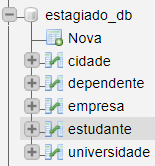
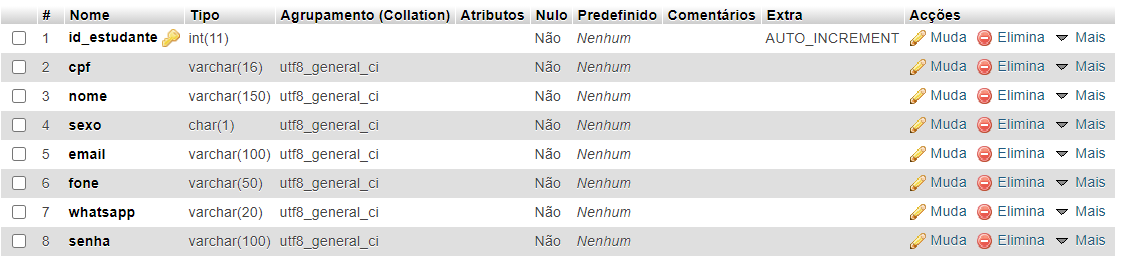
**Checklist e Tut do Projeto Interdisciplinar**

Baseado nas aulas do Professor Caio Malheiros disponíveis em: <https://www.youtube.com/watch?v=vbeesW_sAp8&list=PL69eBMqNvxhWSaTveqjBMuKqGSqL6L6_A>

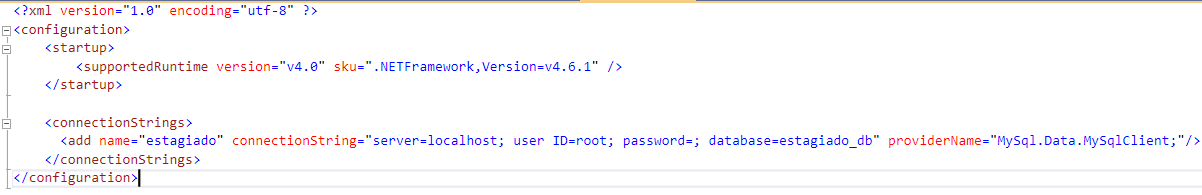
* Setup recomendado do Visual Studio 2019
  + ASP.NET Core desenvolvimento Web
  + Desenvolvimento para Azure
  + **Desenvolvimento para desktop com .NET (obrigatório pra esse projeto)**
  + Desenvolvimento com a Plataforma Universal do Windows
  + Processamento e armazenamento de dados
  + Desenvolvimento de extensão do Visual Studio
  + Desenvolvimento multiplataforma com .NET Core
* Criando o projeto
  + File > New > Project... > Windows Forms App (.NET Framework)
  + Dê um nome e pasta pro projeto
  + Selecione .NET Framework 4.6.1
* Criando e preparando conexão ao banco MySQL
  + Instale o XAMP
  + Inicialize o servidor Apache e o MySQL no painel do XAMP
  + Acesse no navegador <http://localhost/phpmyadmin/>
  + Crie o banco de dados: click em Base de Dados > dê um nome para a base de dados > selecione utf8\_general\_ci > click em Criar
    - Eu chamei nosso banco de estagiado\_db



* + Crie as tabelas: click em Nova > dê um nome para a tabela > selecione o número de colunas > click em executar > dê o nome dos campos, indique os tipos e tamanhos > click em Guarda. Exemplo de tabela criada, a tabela estudantes:



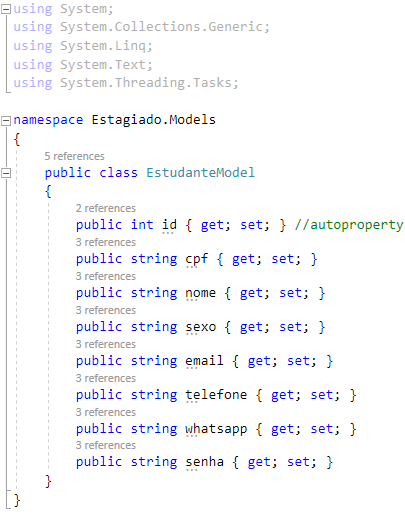
* + Prepare a string de conexão do banco de dados: abra o arquivo App.config do seu projeto > adicione dentro da tag configuration a string de conexão, que deve estar dentro da tag connectionStrings. No meu ficou assim:



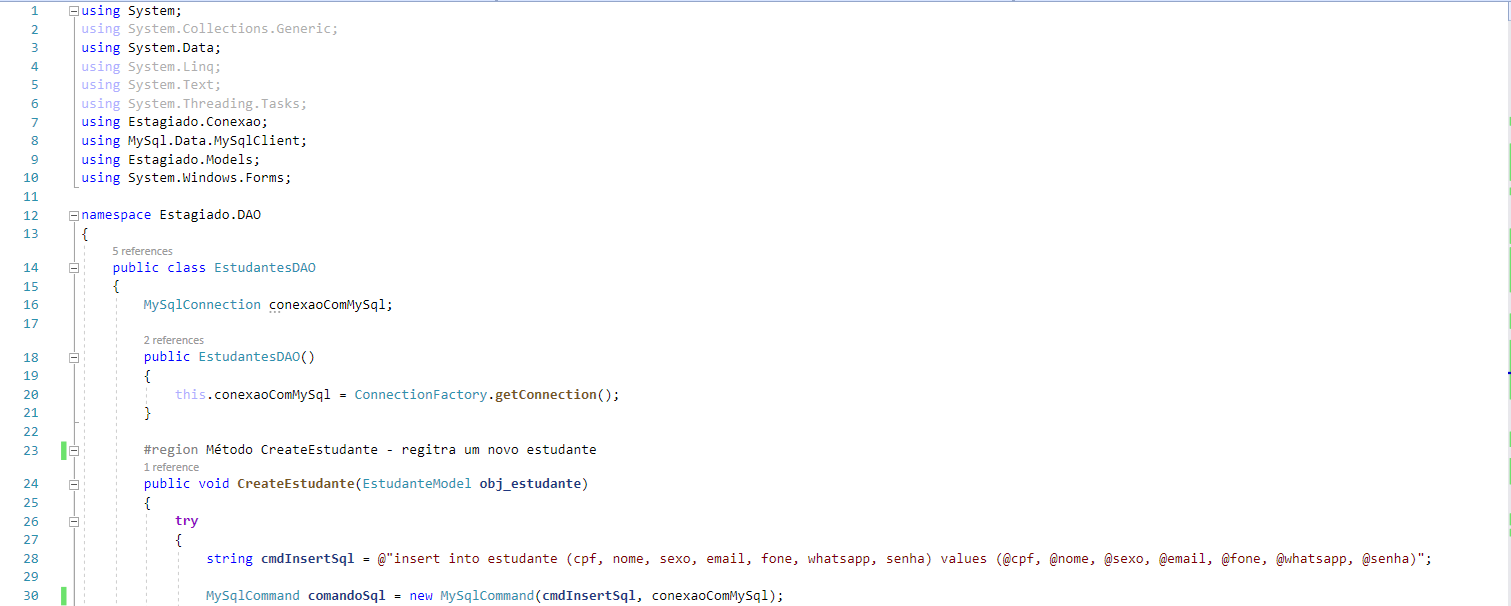
* + - Caso o nome do seu usuário não seja root, você deve mudar isso em user ID. Caso você tenha uma senha de acesso ao banco, você deve indicar isso em password. Em database você coloca o nome do seu banco de dados.
* Crie as pastas para organizar o sistema
  + Crie a pasta Conexao para a classe de conexão ao banco de dados: botão esquerdo no projeto > Add > New Folder > dê o nome Conexao
  + Crie a pasta DAO para as classes de acesso a dados: botão esquerdo no projeto > Add > New Folder > dê o nome DAO
    - Nota: DAO = Data Access Object
  + Crie a pasta Models para as classes do nosso modelo de negócio: botão esquerdo no projeto > Add > New Folder > dê o nome Models
  + Crie a pasta Views para as classes de interação com o usuário: botão esquerdo no projeto > Add > New Folder > dê o nome Views
* Crie uma classe de acesso ao banco de dados
  + Crie uma classe para acessar o banco: botão esquerdo na pasta Conexao > Add > Class... > dê um nome para a classe (eu chamei de ConnectionFactory.cs)
  + Implemente o método da conexão getConnection():

