3º Ano

**Soluções Computacionais**

**<Falcommerce>**

|  |  |
| --- | --- |
| **Nome Completo** | **RA** |
| Andrews Soares Frasson | 22.00908-6 |
| Joao Pedro Crepaldi Fukumura | 22.01243-5 |
| lucca leça aggio | 21.00867-0 |
|  |  |
|  |  |
|  |  |

Documentação de um Produto de Software – Autora do Modelo: Profa. Dra. Ana Paula Gonçalves Serra

# 2024

# 1. Introdução

A Falcommerce foi criada para enfrentar os desafios que pequenos e médios empreendedores encontram ao tentar utilizar ferramentas avançadas de programação. Muitos desses empreendedores dependem de programadores caros para desenvolvimento e manutenção, e as soluções de e-commerce existentes são frequentemente ou muito simples e limitadas, ou extremamente complexas e caras. Há uma lacuna no mercado para soluções intermediárias que sejam flexíveis e escaláveis, permitindo que esses negócios cresçam sem enfrentar altos custos ou complexidade excessiva.

O objetivo técnico da Falcommerce é desenvolver um site web funcional que sirva como ponto de entrada e apresentação do conceito. Isso inclui a criação de páginas essenciais, uma interface intuitiva e responsiva, e pelo menos uma funcionalidade de exemplo para demonstrar o potencial da plataforma. A arquitetura do sistema será projetada para ser escalável, facilitando a adição de novas funcionalidades no futuro e garantindo uma integração simplificada de diversas ferramentas essenciais para operações eficientes.

Em termos de objetivos de negócio, a Falcommerce planeja realizar uma pesquisa de mercado aprofundada para identificar as necessidades mais urgentes dos pequenos e médios e-commerces. Com base nisso, será desenvolvido um catálogo de funcionalidades potenciais, categorizadas por departamentos, e um plano de desenvolvimento futuro priorizando as funcionalidades mais demandadas. A Falcommerce também estabelecerá métricas de engajamento, coletará feedback dos usuários, explorará parcerias acadêmicas e com empreendedores locais, e desenvolverá uma estratégia inicial de marketing digital para promover o conceito e atrair potenciais usuários.

# 2. Plano de Gerenciamento de Projetos

## 2.1. Ciclo de Vida Abordado

Para o desenvolvimento da Falcommerce, optamos por uma abordagem híbrida, combinando elementos dos métodos ágil e preditivo. Inicialmente, será utilizado um planejamento preditivo para definir os requisitos e a arquitetura do sistema, estabelecendo marcos e prazos para os entregáveis chave. Optamos por uma abordagem híbrida, combinando métodos preditivos e ágeis, para a criação do site da Falcommerce devido à necessidade de equilibrar planejamento estruturado e flexibilidade. O método preditivo nos permite definir claramente os requisitos iniciais, a arquitetura do sistema e estabelecer marcos essenciais, garantindo uma visão clara e objetivos bem delineados desde o início.

## 2.2. Principais requisitos das principais entregas/produtos

Estruturar a documentação de requisitos em uma tabela para facilitar a consulta. Adicionar uma categorização entre requisitos obrigatórios e opcionais, ou de alta, média e baixa prioridade

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | | **Descrição** | | **Categoria** | **Prioridade** |
| Página Inicial  (Home) | | Apresentação da empresa, principais  serviços e chamada para ação. | | Obrigatorio | Alta |
| Catálogo de  Funcionalidades | | Lista completa de funcionalidades com descrições e recomendações de uso. | | Obrigatorio | Alta |
| Página de Planos e Preços | | Comparação de planos e preços, com detalhes e opções de assinatura. | | Obrigatorio | Média |
| Página de Sobre  Nós |  | Informações sobre a história, missão, visão e valores da empresa. | | Opcional | Baixa |
| Página  Contato | de | Formulário envio mensagens informações contato | para de e de | Obrigatorio | Alta |
| Página de  Perguntas  Frequentes (FAQ**)** | | Lista perguntas respostas organizadas categorias. | de e  por | Opcional | Média |
| Página de  Login/Cadastro | | Formulários para autenticação e criação de contas. | | Obrigatorio | Alta |

## 2.3. Estrutura Analítica do Projeto – Fases e principais entregas

A EAP deve incluir uma descrição detalhada de cada fase do projeto e suas entregas, com prazos definidos. Se possível, incluir ferramentas visuais para apresentar a EAP.

