PEDRO HENRIQUE SCAPIM T.I 32 SENAC

DOCUMENTAÇÃO CONSULTORIO MEDICO

**1. INTRODUÇÃO**

Nós estudantes do Senac da turma TI32 tivemos como objetivo fazer um projeto no qual desenvolveríamos um programa para um consultório médico.

Poderíamos escolher o formato de fazer, pois ao longo do curso aprendemos diversas formas de fazer CRUD com Windows Forms em c#. Tivemos a opção em fazer de Entity Framwork ou MVC.

O meu Trabalho se baseia em uma criação MVC, na qual me senti mais a vontade e com mais conhecimento para fazer sem muitos problemas ou dificuldades.

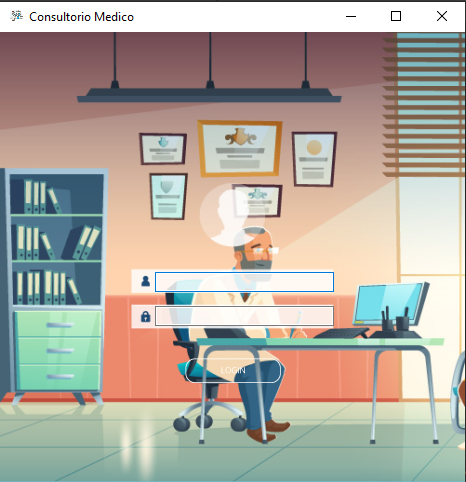
A seguir na documentação será analisado alguns pontos do programa e uma breve informação sobre os códigos.

* 1. **CRONOGRAMA DAS ATIVIDADES DO PROJETO**

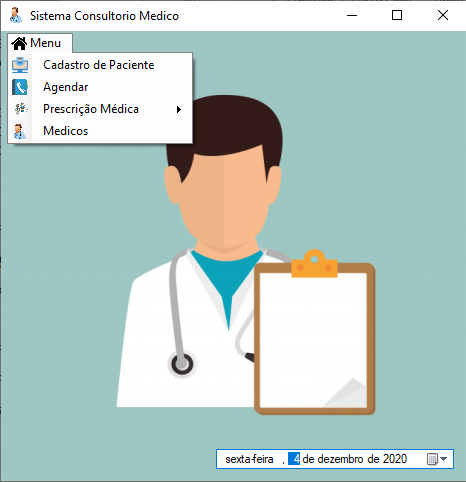
O trabalho foi passado no meio de novembro, aonde tivemos cerca de 3,4 semanas, e o programa foi sido desenvolvido em todo o tempo dado. Comecei com a criação do banco de dados, e em seguida foi passado para a fase do front end.

**2.DESCRIÇÃO GERAL DO SISTEMA**

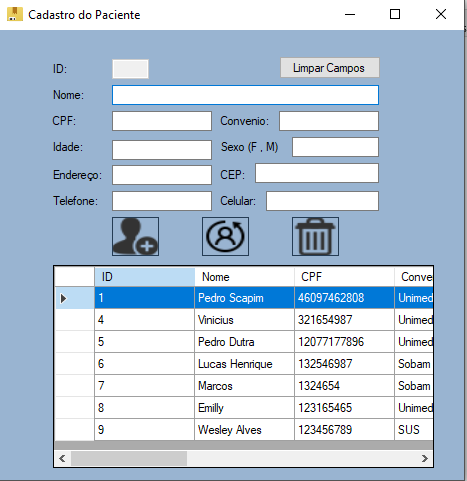
O Sistema é feito para um consultório médico, onde temos telas para: Cadastrar pacientes, agendar horários e consultar os horários marcados, cadastrar médicos que trabalham no hospital/clinica e uma tela aonde você pode consultar todas as vezes que o paciente passou por lá e quais foram os diagnósticos da última consulta.



