Serviço Nacional de Aprendizagem Comercial

Curso Técnico em informática

Vinicius Augusto Vicentin

Projeto Final

Consultório Medico

Jundiaí

2020

Vinicius Augusto Vicentin

Projeto Final

Consultório Medico

Projeto final de conclusão do curso Técnico em informática, da instituição Serviço de Aprendizagem Comercial.

Orientador: Marcos Adriano F. Cozatti

Jundiaí

2020

Sumário

[Introdução 4](#_Toc58083741)

[Documentação 5](#_Toc58083742)

[Conexão com Banco de dados 5](#_Toc58083743)

[Criação e relacionamento de tabelas 5](#_Toc58083744)

[Criação do design das telas do Sistema 9](#_Toc58083745)

[Explicações dos códigos e suas funcionalidades. 14](#_Toc58083746)

[Tela de Login 14](#_Toc58083747)

[Códigos das Telas de Paciente, Medico. 17](#_Toc58083748)

[Códigos Da Tela de Consulta 21](#_Toc58083749)

[Tela de Cadastro de Especialidade 22](#_Toc58083750)

[Tela de manutenção de Usuários. 24](#_Toc58083751)

# Introdução

O tema do projeto constitui-se em um sistema de consultório, o projeto foi pensado com o intuito de mostrar tudo o que foi aprendido durante os ensinamentos do modulo de programação.

O Projeto foi pensado para ser usado em um consultório onde ele vai ter como função ajudar com a organização de dados de pacientes, médicos e consultas.

Para a construção deste foi utilizado as seguintes ferramentas: linguagem de programação em C#, com a plataforma do Visual Studio 2019, Microsoft SQL server management Studio 18.

# Documentação

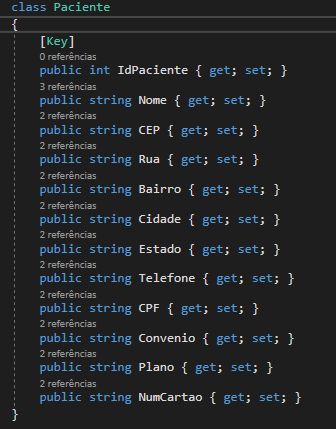
## Conexão com Banco de dados

A construção deste sistema foi feita da seguinte maneira, devido a tecnologia de conexão utilizando o Entity Framework Database First. Primeiramente eu dei início a criação do banco de dados (DbConsultorio), em seguida já dentro do visual Studio instalei a biblioteca EntityFramework, logo após isso fiz com que esse banco e o meu sistema se conectassem por meio da ferramenta fonte de dados do visual, mudei o nome desta conexão que por padrão ela vem com o nome do seu banco mais um dbSet na frente, para a simples palavra “conexão”.

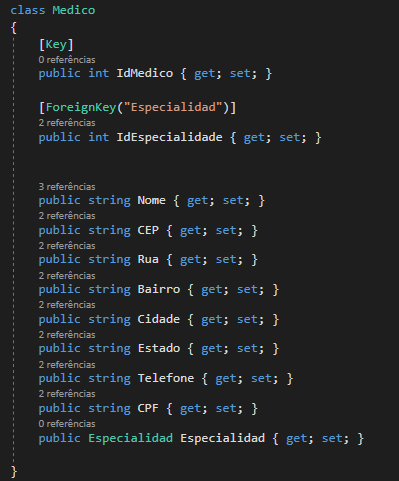
## Criação e relacionamento de tabelas

Em seguida parti para a criação das cinco tabelas e relacionamentos entre elas, como estava usando a tecnologia Database First e o banco já tinha sido criado dentro do SQL Server a criação das tabelas foi feita já dentro do Visual Studio da seguinte maneira:

* Primeira tabela(Paciente)

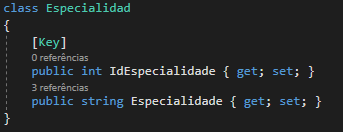


* Segunda Tabela(Medico)



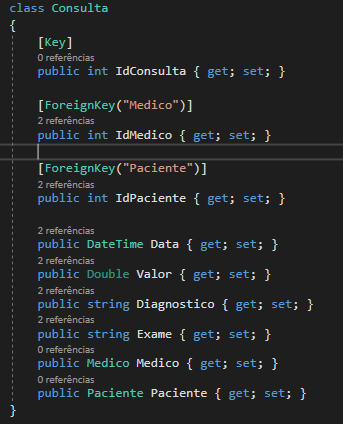
Esta tabela tem um diferencial que é a “ForeignKey” que é a chave estrangeira usada para fazer o relacionamento com outras tabelas neste exemplo ela foi usada para fazer o relacionamento com a tabela “Especialidade” que até esse momento ainda não foi criada.

