HospiCast – Previsão Inteligente de Demandas Hospitalares

Nome do Estudante: Mateus de Faria da Silva

Curso: Engenharia de Software

Data de Entrega: 14/05/2025

Resumo

Este documento apresenta o projeto "HospiCast", uma plataforma que aplica Inteligência Artificial para prever a demanda de atendimentos em unidades hospitalares. A solução visa auxiliar a gestão hospitalar por meio da antecipação de cenários críticos e otimização de recursos.

1. Introdução

Contexto

O sistema de saúde frequentemente enfrenta desafios como superlotação e escassez de recursos. A previsão da demanda por atendimentos é uma necessidade estratégica para melhorar o planejamento hospitalar.

Justificativa

Com base na análise de séries temporais e fatores externos, como clima e sazonalidade, o HospiCast busca aplicar IA de forma prática e acessível, agregando valor ao campo da Engenharia de Software aplicada à saúde.

Objetivos

- Desenvolver uma plataforma para prever demandas hospitalares com IA.
- Exibir os resultados via dashboards interativos.
- Disponibilizar uma API REST para integração com outros sistemas.

2. Descrição do Projeto

Tema do Projeto

Desenvolvimento de uma aplicação web para previsão de atendimentos hospitalares, com base em dados históricos e fatores externos, como clima e calendário.

Problemas a Resolver

- Falta de previsão sobre a demanda hospitalar.
- Dificuldade no planejamento de recursos e pessoal.
- Tomada de decisão reativa em vez de preventiva.

Limitações

- O projeto não abrange previsões de patologias específicas.
- Não inclui integração direta com sistemas hospitalares reais nesta fase inicial.

3. Especificação Técnica

3.1 Requisitos de Software

Requisitos Funcionais

- RF01 - O sistema deve permitir o cadastro de hospitais com localização geográfica.
- RF02 - O sistema deve importar dados históricos de atendimentos hospitalares.
- RF03 - O sistema deve obter dados climáticos automaticamente.
- RF04 - O sistema deve processar os dados históricos e climáticos para gerar previsões.
- RF05 - O sistema deve exibir previsões de atendimentos futuros em um dashboard.
- RF06 - O sistema deve permitir visualização de dados em diferentes períodos (dia, semana, mês).
- RF07 - O sistema deve expor uma API REST para consumo externo das previsões.
- RF08 - O sistema deve permitir que administradores exportem relatórios em PDF.
- RF09 - O sistema deve possuir autenticação para acesso ao painel de gestão.

Requisitos Não-Funcionais

- RNF01 - O sistema deve responder às requisições da API REST em até 1 segundo.
- RNF02 - O sistema deve garantir a integridade dos dados recebidos.
- RNF03 - O sistema deve estar disponível 24/7, com tolerância a falhas.
- RNF04 - O sistema deve utilizar HTTPS para comunicação segura.
- RNF05 - O frontend deve ser responsivo e acessível em dispositivos móveis.

Diagrama de Casos de Uso: ver Apêndice A.

3.2 Considerações de Design

- A arquitetura segue o padrão MVC com separação entre frontend, backend e serviço de IA.
- O sistema é composto por módulos independentes: visualização, previsão, API e banco de dados.

Visão Inicial da Arquitetura

Frontend: WebApp em Blazor/React.
Backend: API REST em FastAPI/ASP.NET Core.
Modelo de IA: Prophet (Python).
Banco de Dados: PostgreSQL.

Padrões de Arquitetura

MVC (Model-View-Controller)
Microserviços (separando API e módulo de previsão)

Modelos C4

[A ser adicionado com diagramas de Contexto, Contêiner, Componente e Código]

3.3 Stack Tecnológica

- Linguagens: Python, C#, JavaScript
- Frameworks: FastAPI, Blazor/React, Prophet
- Banco de Dados: PostgreSQL
- Ferramentas: Docker, GitHub, Grafana, Swagger, Railway/Render

3.4 Considerações de Segurança

- Validação de entrada nas APIs
- Rate limiting
- HTTPS obrigatório
- Autenticação básica para API

4. Próximos Passos

Etapa	Status
Coleta e tratamento de dados	A iniciar
Treinamento do modelo	A iniciar
Desenvolvimento da API	Planejado
Desenvolvimento da interface	Planejado
Testes e deploy	Planejado

5. Referências

- https://facebook.github.io/prophet/
- https://fastapi.tiangolo.com/

- https://docs.microsoft.com/aspnet/core/blazor/
- https://pt.stackoverflow.com
- https://draw.io

6. Apêndices

Apêndice A – Diagrama de Casos de Uso

Diagrama ilustrativo: Casos de uso do sistema para usuário gestor e administrador.

7. Avaliações de Professores

Considerações Professor(a):

Considerações Professor(a):

Considerações Professor(a):