**Estrutura Analítica do Projeto – Fases e Principais Entregas**

### 1. Fase de Planejamento

1.1. Definição de Requisitos o **Descrição:** Coleta e documentação dos requisitos funcionais e não funcionais junto aos stakeholders. o **Entrega:** Documento de Requisitos.

* **Prazo:** 2 semanas (até 15 de outubro).

1.2. Projeto Arquitetural o **Descrição:** Definição da arquitetura de software e seleção de tecnologias adequadas.

* **Entrega:** Documento de Arquitetura.
* **Prazo:** 1 semana (até 22 de outubro).

### 2. Fase de Desenvolvimento

2.1. Configuração do Ambiente de Desenvolvimento o **Descrição:** Configuração do servidor Node.js e setup do Next.js para o front-end.

* **Entrega:** Ambiente de desenvolvimento configurado. o **Prazo:** 1 semana (até 29 de outubro).

2.2. Desenvolvimento do Backend o **Descrição:** Implementação da API no Node.js e integração com o banco de dados Firebird.

* **Entrega:** Backend funcional.
* **Prazo:** 2 semanas (até 12 de novembro).

2.3. Desenvolvimento do Frontend o **Descrição:** Criação de componentes React/Next.js e implementação de rotas e funcionalidades.

* **Entrega:** Frontend desenvolvido.
* **Prazo:** 2 semanas (até 26 de novembro).

2.4. Integração de Sistemas o **Descrição:** Integração entre frontend e backend, incluindo testes de integração.

* **Entrega:** Sistema integrado.
* **Prazo:** 1 semana (até 3 de dezembro).

### 3. Fase de Testes

3.1. Testes de Funcionalidades o **Descrição:** Elaboração e execução de testes unitários e end-to-end com Jest e Cypress.

* **Entrega:** Relatório de Testes Funcionais.
* **Prazo:** 1 semana (até 10 de novembro).

3.2. Testes de Usabilidade

**Descrição:** Realização de testes com usuários reais e ajustes de interface.

* **Entrega:** Relatório de Testes de Usabilidade.
* **Prazo:** 1 semana (até 17 de novembro).

3.3. Preparação para Apresentação o **Descrição:** Revisão final do produto e otimização de performance.

* **Entrega:** Produto finalizado e otimizado.
* **Prazo:** 1 semana (até 24 de novembro).

### 4. Documentação Final

4.1. Documentação Técnica o **Descrição:** Compilação da documentação técnica do projeto. o **Entrega:** Documentação completa. o **Prazo:** 1 semana (até 1 de dezembro).

## 2.4. Marcos

Adicionar uma descrição clara de como cada marco será avaliado e os critérios de sucesso para a entrega da fase.

### 1- Conclusão do Planejamento

* **Descrição:** Finalização da definição de requisitos e projeto arquitetural.
* **Critérios de Sucesso:**

Documento de requisitos aprovado por todos os stakeholders.

* + Arquitetura de software definida e documentada, com tecnologias selecionadas.
  + Aprovação formal do plano de projeto.

### 2- Configuração do Ambiente de Desenvolvimento

* **Descrição:** Ambiente de desenvolvimento configurado para backend e frontend.
* **Critérios de Sucesso:**
  + Servidor Node.js operacional. o Setup do Next.js completo e funcional.
  + Ambiente testado e validado pela equipe de desenvolvimento.

### 3 - Desenvolvimento do Backend

* **Descrição:** Implementação e integração do backend.
* **Critérios de Sucesso:**
  + API Node.js implementada conforme especificações. o Integração bem-sucedida com o banco de dados Firebird.
  + Testes unitários iniciais aprovados.

### 4 - Desenvolvimento do Frontend

* **Descrição:** Criação e implementação de componentes e funcionalidades do frontend.
* **Critérios de Sucesso:**
  + Componentes React/Next.js desenvolvidos conforme o design.

Funcionalidades e rotas implementadas e testadas.

* + Feedback positivo em revisões de código.

### 5- Integração de Sistemas

* **Descrição:** Integração completa entre frontend e backend.
* **Critérios de Sucesso:**
  + Integração testada sem erros críticos. o Funcionalidades principais operacionais.
  + Aprovação em testes de integração.

### 6- Fase de Testes

* **Descrição:** Execução de testes de funcionalidades e usabilidade.
* **Critérios de Sucesso:**
  + Casos de teste documentados e executados com sucesso. o Feedback positivo em testes de usabilidade com usuários reais.
  + Ajustes realizados conforme necessário.

