Documentação do Projeto Hermes

Membros:
Pedro Carneiro

---

1. Introdução

1.1 Apresentação

O projeto Hermes é uma iniciativa inovadora que visa criar uma plataforma web abrangente e intuitiva para facilitar a descoberta, organização e utilização de software livre. Com uma interface amigável e recursos avançados de pesquisa, o Hermes pretende ser o destino central para indivíduos e organizações que buscam soluções de software de código aberto.

Além de simplesmente localizar software livre, o Hermes se propõe a oferecer uma experiência completa aos usuários, fornecendo informações detalhadas sobre cada aplicativo, incluindo sua funcionalidade, requisitos de sistema, documentação, e até mesmo tutoriais e guias de uso. Essa abordagem holística visa capacitar os usuários a tomarem decisões informadas sobre quais softwares atenderão melhor às suas necessidades específicas.

1.2 Objetivos do Projeto

O projeto Hermes tem como objetivo promover a comunidade de software livre, incentivando colaborações, contribuições e feedbacks. Ao facilitar o acesso ao software livre e promover uma cultura de compartilhamento e transparência, o Hermes contribui para o crescimento e aprimoramento contínuo do ecossistema de código aberto.

Além disso, o Hermes pode incluir recursos adicionais, como integração com plataformas de desenvolvimento colaborativo, fóruns de discussão, e até mesmo funcionalidades de recomendação com base nas preferências e nas necessidades dos usuários.

3. Escopo do Projeto

O escopo deste projeto inclui:

- Desenvolvimento de um aplicativo web responsivo.
- Renderiza aplicativos/softwares que auxiliam no desenvolvimento e qualidade/efetividade de projetos.
- Testes unitários para garantir a robustez e funcionalidade do código.

4. Cronograma

Etapa	Duração Estimada	Data de Início	Data de Conclusão
Planejamento	1 dia	18/03	18/03
Design	2 dias	19/03	20/03
Desenvolvimento	4 semanas	21/03	—
Testes Unitários	2 semanas	27/03	—
Testes Integrados	2 semanas	—	—
Revisão e Lançamento	1 semana	—	—

5. Recursos Necessários

5.1 Equipe de Desenvolvimento

(**Front-End:** Irvin Marques e Bia Siquara,**Q.A:** Victor Alves,**Back-End**:xxxxxx,
Equipe de desenvolvimento: (**Front-End:** Irvin Marques e Bia Siquara,**Q.A:** Victor Alves,**Back-End**:xxxxxx,

5.2 Documentação e Pesquisa

Pedro Carneiro e Rafael Teixeira

6 Requisitos Funcionais

6.1 Ferramentas de desenvolvimento

(**IDEs**: Visual Studio , **Frameworks**: Jest, Testing Library, **Biblioteca:** React, **Linguagem de programação:** JavaScript).

6.2 Ferramentas de versionamento

- **Git (é um sistema de controle de versões distribuído)**
- **GitHub (é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git)**
- **GitLab (é um gerenciador de repositório de software baseado em Git)**

6.3 Ferramentas de gerenciamento de tarefas

- **Jira (é uma plataforma que permite o monitoramento de tarefas e o acompanhamento de projetos)**
- **Trello (aplicativo de gerenciamento)**

6.4 Ferramentas de teste unitário

- **Testing Library (frameworks)**
- **Jest (frameworks)**

7 Requisitos não Funcionais

7.1 Desempenho

garantir que a aplicação seja responsiva e rápida com tempos de carregamentos mínimos

7.2 Usabilidade

priorizar a usabilidade e a experiência do usuário, com uma interface intuitiva, design limpo e acessibilidade adequada

7.3 Compatibilidade

garantir compatibilidade com uma variedade de navegadores web, sistemas operacionais e dispositivos, para garantir uma experiência consistente para todos os usuários

8. Estratégias de Testes Unitários

As estratégias de testes unitários incluem:

- **Identificação de Casos de Teste:** Identificar todos os casos de teste relevantes para as unidades de código.
- **Implementação de Testes Unitários:** Desenvolver testes unitários para cada unidade de código, cobrindo diferentes cenários.
- **Automação de Testes:** Automatizar os testes unitários para garantir que sejam executados regularmente durante o desenvolvimento.
- **Integração com Ferramentas de CI/CD:** Integrar os testes unitários ao pipeline de integração contínua/entrega contínua para garantir testes regulares e automáticos através do **GitHub Actions**.
- **Cobertura de Código:** Monitorar e manter a cobertura de código dos testes unitários para garantir uma cobertura adequada do código.
- **Revisão de Código:** Revisar os testes unitários como parte do processo de revisão de código para garantir sua eficácia e qualidade.

9. Responsabilidades

- **Equipe de Desenvolvimento:** Responsáveis pelo desenvolvimento do aplicativo web e auxílio na criação dos testes.
- **Equipe de Documentação e Pesquisa**: Responsáveis pela elaboração da documentação e pesquisa do Projeto Hermes.
- **Equipe de Testadores:** Responsáveis por desenvolver e executar os testes unitários, além de fornecer feedback sobre a qualidade do código.

10. Riscos

- Atrasos no desenvolvimento podem impactar os prazos de entrega.
- Falhas nos testes unitários podem resultar em bugs não detectados.

11. Controle de Mudanças

Qualquer alteração no escopo, cronograma ou recursos do projeto deve ser submetida à aprovação do gerente de projeto antes de ser implementada.