

Documentação Técnica

1. Título e Autoria

- Nome do projeto: **Bookside**
- Nome do desenvolvedor: **Sarah Barcaro de Freitas**
- Data: **20/09/2024**

2. Resumo Executivo Este projeto é um portal de resenhas de livros que permite aos usuários fornecer avaliações sobre livros. O objetivo é proporcionar uma plataforma colaborativa, onde leitores possam inserir suas experiências literárias e administrar sua própria biblioteca de livros, podendo controlar em que etapa de leitura está ou pretende iniciar.

3. Introdução

O portal de resenhas foi desenvolvido para resolver a necessidade de um espaço centralizado onde os leitores possam inserir opiniões, resenhas e receber recomendações sobre livros antes de fazer uma compra ou escolha. Ao permitir que leitores deixem suas avaliações, o portal permite o usuário relembrar alguma opinião sobre certo livro ou declarar algum livro como “favorito” ou “quero ler” por exemplo, podendo administrar suas leituras. O projeto é voltado para amantes da literatura que desejam ter um lugar seguro onde documentar suas opiniões e sentimentos sobre certo livro. O sistema foi desenvolvido como uma aplicação web utilizando tecnologias modernas para garantir uma experiência rápida e eficiente para o usuário.

4. Requisitos

- **Requisitos Funcionais:**

1. Os usuários devem poder visualizar uma lista de livros e suas respectivas resenhas.
2. O sistema deve permitir a avaliação dos livros pelos usuários.
3. Deve ser possível buscar livros por título.
4. O sistema deve listar as resenhas e avaliações de cada livro.
5. Os usuários devem poder adicionar, editar ou remover resenhas de livros.

- **Requisitos Não Funcionais:**

1.O sistema deve ser responsivo e funcionar em dispositivos móveis.

5. Tecnologias Utilizadas

- **Frontend:**

- React
- Tailwind CSS para estilização
- Vite para desenvolvimento e build otimizado

- **Backend:**

- Node.js com Express.js
- MongoDB como banco de dados

- **Outras Ferramentas:**

- Git para controle de versão
- NPM para gerenciamento de dependências

6. Arquitetura do Sistema

A arquitetura do sistema é baseada em uma aplicação web full-stack, com separação clara entre frontend e backend:

- **Frontend (React com Vite):** Responsável pela interface com o usuário, utilizando componentes React para renderizar as páginas dinâmicas de resenhas e avaliações.
- **Backend (Node.js/Express):** Serve como API que lida com as requisições feitas pelo frontend e se comunica com o banco de dados MongoDB.
- **Banco de Dados (MongoDB):** Utilizado para armazenar informações sobre livros e resenhas. As operações de leitura e gravação são feitas via API.

Diagrama de Arquitetura

[React Frontend] <-> [Express API] <-> [MongoDB]

7. Estrutura do Projeto

- **Backend** (back-end/):
 - `index.js`: Código principal do servidor Express. Define as rotas de API para manipulação de livros e resenhas.
 - `package.json`: Lista dependências e scripts para rodar o servidor.
- **Frontend** (front-end/):
 - `src/`: Contém o código fonte React, incluindo componentes e páginas.
 - **Components**: Componentes reutilizáveis como cabeçalhos e listagens.
 - **Pages**: Páginas dinâmicas para exibição de resenhas, home e busca.
 - `public/`: Arquivos estáticos.
 - `tailwind.config.js`: Configuração do Tailwind CSS.
 - `vite.config.js`: Configuração do Vite para build otimizado.

8. Design e Fluxos de Usuário

- **Home Page**: Exibe uma lista de livros com as avaliações e permite navegação para detalhes individuais.
- **Detalhes do Livro**: Exibe informações detalhadas sobre o livro, incluindo as resenhas e avaliações do usuário.
- **Busca**: Permite ao usuário procurar livros por título.
- **Admin Pannel**: Acesso restrito para gerenciar o conteúdo do portal.

9. API

POST /resenhas: Adiciona uma nova resenha ao banco de dados.

- **Parâmetros de Entrada**: Objeto JSON com `livro_id`, `resenha` e `avaliacao`.

- **Exemplo de Requisição:**

```
json {  
  "livro_id": "123",  
  "resenha": "Ótimo livro!",  
  "avaliacao": 5  
}
```

- **PATCH /resenhas/:id:** Atualiza uma resenha existente.
 - **Parâmetros de Entrada:** ID da resenha a ser atualizada e os novos dados.
- **DELETE /resenhas/:id:** Remove uma resenha com base no ID fornecido.

10. Banco de Dados

O banco de dados MongoDB armazena as informações sobre os livros e as resenhas. Abaixo está a estrutura das coleções principais:

- **Resenhas:**
 - `_id`: ID único da resenha.
 - `livro_id`: ID do livro ao qual a resenha pertence.
 - `conteudo`: Texto da resenha.
- **Livros:**
 - `_id`: ID único do livro.
 - `titulo`: Nome do livro.
 - `autor`: Nome do autor.
 - `genero`: Gênero literário do livro.
 - `nota`: Nota do livro dada pelo usuário

11. Testes

- **Testes Unitários:** As funções principais foram testadas manualmente para garantir que as rotas de API respondam corretamente.

- **Testes de Integração:** Testes manuais foram realizados para garantir que o frontend e o backend estão integrados corretamente, e as informações são exibidas no frontend conforme esperado.

12. Deploy e Configuração

- **Ambiente de Desenvolvimento:**

1. Clonar o repositório.
2. Executar npm install no frontend e backend para instalar as dependências.
3. Executar npm run dev no frontend e npm start no backend para iniciar o servidor de desenvolvimento.

- **Deploy para Produção:**

1. Buildar o frontend com npm run build no diretório front-end.
2. Configurar o servidor backend para servir os arquivos estáticos do frontend, além de fornecer a API.

13. Segurança

- O projeto utiliza medidas básicas de segurança como a sanitização de entradas do usuário e verificação de dados enviados via API. Além disso, há planos para implementar autenticação para usuários e administradores.

14. Problemas Conhecidos e Limitações

- O sistema atualmente não possui autenticação, o que significa que qualquer usuário pode acessar o painel administrativo se souber a URL.
- Não há verificação de permissões para as ações no sistema de resenhas, o que pode ser implementado em versões futuras.

15. Conclusão

O projeto de portal de resenhas de livros foi desenvolvido com o objetivo de proporcionar uma plataforma pessoal para leitores. Futuras melhorias podem incluir autenticação de usuários, sistema de recomendação personalizado e otimização de desempenho para grandes volumes de dados.