



Serviço Nacional de Aprendizagem Industrial
Curso Técnico - Desenvolvimento de Sistemas



Sistema de aprendizado ONCOMED

Maria Eduarda Klin de Carvalho
Maria Victória da Cruz Ferreira
Nicholas Serencovich
Rinmarys Alejandra Monagas Marcano

**FLORIANÓPOLIS
2024**

SUMÁRIO

1.	Introdução.....	3
1.1.	Problemas Identificados.....	3
1.2.	Público alvo.....	4
1.3.	Solução.....	4
2.	Stakeholder.....	5
3.	Requisitos de Sistema.....	6
3.1.	Requisitos Funcionais.....	6
3.2.	Requisitos não funcionais.....	8
4.	Protótipos/Wireframes.....	9
4.1.	Link para Figma.....	9
4.2.	Link para Wireframes.....	9
5.	Diagramas UML.....	9
6.	Tecnologias Utilizadas.....	10
6.1.	Para organização e prototipagem.....	10
6.2.	Para a programação das telas.....	11
7.	Banco de Dados.....	13
7.1.	Modelo conceitual.....	13
7.2	Modelo lógico.....	14
7.3	Modelo físico.....	15

1. INTRODUÇÃO

O conceito de agendamento eficiente e acesso a informações de alta qualidade são críticos para os pacientes com câncer no mundo de hoje. É necessário encontrar uma solução para esse problema e facilitar o processo de marcar consultas e fornecer educação constante relacionada ao câncer. Portanto, o nosso projeto visa criar um aplicativo móvel que abordará essa área, oferecendo um serviço para agendar consultas, prestar assistência e garantir o acesso a informações.

1.1. PROBLEMAS IDENTIFICADOS

↳ Problemas enfrentados pelo paciente:

Dificuldades de agendamento: Pacientes enfrentam dificuldades para marcar consultas devido à falta de sistemas eficientes e centralizados.

Informação Fragmentada: Falta de uma fonte centralizada e confiável para informações sobre câncer.

Acesso Limitado: Necessidade de acesso fácil e rápido às informações e serviços sem a necessidade de deslocamentos frequentes.

↳ Problemas enfrentados pelas clínicas/hospitais:

Gestão Ineficiente de Pacientes: Clínicas e hospitais enfrentam dificuldades na administração de agendas, na coordenação dos horários de atendimento e na gestão de pacientes devido à falta de um sistema centralizado. Fazendo com que os períodos de espera para o atendimento sejam demorados, e com isso também, fazendo com que a enfermidade avance.

Comunicação Ineficiente: A comunicação entre clínicas/hospitais e pacientes pode ser deficiente, especialmente quando se trata de *notificações de consultas, lembretes e atualizações sobre o tratamento.*

1.2. PÚBLICO ALVO

Pacientes Oncológicos: Pessoas em tratamento ou em acompanhamento para câncer.

Familiares e Cuidadores: Pessoas que acompanham e ajudam os pacientes em seus tratamentos.

Profissionais de Saúde: Médicos e enfermeiros que atuam em clínicas oncológicas.

Administradores de Clínicas: Pessoas responsáveis pela gestão e operação das clínicas.

1.3. SOLUÇÃO

↳ Desenvolver um aplicativo móvel que:

- Permita o agendamento de consultas em clínicas oncológicas de forma simples e eficiente.
- Ofereça um blog integrado com artigos sobre tipos de câncer, tratamentos e medicina alternativa.
- Forneça lembretes e notificações para consultas e tratamentos.
- Permita a visualização do histórico de consultas e exames.

2. IDENTIFICAÇÃO DE STAKEHOLDERS

↳ Patrocinadores:

Empresas: Que queiram a sua divulgação no Site.

↳ Diretores da Empresa:

Administradores de Clínicas: Pessoas responsáveis pela gestão e operação das clínicas.

↳ Equipe de Projeto:

Maria Eduarda Klin de Carvalho: UI e UX designer, Front-End, Back-end.

Maria Victória da Cruz Ferreira: UI e UX designer, Front-End, Back-end.

Rinmarys Alejandra Monagas Marcano: UI e UX designer, Front-End, Back-end.

Nicholas Serencovich Carvalho: UI e UX designer, Front-End, Back-end.

Ana Carolina Andrade de Araújo: UI e UX designer, Front-End, Back-end.

↳ Gerentes de Projeto:

Rinmarys Alejandra Monagas Marcano: Desenvolvedor de Sistemas.

Maria Eduarda Klin de Carvalho: Desenvolvedor de Sistemas.

