**UNISOCIESC**  
**GESTAO E QUALIDADE DE SOFTWARE**

BIANCA SILVA SANTOS

EZEQUIEL LUIS DA SILVA

LUANA SIQUEIRA RONCHI

GUSTAVO LUCAS SCHNEBERGER

VITOR DA SILVA

**Smart Shop**

**JOINVILLE  
2024**

BIANCA SILVA SANTOS

EZEQUIEL LUIS DA SILVA

LUANA SIQUEIRA RONCHI

GUSTAVO LUCAS SCHNEBERGER

VITOR DA SILVA

**Smart Shop**

Trabalho apresentado a UniSociesc como requisito para conclusão do gestao e qualidade de software.

Orientador: Prof. Marcelo Petri

**JOINVILLE**

**2024**

**RUSUMO**

Este trabalho tem como objetivo aplicar os conceitos estudados em sala de aula sobre gestão, qualidade e teste de software. Os alunos deverão escolher um modelo de negócio e desenvolver um projeto que contemple desde o planejamento de desenvolvimento até a definição de testes e metodologias, identificando riscos e elaborando um cronograma de atividades.

**SUMÁRIO**

**INTRODUÇÃO..................................................................................................................5**

**REUNIÃO..........................................................................................................................6**

**SOLICITAÇÃO DE MUDANÇA........................................................................................7**

**LISTA DOS MARCOS......................................................................................................9**

**DEFINIÇÃO DE TESTES................................................................................................10**

**METODOLOGIA..............................................................................................................11**

**ANÁLISE DE RISCOS....................................................................................................12**

**DIAGRAMAS...................................................................................................................14**

**REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS........................................................19**

**CONCLUSÃO...................................................................................................................21**

**1. INTRODUÇÃO**

Cria um projeto de uma empresa conduzido com as regras de qualidade e usando e metodologia gerenciamento de projeto para cada pessoa da equipe sabe o que tem que fazer e o dia da entra.

Isso e necessário para se uma dia nos entra um trabalho numa empresa nos saber como lida com metodologia da empresa que nos tiver trabalhando.

Nosso objetivo é criar uma plataforma digital que facilite a comparação de preços entre diferentes supermercados, promovendo transparência e competitividade no mercado. Com isso, pretendemos simplificar o processo de decisão de compra para os consumidores, proporcionando uma experiência de compra mais eficiente e econômica. Além disso, buscamos estabelecer parcerias estratégicas com empresas do setor, oferecendo-lhes benefícios em troca de informações sobre pesquisas de produtos e relevância para os clientes. Essa troca de informações permitirá uma coleta de dados mais eficiente sobre os preços praticados no mercado, auxiliando tanto consumidores quanto empresas a tomarem decisões mais informadas e competitivas

**2. REUNIÃO**  
  
08/04/2024   
  
Durante a reunião, discutimos detalhadamente as necessidades e os requisitos do trabalho.  
  
Prototipação  
Definição de Testes  
Escolha de Metodologia   
Análise de Riscos   
Formatação de Documento   
Slides da Apresentação  
  
Optamos pela utilização da metodologia Kanban, cuja implementação ficou sob a responsabilidade de Klaiver. Para gerenciar as atividades e acompanhar o progresso das tarefas, estamos utilizando a tecnologia Trello. Esta ferramenta nos permite visualizar de forma clara e organizada as responsabilidades de cada membro da equipe, bem como os prazos de entrega das funções atribuídas a cada um.  
  
 Prototipação (Luana Siqueira Ronchi, 30 de abr - 22 de abr)  
 Definição de Testes (Ezequiel Luiz da Silva, abr – 29 )  
 Escolha de Metodologia (Klaiver Rodrigues de Cordova, 16 de abr - 30 de abr)  
 Análise de Riscos ( Bianca Silva Santos, 14 de mai )  
 Formatação de Documento (Gustavo Lucas Schneberger, Bianca Silva 31 de mai )  
 Slides da Apresentação (Vitor Ferreira da Silva, 31 de mai )

# 3. SOLICITAÇÃO DE MUDANÇA

# JUSTIFICATIVA

* **Risco Identificado:** A falta de investimento pode comprometer a qualidade e o sucesso do projeto, levando a atrasos, escopo reduzido ou falha na entrega.
* **Impacto Potencial:** Pode afetar negativamente a satisfação do cliente, a reputação da empresa e a viabilidade financeira do projeto.
* **Necessidade de Mudança:** Estabelecer um processo estruturado de gestão de investimentos proporcionará uma abordagem sistemática para alocar recursos financeiros de forma eficiente, priorizando iniciativas que maximizem o valor para o projeto.

