SISTEMA DE TRIAGEM MÉDICA

Documentação Técnica Completa

Sumário

1. Visão Geral

O **Sistema de Triagem Médica** é uma aplicação web completa desenvolvida para modernizar o processo de triagem em ambientes de saúde. Este sistema foi projetado para otimizar o fluxo de atendimento médico, permitindo uma avaliação inicial dos pacientes baseada em seus sintomas e condições de saúde.

O processo inicia-se com o preenchimento de um formulário de triagem pelo paciente, onde são coletadas informações sobre suas condições prévias, como:

- ✓ Status de obesidade, hipertensão ou diabetes
- ✓ Presença de febre e temperatura corporal
- ✓ Presença, localização e intensidade de dor
- ✓ Alergias conhecidas
- ✓ Medicamentos em uso
- ✓ Dados biométricos (peso e idade)

Após o preenchimento do formulário, o sistema implementa uma lógica de classificação que analisa os dados fornecidos e determina a gravidade do caso do paciente. Com base nesta avaliação, o sistema estabelece uma prioridade de atendimento, permitindo que o responsável pela gestão dos agendamentos visualize o resultado de todas as triagens e organize as consultas de acordo com a urgência de cada caso.

Quando o agendamento é concluído, o paciente pode visualizá-lo ao fazer login no sistema, tendo acesso às informações de data, hora, local e médico responsável, além de poder confirmar seu comparecimento. Após a consulta, o administrador ou médico responsável registra o histórico médico do paciente, documentando todo o processo desde a triagem até o resultado final, incluindo diagnóstico, prescrições médicas, orientações e exames solicitados. Este histórico fica permanentemente disponível para acesso pelo paciente.

A solução proporciona uma interface intuitiva que permite a avaliação preliminar automatizada e o acompanhamento de todo o processo, desde a triagem até o atendimento médico e seu histórico clínico. O sistema categoriza os casos por gravidade (crítica, grave ou leve), facilitando a priorização do atendimento e a alocação eficiente de recursos médicos.

2. Objetivos do Sistema

O Sistema de Triagem Médica tem como principais objetivos:

- ✓ Automatizar o processo de triagem médica, reduzindo o tempo de espera dos pacientes
- ✓ Otimizar o agendamento de consultas baseado na gravidade dos casos
- ✓ Manter um registro completo do histórico médico dos pacientes
- ✓ Facilitar o acesso aos dados clínicos para pacientes e profissionais de saúde
- ✓ Aumentar a eficiência operacional em ambientes de assistência médica
- ✓ Proporcionar uma experiência intuitiva para todos os usuários do sistema
- ✓ Reduzir a sobrecarga administrativa das equipes de saúde
- ✓ Melhorar a qualidade do atendimento através da priorização adequada de casos

3. Público-alvo

O sistema foi desenvolvido para atender às necessidades de:

- **Pacientes:** Que necessitam de avaliação médica e desejam um processo simplificado de triagem e agendamento
- **Administradores do sistema:** Responsáveis por gerenciar o fluxo de atendimento, agendamentos e registros médicos
- **Profissionais de saúde:** Médicos e enfermeiros que necessitam de acesso rápido ao histórico dos pacientes
- **Gestores de instituições de saúde:** Interessados em otimizar recursos e melhorar a eficiência operacional
- **Equipe de recepção e acolhimento:** Responsáveis pelo primeiro contato com pacientes e organização do fluxo de atendimento

4. Arquitetura do Sistema

Visão Geral da Arquitetura

O Sistema de Triagem Médica utiliza uma arquitetura cliente-servidor moderna, com separação clara entre frontend e backend:

- ✓ **Frontend:** Desenvolvido em React com Vite, proporciona uma interface de usuário responsiva e intuitiva
- ✓ **Backend:** Construído com Node.js, gerencia a lógica de negócios, autenticação e comunicação com o banco de dados
- ✓ **Banco de Dados:** PostgreSQL, armazena de forma estruturada e relacional todos os dados do sistema
- ✓ **API RESTful:** Proporciona comunicação eficiente entre frontend e backend

Esta arquitetura permite escalabilidade, manutenção simplificada e uma experiência de usuário consistente em diferentes dispositivos.

5. Requisitos do Sistema

5.1 Requisitos Funcionais

RF001 - Autenticação de Usuários

- O sistema deve permitir o cadastro de novos usuários (pacientes)

- O sistema deve autenticar usuários através de e-mail e senha
- O sistema deve permitir os usuários uma recuperação de senha.
- O sistema deve implementar diferentes acesso (paciente e administrador)

RF002 - Triagem Automatizada

- O sistema deve apresentar formulário para coleta de informações de saúde
- O sistema deve processar os dados fornecidos e calcular a gravidade do caso
- O sistema deve classificar os casos em três níveis: crítico, grave e leve
- O sistema deve armazenar todas as informações da triagem

RF003 - Gestão de Consultas

- O sistema deve permitir que administradores visualizem todas as triagens
- O sistema deve sugerir prioridades baseadas na gravidade das triagens
- O sistema deve permitir agendamento de consultas com definição de data, hora, local e médico
- O sistema deve notificar pacientes sobre consultas agendadas

RF004 - Confirmação de Presença

- O sistema deve exibir consultas agendadas para o paciente autenticado
- O sistema deve permitir que pacientes confirmem seu comparecimento
- O sistema deve registrar as confirmações e disponibilizá-las para administradores

RF005 - Histórico Médico

- O sistema deve permitir registro de informações completas após as consultas
- O sistema deve incluir campos para diagnósticos, medicações e observações
- O sistema deve disponibilizar histórico completo para visualização pelo paciente
- O sistema deve permitir pesquisa e filtro de registros históricos

RF006 - Notificações

- O sistema deve enviar notificações sobre consultas agendadas
- O sistema deve alertar administradores sobre casos críticos

5.2 Requisitos Não Funcionais

RNF001 - Usabilidade

- A interface deve ser intuitiva e acessível para todos os perfis de usuários
- O sistema deve ser responsivo e adaptável a diferentes tamanhos de tela

