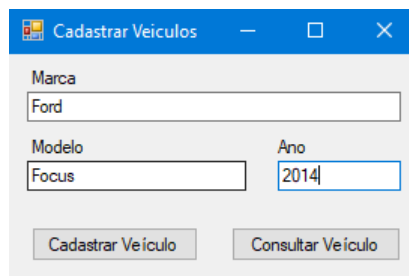
        // Utiliza o método getMarca da classe Veiculo para obter a marca
        marca = veiculo.getMarca();

        // Utiliza o método getModelo da classe Veiculo para obter a modelo
        modelo = veiculo.getModelo();

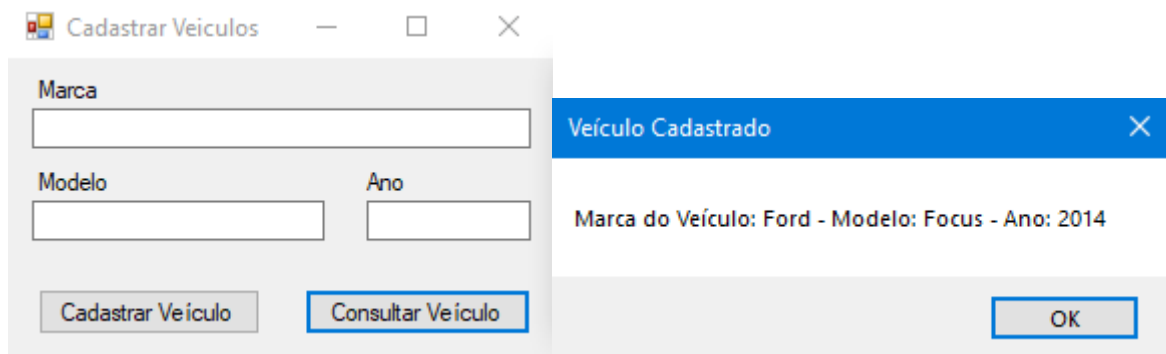
        // Utiliza o método getAno da classe Veiculo para obter a ano
        ano = veiculo.getAno();

        // Exibe uma MessageBox com os dados do veiculo cadastrado anteriormente
        MessageBox.Show("Marca do Veículo: " + marca +
            " - Modelo: " + modelo + " - Ano: " + ano, "Veículo Cadastrado");
    }
}
```

- Altere a classe Program.cs para abrir o formulário frmCadastrarVeiculos.
- Salve a solução e execute o programa. Insira os dados necessários nos campos da tela e após clique em Cadastrar Veículo.



- Agora, clique em consultar e veja o resultado:



- Teste as funcionalidade e resultados ao finalizar, feche o Visual Studio.

AULA 16

Formulário Caixa Eletrônico, Interatividade

- Abra o Visual Studio 2022
- Crie um projeto “Aplicativo do Windows Forms (.NET Framework)” de nome Aula_I6 e estrutura .NET Framework 4.8
- Renomeie o formulário “Form1” para frmCaixaEletronico e altere a propriedade Text do mesmo para Caixa Eletrônico.
- Na janela que irá aparecer clique em “Sim”.
- Com o projeto aberto, adicione uma nova Classe com o nome de ContaBancaria.cs
- Agora, coloque o código abaixo em sua classe ContaBancaria, deixando-a desta forma:

```
// Variáveis privadas para o nome, agência e numero da conta
private string nome;
private string agencia;
private string numeroConta;

// Método que permite definir o nome do titular da conta
public void setNome(string nome)
{
    this.nome = nome;
}

// Método que permite definir a agência
public void setAgencia(string agencia)
{
    this.agencia = agencia;
}

// Método que permite definir o número da conta
public void setNumeroConta(string numeroConta)
{
    this.numeroConta = numeroConta;
}

// Método que permite obter o nome do titular da conta
public string getNome()
{
    return this.nome;
}

// Método que permite obter a agência
public string getAgencia()
{
    return this.agencia;
}

