

UNIVERSIDADE TIRADENTES

Análise e Desenvolvimento de Sistemas

5º Período

DESENVOLVIMENTO MOBILE

Professor: Diógenes Carvalho Matias

ARTUR RAMOS SILVA - 1231309706
CARLOS EDUARDO BARBOSA DE LIMA - 1231309625
LUIZ AUGUSTO DE LIMA BISNETO - 1231309870
PAULO ARTHUR - 1242303569
VINICIUS JOSÉ DA SILVA - 1232303990

Recife
2025

Sumário

Sumário	2
1. Introdução	3
2. Justificativa	3
3. Documento de Requisitos	4
3.1 Requisitos Funcionais (RF)	4
3.2 Requisitos Não Funcionais (RNF)	4
4. Matriz de Rastreabilidade	5
5. Documento de Casos de Uso	6
5.1 Diagrama de caso de uso	6
5.2 Diagrama de Máquina de estado	10
5.3 Diagrama Entidade-Relacionamento (ER)	13
5.4 Diagrama Lógico	13
5.5 Diagrama Físico	15
6. Documentação de Testes	17
6.1 Plano de Testes	17
6.2 Critérios de Aceitação	18
7. Documentação de Segurança	18
7.1 Autenticação e Autorização	18
7.2 Criptografia	19
7.3 Armazenamento Seguro	19
7.4 Proteção contra Ameaças	19
7.5 Conformidade com LGPD	19

1. Introdução

O Sistema de Gestão de Consultas da UNIT precisa se modernizar e otimizar o processo de agendamento, acompanhamento e gerenciamento de consultas médicas na clínica da instituição. Atualmente, o fluxo de atendimento é realizado de forma manual, por meio de planilhas e comunicação via WhatsApp, o que gera lentidão, desorganização e sobrecarga para os responsáveis. Este documento de requisitos tem como objetivo especificar as funcionalidades, regras e arquitetura do sistema, garantindo que ele atenda às necessidades dos usuários (pacientes, profissionais de saúde e administradores) com eficiência, segurança e usabilidade.

2. Justificativa

App Mobile para Gestão de Consultas - UnitSaude

O Sistema de Gestão de Consultas UNIT foi desenvolvido para substituir o atual processo manual - baseado em planilhas e comunicação via WhatsApp - por uma solução digital integrada. A plataforma centralizará todo o fluxo de agendamentos, desde a divulgação de vagas até o acompanhamento de consultas, resolvendo os principais problemas relatados:

- Sobrecarga administrativa devido ao preenchimento manual de planilhas;
- Falta de transparência na disponibilidade de vagas;
- Dificuldade de comunicação com pacientes;
- Organização ineficiente de prontuários e históricos.

3. Documento de Requisitos

3.1 Requisitos Funcionais (RF)

Código	Descrição
RF01	O sistema deve permitir autenticação ao paciente cadastrado.
RF02	O sistema deve permitir a visualização e seleção das consultas do paciente autenticado.
RF03	O sistema deve permitir o cadastro de consultas com área, especialidade, professor, anamnese básica, data e hora.
RF04	O sistema deve permitir o cancelamento de consultas com status pendente e em espera.
RF05	O sistema deve permitir o paciente autenticado deslogar.
RF06	O sistema deve permitir a recuperação de senha do paciente cadastrado.

3.2 Requisitos Não Funcionais (RNF)

Código	Descrição
RNF01	O aplicativo deve estar disponível para Android.
RNF02	O tempo de resposta para qualquer ação do usuário não deve ultrapassar 1 minuto.
RNF03	Os dados do usuário devem ser armazenados de forma segura.
RNF04	A interface deve ser acessível e responsiva.

4. Matriz de Rastreabilidade

Requisito	Caso de Uso	Entidade	Teste
RF01	UC-01/UC-API01	Logar paciente	TC01
RF02	UC02/UC-API02	Visualizar/selecionar consultas	TC02
RF03	UC03/UC-API03	Agendar consulta	TC03
RF04	UC04/UC-API04	Cancelar consulta	TC04
RF05	UC05	Deslogar paciente	TC05
RF06	UC06/UC-API05	Recuperar senha	TC06

