

MyHealth

12 de junho de 2025

VISÃO GERAL

Este documento apresenta a proposta de desenvolvimento de um sistema para gestão de planos de cuidados em saúde, que permitirá o acompanhamento de pacientes com diferentes tipos de planos ativos simultaneamente.

O projeto está em fase de desenvolvimento e tem como objetivo estruturar uma aplicação capaz de gerenciar linhas de cuidado — que são planos contínuos destinados à prevenção ou ao acompanhamento de condições crônicas — e programas de saúde, que são planos pontuais com duração e objetivos específicos.

A aplicação será construída utilizando Next.js, Prisma ORM e Chakra UI, adotando uma arquitetura clara, modular e escalável, que facilite a manutenção, evolução e entendimento do sistema. A solução terá foco na gestão de membros, planos e eventos, bem como no controle de sobreposição de planos e na identificação de possíveis conflitos de agenda.

Este documento acompanhará o andamento do desenvolvimento, refletindo as decisões arquitetônicas, técnicas e eventuais ajustes realizados durante a implementação.

OBJETIVO

O objetivo deste projeto é desenvolver uma aplicação que permita a gestão eficiente de planos de cuidados em saúde, oferecendo às equipes médicas e de enfermagem uma ferramenta capaz de acompanhar e administrar a jornada dos pacientes de forma organizada, segura e rastreável.

A solução deverá possibilitar o gerenciamento de linhas de cuidado e programas de saúde, garantindo que as regras de negócio sejam aplicadas corretamente, que haja controle sobre sobreposição de planos e conflitos de eventos, e que todas as alterações e decisões sejam devidamente registradas no sistema.

Além disso, busca-se construir uma arquitetura escalável, clara e de fácil manutenção, que permita tanto a expansão futura de funcionalidades quanto a sustentação de processos operacionais críticos no contexto de gestão de saúde.

ESPECIFICAÇÕES TÉCNICAS

A aplicação será desenvolvida como um **web app fullstack**, utilizando a stack **Next.js**, **Prisma ORM** e **Chakra UI**, com o objetivo de garantir uma solução moderna, escalável e de fácil manutenção.

Área	Tecnologia
Framework	Next.js (Fullstack React)
UI	Chakra UI
ORM	PRisma
Banco de Dados	SQLite (dev) / PostgreSQL (prod)
Linguagem	TypeScript
Testes	Jest + Testing Library (opcional)
Controle de versão	Git + GitHub
API	API Rest + Server Actions

CRONOGRAMA

O desenvolvimento do projeto foi organizado em etapas, priorizando a construção dos elementos fundamentais do sistema, seguidos pela implementação das regras de negócio, funcionalidades complementares e ajustes finais.

- 1. Setup e Estrutura Inicial**
 - Configuração do projeto com Next.js e Chakra UI.
 - Configuração de ferramentas auxiliares: ESLint, Editorconfig, TypeScript.
- 2. Inicialização do banco de dados**
 - Configuração do PrismaJS.
 - Configuração do banco de dados (SQLite para desenvolvimento).
- 3. Modelagem de Dados**
 - Definição das entidades principais: Patient, Plan, Event.
 - Criação do schema no Prisma.
 - Validação inicial da modelagem, com uma seed.
- 4. Desenvolvimento das Telas Básicas**
 - CRUD de pacientes Patient.

-
- CRUD de planos Plan com suporte à criação, substituição e remoção.
 - CRUD de eventos Event com suporte à criação, edição e exclusão.

5. Desenvolvimento das Regras de Negócio

- Aplicação da regra de limite de 1 linha de cuidado ativa por membro.
- Aplicação da regra de limite de até 3 programas ativos.
- Implementação da lógica de substituição de linhas de cuidado.
- Implementação do encerramento automático de programas após conclusão dos eventos.

6. Detecção de Sobreposição e Conflitos

- Implementação da lógica para detecção de conflitos entre eventos (ex.: eventos na mesma data ou médicos em datas muito próximas).
- Sinalização de conflitos na interface ou retorno pela API.
- Gatilho de notificação para a enfermeira responsável sempre que houver sobreposição de planos.

7. Rastreabilidade e Histórico

- Implementação de logs e registros de alterações em planos e eventos.
- Criação de um histórico que permita visualizar mudanças realizadas no cadastro de planos e eventos.

8. Ajustes, Refinamentos e Testes

- Refinamento da interface e experiência do usuário.
- Validação das regras de negócio aplicadas.
- Implementação de testes unitários e, se possível, testes integrados.
- Correções de bugs e melhorias gerais.