// Método que permite obter o número da conta
public string getNumeroConta()
{
    return this.numeroConta;
}
```

- O comentário de cada linha explica o que ela faz.
- Crie outra classe agora, com o nome de CaixaEletronico

- Agora, faremos com que a classe CaixaEletronico herde a classe ContaBancaria, para que possa utilizar os atributos de uma conta.
- Para isso altere a declaração da classe CaixaEletronico para que fique da seguinte forma:

```
internal class CaixaEletronico : ContaBancaria
{
}
```

- Adicione o seguinte código na classe CaixaEletronico:

```
// Cria variáveis (atributos) privadas para armazenar a matricula e o curso do aluno
private string matricula;
private string curso;

// Método que permite definir o curso da veiculo
public void setMatricula(string matricula)
{
    this.matricula = matricula;
}

// Método que permite definir o curso da veiculo
public void setCurso(string curso)
{
    this.curso = curso;
}

// Método que permite obter a matricula da pessoa
public string getMatricula()
{
    return this.matricula;
}

// Método que permite obter o curso da pessoa
public string getCurso()
{
    return this.curso;
}
```

- Vamos ver, agora, a utilização da classe CaixaEletronico que criamos, herdando os atributos da classe ContaBancaria.
- Para isso, abra o formulário frmCaixaEletronico e insira os controles a seguir com as seguintes propriedades:
 - Label1
 - Text: Nome do Titular da Conta
 - Label2
 - Text: Agência
 - Label3
 - Text: N° da Conta
 - Label4
 - Text: Valor
 - TextBox1
 - Name: txtNome
 - TextBox2
 - Name: txtAgencia
 - TextBox3
 - Name: txtNumeroDaConta

- TextBox4
 - Name: txtValor
 - TextBox5
 - Name: txtExibir
 - BackColor: ScrollBar
 - Multiline: True
 - ScrollBars: Vertical
 - Button1
 - Name: btnCadastrar
 - Text: Cadastrar
 - Button2
 - Name: btnConfirmar
 - Text: Confirmar
 - GroupBox1
 - Text: Abrindo sua Conta
 - Name: groupBoxConta
 - GroupBox2
 - Text: Caixa Eletrônico
 - Name: groupBoxCaixa
 - RadioButton1
 - Text: Depósito
 - Name: optDeposito
 - RadioButton2
 - Text: Saque
 - Name: optSaque
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)

Icaro Daflon
C# MOD I

- Agora dê um duplo clique no botão Cadastrar e acima do event click escreva o código:

```
// Cria um objeto do tipo da classe CaixaEletronico
CaixaEletronico caixaEletronico = new CaixaEletronico();
```

- A classe ficará da seguinte forma:

```
public partial class frmCaixaEletronico : Form
{
    public frmCaixaEletronico()
    {
        InitializeComponent();

        // Cria um objeto do tipo da classe CaixaEletronico
        CaixaEletronico caixaEletronico = new CaixaEletronico();

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

- Veja que estamos criando uma instância da classe CaixaEletronico. Como estamos criando essa instância fora de qualquer método, ela poderá ser usada por qualquer método ou evento que for criado nessa classe.
- Agora, insira dentro do event click do botão btnCadastrar o seguinte código:

```
// Armazena os campos nome, agência, número da conta e o saldo inicial de 500
caixaEletronico.setNome(txtNome.Text);
caixaEletronico.setAgencia(txtAgencia.Text);
caixaEletronico.setNumeroConta(txtAgencia.Text);
caixaEletronico.setSaldo(500);

txtExibir.Text = "Conta Cadastrada com sucesso!!!";

// Limpa os campos da tela
txtNome.Text = null;
txtAgencia.Text = null;
txtNumeroDaConta.Text = null;
```

- Agora volte para o design do formulário, dê um duplo clique no botão btnConfirmar para criar o event click e, em seguida, escreva o código abaixo:

Icaro Daflon

C# MOD I

```
// Cria variáveis para armazenar o nome, agência, número da conta e o saldo
string nome, agencia, numeroDaConta;
double saldo;

// Atribui os valores respectivos às variáveis criadas acima
nome = caixaEletronico.getNome();
agencia = caixaEletronico.getAgencia();
numeroDaConta = caixaEletronico.getNumeroConta();
saldo = caixaEletronico.getSaldo();

// Verifica se a opção Depósito está selecionada
if (optDeposito.Checked)
{
    // Adiciona o valor do controle txtValor da variável saldo
    saldo += double.Parse(txtValor.Text);
}
// Verifica se a opção Saque está selecionada
else if (optSaque.Checked)
{
    // Subtrai o valor do controle txtValor da variável saldo
    saldo -= double.Parse(txtValor.Text);
}

// Atualiza o valor do saldo no Objeto caixaEletronico
caixaEletronico.setSaldo(saldo);

// Exibe o resultado da transação no controle txtExibir
txtExibir.Text = "Transação efetuada com sucesso!!!\r\n\r\n" +
    "Nome do Titular: " + nome + "\r\n" +
    "Número da Conta: " + numeroDaConta + "\r\n" +
    "Agência: " + agencia + "\r\n" + "Saldo Atual: " + saldo;
```

- O código completo deve estar da seguinte maneira:

```
public partial class frmCaixaEletronico : Form
{
    public frmCaixaEletronico()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe CaixaEletronico
    CaixaEletronico caixaEletronico = new CaixaEletronico();

    private void btnCadastrar_Click(object sender, EventArgs e)
    {
        // Armazena os campos nome, agência, número da conta e o saldo inicial de 500
        caixaEletronico.setNome(txtNome.Text);
        caixaEletronico.setAgencia(txtAgencia.Text);
        caixaEletronico.setNumeroConta(txtNumeroConta.Text);
        caixaEletronico.setSaldo(500);

        txtExibir.Text = "Conta Cadastrada com sucesso!!!";

        // Limpa os campos da tela
        txtNome.Text = null;
        txtAgencia.Text = null;
        txtNumeroConta.Text = null;
    }

    private void btnConfirmar_Click(object sender, EventArgs e)
    {
        // Cria variáveis para armazenar o nome, agência, número da conta e o saldo
        string nome, agencia, numeroDaConta;
        double saldo;

        // Atribui os valores respectivos às variáveis criadas acima
        nome = caixaEletronico.getNome();
        agencia = caixaEletronico.getAgencia();
        numeroDaConta = caixaEletronico.getNumeroConta();
        saldo = caixaEletronico.getSaldo();

        // Verifica se a opção Depósito está selecionada
        if (optDeposito.Checked)
        {
            // Adiciona o valor do controle txtValor da variável saldo
            saldo += double.Parse(txtValor.Text);
        }
        // Verifica se a opção Saque está selecionada
        else if (optSaque.Checked)
        {
            // Subtrai o valor do controle txtValor da variável saldo
            saldo -= double.Parse(txtValor.Text);
        }

        // Atualiza o valor do saldo no Objeto caixaEletronico
        caixaEletronico.setSaldo(saldo);

        // Exibe o resultado da transação no controle txtExibir
        txtExibir.Text = "Transação efetuada com sucesso!!!\r\n\r\n" +
            "Nome do Titular: " + nome + "\r\n" +
            "Número da Conta: " + numeroDaConta + "\r\n" +
            "Agência: " + agencia + "\r\n" + "Saldo Atual: " + saldo;
    }
}
```

- Salve o projeto e teste suas funcionalidade, ao terminar feche o Visual Studio.

Aplicando o Conceito de Herança, Páginas 178 a 187

- Peça a seu instrutor a pasta “Aula 15”, copie-a e cole-a em seu local de gravação.
- Abra a pasta “Aula 15”, em seguida a pasta “Curso_CSharp” e por fim abra o arquivo “Curso_CSharp.sln”.
- Com o projeto aberto, adicione uma nova Classe com o nome de Pessoa.cs
- Adicione o código a seguir na classe Pessoa:

```
// Cria variáveis (atributos) privadas para armazenar o nome e a idade da pessoa
private string nome;
private int idade;

// Método que permite definir o nome da pessoa
public void setNome(string nome)
{
    this.nome = nome;
}

// Método que permite definir o idade da pessoa
public void setIdade(int idade)
{
    this.idade = idade;
}

// Método que permite obter o nome da pessoa
public string getNome()
{
    return this.nome;
}

// Método que permite obter o idade da pessoa
public int getIdade()
{
    return this.idade;
}
```