5. Documento de Casos de Uso

5.1 Diagrama de caso de uso

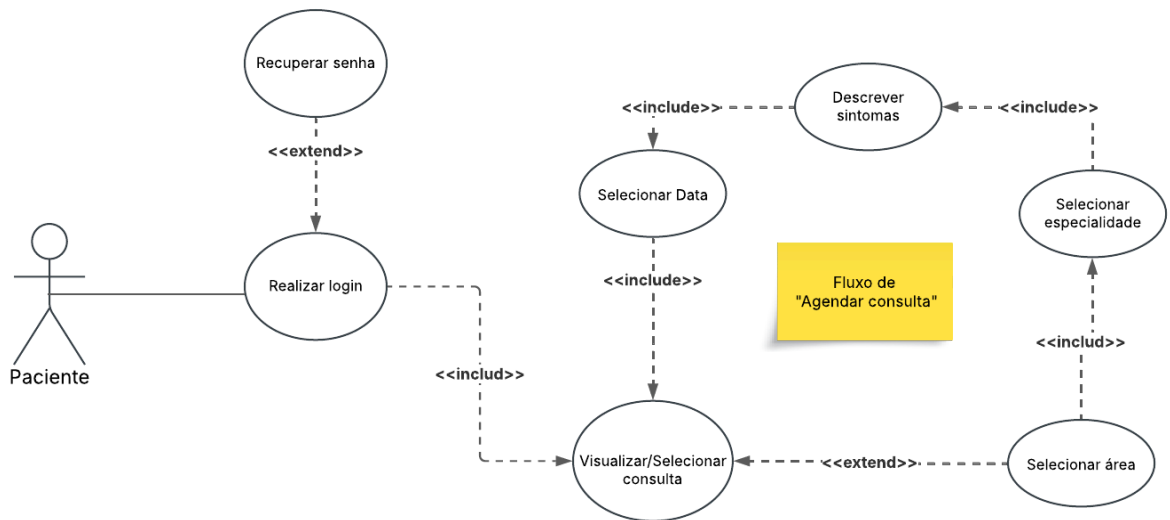


Diagrama de casos de uso - Paciente

UC01 - Realizar login

- **Ator Principal:** Paciente
- **Pré-condição:** O paciente deve estar cadastrado.
- **Fluxo Principal:**
 1. O paciente acessa a tela de login.
 2. Preenche email e senha.
 3. Confirma o login.
 4. O sistema valida e permite o acesso.
 5. O paciente direcionado para a tela inicial
 6. Exibe mensagem "Login realizado com sucesso!"
- **Fluxo Alternativo:** Dados inválidos ou Erro inesperado → exibe mensagem de erro.

UC02 - Visualizar/Selecionar consulta

- **Ator Principal:** Paciente
- **Pré-condição:** O paciente deve estar autenticado.
- **Fluxo Principal:**

1. O paciente acessa a tela inicial.
 2. Clica no botão para escolher o status das consultas.
 3. Visualize as consultas com o status selecionado.
- **Fluxo Alternativo:** Dados inválidos → exibe mensagem de erro.

UC03 - Agendar consulta

- **Ator Principal:** Paciente
- **Pré-condição:** O paciente deve estar autenticado.
- **Fluxo Principal:**
 1. O paciente acessa a tela inicial.
 2. Clica no botão “Agendar consulta”.
 3. Escolha a área da consulta.
 4. Escolha a especialidade da consulta.
 5. Escolha o professor da consulta.
 6. Preenche a anamnese básica.
 7. Escolha a data da consulta.
 8. Escolha a hora da consulta.
 9. Confirma se deseja marcar a consulta.
 10. Visualiza a confirmação da consulta.
 11. Direcionamento para a tela inicial.
- **Fluxo Alternativo:** Dados inválidos → exibe mensagem de erro.

UC04 - Cancelar consulta

- **Ator Principal:** Paciente
- **Pré-condição:** O paciente deve estar autenticado.
- **Fluxo Principal:**
 1. O paciente acessa a tela inicial.
 2. Seleciona a visualização por status pendentes ou em espera.
 3. Clica no botão “Cancelar” da consulta desejada.
 4. Confirma o cancelamento da consulta.
 5. Visualiza a confirmação do cancelamento
- **Fluxo Alternativo:** Dados inválidos → exibe mensagem de erro.

UC05 - Deslogar paciente

- **Ator Principal:** Paciente
- **Pré-condição:** O paciente deve estar autenticado.
- **Fluxo Principal:**
 1. O paciente acessa a tela inicial.
 2. Clica no ícone de porta no header.

3. Direcionamento para a tela de login.

- **Fluxo Alternativo:** Dados inválidos → exibe mensagem de erro.