# Classificação de impacto no projeto:

|  |  |
| --- | --- |
| **Análise de Impacto** | **Descrição** |
| Esforço Estimado (Horas) | 40 |
| Custo Estimado (R$) | R$8.000,00 (Com base no valor de R$200,00/h) |
| Impacto no Prazo (Dias) | 5 |

# 3.1SOLICITAÇÃO DE MUDANÇA NOS REQUISITOS Descrição sumária:

Identificamos a necessidade de implementar novas funcionalidades para o sistema para atribuir melhor usabilidade e solução de problemas para os usuários.

**Adicionar os seguintes requisitos:**

RF016: O sistema deve conter uma API para coletar o dado dos produtos, integrável com o sistema dos mercados;

RF017: O sistema deve permitir o usuário adicionar e remover produtos da sua lista de compras;

RF018: O sistema deve permitir o usuário solicitar um entregador do aplicativo para realizar a sua compra nos mercados da sua lista de compras;

RF019: O sistema deve cobrar uma taxa maior de entrega se os produtos forem de mercados ou estabelecimentos diferentes;

RF020: O sistema deve permitir a lista de compras poder ser gerada semanalmente ou mensalmente baseada na última lista feita;

RF021: O sistema deve permitir o usuário verificar o valor da lista dos seus produtos em vários mercados diferentes;

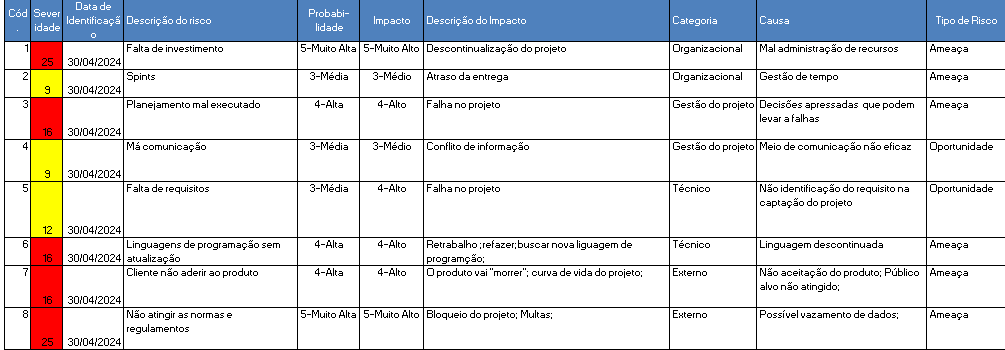
# JUSTIFICATIVA

A mudança é necessária para incluir novas funcionalidades, melhorar a forma de coleta de dados e distribuição dos produtos comprados.

# Classificação de impacto no projeto:

|  |  |
| --- | --- |
| **Análise de Impacto** | **Descrição** |
| Esforço Estimado (Horas) | 8h |
| Custo Estimado (R$) | Indefinido |
| Impacto no Prazo (Dias) | 1 dia |

**3.2 REGISTRO DE RISCOS**



**4. LISTA DOS MARCOS**

A lista dos marcos contém os marcos do projeto.

Ela é criada no processo [Definir as atividades](https://escritoriodeprojetos.com.br/definir-as-atividades), identificando-se em cada fase, os momentos mais importantes do projeto.

Esse documento tem como objetivo documentar e apresentar cada fase do projeto com seus marcos e sua previsão.

# 4.1 FASES-MARCOS

|  |  |  |  |
| --- | --- | --- | --- |
| Fase ou Grupo de Processos | Marcos | Previsão | |
| Iniciação | Realizado reunião para a definir o seguimento do projeto, feito com todo os integrantes do grupo para dar seguimento a prototipação do projeto sendo efetuado o levantando de requisitos funcionais e não funcionais do projeto.  Projeto Aprovado | 08/04/2024 | |
| Planejamento | Efetuar escolha de metodologia.  Efetuar análise de riscos para gestão e qualidade do Software englobando:  **1 - Segurança dos Dados do Usuário**  **2 - Integridade dos Dados**  **3 - Disponibilidade do Serviço**  **4- Concorrência e Violação de Direitos Autorais**  **5- Feedback do Usuário e Reputação da Marca**  **6 - Regulamentações e Compliance**  **7 - Dependência de Terceiros.**  Efetuado definição dos testes que serão aplicados no decorrer da execução do projeto:   1. **Testes de funcionalidade**: 2. **Testes de precisão**. 3. **Testes de desempenho** 4. **Testes de usabilidade** 5. **Testes de integração** 6. **Testes de compatibilidade**.   Efetuar Documentação.  Efetuar Slides de Apresentação.  Plano de Gerenciamento de Projetos Aprovado | 15/04/2024 | |
|  | Linhas de Base de Custos, Prazo e Escopos Salvas |  | |
| Execução, Monitoramento e Controle | 17/04/2024  23/04/2024  01/05/2024  31/05/2024  31/05/2024 | |

