**Documentação do Projeto Laravel – Gerenciador de projetos**

**Projeto:**

Gerenciamento de projetos.

**Escopo:**

O projeto consiste em um sistema em web que gerencia projetos em geral utilizando o Framework Laravel. O usuário irá conseguir se cadastrar, caso não tenha uma conta e se logar, caso já tenha uma. Um usuário de categoria gerente, consegue criar, editar, excluir projetos e gerenciar as inscrições de membros deles. O gerente será também capaz de visualizar todos os projetos criados por ele; já em relação a um usuário de categoria membro, ele consegue se inscrever em algum projeto determinado e visualizar os projetos em que está escrito. A execução dos métodos propostos só acontecerá mediante autenticação do usuário. É possível também acessar o dashboard de projetos a partir da tela inicial e sem autenticação, porém somente para visualização, sem executar nenhuma ação específica. Obs: A validação do usuário de categoria gerente, só vai acontecer mediante a alteração de uma variável diretamente no banco de dados, realizada pelo administrador do site.

**Objetivos do Projeto:**

**Específicos:**

1. Desenvolver um sistema de cadastro e autenticação para gerentes e membros de projetos.

2. Implementar funcionalidades para que gerentes possam criar, editar e excluir projetos, além de gerenciar as inscrições de membros.

3. Permitir que membros se inscrevam e visualizem projetos em que estão participando.

4. Criar dashboards específicos para gerentes e membros, com funcionalidades relevantes para cada perfil.

5. Garantir a segurança dos dados dos usuários e a integridade das transações no sistema.

6. Oferecer uma interface de visualização pública de projetos, acessível sem login, mas limitada a consultas sem possibilidade de edição.

**Mensuráveis:**

1. Garantir que o sistema tenha 80% de cobertura de testes unitários, assegurando a qualidade do código.

2. Reduzir o tempo de resposta do sistema para menos de 2 segundos nas operações de visualização e inscrição de projetos.

3. Atingir um nível de satisfação de usuários acima de 85% em testes de usabilidade.

4. Suportar até 1000 usuários simultâneos sem degradação de desempenho ou falhas de acesso.

**Atingíveis:**

1. Utilizar Laravel, um framework bem documentado e consolidado, para garantir a eficiência no desenvolvimento e manutenção do sistema.

2. Definir ciclos de desenvolvimento em sprints com entregas frequentes, garantindo revisões e feedback constantes durante o desenvolvimento.

3. Estabelecer uma equipe de desenvolvimento com experiência em Laravel e projetos similares, garantindo o cumprimento dos prazos e qualidade.

**Relevantes:**

1. Atender à crescente demanda por sistemas de gerenciamento de projetos eficientes e colaborativos.

2. Facilitar a organização de projetos, promovendo melhor comunicação entre gerentes e membros.

3. Prover uma solução escalável, que permita crescimento contínuo de acordo com o aumento da base de usuários e complexidade dos projetos.

**Temporais:**

1. Concluir o desenvolvimento do sistema em 4 meses, com entregas parciais a cada 2 semanas.

2. Implementar e testar o módulo de autenticação até o terceiro mês.

3. Finalizar o módulo de gerenciamento de projetos e inscrições até o terceiro mês.

4. Realizar testes beta e ajustes finais durante o último mês do cronograma.

**Recursos do Projeto:**

**Equipe:**

- 1 Gerente de Projetos: Responsável por coordenar o planejamento, execução e entrega do projeto, além de garantir o cumprimento dos prazos e a comunicação eficaz entre os membros da equipe.

- 2 Desenvolvedores Back-end: Responsáveis pelo desenvolvimento das funcionalidades do sistema utilizando o framework Laravel, além da integração com banco de dados e APIs.

- 1 Designer UI/UX: Responsável pela criação de interfaces amigáveis e intuitivas, garantindo a melhor experiência de usuário possível.

- 1 Especialista em Testes e QA: Responsável por garantir a qualidade do código e do sistema, realizando testes unitários, de integração e auditorias de segurança.

- 1 Analista de Marketing: Responsável por desenvolver e executar estratégias de marketing, especialmente voltadas para o lançamento e promoção do sistema.

**Tecnologias:**

- Laravel: Framework PHP utilizado para o desenvolvimento de back-end do sistema.

- Postgres SQL: Banco de dados relacional utilizado para armazenar dados dos projetos, usuários e inscrições.

- Bootstrap: Framework front-end para criar interfaces responsivas e compatíveis com dispositivos móveis.

- AWS/Heroku: Serviços de hospedagem escaláveis que permitem o crescimento da aplicação conforme a demanda de usuários aumenta.

- Git/GitHub: Ferramentas de controle de versão para gerenciar o código fonte, colaborar entre a equipe e garantir o rastreamento de alterações.

**Ferramentas de Gestão:**

- Trello: Plataforma de gerenciamento de tarefas e projetos, utilizadas para acompanhar o progresso do desenvolvimento, organizar sprints e formalizar assim então, um cronograma.

- Google Meet: Plataforma de comunicação interna utilizada para facilitar a colaboração entre os membros da equipe por meio de chats, videochamadas e compartilhamento de documentos.

- Figma: Ferramenta de design colaborativa usada para criar e prototipar interfaces de usuário, garantindo que todos os elementos visuais estejam alinhados com os requisitos do projeto.

- Miro: Ferramenta de desenvolvimento de diagramas em geral, utilizada para desenvolvimento do diagrama de uso e do diagrama de fluxo (fluxograma) do projeto.

**Análise de Riscos**

1. Risco: Atrasos no cronograma devido a requisitos não previstos.

Mitigação: Realizar reuniões regulares com stakeholders e a equipe de desenvolvimento para revisar o escopo do projeto, ajustar expectativas e realizar adaptações de requisitos de forma antecipada.

2. Risco: Falhas de segurança que comprometam dados dos usuários.

Mitigação: Aplicar boas práticas de segurança desde o início do desenvolvimento, como criptografia de dados, autenticação segura, revisão de código e auditorias de segurança periódicas, além de utilizar ferramentas específicas de verificação de vulnerabilidades.

3. Risco: Baixa adesão de usuários no lançamento.

Mitigação: Desenvolver uma estratégia de marketing robusta com campanhas de pré-lançamento para gerar interesse no sistema. Isso inclui a criação de landing pages, campanhas de e-mail, parcerias estratégicas e marketing de conteúdo.

4. Risco: Sobrecarga no sistema com o aumento rápido de usuários.

Mitigação: Implementar soluções escaláveis, como o uso de AWS/Heroku, e realizar testes de carga e desempenho regulares para garantir que o sistema possa lidar com picos de demanda. Além disso, ajustar a infraestrutura conforme necessário para garantir a disponibilidade e performance do sistema.