



Programação Orientada a Objetos

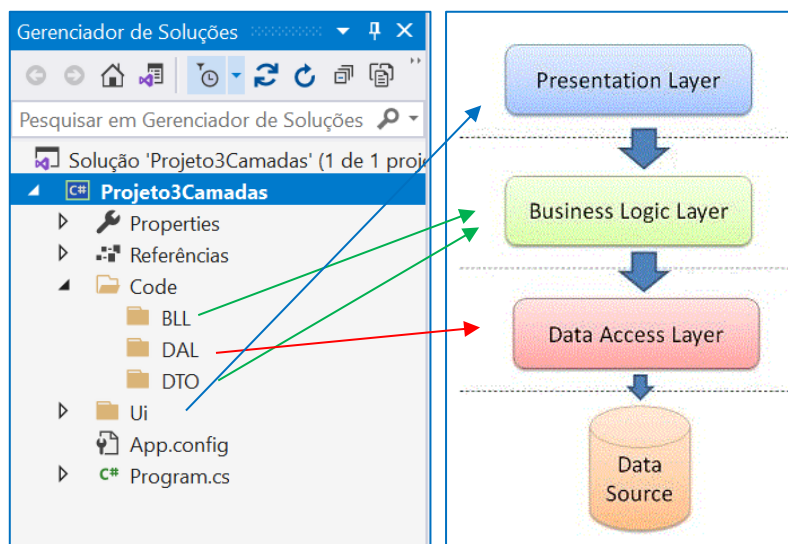
Professora Michelle Belli

PROJETO POO 3ª ETAPA

3-PROGRAMAÇÃO EM TRÊS CAMADAS

□ TEORIA:

- BLL: Business Logic Layer
- DAL: Data Access Layer
- DTO: Data Transfer Object








☐ Crie um projeto:


Criar um novo projeto


forms ✕ Limpar tudo


Todos os idiomas ▼ Todas as plataformas ▼ Todos os tipos de proje... ▼


Modelos de projeto recentes


-  Aplicativo do Windows Forms C#
-  Aplicativo Web ASP.NET (.NET Framework) C#
-  ASP.NET Core Vazio C#
-  Aplicativo do Windows Forms (.NET Framework) C#
-  Aplicativo Web do ASP.NET Core (Model-View-Controller) C#

 Aplicativo do Windows Forms C# Windows Área de Trabalho
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.

 Aplicativo do Windows Forms (.NET Framework) C# Windows Área de Trabalho
Um projeto para criar um aplicativo com uma interface do usuário do Windows Forms (WinForms)

 Aplicativo do Windows Forms Visual Basic Windows Área de Trabalho
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.

 Biblioteca de Controles do Windows Forms (.NET Framework) C# Windows Área de Trabalho Biblioteca
Um projeto para criar controles a serem usados em aplicativos do Windows Forms (WinForms)

 Aplicativo do Windows Forms (.NET Framework) Visual Basic Windows Área de Trabalho

Voltar Próximo

☐ Defina o nome e o local onde será criado e clique em criar

Aplicativo do Windows Forms (.NET Framework) C# Windows Área de Trabalho

Nome do projeto

Projeto3Camadas

Local

C:\Users\Michelle Belli\Desktop\COTEMIG\POO\projetoEmCamadas ...

Nome da solução ⓘ

Projeto3Camadas

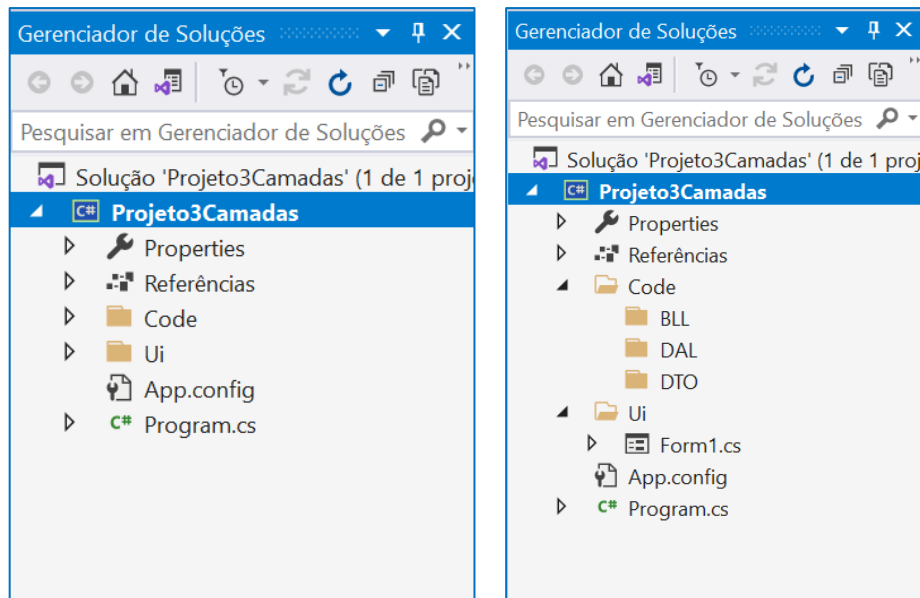
☐ Colocar a solução e o projeto no mesmo diretório

Framework

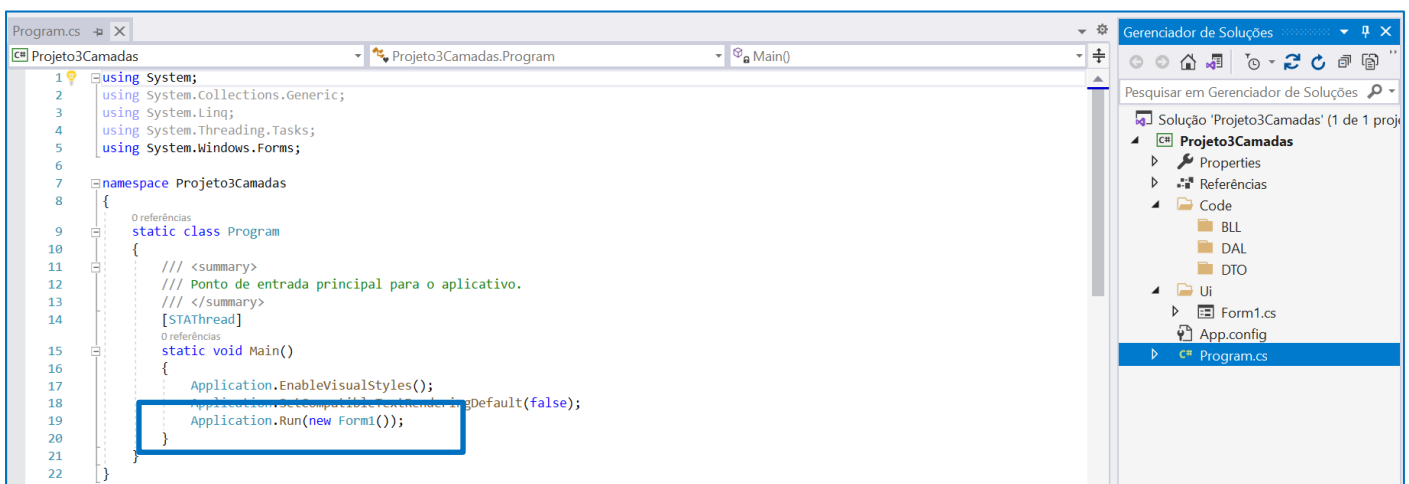
.NET Framework 4.7.2 ▼

Voltar Criar

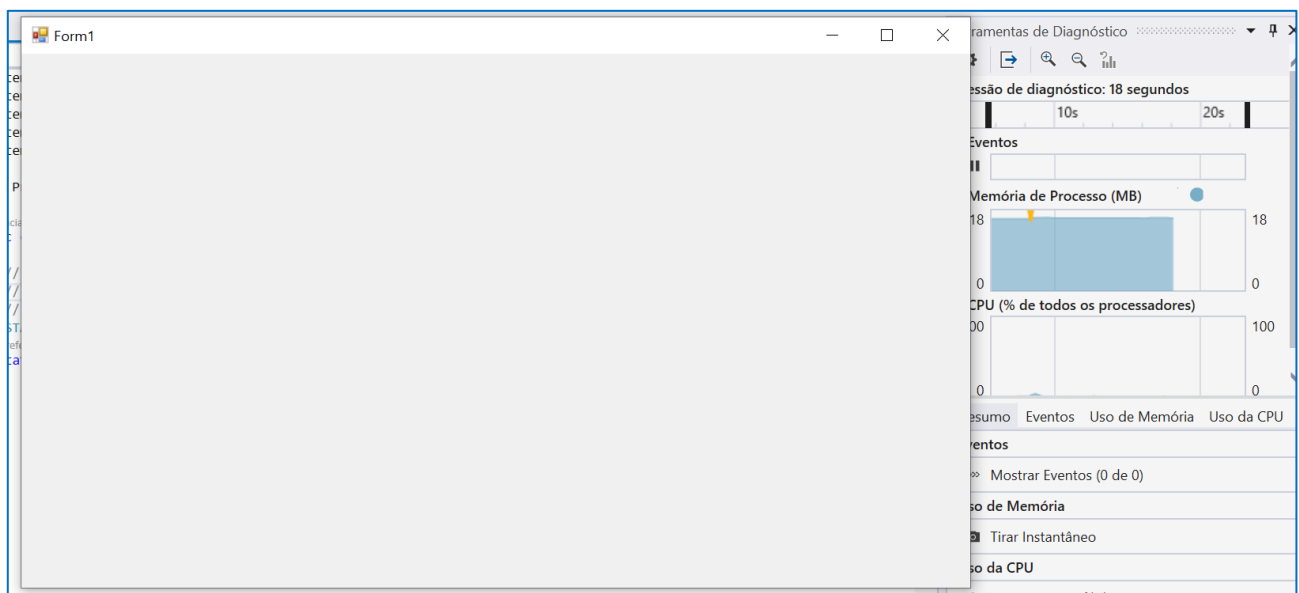
- ❑ Crie as pastas que irão representar as camadas e arraste o form1 para dentro da pasta Ui:



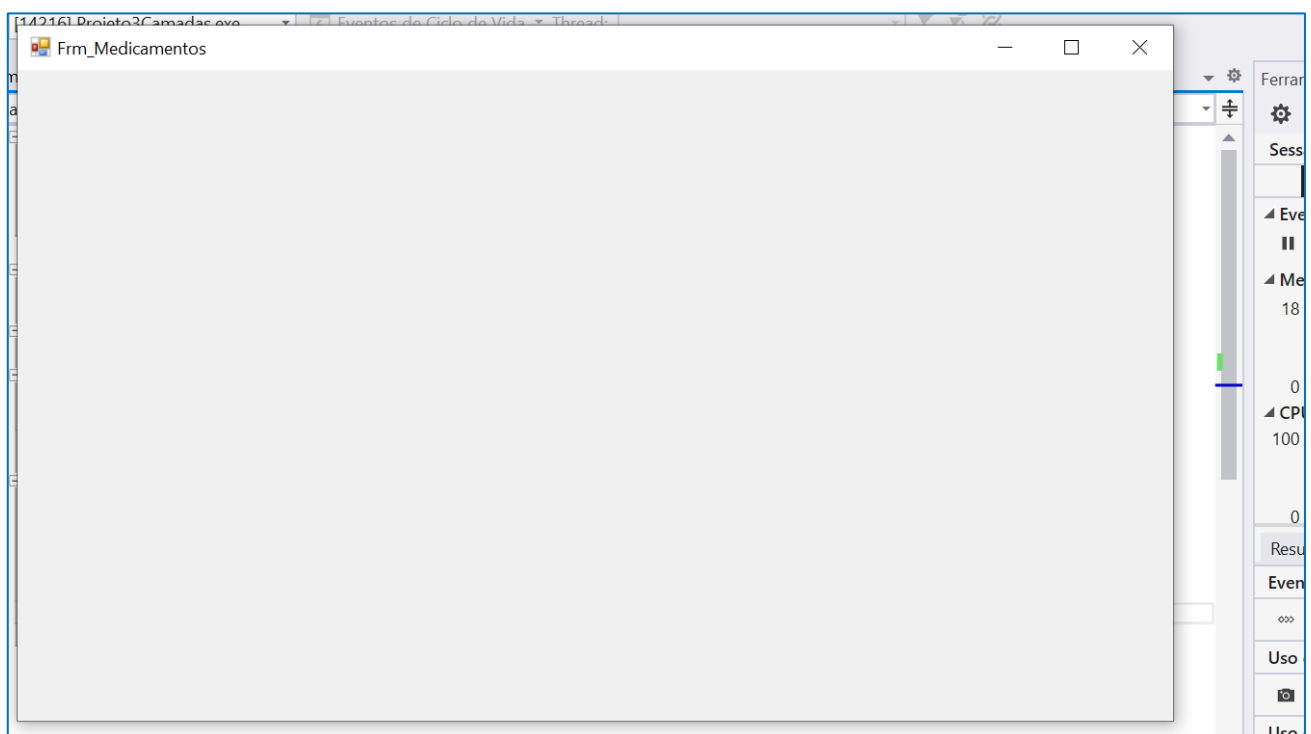
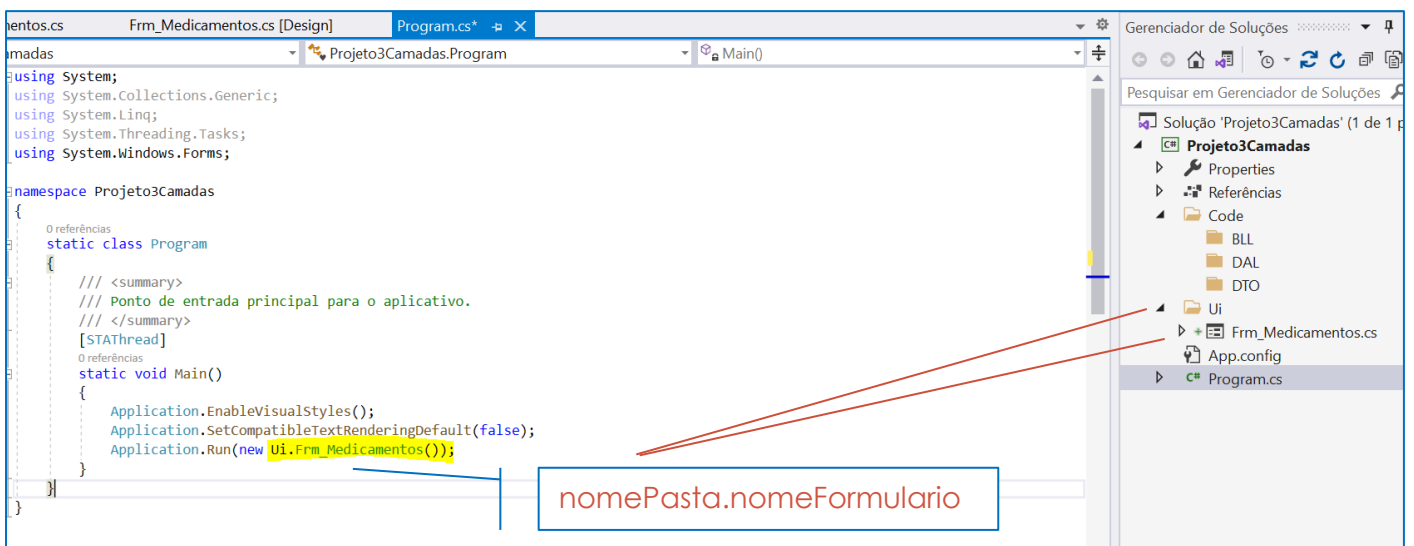
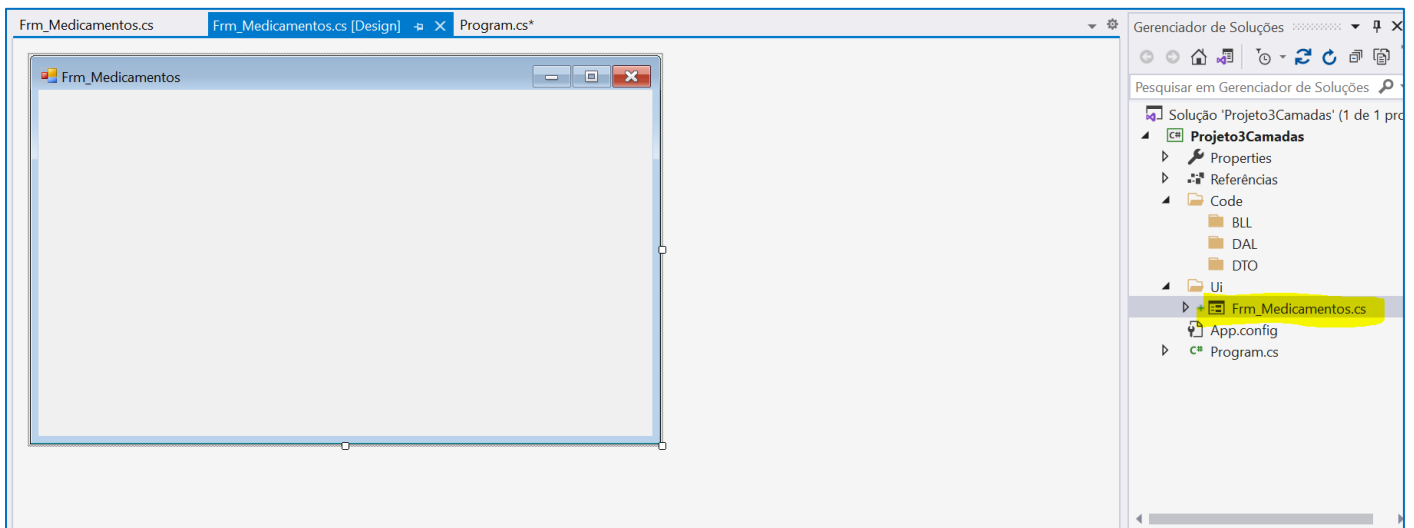
- ❑ Confira se o primeiro formulário a ser executado continua sendo o Form1:



- ❑ Execute e veja se está tudo ok!



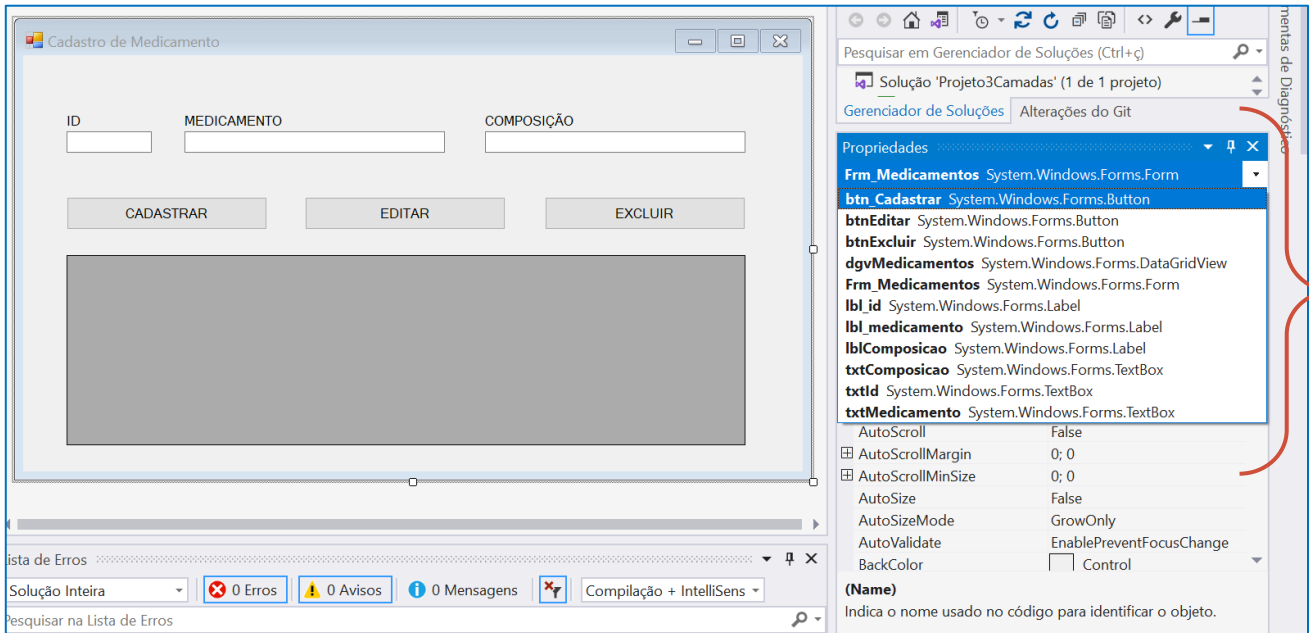
- Como exemplo vou criar um cadastro de medicamentos. Primeiro renomeie o formulário, depois ajuste o Program.cs para acessar corretamente o formulário e teste, sempre realize testes!