**5. DEFINIÇÃO DE TESTES**

1. **Testes de funcionalidade**: Garantir que todas as funcionalidades do software estejam operando corretamente, incluindo a capacidade de comparar preços entre diferentes produtos e supermercados.
2. **Testes de precisão**: Verificar se os preços exibidos pelo software correspondem aos preços reais dos produtos nos supermercados.
3. **Testes de desempenho**: Avaliar a velocidade do software ao comparar preços em diferentes supermercados e ao lidar com grandes volumes de dados.
4. **Testes de usabilidade**: Certificar-se de que a interface do usuário seja intuitiva e fácil de usar, permitindo aos usuários navegar e comparar preços sem dificuldades.
5. **Testes de integração**: Verificar se o software está integrado corretamente com os sistemas dos supermercados para obter informações atualizadas sobre os preços dos produtos.
6. **Testes de compatibilidade**: Garantir que o software funcione corretamente em diferentes dispositivos e navegadores, para que os usuários possam acessá-lo de qualquer lugar.
7. **Testes de segurança**: Avaliar a segurança do software para proteger os dados dos usuários e garantir que as transações sejam seguras.
8. **Testes de localização**: Verificar se o software é capaz de fornecer informações precisas sobre os supermercados próximos à localização do usuário

**6. METODOLOGIA**

A escolha da metodologia adequada para o desenvolvimento de software é crucial para o sucesso do projeto. Entre as diversas metodologias disponíveis, o Kanban se destaca por sua abordagem visual e flexível, que permite um controle eficiente do fluxo de trabalho. Neste trabalho, exploraremos o conceito do Kanban, como funciona, suas vantagens e como implementá-lo na prototipação de software.

O Kanban é uma metodologia de gestão visual originária do sistema de produção da Toyota. A palavra "Kanban" vem do japonês e significa literalmente "cartão visual" ou "sinalização visual". Na prática, o Kanban utiliza cartões ou quadros visuais para representar as tarefas em um processo de produção ou desenvolvimento.

O funcionamento do Kanban é baseado em alguns princípios fundamentais:

* **Visualização do Trabalho:** O fluxo de trabalho é representado visualmente em um quadro Kanban, dividido em colunas que representam os estágios do processo, como "A Fazer", "Em Andamento" e "Concluído".
* **Limitação do Trabalho em Progresso (WIP)**: Cada coluna do quadro tem um limite de tarefas que podem ser realizadas simultaneamente. Isso evita a sobrecarga de trabalho e ajuda a manter um fluxo constante.
* **Gestão do Fluxo:** As tarefas movem-se pelo quadro de acordo com o progresso, desde a entrada no sistema até a conclusão. Isso permite uma visão clara do fluxo de trabalho e identificação rápida de gargalos ou problemas.
* **Feedback Contínuo:** O Kanban facilita a comunicação entre os membros da equipe, permitindo o feedback contínuo sobre o andamento das tarefas e possíveis ajustes no processo.

**6.1 VANTAGENS DO KANBAN:**

A metodologia Kanban oferece várias vantagens para a prototipação de software, como:

* + **Flexibilidade:** O Kanban é altamente adaptável a diferentes tipos de projetos e equipes. Ele pode ser facilmente personalizado para atender às necessidades específicas de desenvolvimento de software.
  + **Visibilidade do Processo:** A visualização do fluxo de trabalho em um quadro Kanban proporciona uma compreensão clara do estado atual do projeto, facilitando o acompanhamento e a tomada de decisões.
  + **Redução de Desperdícios:** Ao limitar o trabalho em progresso e identificar gargalos, o Kanban ajuda a reduzir desperdícios de tempo e recursos, aumentando a eficiência do processo.
  + **Melhoria Contínua:** A ênfase no feedback contínuo e na melhoria incremental torna o Kanban uma ferramenta poderosa para aprimorar constantemente o processo de desenvolvimento de software.

**6.2 IMPLEMENTAÇÃO**

A implementação do Kanban na prototipação de software envolve os seguintes passos:

* + **Definir o Quadro Kanban:** Crie um quadro Kanban com colunas que representem os estágios do processo de prototipação, como "Ideação", "Desenvolvimento", "**Testes" e "Entrega".**
  + **Identificar Tarefas:** Liste as tarefas específicas necessárias para cada etapa do processo de prototipação, como criar wireframes, desenvolver interfaces, realizar testes de usabilidade, etc.
  + **Estabelecer Limites WIP:** Defina limites para o número máximo de tarefas que podem estar em progresso simultaneamente em cada coluna do quadro Kanban.
  + **Monitorar e Atualizar:** Mantenha o quadro Kanban atualizado, movendo as tarefas conforme progridem no processo e atualizando o status regularmente.
  + **Revisão e Melhoria:** Realize revisões periódicas do processo Kanban para identificar áreas de melhoria e ajustar o fluxo de trabalho conforme necessário.