UC06 - Recuperar senha

- **Ator Principal:** Paciente

- **Pré-condição:** O paciente deve estar cadastrado com email válido.

- **Fluxo Principal:**

1. O paciente acessa a tela de login.
2. Clique em “esqueceu sua senha?”.
3. Direcionamento para a tela de recuperação de senha.
4. Preenche o email e clica em enviar.
5. Recebe um link para recuperação de senha no email.
6. Direcionamento para a tela de “Nova senha”.
7. Preenche a nova senha e a confirmação da nova senha.
8. Direcionamento para a tela de login.

- **Fluxo Alternativo:** Dados inválidos → exibe mensagem de erro.

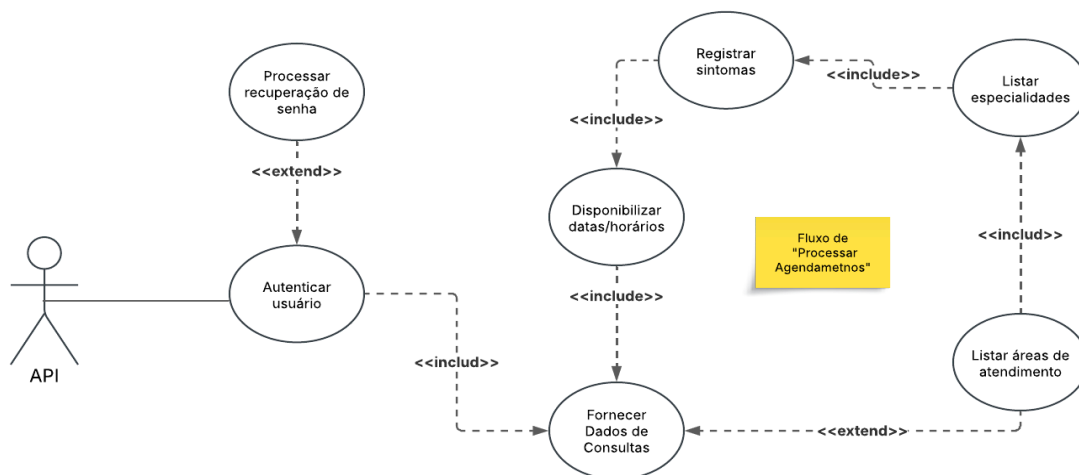


Diagrama de casos de uso - API

UC- API01 - Autenticar usuário

- **Ator Principal:** API

- **Pré-condição:** Credenciais válidas devem ser fornecidas e Usuário cadastrado

- **Fluxo Principal:**

1. Recebe requisição com email e senha.
2. Valida estrutura dos dados.
3. Busca usuário no banco de dados.

4. Valida correspondência de senha (hash).
 5. Gera token JWT válido por 4 horas.
 6. Retorna token de autenticação.
- **Fluxo Alternativo:** Credenciais inválidos → retorna 401.

UC- API02 - Fornecer Dados de Consultas

- **Ator Principal:** API
- **Pré-condição:** Token JWT válido
- **Fluxo Principal:**
 1. Valida token
 2. Decodifica token para obter patientId
 2. Verifica permissões
 3. Busca consultas no banco filtrando pelo patientId (Através do GET)
 4. Retorna lista formatada de acordo com o status escolhido.
- **Fluxo Alternativo:** Erro inesperado → retorna mensagem de erro.

UC- API03 - Processar Agendamentos

- **Ator Principal:** API
- **Pré-condição:** Token JWT válido
- **Fluxo Principal:**
 1. Valida token
 2. Faz o carregamento dos dados prévios via GET (áreas, especialidades, professores, horários).
 3. Verifica conflitos de horário
 4. Valida dados da consulta nova consulta
 5. Persiste no banco com status “Pendente” ou “Em espera” de acordo com horário escolhido.
 6. Retorna consulta criada (status 200)
 7. Envia email com as informações sobre a consulta para o Paciente.
- **Fluxo Alternativo:** Erro inesperado → retorna mensagem de erro.

UC- API04 - Processar Cancelamentos

- **Ator Principal:** API
- **Pré-condição:** Token JWT válido, Consulta existente
- **Fluxo Principal:**
 1. Recebe ID da consulta via POST para cancelar.