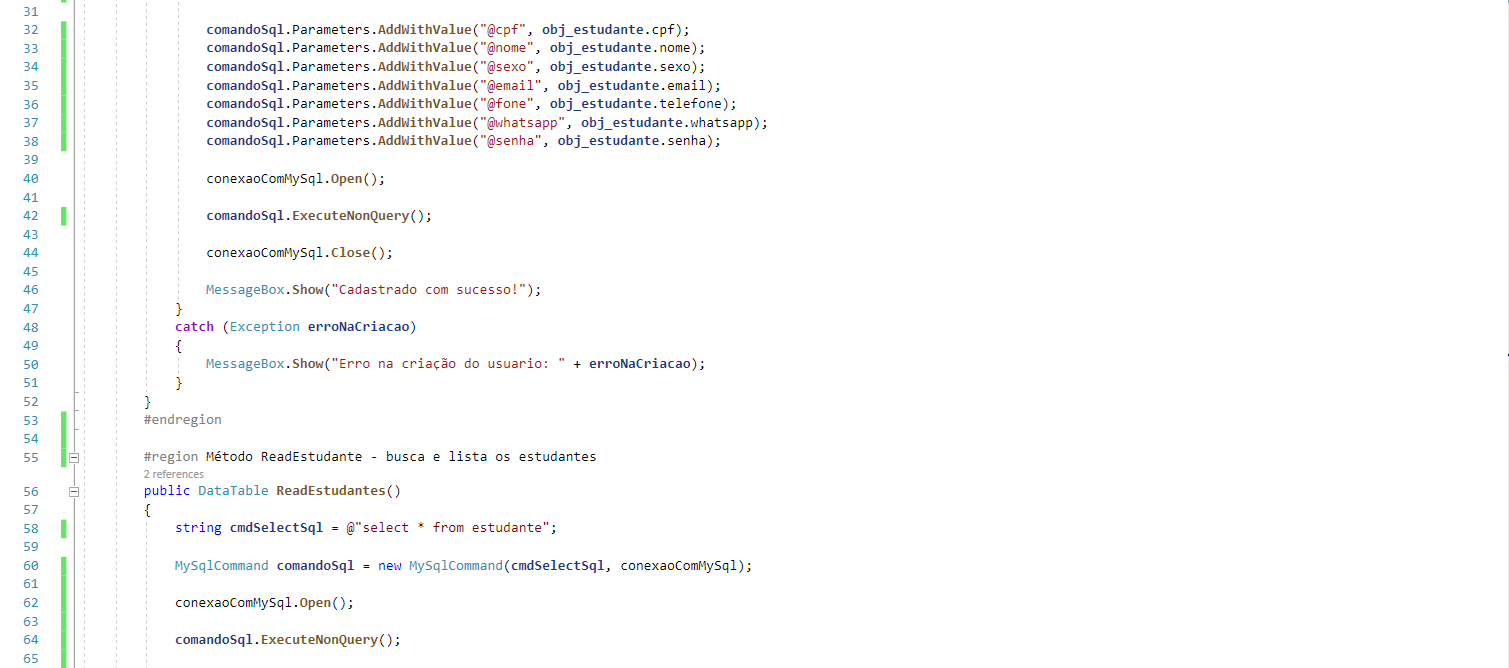


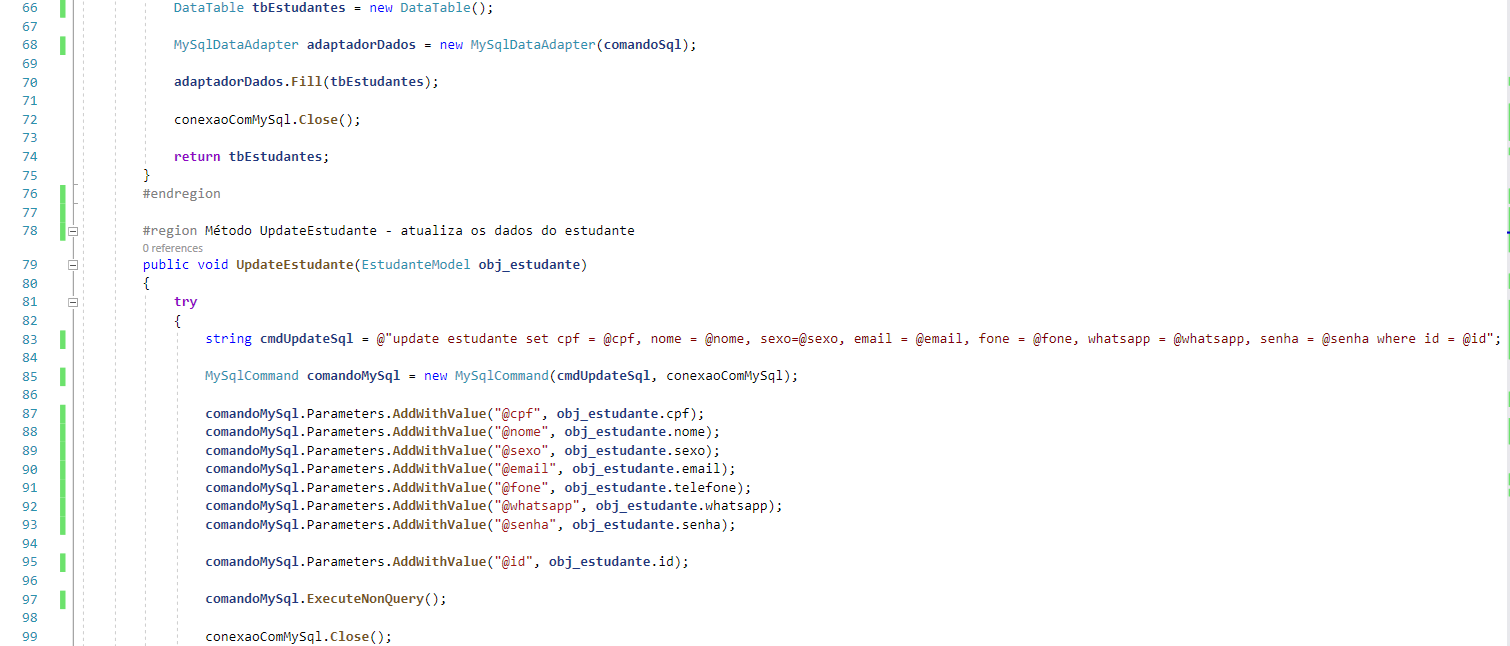
* Classe para representar uma entidade do nosso negócio (EstudanteModel)
  + Criar a classe para representar estudantes: botão esquerdo na pasta Models > Add > Class... > dê um nome para a classe (eu chamei de EstudanteModel.cs)
  + Implemente a classe EstudanteModel:



* + - Nota: eu usei a autoproperty do C#, mas vocês podem fazer do modo convencional. É melhor para aprender.
* Classe para acesso e manipulação de dados da entidade estudante (CRUD = Create, Read, Update and Delete)
  + Criar a classe para fazer o CRUD de estudante: botão esquerdo na pasta DAO > Add > Class... > dê um nome para a classe (eu chamei de EstudantesDAO.cs).
  + Declare um objeto do tipo MySqlConnection. Use ele para acessar o método getConnection() da classe ConnectionFactory
  + Implemente os métodos CreateEstudante(), ReadEstudante(), UpdateEstudante() e DeleteEstudante():



