

**ClínicaCare: Promovendo Saúde e Bem-Estar com um Sistema Integrado de
Gestão**

Camilly Vitoria Quirino da Silva

Kauã Victor Seles Santos

Luis Henrique Magueta Santos

Rafael Martin

Vitória Sousa dos Santos Silva

Professoras:

Priscila Angela Berton e Magda Miyashiro

Sumário

Introdução.....	3
Visão Geral do Software.....	4
Requisitos Funcionais.....	5
Cadastro de Pacientes:.....	5
Agendamento de Consultas:.....	5
Registro de Consultas e Exames:.....	5
Gerenciamento de Médicos e Funcionários:.....	5
Portal do Paciente:.....	6
Requisitos Não Funcionais.....	6
Desempenho:.....	6
Usabilidade:.....	6
Confiabilidade:.....	6
Disponibilidade:.....	6
Manutenção:.....	6
Regra de Negócio.....	7
Cadastro de Pacientes:.....	7
Agendamento de Consultas:.....	7
Registro de Consultas e Exames:.....	7
Gerenciamento de Médicos e Funcionários:.....	7
Métodos e atributos:.....	7
Paciente.....	7
Descrição:.....	7
Atributos:.....	7
Métodos:.....	8
consulta.....	8
Descrição:.....	8
Atributos:.....	8
Métodos:.....	9
médico.....	9
Descrição:.....	9
Atributos:.....	9
Métodos:.....	9
Diagrama de Sequência.....	10
Diagrama de Classe.....	11
Diagrama de Atividades.....	11
Diagrama e Especificação de caso de uso.....	12
Diagrama Modelo Relacional.....	13

Introdução

O ClínicaCare é um projeto em desenvolvimento com o objetivo de proporcionar uma solução abrangente e eficiente para o gerenciamento de informações médicas e operacionais em ambientes de cuidados de saúde. Esta plataforma visa promover o alcance das metas estabelecidas pela ODS 3 - Saúde e Bem-Estar, contribuindo para a melhoria dos serviços de saúde e o bem-estar das comunidades.

Visão Geral do Software

Nosso objetivo com o ClínicaCare é tornar a vida das clínicas e hospitais mais fácil, ao mesmo tempo em que garantimos que todos recebam o melhor atendimento possível. Com o ClínicaCare, queremos que as equipes médicas tenham mais tempo para cuidar das pessoas, e menos tempo lidando com papelada.

Queremos que seja fácil para as clínicas e hospitais acompanhar cada paciente, desde o momento em que entram pela porta até o acompanhamento de seu progresso ao longo do tempo. E não queremos apenas ajudar no atendimento aos pacientes, mas também tornar os bastidores mais tranquilos, automatizando tarefas administrativas e simplificando processos.

Requisitos Funcionais

Cadastro de Pacientes:

Permitir o registro de informações detalhadas dos pacientes, incluindo:

- Nome completo, data de nascimento, gênero, endereço, contato e outras informações pessoais relevantes.
- Histórico médico, como doenças pré-existentes, cirurgias anteriores, alergias e medicamentos em uso.
- Registros de consultas anteriores, diagnósticos, tratamentos e exames realizados.
- Oferecer funcionalidades para editar ou atualizar informações dos pacientes conforme necessário.

Agendamento de Consultas:

- Possibilitar o agendamento de consultas médicas pelos funcionários da clínica ou hospital.
- Incluir uma interface para visualizar a disponibilidade dos médicos e agendar consultas conforme necessário.
- Enviar lembretes automáticos para consultas agendadas via e-mail ou SMS.

Registro de Consultas e Exames:

- Registrar consultas médicas e exames realizados, incluindo:
- Data, hora e duração da consulta.
- Manter um histórico completo de saúde para cada paciente, com acesso fácil para médicos e outros profissionais autorizados.
- Integrar resultados de exames diretamente no histórico do paciente para facilitar o monitoramento.

Gerenciamento de Médicos e Funcionários:

- Cadastrar médicos e funcionários com suas especialidades e horários de trabalho.