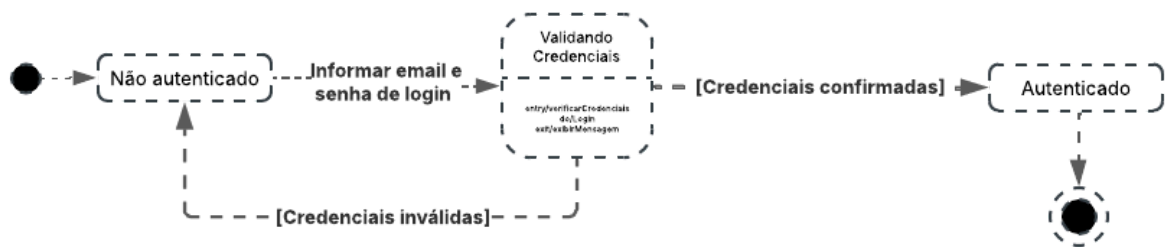
2. Verifica se consulta pertence ao paciente.
 2. Valida se cancelamento é permitido.
 3. Atualiza status para "Cancelada".
 4. Dispara email para o Paciente.
- **Fluxo Alternativo:** Erro inesperado → retorna mensagem de erro.

UC- API05 - Processar recuperação de senha

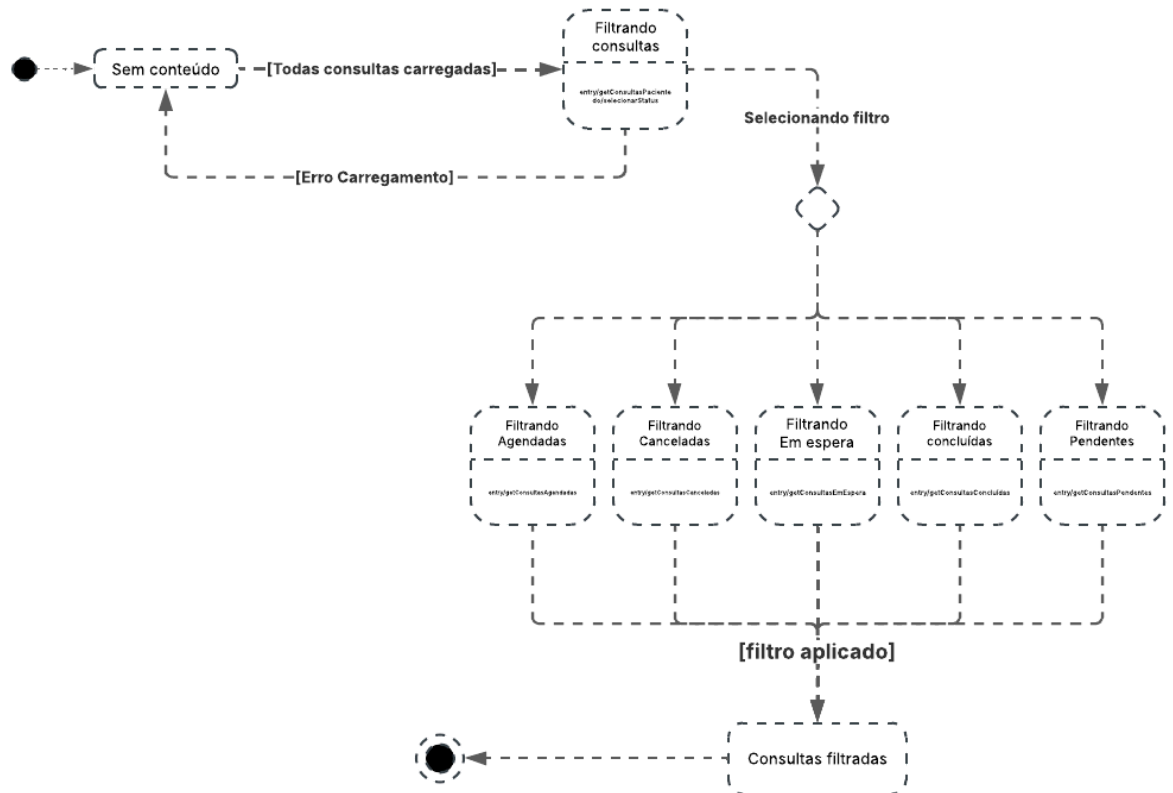
- **Ator Principal:** API
- **Pré-condição:** Email válido cadastrado
- **Fluxo Principal:**
 1. Recebe email via POST
 2. Gera token temporário (válido por 10 minutos)
 2. Envia email com link contendo token.
 3. Receber nova senha
 - 4 Valida token temporário
 - 5 Atualiza hash da senha no banco
 - 6 Inválida token temporário
- **Fluxo Alternativo:** Erro inesperado → retorna mensagem de erro.