□ Coloque no formulário:

- 3 label: lblId, lblMedicamento, lblComposicao
- 3 textbox: txtId, txtMedicamento, txtComposicao
- 3 botoes: btnCadastrar, btnEditar, btnExcluir
- 1 dataGridView: dgvMedicamentos

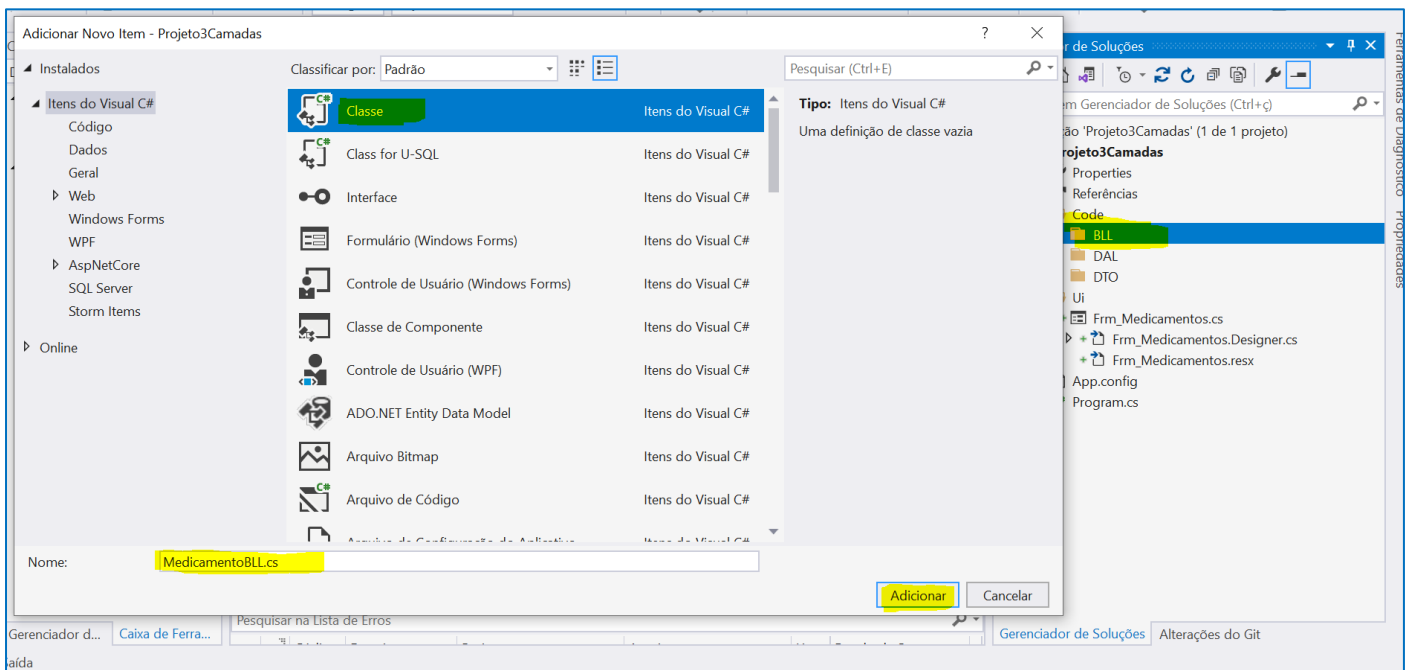
□ A imagem abaixo demonstra a tela e em propriedades é possível ver os nomes de todos os elementos.



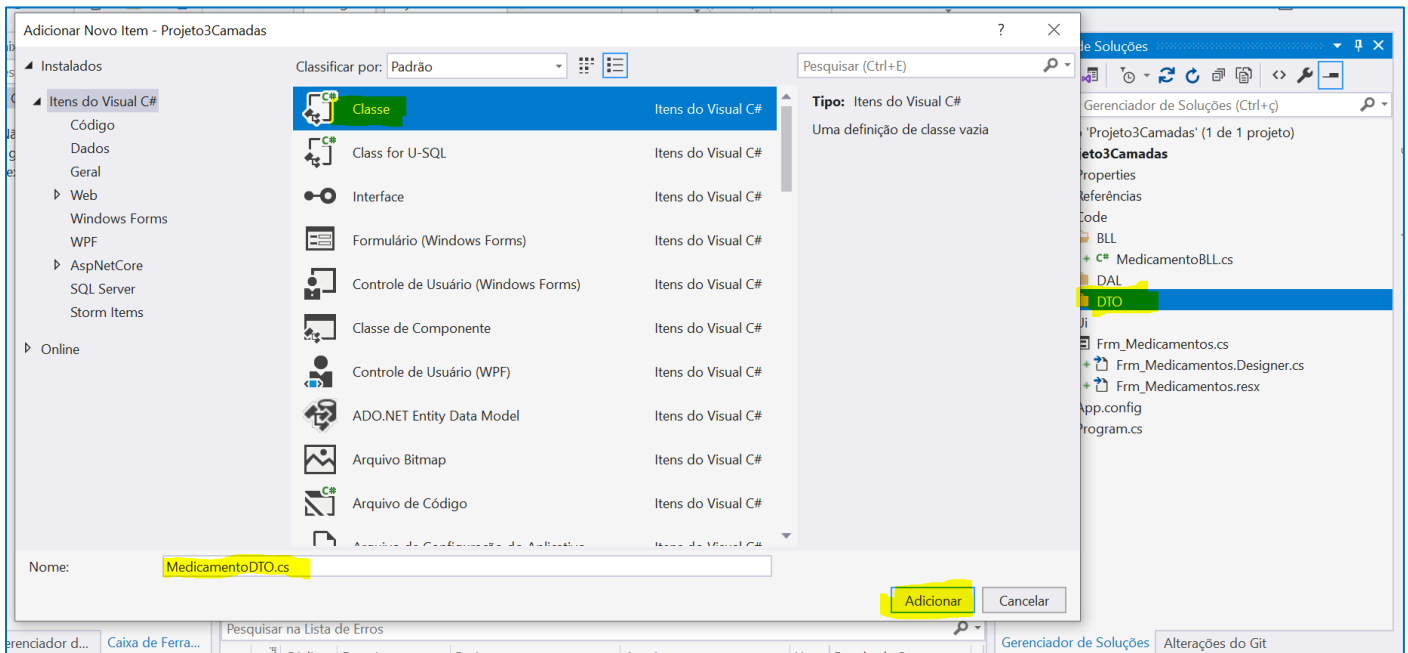
□ A classe de Medicamento que normalmente criávamos em um arquivo somente agora vai ser dividida em dois arquivos. Criaremos duas classes:

- BLL: Aqui ficarão os métodos
- DTO: Aqui ficarão as propriedades

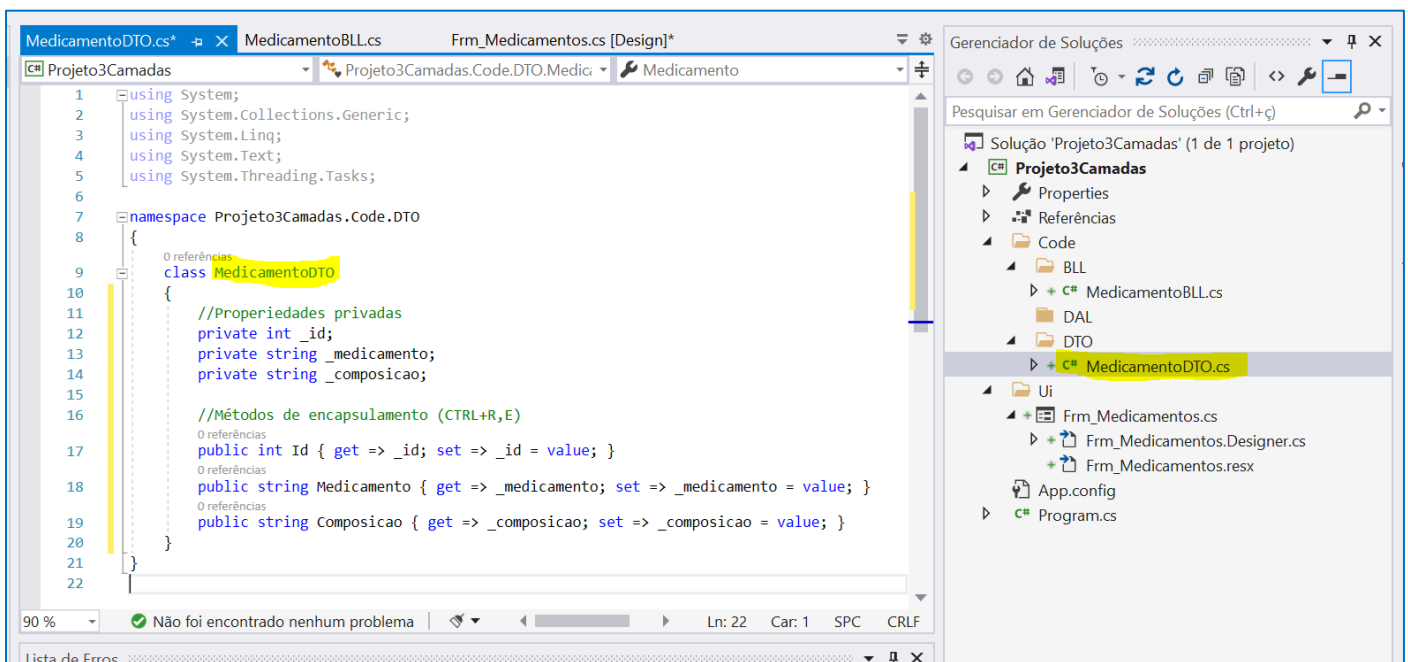
□ Para criar a classe MedicamentoBLL clique com o botão direito na pasta BLL/Adicionar/Classe e dê o nome conforme abaixo:



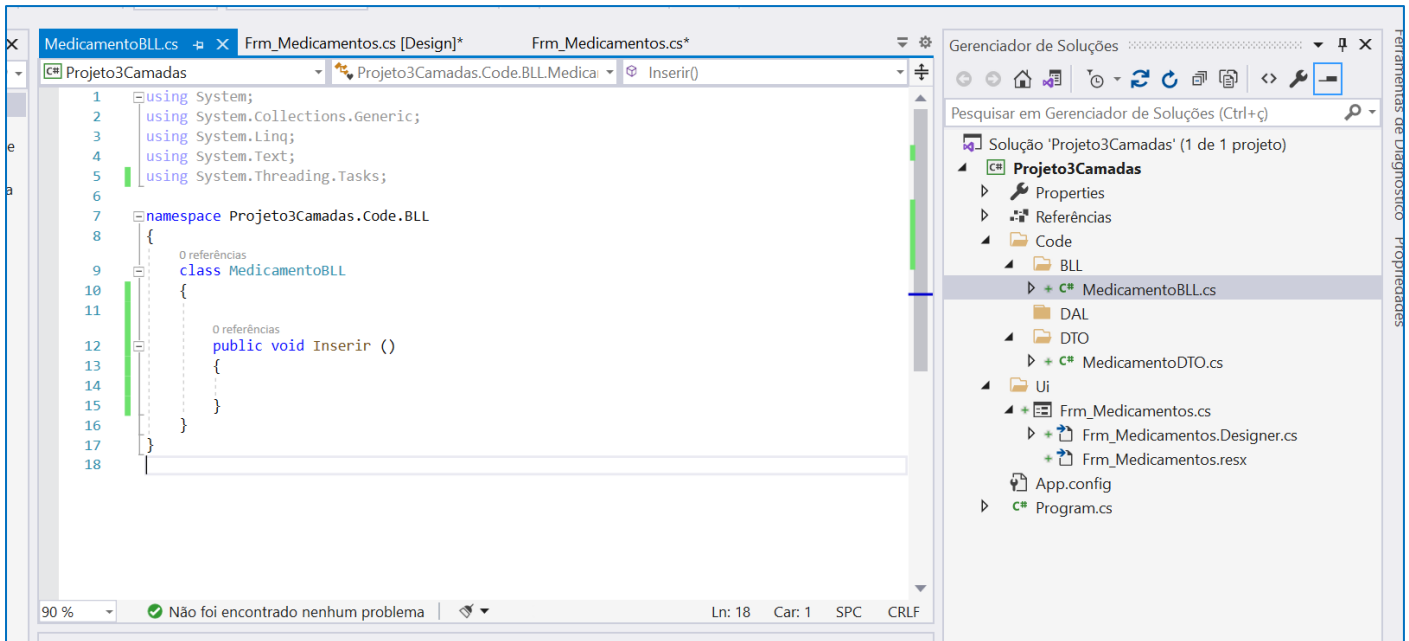
- Para criar a classe MedicamentoDTO clique com o botão direito na pasta DTO/Adicionar/Classe e dê o nome conforme abaixo:



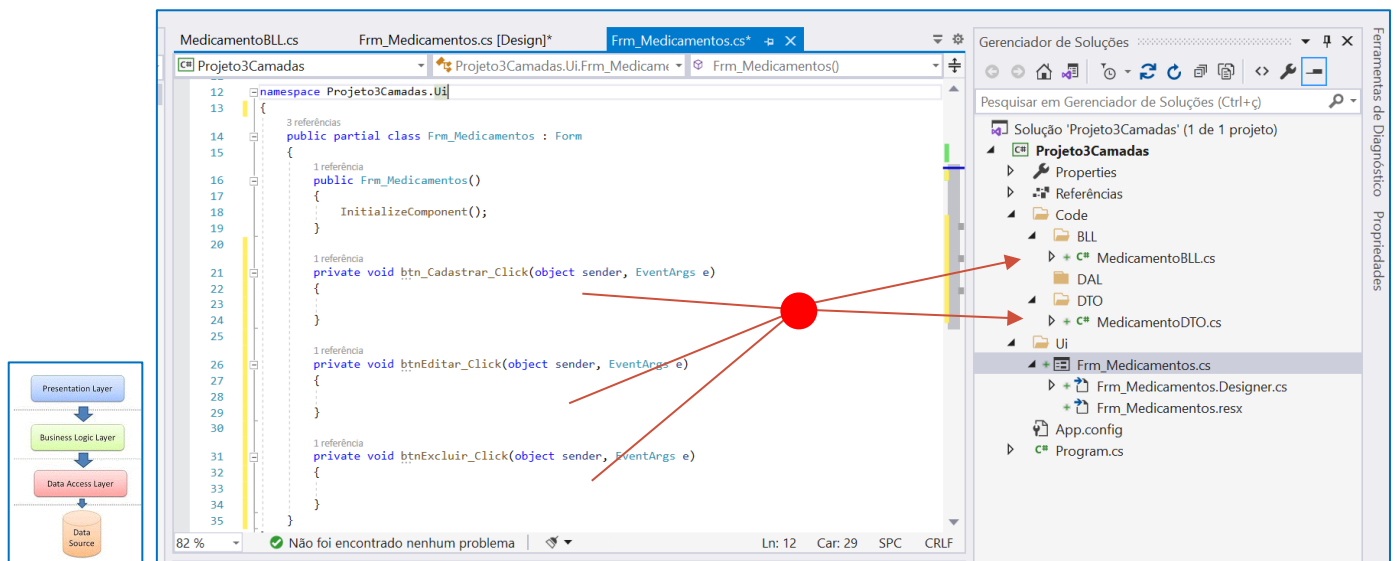
- Crie as propriedades que serão os "Dados Trafegados do Objeto", ou seja, DTO:
 - Digite somente as propriedades privadas e utilize o atalho CTRL+R,E para gerar o encapsulamento de forma automática.



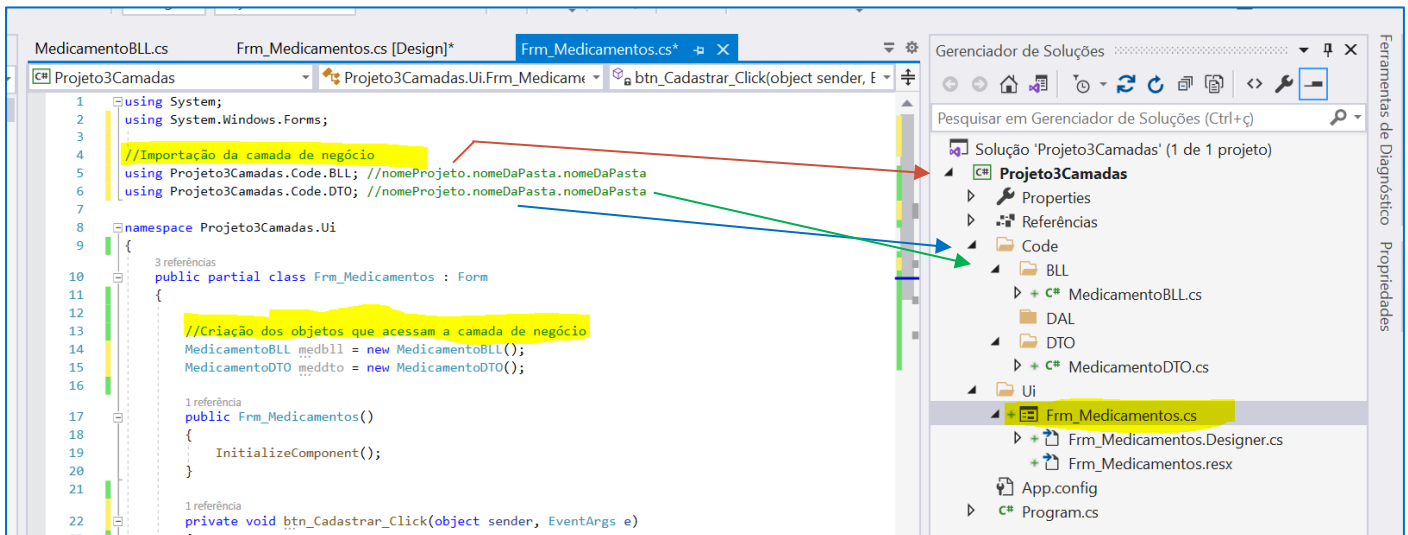
- Crie o primeiro método que iremos programar, o Inserir:



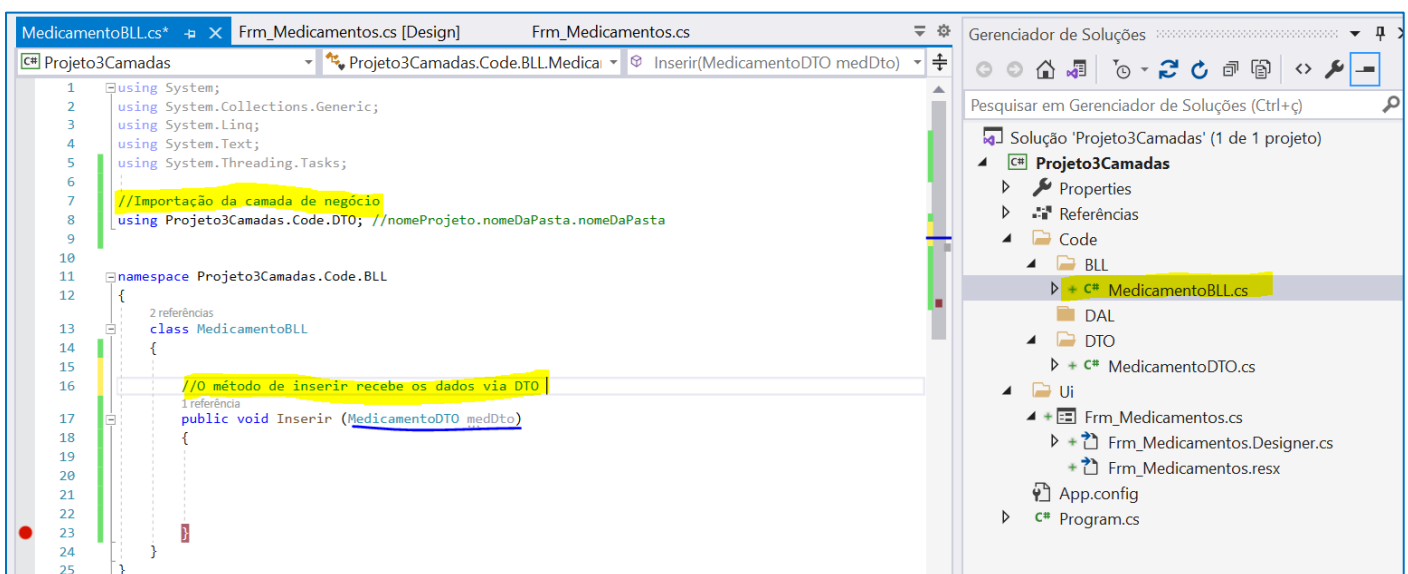
- A camada de apresentação é composta pelos formulário que está na pasta `Ui` e sabemos que será necessário programar os eventos destes formulário para que possamos enviar os dados à camada de negócio. Veja:



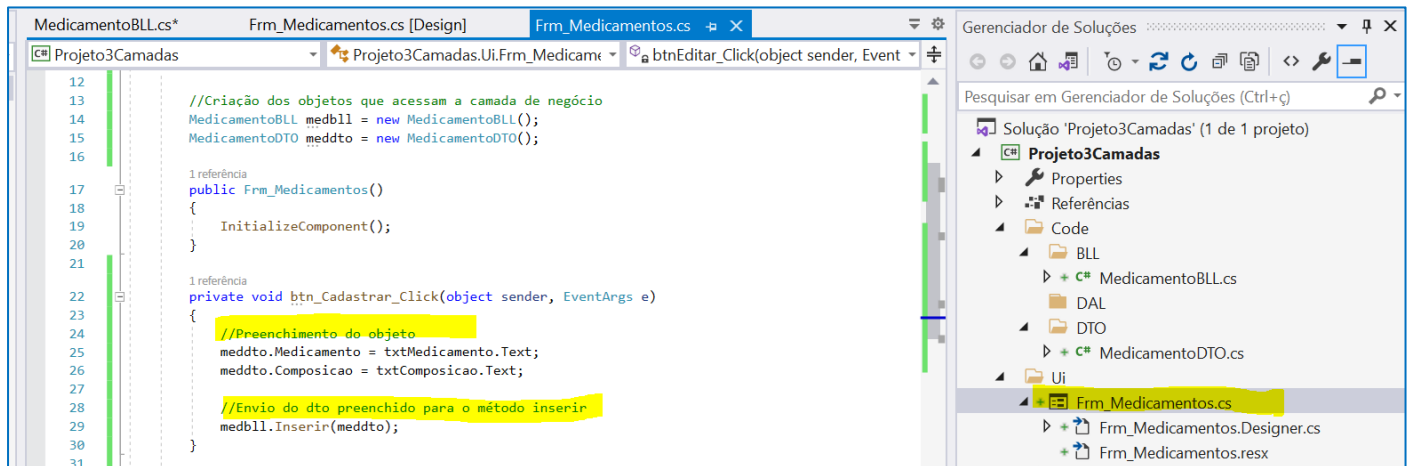
- Portanto será necessário criar dois objetos, sendo um para acessar MedicamentoBLL e outro para acessar MedicamentoDTO:



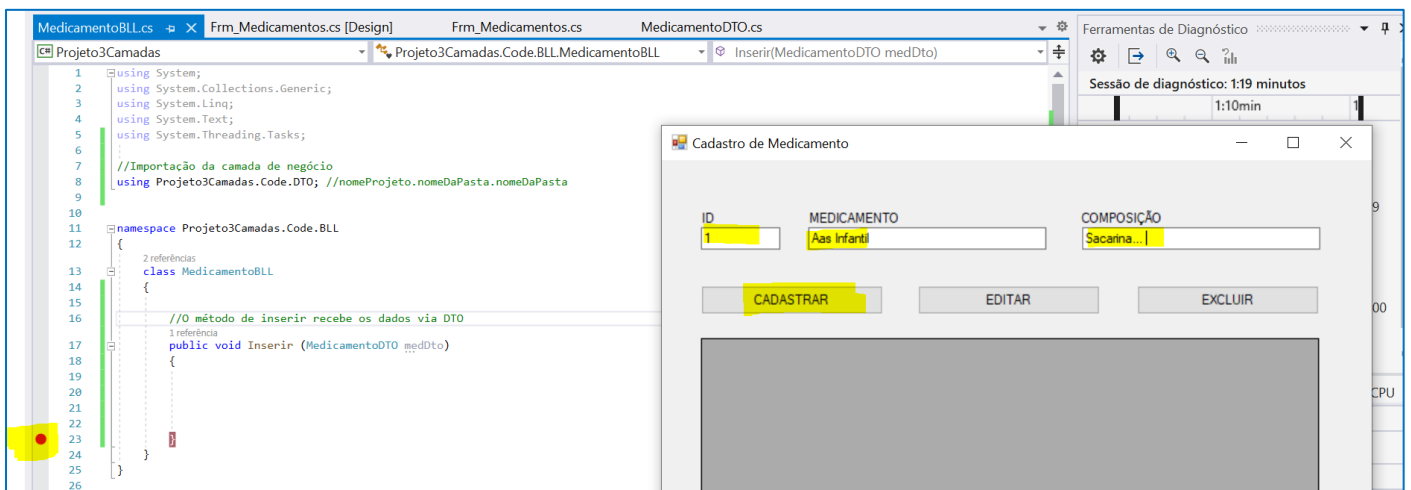
- A partir da importação e da criação do objeto os passos serão preencher o DTO e chamar o método Inserir. Porém, lembre-se que uma camada não conhece enxergar a outra então deveremos ajustar a classe BLL para que ela reconheça e receba os dados da DTO:



- camada de apresentação, Frm_Medicamentos preencha o objeto e envie ele pelo método Inserir:



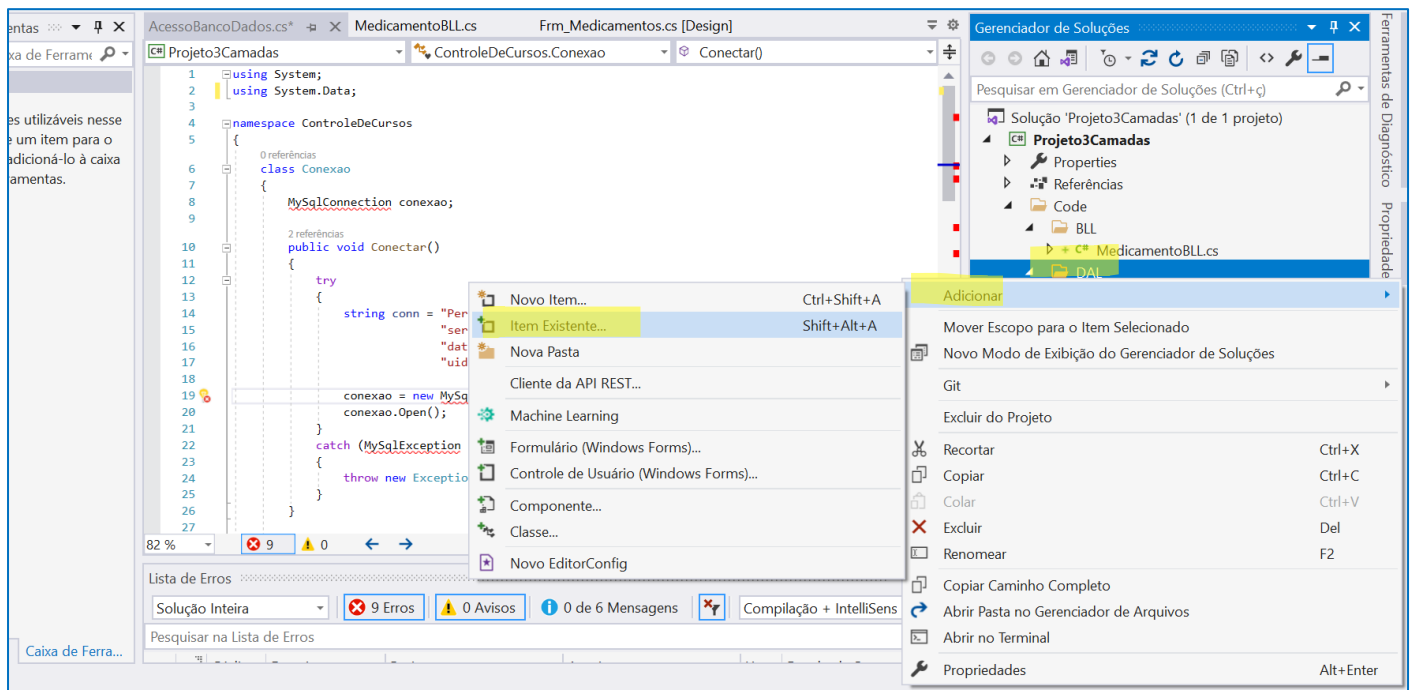
- Confirme se deu certo. Adicione um breakpoint e execute:



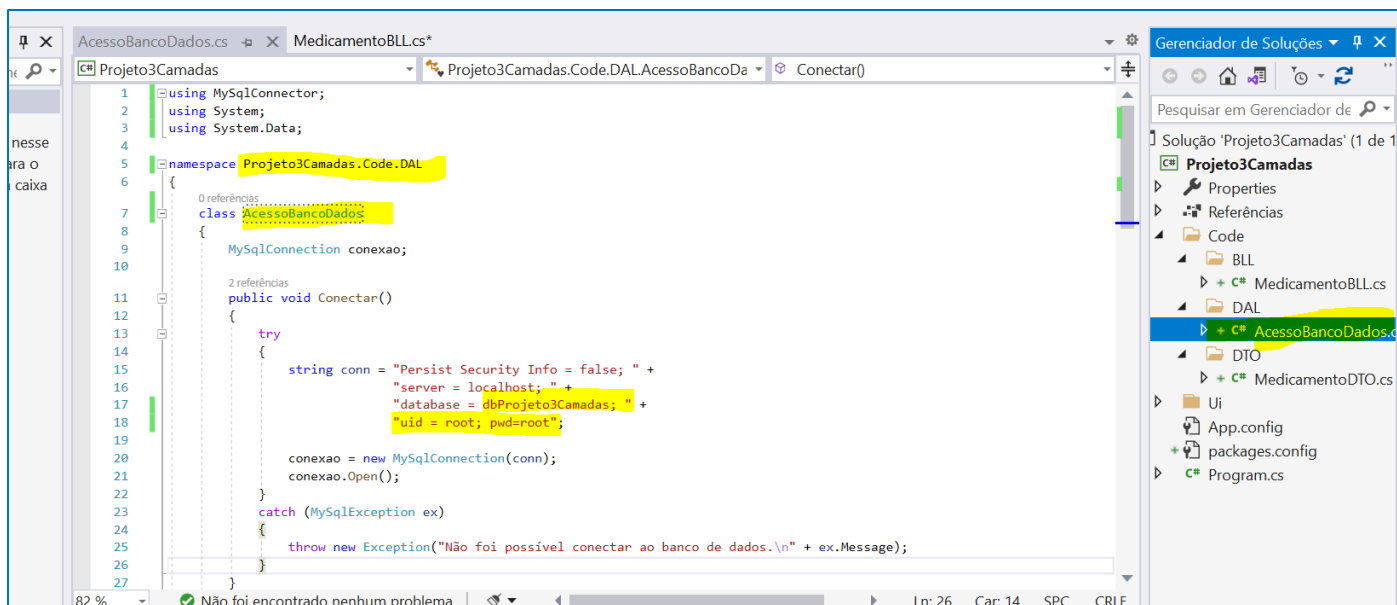
- Revisando, fizemos a camada de Apresentação conversar com a camada de Negócio através da importação. Toda vez que precisar que um arquivo enxergue o outro basta utilizar a importação e indicar as pastas com o caminho correto.

Continuação...