* Terceira Tabela(Especialidad)



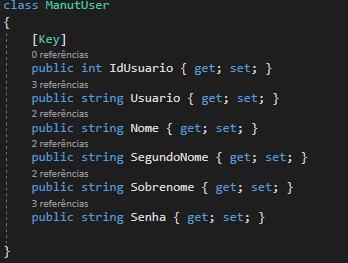
Como já foi dito a segunda tabela(Medico) possui relacionamento com a Terceira Tabela(Especialidad) porém a Terceira Tabela(Especialidad)Não possui relação com a segunda tabela(Medico).

* Quarta tabela(Consulta)



Essa tabela assim como a segunda possui relação com outra tabela, no casso essa possui conexão com outras duas tabelas, umas delas é Segunda Tabela (Medico) e com a Primeira Tabela (Paciente).

* Quinta Tabela(ManutUser)



Essa tabela assim com a primeira não tem relação com nenhuma outra.

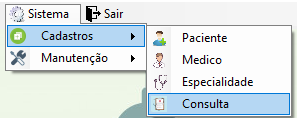
Depois da criação de todas as tabelas ficou faltando “upar” essas tabelas no banco de dados, para isso utilizei outra ferramenta do visual chamada de Console do Gerenciado de Pacotes que é basicamente um terminal, dentro deste “Terminal” eu utilizei três comando, um deles foi o enable-migrations que é como o próprio nome diz o modo de habilitar a migração de algo para o banco neste casso foram as tabelas, o segundo comando foi o Add-migration que faz com que você de um nome para essa migração, e o terceiro comando foi o Update-database que “upa” essa migração para o banco de dados.

## Criação do design das telas do Sistema

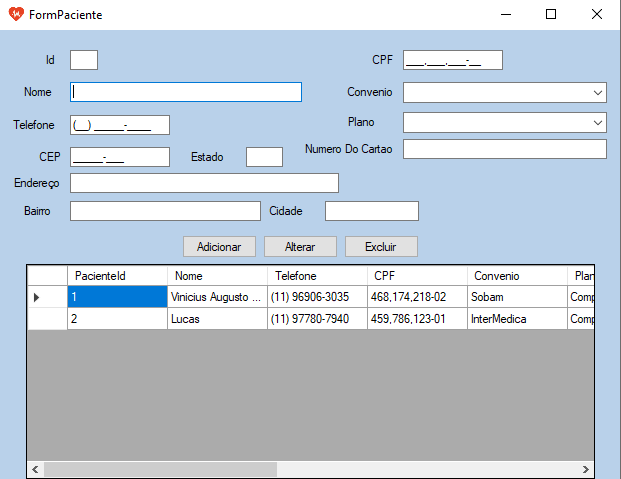
A criação destes design foi em resumo a parte mais simples do projeto, o primeiro design que foi criado foi o da tela inicial do sistema:



Este “Forms” tem como seu único diferencial uma ferramenta chamado MenuStrip que serve como um DropBox que nada mais é do que um botão que quando clicado traz mais “Botões”

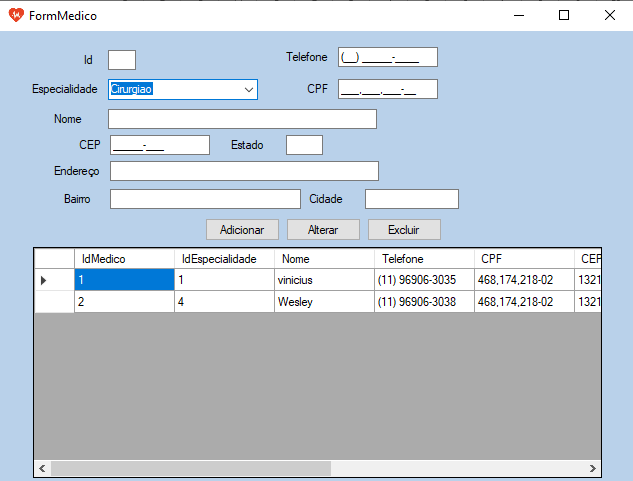


O segundo design que eu criei foi feito para o cadastro de pacientes:



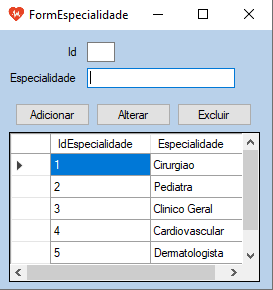
Esse “Forms” tem como diferencial algumas ferramentas, entre elas está o MaskedTextBox que é um TextBox porém com o diferencial de você conseguir criar mascaras para esse textBox, por exemplo (números de telefone, cep, cpf, entre outras), outra ferramenta utiliza foi o comboBox que também é um textBox porém com um DropDown que é basicamente igual ao menustrip, também possui três botões para que seja possível cadastrar, excluir e editar os dados inserido no banco e por último a GridView que está sendo usada para exibir os dados que foram cadastrados no banco de dados.

O Terceiro design que eu criei foi feito para o cadastro de Médicos:



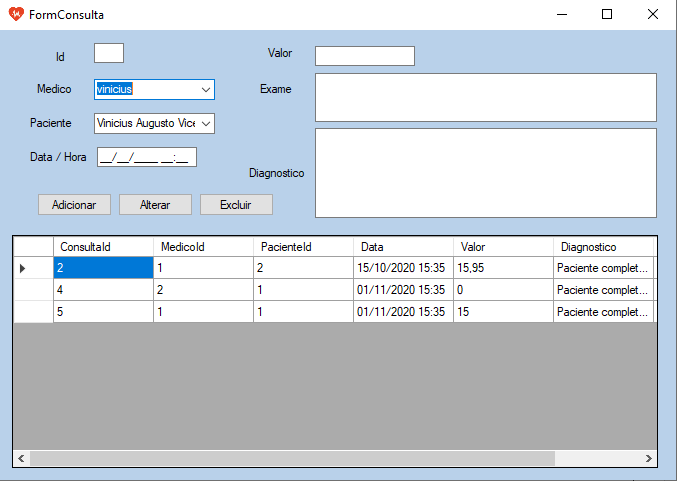
Esse “Forms” é bem semelhante ao segundo então suas funções também são as mesmas.

O Quarto design que eu criei foi feito para o cadastro de especialidade medicas:



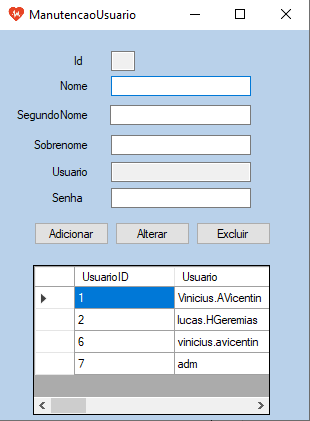
Esse “Forms” é sem dúvidas o mais simples então não possui nada de diferente dos outros.

O Quinto design que eu criei foi feito para o cadastro de consultas:

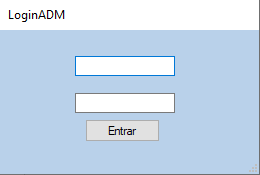


Esse “Forms” também possui as mesmas funcionalidades porém alterei a propriedade de dois textBox habilitando as multilinhas para eles ficarem maiores.

O Sexto design que eu criei foi feito para o cadastro de usuários:

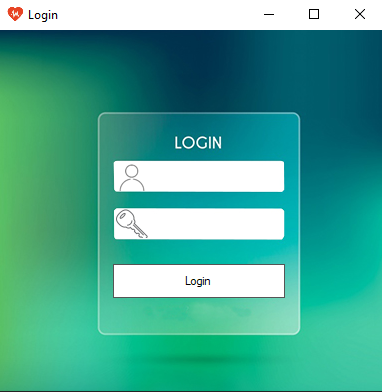


Esse “Forms” também possui as mesmas funcionalidade porém no TextBox de Usuário possui uma configuração de duplo clique que abri um outro formulário:



Que serve para fazer o Login de ADM para conseguir alterar algum usuário.