5.2 Diagrama de Máquina de estado

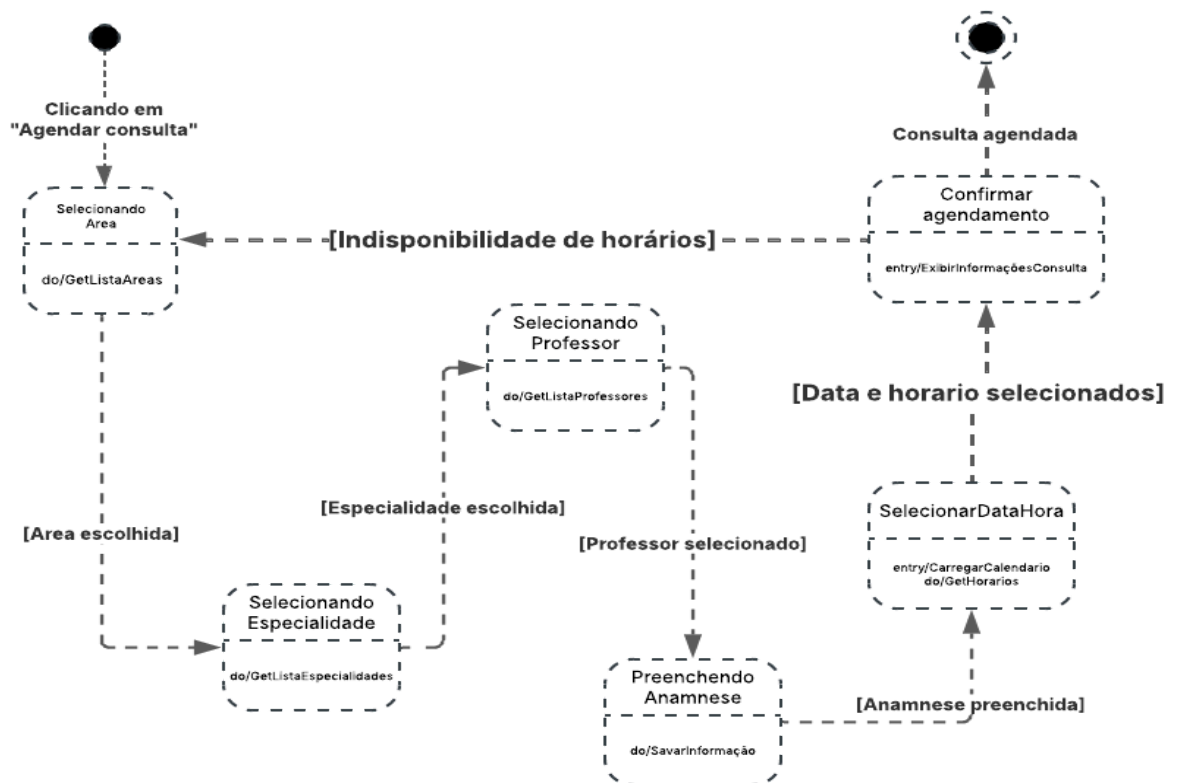
UC-01/UC - API01



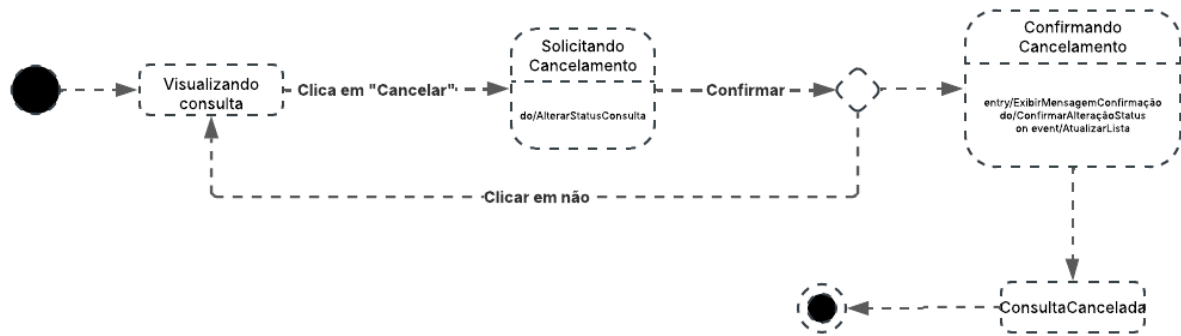
UC-02/UC - API02



UC-03/UC - API03