↳ Clientes:

Pacientes Oncológicos: Pessoas em tratamento ou em acompanhamento para câncer.

Familiares e Cuidadores: Pessoas que acompanham e ajudam os pacientes em seus tratamentos.

3. REQUISITOS DE SISTEMA

↳ 3.1. Requisitos funcionais (RF)

[RF001] – O paciente deve poder agendar consultas.

- O paciente deve inserir informações como data, hora e tipo de consulta que deseja agendar.

[RF002] – O paciente/administrador deve poder cancelar consultas.

- O paciente/administrador deve confirmar a exclusão antes de remover uma consulta.

[RF003] – O sistema deve permitir que pacientes visualizem suas consultas agendadas.

- Os pacientes devem ter acesso a uma lista de suas consultas futuras e passadas.

[RF004] – O sistema deve permitir que os médicos visualizem a agenda de consultas.

- Os médicos devem poder ver todas as consultas agendadas para o dia ou semana.

[RF005] – O sistema deve permitir que os pacientes e médicos atualizem suas informações pessoais.

- Os pacientes e médicos devem poder atualizar dados como telefone, endereço e e-mail.

[RF006] – O sistema deve permitir a categorização dos artigos do blog.

- Os artigos devem ser organizados em categorias (ex: Tratamentos, Dicas de Saúde, etc.) para facilitar a navegação.

[RF007] – O sistema deve validar as informações de login.

- O sistema deve permitir que os usuários façam login utilizando seu e-mail e senha. O login só deve ser aceito se as informações estiverem corretas de acordo com a conta correspondente. Caso as informações estejam incorretas, deverá aparecer uma mensagem de erro.

↳ **3.2. Requisitos não funcionais (RNF)**

[RNF001] – Aceitar os termos de uso e a política de privacidade.

- O usuário deve aceitar os termos de uso e a política de privacidade ao se cadastrar no sistema.

[RNF002] – O sistema não aceita o mesmo e-mail mais de uma vez.



- O sistema deve notificar o usuário com uma mensagem de erro ao tentar cadastrar um e-mail já registrado.

[RNF003] – O usuário só pode marcar uma consulta se a data e o horário estiverem disponíveis.

- O sistema deve verificar a disponibilidade em tempo real antes de confirmar o agendamento.

[RNF004] – Pessoas só podem se cadastrar se forem maiores de idade.

- O sistema deve solicitar a data de nascimento e validar se o usuário é maior de idade.

[RNF005] – O sistema só vai aceitar o cadastro se todas as informações forem preenchidas.

- O sistema deve exigir que os campos nome, e-mail e senha sejam preenchidos para permitir o cadastro.

[RNF006] – O sistema deve ter uma interface intuitiva.

- A navegação deve ser fácil e os usuários devem conseguir realizar ações sem dificuldade.

4. PROTÓTIPOS / WIREFRAMES

↳ 4.1. Link para Telas Figma:

<https://www.figma.com/design/VizoKBMnKxm9Z9LTnxk8gb/OncoMed-S.A?node-id=0-1&t=3k7196boEcNdW34N-1>

↳ 4.2. Link para Telas de Wireframes:

<https://excalidraw.com/#json=blf0ZZN0bmOgWOnH2P5Oy,32nCOhfFjSqKz5m0k9sqXQ>

5. DIAGRAMAS UML

https://miro.com/welcomeonboard/bE9JM3hGM3FqUVBqWHZ5NThLUkYzdVJHaEwvU2lsc25LKzMzSjV4RFk0dDFQdWV6WklhZ0lSMi9mT09ncFVnNy9nV25mNVZJbVRKOUUp0NEE4azg0LzV6VUhlc0QyVUJqVmcyYkZVcVhGbktOMTE4TGpzaklJdWhWYXgza2pWVnchZQ==?share_link_id=599870206894

6. TECNOLOGIAS UTILIZADAS

➤ 6.1. Para a organização e prototipagem:

Trello: Para facilitar a colaboração, pois todos conseguem ver o andamento das tarefas e deixar comentários nos cartões. Assim, todo mundo fica alinhado e pode acompanhar o progresso em tempo real do projeto.



Figma: Para criar e colaborar em designs de projetos, como protótipos. A ferramenta permite edição em tempo real e comentários diretos nas telas, facilitando a comunicação e o alinhamento com a equipe.



Excalidraw: Uso para criar esboços rápidos e wireframes, visualizando ideias de forma intuitiva. É colaborativo, permitindo trabalhar em tempo real com a equipe, ideal para brainstorms e planejamentos visuais.