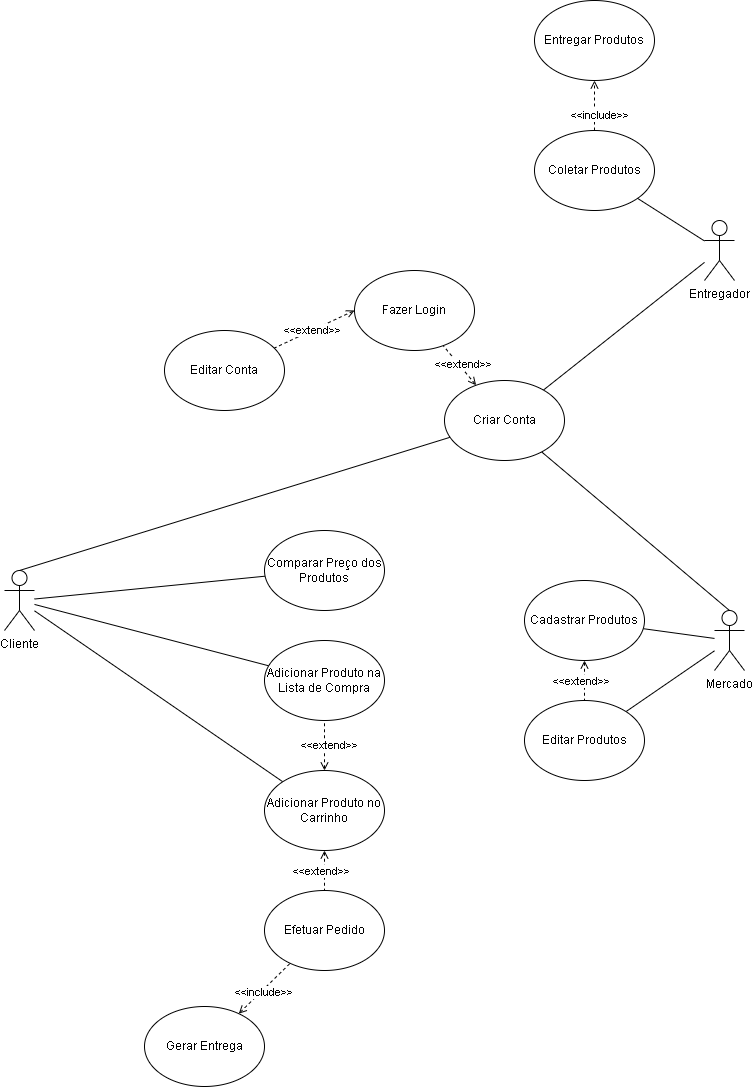
O Kanban é uma metodologia eficaz para a prototipação de software devido à sua abordagem visual, flexibilidade e foco na melhoria contínua. Ao implementar o Kanban, as equipes de desenvolvimento de software podem melhorar a visibilidade do processo, reduzir desperdícios e aumentar a eficiência, resultando em protótipos de software de maior qualidade e maior satisfação do cliente.

**7. ANÁLISE DE RISCOS**

Ao realizar uma análise de risco para um modelo de negócio em gestão e qualidade de software, alguns tópicos importantes a considerar são:

* **Segurança dos Dados do Usuário:**
  + Risco: Vazamento de informações pessoais, como dados de pagamento, endereços e histórico de compras.
  + Mitigação: Implementar protocolos robustos de segurança, como criptografia de dados, autenticação de dois fatores e conformidade com regulamentações de proteção de dados, como GDPR ou LGPD.
* **Integridade dos Dados:**
  + Risco: Manipulação de dados por parte de usuários mal-intencionados, resultando em informações falsas sobre os preços dos produtos.
  + Mitigação: Implementar controles de integridade de dados, como hashes criptográficos e controle de versão de dados. Validação de entrada do usuário e revisão manual de dados suspeitos também são úteis.
* **Disponibilidade do Serviço:**
  + Risco: Interrupção do serviço devido a falhas de servidor, ataques de negação de serviço (DDoS) ou problemas de infraestrutura.
  + Mitigação: Implementar redundância de servidor, balanceamento de carga, monitoramento proativo de desempenho e resposta rápida a incidentes por meio de equipes de operações de TI.
* **Concorrência e Violação de Direitos Autorais:**
  + Risco: Possibilidade de que concorrentes ou proprietários de direitos autorais acusem a aplicação de violação de direitos autorais ou práticas comerciais desleais.
  + Mitigação: Garantir que a aplicação cumpra todas as leis de propriedade intelectual e direitos autorais. Além disso, implementar mecanismos para monitorar e responder a possíveis disputas legais de forma rápida e eficaz.
* **Feedback do Usuário e Reputação da Marca:**
  + Risco: Feedback negativo dos usuários devido a problemas de usabilidade, precisão dos dados ou serviço insatisfatório.
  + Mitigação: Realizar testes de usabilidade frequentes, coletar feedback dos usuários de forma regular e responder rapidamente aos problemas identificados. Investir em uma estratégia de gerenciamento de reputação online para lidar com críticas negativas de forma construtiva.
* **Regulamentações e Compliance:**
  + Risco: Não conformidade com regulamentações locais ou globais relacionadas a comércio eletrônico, proteção de dados e práticas comerciais.
  + Mitigação: Manter-se atualizado com as regulamentações relevantes e garantir que a aplicação esteja em conformidade com todas as leis aplicáveis. Isso pode incluir consultas regulares com advogados especializados em direito digital e compliance.
* **Dependência de Terceiros:**
  + Risco: Problemas causados por falhas ou mudanças nas APIs de supermercados ou fornecedores de dados de preços.
  + Mitigação: Diversificar fontes de dados sempre que possível e ter planos de contingência para lidar com interrupções nos serviços de terceiros. Estabelecer contratos de serviço robustos que definam claramente as responsabilidades e obrigações de todas as partes envolvidas.