- O tempo médio para completar a triagem não deve exceder 5 minutos
- O sistema deve seguir padrões de acessibilidade WCAG 2.1 nível AA

RNF002 - Segurança

- Todas as senhas devem ser armazenadas utilizando algoritmos com tokem
- A comunicação entre cliente e servidor deve ser criptografada (HTTPS)
- Os tokens de autenticação devem expirar em 24 horas

RNF003 - Desempenho

- O tempo de resposta para operações regulares não deve exceder 2 segundos
- O sistema deve suportar até 100 usuários simultâneos
- O tempo de carregamento inicial não deve exceder 3 segundos em conexões
- As consultas ao banco de dados devem ser otimizadas para evitar gargalos

RNF004 - Escalabilidade

- O banco de dados deve ser dimensionado para suportar crescimento de até 50% ao ano
- O sistema deve ser modular para permitir expansão de funcionalidades

RNF005 - Disponibilidade

- O sistema deve estar disponível 99,9% do tempo (máximo de 8,76 horas/ano)
- Manutenções programadas devem ocorrer fora do horário comercial

RNF006 - Compatibilidade

- O sistema deve funcionar nos navegadores Chrome, Firefox, Safari e Edge (duas versões mais recentes)
- A interface deve ser responsiva para dispositivos móveis, tablets e desktops