- Agora, crie uma outra classe, com o nome Aluno.cs
- Nós faremos com que a classe Aluno herde a classe Pessoa, para que possa utilizar os atributos de uma pessoa.
- Para isso altere a declaração da classe Aluno para que fique da seguinte forma:

```
internal class Aluno : Pessoa
{
}
```

- Adicione o seguinte código na classe Aluno:

```
// Cria variáveis (atributos) privadas para armazenar a matrícula e o curso do aluno
private string matricula;
private string curso;

// Método que permite definir o curso da pessoa
public void setCurso(string curso)
{
    this.curso = curso;
}

// Método que permite definir a matrícula da pessoa
public void setMatricula(string matricula)
{
    this.matricula = matricula;
}

// Método que permite obter o curso da pessoa
public string getCurso()
{
    return this.curso;
}

// Método que permite obter a matrícula da pessoa
public string getMatricula()
{
    return this.matricula;
}
```

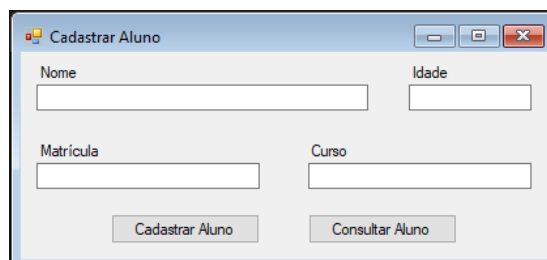
- Vamos ver, agora, a utilização da classe Aluno que criamos, herdando os atributos da classe Pessoa.
- Para isso crie um novo formulário com o nome frmCadastrarAluno e altere sua propriedade Text para Cadastrar Aluno

Icaro Daflon
C# MOD I

➤ Com o formulário aberto insira os controles a seguir com as seguintes propriedades:

- Label1
 - Text: Nome
- Label2
 - Text: Idade
- Label3
 - Text: Matrícula
- Label4
 - Text: Curso
- TextBox1
 - Name: txtNome
- TextBox2
 - Name: txtIdade
- TextBox3
 - Name: txtMatricula
- TextBox4
 - Name: txtCurso
- Button1
 - Name: btnCadastrar
 - Text: Cadastrar Aluno
- Button2
 - Name: btnConsultar
 - Text: Consultar Aluno

➤ Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



➤ Agora dê um duplo clique no botão Cadastrar Aluno e acima do event click escreva o código:

```
// Cria um objeto do tipo da classe Aluno  
Aluno aluno = new Aluno();
```

- O código deve estar assim:

```
public partial class frmCadastrarAluno : Form
{
    public frmCadastrarAluno()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe Aluno
    Aluno aluno = new Aluno();

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

- Agora, dentro do event click do botão btnCadastrar, insira o código abaixo:

```
// Utiliza o método setNome da classe Aluno para armazenar a nome
aluno.setNome(txtNome.Text);

// Utiliza o método setIdade da classe Aluno para armazenar a idade
aluno.setIdade(int.Parse(txtIdade.Text));

// Utiliza o método setMatricula da classe Aluno para armazenar a matrícula
aluno.setMatricula(txtMatricula.Text);

// Utiliza o método setCurso da classe Aluno para armazenar a curso
aluno.setCurso(txtCurso.Text);

// Limpa os campos da tela
txtNome.Text = null;
txtIdade.Text = null;
txtMatricula.Text = null;
txtCurso.Text = null;
```

- Com o botão btnCadastrar finalizado, retorne ao design do formulário e dê um duplo clique sobre o botão btnConsultar e, em seguida, escreva, dentro do event click, o código abaixo:

```
// Cria variáveis para receber os dados do veículo
string nome, matricula, curso;
int idade;