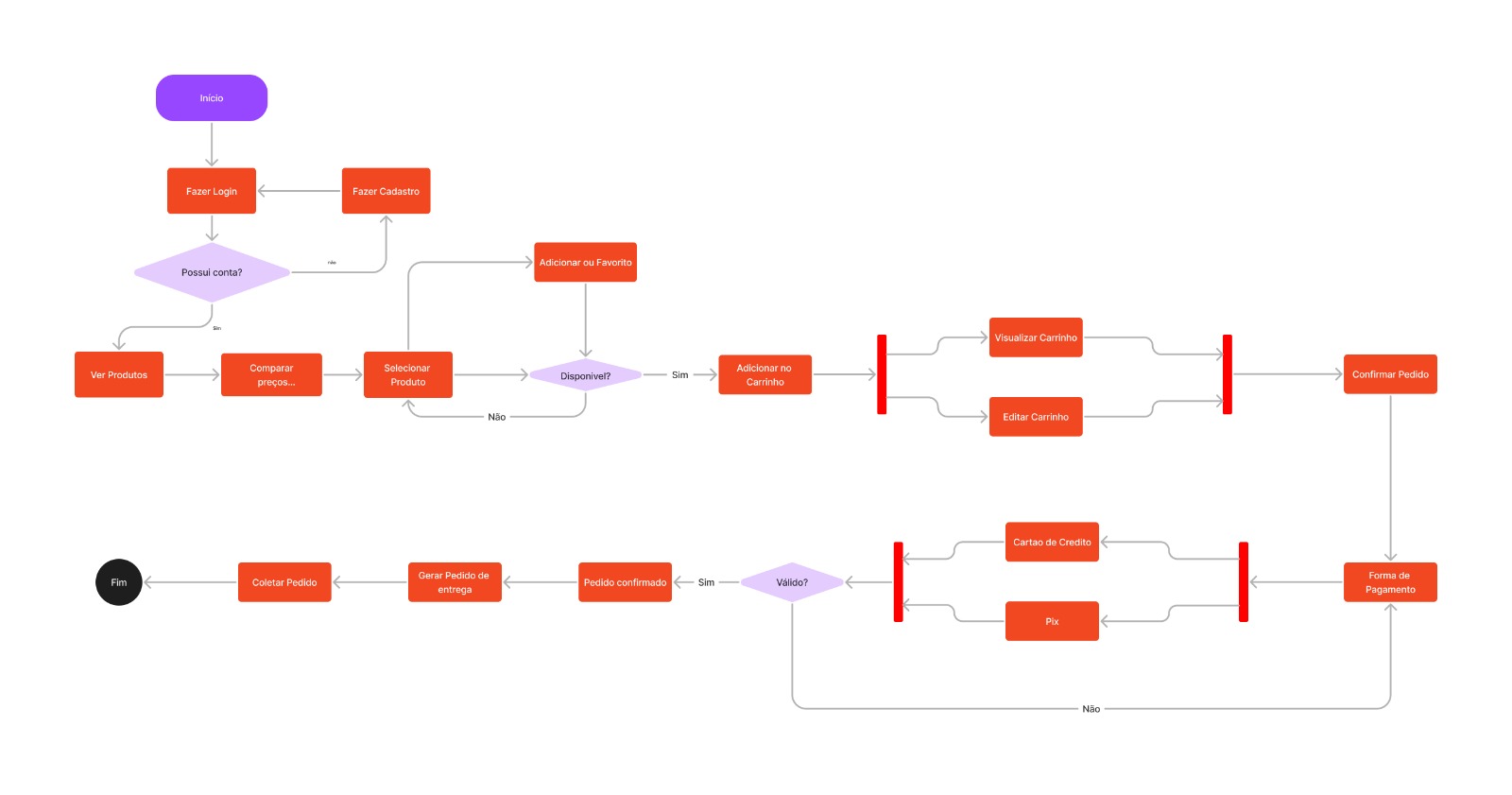
**8. DIAGRAMAS**



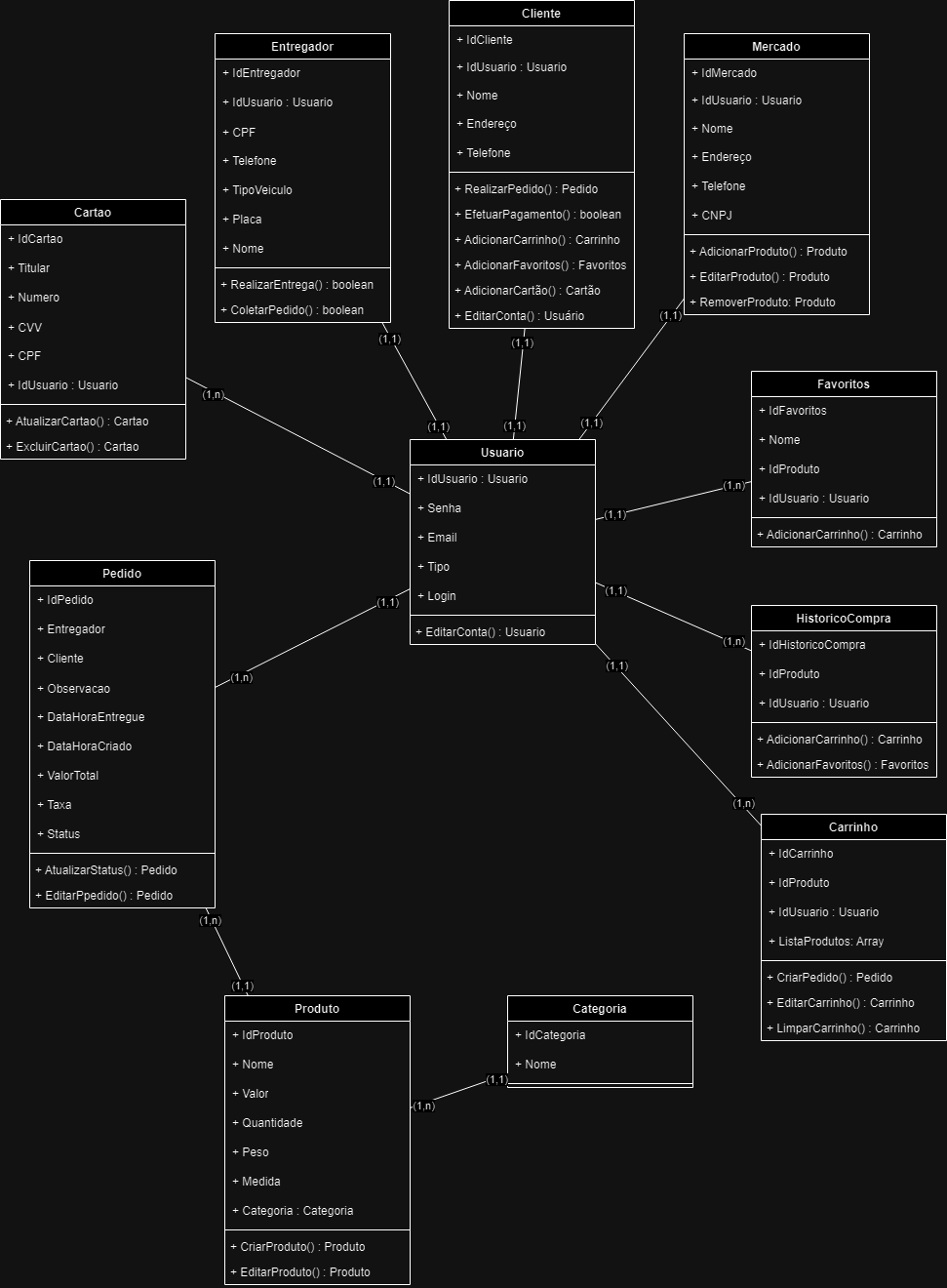
**8.1 CASO DE USO**

* O diagrama de caso de uso é uma ferramenta da Engenharia de Software que descreve as interações entre usuários e o sistema. Ele ajuda a compreender o sistema, facilita a comunicação entre as partes envolvidas, identifica requisitos, valida as expectativas dos usuários e serve como base para o design do sistema

**8.2 DIAGRAMA DE ATIVIDADES**



* Um diagrama de atividades é uma representação gráfica que descreve o fluxo de atividades em um processo ou sistema. Ele é uma ferramenta essencial em engenharia de software e análise de sistemas, usada para modelar e visualizar fluxos de trabalho de maneira clara e intuitiva.
* Visualização de Processos: Representa graficamente o fluxo de atividades.
* Identificação de Decisões: Mostra pontos de decisão no processo.
* Modelagem de Fluxos Paralelos: Representa atividades paralelas.
* Descrição de Sequências: Mostra a sequência lógica das atividades.
* Definição de Início e Fim: Claramente define onde o processo começa e termina.

