

## Projeto e Desenvolvimento de Sistemas II

Guilherme Silva Daires  
Izabela Caroline Sousa Silva  
Lucas Rosa Alves  
Marcus Vinicius Thomé

### Nome do sistema

Clínica Médica - ConsultEasy

### Versão do Documento

4.0

### Descrição

O sistema será desenvolvido com o objetivo de realizar toda a gestão de consultas médicas. Além disso, realizar o controle dos dados dos pacientes e médicos, trazendo consigo a possibilidade de se obter uma agenda, onde os mesmos poderão administrar suas consultas e agendamentos, de acordo com datas e horários. Outro ponto que o sistema apresenta é a possibilidade de re-agendar as consultas, para os casos de não comparecimento da mesma, sendo possível justificar e incluir anotações nas observações. Em relação aos usuários, cada um deles terá a opção de criar seus respectivos login e senha, considerando que os acessos serão parametrizados de acordo com a permissão de cada usuário cadastrado. O sistema possui inúmeras funcionalidades que buscam facilitar o agendamento de consultas e gerenciamento de suas agendas.

### Motivação

De acordo com todas as dificuldades, desencontros e equívocos que ocorrem ao realizar o agendamento das consultas, visto que, neste âmbito faz-se necessário obter uma gestão assertiva no agendamento das mesmas, o sistema busca informatizar e facilitar estes processos. A maior motivação é aplicar e desenvolver um sistema que traga eficiência ao realizar os agendamentos, maior visibilidade e organização das consultas, tanto para os pacientes quanto para os administradores, sendo médicos, secretárias, etc. Além de ser possível controlar a agenda de pacientes e médicos, o sistema também possibilita a realização de reagendamentos, históricos médicos, e anotações.

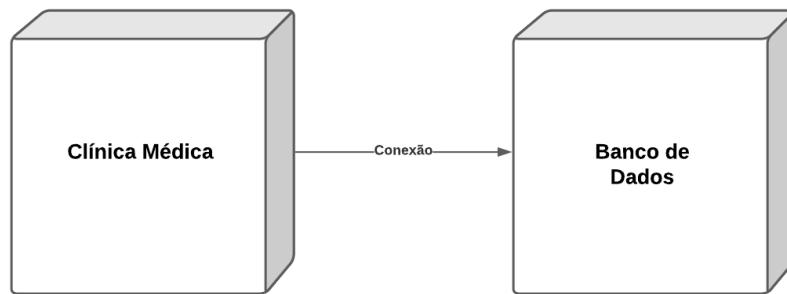
### Plataforma-alvo

Navegador com conexão à internet

### Principais tecnologias utilizadas

ASP.NET Core MVC, Docker, SQL Server, C#, HTML, CSS, JavaScript, JQuery e Bootstrap 4.

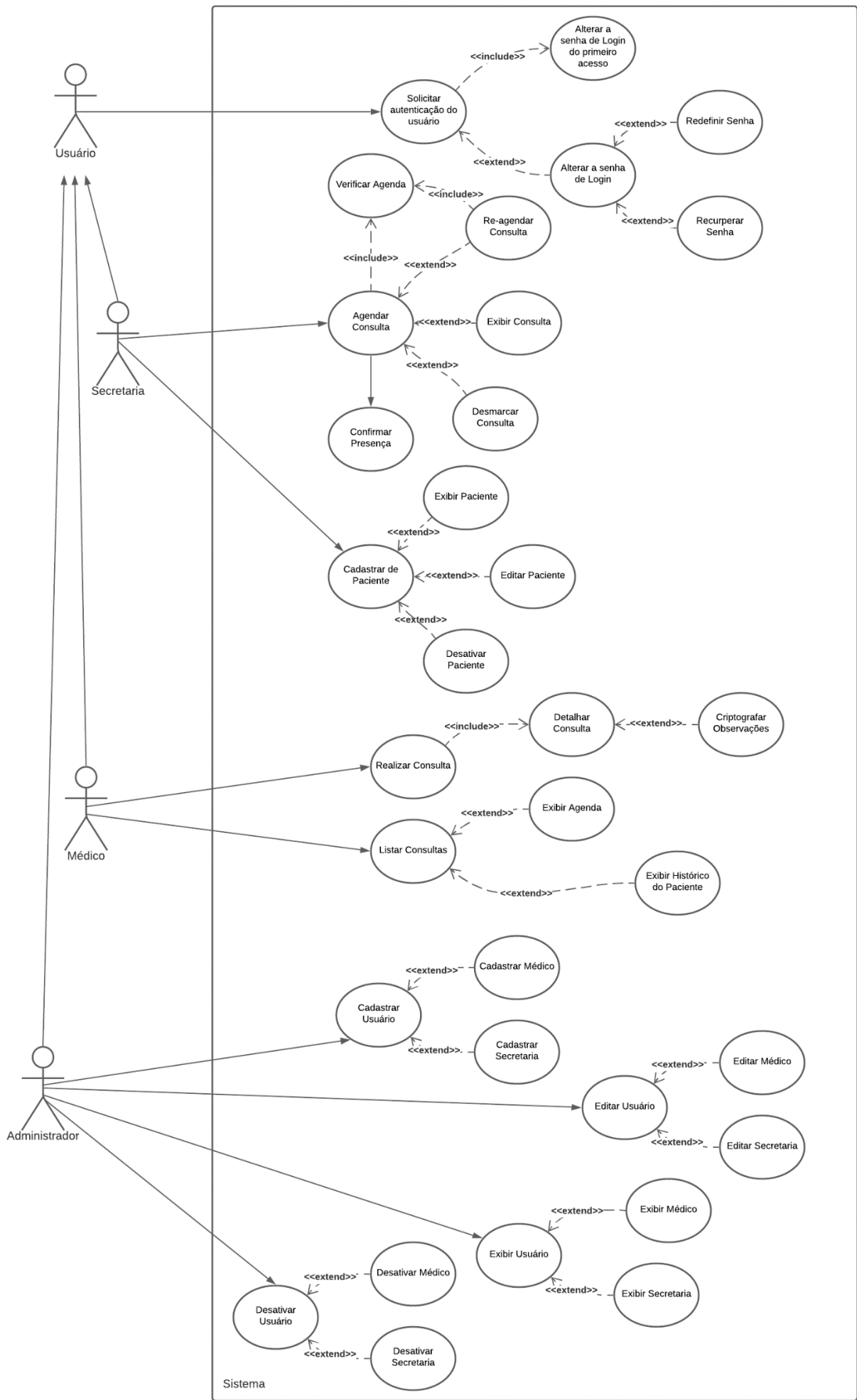
## Diagrama de Implantação



O *software* ConsultEasy opera em um único computador, sendo composto apenas por suas interfaces visuais, suas classes e pela integração com o banco de dados. É necessária conexão com *internet*, porém sem outros computadores para a sua execução.

ConsultEasy: O primeiro módulo do sistema, é responsável pela interface visual do sistema e pela manipulação dos dados. Já o segundo módulo, bem como responsável pelo armazenamento das informações dos usuários, pacientes e das consultas no banco de dados. Escrita utilizando a linguagem C#, HTML (Linguagem de Marcação e Hipertexto), CSS, JavaScript, Bootstrap 4, utilizando o framework Docker para realizar a conexão com o Base de dados.

Diagrama de casos de uso



### **Usuário - funções comuns para todos os tipos de usuários.**

O ator usuário pode logar no sistema e será feita a autenticação, caso seja seu primeiro acesso, será solicitada a alteração da senha do usuário.

Após o primeiro acesso, o usuário tem a possibilidade de alteração da própria senha de login, além da opção de recuperar a senha caso esqueça.

### **Secretaria**

O ator Secretaria tem, além das funcionalidades do ator Usuário, as opções:

Agendar consulta: serão informados os dados pessoais de cada paciente após informar o médico, data e horário. No momento de agendar a consulta, será feita pelo sistema, a verificação da agenda, com base em consultas previamente cadastradas.

Exibir consulta: será exibida uma lista com todas as consultas e seus respectivos pacientes de acordo com o médico, a data e o horário.

Desmarcar consulta: será possível realizar o cancelamento da consulta.

Reagendar consulta: será possível realizar o reagendamento da consulta para uma data disponível, de acordo com a verificação da agenda, que será feita pelo sistema, com base em consultas previamente cadastradas.

Cadastrar paciente: o cadastro do paciente será realizado através do preenchimento dos dados pessoais de cada paciente.

Editar paciente: é possível editar todos os cadastros dos pacientes.

Desativar paciente: é possível desativar um paciente, para os casos onde os pacientes se desligam da clínica.

### **Médico**

O ator Médico tem, além das funcionalidades do ator Usuário, as opções:

Realizar consulta: o ator Médico poderá realizar consultas.

Detalhar consulta: a consulta deve ser obrigatoriamente iniciada pelo ator Médico e o mesmo deverá fazer o seu detalhamento/preenchimento.