// Utiliza o método getNome da classe Aluno para obter a nome
nome = aluno.getNome();

// Utiliza o método getIdade da classe Aluno para obter a idade
idade = aluno.getIdade();

// Utiliza o método getMatricula da classe Aluno para obter a matrícula
matricula = aluno.getMatricula();

// Utiliza o método getCurso da classe Aluno para obter a curso
curso = aluno.getCurso();

// Exibe uma MessageBox com os dados do aluno cadastrado anteriormente
MessageBox.Show("Nome do Aluno: " + nome +
    " - Idade: " + idade + " - Matrícula: " +
    matricula + " - Curso: " + curso, "Aluno Cadastrado");
```

- O código completo ficará da seguinte forma:

Icaro Daflon C# MOD I

```
public partial class frmCadastrarAluno : Form
{
    public frmCadastrarAluno()
    {
        InitializeComponent();
    }

    // Cria um objeto do tipo da classe Aluno
    Aluno aluno = new Aluno();

    private void btnCadastrar_Click(object sender, EventArgs e)
    {
        // Utiliza o método setNome da classe Aluno para armazenar a nome
        aluno.setNome(txtNome.Text);

        // Utiliza o método setIdade da classe Aluno para armazenar a idade
        aluno.setIdade(int.Parse(txtIdade.Text));

        // Utiliza o método setMatricula da classe Aluno para armazenar a matricula
        aluno.setMatricula(txtMatricula.Text);

        // Utiliza o método setCurso da classe Aluno para armazenar a curso
        aluno.setCurso(txtCurso.Text);

        // Limpa os campos da tela
        txtNome.Text = null;
        txtIdade.Text = null;
        txtMatricula.Text = null;
        txtCurso.Text = null;
    }

    private void btnConsultar_Click(object sender, EventArgs e)
    {
        // Cria variáveis para receber os dados do veiculo
        string nome, matricula, curso;
        int idade;

        // Utiliza o método getNome da classe Aluno para obter a nome
        nome = aluno.getNome();

        // Utiliza o método getIdade da classe Aluno para obter a idade
        idade = aluno.getIdade();

        // Utiliza o método getMatricula da classe Aluno para obter a matricula
        matricula = aluno.getMatricula();

        // Utiliza o método getCurso da classe Aluno para obter a curso
        curso = aluno.getCurso();

        // Exibe uma MessageBox com os dados do aluno cadastrado anteriormente
        MessageBox.Show("Nome do Aluno: " + nome +
            "\n - Idade: " + idade + "\n - Matricula: " +
            matricula + "\n - Curso: " + curso, "Aluno Cadastrado");
    }
}
```

- Altere a classe Program.cs para abrir o formulário frmCadastrarAluno.
- Salve a solução e execute o programa. Após finalizar os testes feche o Visual Studio.

AULA 17

Formulário Polimorfismo, Iteratividade

- Crie um novo projeto Aplicativo do Windows Forms (.NET Framework) com o nome Aula_17 e Estrutura .NET Framework 4.8
- Com o projeto aberto, adicione uma nova Interface com o nome de Operacao.cs
- Deixe o código da Interface Operacao como abaixo:

```
internal interface Operacao
{
    // Cria a definição do método Calcular
    int Calcular(int numero1, int numero2);
}
```

Icaro Daflon
C# MOD I

- Agora, crie uma classe, com o nome Soma.cs
- Deixe a classe Soma da seguinte forma:

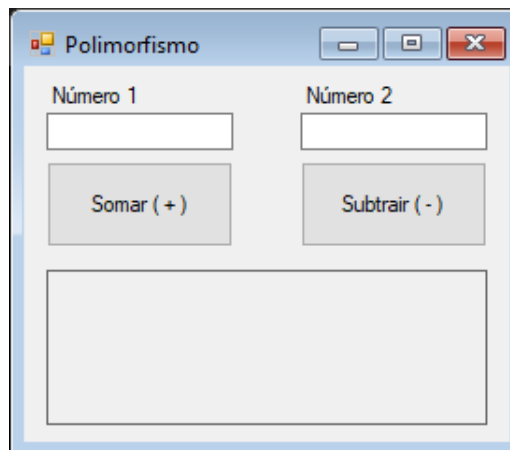
```
internal class Soma : Operacao
{
    // Implementa o método Calcular declarado na Interface Operacao
    public int Calcular(int numero1, int numero2)
    {
        return numero1 + numero2;
    }
}
```

- Agora, crie outra classe, agora com o nome Subtracao.cs
- Deixe a classe Subtracao da seguinte forma (Não esqueça de alterar o sinal para (-) no comando return):

