Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS)

Aluno:

**SEU NOME COMPLETO AQUI**

RU:

**SEU RU AQUI**

Curso:

**NOME DO CURSO**

Polo:

**NOME DO POLO**

Professor:

**NOME DO PROFESSOR**

# Sumário

1. Introdução

2. Requisitos Funcionais e Não Funcionais

3. Modelagem

4. Implementação

5. Plano de Testes

6. Conclusão

7. Referências

# 1. Introdução

Este projeto tem como objetivo desenvolver um sistema de gestão hospitalar e de serviços de saúde (SGHSS), com foco na trilha de Back-end. A proposta visa gerenciar dados de pacientes, consultas e usuários, garantindo segurança, integridade dos dados e autenticação via JWT.

O projeto foi desenvolvido em Node.js com Express, utilizando PostgreSQL como banco de dados relacional, com autenticação baseada em tokens JWT.

# 2. Requisitos Funcionais e Não Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Descrição | Tipo | Prioridade |
| RF001 | Cadastrar pacientes | Funcional | Alta |
| RF002 | Listar consultas | Funcional | Alta |
| RF003 | Autenticar usuários via JWT | Funcional | Alta |
| RF004 | Cadastrar consultas com vínculo ao paciente | Funcional | Alta |
| RNF001 | Segurança com autenticação JWT | Não Funcional | Alta |
| RNF002 | Persistência em banco de dados relacional | Não Funcional | Alta |

# 3. Modelagem

O modelo relacional utilizado para este projeto é composto por três entidades principais: usuários, pacientes e consultas. O relacionamento entre elas está demonstrado no Diagrama Entidade-Relacionamento (DER) abaixo.

**[INSIRA A IMAGEM DO DER AQUI]**

# 4. Implementação

A aplicação foi construída utilizando as seguintes tecnologias:

- Node.js

- Express.js

- PostgreSQL

- JWT para autenticação

- curl/Postman para testes

Repositório GitHub:

**[INSIRA O LINK DO REPOSITÓRIO AQUI]**

Documentação dos Endpoints:

**[COLAR CONTEÚDO DO ARQUIVO .MD AQUI]**

# 5. Plano de Testes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Teste | Entrada usada | Esperado | Obtido | Status |
| CT01 | Cadastro de usuário | email/senha | 201 Created | OK | ✅ |
| CT02 | Login com JWT | email/senha | 200 OK + token | OK | ✅ |
| CT03 | Criar paciente | nome, cpf, nascimento + token | 201 Created | OK | ✅ |
| CT04 | Criar consulta | data, paciente\_id + token | 201 Created | OK | ✅ |
| CT05 | Acesso sem token | curl sem header Authorization | 401 Unauthorized | OK | ✅ |

# 6. Conclusão

Durante o desenvolvimento deste projeto, foi possível consolidar conhecimentos em criação de APIs RESTful, autenticação com JWT e persistência de dados em banco relacional.

A principal dificuldade enfrentada foi a conexão com banco de dados remoto e o controle de tokens. Futuramente, o sistema pode ser estendido com filtros avançados, integração com front-end e deploy online.

**[ADICIONE QUALQUER OBSERVAÇÃO PESSOAL AQUI]**

# 7. Referências

- Node.js - https://nodejs.org

- Express - https://expressjs.com

- PostgreSQL - https://www.postgresql.org/

- JWT - https://jwt.io/

- DB Diagram - https://dbdiagram.io

**[ADICIONE OUTRAS REFERÊNCIAS UTILIZADAS]**