O Sétimo e último design que eu criei foi feito para o acesso ao sistema “Login”:

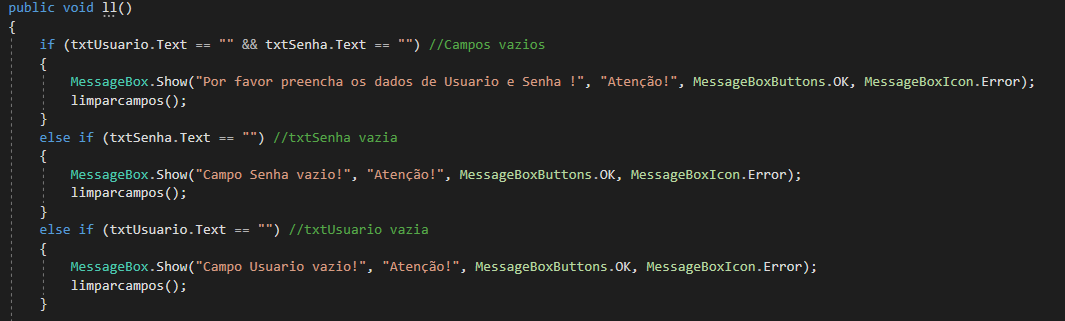


Esse “Forms” também é bem simples possuindo apenas dois campos de texto, um para o usuário e outro para a senha e um botão para fazer o login.

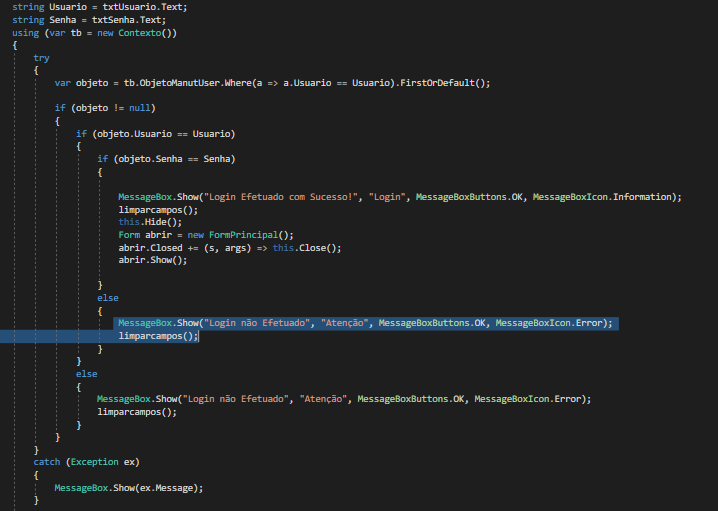
## Explicações dos códigos e suas funcionalidades.

## Tela de Login

Começando pelos códigos da tela de login:



Estes códigos acima são basicamente para alertar o usuário de que a campos vazios, o primeiro if é para alertar de que ambos os campos (Usuário e Senha) estão vazios, se isso acontecer ambos os campos são limpos, o primeiro else if é para alertar de que o campo (Senha) não foi preenchido, se isso acontecer ambos os campos são limpos, e o segundo else if é para alertar que o campo (Usuário) está vazio, se isso acontecer ambos os campos também são limpos.



Os códigos acimas estão fazendo exatamente isso:

A primeira linha de código está criando uma variável do tipo string que pode receber basicamente qualquer caractere, como números, letras, símbolos e etc. essa variável que está sendo criada recebe os dados que estão sendo digitados pelo usuário no campo de (txtUsuario) e armazenando esses dados.

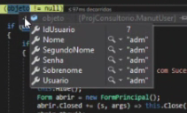
A segunda linha de código está criando outra variável do tipo string. Essa variável que está recebendo os dados que estão sendo digitados pelo usuário no campo de (txtSenha) e armazenado esses dados.

A terceira linha também cria uma variável (tb) porém os dados que ela está recebendo não são digitados pelo usuário os dados que ela recebe estão em uma outra classe que foi criada para fazer a conexão do Sistema com o Banco de Dados (Contexto), então devido a esses dados que ela recebe a variável consegue buscar os dados que foram inseridos dentro do banco.