6. Diagramas

Diagrama de Arquitetura

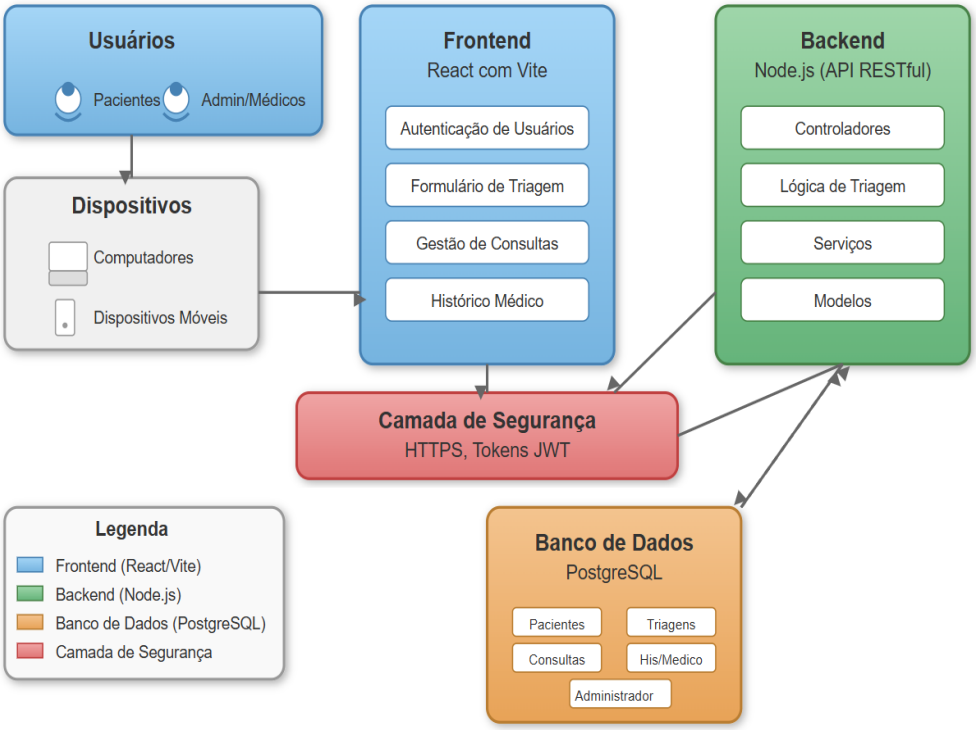


Diagrama de Casos de Uso

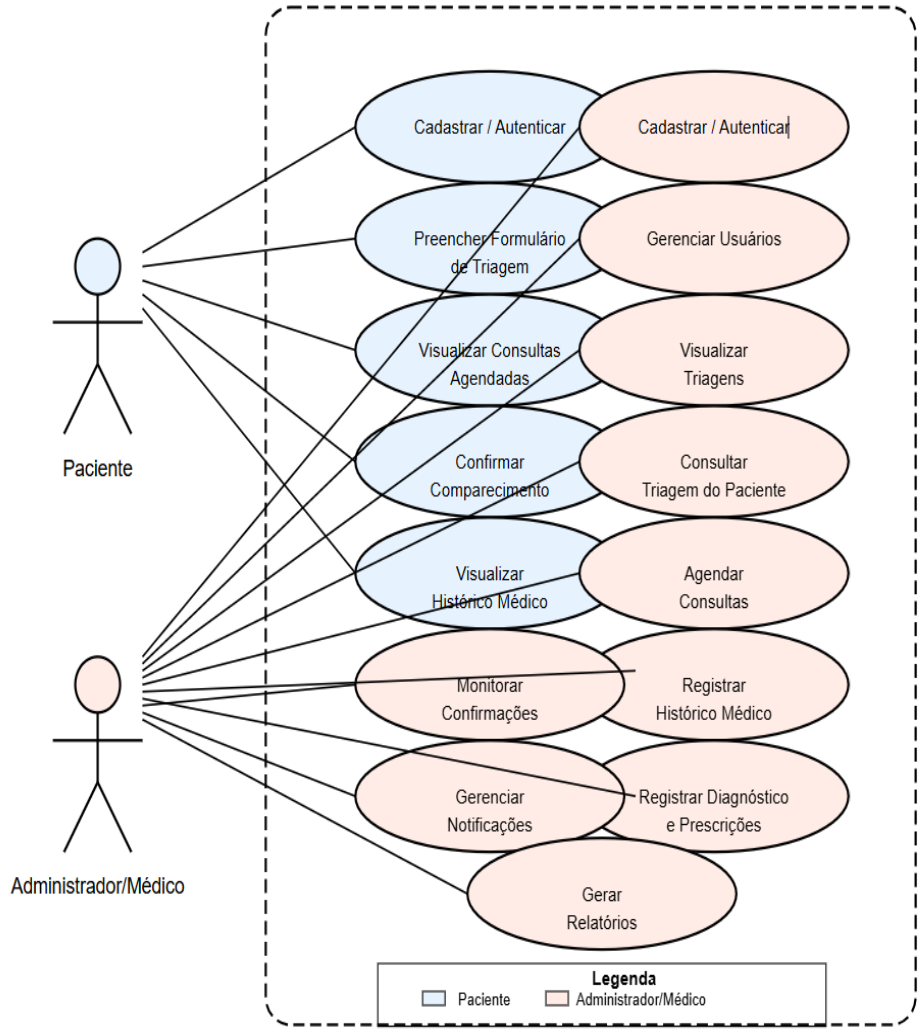


Diagrama de Classes

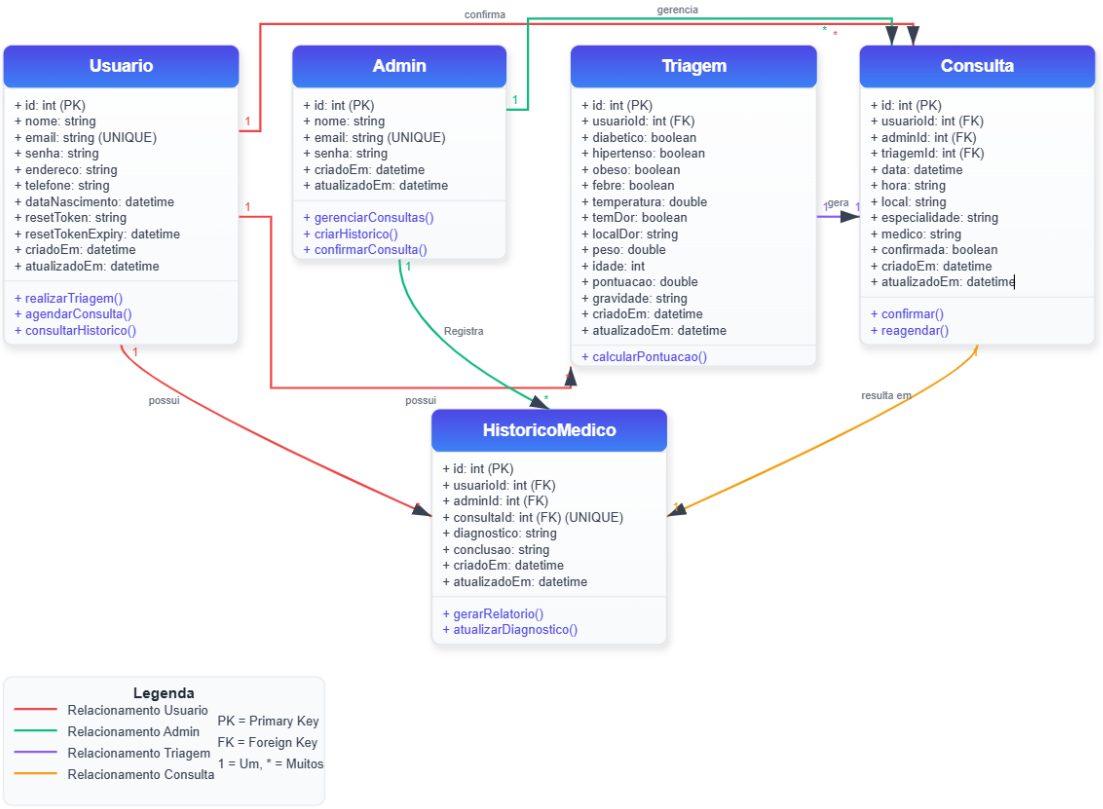


Diagrama Entidade-Relacionamento (DER).