- Gerenciar as agendas dos médicos e outros profissionais de saúde.

Portal do Paciente:

- Permitir que os pacientes acessem suas informações médicas, histórico de consultas, resultados de exames e agendem consultas.

Requisitos Não Funcionais

Desempenho:

- O sistema deve responder às consultas de busca de pacientes e agendamento de consultas em menos de 2 segundos para garantir uma experiência ágil para os usuários;
- O tempo de carregamento das páginas do sistema não deve exceder 3 segundos para evitar frustração dos usuários.

Usabilidade:

- A interface do usuário deve ser intuitiva e de fácil navegação, com menus claros e organizados;

Confiabilidade:

- O sistema deve estar disponível 99.9% do tempo, minimizando interrupções no atendimento aos pacientes;

Disponibilidade:

- O ClínicaCare deve estar acessível 24 horas por dia, 7 dias por semana, para atender às necessidades dos profissionais de saúde e pacientes.

Manutenção:

- O sistema deve ser modular e de fácil manutenção, permitindo atualizações regulares sem impactar a disponibilidade;
- As atualizações de software devem ser realizadas fora do horário de pico para evitar interrupções no uso.

Regra de Negócio

Cadastro de Pacientes:

- Pacientes devem ser registrados com informações precisas e atualizadas.
- Somente funcionários autorizados podem acessar e editar os dados dos pacientes.

Agendamento de Consultas:

- As consultas devem ser agendadas de acordo com a disponibilidade dos médicos e as necessidades dos pacientes.
- Notificações automáticas devem ser enviadas aos pacientes para confirmar consultas agendadas.

Registro de Consultas e Exames:

- Resultados de exames devem ser integrados ao histórico do paciente.
- O acesso aos registros médicos deve ser restrito a médicos e profissionais de saúde autorizados.

Gerenciamento de Médicos e Funcionários:

- Médicos e funcionários devem ser cadastrados com informações detalhadas sobre suas especialidades e horários de trabalho.

Métodos e atributos:

Paciente

Objeto: Paciente

Descrição:

O objeto Paciente representa os indivíduos que recebem atendimento médico na clínica ou hospital.

Atributos:

nomeCompleto (String): Nome completo do paciente.

dataNascimento (LocalDate): Data de nascimento do paciente.

genero (String): Gênero do paciente.

endereço (String): Endereço do paciente.

contato (String): Informações de contato do paciente.

historicoMedico (ArrayList<String>): Histórico médico do paciente, incluindo doenças pré-existentes, cirurgias anteriores, alergias e medicamentos em uso.

Métodos:

adicionarPaciente(infoPaciente: HashMap<String, Object>): Adiciona um novo paciente ao sistema com as informações fornecidas.

editarInformacoesPaciente(idPaciente: int, novasInformacoes: HashMap<String, Object>): Permite editar as informações de um paciente específico.

agendarConsulta(idPaciente: int, dataConsulta: LocalDate): Agenda uma consulta para o paciente especificado.

visualizarHistoricoMedico(idPaciente: int): Retorna o histórico médico de um paciente específico.

enviarLembreteConsulta(idPaciente: int): Envia um lembrete ao paciente sobre uma consulta agendada para ele.

consulta

Objeto: Consulta

Descrição:

O objeto Consulta representa as interações entre médicos e pacientes, registrando informações sobre os encontros médicos.

Atributos:

dataHora (LocalDateTime): Data e hora da consulta.

duracao (int): Duração da consulta em minutos.

idMedico (int): Identificação do médico responsável pela consulta.

idPaciente (int): Identificação do paciente atendido na consulta.