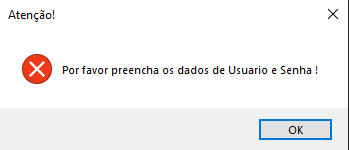
Explicação de toda a linha de comando Try:

Os códigos que estão dentro desta linha de comando tem as seguintes funções.

Primeira parte do código – Estão criando uma variável (objeto) onde ela recebe os dados da variável (tb), esses dados são (ObjetoManutUser) que é a tabela da manutenção de usuários onde está cadastrado os logins. Ela recebe todos os dados da linha de cadastro e armazena na memória.



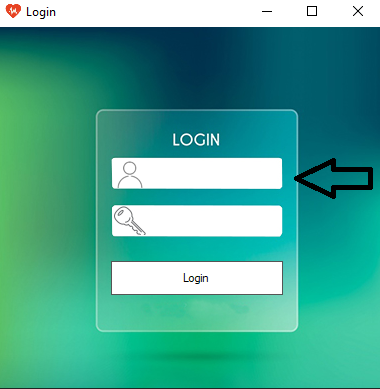
Segunda parte do código – O primeiro if verifica se a variável objeto não está vazia, se ela estiver vazia o programa já mostra uma mensagem q o login não foi efetuado.



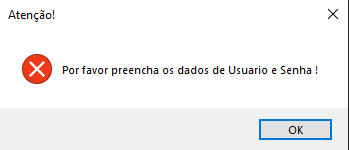
Se a variável possuir algum dado então ela passa para o segundo if que pega os dados que foram cadastrados no campo usuário



E verifica se os dados que estão digitados ali são iguais aos dados digitados pelo usuário dentro do textBox(Usuário)

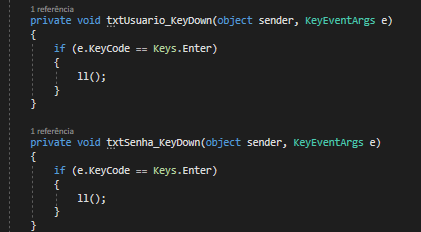


Se esses dados baterem então ele passa para o próximo if se não ele dá uma mensagem onde o login não foi efetuado.



Terceira Parte – O terceiro if função do primeiro com a diferença de que ele verifica a senha e não o usuário, se os dados baterem ele entra no sistema e fecha a tela de login, se os dados não baterem e mostra outra mensagem onde o login não foi efetuado limpa os campo e foca o texto no txtUsuario.

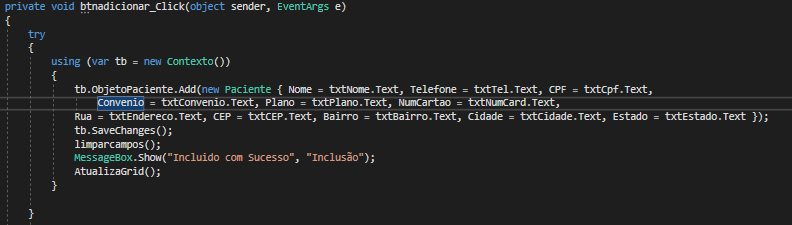
Quarta e Última parte da tela de login.



Esta parte do código serve para você consegui logar apertando a tecla enter.

## Códigos das Telas de Paciente, Medico.

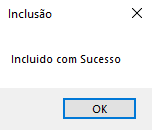
Em seguida temos os códigos da tela de cadastro de pacientes:



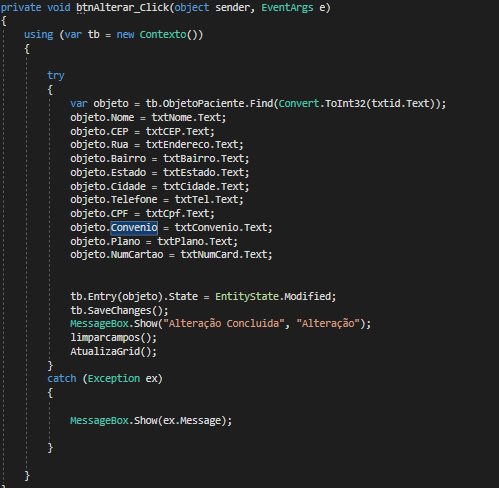
O código acima tem como funcionalidade a inserção de dados no banco.