Criptografar observações: o ator Médico terá a possibilidade de criptografar ou não as observações por ele descritas durante a realização da consulta.

Listar consultas: o ator Médico pode listar as consultas.

Exibir agenda: há a possibilidade de listar todas as consultas pendentes do ator Médico.

Exibir histórico do paciente: o ator Médico pode listar todas as consultas já finalizadas de determinado paciente.

## **Administrador**

O ator Administrador tem, além das funcionalidades do ator Usuário, as opções:

Cadastrar usuário: o ator Administrador pode cadastrar um usuário no sistema.

Cadastrar médico: poderá ser feito o cadastro de um usuário do tipo médico.

Cadastrar secretaria: poderá ser feito o cadastro de um usuário do tipo secretaria.

Editar usuário: o ator Administrador pode editar um usuário no sistema.

Editar médico: poderá ser feita a edição dos dados de um usuário do tipo médico, exceto a alteração de sua senha.

Editar secretaria: poderá ser feita a edição dos dados de um usuário do tipo secretaria, exceto a alteração de sua senha.

Exibir usuário: o ator Administrador pode listar os usuários do sistema.

Exibir médico: poderá ser feita a listagem de todos os usuários do tipo médico.

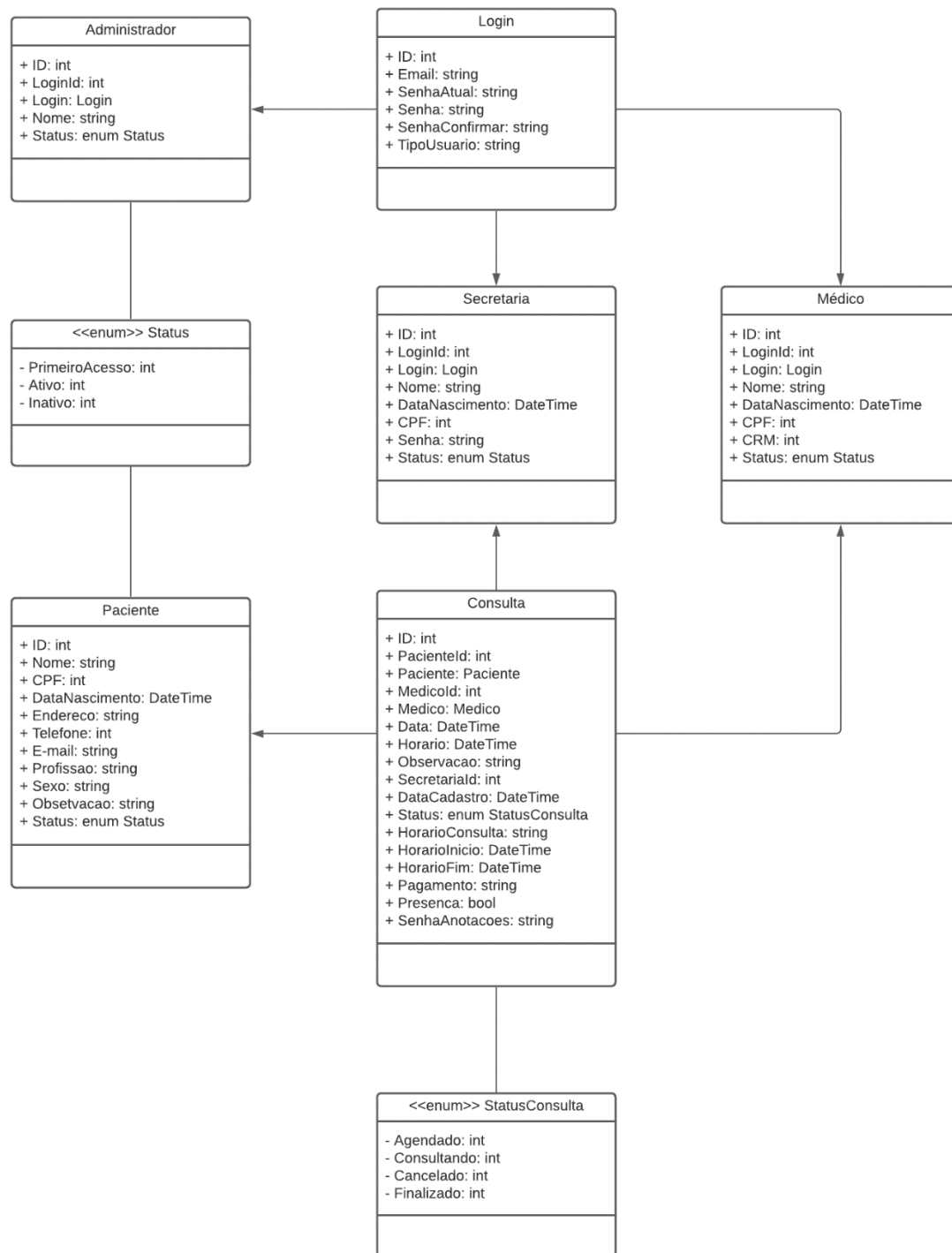
Exibir secretaria: poderá ser feita a listagem de todos os usuários do tipo secretaria.

Desativar usuário: o ator Administrador pode desativar um usuário no sistema.

Desativar médico: poderá ser feita a desativação de um usuário do tipo médico.

Desativar secretaria: poderá ser feita a desativação de um usuário do tipo secretaria.

## Diagrama de classes



Paciente: classe que possui onze propriedades, ID, Nome, CPF, DataNascimento, Endereço, Telefone, E-mail, Profissao, Sexo, Observação e Status que devem ser utilizadas para armazenar as informações do objeto do tipo Paciente que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- Nome: Uma propriedade do tipo string, que recebe o nome do paciente;
- CPF: Uma propriedade do tipo inteiro de forma única(unique), que recebe o cpf do paciente;

- DataNascimento: Uma propriedade do tipo DateTime que recebe a data de nascimento do paciente;
- Endereço: Uma propriedade do tipo string, que recebe o endereço do paciente;
- Telefone: Uma propriedade do tipo int, que recebe o telefone do paciente;
- E-mail: Uma propriedade do tipo string, que recebe o e-mail do paciente;
- Profissao: Propriedade do tipo string, que recebe a profissão do paciente;
- Sexo: Uma propriedade do tipo string, que recebe o sexo do paciente;
- Observação: Uma propriedade do tipo string, que recebe uma observação opcional.

Medico: classe que possui oito propriedades, ID, LoginId, Login, Nome, DataNascimento, CPF, CRM e Status que devem ser utilizadas para armazenar as informações do objeto do tipo Secretaria que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- LoginId: Uma propriedade do tipo int, que receberá o id do login cadastrado;
- Login: Uma propriedade do tipo Login, que receberá um objeto com as informações de um Login cadastrado em um banco de dados;
- Nome: Uma propriedade do tipo string, que recebe o nome do médico;
- DataNascimento: Uma propriedade do tipo DateTime que recebe a data de nascimento do médico;
- CPF: Uma propriedade do tipo inteiro de forma única(unique), que recebe o cpf do médico;
- CRM: Uma propriedade do tipo int, que recebe o CRM do médico;
- Status: Uma propriedade do tipo enum Status, que recebe uma enumeração do status atual do usuário.

Secretaria: classe que possui sete propriedades, ID, LoginId, Login, Nome, DataNascimento, CPF, e Status que devem ser utilizadas para armazenar as informações do objeto do tipo Secretaria que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- LoginId: Uma propriedade do tipo int, que receberá o id do login cadastrado;
- Login: Uma propriedade do tipo Login, que receberá um objeto com as informações de um Login cadastrado em um banco de dados;
- Nome: Uma propriedade do tipo string, que recebe o nome do usuário secretaria;
- DataNascimento: Uma propriedade do tipo DateTime que recebe a data de nascimento do usuário secretaria;
- CPF: Uma propriedade do tipo inteiro de forma única(unique), que recebe o cpf do usuário secretaria;
- Status: Uma propriedade do tipo enum Status, que recebe uma enumeração do status atual do usuário.

Administrador: classe que possui seis propriedades, ID, Nome, E-mail, NomeUsuario, Senha e Status, que devem ser utilizadas para armazenar as informações do objeto do tipo Administrador que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- Nome: Uma propriedade do tipo string, que recebe o nome do administrador;
- LoginId: Uma propriedade do tipo int, que receberá o id do login cadastrado;
- Login: Uma propriedade do tipo Login, que receberá um objeto com as informações de um Login cadastrado em um banco de dados;

- Status: Uma propriedade do tipo enum Status, que recebe uma enumeração do status atual do administrador.

Status: enumeração que possui três itens, PrimeiroAcesso, Ativo, Inativo.

- PrimeiroAcesso: atribui o status “PrimeiroAcesso” para o usuário que foi cadastro, porém não efetuou o primeiro acesso;
- Ativo: atribui o status “Ativo” para o usuário que for cadastrado;
- Inativo: atribui o status “Inativo” para o usuário que não pertence mais o acesso ao sistema.