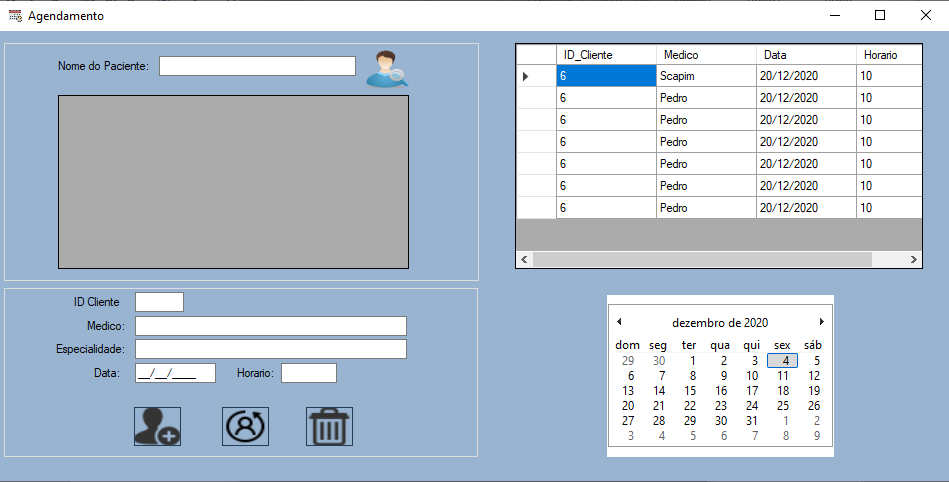
Essa é a tela de login, onde apenas funcionários tem permissão para entrar no sistema e consultar as informações que tem no sistema. Caso a pessoa tente e não é cadastrada o programa continua bloqueado.



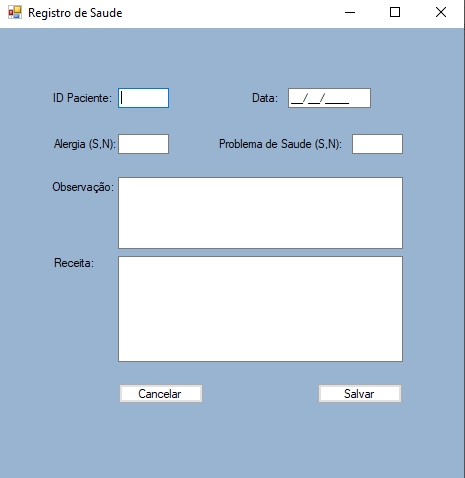
A tela de menu é onde há possibilidades de escolher quais telas você pretende usar, todas são marcadas com nome, e um ícone ao lado para melhor entendimento do usuário. Nessa tela também há um simples calendário que a pessoa consegue ver qual o dia atual e consegue consultar os outros dias do ano.