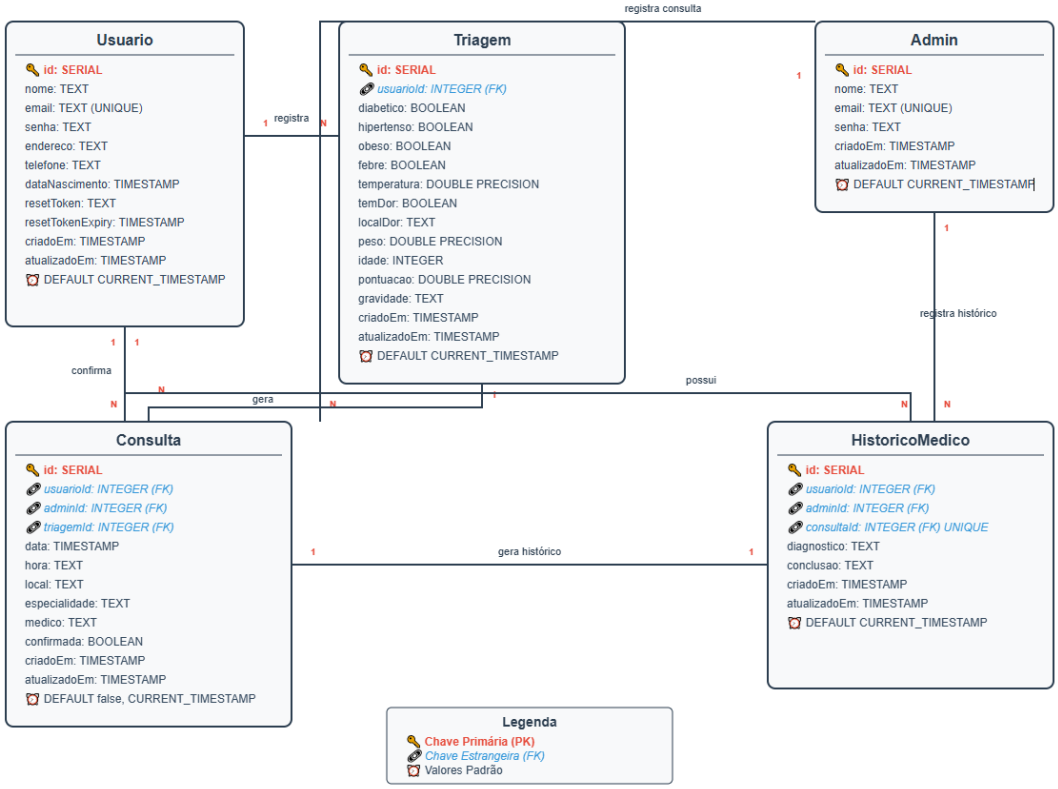
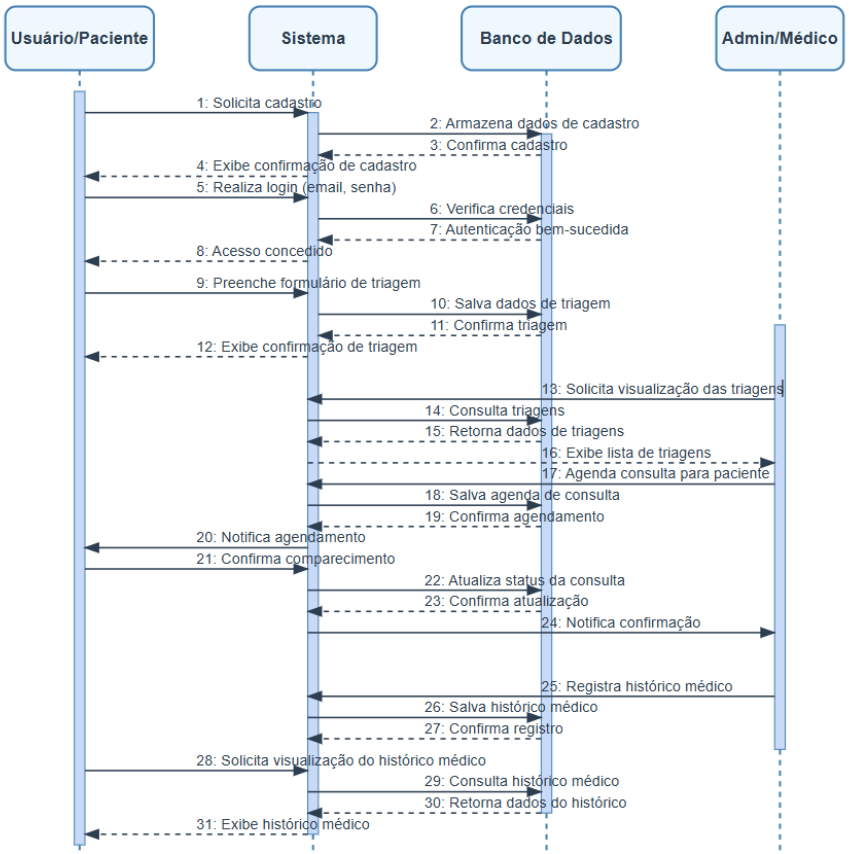


Diagrama de Sequência para Triagem



7. Stack Tecnológica

Frontend

- React
- Vite
- Axios

Backend

- Node.js
- Prisma
- JWT (JSON Web Tokens)

Banco de Dados

- PostgreSQL
- pgAdmin 4
- Prisma ORM

8. Dicionário de Dados

Tabela: Usuario

Armazena informações de usuários do sistema (pacientes).

Campo	Tipo	Descrição	Restrições
id	SERIAL	Identificador único do usuário	Chave primária, auto-incremento
nome	TEXT	Nome completo do usuário	Não nulo
email	TEXT	Endereço de e-mail do usuário	Não nulo, único
senha	TEXT	Senha do usuário (criptografada)	Não nulo
endereco	TEXT	Endereço residencial	Não nulo
telefone	TEXT	Número de telefone para contato	Não nulo
dataNascimento	TIMESTAMP(3)	Data de nascimento do usuário	Não nulo
resetToken	TEXT	Token de Recuperação de senha	Não nulo
resetTokenExpiry	TIMESTAMP()	Tempo de expiração do tokem	Não nulo

criadoEm	TIMESTAMP(3)	Data e hora de criação do registro	Não nulo, valor padrão: timestamp atual
atualizadoEm	TIMESTAMP(3)	Data e hora da última atualização	Não nulo

Tabela: Admin

Armazena informações de administradores do sistema.

Campo	Tipo	Descrição	Restrições
id	SERIAL	Identificador único do administrador	Chave primária, auto-incremento
nome	TEXT	Nome completo do administrador	Não nulo
email	TEXT	Endereço de e-mail do administrador	Não nulo, único
senha	TEXT	Senha do administrador (criptografada)	Não nulo
criadoEm	TIMESTAMP(3)	Data e hora de criação do registro	Não nulo, valor padrão: timestamp atual
atualizadoEm	TIMESTAMP(3)	Data e hora da última atualização	Não nulo

Tabela: Triagem

Armazena informações da triagem médica inicial dos usuários.

Campo	Tipo	Descrição	Restrições
id	SERIAL	Identificador único da triagem	Chave primária, auto-incremento
usuarioId	INTEGER	ID do usuário relacionado	Não nulo, chave estrangeira referenciando Usuario(id)
diabetico	BOOLEAN	Indica se o paciente é diabético	Não nulo
hipertenso	BOOLEAN	Indica se o paciente é hipertenso	Não nulo
obeso	BOOLEAN	Indica se o paciente é obeso	Não nulo
febre	BOOLEAN	Indica se o paciente está com febre	Não nulo
temperatura	DOUBLE PRECISION	Temperatura corporal do paciente	Pode ser nulo
temDor	BOOLEAN	Indica se o paciente está com dor	Não nulo
localDor	TEXT	Localização da dor no corpo	Pode ser nulo

peso	DOUBLE PRECISION	Peso do paciente em kg	Não nulo
idade	INTEGER	Idade do paciente	Não nulo
pontuacao	DOUBLE PRECISION	Pontuação de gravidade da triagem	Não nulo
gravidade	TEXT	Classificação de gravidade do caso	Não nulo
criadoEm	TIMESTAMP(3)	Data e hora de criação do registro	Não nulo, valor padrão: timestamp atual
atualizadoEm	TIMESTAMP(3)	Data e hora da última atualização	Não nulo

Tabela: Consulta

Armazena informações das consultas médicas.