Login: classe que possui seis propriedades, ID, E-mail, SenhaAtual, Senha, SenhaConfirmar e TipoUsuario, que devem ser utilizadas para armazenar as informações do objeto do tipo Login que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- Email: Uma propriedade do tipo string, que recebe o e-mail do usuário;
- SenhaAtual: Uma propriedade do tipo string, que recebe a senha atual do usuário;
- Senha: Uma propriedade do tipo string, que recebe a senha do usuário;
- SenhaConfirmar: Uma propriedade do tipo string, que recebe a senha que virá ser de confirmação do usuário;
- TipoUsuário: Uma propriedade do tipo string, que recebe o tipo de usuário cadastrado naquele login.

Consulta: classe que possui dezessete propriedades, ID, PacienteId, Paciente, MedicId, Medico, Data, Horario, Observacao, Secretariald, DataCadastro, Status, HorarioConsulta, HorarioInicio e HorarioFim, Pagamento, Presenca e SenhaAtendimento que devem ser utilizadas para armazenar as informações do objeto do tipo Consulta que virá a ser instanciado.

- ID: Uma propriedade do tipo inteiro, que será autoincrementado no banco de dados;
- PacienteId: Uma propriedade do tipo int, que receberá o id do paciente previamente cadastrado;
- Paciente: Uma propriedade do tipo Paciente, que receberá um objeto com as informações de um Paciente cadastrado em um banco de dados;
- MedicId: Uma propriedade do tipo int, que receberá o id do médico previamente cadastrado;
- Medico: Uma propriedade do tipo Medico, que receberá um objeto com as informações de um Médico cadastrado em um banco de dados;
- Data: Uma propriedade do tipo DateTime, que recebe a data da consulta;
- Horario: Uma propriedade do tipo DateTime, que recebe a hora da consulta;
- Observacao: Uma propriedade inicialmente nula do tipo string, que pode ser alterada após o início de uma consulta;
- Status: Uma propriedade do tipo enum StatusConsulta, que receberá o status de uma consulta;
- Secretariald: Uma propriedade do tipo int, que receberá o id da secretaria previamente cadastrada;
- DataCadastro: Uma propriedade do tipo DateTime, que recebe a data do cadastro da consulta;
- Status: Uma propriedade do tipo enum StatusConsulta, que recebe uma enumeração do status atual da consulta;



- **HorarioConsulta:** Uma propriedade do tipo string, que recebe a hora da consulta para uma posterior exibição da consulta na data grid;
- **HorarioInicio:** Uma propriedade do tipo DateTime, que recebe a hora do início da consulta. Salvo apenas para registro da consulta;
- **HorarioFim:** Uma propriedade do tipo DateTime, que recebe a hora do fim da consulta. Salvo apenas para registro da consulta;
- **Pagamento:** Uma propriedade do tipo string, que armazena a forma de pagamento ao realizar a consulta;
- **Presença:** Uma propriedade do tipo bool, que armazena se o paciente estará presente ou não, sendo true para sim e false para não;
- **SenhaAnotacoes:** Uma propriedade do tipo string, que armazena o hash duplo da senha informada pelo sistema.

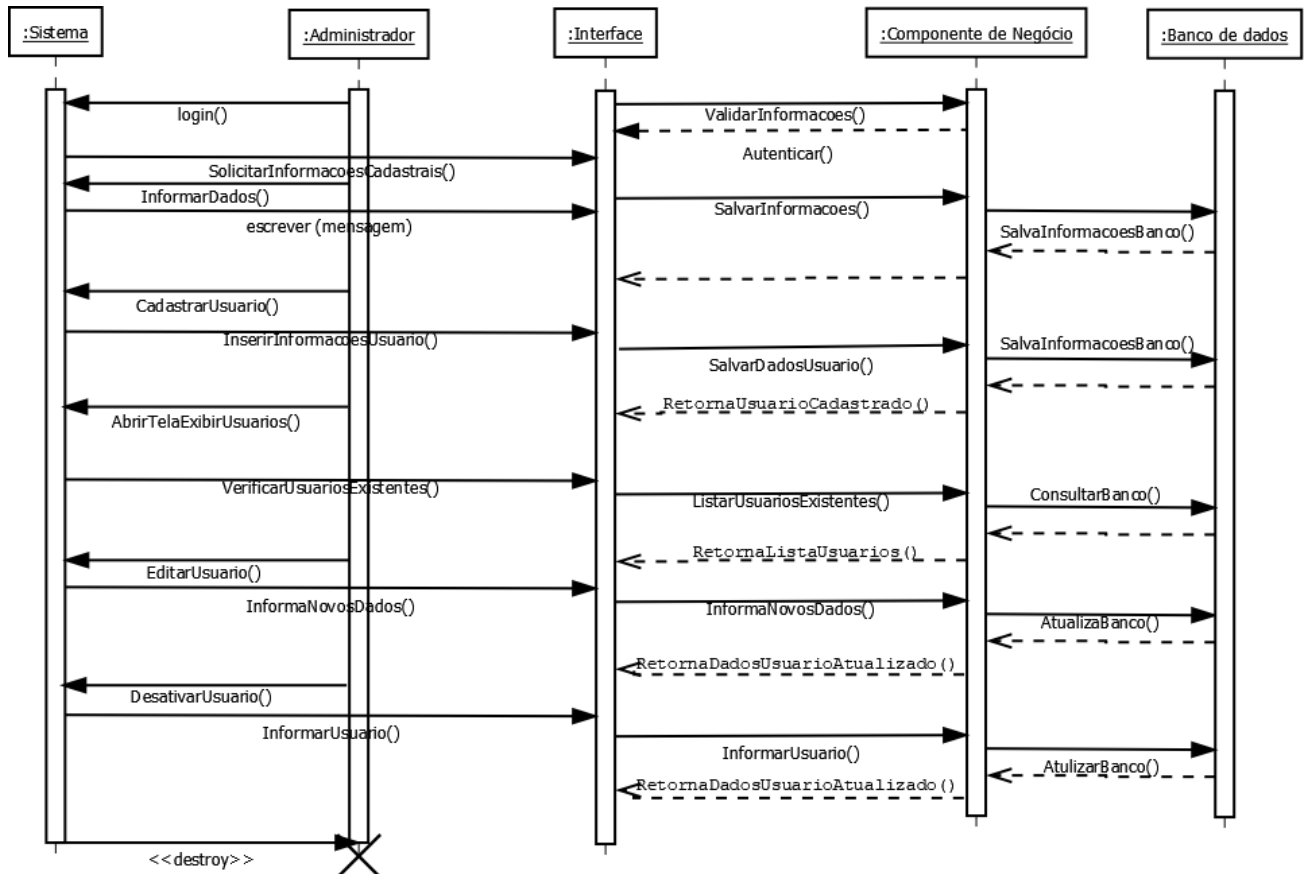
**StatusConsulta:** enumeração que possui quatro itens, Agendado, Consultando, Cancelado e Finalizado.

- **Agendado:** atribui o status “Agendado” para a consulta que for agendada;
- **Consultando:** atribui o status “Consultando” para a consulta que foi iniciada, porém não finalizada;
- **Cancelado:** atribui o status “Cancelado” para a consulta que foi iniciada, porém cancelada;
- **Finalizado:** atribui o status “Finalizado” para a consulta que foi encerrada.