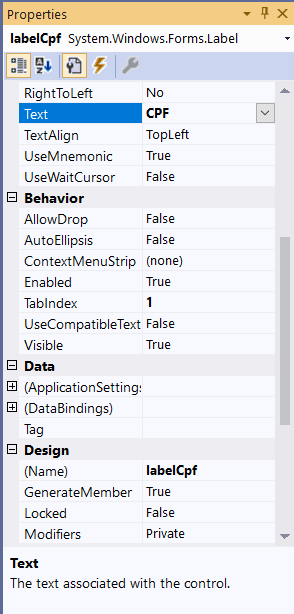




* Formulário de cadastro de estudantes
  + Criar um formulário de cadastro: botão esquerdo na pasta Views > Add > Form (Windows Forms)... > dê um nome para o formulário (eu chamei de FormEstudante)
  + Adicione os campos do formulário: Toolbox > Common Controls > escolha e arraste o que você quer adicionar:
    - Para Nome eu escolhi um Label, pro campo onde vai ser digitado o nome eu escolhi um TextBox, pros demais campos parecidos (CPF, Email, Senha, Telefone, WhatsApp) escolhi também TextBox. Para Sexo eu escolhi um Label e um ComboBox. Adicionei ao lado dos TextBox um DataGridView. Abaixo coloquei um Button e chamei ele de Criar Cadastro
    - Para cada item na tela vocês podem mudar o texto e o nome deles na aba Properties. É só clicar no item que essa aba abre no canto inferior direito. Exemplo:
      * Clica sobre a Label CPF



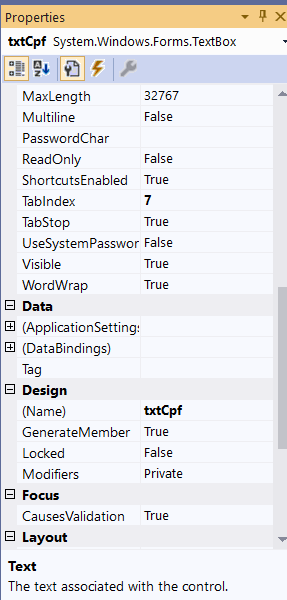
* Abre as Propriedades da Label CPF



* Clica sobre a TexBox do CPF



* Abre as Propriedades da TextBox do CPF

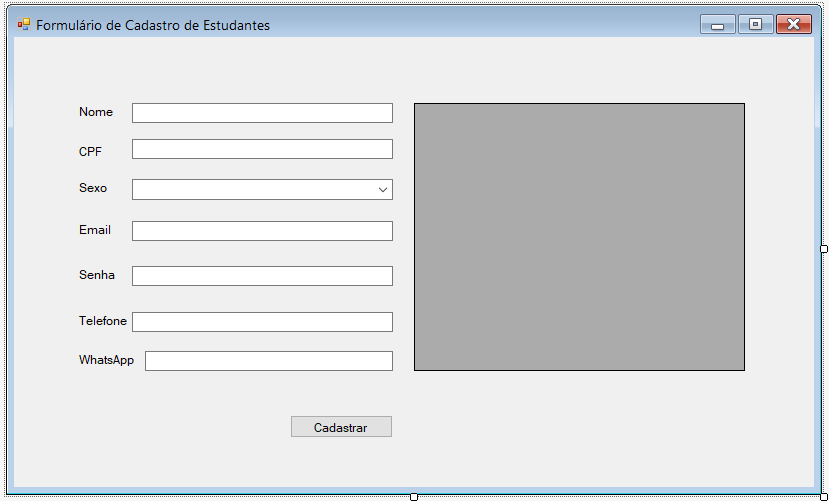


**>Deem um Name único. Esse Name será usado para recuperar os dados digitados nessa área**

Os Names que coloquei nas entradas de dados foram:

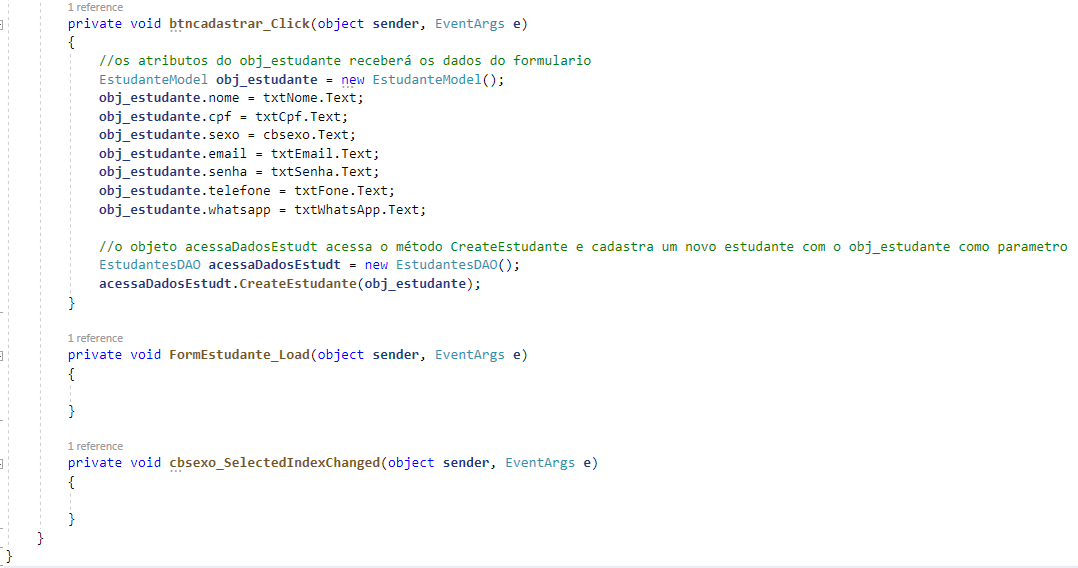
* + TextBox para Nome: txtNome
  + TextBox para CPF: txtCpf
  + ComboBox para Sexo: cbsexo
  + TextBox para Email: txtEmail
  + TextBox para Senha: txtSenha
  + TextBox para Telefone: txtFone
  + TextBox para WhatsApp: txtWhatsApp
  + Button Criar Cadastro: btncadastrar
  + DataGridView: dgvEstudante