### 7- Preparação para Apresentação

* **Descrição:** Revisão final e otimização do produto.
* **Critérios de Sucesso:**
  + Produto revisado e aprovado pela equipe de qualidade. o Performance otimizada conforme critérios estabelecidos.
  + Produto pronto para lançamento.

### 8- Documentação Final

* **Descrição:** Compilação e entrega da documentação técnica.
* **Critérios de Sucesso:**
  + Documentação completa e revisada.
  + Disponível para a equipe de suporte e clientes. o Aprovação final pelo gerente de projeto.

## 2.5. Partes interessadas do projeto

Incluir um gráfico RACI (Responsibility, Accountability, Consulted, Informed) para descrever melhor as funções e responsabilidades de cada parte interessada, garantindo clareza nos papéis e na comunicação.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Marcos/Funções | Gerent  e de  Projeto | Equipe de  Desenvolvimen to | Analista de Requisito s | Equip e de  Teste s | Stakeholde rs |
| Conclusão do  Planejamento | A | R | C | I | I |
| Configuraç ão do Ambiente de  Desenvolvimento | I | R | C | I | I |
| Desenvolvimento do Backend | I | R | C | I | I |
| Desenvolvimento do Frontend | I | R | C | I | I |
| Integração de  Sistemas | A | R | C | C | I |
| Fase de Testes | I | R | I | R | I |
| Preparação para  Apresentação | A | C | C | C | I |
| Documentação  Final | A | R | C | I | I |

## 2.6. Restrições

Relacione as restrições do projeto, ou seja, limitação aplicável ao projeto, a qual afetará seu desempenho. Limitações reais: orçamento, recursos, tempo de alocação, ... Ex.: Entrega no dia 30/11/2024.

### 1-Prazo de Entrega

* **Descrição:** O projeto deve ser concluído e entregue até o dia 30/11/2024.
* **Impacto:** O prazo apertado pode limitar o tempo disponível para desenvolvimento, testes e ajustes finais, aumentando o risco de entrega de um produto com falhas ou incompleto.

### 2- Orçamento

* **Descrição:** O orçamento total disponível para o projeto é de R$ 150.000,00.
* **Impacto:** Com um orçamento restrito, a alocação de recursos financeiros deve ser extremamente cuidadosa, o que pode limitar a contratação de mão de obra adicional, aquisição de ferramentas e tecnologias, e outras despesas necessárias para o desenvolvimento do projeto.

### 3- Recursos Humanos

* **Descrição:** Apenas um desenvolvedor está disponível para todo o ciclo de desenvolvimento do projeto.
* **Impacto:** A limitação de recursos humanos pode levar a sobrecarga de trabalho, atrasos no cronograma e potencial comprometimento da qualidade do código e funcionalidades implementadas.

### 4- Disponibilidade dos Stakeholders

* **Descrição:** A disponibilidade dos stakeholders, que são proprietários de ecommerce de pequeno e médio porte, é limitada.
* **Impacto:** A coordenação e comunicação com os stakeholders podem ser dificultadas, resultando em atrasos na coleta de requisitos, feedback e validação das funcionalidades implementadas.

### 5- Integração com Sistemas Externos

* **Descrição:** Necessidade de integração com sistemas externos já utilizados pelos e-commerces para gerenciamento de negócios.
* **Impacto:** As integrações podem ser complexas e demandar mais tempo e recursos do que o previsto, impactando o cronograma e o orçamento do projeto. Além disso, a compatibilidade e a estabilidade das integrações podem ser desafiadoras.

## 2.7. Premissas

### Disponibilidade dos Stakeholders

* **Descrição:** Os stakeholders, proprietários de e-commerce de pequeno e médio porte, estarão disponíveis para reuniões de feedback e validação de requisitos por pelo menos 50% do tempo necessário durante as fases críticas do projeto.
* **Impacto:** A disponibilidade dos stakeholders é crucial para garantir que o desenvolvimento esteja alinhado com as necessidades e expectativas dos usuários finais. A falta de disponibilidade pode resultar em atrasos e retrabalho.

### Estabilidade dos Requisitos

* **Descrição:** Os requisitos coletados durante a fase de planejamento permanecerão estáveis e não sofrerão mudanças significativas ao longo do desenvolvimento.
* **Impacto:** A estabilidade dos requisitos permite um planejamento mais preciso e reduz o risco de atrasos e aumentos de custo devido a mudanças inesperadas.

