

Plano de Gestão do Projeto - Sistema Web

1. Período: até 30 de junho ?

2. Comunicação

Estratégia de Comunicação com os Stakeholders:

Reuniões periódicas com stakeholders.

Relatórios de progresso semanal.

Definição de canais de comunicação (XXXXX).

Descrição:

Definir uma abordagem eficaz de comunicação é essencial para manter os stakeholders informados e engajados. Serão realizadas reuniões periódicas, envio de relatórios de progresso semanal e o uso da plataforma Moodle como canal central de comunicação.

3. Ferramenta de Gestão Ágil e Sprints

Ferramenta de Gestão Ágil: Escolha e configuração (ex: Jira, Trello, Azure DevOps, ClickUp).

Duração das Sprints: Sprints quinzenais (2 semanas).

Cerimônias Scrum:

Planejamento da Sprint.

Daily Scrum.

Review (Revisão da Sprint).

Retrospectiva da Sprint.

Descrição:

Utilização de ferramentas ágeis (ex.: Jira, Trello, Azure DevOps, ClickUp) para organizar e acompanhar o progresso do projeto. As Sprints serão quinzenais e seguirão as cerimônias do Scrum para garantir alinhamento e melhoria contínua.

4. Definição dos Stakeholders

Identificação e mapeamento dos envolvidos:

Clientes.

Usuários finais.

Patrocinadores.

Equipe técnica.

Descrição:

Mapeamento e identificação dos principais envolvidos no projeto, como clientes, usuários finais, patrocinadores e equipe técnica, a fim de garantir que todas as expectativas e necessidades sejam consideradas.

5. Produto: Entendimento e Objetivos

Entendimento do Produto:

Alinhamento com o Product Owner sobre visão e metas.

Estudo do protótipo no Figma.

Avaliação da experiência do usuário (UX).

Descrição:

Compreensão aprofundada da visão do produto por meio do alinhamento com o Product Owner, análise do protótipo no Figma e avaliação da experiência do usuário (UX) para garantir que o desenvolvimento atenda aos objetivos do projeto.

6. Planilhas dos Médicos: Análise de Dados

Análise dos Dados Fornecidos:

Identificação de informações sensíveis (LGPD).

Relação com funcionalidades do app (cadastro, relatórios).

Descrição:

Avaliação dos dados fornecidos por médicos para identificar informações sensíveis de acordo com a LGPD e correlacionar essas informações com funcionalidades do aplicativo, como cadastro e relatórios.

7. Levantamento e Documentação de Requisitos

Requisitos Funcionais e Não-Funcionais.

Visão Holística do Sistema.

Mapeamento de Dados Pessoais conforme LGPD.

Documentos Utilizados:

Documento de Visão.

Especificação Técnica.

Descrição:

Coleta e registro de requisitos funcionais e não-funcionais, mapeando dados pessoais em conformidade com a LGPD. Documentação realizada por meio de artefatos como Documento de Visão e Especificação Técnica.

8. Definição da Arquitetura do Projeto

Arquitetura de Software:

Definição do padrão arquitetural (ex.: arquitetura em camadas, microserviços, arquitetura hexagonal).

Escolha dos padrões de comunicação entre sistemas (REST, GraphQL, gRPC, etc.).

Definição dos componentes principais do sistema e suas responsabilidades (backend, frontend, banco de dados, serviços externos).

Integração e Comunicação entre Módulos:

Definição de API Gateway ou integrações diretas entre serviços.

Estratégias de segurança para comunicação (autenticação e autorização entre módulos).

Escolha de protocolos e formatos de dados (JSON, XML, etc.).

Escalabilidade e Desempenho:

Planejamento para escalabilidade horizontal e vertical (uso de containers, Kubernetes, etc.).

Definição de estratégias de cache e balanceamento de carga.

Ferramentas e Tecnologias de Suporte à Arquitetura:

Ferramentas de CI/CD (ex.: Jenkins, GitLab CI).

Ferramentas de monitoramento e logging (ex.: Prometheus, Grafana, ELK Stack).

