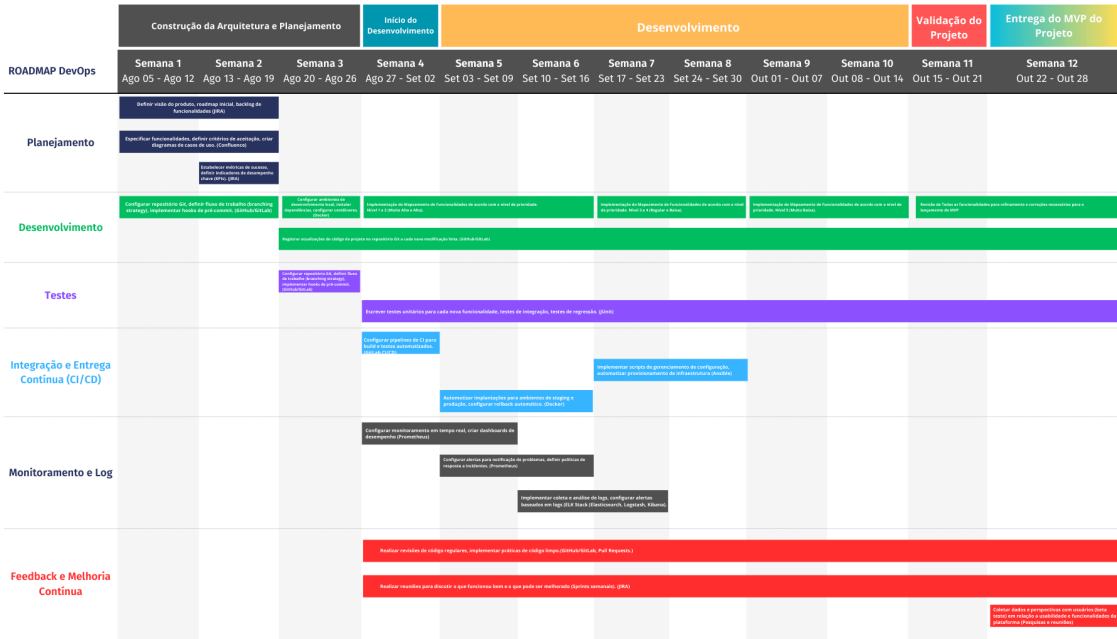


Roadmap - Projeto Plataforma de Marketplace de Energia



* A imagem também estará disponível em formato PDF no GitHub, no qual é possível utilizar o zoom para melhor visualização.

Este Roadmap envolve 5 etapas que sugerem a construção da plataforma de maneira eficiente e contato com os stakeholders, levando em consideração o uso de métodos ágeis e o ciclo DevOps para impulsionar o desenvolvimento. São eles:

- 1. Construção da Arquitetura e Planejamento
- 2. Início do Desenvolvimento
- 3. Desenvolvimento
- 4. Validação do Projeto
- 5. Entrega do MVP do Projeto

A partir disso, são segmentados em 12 semanas de desenvolvimento e construção do projeto envolvendo o Planejamento, Desenvolvimento, Testes, Integração e Entrega Contínua (CI/CD), Monitoramento e Log e Feedback e Melhoria Contínua, conforme descrito abaixo de maneira detalhada, como ponto de partida definimos a data de 05 de Agosto de 2024 (Data meramente ilustrativa, levando em consideração a sequência ininterrupta de 12 semanas do projeto.)

Semana 1 (Ago 05 - Ago 12)

- **Planejamento:**
 - Definir visão do produto, roadmap inicial, backlog de funcionalidades (JIRA)
 - Especificar funcionalidades, definir critérios de aceitação, criar diagramas de casos de uso. (Confluence)
- **Desenvolvimento:**
 - Configurar repositório Git, definir fluxo de trabalho (branching strategy), implementar hooks de pré-commit. (GitHub/GitLab)

Semana 2 (Ago 13 - Ago 19)

- **Planejamento:**
 - Estabelecer métricas de sucesso, definir indicadores de desempenho chave (KPIs). (JIRA)
- **Desenvolvimento:**
 - Configurar repositório Git, definir fluxo de trabalho (branching strategy), implementar hooks de pré-commit. (GitHub/GitLab)

Semana 3 (Ago 20 - Ago 26)

- **Desenvolvimento:**
 - Configurar ambientes de desenvolvimento local, instalar dependências, configurar contêineres. (Docker)
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Configurar repositório Git, definir fluxo de trabalho (branching strategy), implementar hooks de pré-commit. (GitHub/GitLab)

Semana 4 (Ago 27 - Set 02)

- **Desenvolvimento:**
 - Implementação do mapeamento de funcionalidades de nível de prioridade 1 e 2 (Muito Alta e Alta).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**

- Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Integração e Entrega Contínua (CI/CD):**
 - Configurar pipelines de CI para build e testes automatizados. (GitLab CI/CD)
- **Monitoramento e Log:**
 - Configurar monitoramento em tempo real, criar dashboards de desempenho (Prometheus)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 5 (Set 03 - Set 09)

- **Desenvolvimento:**
 - Implementação do mapeamento de funcionalidades de nível de prioridade 1 e 2 (Muito Alta e Alta).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Integração e Entrega Contínua (CI/CD):**
 - Automatizar implantações para ambientes de staging e produção, configurar rollback automático. (Docker)
- **Monitoramento e Log:**
 - Configurar monitoramento em tempo real, criar dashboards de desempenho (Prometheus)
 - Configurar alertas para notificação de problemas, definir políticas de resposta a incidentes. (Prometheus)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 6 (Set 10 - Set 16)

- **Desenvolvimento:**
 - Implementação do mapeamento de funcionalidades de nível de prioridade 1 e 2 (Muito Alta e Alta).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Integração e Entrega Contínua (CI/CD):**
 - Automatizar implantações para ambientes de staging e produção, configurar rollback automático. (Docker)
- **Monitoramento e Log:**
 - Configurar alertas para notificação de problemas, definir políticas de resposta a incidentes. (Prometheus)
 - Implementar coleta e análise de logs, configurar alertas baseados em logs (ELK Stack (Elasticsearch, Logstash, Kibana)).
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 7 (Set 17 - Set 23)

- **Desenvolvimento:**
 - Implementação do Mapeamento de Funcionalidades de acordo com o nível de prioridade. Nível 3 e 4 (Regular e Baixa).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Integração e Entrega Contínua (CI/CD):**
 - Implementar scripts de gerenciamento de configuração, automatizar provisionamento de infraestrutura (Ansible)
- **Monitoramento e Log:**

- Implementar coleta e análise de logs, configurar alertas baseados em logs (ELK Stack (Elasticsearch, Logstash, Kibana)).
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 8 (Set 24 - Set 30)

- **Desenvolvimento:**
 - Implementação do Mapeamento de Funcionalidades de acordo com o nível de prioridade. Nível 3 e 4 (Regular e Baixa).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Integração e Entrega Contínua (CI/CD):**
 - Implementar scripts de gerenciamento de configuração, automatizar provisionamento de infraestrutura (Ansible)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 9 (Out 01 - Out 07)

- **Desenvolvimento:**
 - Implementação do Mapeamento de Funcionalidades de acordo com o nível de prioridade. Nível 5 (Muito Baixa).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Feedback e Melhoria Contínua:**

- Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
- Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 10 (Out 08 - Out 14)

- **Desenvolvimento:**
 - Implementação do Mapeamento de Funcionalidades de acordo com o nível de prioridade. Nível 5 (Muito Baixa).
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 11 (Out 15 - Out 21)

- **Desenvolvimento:**
 - Revisão de Todas as funcionalidades para refinamento e correções necessárias para o lançamento do MVP
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)

Semana 12 (Out 22 - Out 28)

- **Desenvolvimento:**
 - Revisão de Todas as funcionalidades para refinamento e correções necessárias para o lançamento do MVP
 - Registrar atualizações do código do projeto no repositório Git a cada nova modificação feita. (GitHub/GitLab)
- **Testes:**
 - Escrever testes unitários para cada nova funcionalidade, testes de integração, testes de regressão. (JUnit)
- **Feedback e Melhoria Contínua:**
 - Realizar revisões de código regulares, implementar práticas de código limpo.(GitHub/GitLab, Pull Requests.)
 - Realizar reuniões para discutir o que funcionou bem e o que pode ser melhorado (Sprints semanais). (JIRA)
 - Coletar dados e perspectivas com usuários (beta teste) em relação a usabilidade e funcionalidades da plataforma (Pesquisas e reuniões)

Backlog do Produto:

- Cadastro de Usuários
- Busca e Comparação de Ofertas
- Negociação e Contratação de Energia
- Monitoramento e Gestão de Contratos
- Sistema de Notificações
- Relatórios e Análises

Ciclos de Desenvolvimento:

- **Sprints:** Cada sprint terá a duração de 3 meses (Total de 1 ano para lançamento).
- **Sprint 1:**
 - Implementação do Cadastro de Usuários
 - Desenvolvimento da Interface de Busca e Comparação
 - Testes e Feedback dos Usuários
- **Sprint 2:**
 - Implementação da Negociação e Contratação de Energia
 - Desenvolvimento do Sistema de Notificações
 - Testes e Feedback dos Usuários

- **Sprint 3:**
 - Desenvolvimento de Funcionalidades de Monitoramento e Gestão de Contratos
 - Criação de Relatórios e Análises
 - Testes e Feedback dos Usuários
- **Sprint 4:**
 - Ajustes e Melhorias Baseados nos Feedbacks
 - Preparação para o Lançamento MVP