Campo	Tipo	Descrição	Restrições
id	SERIAL	Identificador único da consulta	Chave primária, auto-incremento
usuarioId	INTEGER	ID do usuário (paciente)	Não nulo, chave estrangeira referenciando Usuario(id)
adminId	INTEGER	ID do administrador	Não nulo, chave estrangeira referenciando Admin(id)
triagemId	INTEGER	ID da triagem relacionada	Não nulo, chave estrangeira referenciando Triagem(id)
data	TIMESTAMP(3)	Data da consulta	Não nulo
hora	TEXT	Hora da consulta	Não nulo
local	TEXT	Local da consulta	Não nulo
especialidade	TEXT	Especialidade médica	Não nulo
medico	TEXT	Nome do médico responsável	Não nulo
confirmada	BOOLEAN	Status de confirmação da consulta	Não nulo, valor padrão: false
criadoEm	TIMESTAMP(3)	Data e hora de criação do registro	Não nulo, valor padrão: timestamp atual
atualizadoEm	TIMESTAMP(3)	Data e hora da última atualização	Não nulo

Tabela: HistoricoMedico

Armazena informações do histórico médico relacionado a cada consulta.

Campo	Tipo	Descrição	Restrições
id	SERIAL	Identificador único do histórico	Chave primária, auto-incremento

usuarioId	INTEGER	ID do usuário (paciente)	Não nulo, chave estrangeira referenciando Usuario(id)
adminId	INTEGER	ID do administrador	Não nulo, chave estrangeira referenciando Admin(id)
consultaId	INTEGER	ID da consulta relacionada	Não nulo, chave estrangeira referenciando Consulta(id), único
diagnostico	TEXT	Diagnóstico médico	Não nulo
conclusao	TEXT	Conclusão do atendimento	Não nulo
criadoEm	TIMESTAMP(3)	Data e hora de criação do registro	Não nulo, valor padrão: timestamp atual
atualizadoEm	TIMESTAMP(3)	Data e hora da última atualização	Não nulo

9 Relacionamentos

1. Usuario → Triagem: Um usuário pode ter várias triagens (1:N)
    a. Chave estrangeira: Triagem.usuarioId → Usuario.id
2. Usuario → Consulta: Um usuário pode ter várias consultas (1:N)
    a. Chave estrangeira: Consulta.usuarioId → Usuario.id
3. Admin → Consulta: Um administrador pode gerenciar várias consultas (1:N)
    a. Chave estrangeira: Consulta.adminId → Admin.id
4. Triagem → Consulta: Uma triagem está associada a uma consulta (1:1)
    a. Chave estrangeira: Consulta.triagemId → Triagem.id
5. Usuario → HistoricoMedico: Um usuário pode ter vários registros no histórico médico (1:N)
    a. Chave estrangeira: HistoricoMedico.usuarioId → Usuario.id
6. Admin → HistoricoMedico: Um administrador pode criar vários registros no histórico médico (1:N)
    a. Chave estrangeira: HistoricoMedico.adminId → Admin.id
7. Consulta → HistoricoMedico: Uma consulta tem exatamente um registro no histórico médico (1:1)
    a. Chave estrangeira: HistoricoMedico.consultaId → Consulta.id (com restrição UNIQUE)

Triagem → Usuario

- Uma triagem pertence a um usuário específico
- Um usuário pode ter múltiplas triagens

Consulta → Usuario

- Uma consulta pertence a um usuário específico
- Um usuário pode ter múltiplas consultas

Consulta → Admin

- Uma consulta é gerenciada por um administrador específico
- Um administrador pode gerenciar múltiplas consultas

Consulta → Triagem

- Uma consulta está associada a uma triagem específica
- Uma triagem pode resultar em uma consulta

HistoricoMedico → Usuario

- Um histórico médico pertence a um usuário específico
- Um usuário pode ter múltiplos históricos médicos

HistoricoMedico → Admin

- Um histórico médico é registrado por um administrador específico
- Um administrador pode registrar múltiplos históricos médicos

HistoricoMedico → Consulta

- Um histórico médico está associado a uma consulta específica
- Uma consulta gera apenas um histórico médico (relacionamento 1:1)

10. Funcionalidades do Sistema

10.1 Gestão de Usuários

O módulo de Gestão de Usuários permite:

1. Cadastro de Novos Usuários

- ✓ Formulário completo para captação de dados pessoais
- ✓ Validação de informações em tempo real
- ✓ Criação de credenciais de acesso seguras

2. Autenticação

- ✓ Login seguro com verificação de credenciais
- ✓ Diferenciação entre perfis de acesso (paciente/administrador)

3. Controle de Acesso

- ✓ Permissionamento baseado em perfis
- ✓ Acesso restrito a funcionalidades administrativas
- ✓ Registro de logs de atividades
- ✓ Auditoria de ações sensíveis

10.2 Processo de Triagem

O módulo de Triagem automatiza a avaliação inicial de pacientes:

1. Coleta de Informações

- Questionário estruturado sobre condições de saúde:
  - Condições pré-existentes (diabetes, hipertensão, obesidade)
  - Sintomas atuais (febre, dor)
  - Dados biométricos (peso, temperatura)
- Interface acessível e intuitiva
- Progresso do preenchimento visível ao usuário

2. Algoritmo de Classificação

- Processamento das informações coletadas
- Cálculo de pontuação baseado em critérios médicos predefinidos
- Categorização da gravidade (crítica, grave, leve)
- Regras de negócio implementadas para avaliação consistente

3. Resultado da Triagem

- Exibição do resultado com classificação de gravidade
- Recomendações iniciais baseadas na classificação
- Armazenamento dos dados para consulta posterior
- Alertas automáticos para casos críticos

4. Métricas e Relatórios

- Distribuição de casos por gravidade
- Tempo médio entre triagem e atendimento
- Relatórios exportáveis para análise

