



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO



DOCUMENTAÇÃO SISTEMA PRONTUÁRIO

ALUNO: RAFAEL NASCIMENTO ANDRADE

MATRÍCULA: 202200014480

Introdução

Esse projeto visa construir um sistema de gerenciamento de prontuário eletrônico para uma clínica de nutrição da UFS de Lagarto. O objetivo é permitir o cadastro dos dados de pacientes e, para cada paciente, permitir o cadastro de N registros de N tipos.

Os tipos de registros informados pelo cliente são:

- Anamnese de 1º Consulta
- Anamnese Infantil
- Anamnese de Retorno
- Plano Alimentar
- Plano de Recordatório

Nos exemplos fornecidos pelo cliente, as informações das anamneses e do paciente estavam juntas, porém, para permitir a possibilidade de cadastrar vários registros para um mesmo paciente, foi criado um formulário de cadastro separado para o paciente.

Os dados para cadastro do paciente são:

- Nome
- Endereço
- Data de Nascimento
- Idade
- Telefone

No repositório do git encontra-se exemplos de anamneses e plano alimentar fornecidos pelo cliente

Link do repositório no GitHub:

<https://github.com/RafaelAndr/prontuarioUFS>

Ferramentas Utilizadas

O projeto será um sistema web que utilizará React + Bootstrap no frontend e python (FastAPI) no backend, o banco de dados utilizado foi o postgresql(Mas poderá mudar caso necessite).

Requisitos Funcionais

RF01: O sistema deverá permitir o cadastro de pacientes.

RF02: O sistema deverá permitir a edição dos dados dos pacientes.

RF03: O sistema deverá permitir a exclusão dos dados de um paciente, removendo automaticamente todos os registros vinculados a ele.

RF05: O sistema deverá permitir a busca dos dados dos pacientes por nome.

RF06: O sistema deverá permitir o cadastro de anamneses de 1º consulta (padrão)

RF07: O sistema deverá permitir a edição da anamnese de 1º consulta

RF08: O sistema deverá permitir o exclusão da anamnese de 1º consulta

RF09: O sistema deverá permitir a visualização dos dados da anamneses de 1º consulta

RF10: O sistema deverá permitir o cadastro de anamnese infantil

RF11: O sistema deverá permitir a edição de anamnese infantil

RF12: O sistema deverá permitir o exclusão de anamnese infantil

RF13: O sistema deverá permitir a visualização dos dados da anamnese infantil

RF14: O sistema deverá permitir o cadastro de anamnese de retorno

RF15: O sistema deverá permitir a edição da anamnese de retorno

RF16: O sistema deverá permitir o exclusão da anamnese de retorno

RF17: O sistema deverá permitir a visualização dos dados da anamnese de retorno

RF18: O sistema deverá permitir o cadastro de plano alimentar

RF19: O sistema deverá permitir a edição do plano alimentar

RF20: O sistema deverá permitir o exclusão do plano alimentar

RF21: O sistema deverá permitir a visualização dos dados do plano alimentar

RF22: O sistema deverá permitir o download do plano alimentar em formato PDF

RF23: O sistema deverá permitir o cadastro de ficha de recordatório alimentar

RF24: O sistema deverá permitir a edição da ficha de recordatório alimentar

RF25: O sistema deverá permitir o exclusão da ficha de recordatório alimentar

RF26: O sistema deverá permitir a visualização dos dados do recordatório alimentar

Tutorial de Como executar o sistema

1. Backend (FastAPI)

Abra o terminal na pasta do projeto e execute:

```
cd backend
```

```
poetry install
```

```
poetry run uvicorn src.main:app --port 8080 --reload
```

(Pode mudar a porta no arquivo env no frontend se quiser)

Isso irá:

- **criar o ambiente virtual do Poetry**
- **instalar todas as dependências**
- **iniciar o servidor FastAPI em <http://localhost:8080>**

2. Frontend (React)

Em outro terminal, execute:

```
cd frontend
```

```
npm install
```

```
npm run dev
```

O React iniciará normalmente (geralmente em <http://localhost:5173>).

2. Banco de Dados (Postgres)

É necessário instalar o postgresql, configurar o usuário e senha, e criar uma database, em seguida colocar no arquivo .env do backend, exemplo:

```
DATABASE_URL=postgresql://postgreuser:1234@localhost:5432/prontuarioufs
```

Pendências Atuais do Projeto – Sistema de Prontuário UFS

No estágio atual de desenvolvimento, o projeto do sistema de prontuário encontra-se funcional e com todos os módulos principais implementados. As etapas de modelagem, construção do backend, desenvolvimento do frontend e integração das funcionalidades essenciais já foram concluídas.

Entretanto, ainda existem algumas pendências que precisam ser finalizadas para que o sistema possa ser utilizado em ambiente real pelo cliente. São elas:

1. Implantação do sistema no ambiente do cliente

É necessário realizar o processo de implantação — configurar o servidor, banco de dados e demais ajustes necessários para disponibilizar a aplicação. É necessário decidir como a aplicação vai ser disponibilizada para o cliente e como ele irá acessá-la da forma mais fácil, seja por meio da web ou um aplicativo desktop criado a partir de frameworks que criam aplicativos desktops a partir de aplicações web como o Electron.

2. Realização de testes completos do sistema antes da implantação

Antes de disponibilizar o sistema ao cliente, é indispensável executar uma bateria de testes em massa, abrangendo todos os módulos e fluxos de uso. Esses testes têm como objetivo validar cada funcionalidade, identificar possíveis falhas, verificar o comportamento da aplicação em cenários variados e assegurar que todas as funcionalidades estejam operando corretamente.

3. Correção de erros identificados durante os testes

Durante a execução dos testes, erros ou comportamentos inesperados poderão ser identificados. Será necessário realizar a correção de bugs, ajustes de interface, melhorias de usabilidade e refinamentos no backend ou frontend para garantir o funcionamento pleno e consistente do sistema.

3. Ajustes e melhorias conforme feedback da cliente

Após o início do uso real, será necessário recolher o feedback da cliente para realizar ajustes finos, otimizações e possíveis novas funcionalidades que agreguem valor ao sistema. Essa etapa garante que a solução se alinhe totalmente às necessidades práticas do consultório/usuário final.