Whatsapp: Uso para comunicação rápida e constante com a equipe, tirando dúvidas, compartilhando atualizações e enviando arquivos. Criamos grupos específicos para cada parte do projeto, facilitando a organização e o alinhamento.



GitHub: uso para controlar versões e colaborar no código, gerenciando branches e integrando mudanças através de pull requests, garantindo que todos no projeto fiquem alinhados.



Canva: uso para criar apresentações. A ferramenta permite que todos trabalhem juntos em tempo real, fazendo ajustes e deixando comentários diretamente nos designs, o que facilita a colaboração e o alinhamento das ideias.



➤ 6.2. Para programação das telas:

Visual Studio Code: No Visual Studio Code, uso o Live Share para editar o código em tempo real com a equipe, além de integrar Git para controlar versões e colaborar de forma eficiente no desenvolvimento do projeto.



HTML: usamos para estruturar as páginas do projeto, com cada membro responsável por seções específicas, garantindo organização e integração com outras tecnologias.



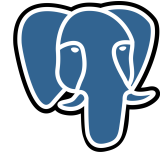
CSS: Usamos para estilizar as páginas, ajustando layout, cores e responsividade, com cada membro cuidando de diferentes partes do design.



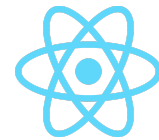
JavaScript: Usamos para adicionar interatividade e funcionalidades dinâmicas, com a equipe implementando recursos como formulários e animações.



PostgreSQL: uso para gerenciar o banco de dados do projeto, organizando e acessando dados de forma eficiente, enquanto a equipe colabora na criação de tabelas e consultas.



React: Facilita a criação de interfaces rápidas e eficientes com componentes reutilizáveis, e sua atualização de DOM virtual melhora a performance. A grande comunidade oferece várias ferramentas úteis.



Node JS: É rápido, escalável e permite programar em JavaScript no front-end e back-end. Ele lida bem com múltiplas requisições e tem uma ótima performance graças ao motor V8.