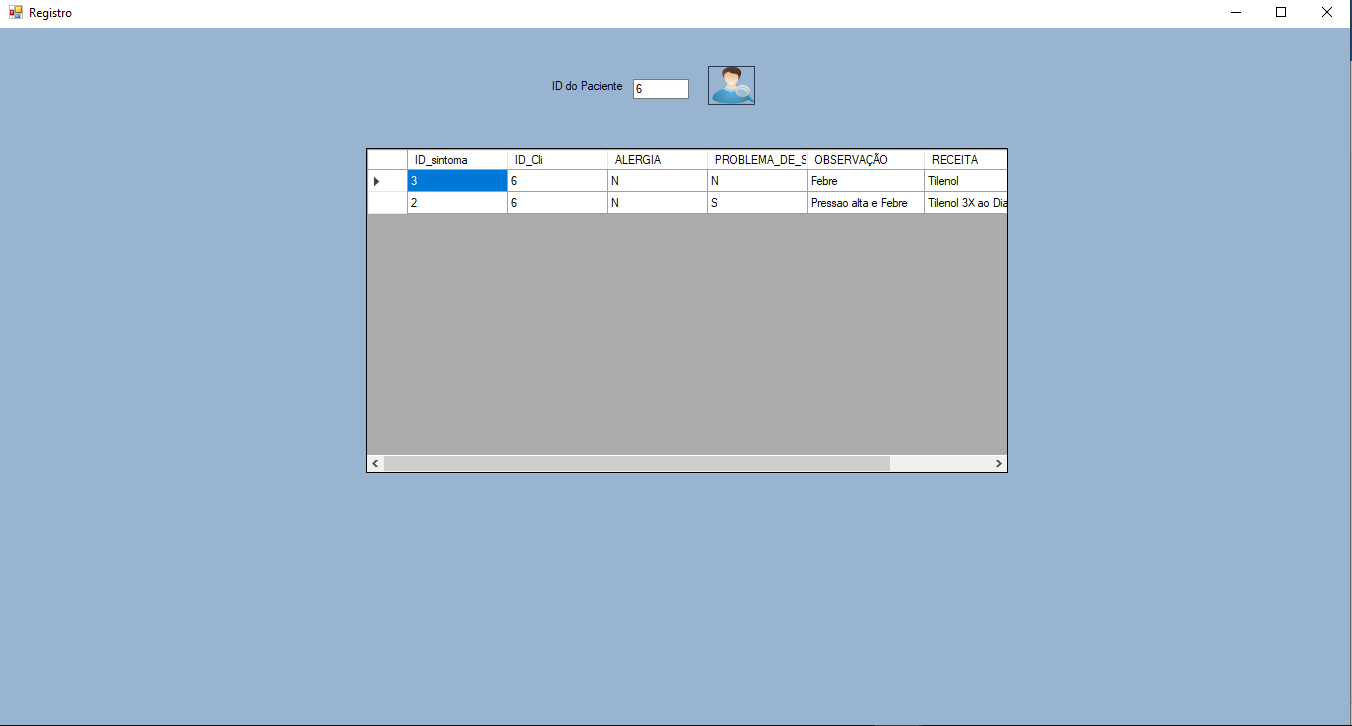
Essa é a tela de cadastro do paciente. Nela você tem as informações essenciais e necessárias para cadastro do paciente. Nela você consegue inserir, alterar e até apagar um usuário, mas claro que com restrições. Se esse paciente tem hora marcada na agenda, ele não poderá ser removido do cadastro, pois o banco não permite apagar um dado quando tem relação com outra coluna.



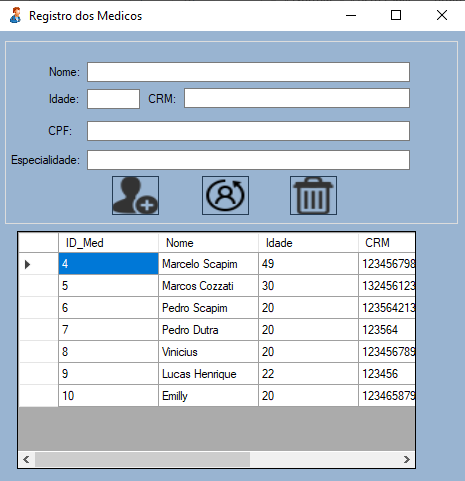
Aqui temos a tela de agendamento. Nela você consegue buscar o ID do paciente que você precisa para fazer o agendamento. Ao fazer a buscar e clicar no paciente, o id é encaminhado para o campo ID Cliente e você precisa apenas terminar de preencher o resto. Ao lado temos uma grid onde aparece todos os horários marcados, sempre em ordem de dia. E em baixo um calendário para que quem esteja agendando o horário saiba o dia da semana e o me no qual esta marcando, assim evitando confusões.  
Como na tela de cima, você altera, exclui e inseri um agendamento.



Ao passar com o médico, ele conseguirá salvar as informações de quando você passou com ele sobre seu estado de saúde, e também sobre os remédios que você foi receitado.



Essa tela o médico consegue ver o histórico de quantas vezes você passou por lá, quais foram os dias e como estava seu estado de saúde atual.



Por fim a tela de cadastro de médicos, onde nela você cadastra, apaga e altera todos os médicos que trabalham no seu hospital/clinica. Assim como todas as telas de cadastro nela há botões de inserir, alterar e deletar.

**2.1 UTILIZADORES DO SISTEMA**

Os utilizadores do sistema serão os médicos do hospital, enfermeiras e recepcionistas.