O formulário ficou assim:

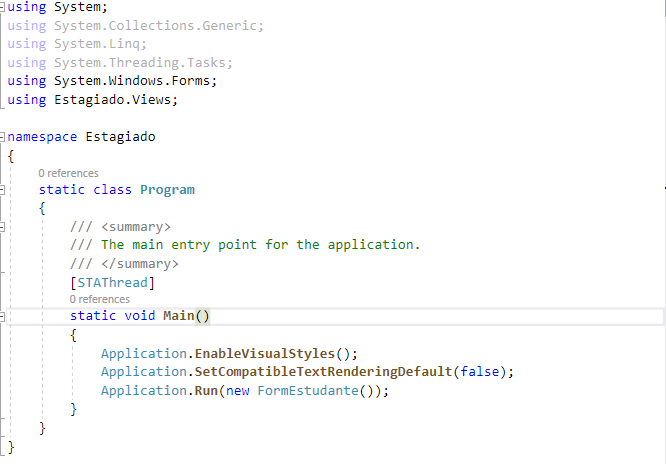


* Recuperar dados do formulário e enviar para o banco de dados
  + Edite a classe do formulário: dois clicks sobre o botão Cadastrar para abrir a classe do formulário (FormEstudante no meu caso). Edite ela da seguinte forma:



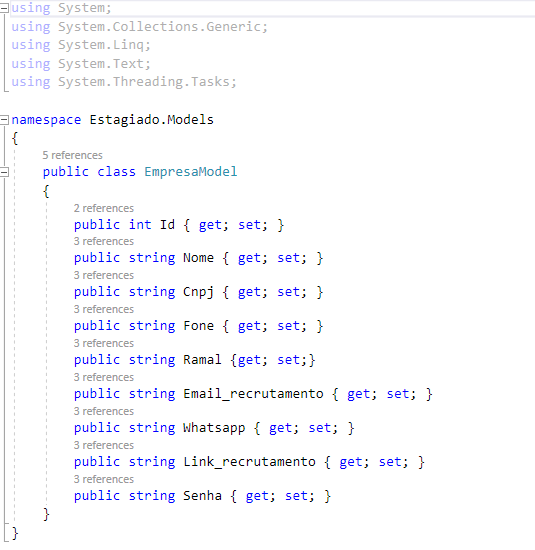


* Mudar o entry point do programa para rodar o formulário FormEstudante
  + Acesse o arquivo Program.cs
  + Modifique-o para ficar assim:

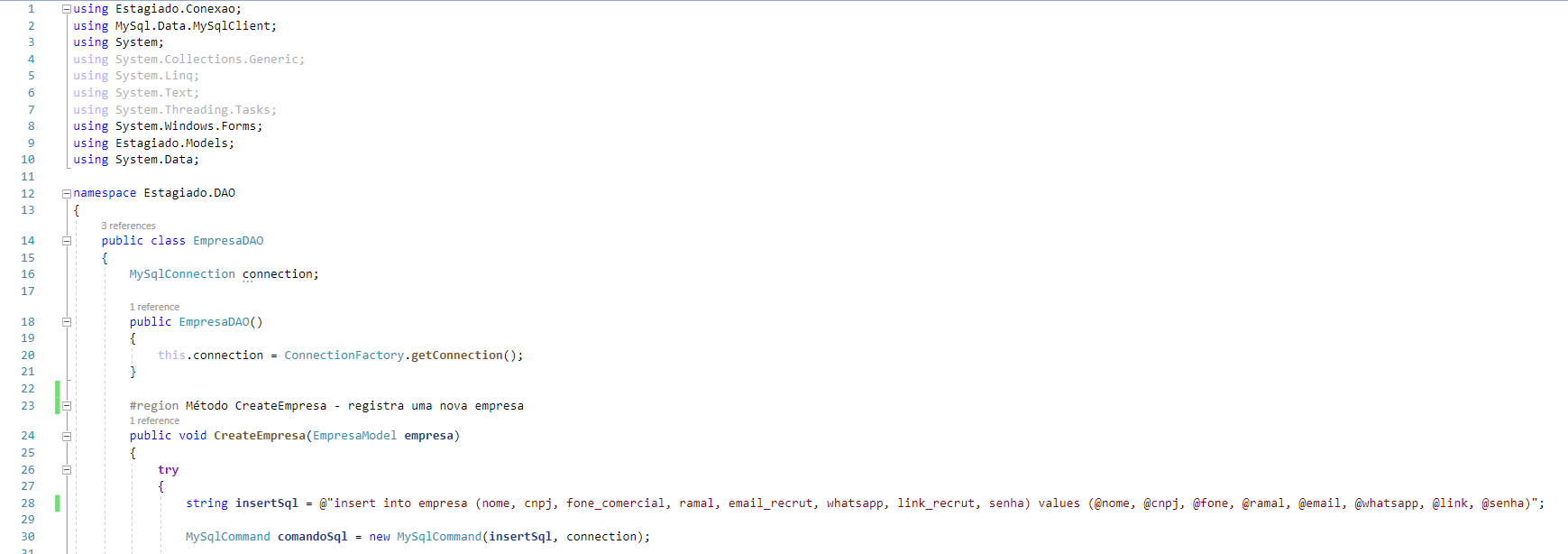


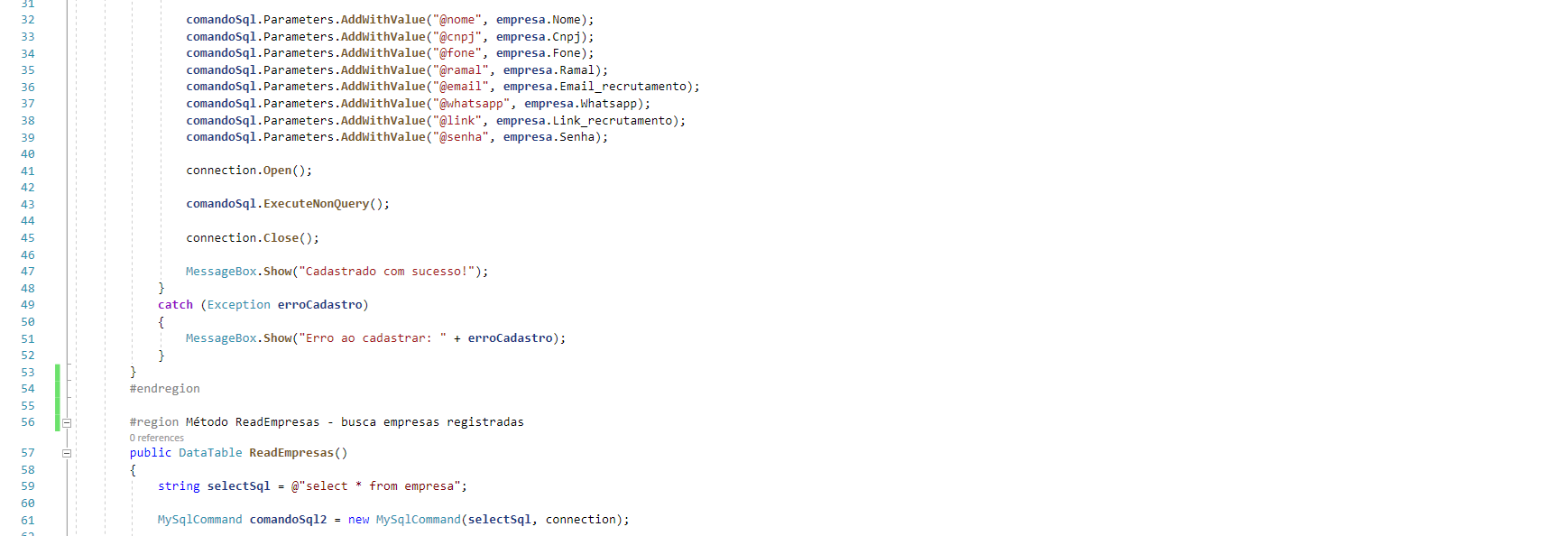
Rode a aplicação e veja se há erro. Caso não haja, preencha o formulário e click em Cadastrar. Veja no [localhost/phpmyadmin/](http://localhost/phpmyadmin/) se os dados foram enviados para o banco. Se sim, a aplicação está funcionando. Caso haja dúvida veja o tutorial do Professor Caio Malheiros em que me baseei: <https://www.youtube.com/watch?v=vbeesW_sAp8&list=PL69eBMqNvxhWSaTveqjBMuKqGSqL6L6_A>