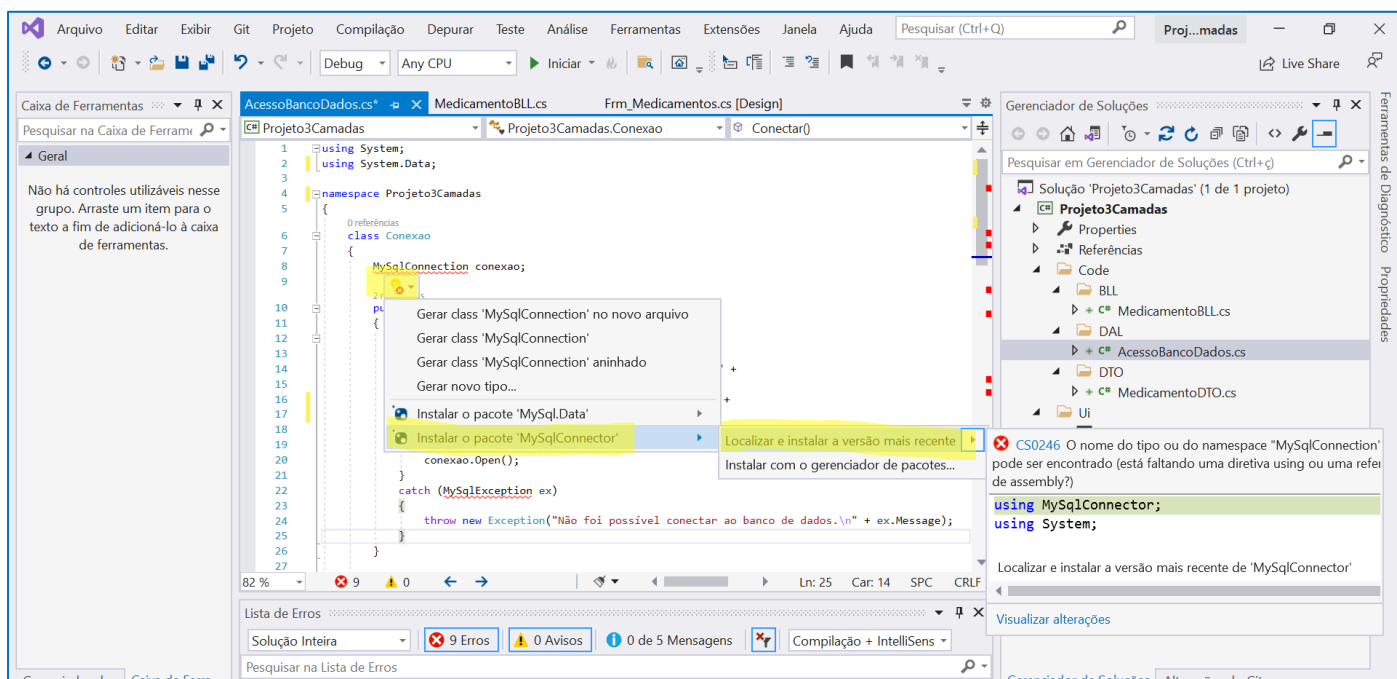
- O próximo passo consiste em fazer a camada de negócio acessar a camada de Acesso aos Dados. Utilizei exatamente a mesma classe de conexão que vocês já conhecem. Basta adicionar um item existente e utilizar o arquivo que disponibilizei.



- Fique atento para realizar as alterações, pois são projetos diferentes:
 - Namespace é o nomeDoProjeto.nomeDaPasta.NomeDaPasta
 - Nome do banco de dados utilizado
 - Usuário e senha
 - E, troquei o nome do arquivo para AcessoBancoDados
 - Ps: No meu não aparece, mas está cheio de erros de MySQL



- Além disso para que os erros saiam da tela falta instalar as referências/dependências de MySQL. Encoste o mouse em MySqlConnection e instale conforme abaixo:



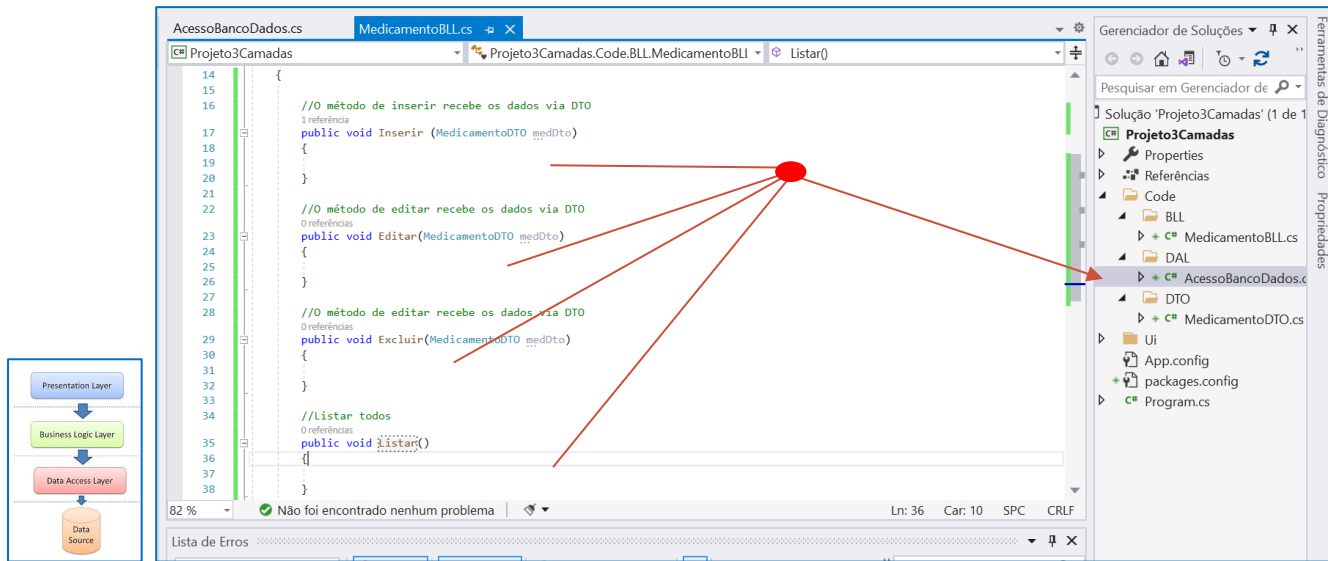
- Todos os erros saem:

AcessoBancoDados.cs* MedicamentoBLL.cs

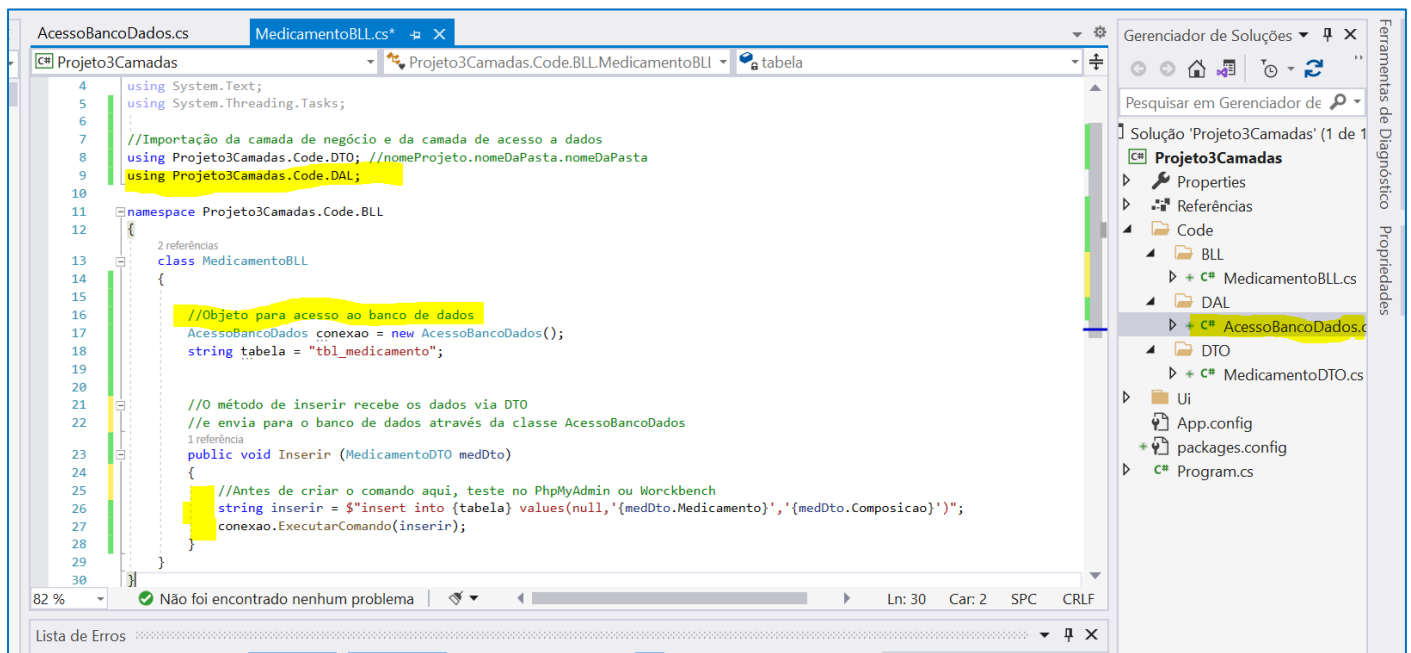
Projeto3Camadas Projeto3Camadas.Code.DAL.Conexao Conectar()

```
1 using MySqlConnection;
2 using System;
3 using System.Data;
4
5 namespace Projeto3Camadas.Code.DAL
6 {
7     0 referências
8     class Conexao
9     {
10         MySqlConnection conexao;
11
12         2 referências
13         public void Conectar()
14         {
15             try
16             {
17                 string conn = "Persist Security Info = false; " +
18                     "server = localhost; " +
19                     "database = dbProjeto3Camadas; " +
20                     "uid = root; pwd=root";
21
22                 conexao = new MySqlConnection(conn);
23                 conexao.Open();
24             }
25             catch (MySqlException ex)
26             {
27                 throw new Exception("Não foi possível conectar ao banco de dados.\n" + ex.Message);
28             }
29         }
30     }
31 }
```

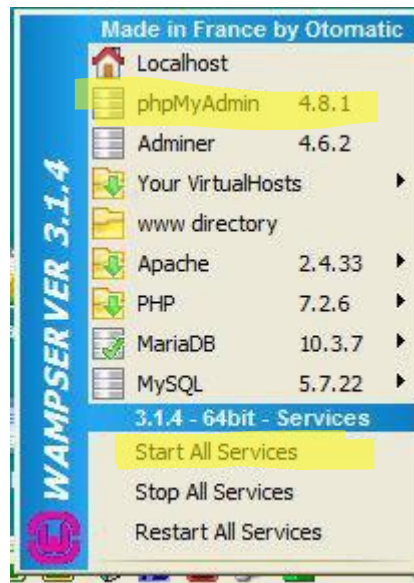
- Vamos criar em Medicamento BLL o acesso ao banco de dados:



- Veja o método Insert:



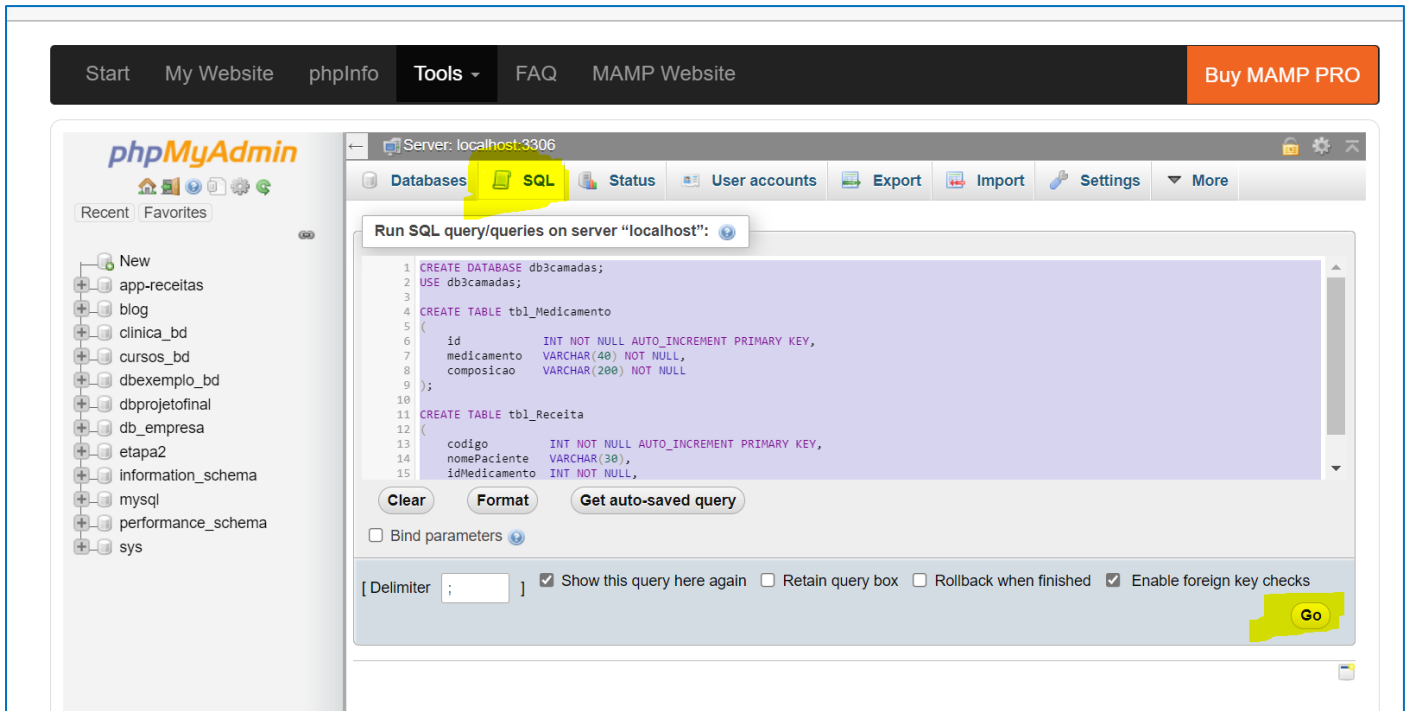
- Neste ponto já criamos o comando em Sql e ele foi enviado para a classe de acesso a dados. Vamos criar o banco de dados usando o PhpMyAdmin. Para isso inicie todos os serviços e clique em phpMyAdmin



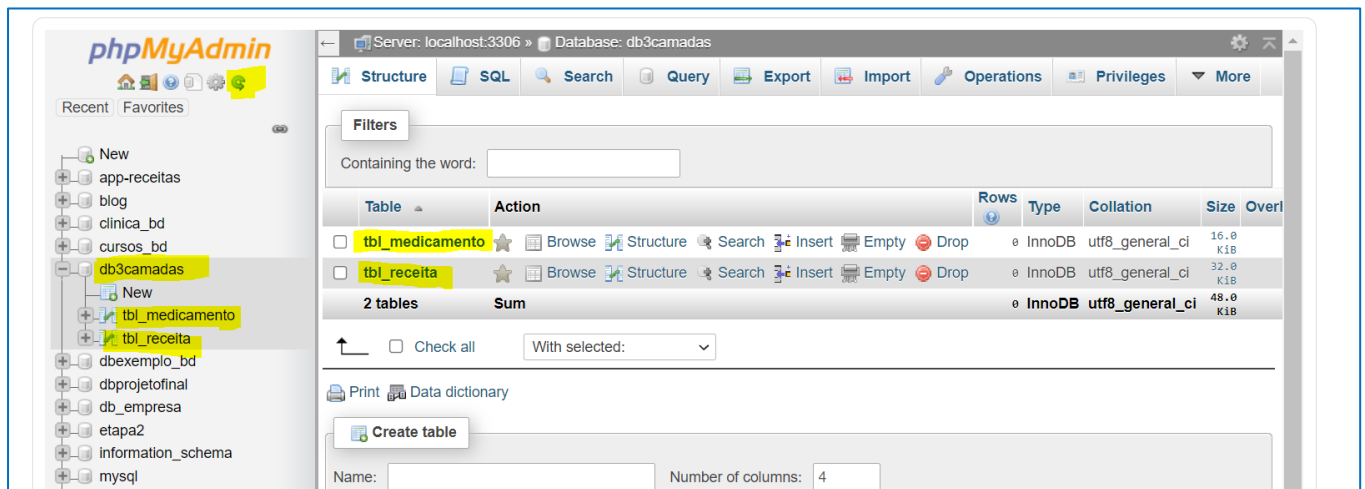
- No phpMyAdmin existe a possibilidade de criar o banco via design ou via código, quero que vocês criem via código para poder disponibilizar o script. Abra um bloco de notas ou o NotePad++ e monte um script para criar o banco e as tabelas. Salve, pois ele será disponibilizado junto ao projeto.

```
Agenda.txt x scriptDB.sql x
1 CREATE DATABASE db3camadas;
2 USE db3camadas;
3
4 CREATE TABLE tbl_Medicamento
5 (
6     id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
7     medicamento VARCHAR(40) NOT NULL,
8     composicao VARCHAR(200) NOT NULL
9 );
10
11 CREATE TABLE tbl_Receita
12 (
13     codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
14     nomePaciente VARCHAR(30),
15     idMedicamento INT NOT NULL,
16     FOREIGN KEY(idMedicamento) REFERENCES tbl_Medicamento(id)
17 );
```

- Execute no phpMyAdmin



- Confira se a execução foi correta:

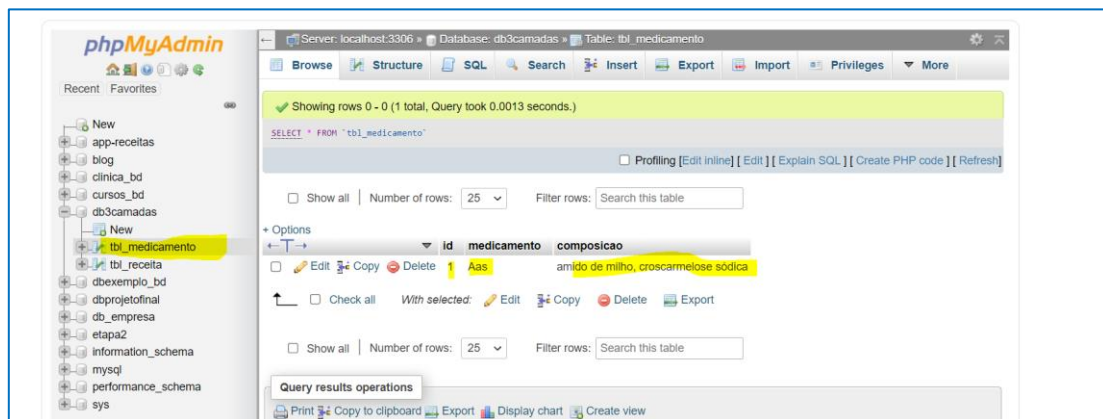
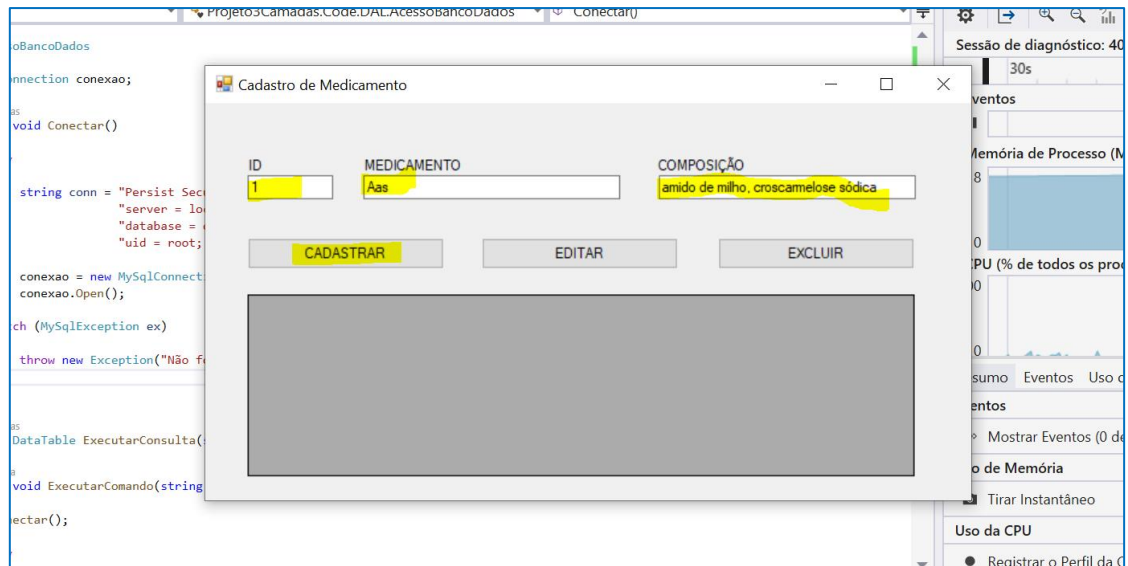


- Ok, agora que temos o banco de dados vamos conferir os dados de conexão e executar o projeto no C#.

```
try
{
    string conn = "Persist Security Info = false; " +
        "server = localhost; " +
        "database = db3camadas; " +
        "uid = root; pwd=root";

    conexao = new MySqlConnection(conn);
    conexao.Open();
}
catch (MySqlException ex)
```

□ Execute o projeto e teste a inserção:



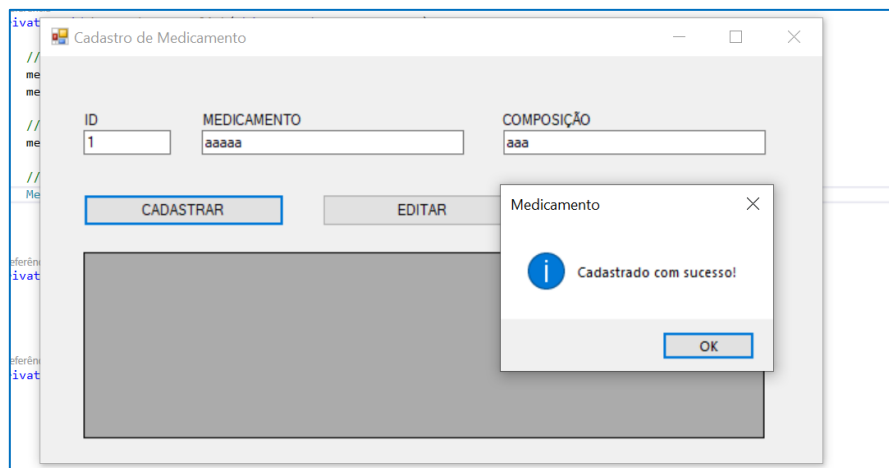
□ Emita mensagem de sucesso ao cadastrar.

```
1 referencia
private void btn_Cadastrar_Click(object sender, EventArgs e)
{
    //Preenchimento do objeto
    meddto.Medicamento = txtMedicamento.Text;
    meddto.Composicao = txtComposicao.Text;

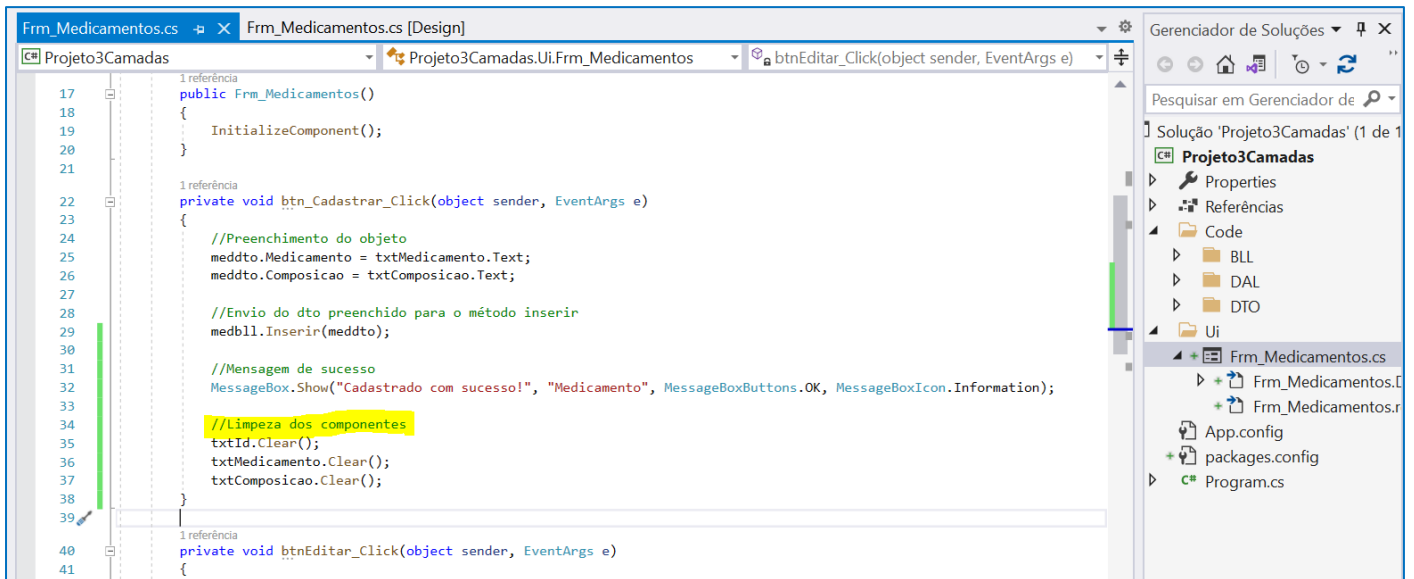
    //Envio do dto preenchido para o método inserir
    medbll.Inserir(meddto);

    //Mensagem de sucesso
    MessageBox.Show("Cadastrado com sucesso!", "Medicamento", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

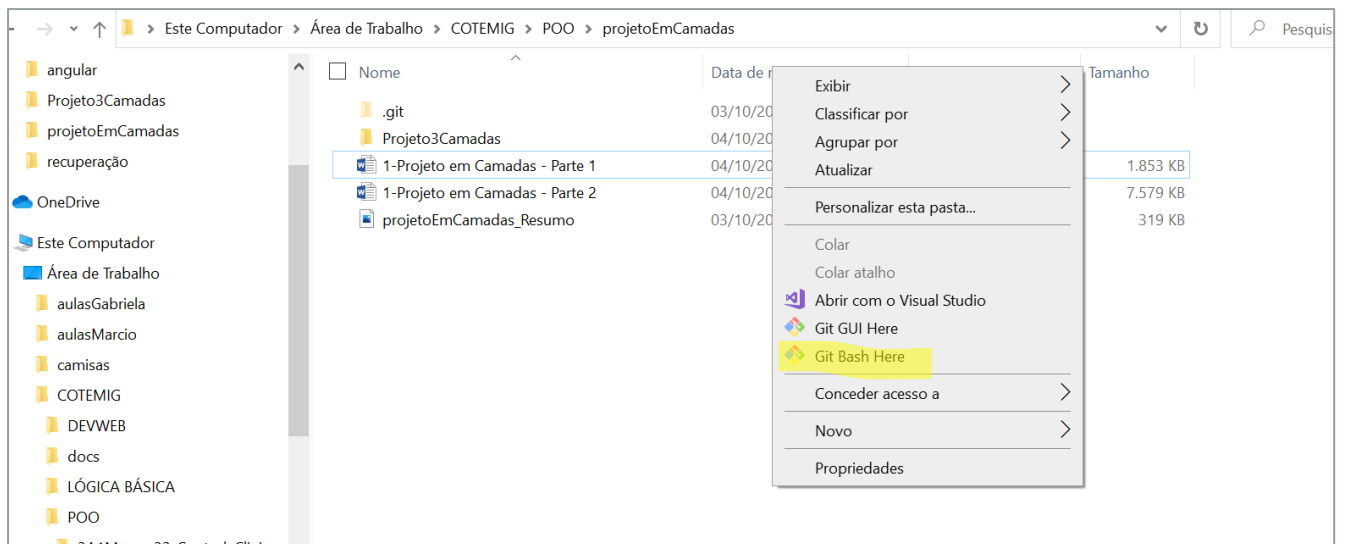


- Limpe os componentes de tela:



ENVIO PARA O GITHUB

- Concorda que agora temos o projeto e muitos novos arquivos? Então vamos enviar para o GitHub:
- Clique com o botão direito na pasta do projeto e abra o Git Bash Here



- ❑ Adicione todos os arquivos que deverão ser enviados:
 - `git add .`

```
MINGW64:/c/Users/Michelle Belli/Desktop/COTEMIG/POO/projetoEmCamadas
Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git add .
warning: LF will be replaced by CRLF in Projeto3Camadas/packages/MySQLConnector.1.3.13/README.md.
The file will have its original line endings in your working directory
Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$
```

- ❑ Faça um novo commit:
 - `git commit -m "Parte 2"`
 - Importante, são muitos arquivos novos criados com o projeto, é isso mesmo. Está certo!

```
MINGW64:/c/Users/Michelle Belli/Desktop/COTEMIG/POO/projetoEmCamadas
Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git add .
warning: LF will be replaced by CRLF in Projeto3Camadas/packages/MySQLConnector.1.3.13/README.md.
The file will have its original line endings in your working directory
Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git commit -m "Parte 2"
[master 6886ffc] Parte 2
177 files changed, 80765 insertions(+)
create mode 100644 1-Projeto em Camadas - Parte 1.docx
create mode 100644 1-Projeto em Camadas - Parte 2.docx
create mode 100644 Projeto3Camadas/.vs/Projeto3Camadas/v16/.suo
create mode 100644 Projeto3Camadas/Projeto3Camadas.sln
create mode 100644 Projeto3Camadas/Projeto3Camadas/App.config
create mode 100644 Projeto3Camadas/Projeto3Camadas/Code/BLL/MedicamentoBLL.cs
create mode 100644 Projeto3Camadas/Projeto3Camadas/Code/DAL/AcessoBancoDados.cs
create mode 100644 Projeto3Camadas/Projeto3Camadas/Code/DTO/MedicamentoDTO.cs
create mode 100644 Projeto3Camadas/Projeto3Camadas/Program.cs
create mode 100644 Projeto3Camadas/Projeto3Camadas/Projeto3Camadas.csproj
create mode 100644 Projeto3Camadas/Projeto3Camadas/Properties/AssemblyInfo.cs
```

- ❑ Configure seu e-mail e senha:
 - `git config --global user.email michellebelli@cotemig.com`
 - `git config --global user.name "Michelle Belli"`

```
MINGW64:/c/Users/Michelle Belli/Desktop/COTEMIG/POO/projetoEmCamadas
.3.0/dotnet_library_license.txt
create mode 100644 Projeto3Camadas/packages/System.Threading.Tasks.Extensions.4
.3.0/lib/netstandard1.0/System.Threading.Tasks.Extensions.dll
create mode 100644 Projeto3Camadas/packages/System.Threading.Tasks.Extensions.4
.3.0/lib/netstandard1.0/System.Threading.Tasks.Extensions.xml
create mode 100644 Projeto3Camadas/packages/System.Threading.Tasks.Extensions.4
.3.0/lib/portable-net45+win8+wp8+wpa81/System.Threading.Tasks.Extensions.dll
create mode 100644 Projeto3Camadas/packages/System.Threading.Tasks.Extensions.4
.3.0/lib/portable-net45+win8+wp8+wpa81/System.Threading.Tasks.Extensions.xml
create mode 100644 Projeto3Camadas/scriptDB.sql
create mode 100644 ~$Projeto em Camadas - Parte 2.docx
create mode 100644 ~WRL0761.tmp

Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git config --global user.email michellebelli@cotemig.com

Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git config --global user.nome "Michelle Belli"

Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ |
```

- ☐ Envie os arquivos para o repositório já configurado na etapa 1
 - `git push -u origin master`

```
MINGW64:/c/Users/Michelle Belli/Desktop/COTEMIG/POO/projetoEmCamadas

Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ git push -u origin master
Enumerating objects: 157, done.
Counting objects: 100% (157/157), done.
Delta compression using up to 4 threads
Compressing objects: 100% (142/142), done.
Writing objects: 100% (156/156), 14.75 MiB | 1.73 MiB/s, done.
Total 156 (delta 35), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (35/35), done.
To https://github.com/michellebellicotemig/projetoEmCamadas.git
 c88bc67..6886ffc master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Michelle Belli@DESKTOP-6J55SQD MINGW64 ~/Desktop/COTEMIG/POO/projetoEmCamadas (master)
$ |
```

- ☐ Pronto!! Só um comentário, na minha maquina pessoal só preciso executar 3 comandos:
 - `git add .`
 - `git commit -m "Texto explicativo do comit"`
 - `git push -u origin máster`

DISTRIBUIÇÃO DE PONTOS (Total= 2pts)

- ☐ Faça seu projeto de acordo com o tema, não aceitarei entregas com outros temas.
- ☐ Ao abrir o link do GitHub é possível acessar o projeto completo? (0.25)
- ☐ A estrutura de pastas para 3 camadas foi respeitada? (0.25)
- ☐ O formulário apresenta criatividade, nomes, elementos funcionando? (0.25)
- ☐ A classe DTO possui propriedades e encapsulamento adequados? (0.25)
- ☐ A classe BLL possui importação, conexão e método inserir adequados? (0.5)
- ☐ Entregou no prazo? (0.5)