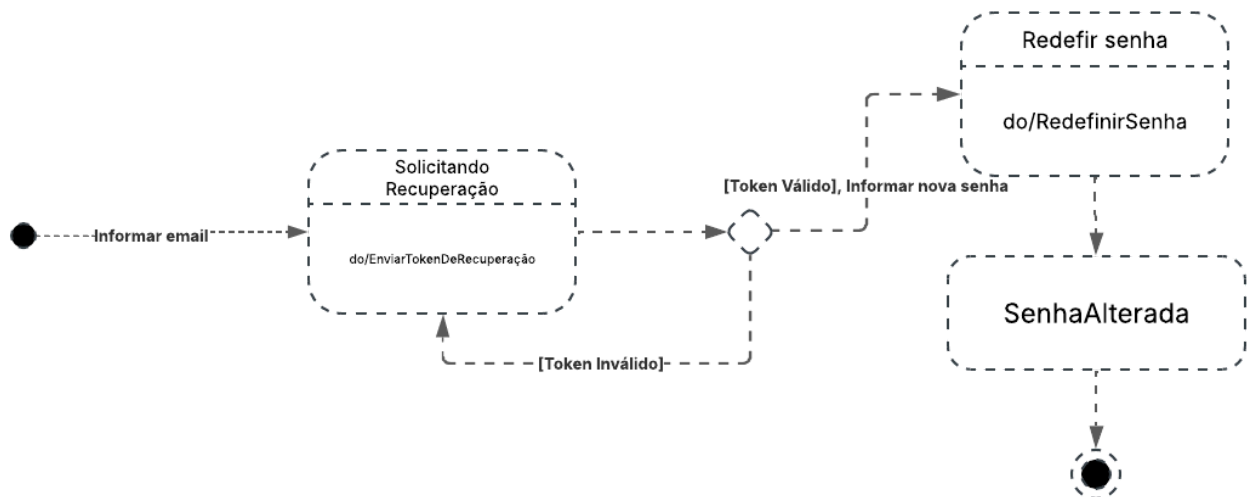
UC-04/UC - API04



UC05



UC-06/UC - API05



5.3 Diagrama Entidade-Relacionamento (ER)

Representa as entidades principais do sistema e seus relacionamentos:

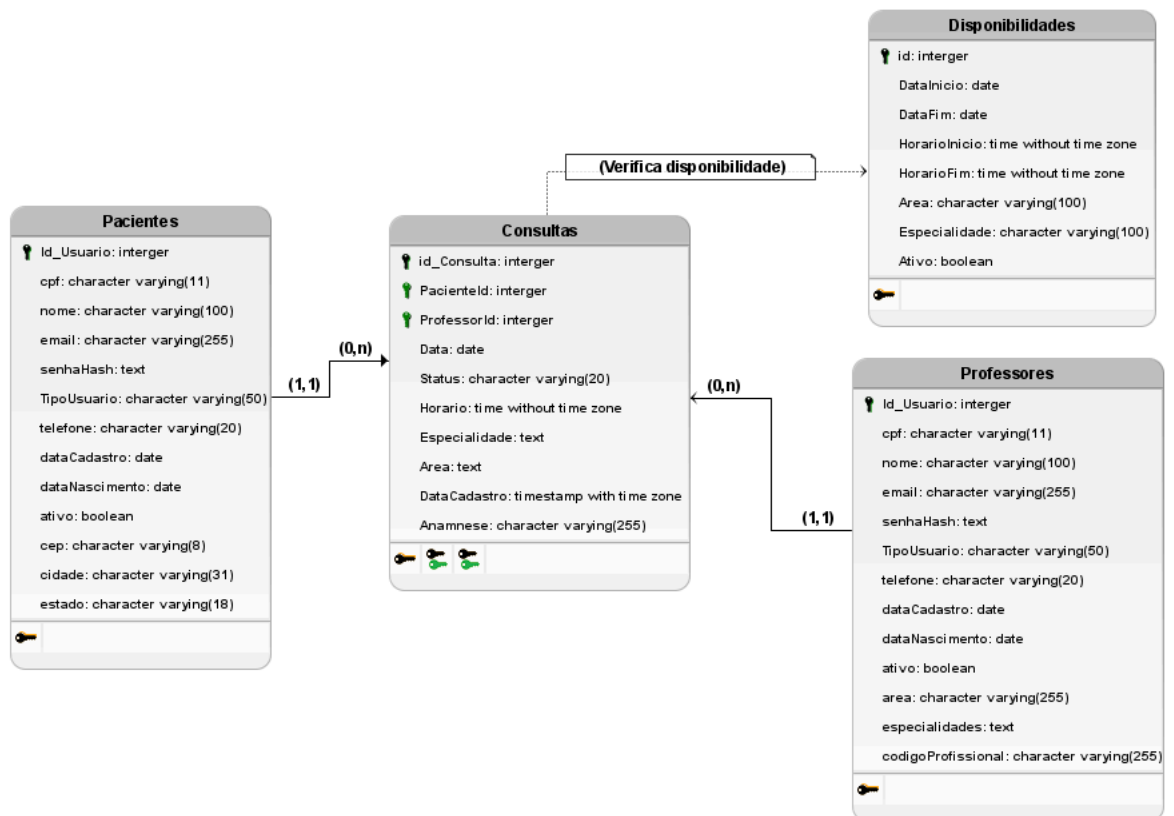
- **Paciente registra Consulta**

- Consultas verifica Disponibilidades
- Consultas integra Professor

5.4 Diagrama Lógico

Mostra as tabelas com seus atributos e relacionamentos lógicos:

- Chaves primárias (PK) e estrangeiras (FK)
- Relacionamentos 1:N e 1:1



5.5 Diagrama Físico

Detalha os tipos de dados e estrutura física do banco de dados:

Tabela pacientes

```
CREATE TABLE public."Pacientes" (  
  "Id_Usuario" integer NOT NULL GENERATED BY DEFAULT AS IDENTITY,  
  cpf character varying(11) NOT NULL,  
  nome character varying(100) NOT NULL,  
  email character varying(255) NOT NULL,  
  "senhaHash" text NOT NULL,  
  telefone character varying(20) NOT NULL,  
  "dataCadastro" date NOT NULL,  
  "dataNascimento" date,  
  "TipoUsuario" character varying(50) NOT NULL,  
  ativo boolean DEFAULT true NOT NULL,  
  cep character varying(8) DEFAULT "::character varying NOT NULL,  
  cidade character varying(31) DEFAULT "::character varying NOT NULL,  
  estado character varying(18) DEFAULT "::character varying NOT NULL,  
  CONSTRAINT "PK_Pacientes" PRIMARY KEY ("Id_Usuario")  
);  
  
CREATE UNIQUE INDEX "IX_Pacientes_cpf" ON public."Pacientes" USING btree (cpf);  
CREATE UNIQUE INDEX "IX_Pacientes_email" ON public."Pacientes" USING btree (email);  
CREATE UNIQUE INDEX "IX_Pacientes_telefone" ON public."Pacientes" USING btree (telefone);
```

Tabela Professores

```
CREATE TABLE public."Professores" (  
  "Id_Usuario" integer NOT NULL GENERATED BY DEFAULT AS IDENTITY,  
  area character varying(255) NOT NULL,  
  "codigoProfissional" character varying(255) NOT NULL,  
  cpf character varying(11) NOT NULL,  
  nome character varying(100) NOT NULL,  
  email character varying(255) NOT NULL,  
  "senhaHash" text NOT NULL,  
  telefone character varying(20) NOT NULL,  
  "dataCadastro" date NOT NULL,  
  "dataNascimento" date,  
  "TipoUsuario" character varying(50) NOT NULL,  
  ativo boolean DEFAULT true NOT NULL,  
  especialidades text DEFAULT "":text NOT NULL,  
  CONSTRAINT "PK_Professores" PRIMARY KEY ("Id_Usuario")  
);  
  
CREATE UNIQUE INDEX "IX_Professores_cpf" ON public."Professores" USING btree (cpf);  
CREATE UNIQUE INDEX "IX_Professores_email" ON public."Professores" USING btree (email);  
CREATE UNIQUE INDEX "IX_Professores_telefone" ON public."Professores" USING btree (telefone);
```