Métodos:

registrarConsulta(infoConsulta: HashMap<String, Object>): Registra uma nova consulta no sistema com as informações fornecidas.

editarConsulta(idConsulta: int, novasInformacoes: HashMap<String, Object>): Permite editar as informações de uma consulta específica.

cancelarConsulta(idConsulta: int): Cancelar uma consulta agendada.

visualizarConsultas(idPaciente: int): Retorna a lista de consultas de um paciente específico.

visualizarConsultasMedico(idMedico: int): Retorna a lista de consultas agendadas para um médico específico.

médico

Objeto: Médico

Descrição:

O objeto Médico representa os profissionais de saúde que prestam serviços na clínica ou hospital.

Atributos:

nomeCompleto (String): Nome completo do médico.

especialidade (String): Especialidade médica do médico.

horarioTrabalho (ArrayList<HashMap<String, String>>): Lista que registra o horário de trabalho do médico, incluindo dia da semana e horário de início/fim.

Métodos:

adicionarMedico(infoMedico: HashMap<String, Object>): Permite adicionar um novo médico ao sistema com as informações fornecidas.

`editarInformacoesMedico(idMedico: int, novasInformacoes: HashMap<String, Object>)`: Permite editar as informações de um médico específico.

`visualizarAgendaMedico(idMedico: int)`: Retorna a agenda de um médico específico.

`visualizarPacientesAtendidos(idMedico: int)`: Retorna a lista de pacientes atendidos por um médico específico.

Diagramas:

Diagrama de Sequência

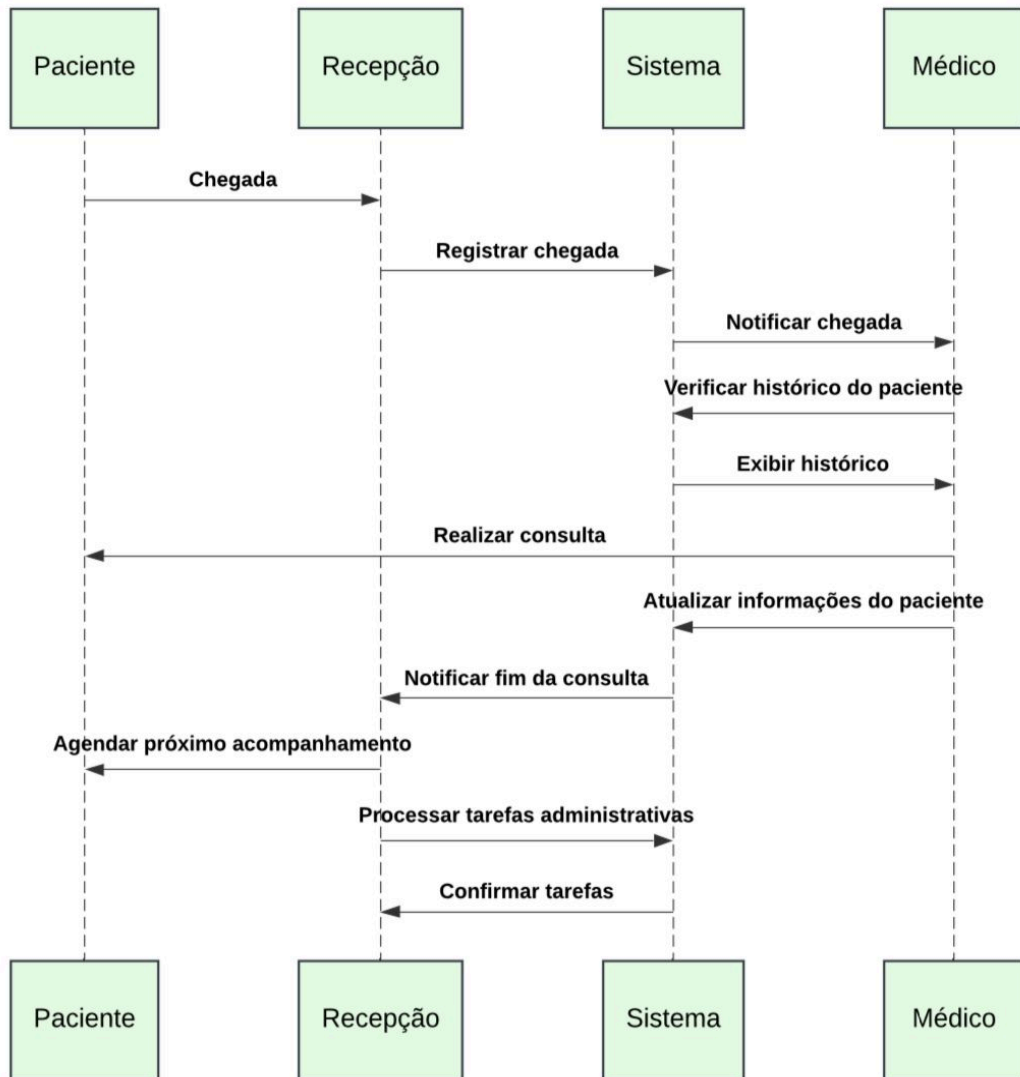


Diagrama de Classe