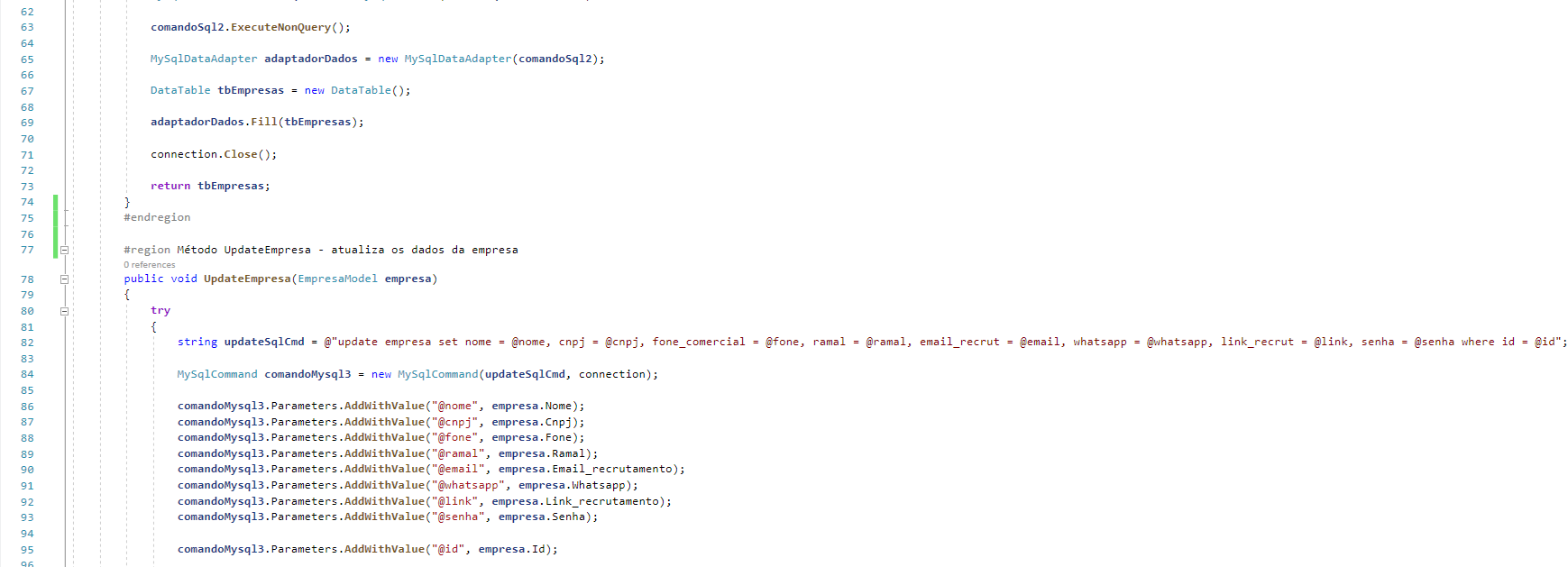
* Outra entidade do nosso negócio (EmpresaModel)
  + Criar a classe EmpresaModel > botão direito na pasta Models > Add > Class... > dê o nome de EmpresaModel.cs
  + Implementar a classe EmpresaModel

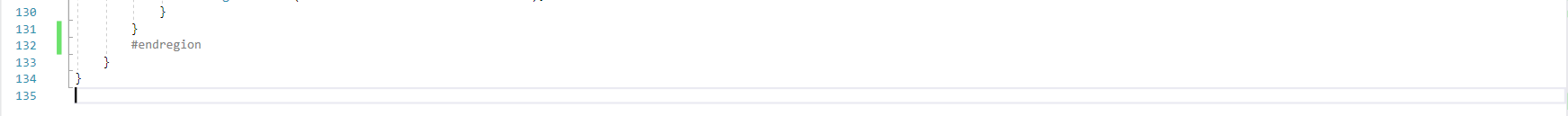


* CRUD da entidade empresa
  + Criar a classe para fazer o CRUD da empresa: botão esquerdo na pasta DAO > Add > Class... > dê um nome para a classe (eu chamei de EmpresaDAO.cs)
  + Declare um objeto do tipo MySqlConnection. Use ele para acessar o método getConnection() da classe ConnectionFactory
  + Implemente os métodos CreateEmpresa(), ReadEmpresa(), UpdateEmpresa() e DeleteEmpresa():