**2.2 DESENVOLVEDORES DO SISTEMA**

Pedro Henrique Scapim

**3.3 CODIGOS INSERIDOS NO PROGRAMA.**

Os códigos inseridos nas telas foram praticamente padrões. Foi feito uma conexão MVC que consiste em:

● Model

A responsabilidade dos models é representar o negócio. Também é responsável pelo acesso e manipulação dos dados na sua aplicação.

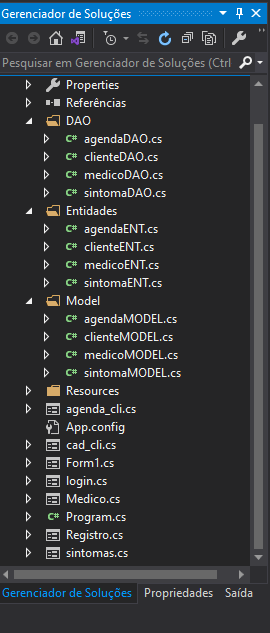
● View

A view é responsável pela interface que será apresentada, mostrando as informações do model para o usuário.

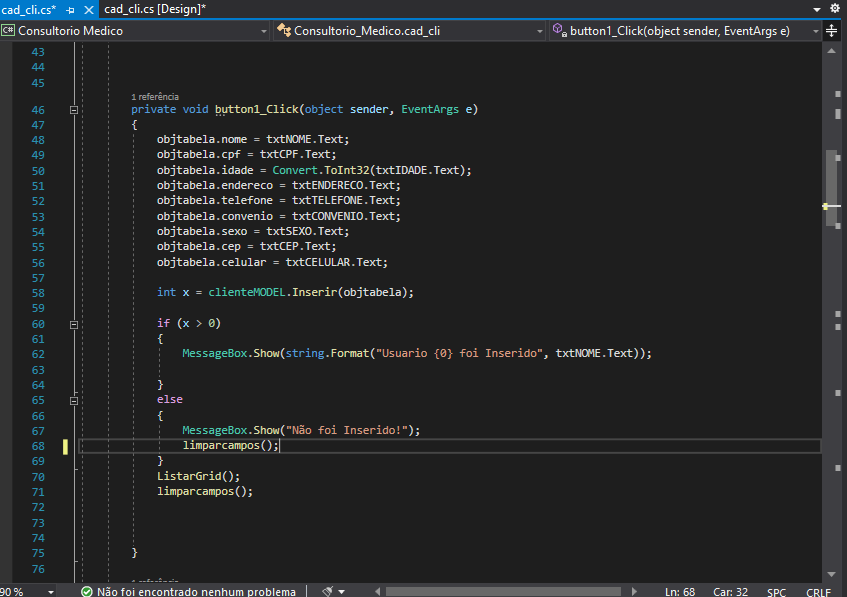
● Controller

É a camada de controle, responsável por ligar o model e a view, fazendo com que os models possam ser repassados para as views e vice-versa.

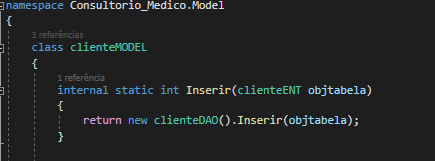
Foi criado Classes de DAO, ENTIDADES e MODELO.



