|  |  |
| --- | --- |
| **Nome do Sistema**    Clínica médica | **Versão**    1.0 |

# Descrição

O software Clínica médica é utilizado para controle de consultas, o mesmo traz a possibilidade de cadastrar médicos, pacientes e consultas. O mesmo conta também com telas de cadastro intuitivas e controle de agenda, evitando que mais de uma consulta seja marcada para um mesmo medico, sendo possível também editar e excluir as consultas agendadas.

# Motivação

Hoje em dia é algo muito comum ser necessário que marquemos consultas medicas, seja para acompanhar a nossa saúde ou para tratar algo que nos incomoda, essa tarefa parece algo bem simples, mas quando olhamos de forma micro percebemos que ela envolve inúmeros fatores, o Clínica médica vem para nos ajudar em toda a burocracia, para assim conseguirmos nos livrar de pilhas de prontuários ocupando espaço físico, erros em agendamentos de consultas e um melhor controle na agenda dos médicos e de seus históricos de consultas.

|  |  |
| --- | --- |
| **Plataforma-alvo**    Microsoft Windows 10 | **Principais tecnologias utilizadas**    .NET Framework 5.0, Linguagem de programação C# e SQL Server |

# Interface gráfica do usuário Descrição gerada automaticamente com confiança médiaDiagrama de Implantação

O software Clínica Médica é executado localmente na máquina do cliente sem a necessidade de conexão com a internet. O programa é composto pela interface visual, suas classes e banco de dados.

Clínica Médica: módulo principal do sistema, responsável pela interface visual e operações para seu funcionamento. É escrito utilizando a linguagem C#, Windows.Forms e SQL Server.

# Diagrama de casos de uso

Diagrama

Descrição gerada automaticamente

Usuário: usuário do sistema que realiza as ações.

Tela Principal: porta de entrada do sistema, a partir dela é possível selecionar o menu que se deseja acessar.

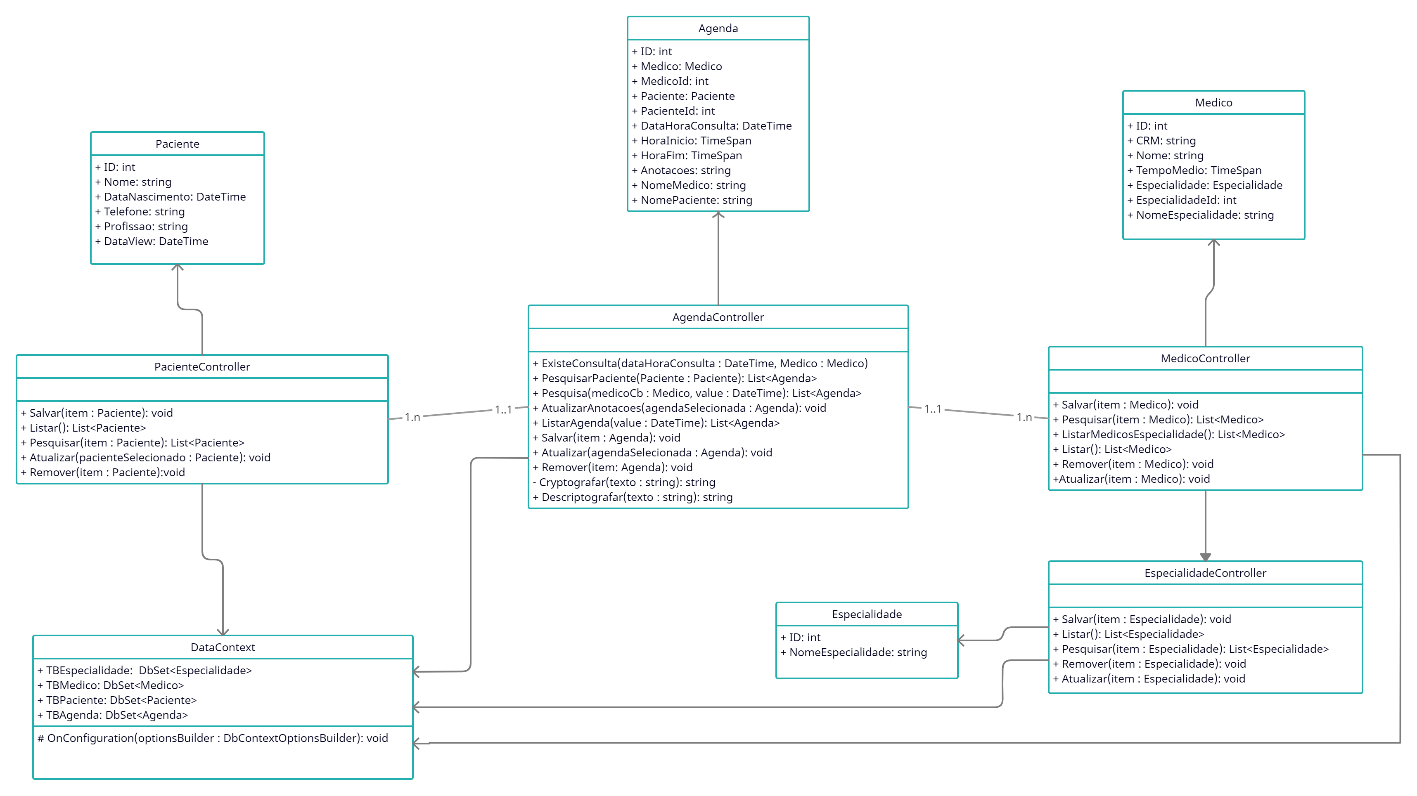
Adicionar: efetua o cadastro de especialidade, médicos, pacientes e agendamento de consultas.

