

Documentação do projeto interdisciplinar

Autores: Kauã Neres Moura, Eric Silva Patricio, Francisco Osmar Santos Silva.

1. Tema

Em um ambiente de pronto socorro, a demanda de atendimento varia significativamente, dependendo da gravidade dos casos. Muitos pacientes chegam simultaneamente, exigindo uma triagem rápida e precisa para determinar a ordem de atendimento. Isso é fundamental para garantir que aqueles com maior risco de vida recebam tratamento imediato, enquanto casos menos urgentes aguardem em fila de espera. No cenário real dos pronto-socorros, a eficiência na gestão desses fluxos pode fazer a diferença entre a vida e a morte.

1.1 Objetivo do Projeto

O projeto interdisciplinar tem como objetivo dividir os alunos do TSC (Curso de tecnologia em Sistemas de Computação), da UESPI, em grupos. Estes deverão desenvolver um sistema que seja capaz de gerenciar o fluxo de atendimento dos pacientes em um pronto socorro. Esse sistema precisa realizar a triagem dos pacientes com base em critérios de prioridade, seguindo as diretrizes estabelecidas pelo Ministério da Saúde brasileiro. Dessa forma, o sistema deve auxiliar no registro da identificação e classificação do risco de cada paciente, garantindo que os casos mais críticos sejam atendidos primeiro, mesmo em uma situação de alta demanda.

1.2 Delimitação do problema

O projeto visa construir um sistema de fila de prioridade que simule um sistema de um pronto socorro e que mantenha o fluxo de atendimento de maneira eficiente e abrangente, disponibilizando as devidas funções para os usuários do sistema, sendo elas: triagem de paciente baseada em prioridades, processos de operação do sistema, interface e usabilidade usando todos os conhecimentos adquiridos nas matérias envolvidas (programação orientada a objetos, estrutura de dados e banco de dados).

1.3 Justificativa da Escolha do Tema

Colocar em prática os conhecimentos passados em sala de aula através de um tema de grande importância e necessidade, para a avaliação do grupo e sua solução, que ocorre de acordo com os critérios passados.

1.4 Método de Trabalho

O tipo de processo de desenvolvimento de software escolhido foi o Modelo Incremental, por conta do tempo de intervalo entre as entregas parciais 1, 2 e 3.

A modelagem de software especificada nas regras de negócio foi o modelo de Programação Orientada a Objetos, por conta da disciplina ministrada no presente momento, e a linguagem especificada foi Typescript.

1.5 Organização do Trabalho

Os 3 (três) membros do grupo dividiram os assuntos das disciplinas componentes do projeto de acordo com as competências e facilidades de entendimento de cada um, porém também auxiliaram-se entre eles para um completo entendimento do sistema e suas funcionalidades. A organização ficou da seguinte forma:

- Eric Silva Patrício (Programação Orientada a Objetos - Prof. Dr. Eyder Sousa Rios)
- Kauã Neres Moura (Estrutura de Dados - Prof. Dr. Dario Brito Calçada)
- Francisco Osmar Santos Silva (Banco de Dados - Prof. Me. Francisco das Chagas Rocha)

2. Descrição Geral do Sistema

O sistema desenvolvido se trata de um gerenciador de fila de prioridade de um pronto socorro.

- Cadastro de pacientes pela recepção
- Atribuição de triagem ao atendimento pelo enfermeiro
- Fila de prioridade de atendimento com base no risco da triagem atribuída
- Registro completo de atendimentos
- Integração com banco de dados SQLite (Local)
- Interface gráfica para uso em tempo real
- API REST para comunicação com o front-end
- Autenticação de usuários por cargo

2.1 Principais Envolvidos e Suas Características

2.1.1. Usuários do Sistema

O sistema será utilizado por 4 usuários principais:

- Um administrador, que fará o cadastro de médicos(as), enfermeiros(as) e recepcionistas
- Um(a) médico(a), que fará a chamada e retirada de pacientes da fila, e consultará os pacientes.
- Um(a) enfermeiro(a), que fará a chamada para a triagem dos pacientes que tiverem sido atendidos, e também fará a inserção de pacientes triados na fila de espera.
- Um(a) recepcionista que fará o cadastro de pacientes e os inserirá no banco de dados do sistema.