## Diagrama de sequência

### Administrador

#### Primeiro acesso, Cadastro, Edição e Inativação de Usuários

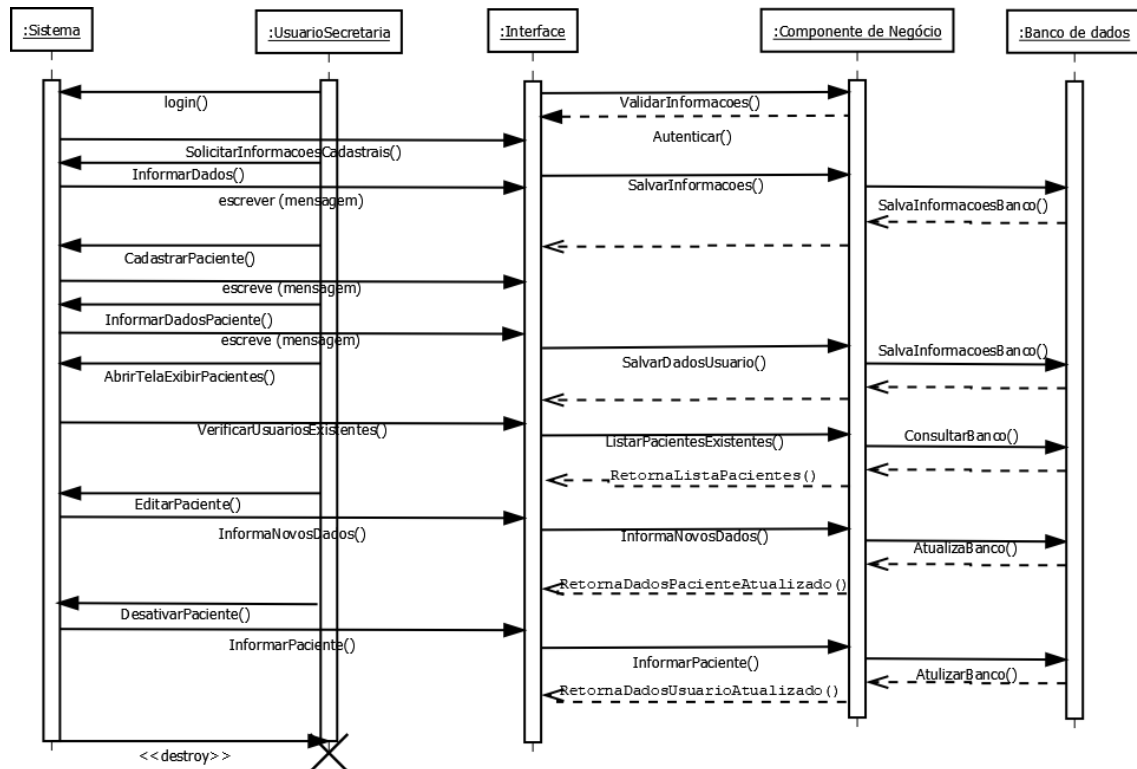


O diagrama apresenta 5 componentes sendo Sistema, Administrador, Interface, Componente de Negócio e Banco de dados. Nele é descrito o primeiro acesso ao sistema do administrador no qual valida as informações, faz a autenticação e solicita que o administrador atualize seus dados. Após isso, mostra o fluxo do processo entre as camadas para a inserção dos dados até o banco. Logo, mostra a sequência do processo de cadastrar um usuário, no qual é informado a interface, que passa pela regra de negócio, armazena no banco e retorna o usuário cadastrado.

É apresentado também, o fluxo para exibir uma lista de usuários previamente cadastrados no sistema, verificando os existentes e retornando a lista, podendo até editar um usuário, no qual são passados os novos dados e atualizados no banco, retornando o usuário atualizado no sistema. Por fim é possível através do administrador ver o fluxo de dados para a inativação do usuário, atualizando o status para inativo no banco, e retornando o usuário com o status alterado.

## Secretaria

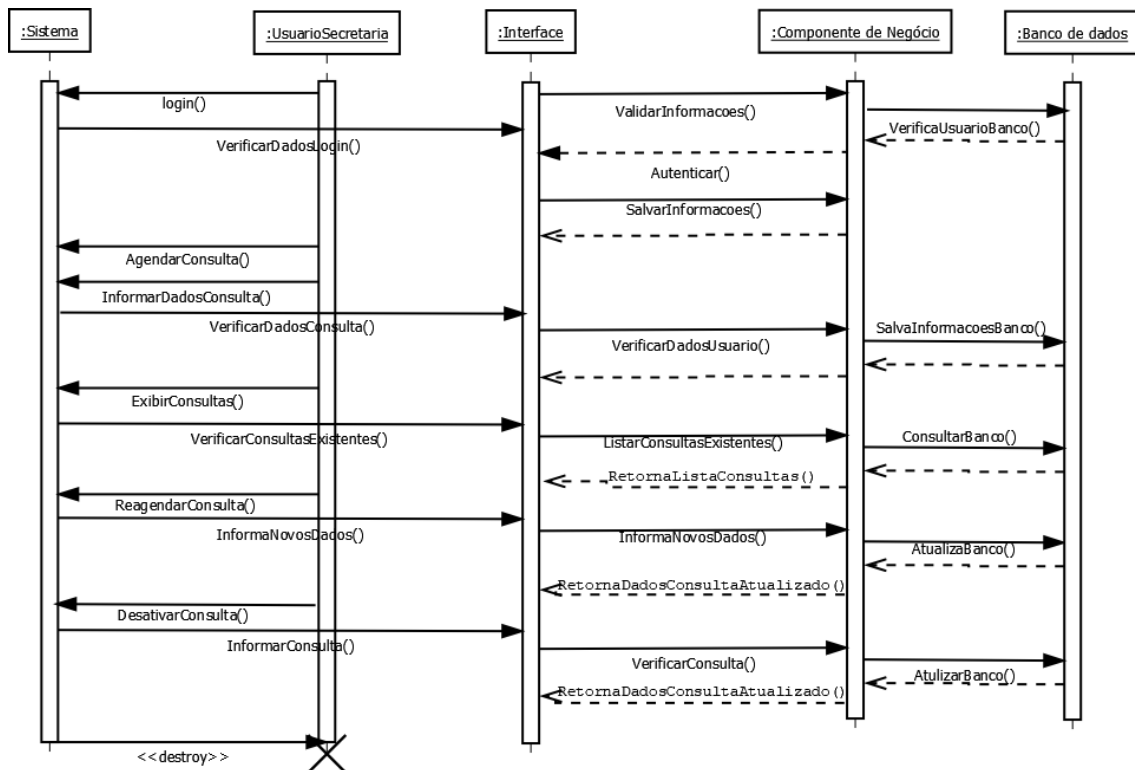
### Primeiro acesso ao sistema, Cadastro, Edição e Inativação de Pacientes



Neste diagrama da secretaria apresenta 5 componentes sendo Sistema, UsuarioSecretaria, Interface, Componente de Negócio e Banco de dados. É descrito o primeiro acesso ao sistema do usuario da secretaria no qual valida as informações e atualiza a senha. Após isso, mostra a sequência do processo de cadastrar um paciente, no qual é informado a interface, que passa pela regra de negócio, armazena no banco e retorna o paciente cadastrado. É apresentado também, o fluxo para exibir uma lista de pacientes previamente cadastrados no sistema, verificando os existentes e retornando a lista, podendo até editar um paciente, no qual são passados os novos dados e atualizados no banco, retornando o paciente atualizado no sistema.

Por fim é possível através do usuário ver o fluxo de dados para a inativação do paciente, atualizando o status para inativo no banco, e retornando o usuário com o status alterado.

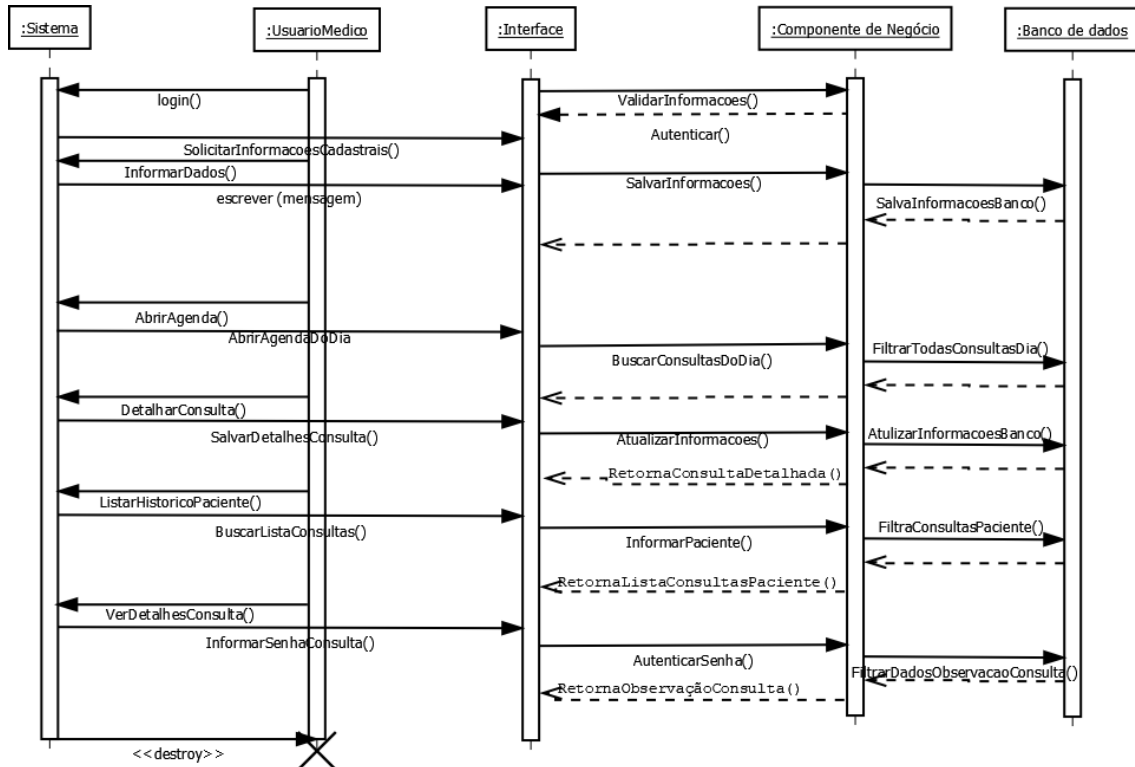
## Cadastro, Edição e Cancelamento de Consultas



Neste diagrama da secretaria apresenta 5 componentes sendo Sistema, UsuarioSecretaria, Interface, Componente de Negócio e Banco de dados. É descrito o login ao sistema do usuário da secretaria. Após isso, mostra a sequência do processo de cadastrar uma consulta, no qual é informado a interface, que passa pela regra de negócio, armazena no banco e retorna à consulta agendada. É apresentado também, o fluxo para exibir uma lista de consultas previamente cadastradas no sistema e retornando a lista, podendo até reagendar uma consulta, no qual são passados os novos dados e atualizados no banco, retornando à consulta atualizada e reagendada no sistema. Por fim é possível através do usuário ver o fluxo de dados para o cancelamento de uma consulta, atualizando o status para cancelado no banco.