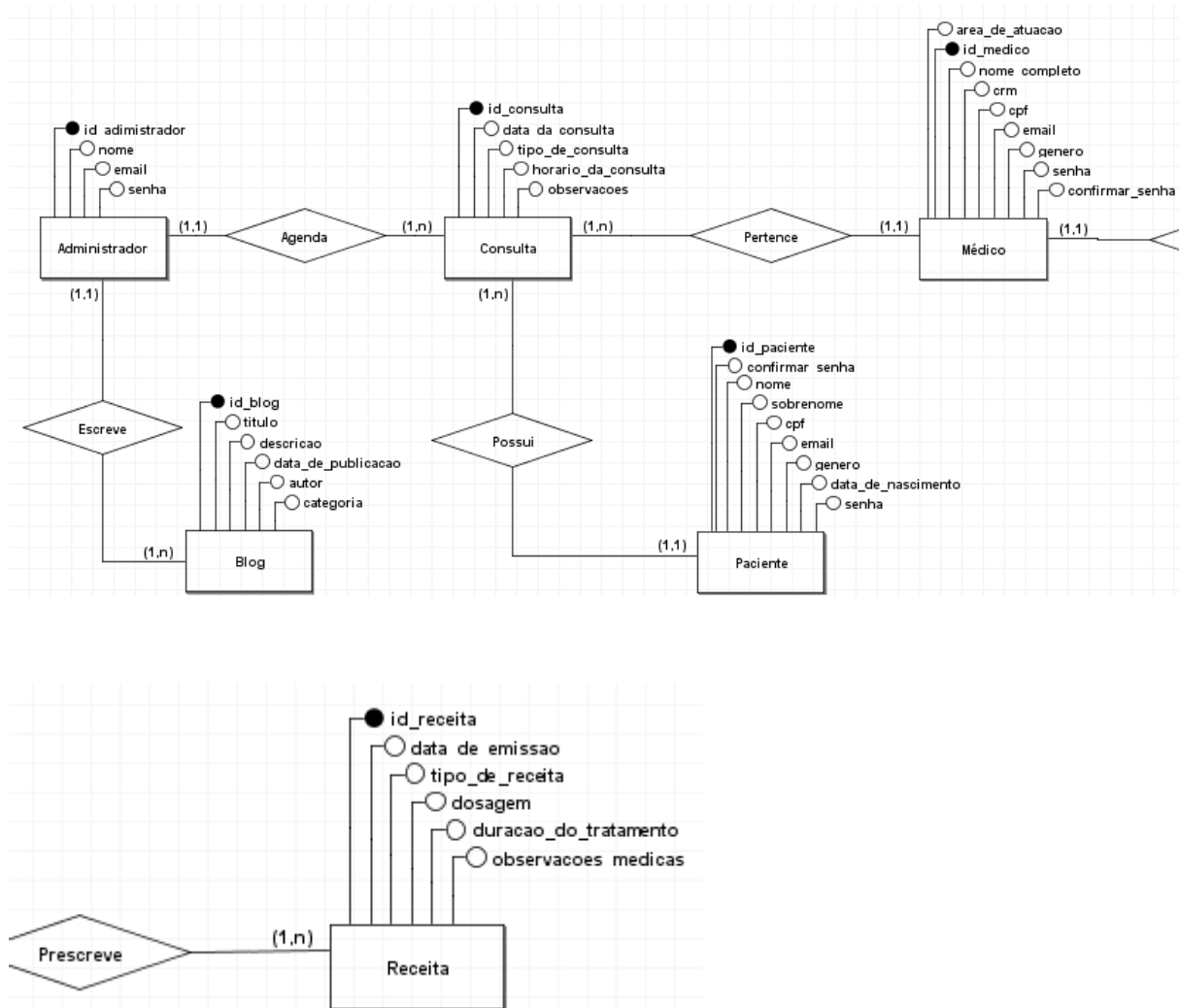


Postman: Usamos o Postman pela sua facilidade no teste, desenvolvimento e documentação de APIs, além de permitir colaboração eficiente e integração com outras ferramentas, otimizando o fluxo de trabalho.

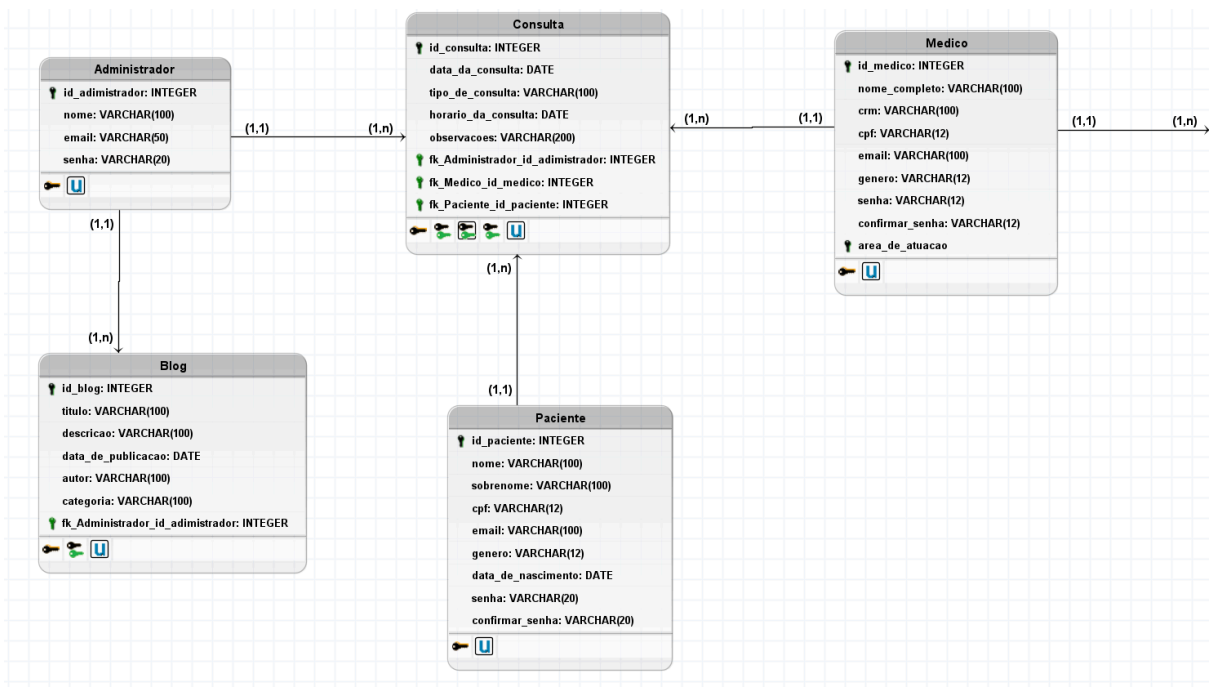
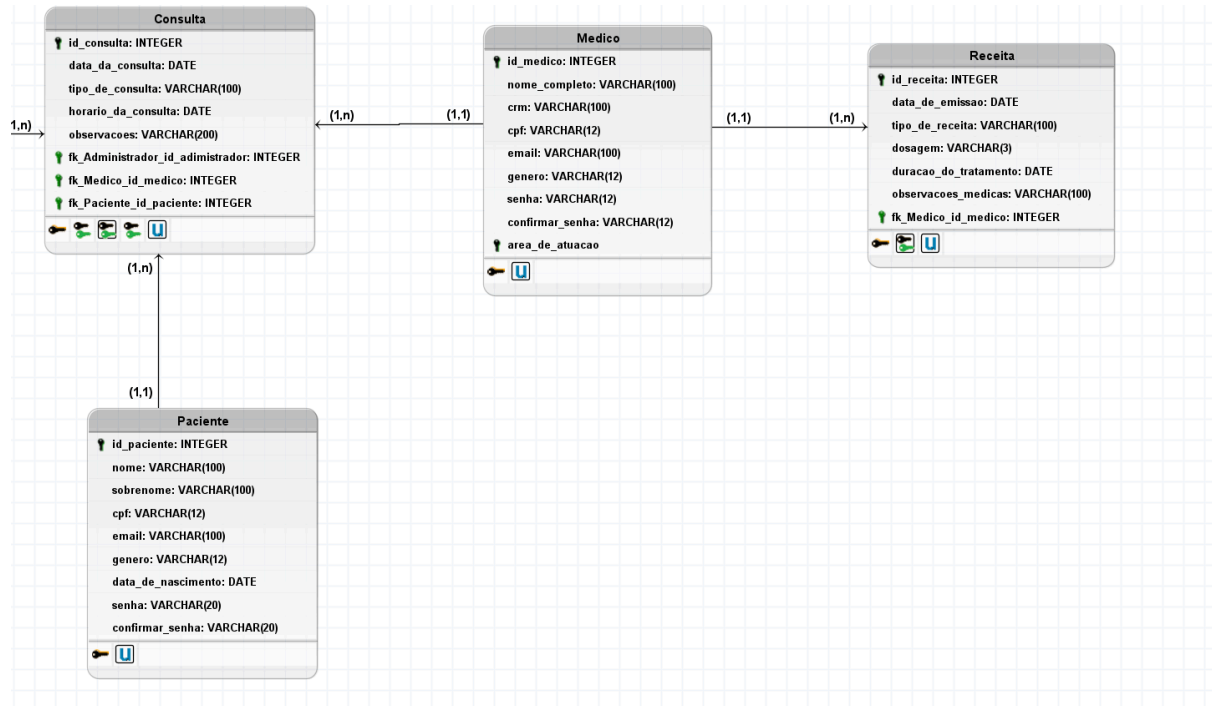