**8.3 DIAGRAMA DE CLASSES**

* Um diagrama de classes é uma ferramenta essencial na modelagem de sistemas orientados a objetos, particularmente em engenharia de software. Ele serve para representar a estrutura estática de um sistema, mostrando as classes, atributos, métodos e os relacionamentos entre os objetos.

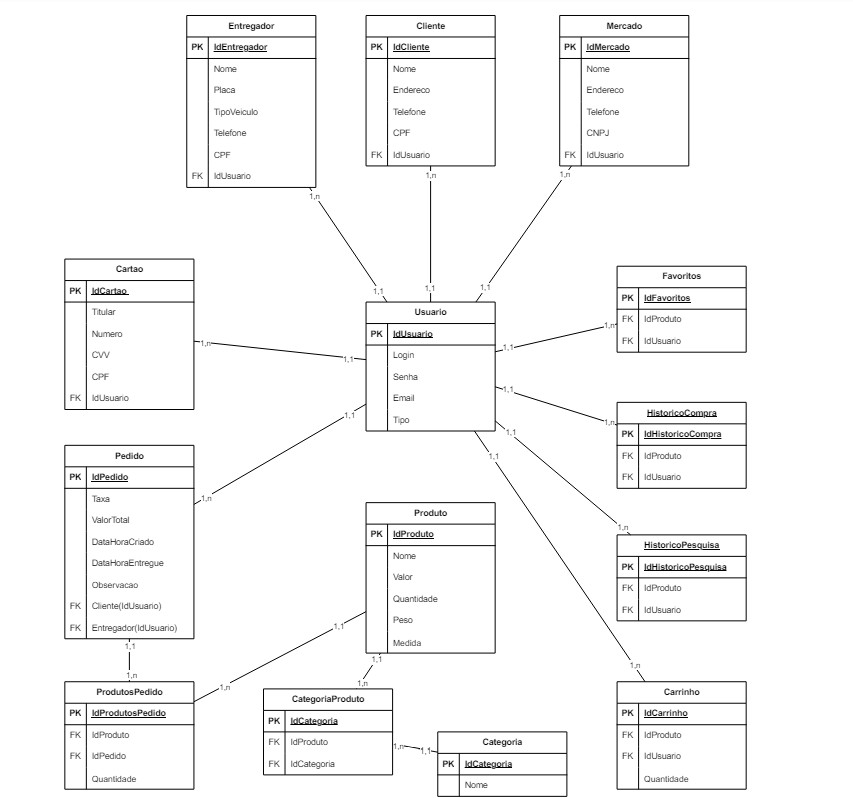
**PRINCIPAIS FUNÇÕES:**

* Modelagem Estrutural: Descreve a arquitetura estática do sistema.
* Visualização: Fornece uma visão clara da estrutura do sistema.
* Documentação: Serve como documentação detalhada para manutenção e desenvolvimento futuro.
* Análise e Design: Ajuda na análise de requisitos e no design do sistema.
* Implementação: Guia os desenvolvedores durante a codificação.

**PRINCIPAIS COMPONENTES:**

* Classes: Representadas por retângulos divididos em nome, atributos e métodos.
* Atributos: Propriedades das classes.
* Métodos (Operações): Funções que a classe pode executar.
* Relacionamentos:
* Associações: Linhas que conectam classes.
* Generalização/Especialização (Herança): Relacionamento entre superclasse e subclasse.
* Agregação e Composição: Indicam relações de parte-todo.
* Visibilidade: Nível de acesso dos atributos e métodos (público, privado, protegido).

**8.4 DIAGRAMA DE MODELO ENTIDADE-RELACIONAMENTO**

****

* O Diagrama de Modelo Entidade-Relacionamento (MER) é uma ferramenta usada na modelagem de banco de dados para representar a estrutura lógica dos dados. Ele ilustra as entidades relevantes de um sistema, os atributos dessas entidades e os relacionamentos entre elas.

**PRINCIPAIS FUNÇÕES:**

* Modelagem de Dados: Representa graficamente os dados e seus relacionamentos em um sistema.
* Visualização Estrutural: Fornece uma visão clara e organizada das entidades e seus vínculos.
* Planejamento de Banco de Dados: Ajuda na concepção e organização de um banco de dados relacional.