### Capacidade Técnica do Desenvolvedor

* **Descrição:** O desenvolvedor único designado para o projeto possui todas as habilidades técnicas necessárias para implementar as funcionalidades planejadas, incluindo conhecimentos em Node.js, Next.js e integração com sistemas externos.
* **Impacto:** A capacidade técnica do desenvolvedor é fundamental para o sucesso do projeto. Qualquer lacuna de conhecimento pode resultar em atrasos ou na necessidade de contratar recursos adicionais. **Acesso a Ferramentas e Tecnologias**
* **Descrição:** Todas as ferramentas e tecnologias necessárias para o desenvolvimento, testes e implantação do projeto estarão disponíveis e funcionarão conforme esperado.
* **Impacto:** A disponibilidade e funcionalidade das ferramentas e tecnologias são essenciais para manter o cronograma do projeto. Problemas com ferramentas podem causar atrasos significativos.

### Suporte Técnico e Infraestrutura

* **Descrição:** A infraestrutura técnica necessária, incluindo servidores, ambientes de desenvolvimento e ferramentas de colaboração, estará disponível e será suportada adequadamente durante todo o projeto.
* **Impacto:** A infraestrutura técnica é vital para o desenvolvimento contínuo e eficiente. Qualquer falha ou indisponibilidade pode interromper o progresso do projeto.

### Aceitação do Produto pelo Mercado

* **Descrição:** O mercado de e-commerce de pequeno e médio porte estará receptivo à plataforma Falcommerce, considerando-a uma solução viável e desejável para suas necessidades.
* **Impacto:** A aceitação do produto pelo mercado é crucial para o sucesso comercial do projeto. Se a premissa não se confirmar, pode haver necessidade de ajustes na estratégia de marketing e posicionamento do produto.

## 2.8. Formas de Comunicação

Identifique a forma de comunicação e periodicidade com as partes interessada.

Pode-se utilizar o modelo de matriz utilizado na aula de Gestão de Projetos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Partes**  **Interessadas** | **Tipo de**  **Comunicação** | | **Frequência** | **Responsável** | |
| Gerente de Projeto | Reunião  Alinhamento  Interno | de | Semanal | Gerente de  Projeto | |
| Equipe de  Desenvolvimento | Relatório  Progresso  Técnico | de | Diário | Desenvolvedor  Líder |  |
| Clientes (E-  commerce) | Atualização  Status do Proje | de to | Quinzenal | Gerente  Projeto | de |
| Todos os  Stakeholders | Relatório de  Status Geral do  Projeto | | Mensal | Gerente  Projeto | de |

## 2.9. Riscos

Identifique os riscos do projeto, categoria e ações. Pode-se utilizar o modelo de matriz de risco utilizado na aula de Gestão de Projetos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risco | Categoria | Probabilidade | Impacto | Ações de  mitigação |
| Atrasos no  desenvolvimento | Cronograma | Alta | Alto | Revisar cronograma regularmente; alocar recursos  extras se necessário. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mudanças  requisitos cliente | nos do | Escopo | Média | Alto | Estabelecer um processo claro de gerenciamento de mudanças; comunicação constante com clientes. |
| Falhas segurança plataforma | de na | Segurança | Baixa | Alto | Realizar auditorias de segurança; implementar protocolos de segurança rigorosos. |
| Indisponibilidade de recursos  técnicos | | Recursos | Média | Médio | Planejamento de  contingência; treinamento  cruzado da equipe. |
| Aceitação baixa pelo mercado | | Comercial | Média | Alto | Pesquisa de mercado contínua; ajustar estratégias de marketing |
|  | |  |  |  | conforme necessário. |

**2.10. Cronograma e etapas do projeto.**

Incorporar cronograma com fases detalhadas e prazos realistas. Adicione também recursos alocados para cada fase, destacando dependências entre as tarefas. O cronograma deve estar de acordo com o modelo de processo escolhido.

Para garantir uma execução eficiente e organizada do projeto Falcommerce, desenvolvemos um cronograma detalhado utilizando o Microsoft Project. Este cronograma foi elaborado com base em um modelo de processo ágil, permitindo flexibilidade e ajustes contínuos durante o desenvolvimento. A seguir, apresentamos as principais fases e prazos do projeto, bem como os recursos alocados para cada etapa e as dependências entre as tarefas.

***Observações***

***OBS 1: Caso o item ciclo de vida abordado seja desenvolvimento ágil realize o preenchimento da documentação de software a partir do item 5 e remova os itens de 3 a 8 (AZUL NO TEXTO).***

***OBS 2: Caso o item ciclo de vida abordado seja cascata realize o preenchimento da documentação de software a partir do item 3 e remova o item 9 A 11 (VERDE NO TEXTO).***

**O item 10 deve ser elaborado para qualquer ciclo de vida.**

## 3. Requisitos do Sistema

**3.1.Extração de Requisitos**

Especifique como os requisitos foram coletados, caso tenha sido entrevista ou questionário, inclua as evidências no apêndice.