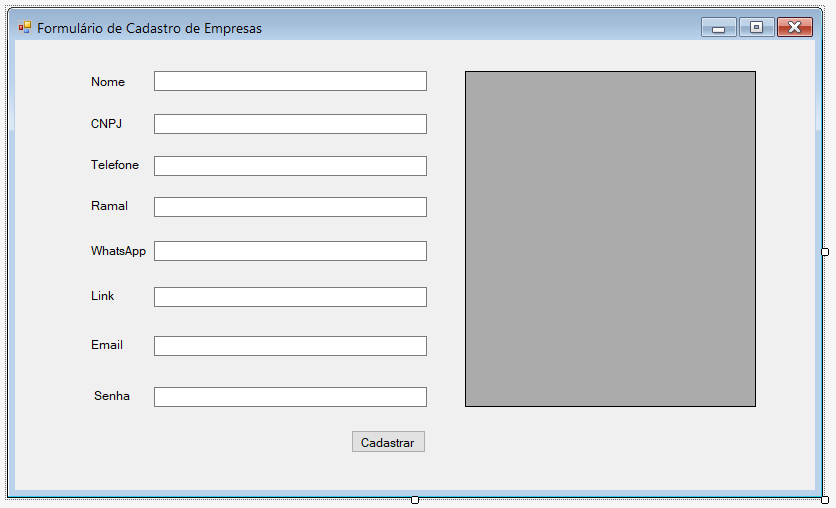






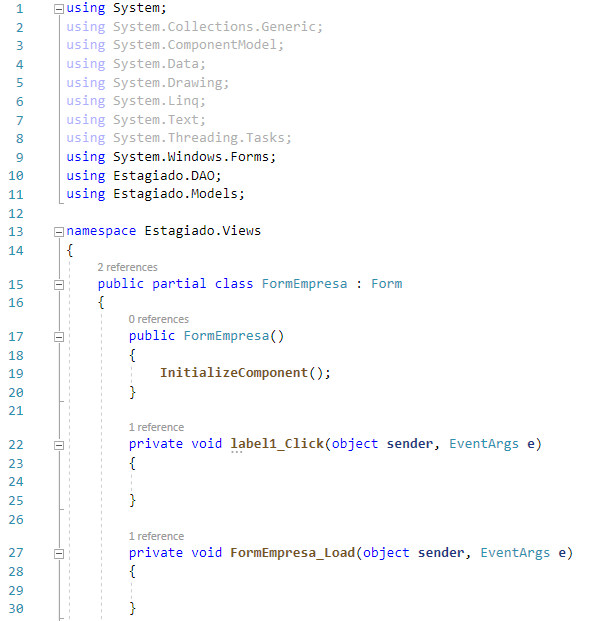


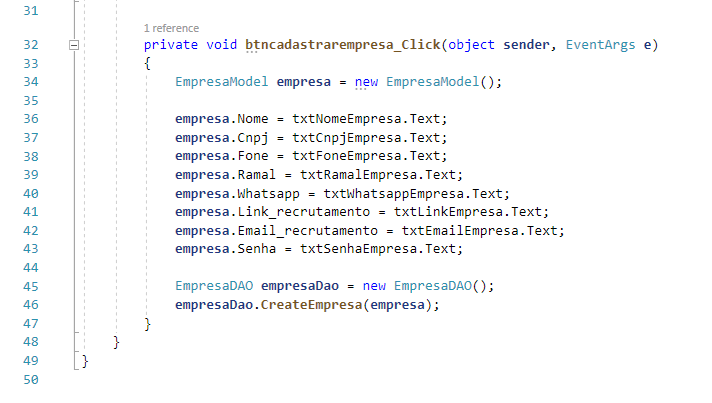
* Formulário de cadastro de empresas
  + Criar um formulário de cadastro: botão esquerdo na pasta Views > Add > Form (Windows Forms)... > dê um nome para o formulário (eu chamei de FormEmpresa)
  + Monte o formulário clicando e arrastando os itens para a tela. Veja o do estudante se estiver com dificuldade. O meu ficou assim:



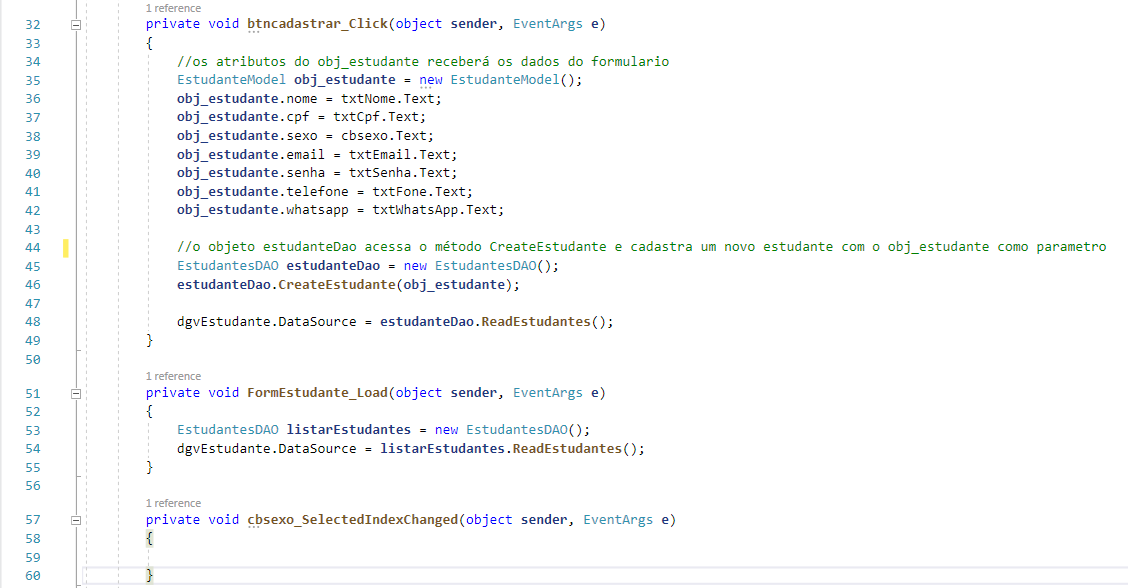
Os Names que coloquei nas entradas de dados foram:

* + TextBox para Nome: txtNomeEmpresa
  + TextBox para CNPJ: txtCnpjEmpresa
  + TextBox para Telefone: txtFoneEmpresa
  + TextBox para Ramal: txtRamalEmpresa
  + TextBox para WhatsApp: txtWhatsappEmpresa
  + TextBox para Link: txtLinkEmpresa
  + TextBox para Email: txtEmailEmpresa
  + TextBox para Senha: txtSenhaEmpresa
  + Button Cadastrar: btncadastrarempresa
  + DataGridView: dgvEmpresa
* Recuperar dados do formulário e enviar para o banco de dados
  + Edite a classe do formulário: dois clicks sobre o botão Cadastrar para abrir a classe do formulário (FormEmpresa no meu caso). Edite ela da seguinte forma:





* Listar estudantes cadastrados no DataGridView
  + Clique sobre formulário de estudante e edite as classes FormEstudante\_Load() e btncadastrar\_Click()



Nota: repare que dgvEstudante é o Name que foi dado para o DataGridView do formulário do estudante