Documentação do Projeto: Sistema de Criação de Lembretes Full-Stack

Abstract

This document describes the development of the full-stack reminder creation system, covering the design decisions, assumptions made, and instructions for running the system. The system allows the creation and deletion of reminders, where each reminder must contain a name and a future date. The system is implemented with a frontend in React and a backend in C# with ASP.NET.

Contents

1 Introdução

O Sistema de Criação de Lembretes Full-Stack é uma aplicação web que permite aos usuários criar, excluir e visualizar lembretes com uma data futura. Cada lembrete é composto por um nome e uma data, com a exigência de que a data seja sempre no futuro. O sistema exibe os lembretes de forma cronológica e organiza os lembretes por dia, criando novos dias conforme necessário.

2 Arquitetura do Sistema

O sistema é dividido em duas partes principais:

- **Frontend:** Implementado em React com Vite, utilizando a biblioteca `styled-components` para estilização, e hooks como `useState` e `useEffect` para gerenciamento de estado e ciclo de vida dos componentes.
- **Backend:** Implementado em C# com ASP.NET, utilizando a arquitetura MVC (Model-View-Controller) para organizar a estrutura do código.

3 Premissas Assumidas

- A data de cada lembrete deve ser sempre no futuro. Lembretes com datas no passado não são permitidos.
- O sistema deve exibir os lembretes em ordem cronológica.
- Cada dia deve conter uma lista de lembretes para aquela data específica.
- O frontend e o backend estão separados e se comunicam através de APIs REST.
- O sistema deve ser simples de configurar e executar, com pouca necessidade de configuração manual.

4 Decisões de Projeto

4.1 Frontend

- O Vite foi escolhido como bundler por sua rapidez no processo de construção e recarga durante o desenvolvimento.

- A biblioteca `styled-components` foi usada para estilização, oferecendo uma solução modular e escalável para o gerenciamento de estilos.
- O uso de hooks como `useState` e `useEffect` foi adotado para o gerenciamento do estado e efeitos colaterais de componentes.

4.2 Backend

- O ASP.NET foi escolhido para o backend por ser uma framework robusta e amplamente utilizada para criar APIs RESTful.
- A arquitetura MVC foi utilizada para garantir a separação de responsabilidades, tornando o código mais organizado e fácil de manter.
- A validação de dados, como a verificação de datas no futuro, foi implementada tanto no frontend quanto no backend para garantir a integridade dos dados.

5 Rotas do Backend

5.1 Criar Lembrete

- **Rota:** POST /api/reminder
- **Descrição:** Cria um novo lembrete, verificando se a data fornecida é válida (no futuro).
- **Respostas Possíveis:**
  - **200 OK:** Lembrete criado com sucesso.
  - **400 BadRequest:** Dados inválidos ou data no passado.

5.2 Deletar Lembrete

- **Rota:** DELETE /api/reminder
- **Descrição:** Deleta um lembrete especificado pelo nome e data, verificando se a data não passou.
- **Respostas Possíveis:**
  - **200 OK:** Lembrete removido com sucesso.
  - **400 BadRequest:** Data no passado ou dados inválidos.
  - **404 NotFound:** Lembrete não encontrado.

5.3 Obter Todos os Lembretes

- **Rota:** GET /api/reminder
- **Descrição:** Retorna todos os lembretes cadastrados no sistema.
- **Respostas Possíveis:**
  - **200 OK:** Lista de lembretes.

5.4 Testes

- O Vitest e a React Testing Library foram usados no frontend para garantir que os componentes funcionem corretamente e atendam aos requisitos de usabilidade.
- O xUnit e Moq foram usados no backend para garantir que os métodos de criação, exclusão e recuperação de lembretes funcionem conforme esperado, e para simular dependências durante os testes.

6 Instruções para Executar o Sistema

1. Clone o repositório.

6.1 Frontend

Para executar o frontend, siga as etapas abaixo:

1. Navegue até o diretório `frontend/reminderwebapp`.
2. Instale as dependências com o comando:

   ```
   npm install
   ```

3. Execute o servidor de desenvolvimento com o comando:

   ```
   npm run dev
   ```

4. O frontend estará disponível em `http://localhost:5173`.
5. Para rodar os testes:

   ```
   npm test
   ```

6.2 Backend

Para executar o backend, siga as etapas abaixo:

1. Navegue até o diretório `backend/ReminderWebAPI`.
2. Restaure as dependências com o comando:

   ```
   dotnet restore
   ```

3. Compile o projeto com o comando:

   ```
   dotnet build
   ```

4. Execute a aplicação com o comando:

   ```
   dotnet run
   ```

5. O backend estará disponível em `https://localhost:7108`.
6. Para rodar os testes:

   ```
   dotnet test
   ```

7 Conclusão

O sistema de criação de lembretes foi desenvolvido com foco na simplicidade e na usabilidade, garantindo que os lembretes sejam armazenados corretamente e exibidos de forma cronológica. O uso de React e ASP.NET proporcionou uma arquitetura robusta e escalável, enquanto os testes unitários ajudaram a garantir a confiabilidade do sistema.