## Médico

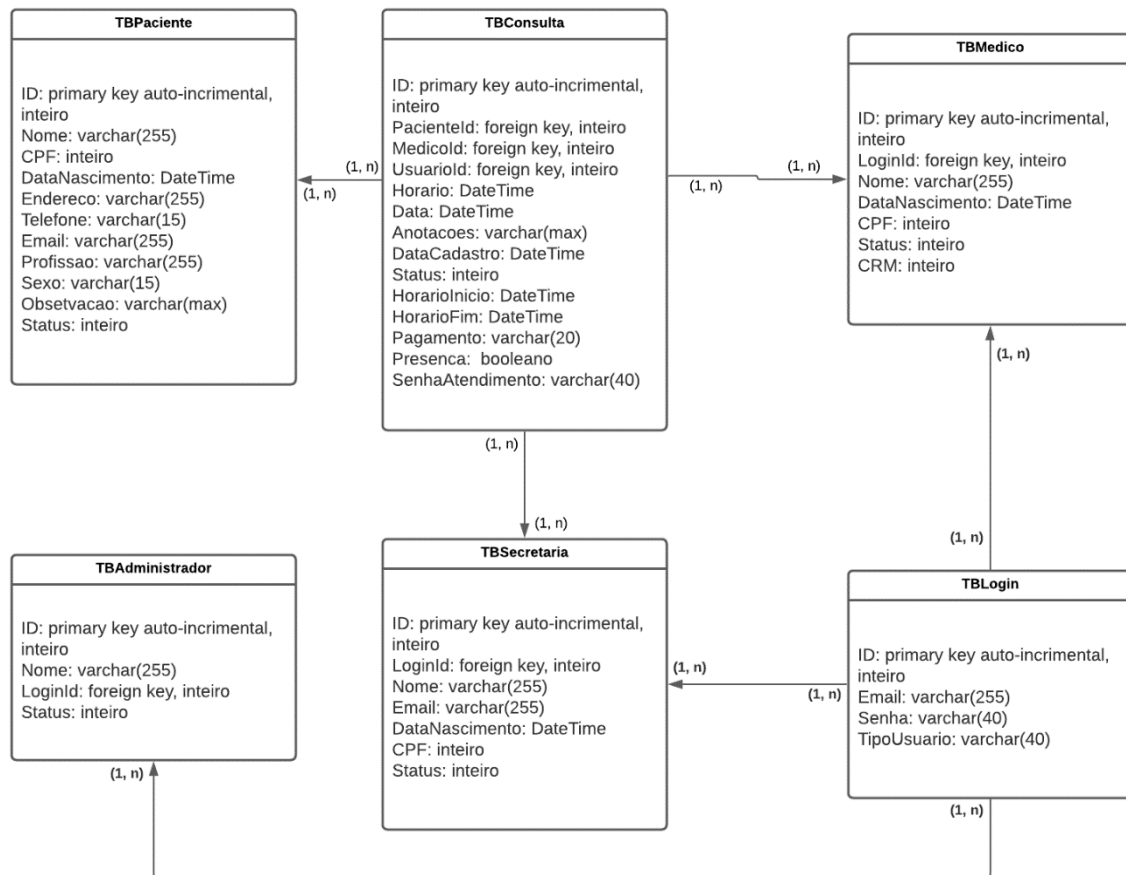
### Primeiro Acesso, Detalhamento de Consulta, Histórico de Paciente e Ver Detalhes da Consulta



O diagrama do médico apresenta 5 componentes sendo Sistema, UsuarioMedico, Interface, Componente de Negócio e Banco de dados. Nele é descrito o primeiro acesso ao sistema do usuário do tipo médico no qual valida as informações e atualiza a senha. Após isso, mostra o fluxo do processo entre as camadas para a inserção dos dados até o banco. Logo, mostra a sequência do processo de exibir a agenda do dia, verificando as consultas do dia e retornando uma lista de consultas do médico respectivo. É apresentando também, o processo de detalhar uma consulta, no qual é passado pelo usuário as informações, enviado para a regra de negócio e por fim armazenado no banco e atualizando a consulta.

É possível que o médico veja o histórico do paciente, informando o paciente e retornando uma lista de consultas do paciente respectivo. Por fim é possível através do diagrama ver o fluxo de dados para ver os detalhes da consulta de um paciente, informando a senha, autenticando e retornando as observações descriptografada para o médico.

## Diagrama relacional do banco de dados



TBPaciente: tabela que possui onze atributos, ID, Nome, CPF, DataNascimento, Endereco, Telefone, Email, Profissao, Sexo, Observação e Status que devem ser utilizadas para armazenar as informações do objeto do tipo Paciente do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- Nome: Um atributo do tipo varchar, não nulo, que armazenará o nome do paciente;
- CPF: Um atributo do tipo inteiro, não nulo, que armazenará o cpf do paciente;
- DataNascimento: Um atributo do tipo DateTime, não nulo, que armazenará a data de nascimento do paciente;
- Endereco: Um atributo do tipo varchar, não nulo, que armazenará o endereço do paciente;
- Telefone: Um atributo do tipo inteiro, não nulo, que armazenará o telefone do paciente;
- Email: Um atributo do tipo varchar, não nulo, que armazenará o email do paciente;
- Profissao: Um atributo do tipo varchar, não nulo, que recebe a profissão do paciente;
- Sexo: Um atributo do tipo varchar, não nulo, que recebe o sexo do paciente;
- Observação: Um atributo do tipo varchar, nulo, que recebe a observação do paciente;
- Status: Um atributo do tipo inteiro, não nulo, que recebe uma enumeração do status do paciente.

TBMedico: tabela que possui sete atributos, ID, LoginId, Nome, DataNascimento, CPF, Status e CRM que devem ser utilizadas para armazenar as informações do objeto do tipo Médico do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- LoginId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBLogin para a obtenção dos dados), não nulo, que receberá o id do login cadastrado;
- Nome: Um atributo do tipo varchar, não nulo, que armazenará o nome do médico;
- DataNascimento: Um atributo do tipo DateTime, não nulo, que armazenará a data de nascimento do médico;
- CPF: Um atributo do tipo inteiro, unique, não nulo, que armazenará o cpf do médico;
- Status: Um atributo do tipo inteiro, não nulo, que recebe uma enumeração do status do médico;
- CRM: Um atributo do tipo inteiro, unique, não nulo, que armazenará o CRM do médico.

TBConsulta: tabela que possui catorze atributos, ID, PacienteId, MedicId, UsuarioId, Horario, Data, Anotacoes, DataCadastro, Status, HorarioInicio, HorarioFim, Pagamento, Presenca e SenhaAtendimento que devem ser utilizadas para armazenar as informações do objeto do tipo Consulta do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- PacienteId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBPaciente para a obtenção dos dados), não nulo, que receberá o id do paciente previamente cadastrado;
- MedicId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBMedico para a obtenção dos dados), não nulo, que receberá o id do médico previamente cadastrado;
- UsuarioId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBSecretaria para a obtenção dos dados), não nulo, que receberá o id do usuário do tipo secretaria previamente cadastrado;
- Horario: Um atributo do tipo DateTime, não nulo, que armazenará a hora da consulta;
- Data: Um atributo do tipo DateTime, não nulo, que armazenará a data da consulta;
- Anotacoes: Um atributo inicialmente nulo do tipo varchar;
- DataCadastro: Um atributo do tipo DateTime, não nulo, que armazenará a data da consulta;
- Status: Um atributo do tipo inteiro, que armazenará o índice do enum StatusConsulta;
- HorarioFim: Um atributo do tipo DateTime, inicialmente nulo, que armazenará a hora do fim da consulta;
- HorarioInicio: Um atributo do tipo DateTime, inicialmente nulo, que armazenará a hora do início da consulta;
- Pagamento: Um atributo do tipo varchar, não nulo, que armazenará a forma de pagamento da consulta;
- Presenca: Um atributo do tipo booleano, não nulo, que armazenará o se o paciente irá realizar a consulta ou não.
- SenhaAtendimento: Um atributo do tipo varchar, inicialmente nulo, que irá armazenar o hash duplo da senha do médico para acessar as observações da consulta.