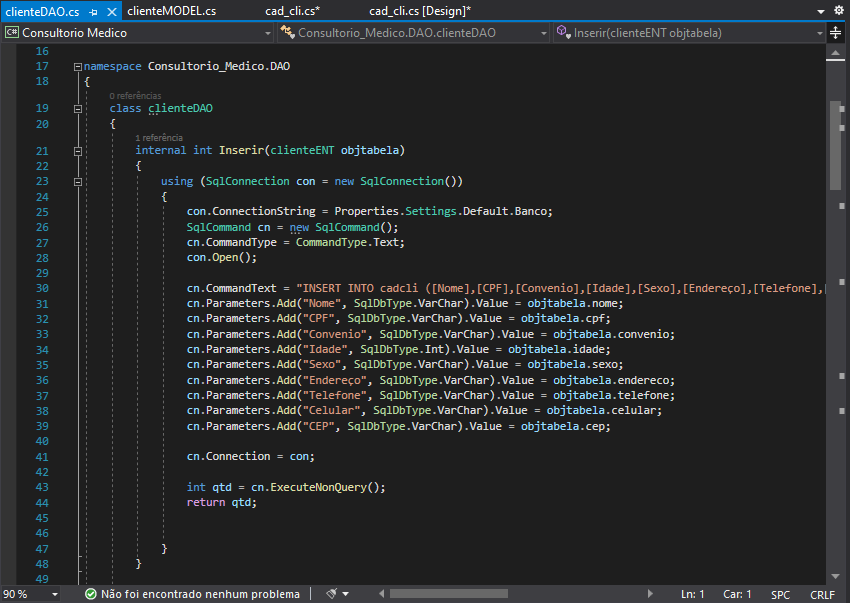
A maioria das telas tem um botão inserir, alterar e excluir, onde possuem o mesmo código, mudando apenas o código SQL, pois existem tabelas diferentes. Aqui vai um exemplo:



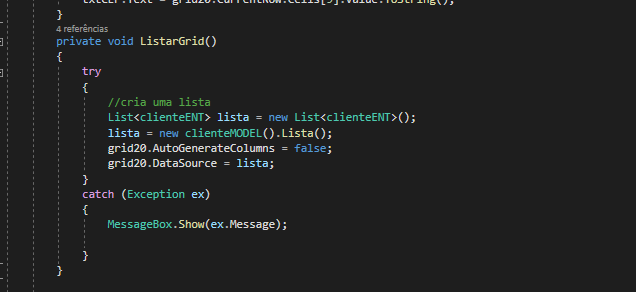
Aqui temos o botão inserir, onde ele faz uma referencia ao MODEL, que possui o seguinte código:

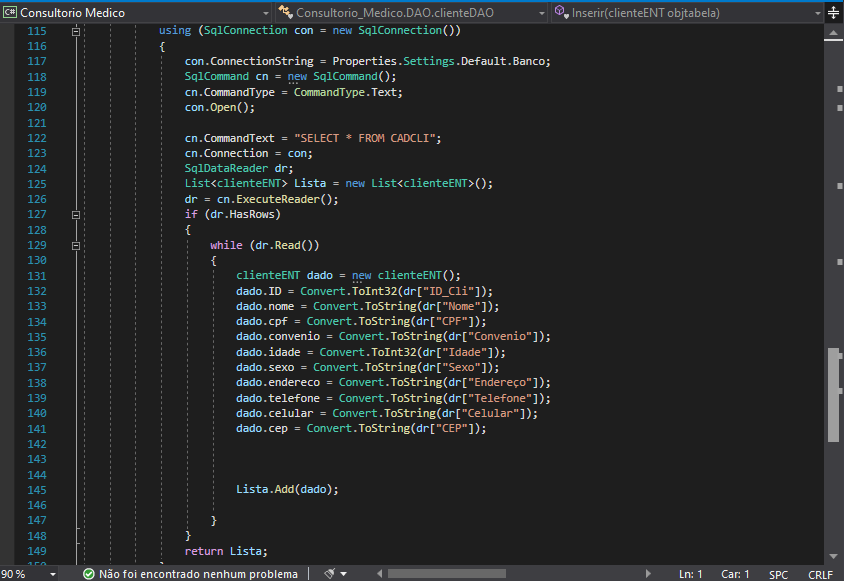


Neste código ele faz uma referencia ao DAO, onde será feita a conexão ao banco no evento de click e executado um código SQL:

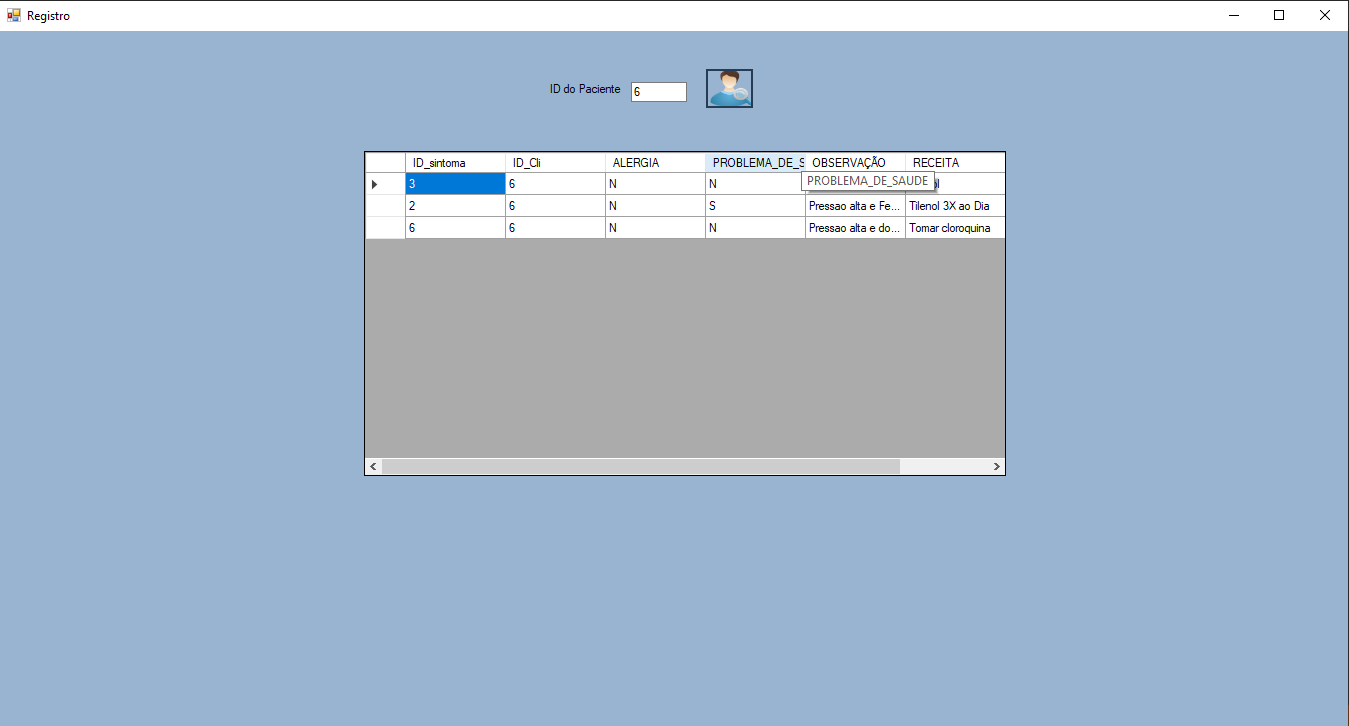


Todas as telas são padrão. E para as grids, normalmente é usado uma SELECT, como na próxima imagem:

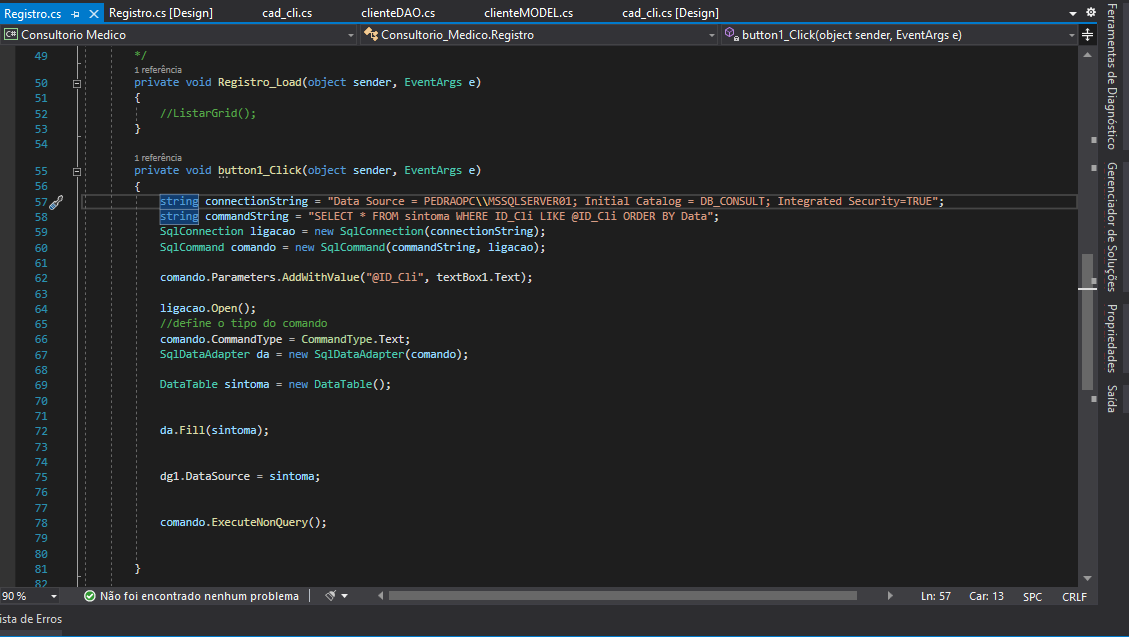
****

****

As telas normalmente seguem o mesmo padrão de código, apenas uma tela de consultar pacientes que é diferente, pois nela eu consigo fazer uma busca através de um TextBox e ele inseri os dados para mim na grid. Acompanhe o exemplo a seguir:



Ao escolher um ID, você consegue ver todas as passagem do paciente pelo consultório, e ordena por datas.



Nesse código ele faz a select e joga os dados direto para a grid, de acordo com sua busca.

**3.4. CODIGOS DO SQL/BANCO DE DADOS**

SCRIPTS:

**Tabela do Paciente:**

CREATE TABLE cadcli(  
ID\_Cli INT IDENTITY (1,1) PRIMARY KEY,  
Nome VARCHAR(100) NOT NULL,  
CPF VARCHAR(13) NOT NULL,  
Convenio VARCHAR(30),  
Idade INT,  
Sexo VARCHAR(1),  
Endereço VARCHAR(50) NOT NULL,  
Telefone VARCHAR(15),  
Celular VARCHAR(20) ,  
CEP VARCHAR(15)

CONSTRAINT CH\_Sexo CHECK (Sexo in (‘F’,’M’))

)

**Tabela do agendamento**

CREATE TABLE agendacli(  
ID\_agenda INT IDENTITY(1,1) PRIMARY KEY,  
ID\_Cli INT,  
Medico VARCHAR(40),  
Especialidade VAARCHAR(30),  
Data DATETIME,  
Horario VARCHAR(15),

CONSTRAINT FK\_ID\_CLIENTE FOREIGN KEY(ID\_Cli) REFERENCES cadcli(ID\_Cli)

)

**Tabela para diagnostico do Paciente:**

USE DB\_CONSULT  
CREATE TABLE sintoma(  
ID\_sintoma int identity(1,1) PRIMARY KEY,  
ID\_Cli INT,  
ALERGIA VARCHAR (1),  
PROBLEMA\_DE\_SAUDE VARCHAR(1),  
OBSERVAÇÃO VARCHAR(100),  
RECEITA VARCHAR(100),  
DATA DATE,

CONSTRAINT CH\_ALERGIA CHECK (ALERGIA IN('S','N')),  
CONSTRAINT CH\_PRO CHECK (PROBLEMA\_DE\_SAUDE IN ('S','N')),  
CONSTRAINT FK\_ID\_CLI FOREIGN KEY (ID\_Cli)REFERENCES cadcli(ID\_Cli)  
)

**Tabela para cadastrar médicos:**

CREATE TABLE Medico(  
ID\_Med int identity (1,1),  
Nome VARCHAR(50),  
Idade VARCHAR(50),  
CRM VARCHAR(50),  
CPF VARCHAR (50),  
Especialidade VARCHAR(50)