10.3 Agendamento de Consultas

O módulo de Agendamento gerencia o processo de marcação de consultas:

1. Criação de Agendamentos

- Interface administrativa para definição de:
  - Data e hora
  - Local de atendimento
  - Especialidade médica
  - Profissional responsável
- Verificação de disponibilidade em tempo real

2. Priorização Baseada em Triagem

- Alertas para casos críticos
- Visualização de triagens pendentes ordenadas por urgência
- Dashboard com resumo de casos aguardando agendamento

3. Notificação ao Paciente

- Comunicação automática sobre detalhes do agendamento
- Solicitação de confirmação de presença
- Lembretes próximos à data da consulta

4. Confirmação de Comparecimento

- Interface para que o paciente confirme sua presença
- Registro da confirmação no sistema

5. Gerenciamento de Agenda

- Visualização de calendário com todas as consultas
- Filtros por médico, especialidade ou período
- Detecção de sobrecarga de agenda
- Ajustes em tempo real quando necessário

10.4 Histórico Médico

O módulo de Histórico Médico mantém o registro completo de atendimentos:

1. Registro Pós-Consulta

- Interface para administradores registrarem:
  - Diagnóstico
  - Conclusões médicas
  - Medicamentos prescritos
  - Exames solicitados
  - Orientações ao paciente
- Vinculação automática com a consulta realizada

2. Visualização do Histórico

- Interface para pacientes acessarem seu histórico completo
- Organização cronológica dos atendimentos
- Detalhamento de cada consulta e seus resultados
- Visualização de evolução do quadro de saúde

3. Pesquisa e Filtros

- Filtros para visualização específica de informações

4. Privacidade e Segurança

- Acesso restrito apenas ao próprio paciente e profissionais autorizados
- Criptografia de dados sensíveis
- Conformidade com legislações de proteção de dados

11. Estrutura do Banco de Dados

```
-- CreateTable
```

```
CREATE TABLE "Usuario" (
"id" SERIAL NOT NULL,
"nome" TEXT NOT NULL,
"email" TEXT NOT NULL,
"senha" TEXT NOT NULL,
"endereco" TEXT NOT NULL,
"telefone" TEXT NOT NULL,
“resetToken” TEXT NOT NULL,
“resetTokenExpiry” TIMESTAMP(3) NOT NULL,
"dataNascimento" TIMESTAMP(3) NOT NULL,
"criadoEm" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP, "atualizadoEm" TIMESTAMP(3)
NOT NULL,
CONSTRAINT "Usuario_pkey" PRIMARY KEY ("id")
);


-- CreateTable
CREATE TABLE "Admin" (
"id" SERIAL NOT NULL,
"nome" TEXT NOT NULL,
"email" TEXT NOT NULL,
"senha" TEXT NOT NULL,
"criadoEm" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP, "atualizadoEm" TIMESTAMP(3)
NOT NULL,
CONSTRAINT "Admin_pkey" PRIMARY KEY ("id")
);

-- CreateTable
CREATE TABLE "Triagem" (
"id" SERIAL NOT NULL,
"usuarioId" INTEGER NOT NULL,
"diabetico" BOOLEAN NOT NULL,
"hipertenso" BOOLEAN NOT NULL,
"obeso" BOOLEAN NOT NULL,
"febre" BOOLEAN NOT NULL,
"temperatura" DOUBLE PRECISION,
"temDor" BOOLEAN NOT NULL,
"localDor" TEXT,
```

```
"peso" DOUBLE PRECISION NOT NULL,
"idade" INTEGER NOT NULL,
"pontuacao" DOUBLE PRECISION NOT NULL,
"gravidade" TEXT NOT NULL,
"criadoEm" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP, "atualizadoEm" TIMESTAMP(3) NOT NULL,
CONSTRAINT "Triagem_pkey" PRIMARY KEY ("id")
);

-- CreateTable
CREATE TABLE "Consulta" (
"id" SERIAL NOT NULL,
"usuarioId" INTEGER NOT NULL,
"adminId" INTEGER NOT NULL,
"triagemId" INTEGER NOT NULL,
"data" TIMESTAMP(3) NOT NULL,
"hora" TEXT NOT NULL,
"local" TEXT NOT NULL,
"especialidade" TEXT NOT NULL,
"medico" TEXT NOT NULL,
"confirmada" BOOLEAN NOT NULL DEFAULT false,
"criadoEm" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP, "atualizadoEm" TIMESTAMP(3) NOT NULL,
CONSTRAINT "Consulta_pkey" PRIMARY KEY ("id")
);

-- CreateTable
CREATE TABLE "HistoricoMedico" (
"id" SERIAL NOT NULL,
"usuarioId" INTEGER NOT NULL,
"adminId" INTEGER NOT NULL,
"consultaId" INTEGER NOT NULL,
"diagnostico" TEXT NOT NULL,
"conclusao" TEXT NOT NULL,
"criadoEm" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP, "atualizadoEm" TIMESTAMP(3) NOT NULL,
CONSTRAINT "HistoricoMedico_pkey" PRIMARY KEY ("id")
);
```

```
-- CreateIndex
CREATE UNIQUE INDEX "Usuario_email_key" ON "Usuario"("email");

-- CreateIndex
CREATE UNIQUE INDEX "Admin_email_key" ON "Admin"("email");

-- CreateIndex
CREATE UNIQUE INDEX "HistoricoMedico_consultaId_key" ON "HistoricoMedico"("consultaId");

-- AddForeignKey
ALTER TABLE "Triagem" ADD CONSTRAINT "Triagem_usuarioId_fkey" FOREIGN KEY ("usuarioId")
REFERENCES "Usuario"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "Consulta" ADD CONSTRAINT "Consulta_usuarioId_fkey" FOREIGN KEY ("usuarioId")
REFERENCES "Usuario"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "Consulta" ADD CONSTRAINT "Consulta_adminId_fkey" FOREIGN KEY ("adminId")
REFERENCES "Admin"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "Consulta" ADD CONSTRAINT "Consulta_triagemId_fkey" FOREIGN KEY ("triagemId")
REFERENCES "Triagem"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "HistoricoMedico" ADD CONSTRAINT "HistoricoMedico_usuarioId_fkey" FOREIGN KEY
("usuarioId") REFERENCES "Usuario"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "HistoricoMedico" ADD CONSTRAINT "HistoricoMedico_adminId_fkey" FOREIGN KEY
("adminId") REFERENCES "Admin"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "HistoricoMedico" ADD CONSTRAINT "HistoricoMedico_consultaId_fkey" FOREIGN KEY
("consultaId") REFERENCES "Consulta"("id") ON DELETE RESTRICT ON UPDATE CASCADE;
```