**PRINCIPAIS COMPONENTES:**

* Entidades: Representadas por retângulos, são objetos ou conceitos do mundo real com existência independente. Ex: Cliente, Produto.
* Atributos: Propriedades ou características das entidades, representadas por elipses conectadas às entidades. Ex: Nome, ID.
* Relacionamentos: Representados por losangos, mostram como as entidades estão relacionadas entre si. Ex: Compra (relacionamento entre Cliente e Produto).
* Cardinalidade: Indica o número de instâncias de uma entidade que podem se relacionar com instâncias de outra entidade. Ex: 1:1, 1

**9. REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS**

## **9.1 REQUISITOS FUNCIONAIS**

* **RF001:** O sistema deve conter uma interface para o usuário criar sua conta com nome, CPF, e-mail e senha, podendo realizar o login quando já cadastrado;
* **RF002:** O sistema deve conter uma interface de login com opções para cadastro de pessoa física e cadastro para mercados;
* **RF003:** O sistema deve permitir a redefinição de senha de uma conta já cadastrada através do e-mail ou CPF/CNPJ associado à conta;
* **RF004:** O sistema deve registrar uma conta de entregador para cada entregador terceirizado para realizarem as interações com a aplicação;
* **RF005:** O sistema deve possibilitar ao usuário selecionar um ou mais produtos e comparar o valor deles em mercados diferentes;
* **RF006:** O sistema deve conter uma interface para os mercados parceiros cadastrarem suas tabelas (CSV) de produtos e preços;
* **RF007:** O sistema deve conter um sistema de categorização de produtos para facilitar a busca e visualização dos preços em cada mercado;
* **RF008:** O sistema deve conter uma interface do histórico de produtos e mercados vistos recentemente para facilitar o acesso a informações recentes;
* **RF009:** O sistema deve permitir que o usuário exclua os produtos vistos recentemente do histórico;
* **RF010:** O sistema deve fornecer uma forma de visualizar as categorias de produtos nos mercados cadastrados;
* **RF011:** O sistema deve permitir que o usuário visualize as categorias de vários produtos para facilitar a busca e análise de diferentes itens;
* **RF012:** O sistema deve conter uma API para coletar o dado dos produtos, integrável com o sistema dos mercados;
* **RF013:** O sistema deve permitir o usuário adicionar e remover produtos da sua lista de compras;
* **RF014:** O sistema deve permitir o usuário solicitar um entregador do aplicativo para realizar a sua compra nos mercados da sua lista de compras;
* **RF015:**  O sistema deve cobrar uma taxa maior de entrega se os produtos forem de mercados ou estabelecimentos diferentes;
* **RF016:** O sistema deve permitir a lista de compras poder ser gerada semanalmente ou mensalmente baseada na última lista feita;
* **RF017:** O sistema deve permitir o usuário verificar o valor da lista dos seus produtos em vários mercados diferentes;
* **RF018:** O sistema deve permitir o usuário entregador de visualizar os pedidos e atende-los;

**9.2 REQUISITOS NÃO FUNCIONAIS**

* **RNF001:** O sistema deve garantir a proteção dos dados pessoais dos usuários, utilizando medidas de criptografia e protocolos de segurança para evitar acessos não autorizados;
* **RNF002:** O sistema deve ser capaz de lidar com um grande volume de usuários simultâneos e processar as consultas de preços de forma rápida e eficiente, garantindo uma resposta rápida mesmo em horários de pico;
* **RNF003:** O sistema deve estar disponível 24 horas por dia, 7 dias por semana, com um tempo de inatividade mínimo para garantir que os usuários possam acessar as informações a qualquer momento;
* **RNF004:** O sistema deve ser compatível para Web e aparelhos Mobile (Android/IOS);
* **RNF005:** O sistema deve possuir uma interface intuitiva e fácil de usar, permitindo que os usuários naveguem facilmente pelas funcionalidades e encontrem as informações desejadas sem dificuldades;
* **RNF006:** O sistema deve ser projetado para escalabilidade, permitindo que ele cresça conforme o aumento da demanda sem comprometer o desempenho ou a qualidade do serviço;
* **RNF007:** O sistema deve ser facilmente mantido e atualizado, com código limpo e bem documentado, facilitando a incorporação de novas funcionalidades e correções de bugs;
* **RNF008:** Para concluir o cadastro do mercado, o usuário mercado receberá um formulário por e-mail para preenchimento;
* **RNF009:** O sistema deve oferecer suporte técnico eficiente e rápido para resolver quaisquer problemas ou dúvidas dos usuários, garantindo uma experiência positiva e confiável.

**CONCLUSÃO**

Podemos concluir por fim que O SmartShop é um aplicativo de comparação de preços que revoluciona a forma como os consumidores lidam com o mercado varejista. Ao oferecer uma plataforma intuitiva e abrangente para comparar preços em diferentes estabelecimentos, o SmartShop promove transparência e capacita os consumidores a fazerem escolhas mais informadas. Além disso, ao compartilhar dados com empresas parceiras, o SmartShop impulsiona a concorrência saudável e promove a inovação no mercado. Em suma, o SmartShop representa uma ferramenta essencial para consumidores e fornecedores, transformando a experiência de compra e impulsionando a eficiência no mercado varejista.