Devido a biblioteca EntityFramework esse código fica muito mais simples, basicamente ele está criando(Instanciando) uma variável (tb) onde ela recebe dados da classe contexto que como já foi explica é responsável por fazer a conexão com o banco de dados, devido a isso a “variável” (tb) esta recebendo tudo o que a classe contexto possui, neste caso ela está recebendo a entidade/tabela(Paciente). O Add que vem a seguir no código é devido a biblioteca EntityFramework que é responsável por fazer a adição dos dados, logo após você instancia a tabela que vai receber os dados no caso (Paciente) e por fim você chama os campos da tabela e define o que esses campos estarão recebendo e finaliza salvando eles.

Se este código funcionar ele traz uma caixa de texto contendo a seguinte frase.

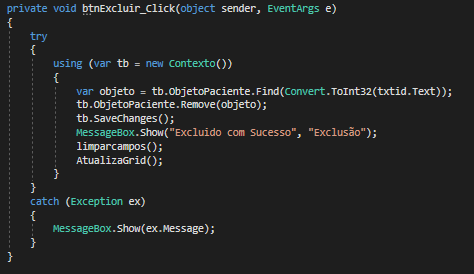


O próximo código tem como funcionalidade a alteração de dados no banco.



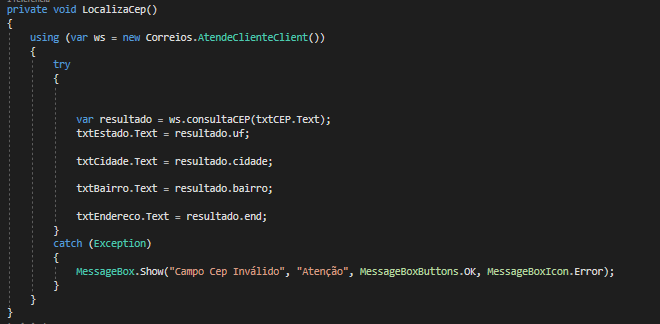
O código da alteração de dados tem o seu início bem parecido com o código anterior, ele possui algumas diferenças uma é que ele cria uma variável (objeto) que recebe dados do banco e o comando FIND, que também é da biblioteca EntityFramework e sua função é fazer uma busca em cima do campo (txtId) e em seguida ele faz um select desses dados, depois como a variável (objeto) possui os dados do banco, ela consegue definir o que cada campo da tabela vai receber, e por fim ele altera esses dados e salva.

O próximo código tem como funcionalidade a exclusão de dados no banco.



O código de exclusão tem suas primeiras linhas de comando idênticas as do código anterior porém após ele receber os dados e fazer um select em cima do campo (txtid) ele faz a exclusão destes dados.

O próximo código tem como funcionalidade ajudar na inserção de dados de endereço no banco.



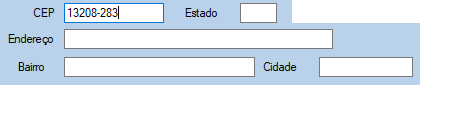
Primeiramente para que esse código funcionasse foi preciso fazer uma conexão em uma outra base de dados nesse casso alguma que continha dados de endereço.



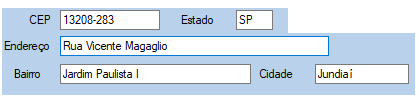
Assim como a biblioteca EntityFramework possui comados que você consegue utilizar após fazer a sua instalação, essa conexão que fizemos também possui.

Para os códigos anteriores estávamos criando uma variável que recebia dados do nosso banco, neste casso estamos criando uma variável que recebe dados desta nova conexão que fizemos.