Visualizar: lista todos os itens cadastrados até o momento com a possibilidade de filtrar os resultados obtidos.

Editar: edita o registro desejado.

Excluir: exclui o registro desejado.

# Diagrama de classes



Especialidade: Classe responsável por encapsular os dados referentes a especialidade.

Medico: Classe responsável por agrupar todos dos dados do médico.

Paciente: Sua função primordial é armazenar o objeto que compõe o paciente.

Agenda: Responsável por criar o objeto agenda e realizar os armazenamentos dos dados da consulta como médico, paciente, o agendamento da consulta e as anotações realizadas pelo médico sobre o paciente.

EspecialidadeController: Intermédia as solicitações da interface com o banco de dados referente as especialidades.

* Salvar: grava as informações passadas a ela no banco de dados.
* Listar: busca todos os itens cadastrado no banco de dados.
* Pesquisar: busca informações no banco de dados mediante ao termo passado.
* Remover: exclui o item passado como parâmetro do banco de dados
* Atualizar: atualiza o dado no banco de dados com o termo passado como parâmetro.

MedicoController: Controla o fluxo de requisições ao banco na busca da tabela médico.

* Salvar: cadastra o registro passado como parâmetro.
* Pesquisar: busca registro no banco se satisfeito a condição de busca.
* ListarMedicoEspecialidade: busca os registros de um médico se for pertencente ao grupo da especialidade solicitada.
* Listar: busca todos os registros no banco de dados.
* Remover: remove o registro passado como parâmetro do banco de dados.
* Atualizar: atualiza o registro com o nova recebido como parâmetro.

PacienteController: Gerencia os requerimentos solicitados do paciente no banco de dados.

* Salvar: salva o registro no banco de dados,
* Lista: lista todos os pacientes do banco de dados.
* Pesquisar: lista registros se eles satisfizerem a condição de similaridade com o parâmetro recebido.
* Atualizar: edita o paciente com os dados passados como parâmetro.
* Remover: exclui o paciente recebido como parâmetro.

AgendaController: Administra as solicitações do programa que deseja operar sobre os dados da agenda.

* ExisteConsulta: verifica se o horário solicitado para o agendamento da consulta já não está preenchido.
* PesquisarPaciente: busca os registros do banco de dados que pertencem a um paciente recebido como parâmetro.
* Pesquisar: lista consultas do médico e do dia que são filtrados a partido dos parâmetros recebidos.
* AtualizarAnotacoes: grava as alterações feitas pelo médico a respeito da consulta em atendimento.
* ListarAgenda: lista as consultas do dia recebido como parâmetro.
* Salvar: adiciona uma nova consulta ao banco de dados.
* Atualizar: realiza uma edição na consulta recebida como parâmetro.
* Remover: remove o registro recebido como parâmetro.
* Cryptografar: recebe um texto e o cryptografa para ser salvo.
* Descriptografar: recebe um texto e o descriptografa para ser exibido.

DataContext: Realiza a conexão com o banco.

* OnConfiguration: configura as credenciais para acessar ao banco.

# Requisitos funcionais

RF01: O sistema deve conter uma tela para que seja realizado o cadastro dos médicos que atende na clínica, a tela deve conter os campos para o cadastro do CRM, Nome, Especialidade e tempo médio de cada consulta do médico.

RF02: O sistema deve conter uma tela para o cadastro de pacientes, a mesma deve conter os campos para o cadastro do Nome, Data de Nascimento, Telefone e profissão do paciente.

RF03: O sistema deve conter uma tela para o cadastro das consultas, sendo possível selecionar o médico, paciente e horário da consulta, sem que o horário conflita com outra consulta já marcada.

RF04: O sistema deve conter uma tela com as informações sobre a consulta, para que o médico possa realizar anotações sobre a consulta e que possa também registra o horário de início e de fim da mesma, assim também mantendo um histórico da consulta.

# Requisitos não-funcionais

RNF01: O sistema deve ser executado em computadores com sistema operacional Windows e com o .Net *Framework* 5.0.

RNF02: O armazenamento das consultas deve ser por meio de um banco de dados SQL Server.

RNF03: O sistema deve ser executado de maneira *Off-line*.

RNF04: O sistema deve rodar em uma máquina que tenha pelo menos 4 GB de memória