**3.2.Requisitos Funcionais**

Descreva as funcionalidades que o software deve ter, de acordo, com a extração de requisitos.

**3.3.Requisitos Não Funcionais**

Descreva os requisitos não funcionais, com as características de qualidade (performance, usabilidade, segurança, etc.).

## 4. Modelagem do Sistema

**4.1.Diagrama de Casos de Uso**

Represente dos requisitos funcionais, por meio do diagrama de caso de uso e se necessário as especificações detalhadas de cada ator e caso de uso.

**4.2.Diagrama de Classes**

Elabore a estrutura de solução, por meio do diagrama de classes.

**4.3.Outros Diagramas UML**

Elabore diagramas complementares se necessário para uma melhor modelagem, e proposta de solução. Exemplos de outros diagramas: sequência, atividades, componentes, estados etc.

## 5. Tecnologias e Ferramentas Utilizadas para Desenvolvimento

Descreva a arquitetura, tecnologias utilizadas para a implementação, banco de dados e ambiente de implantação.

## 6. Design de Interface

Descreva as características de experiência do usuário e como será essa interface do usuário, justificando cores, usabilidade, acessibilidade etc.

Elabore o protótipo das telas.

**7. Desenvolvimento e Implementação**

Indique o link do repositório com o código implementado.

## 8. Testes

Detalhe o plano de testes. Defina os tipos de testes (unitários, integração, aceitação, carga) e as ferramentas utilizadas. Especificar métricas de qualidade para a validação dos testes e indique o repositório com os testes

**9. Levantamento dos Requisitos do Sistema de Software – Product**

# Backlog

Crie um Product Backlog, de acordo com a extração de requisitos, caso tenha sido entrevista ou questionário, inclua as evidências no apêndice.

as reuniões realizadas com o parceiro. O Product Backlog deve conter:

* Cartões de histórias que representam os requisitos do software.
* Prioridade de cada história.
* Estimativa de esforço.
* Sprint em que será implementada.

Esse Product Backlog deve ser criado em alguma ferramenta, como: Azure, Trello ou Jira.

## 10. Tecnologias e Ferramentas Utilizadas para Desenvolvimento

Descreva a arquitetura, tecnologias utilizadas para a implementação, banco de dados e ambiente de implantação.

## 11. Sprint X

Apresente o planejamento, desenvolvimento, testes, entrega e retrospectiva da Sprint 1. Lembre-se que você pode utilizar modelos da UML para detalhar a sprint, como: diagrama de classes, sequência, implantação etc.

Este item deve se repetir com todos os itens para as demais Sprints que serão desenvolvidas no projeto.

### 11.1.1. Planejamento da Sprint 1 – Sprint Backlog

Apresente um detalhamento de como será implementada a Sprint, definindo as regras de negócio, proposta de arquitetura, solução técnica e implementação. Criar um quadro de tarefas detalhado de cada Sprint. Esse quadro de tarefas deve ser criado em alguma ferramenta, como: Azure, Trello ou Jira.

### 11.1.2. Design de Interface

Descreva as características de experiência do usuário e como será essa interface do usuário, justificando cores, usabilidade, acessibilidade etc.

Elabore o protótipo das telas.

### 11.1.3. Implementação e Testes – Sprint 1

Indique o link do repositório com o código e os testes implementados na Sprint 1.

Os testes a serem realizados são de BDD (Behavior Driven Development) e TDD (Test Driven Development).

### 11.1.4. Review da Sprint 1

Descreva o evento de Review da Sprint. Realizar um breve relato dos resultados desta Sprint, o objetivo foi cumprido? E o que será realizado na próxima Sprint.

Se possível descreva o parecer do parceiro testando essa Sprint.

### 11.1.5. Retrospectiva da Sprint 1

Descreva o evento de Retrospectiva da Sprint. Realizar um breve relato de como foi a Sprint em relação a indivíduos, interações, pontos fontes e fracos e o que deve ser mantido e o que deve ser melhorado para a próxima Sprint.

## 12. Conclusão

Adicione uma análise mais completa dos resultados alcançados, comparando com os objetivos estabelecidos inicialmente. Incluir uma seção de 'Próximos Passos', indicando o que será feito após a conclusão do projeto.