TBSecretaria: tabela que possui seis atributos, ID, LoginId, Nome, DataNascimento, CPF, Status que devem ser utilizadas para armazenar as informações do objeto do tipo Secretaria do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- LoginId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBLogin para a obtenção dos dados), não nulo, que receberá o id do login cadastrado;
- Nome: Um atributo do tipo varchar, não nulo, que armazenará o nome do usuário do tipo secretaria;
- DataNascimento: Um atributo do tipo DateTime, não nulo, que armazenará a data de nascimento do usuário do tipo secretaria;
- CPF: Um atributo do tipo inteiro, não nulo, que armazenará o cpf do usuário do tipo secretaria;
- Status: Um atributo do tipo inteiro, não nulo, que recebe uma enumeração do status do o usuário do tipo secretaria.

TBAdministrador: tabela que possui quatro atributos, ID, LoginId, Nome e Status que devem ser utilizadas para armazenar as informações do objeto do tipo Administrador do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- LoginId: Um atributo do tipo inteiro, chave estrangeira (Fará o relacionamento com a tabela TBLogin para a obtenção dos dados), não nulo, que receberá o id do login cadastrado;
- Nome: Um atributo do tipo varchar, não nulo, que armazenará o nome do administrador;
- Status: Um atributo do tipo inteiro, não nulo, que recebe uma enumeração do status do administrador.

TBLogin: classe que possui quatro propriedades, ID, Email, Senha e TipoUsuario, que devem ser utilizadas para armazenar as informações do objeto do tipo Login do sistema no banco de dados.

- ID: Um atributo do tipo inteiro, de chave primária, não nulo que será autoincrementado;
- Email: Um atributo do tipo varchar, não nulo, unique, que armazenará o email do usuario;
- Senha: Um atributo do tipo varchar, não nulo, que armazenará o hash duplo da senha do usuário;
- TipoUsuário: Uma propriedade do tipo varchar, que armazenará o tipo de usuário cadastrado.

Relacionamentos:

- TBConsulta – TBMedico: (1, n) Um médico pode estar relacionado a várias consultas, mas uma consulta, poderá possuir apenas um único médico;
- TBConsulta – TBPaciente: (1, n) Um paciente pode estar relacionado a várias consultas, mas uma consulta, poderá possuir apenas um único paciente.
- TBConsulta – TBSecretaria: (1, n) Um usuário do tipo secretaria pode estar relacionado a várias consultas, mas uma consulta, poderá possuir apenas um usuário do tipo secretaria que o realizou o cadastro da mesma.



- TBSecretaria – TBLLogin: (1, n) Um login pode estar relacionado a vários usuários do tipo secretaria, mas um secretário, poderá possuir apenas um login que o realizou o cadastro do mesmo.
- TBMedico – TBLLogin: (1, n) Um login pode estar relacionado a vários usuários do tipo médico, mas um médico, poderá possuir apenas um login que o realizou o cadastro do mesmo.
- TBAdministrador – TBLLogin: (1, n) Um login pode estar relacionado a vários usuários do tipo secretaria, mas um administrador, poderá possuir apenas um login que o realizou o cadastro do mesmo.

## Requisitos Funcionais

RF01: Ao acessar o site, será feito a verificação se algum administrador já se autenticou ao menos uma vez no site. Caso seja o primeiro acesso, a página apresentará uma tela de login, com um formulário contendo dois inputs do tipo "text" para informar a senha e o login previamente definidos pelo desenvolvedor como "admin" para ambos os campos e um input do tipo "button" com o nome de "Login" para enviar o formulário.

Se os dados informados nos campos forem incorretos, será disparada uma mensagem de alerta informando que os dados estão incorretos e não os aceitarão, portanto não será feito o primeiro acesso.

Quando o formulário for enviado com sucesso (dados informados corretos), será mudado o valor do atributo "Status" da tabela "TBAadministrador" do banco de dados, de "PrimeiroAcesso"(que representa não logado) para "Ativo"(que representa logado).

RF02: Todas as senhas seja para administrador ou usuário, devem conter ao menos 8 caracteres e as características seguintes: letras em minúsculas, letras em maiúsculas, ao menos um caractere alfanumérico (como %, &, !) e ao menos um número. Além disso, deverão ser convertidas e armazenadas no banco de dados com o padrão SHA 256 e hash duplo.

RF03: Se o valor do atributo "Status" da tabela "TBAadministrador" for "PrimeiroAcesso", o primeiro acesso ainda não foi feito, portanto, será requisitada a alteração da senha do administrador, através de uma janela pop-up com quatro campos input do tipo "text" para informar o nome completo do administrador, e-mail, a nova senha e a confirmação da nova senha, além disso, deve conter um input do tipo "submit", ou seja, um botão para enviar o formulário e salvar as informações. O formulário só será enviado caso os dois campos informados sejam exatamente iguais e atendam aos requisitos de criptografia. Caso a senha não atenda aos padrões requisitados, o sistema irá retornar uma mensagem de erro e não irá aceitar a senha definida.

Caso a senha seja aceita, será mudado o valor do atributo "Status" da tabela "TBAadministrador" do banco de dados, de "PrimeiroAcesso"(que representa nunca autenticado) para "Ativo"(que representa autenticado). O usuário administrador será autenticado e não será necessário mais fazer o login a menos que a sessão seja encerrada.

Caso algum administrador já tenha se autenticado, o usuário será redirecionado para a tela de login, onde poderá autenticar com uma conta de administrador, médico ou secretaria.

RF04: Após a autenticação no sistema, o administrador será redirecionado a página principal que contém, um menu com as funcionalidades dos sistemas e submenus para direcionar as páginas para estas funcionalidades.

O cabeçalho contará com um botão para as informações do usuário, este botão quando clicado apresentará um menu "dropdown" com as opções "redefinir senha" e "sair". Quando clicado na opção "redefinir senha", será carregada na página de informações do usuário que contém dois campos input do tipo "text" para exibir o nome e e-mail, e mais três campos para informar

a “senha atual”, a “nova senha” e a “confirmação da nova senha”, caso o administrador queira alterar, além disso, deve conter um input do tipo "button" para atualizar os dados e um link para redefinição de senha com o nome “Esqueci minha senha”.

Ao clicar no botão para a alteração da senha, se a senha digitada no campo de “senha atual” (aplicando as regras de criptografia) forem idênticas a senha armazenada no banco e as senhas digitadas nos campos “nova senha” e “confirmação da nova senha” forem idênticas e atenderem as regras de criptografia, será alterada a senha do banco de dados e o administrador será desautenticado e encaminhado para a tela de login. Caso a menos um dos três campos não atendam as regras de criptografia ou às comparações citadas anteriormente, será disparada uma mensagem de erro informando os campos a serem corrigidos.

Ao clicar na opção “Esqueci minha senha” será enviado um e-mail com um link temporário de redefinição de senha para o e-mail cadastrado no primeiro acesso do administrador. Este link permite a alteração da senha sem a necessidade da senha atual, ao clicar no link, o usuário é redirecionado a uma página contendo dois campos input do tipo “text” para informar a nova senha e a confirmação da nova senha. A nova senha deverá ter os dois campos informados exatamente iguais e atendam aos requisitos de criptografia. Caso a senha seja aceita, o administrador terá sua sessão encerrada, sendo necessário fazer a autenticação no sistema com a nova senha.

Quando clicado no botão “Sair”, o usuário será redirecionado para a tela de login.

RF05: Na página, terá um menu, no qual apresentará uma opção de “Médicos”. Ao clicar na opção, será exibido uma lista de todos os médicos cadastrados. Ao clicar em “Criar um novo usuário” será aberto um formulário no site para cadastrar médico, no qual será visível 5 campos do tipo “text” para que seja inserida as informações dele como, Nome, CRM, E-mail, Data de Nascimento e CPF. Terá um input do tipo “submit”, ou seja, um botão que fará o envio das informações ao banco de dados, no qual alertará ao administrador de que o usuário foi cadastrado.

Caso o administrador selecione “Secretarios” no select, o campo CRM não estará mais visível para armazenar informações, possibilitando informar apenas Nome, E-mail, Data de Nascimento, CPF. Todos os campos são obrigatórios, portanto, se houver ao menos um campo sem informação nenhuma, ele deverá alertar o usuário de qual campo deve ser preenchido. O campo do CPF terá um validador e caso o CPF digitado não seja válido, o usuário será alertado a digitar um que seja de fato válido. Caso não haja problemas com a entrada, o sistema deverá salvar as informações do usuário e armazenar no banco de dados. Após a criação do usuário, será disparado um e-mail com destino ao e-mail cadastrado para que o usuário possa se autenticar no sistema posteriormente.