Descrição:

Estruturação da arquitetura de software do sistema, definindo padrões (ex.: em camadas, microserviços), formas de comunicação entre módulos (REST, GraphQL etc.), estratégias de escalabilidade, segurança e uso de ferramentas de suporte como CI/CD e monitoramento.

9. UX/UI Design

Aprovação dos Protótipos com Stakeholders.

Planejamento das Tarefas de Frontend.

Registro de Domínio e Hospedagem.

Definição do Painel Administrativo Responsivo.

Descrição:

Validação dos protótipos com stakeholders, planejamento das tarefas de frontend, definição do painel administrativo responsivo e registro do domínio e hospedagem para garantir uma experiência de usuário eficiente e amigável.

10. Backend

Definição da Stack Tecnológica:

Node.js

Spring Boot (Java)

Python (Flask, Django, FastAPI)

Organização dos Repositórios no GitLab.

Ferramentas Utilizadas:

VSCode

Postman

Swagger

Eclipse (para desenvolvimento em Java)

Implementação de APIs REST e Autenticação.

Descrição:

Escolha e definição da stack tecnológica (Node.js, Spring Boot, Python), organização dos repositórios no GitLab e uso de ferramentas como VSCode, Postman, Swagger e Eclipse para desenvolvimento das APIs e implementação de mecanismos de autenticação.

11. Banco de Dados

Escolha do Sistema Gerenciador de Banco de Dados (SGBD).

Modelagem do Banco.

Scripts Iniciais para MVP.

Infraestrutura e Segurança do Banco.

Ferramentas de Apoio:

DBDiagram

pgAdmin

Descrição:

Seleção do SGBD mais adequado, modelagem do banco de dados, elaboração de scripts iniciais para o MVP, estruturação da infraestrutura e segurança do banco, utilizando ferramentas como DBDiagram e pgAdmin.

12. Testes e Qualidade

Estratégia de Testes: (unitários, integração, ponta-a-ponta).

Ferramentas de QA:

JUnit

Cypress

Jest

Planejamento de QA Manual e Automatizado.

Descrição:

Planejamento da estratégia de testes (unitários, integração e ponta-a-ponta) com suporte de ferramentas como JUnit, Cypress e Jest. Serão realizados testes manuais e automatizados para garantir a qualidade contínua do sistema.

13. Segurança

Implementação de Controle de Acesso: (JWT, OAuth2, RBAC).

Funcionalidade de Recuperação de Senha.

Criptografia e Proteção de Dados.

Políticas de Segurança.

Descrição:

Implementação de mecanismos de segurança no sistema, incluindo controle de acesso com JWT, OAuth2 e RBAC, funcionalidades como recuperação de senha, uso de criptografia, proteção de dados e definição de políticas de segurança.

14. Cronograma

Descrição:

O cronograma do projeto estabelece marcos importantes e prazos para cada fase do desenvolvimento, permitindo o acompanhamento do progresso e o cumprimento dos prazos. Será dividido em etapas como levantamento de requisitos, design, desenvolvimento, testes e implantação, com entregas parciais a cada Sprint quinzenal até a data final prevista (30 de junho).

Observação: É recomendável manter esse cronograma atualizado em uma ferramenta de gestão visual (ex.: Gantt, roadmap no Jira/Trello) e revisá-lo a cada Sprint para refletir ajustes e imprevistos.

15. Riscos

Descrição:

A identificação e gestão de riscos visa prevenir possíveis problemas que possam comprometer o andamento do projeto. Cada risco será avaliado quanto à sua probabilidade de ocorrência e impacto, com definição de ações preventivas e planos de contingência.

Exemplos de riscos mapeados:

- **Atrasos nas entregas das Sprints:** Mitigado por reuniões diárias e boa definição de tarefas.
- **Mudanças frequentes nos requisitos:** Mitigado por reuniões de alinhamento com o Product Owner e documentação contínua.
- **Problemas com integração entre sistemas:** Mitigado por planejamento arquitetural detalhado e uso de testes automatizados.
- **Risco de não conformidade com a LGPD:** Mitigado com mapeamento dos dados sensíveis e validação com especialistas em proteção de dados.