7. BANCO DE DADOS

7.1 Modelo Conceitual:



➤ 7.2 Modelo Lógico:



➤ 7.3 Modelo Físico:

```
12
13 CREATE TABLE pacientes(
14
15     id_paciente SERIAL PRIMARY KEY,
16     nome VARCHAR(100) NOT NULL,
17     cpf CHAR(14) UNIQUE NOT NULL,
18     cep CHAR(9) NOT NULL,
19     email VARCHAR(100) UNIQUE NOT NULL,
20     data_de_nascimento DATE NOT NULL,
21     genero VARCHAR(20) NOT NULL,
22     senha VARCHAR(14) NOT NULL
23 );
24
25 CREATE TABLE marcarConsulta(
26     id_marcar_consulta SERIAL PRIMARY KEY,
27     data_agendamento DATE NOT NULL,
28     tipo_consulta VARCHAR(200) NOT NULL,
29     horario TIME NOT NULL,
30     observacoes VARCHAR(100),
31     id_paciente INT,
32     FOREIGN KEY (id_paciente) REFERENCES paciente(id_paciente)
33 );
34
35 CREATE TABLE admin(
36     id_admin SERIAL PRIMARY KEY,
37     nome VARCHAR(100) NOT NULL,
38     email VARCHAR(50) NOT NULL,
39     senha VARCHAR(20) NOT NULL
40 );
41
42 CREATE TABLE blog (
43     id_blog SERIAL PRIMARY KEY,
44     titulo VARCHAR(100) NOT NULL,
45     descricao VARCHAR(300) NOT NULL,
46     data_publicacao DATE NOT NULL,
47     autor VARCHAR(100) NOT NULL,
48     categoria VARCHAR (100) NOT NULL,
49     id_admin INT,
50     FOREIGN KEY (id_admin) REFERENCES admin(id_admin)
51 );
```