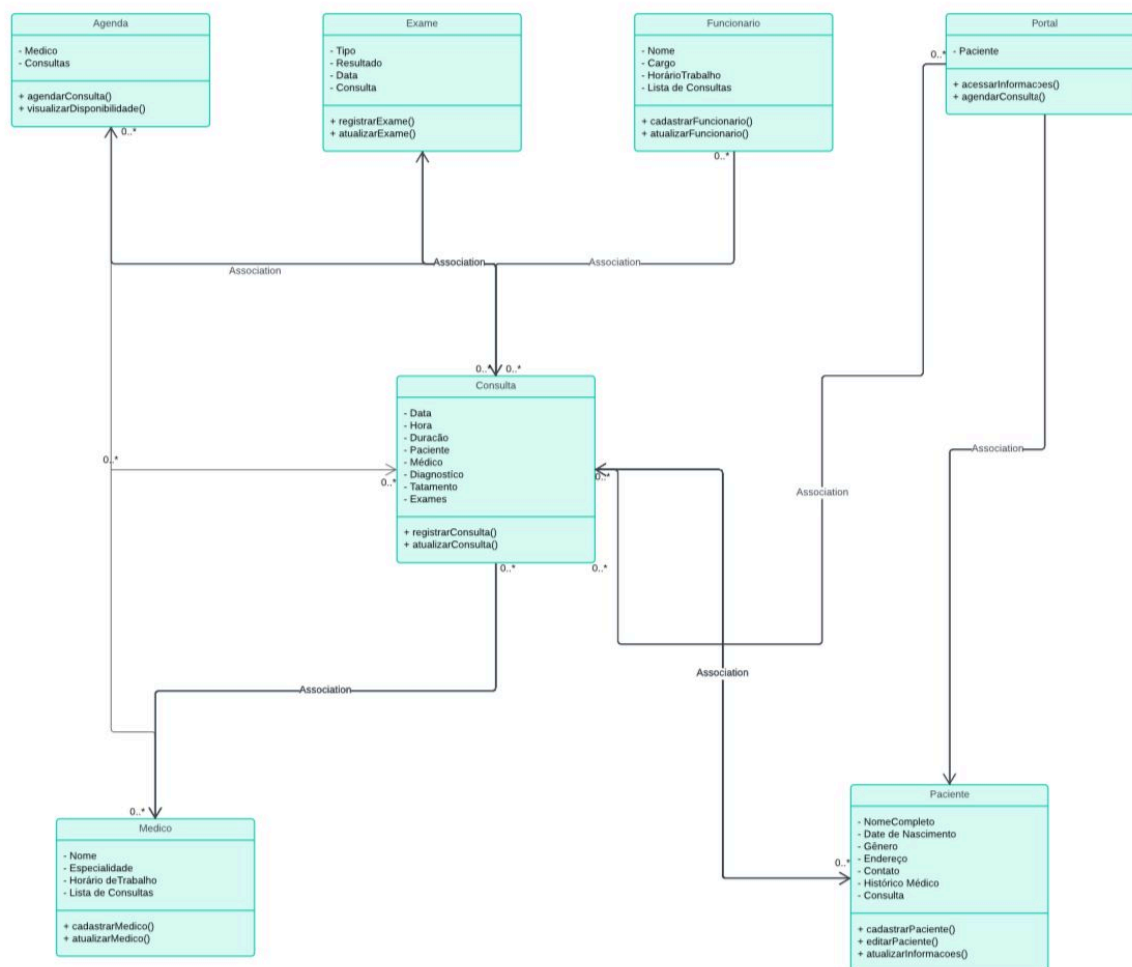


Diagrama de Atividades

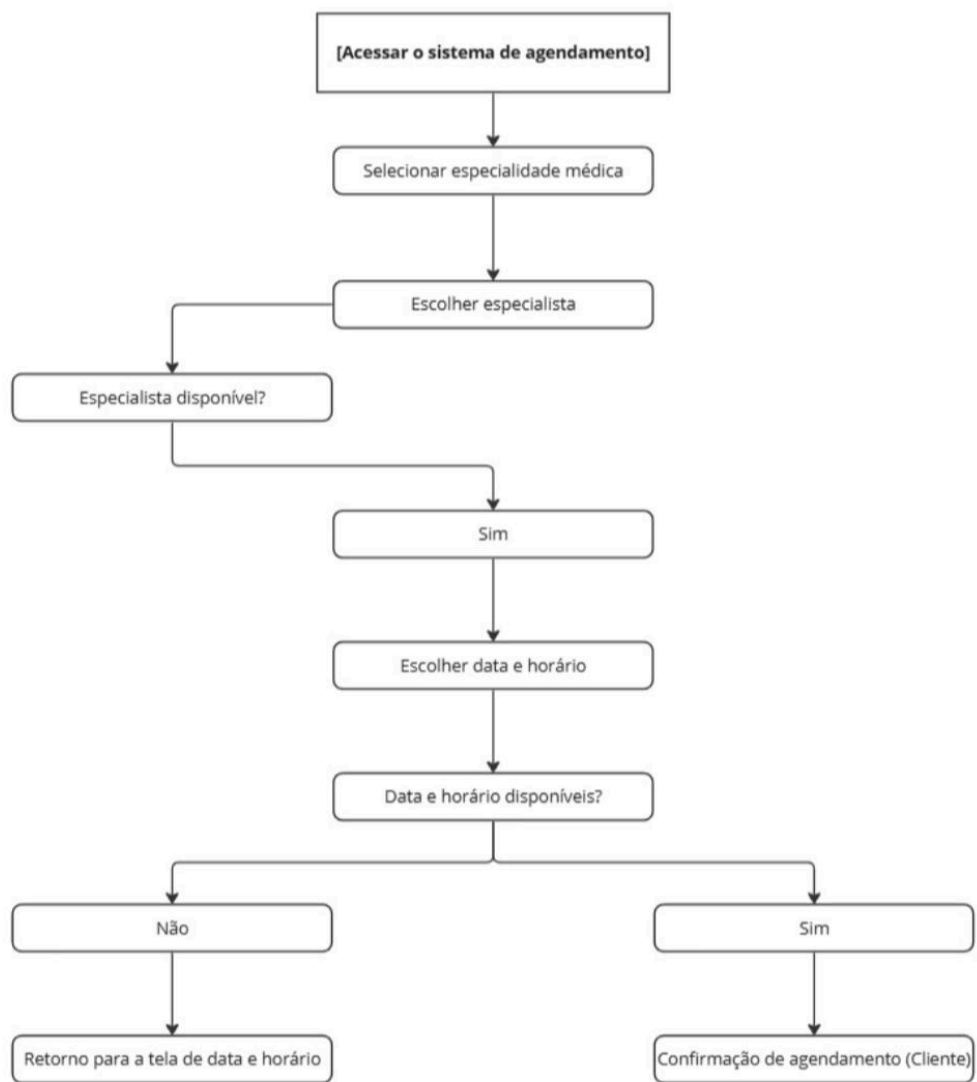