2.1.2. Desenvolvedores do Sistema

- Eric Silva Patrício (Graduando em Sistemas de Computação e programador fullstack).
- Kauã Neres Moura (Graduando em Sistemas de Computação, programador fullstack e designer UI/UX).
- Francisco Osmar Santos Silva (Graduando em Sistemas de Computação e programador front-end).

2.3 Regras de Negócio

- Desenvolvimento em linguagem Typescript.
- Utilizar PostgreSQL como Gerenciador de Banco de Dados *(Posteriormente essa regra de negócio foi modificada, e foi permitido usar outros SGBD's).
- Utilizar dos princípios de UI/UX para melhor performance de usabilidade.

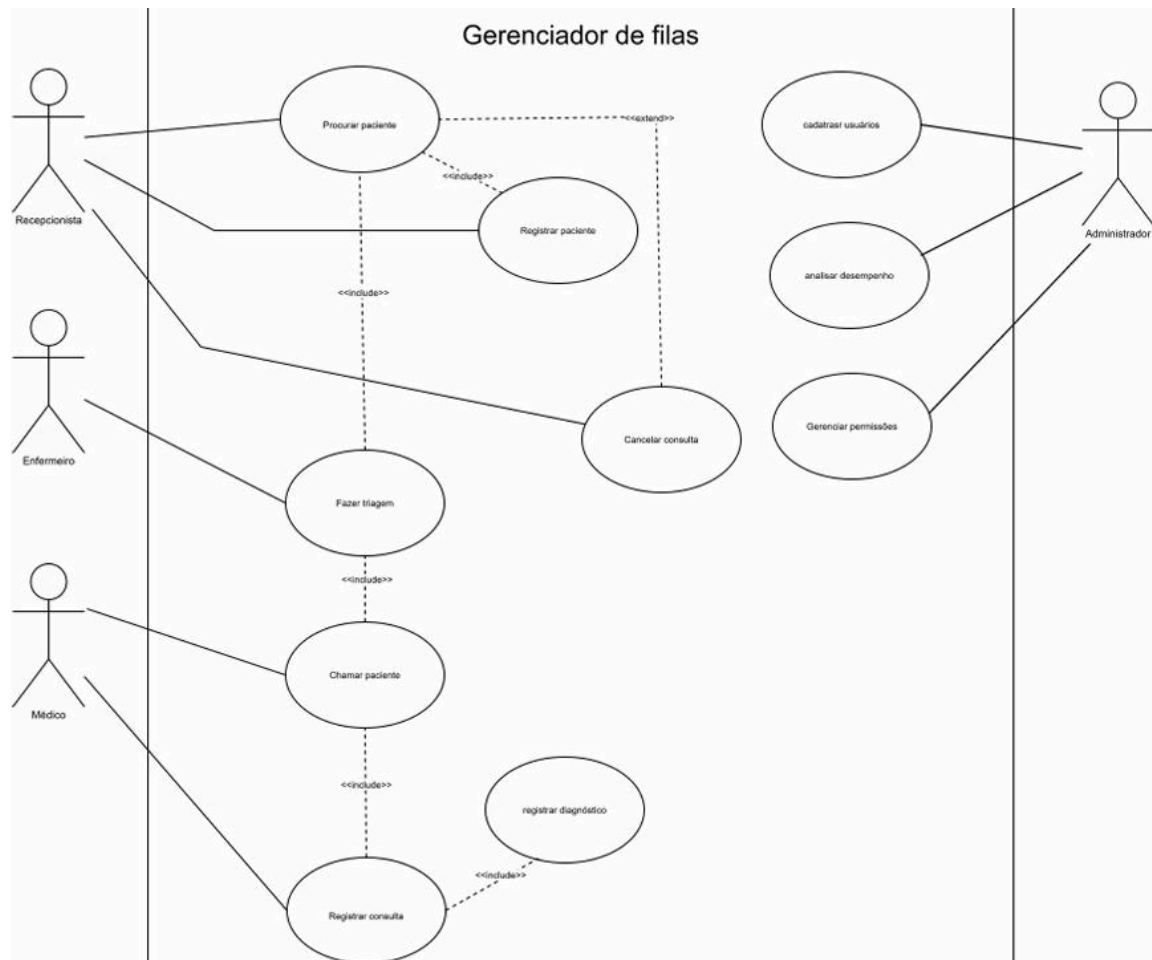
3. Métricas e Cronograma

O Projeto Interdisciplinar foi planejado para ser entregue completo no fim do semestre 2025.1, tendo 3 (três) entregas ao todo: 2 (duas) entregas parciais e 1 (uma) entrega final, e também a documentação completa do sistema.O cronograma ficou definido da seguinte forma:

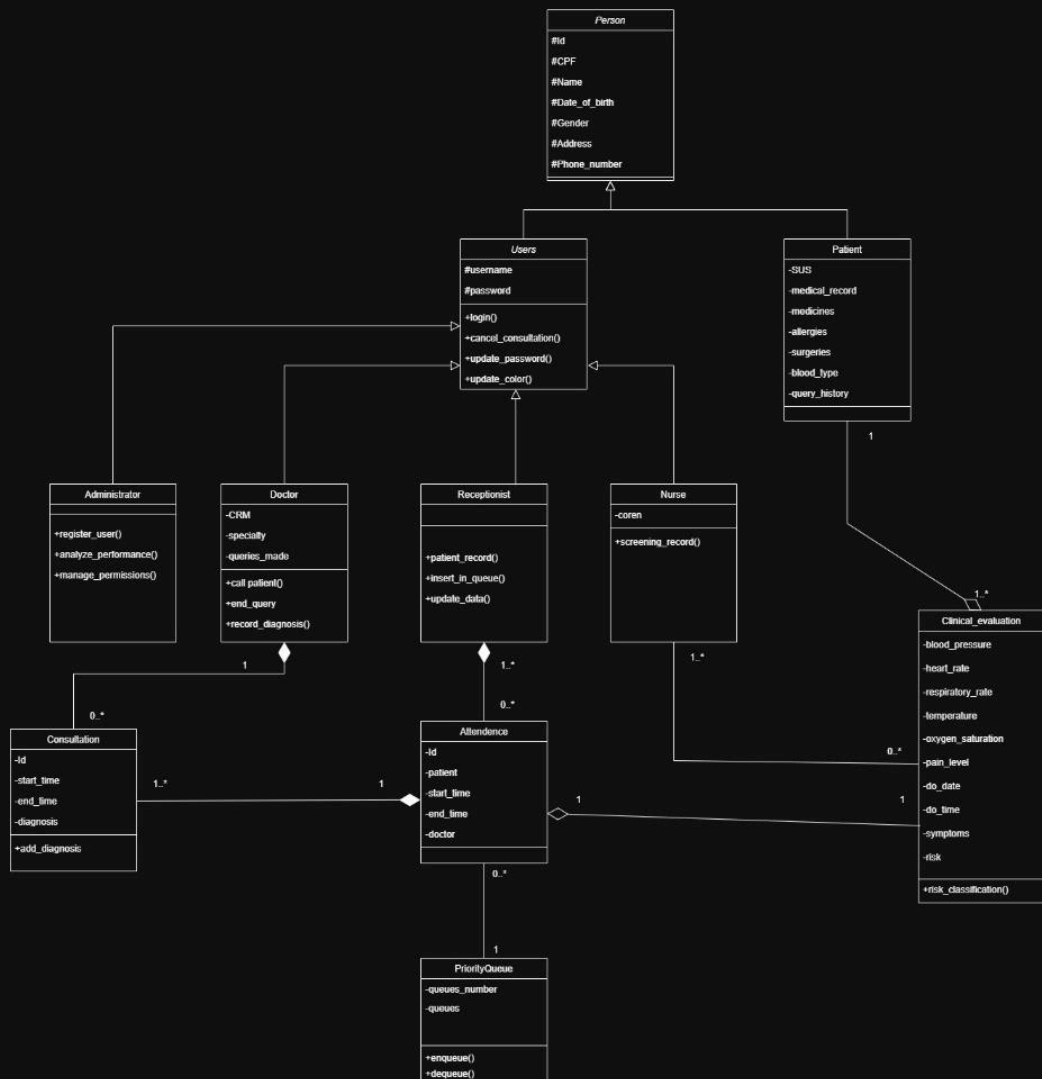
- Entregas Parciais 01: 25/04/2025
- Entregas Parciais 02: 23/05/2025
- Entrega Final (Apresentações e documentação) : 27/06/2025