```
internal class Subtracao : Operacao
{
    // Implementa o método Calcular declarado na Interface Operacao
    public int Calcular(int numero1, int numero2)
    {
        return numero1 - numero2;
    }
}
```

- Agora iremos editar o formulário já criado. Portanto altere o nome do Formulário para frmPrincipal e sua propriedade Text para Polimorfismo
- Com o formulário aberto insira os controles a seguir com as seguintes propriedades:
 - Label1
 - Text: Número 1
 - Label2
 - Text: Número 2
 - Label3
 - Text: *Apague o conteúdo*
 - Name: lblMostrar
 - BorderStyle: FixedSingle
 - AutoSize: False
 - TextBox1
 - Name: txtNumero1
 - TextBox2
 - Name: txtNumero2
 - Button1
 - Name: btnSomar
 - Text: Somar (+)
 - Button2
 - Name: btnSubtrair
 - Text: Subtrair (-)

- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Agora dê um duplo clique no botão Somar escreva o código:

```
// Verifica se as caixas de texto estão vazias
if (txtNumero1.Text != "" & txtNumero2.Text != "")
{
    // Cria uma instância da Interface Operacao baseada na classe Soma
    Operacao operacao = new Soma();

    // Cria uma variavel resultado atribuindo-a com o retorno do método Calcular
    int resultado = operacao.Calcular(int.Parse(txtNumero1.Text),
                                     int.Parse(txtNumero2.Text));

    // Exibe o resultado no controle lblMostrar
    lblMostrar.Text = "A soma dos números informados é: " + resultado;
}
```

- Volte para a guia Design do formulário e agora dê um duplo clique no botão Subtrair e, após, escreva o código abaixo:

```
// Verifica se as caixas de texto estão vazias
if (txtNumero1.Text != "" & txtNumero2.Text != "")
{
    // Cria uma instância da Interface Operacao baseada na classe Subtracao
    Operacao operacao = new Subtracao();

    // Cria uma variavel resultado atribuindo-a com o retorno do método Calcular
    int resultado = operacao.Calcular(int.Parse(txtNumero1.Text),
                                     int.Parse(txtNumero2.Text));

    // Exibe o resultado no controle lblMostrar
    lblMostrar.Text = "A subtração dos números informados é: " + resultado;
}
```

- Salve a solução e execute o programa. Após finalizar os testes feche o Visual Studio.

Interface, Páginas 190 a 199

- Crie um novo projeto Aplicativo do Windows Forms (.NET Framework) com o nome Polimorfismo e Estrutura .NET Framework 4.8
- Com o projeto aberto, adicione uma nova Interface com o nome de Figura.cs
- Deixe o código da Interface figura como abaixo:

```
namespace Polimorfismo
{
    internal interface Figura
    {
        // Cria a definição dos métodos CalcularArea e CalcularPerimetro
        double CalcularArea(double medida);
        double CalcularPerimetro(double medida);
    }
}
```

- Agora, crie uma classe, com o nome Quadrado.cs
- Deixe a classe Quadrado da seguinte forma:

```
internal class Quadrado : Figura
{
    // Implementa o método CalcularArea declarado na Interface Figura
    public double CalcularArea(double medida)
    {
        return Math.Pow(medida, 2);
    }