Diagrama e Especificação de caso de uso

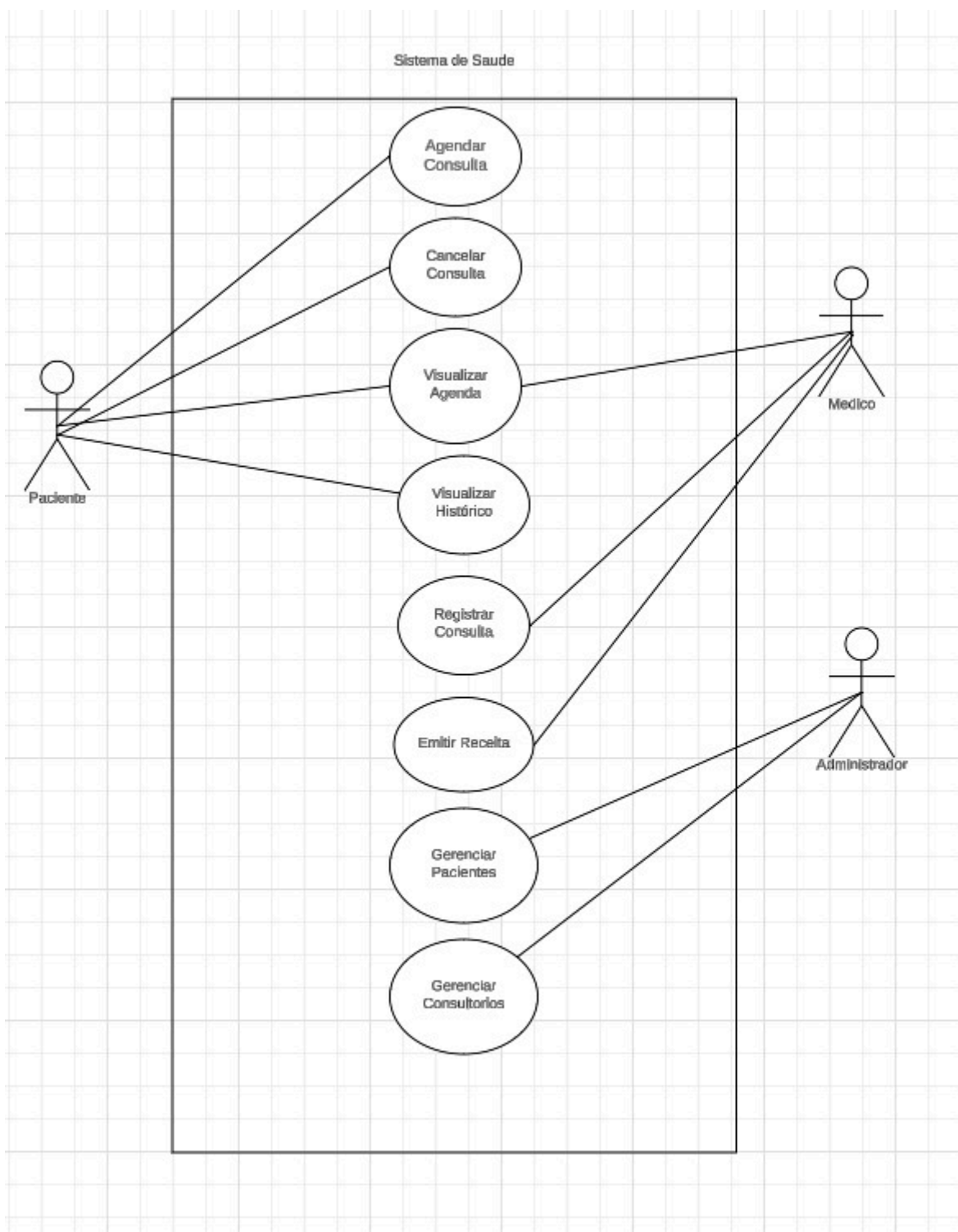


Diagrama Modelo Relacional