12. Configuração e Instalação

Pré-requisitos

- Node.js ≥ 18.x
- PostgreSQL ≥ 13.x
- PgAdmin 4
- Git

Guia de Instalação

Clone o Repositório

git clone https://github.com/seu-usuario/sistema-triagem-medica.git
cd sistema-triagem-medica

Instale as Dependências

npm install

Instale a Dependência para o envio de Email de recuperação de senha

npm install nodemailer

Configure o Ambiente

Crie um arquivo .env na raiz do projeto com o seguinte conteúdo:

DATABASE_URL="postgresql://postgres:sua_senha@localhost:5432/triagem_medica?schema=public"
JWT_SECRET="sua_chave_super_secreta"

Importante: Substitua sua_senha e sua_chave_super_secreta pelos valores adequados ao seu ambiente.

Configure o Banco de Dados com pgAdmin 4

1. Abra o pgAdmin 4
2. Registre um Novo Servidor
    a. Clique com o botão direito em "Servidores"
    b. Selecione "Registrar" e depois "Servidor"
3. Configure a Aba "Geral"
    a. Nome: triagem_medica
4. Configure a Aba "Conexão"
    a. Host: localhost
    b. Porta: 5432
    c. Banco de dados de manutenção: postgres
    d. Nome de usuário: postgres
    e. Senha: sua_senha
    f. Clique em "Salvar"
5. Crie um Novo Banco de Dados
    a. No servidor recém-criado, clique com o botão direito em "Bancos de Dados"
    b. Selecione "Criar" e depois "Banco de Dados"
    c. Nome do banco de dados: triagem_medica
    d. Proprietário: postgres
    e. Clique em "Salvar"

Configure o Prisma e Realize a Migração do Banco

npx prisma migrate
npx prisma migrate dev --name init

Inicie o Servidor Frontend

npm run dev
Acesse em: http://localhost:5173 (ou outra porta indicada)

Inicie o Servidor Backend

npm run server
Servidor rodando na porta 3001

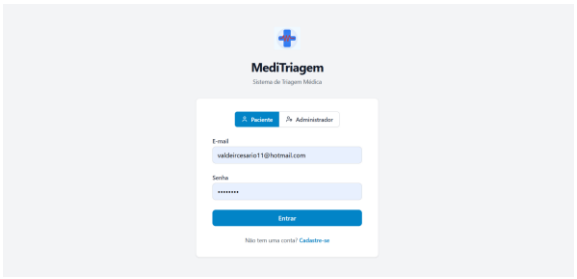
Prisma Studio (GUI para gerenciamento do banco de dados)

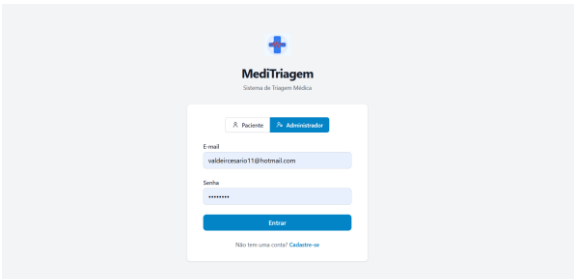
npx prisma studio
Acesse em: http://localhost:5555