RF06: Ao clicar em “Médicos” no menu ou “Secretários”, é será apresentado na div principal uma tabela com todos os usuários registrados no sistema do respectivo tipo, podendo filtrar por Nome, CPF e se for Médico, CRM. Nesta tabela haverá todas as informações do usuário. Embaixo do nome de cada usuário da tabela, será visível duas opções, “Editar” e “Remover”. Ao clicar em “Editar” o administrador será redirecionado para uma página no qual conterá as informações do respectivo usuário selecionado, no qual possibilitará apenas a edição de todos os campos

Nome, CRM, E-mail, Data de Nascimento, CPF, nome de usuário, se for médico ou somente Nome, E-mail, Data de Nascimento, CPF, nome de usuário, se for da Secretaria. A senha não estará disponível para alterar pois somente o usuário pertencente aquela conta poderá alterá-la, o administrador não terá permissão. Todos os campos devem estar preenchidos para que possa ser atualizado os dados. Caso não haja problemas com a entrada, o sistema deverá atualizar as informações do usuário no banco de dados.

Caso o administrador selecione a opção “Desativar usuário”, o sistema deverá apresentar uma mensagem alertando se de fato o administrador queira desativar o usuário, se sim, o usuário terá o seu Status alterado de “Ativo” para “Inativo”, no qual irá finalizar a sessão do usuário e não permitirá que ele faça o acesso ao sistema. Se não, o usuário permanecerá com o Status “Ativo” normalmente, podendo ter acesso ao sistema.

RF07: Para que seja possível realizar o primeiro acesso no sistema, o usuário comum receberá um link em seu e-mail. Este link irá conter uma chave de acesso como se fosse uma “senha pré-definida” pelo administrador. Ao clicar no link, ele será redirecionado para a tela de Login.

Nesta tela, será exibido um formulário solicitando a mudança de sua senha em seu primeiro acesso por questões de segurança. O formulário conterá os campos, senha atual, nova senha e confirmar a nova senha.

Também haverá um botão chamado Acessar, e logo abaixo as especificações de como deverá ser a senha a nível de segurança da informação, atendendo a todos os requisitos para a construção de uma senha forte, os títulos possuirão um asterisco “\*” em vermelho para especificar as características que a senha deve possuir os requisitos de criptografia.

Após realizado o primeiro acesso no sistema, no canto superior direito será exibido um Login juntamente com o título: “SEJA BEM VINDO, nome do seu usuário”. Neste campo, haverá uma seta indicada para baixo que exibirá uma listagem de opções. Levando em consideração que a única opção permitida para um usuário comum será alterar a própria senha, existirá apenas as opções “alterar senha” e “sair da conta”.

RF08: Para os casos onde o usuário tem conhecimento de sua senha atual e deseja realizar a sua mudança, ao clicar na opção “alterar senha”, o usuário será redirecionado para um formulário que conterá 3 inputs do tipo “text”, será solicitado digitar sua senha atual, a nova senha e a confirmação da nova senha, além de um input do tipo “submit”, ou seja, um botão escrito “salvar”, para salvar as alterações.

Além disso, existirá um texto informando que o usuário não poderá utilizar a mesma senha anterior. Após clicar no botão de Salvar, será realizado o Logoff automático do sistema, para que o usuário possa realizar um novo acesso utilizando a nova senha.

RF09: Nos casos em que o usuário não se recordar de sua senha, na tela inicial de Login do sistema existirá a opção abaixo do campo de efetuar login, um título clicável com o nome: Recuperar sua senha através do E-mail. Após o usuário clicar neste link, será redirecionado a uma página contendo um formulário que solicitará o endereço de sua conta de E-mail para que seja enviado um link de redefinição de senha. Neste formulário, existirá dois campos input do

tipo “text”, “informar seu endereço de e-mail” e “confirmar seu endereço de e-mail” e um input do tipo “submit”, ou seja, um botão com o nome “Enviar”. Ao clicar em “Enviar”, será exibida uma mensagem indicando ao usuário que deverá confirmar o recebimento do e-mail enviado em sua caixa de entrada para recuperar a senha.

Além disso, também será exibido um texto informando que o link expira em 10 minutos. Caso o usuário não o acesse e realize a troca de sua senha dentro desses minutos, o mesmo deverá solicitar um novo link de redefinição. Será encaminhado o mesmo link enviado anteriormente, com o prazo de expiração renovado em 10 minutos.

RF10: Quando o usuário clicar no botão “Logar”, será necessário o sistema verificar o Status daquele usuário, se for “Ativo”, libera o acesso e o redireciona para a página principal. Caso o usuário esteja com o status “Inativo”, será alertado de que esta conta não pode realizar acesso pois está desativada.

RF11: Caso o login seja feito com o usuário do tipo “Secretaria”, será permitido que faça o controle de paciente e o agendamento da consulta. Para cadastrar um paciente, o usuário precisará ir no menu, que apresentará uma opção de “Cadastro”. Ao clicar na opção, será exibido um submenu com a opção de “Paciente”. Quando clicado em “Paciente”, haverá outro submenu com as opções “Criar um novo” e “Todos os pacientes”. Quando clicado em “Criar um novo”, será aberto um formulário contendo 9 inputs do tipo “text” para armazenar as informações do Paciente como, Nome, CPF, Data de nascimento, Endereço, Telefone, E-mail, Profissão, Sexo, Observação (opcional). Também contará com um input do tipo “submit”, ou seja, um botão com o nome de “Cadastrar” para realizar o cadastro no banco. Todos os campos são obrigatórios, portanto, se houver ao menos um campo sem informação nenhuma, ele deverá alertar o usuário de qual campo deve ser preenchido. O campo do CPF terá um validador e caso o CPF digitado não seja válido, o usuário será alertado a digitar um que seja de fato válido. Caso não haja problemas com a entrada, o sistema deverá salvar as informações do Paciente e armazenar no banco de dados.

RF12: Ao clicar em “Todos os Pacientes” no submenu de “Paciente”, será apresentado na página uma tabela com todos os pacientes registrados no sistema, podendo filtrar por Nome, CPF ou Data de Nascimento. Nesta tabela haverá todas as informações do Paciente. Embaixo do nome de cada usuário da tabela, será visível duas opções, “Editar” e “Desativar”. Ao clicar em “Editar” o Secretário será redirecionado para uma página no qual conterá as informações do respectivo Paciente selecionado, no qual possibilitará a edição de todos os campos Nome, CPF, Data de nascimento, Endereço, Telefone, E-mail, Profissão, Sexo, Observação (opcional). Todos os campos devem estar preenchidos para que possa ser atualizado os dados. Caso não haja problemas com a entrada, o sistema deverá atualizar as informações do Paciente no banco de dados.

Caso o Secretário selecione a opção “Desativar Paciente”, o sistema deverá apresentar uma mensagem de confirmação alertando se de fato o Secretário deseja desativar o Paciente, se sim, o Paciente terá o seu Status alterado de “Ativo” para “Inativo”, no qual não permitirá que ele tenha uma consulta agendada.

RF13: No menu da página, terá uma opção chamada “Consultas” no qual exibe, quando clicada, duas opções “Nova Consulta” e “Todas as Consultas”. Ao clicar em “Nova Consulta”, será exibido um formulário para o Secretário a fim de realizar o agendamento de uma consulta. A página contará com 6 inputs, sendo dois inputs do tipo “select” retornando uma lista de pacientes no select “Paciente” e uma lista de médicos no select “Médico”, para que o secretário selecione qual o Paciente e Médico serão consultados na respectiva consulta que está sendo agendada. Seguido terá um input do tipo “date” para que seja selecionado a data da consulta, não podendo selecionar uma data anterior a de hoje. Terá um input do tipo “time” para que seja definido o horário da consulta, um input do tipo “textarea” para que seja possível adicionar observações a respeito da consulta e o input do tipo “select” para selecionar se a consulta será feita de forma “Gratuita”, pelo “Nome do Plano de Saúde” ou “Particular”.

O secretário após informar todos os dados deverá clicar no input do tipo “submit”, ou seja, um botão que salva e realiza o agendamento da consulta. Todos os campos são obrigatórios, exceto o campo de Observação, que será opcional, portanto, se houver ao menos um campo obrigatório sem informação nenhuma, ele deverá alertar o usuário de qual campo deve ser preenchido. O sistema deve verificar se há alguma consulta agendada para este médico no mesmo dia e horário, se sim, enviar uma mensagem de alerta ao usuário informando que o médico já possui um paciente agendado neste horário, não permitindo que ele faça o agendamento. Caso não haja problemas com a entrada, o sistema deverá salvar as informações da consulta e armazenar no banco de dados. O sistema deve armazenar também, na tabela “TBConsultas”, o nome do secretário que está realizando o agendamento e a data que está sendo agendada a consulta.

RF14: Ao clicar em “Todos as Consultas” no submenu de “Consultas”, é será apresentado na página uma tabela com todos as consultas agendadas no sistema no dia de hoje, podendo filtrar por dia, paciente ou médico. Nesta tabela haverá todas as informações da Consulta. Embaixo do nome de cada consulta da tabela, será visível duas opções, “Cancelar”, “Re-agendar” e “Confirmar Presença”.