Tabela Disponibilidades

```
CREATE TABLE public."Disponibilidades" (  
  "Id" integer NOT NULL GENERATED BY DEFAULT AS IDENTITY,  
  "DataInicio" date NOT NULL,  
  "DataFim" date NOT NULL,  
  "HorarioInicio" time without time zone NOT NULL,  
  "HorarioFim" time without time zone NOT NULL,  
  "Area" character varying(100) NOT NULL,  
  "Especialidade" character varying(100) NOT NULL,  
  "Ativo" boolean DEFAULT true NOT NULL,  
  CONSTRAINT "PK_Disponibilidades" PRIMARY KEY ("Id")  
);
```

Tabela Consultas

```
CREATE TABLE public."Consultas" (
  "id_Consulta" integer NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  "Data" date,
  "Horario" time without time zone,
  "Status" character varying(20) NOT NULL,
  "PacientId" integer NOT NULL,
  "ProfessorId" integer NOT NULL,
  "Especialidade" text DEFAULT "":text NOT NULL,
  "Area" text DEFAULT "":text NOT NULL,
  "DataCadastro" timestamp with time zone DEFAULT '-infinity':timestamp with time zone NOT NULL,
  "Anamnese" character varying(255) DEFAULT "":character varying NOT NULL,
  CONSTRAINT "PK_Consultas" PRIMARY KEY ("id_Consulta"),
  CONSTRAINT "FK_Consultas_Pacientes_PacientId" FOREIGN KEY ("PacientId") REFERENCES
public."Pacientes"("Id_Usuario") ON DELETE RESTRICT,
  CONSTRAINT "FK_Consultas_Professores_ProfessorId" FOREIGN KEY ("ProfessorId")
REFERENCES public."Professores"("Id_Usuario") ON DELETE RESTRICT
);

CREATE INDEX "IX_Consultas_PacientId" ON public."Consultas" USING btree ("PacientId");
CREATE INDEX "IX_Consultas_ProfessorId" ON public."Consultas" USING btree ("ProfessorId");
```

6. Documentação de Testes

6.1 Plano de Testes

Objetivo: Verificar se o sistema atende aos requisitos funcionais e não funcionais.

Escopo:

- Testes funcionais (cadastro, agendamento, notificações)
- Testes de interface
- Testes de desempenho
- Testes de segurança

Ambiente de Testes:

- Android 8+
- Banco de dados PostgreSQL

Casos de Teste

Código	Caso de Teste	Entrada	Resultado	Esperado	Requisito
TC01	Realizar	Email e senha	Direcionament	Direcionamento para tela	Estar

	login		o para tela inicial	inicial	cadastrado
TC02	Visualizar/ Selecionar status da consulta	Seleção de Status	Visualização das consultas do paciente pelo status escolhido	Visualização das consultas do paciente pelo status escolhido	Estar autenticado
TC03	Agendar consulta	Área, especialidade, professor, anamnese básica, data e hora.	Agendamento da consulta	Agendamento da consulta	Estar autenticado
TC04	Cancelar consulta	Confirmação de cancelamento	A consulta é cancelada	A consulta é cancelada e movida para canceladas	Estar autenticado
TC05	Deslogar		Direcionament o para a tela de login	Direcionamento para a tela de login	Estar autenticado
TC06	Recuperar senha	Email	Recebimento de email com link para alteração de senha	Recebimento de email com link para alteração de senha	Estar cadastrado com email válido

6.2 Critérios de Aceitação

- Todos os testes devem passar com sucesso.
- Nenhum erro crítico deve ser identificado.
- A interface deve ser responsiva em diferentes tamanhos de tela.
- O tempo de resposta deve ser inferior a 1 minuto (RNF02).

7. Documentação de Segurança

7.1 Autenticação e Autorização

- **Autenticação por e-mail e senha** com validação de formato e força da senha.
- **Tokens JWT** para sessões seguras e controle de acesso.

- **Controle de sessão** com expiração automática e logout manual.

7.2 Criptografia

- **Hash de senhas** com **bcrypt** antes do armazenamento.
- **Comunicação segura** via **HTTPS/TLS 1.3** para todas as requisições.

7.3 Armazenamento Seguro

- Dados locais armazenados com **SQLite criptografado**.
- Backup criptografado e sincronizado com o servidor.

7.4 Proteção contra Ameaças

- **Validação de entrada** para evitar injeção de SQL e XSS.

7.5 Conformidade com LGPD

- Consentimento explícito para coleta de dados.
- Opção de **excluir conta e dados** permanentemente.
- Política de privacidade clara e acessível no app.