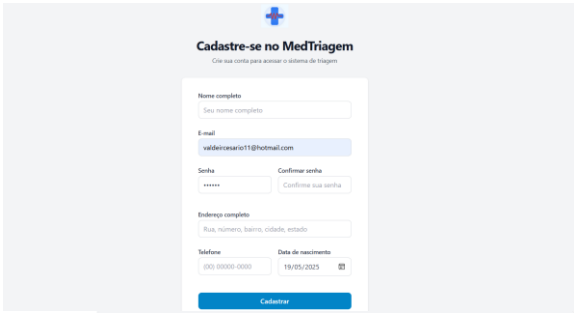
13. Demonstração

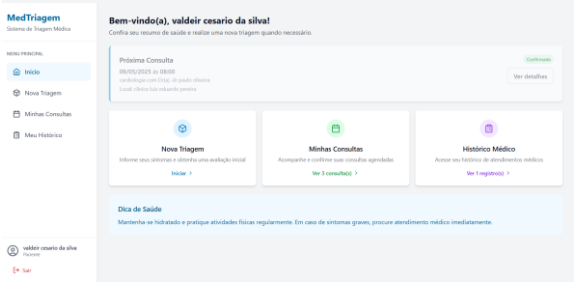
Principais Telas

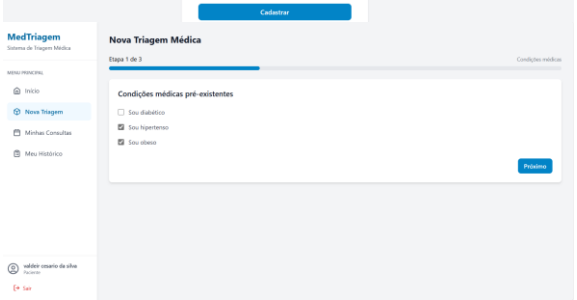
Visualização das Telas

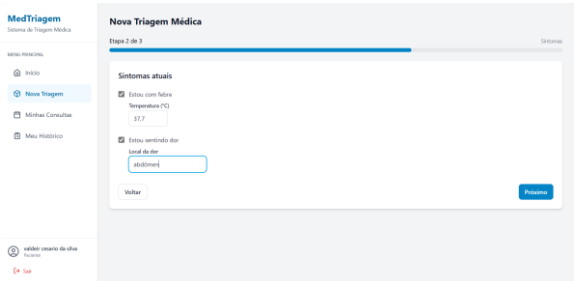


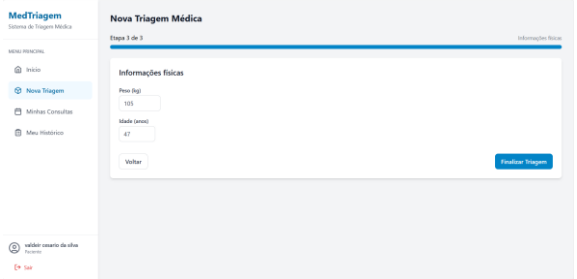


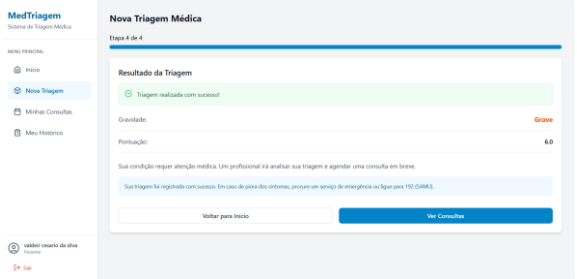


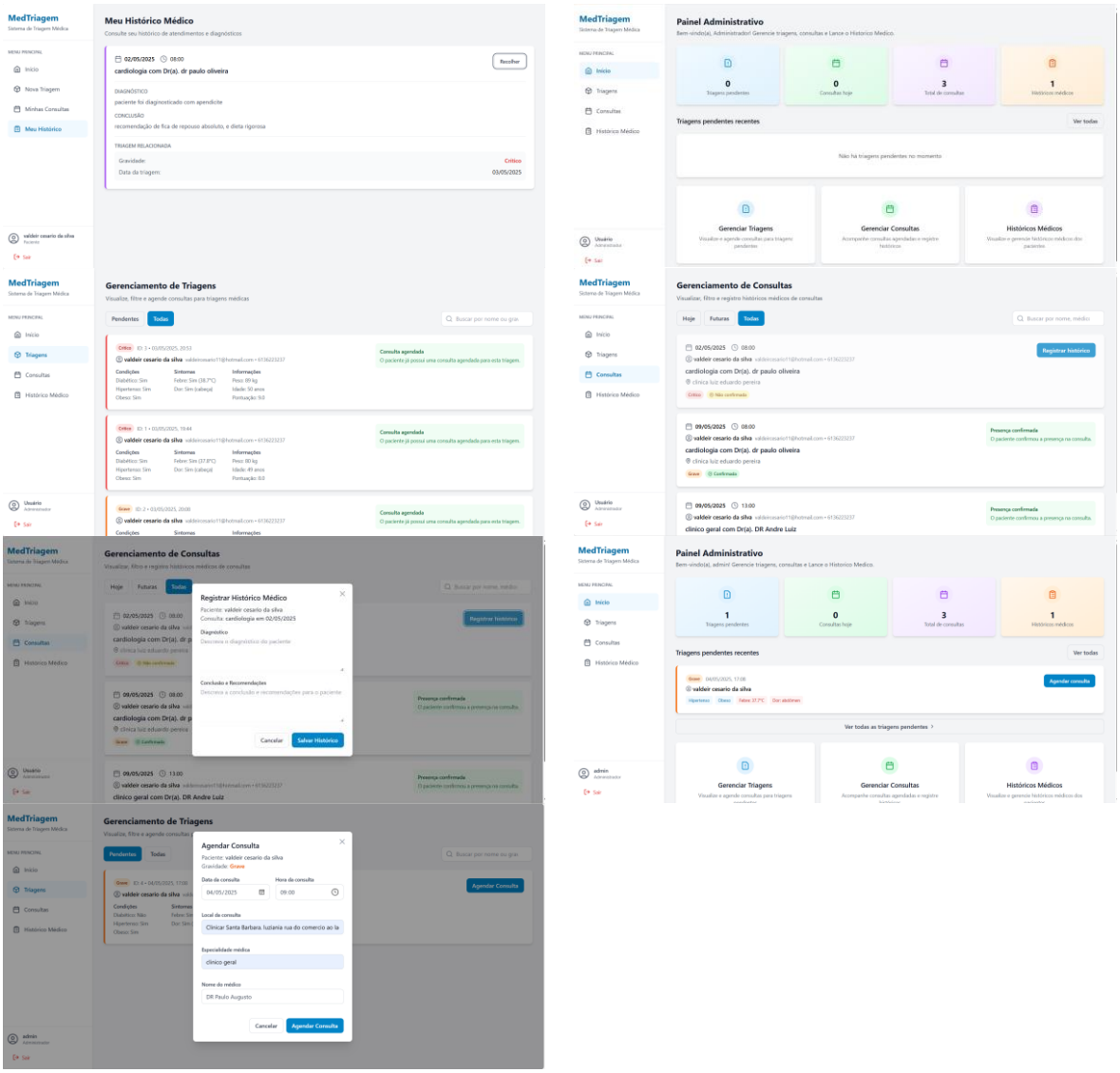












Fluxo de Utilização

1. Login no Sistema
2. Preenchimento da Triagem
    a. Informação de sintomas e condições de saúde
    b. Cálculo automático da gravidade
3. Visualização de Consultas Agendadas
4. Acesso ao Histórico Médico

14. Contribuição

Contribuições são sempre bem-vindas! Para contribuir:

1. Faça um Fork do projeto
2. Crie uma branch para sua feature (git checkout -b feature/nova-funcionalidade)
3. Faça commit das alterações (git commit -m 'feat: Adiciona nova funcionalidade')
4. Faça push para a branch (git push origin feature/nova-funcionalidade)
5. Abra um Pull Request

15. Desenvolvedor

Valdeir Cesario

- LinkedIn: valdeircesario2023
- GitHub: valdeircesario
- Email: valdeircesario11@hotmail.com