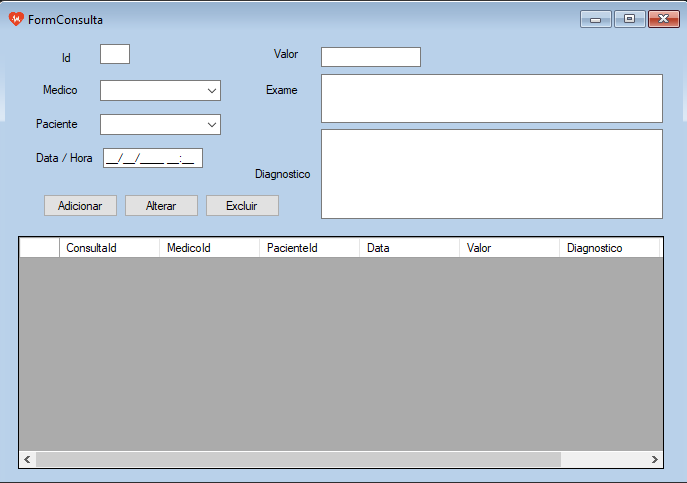
Após isso estamos criando uma variável que recebe os dados que estão sendo digitados pelo usuário no campo de cep e faz uma consulta nessa base de dados de acordo com o cep digitado. Devido a isso para preencher os campos de estado, cidade, bairro e endereço basta você digitar o seu cep.



Quando você sair desta caixa de texto os dados serão preenchidos.

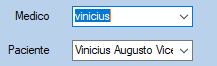
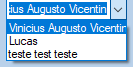


## Códigos Da Tela de Consulta

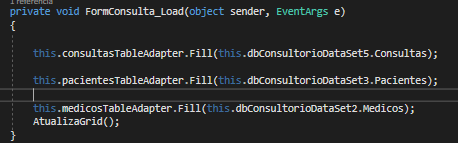


Os códigos desta tela assim os códigos acima de inserção, alteração e exclusão essa tela também possui portanto não estará especificado como eles funcionam aqui.

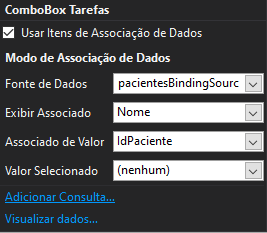
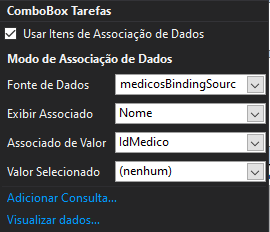
Os diferencias desta tela são os comboBox.

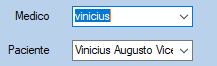
As funcionalidades deles são guardar dados para facilitar na inserção deles, devido ao relacionamento de tabelas que fizemos, conseguimos buscar dados de outras tabelas, aqui o comboBox está exibindo dados de duas outras tabelas, uma delas é a medico e a outra paciente.

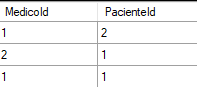


Para exibir esses dados fizemos o seguinte.

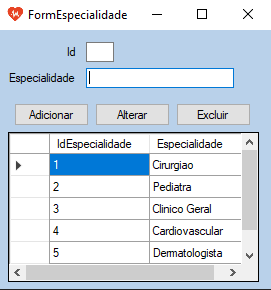
 

O ComboBox te permite adicionar fontes de dados para buscar dados de outro “lugar”. Fonte de dados é o lugar onde ele vai buscar esses dados.

Exibir Associado  é o que vai aparecer no comboBox.

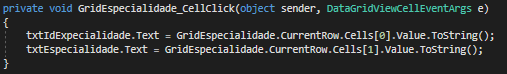
Associado de Valor  é o local (id) onde ele busca os dados.

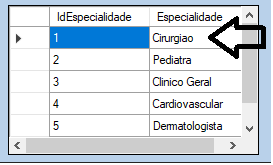
## Tela de Cadastro de Especialidade



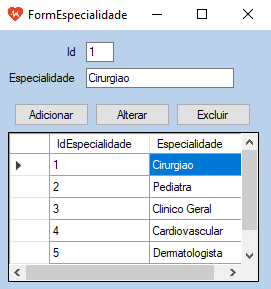
Assim como as demais essa tela também possui os mesmo comando para inserir, alterar e excluir algum dado.

Algo que ela possui e as outras também e que ainda não foi explicado é o seguinte código.

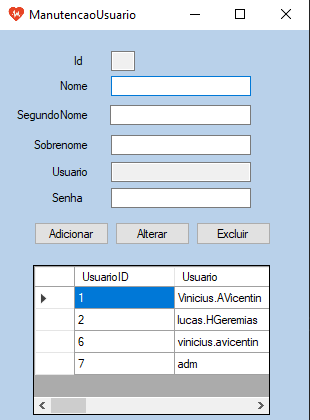


Que faz com que quando você clica dentro da grid em algum dado.

Ele preenche os campos de texto.