    // Implementa o método CalcularPerimetro declarado na Interface Figura
    public double CalcularPerimetro(double medida)
    {
        return medida * 4;
    }
}
```

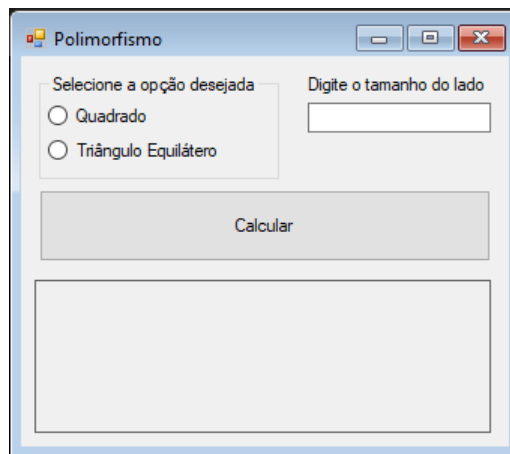
- Agora, crie outra classe, agora com o nome Triangulo.cs
- Deixe a classe Triangulo da seguinte forma:

```
internal class Triangulo : Figura
{
    // Implementa o método CalcularArea declarado na Interface Figura
    public double CalcularArea(double medida)
    {
        return (Math.Pow(medida, 2) * Math.Sqrt(3)) / 4;
    }

    // Implementa o método CalcularPerimetro declarado na Interface Figura
    public double CalcularPerimetro(double medida)
    {
        return medida * 3;
    }
}
```

- Agora iremos editar o formulário já criado. Portanto altere o nome do Formulário para frmPrincipal e sua propriedade Text para Polimorfismo
- Com o formulário aberto insira os controles a seguir com as seguintes propriedades:
 - Label1
 - Text: Digite o tamanho do lado
 - Label2
 - Text: *Apague o conteúdo*
 - Name: lblMostrar
 - BorderStyle: FixedSingle
 - AutoSize: False
 - TextBox1
 - Name: txtMedida

- ButtonI
 - Name: btnCalcular
 - Text: Calcular
 - GroupBox
 - Text: Selecione a opção desejada
 - Name: groupBoxOpcao
 - RadioButton
 - Text: Quadrado
 - Name: optQuadrado
 - RadioButton
 - Text: Triângulo Equilátero
 - Name: optTriangulo
- Organize os controles da seguinte forma: (Não esqueça de redimensionar o tamanho do formulário para se adequar aos itens inseridos)



- Agora dê um duplo clique no botão Calcular escreva o código:

Icaro Daflon

C# MOD I

```
// Verifica se a caixa de texto está vazia ou não
if (txtMedida.Text != "")
{
    //Verifica se a opção Quadrado está selecionada
    if (optQuadrado.Checked)
    {
        // Cria uma instância da interface Figura baseada na classe Quadrado
        Figura figura = new Quadrado();

        // Cria as variáveis para armazenar os resultados dos métodos criados
        double area = figura.CalcularArea(double.Parse(txtMedida.Text));
        double perimetro = figura.CalcularPerimetro(double.Parse(txtMedida.Text));

        // Exibe o resultado no controle lblMostrar
        lblMostrar.Text = "Perimetro: " + perimetro + "\r\nÁrea: " + area;
    }
    // Verifica se a opção Triângulo Equilátero está selecionada
    else if (optTriangulo.Checked)
    {
        // Cria uma instância da interface Figura baseada na classe Triangulo
        Figura figura = new Triangulo();

        // Cria as variáveis para armazenar os resultados dos métodos criados
        double area = figura.CalcularArea(double.Parse(txtMedida.Text));
        double perimetro = figura.CalcularPerimetro(double.Parse(txtMedida.Text));

        // Exibe o resultado no controle lblMostrar
        lblMostrar.Text = "Perimetro: " + perimetro + "\r\nÁrea: " + area;
    }
    // Será executado caso o usuário não selecione uma opção
    else
    {
        MessageBox.Show("Você deve selecionar a forma geométrica desejada!");
    }
}
// Será executado caso o usuário deixa a caixa de texto da Medida vazia
else
{
    MessageBox.Show("Informe o valor do lado!");
}
```

- Salve a solução e execute o programa. Após finalizar os testes feche o Visual Studio.