RF15: Na listagem de "Todas as Consultas" haverá a opção "Cancelar" com um evento de clique que abrirá um formulário com os detalhes da consulta na página, após esses detalhes terão 2 inputs do tipo “submit”, o primeiro com o nome "Cancelar" e uma estilização da cor vermelha que após clicar nele irá exibir um campo de texto pra definir o motivo de cancelamento da consulta, o segundo com o nome "Voltar" e uma a estilização da cor cinza, caso o usuário não queria prosseguir com o cancelamento.

RF16: Ao clicar em “Re-agendar consulta”, será exibido um formulário que deverá ser preenchido de acordo com os dados do solicitante. Neste formulário, virá como padrão: Nome do paciente, nome do médico, data da consulta e horário em que estava agendada, e descrição do motivo pela qual será reagendada. A página também contará com 5 inputs sendo dois inputs do tipo “select” que retornará uma lista de pacientes no select “Paciente” e uma lista de médicos no select “Médico”, para que seja possível selecionar o paciente que gostaria de reagendar a

consulta e o médico em questão. Também haverá um input do tipo “date” para que seja possível selecionar a data da consulta atual que deseja remarcar, e além disso, conterà um input do tipo “time” para selecionar o horário da consulta que já havia sido marcada. Após isso, basta clicar no Botão do tipo “submit” com o nome de Buscar, ele será responsável por realizar a busca da consulta que já havia sido agendada. Após retornar os dados da consulta, o usuário deverá clicar no botão Re-agendar novamente, e então será aberto um novo formulário da nova consulta, ou seja, a que será postergada para uma próxima data. Neste formulário, irá conter um input do tipo “textarea” possibilitando o usuário realizar a descrição e observações a respeito do motivo pelo qual a mesma será re agendada, um input do tipo “date” para selecionar a data que será realizada a nova consulta, e um outro input do tipo “time” para escolher o horário. Após serem realizadas todas essas alterações pelo usuário, basta clicar no botão Reagendar, imediatamente será exibido um “pop-up” perguntando se o usuário deseja salvar as informações e reagendar a consulta, haverá 2 botões com as opções: Salvar e Cancelar. Para que os novos dados sejam salvos no sistema, o usuário deverá clicar em Salvar e consulta será reagendada. Para os casos de desistência, basta clicar em Cancelar e o sistema retornará ao formulário anterior.

RF17: Em todos os formulários do sistema que estejam relacionados com as Consultas, nestes, haverá um botão do tipo “submit” escrito Confirmar presença localizado ao lado direito. Após clicar neste botão, será retornado um pop-up com a questão: “Deseja confirmar?” e com dois botões inclusos de SIM e NÃO. Caso o usuário selecione a opção SIM, para confirmar sua presença na consulta, o mesmo será redirecionado para uma página onde será possível realizar o check-in online de acordo com o lugar que o usuário estiver, neste caso, o mesmo deverá estar com a localização da clínica onde agendou sua consulta, para autenticar o seu check-in e confirmação de sua presença. Este check-in deverá ser realizado no ato da consulta pelo usuário ou até mesmo administrador. Caso o usuário selecione a opção “NÃO” o formulário será fechado.

RF18: Ao clicar em Agenda médica, será exibido um formulário contendo controles de calendário/data e horários, podendo ser preenchido de acordo com a preferência do médico. Neste formulário, serão solicitados data atual ou anterior, até mesmo posterior em que se deseja buscar consultas que já foram realizadas ou que estão marcadas no futuro. A página também exibirá uma lista contendo os dados de todos os pacientes que estão agendados naquele respectivo dia.

A página conterà um input do tipo “date” para que seja possível selecionar a data em que se deseja visualizar os pacientes, e além disso, conterà um input do tipo “time” para selecionar o horário da consulta caso a mesma já tenha sido marcada. Após isso, basta clicar no Botão do tipo “submit” com o nome de Pesquisar, ele será responsável por realizar a busca da consulta que já havia sido agendada ou que já foram realizadas anteriormente, trazendo consigo os dados dos pacientes e horários que foram agendadas.

RF19: Ao abrir a agenda, o médico poderá visualizar as consultas a serem realizadas pelo mesmo. Para cada consulta com o status agendado, será exibido em uma div “row” que representa uma linha, quatro divs contendo o horário e a data da consulta e os nomes do médico e do paciente, que serão extraídos do banco de dados. Além disso, na div “row” haverá um input do tipo

“button” com o valor “Ver Consulta” que quando clicado deverá abrir um formulário que possibilita a visualização dos dados da consulta, possuindo 8 caixas de texto, disponíveis apenas para leitura, são elas: Médico(que exibe o nome do médico), Data(que exibe a data da consulta), Hora(que exibe o horário da consulta), Paciente(que exibe o nome do paciente), Data de Nascimento(que exibe a data de nascimento do paciente), Telefone(que exibe o telefone do paciente), Profissão(que exibe a profissão do paciente) e Anotações(campo inicialmente vazio, que permitirá a adição de valores quando iniciada a consulta). Todos os dados dos campos, deverão ser obtidos no banco de dados. O formulário contará também com 3 botões, "Iniciar Consulta" e "Encerrar Consulta".

Quando clicado no botão “Privar Anotações”, será feita a verificação se os campos senha e confirmação de senha coincidem. Caso os campos estejam iguais será salvo no banco a senha já criptografada seguindo os padrões previamente definidos. Se os campos estiverem vazios e o botão for clicado, será notificado ao usuário que é necessário preencher os campos para que seja possível privar as anotações. Se os campos não coincidirem, será notificado ao usuário a necessidade de que os campos sejam preenchidos de forma igual.

Se o botão “Privar Anotações” não for clicado, ao finalizar a consulta o campo observações será automaticamente privado de acordo com o seguinte padrão: em um for que contará de 0 a 7, totalizando 8 repetições, serão gerados 8 números randômicos de 0 a 26, estes números serão o índice de uma lista de caracteres contendo o alfabeto que vai de 0 a 26. Em uma variável “senhaAnotacoes” será concatenado um valor da lista por repetição, totalizando ao final das repetições 8 caracteres. Será feito o hash duplo da variável “senhaAnotacoes” e será salvo no banco de dados.

Além disso, este botão altera o status da consulta de "Agendado" para "Consultando", além de atribuir o horário do início da consulta no campo "HorarioInicio" da tabela "TBConsulta", sendo atribuído pelo sistema no qual pega a informação de uma função e retorna o horário atual.

O Botão "Encerrar Consulta" salva os dados inseridos na caixa de texto "Observações" no banco de dados, além de mudar o Status para "Finalizado" e também atribui o horário do salvamento no campo "HorarioFim" da tabela "TBConsulta" através do sistema no qual pega a informação de uma função e retorna o horário atual.

RF20: Ao realizar os atendimentos o médico terá a opção de cifrar as observações do atendimento com uma senha, fazendo com que somente ele possa ter acesso aos detalhes do atendimento. Ao realizar o termino da consulta, o médico terá um input do tipo “button” no qual será responsável por salvar as informações no banco. Porém antes de salvar, ele dará a opção de inserir uma senha, caso ela fique em branco, as observações serão visíveis para todos os usuários do tipo médico. Caso ele decida utilizar, será apresenta um input do tipo “text” para que ele informa a senha, após informada, a as observações será criptografada e armazenada no banco. Para realizar o acesso aos detalhes das consultas, será necessário que o médico informe a senha novamente para que seja apresentado as informações, caso contrário, o médico ficará sem acesso pois não haverá quaisquer formas de recuperação ou redefinição de senha.

RF21: Na página de histórico do paciente, o médico digitará em um campo o nome do paciente no qual ele deseja ver as consultas anteriores e filtrar pela data caso queira. Após isso, o sistema trará uma tabela com as consultas realizadas por aquele paciente, sendo pela data caso informado, ou pela ordem das consultas realiza. Porém, ficará limitado a ver somente as informações básicas da consulta como horário de início e fim e com qual médico foi realizado a



consulta. Caso o médico deseja ver os detalhes do atendimento, será apresentado um input do tipo “text” para que ele informe a senha para o respectivo atendimento, se aceita, as informações de observação serão apresentadas ao usuário, caso contrário, ele não permitirá o acesso até que a senha esteja correta.

### **Requisitos Não-Funcionais**

RNF01: O sistema deve ser executado em dispositivos móveis ou computadores em navegadores compatíveis com ASP.NET Core;

RNF02: Para facilitar a utilização, os controles devem ser selecionados na seguinte ordem: de cima para baixo, da esquerda para a direita.

RNF03: Ao tentar efetuar o login, o usuário não será desativado caso haja muitas tentativas de acesso inválidas.

RNF04: Somente usuários maiores de 16 anos poderão realizar o cadastro no sistema.

RNF05: O sistema não possui sistema operacional específico para sua execução e acesso, nem mesmo memória, hardware ou quaisquer outros periféricos que impactam em seu desempenho, além da própria conexão com internet.

RNF06: O sistema seguirá todos os padrões e precedências estando em acordo com a nova LGPD (Lei Geral de Proteção de Dados) e os usuários deverão ler e concordar com os termos de autorização do uso de seus dados.