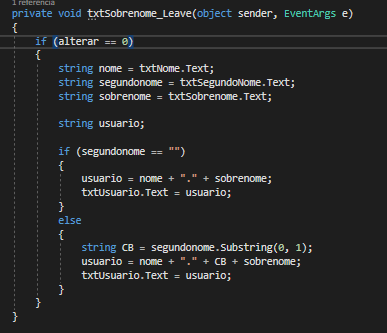
## Tela de manutenção de Usuários.



Essa tela também possui os mesmo comandos para adicionar, alterar e excluir dados.

Os diferenciais desta tela são:

* Concatenação de nomes para cadastro de usuários-



É feita da seguinte maneira, primeiro criamos três variáveis que estão recebendo os dados digitados pelo usuário.

O primeiro if deste código está verificando se o campo (txtSobrenome) está vazio ou não, se ele estiver então o programa vai pegar o primeiro nome do usuário adicionar um ( . ) e em seguida concatenar o sobrenome dele.

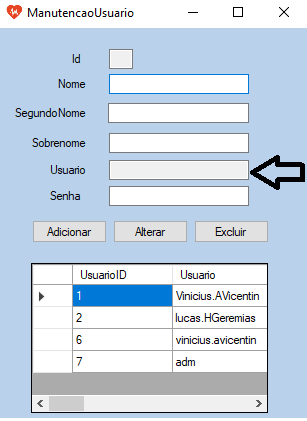
Se esse campo não estiver vazio então ele cria uma nova variável que recebe o segundo nome digitado por esse usuário e pega apenas a primeira letra deste nome. Logo após ele insere o primeiro nome deste usuário assim como explicado anteriormente e adiciona um ( . ) em seguida ela concatena com a variável que foi criada para receber a primeira letra do segundo nome e por fim concatena com o sobrenome do usuário.

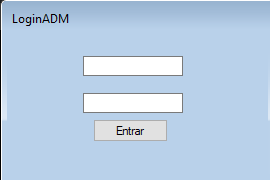
Exemplo: Nome—Vinicius || SegundoNome –Augusto || **SobreNome** -- Vicentin

Usuário – Vinicius.AVicentin

* Alteração de usuário feita apenas por um login de adm.

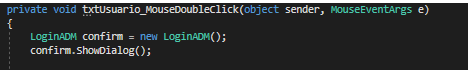
Quando clicamos duas vezes com o botão esquerdo do mouse em cima do campo usuário o programa abre uma outra tela para q o usuário faço um login de administrador.



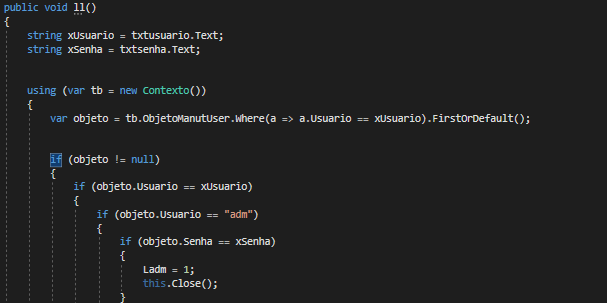


Após o login ser efetuado o programa volta para a tela de manutenção de usuários porém ele desabilita a configuração feita para um usuário comum não conseguir alterar o seu usuário.

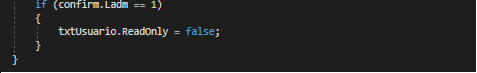
O código por de traz disso é o seguinte.



Está parte ele está trazendo a tela de loginADM.



Está parte é idêntica ao login padrão que já foi explicado no início, basicamente ele está verificando se o usuário e a senha que foi digitada está de acordo com o usuário e a senha cadastrada para o administrador. Se esses dados estiverem corretos então o programa volta para a tela de Manutenção de usuários.



Esta parte do código está confirmando se a variável criada (Ladm) tem como seu valor o número “1” se ela possuir esse valor então ela desabilita a função de apenas ler o campo txtUsuario, fazendo com que assim torne-se possível fazer sua alteração.

Vinicius Augusto Vicentin

Projeto Final

Consultório Medico

Projeto final de conclusão do curso Técnico em informática, da instituição Serviço de Aprendizagem Comercial.

Orientador: Marcos Adriano F. Cozatti

Jundiaí

2020