4. Diagramas

4.1 Diagrama de Casos de Uso

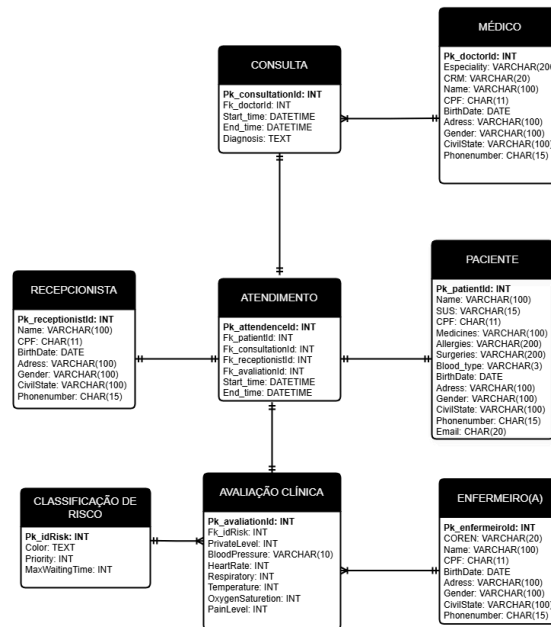


4.2 Diagramas de Classes



5. Modelo de Dados

5.1 Modelo Lógico da Base de Dados



5.2 Criação Física do Modelo de Dados

https://drive.google.com/file/d/1GfGj_fXZbTUfjsqckZI7jVoymjY9RdTp/view

5.3 Ambiente de Desenvolvimento

A linguagem utilizada na lógica do sistema foi Typescript, obedecendo assim uma das regras de negócio especificadas. O Sistema Gerenciador de Banco de Dados (SGBD) foi o SQLite, o qual é um SGBD relacional, leve e embarcado. Também foi utilizado o Next.js como framework para a criação de páginas web interligadas com o Banco de Dados.

6. Implementação

<https://github.com/ericSilvaP/filaProntoSocorro>

7. Testes

Critérios Avaliados	Código de comparação	Código do time
Linguagem	Java	Typescript
Estrutura de dados	Heap	Fila
Critérios de prioridade	Severidade de condição e idade	Prioridade médica e tempo de espera
Memória RAM máxima	46.884 KB	227.384 KB
Porcentagem (uso) da CPU	146%	78%
Tempo de usuário	0.38 s	7.64 s
Tempo de sistema	0.07 s	0.50 s
Tempo total decorrido	0.31 s	10.37 s

7.1 Relatório de Testes

Para a realização dos testes, foi utilizado um repositório de outro desenvolvedor como base de comparação, com o objetivo de avaliar o desempenho entre o sistema proposto e o sistema de referência. Foram executadas 10.000 operações de inserção e 15.000 operações de remoção em ambos os sistemas.

A mensuração do tempo de execução foi feita por meio do utilitário `/usr/bin/time`, disponível no ambiente Linux. Após a execução dos testes, foram coletadas e analisadas as métricas mais relevantes como: memória RAM máxima, porcentagem de uso da CPU, tempo de usuário e tempo de sistema percorrido, a fim de identificar possíveis ganhos ou perdas de desempenho.

Link do repositório utilizado para comparação:
<https://github.com/shreyanimbali/Patient-Triage-System>

8. Relatório Final

Desenvolvimento do Projeto

Após a definição dos requisitos a serem implementados e a elaboração dos diagramas correspondentes, iniciou-se a fase de desenvolvimento do projeto físico. Essa etapa teve duração aproximada de três semanas e foi caracterizada por ajustes recorrentes nos requisitos e fluxos inicialmente propostos, com maior ênfase na modelagem do banco de dados e na implementação da lógica do sistema.

A equipe de desenvolvimento foi dividida em funções específicas para otimizar o trabalho:

- Osmar: responsável pela elaboração e criação do banco de dados, utilizando o sistema gerenciador SQLite;
- Eric: responsável pelo desenvolvimento das camadas de front-end e back-end, utilizando o framework Next.js e a biblioteca sqlite3 para construção das APIs;
- Kauã: responsável pelo desenvolvimento do back-end, também utilizando Next.js e sqlite3.

Ao final do período de desenvolvimento, foi possível atender aos requisitos funcionais estabelecidos inicialmente, mesmo diante dos desafios enfrentados durante o processo. Essa etapa consolidou a base técnica do projeto, garantindo o funcionamento dos principais módulos previstos.