

**UNISOCIESC**

**MODELOS MÉTODOS E TÉCNICAS DA ENGENHARIA DE  
SOFTWARE**

BIANCA SILVA SANTOS

GUSTAVO LUCAS SCHNEBERGER

LUANA SIQUEIRA RONCHI

**SMART SHOP**



JOINVILLE, 2024

## **RESUMO**

O projeto propõe a criação de um benefício para empresas que se tornem parceiras em um site, visando facilitar a comparação de preços para os consumidores em supermercados. Isso seria realizado através da troca de informações sobre pesquisas de produtos e seus preços, permitindo uma coleta de dados mais eficiente e oferecendo aos consumidores uma ferramenta para tomar decisões de compra mais informadas. A variedade de preços nos supermercados representa um desafio para os consumidores e empresas, mas um aplicativo de comparação de preços poderia resolver esse problema, promovendo a transparência nos preços e incentivando a concorrência saudável.

## **1. REQUISITOS E FUNÇÕES PRINCIPAIS**

### **1.1 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 empresa;

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 conter uma interface para o administrador com acesso total às funcionalidades do sistema;

RF005: O sistema deve possibilitar ao usuário selecionar produtos, clicar e ser redirecionado para os preços dos mesmos;

RF006: O sistema deve conter uma interface para as empresas parceiras cadastrarem suas tabelas 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 empresa;

RF008: O sistema deve permitir que empresas enviem arquivos em Excel com suas tabelas de preços para cadastro dos produtos;

RF009: O sistema deve incluir uma conta de administrador com acesso total a todas as funcionalidades do sistema;

RF010: O sistema deve conter uma interface de produtos e mercados vistos recentemente para facilitar o acesso a informações recentes;

RF011: O sistema deve permitir que o administrador aprove os cadastros de empresas;

RF012: Para concluir o cadastro de empresa, o usuário empresa receberá um formulário por e-mail para preenchimento;

RF013: O sistema deve fornecer uma visualização das categorias de produtos disponíveis nas empresas cadastradas;

RF014: O sistema deve permitir que o usuário exclua os produtos vistos recentemente;

RF015: O sistema deve permitir que o usuário visualize as categorias de vários produtos para facilitar a busca e análise de diferentes itens.

## **1.2 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;

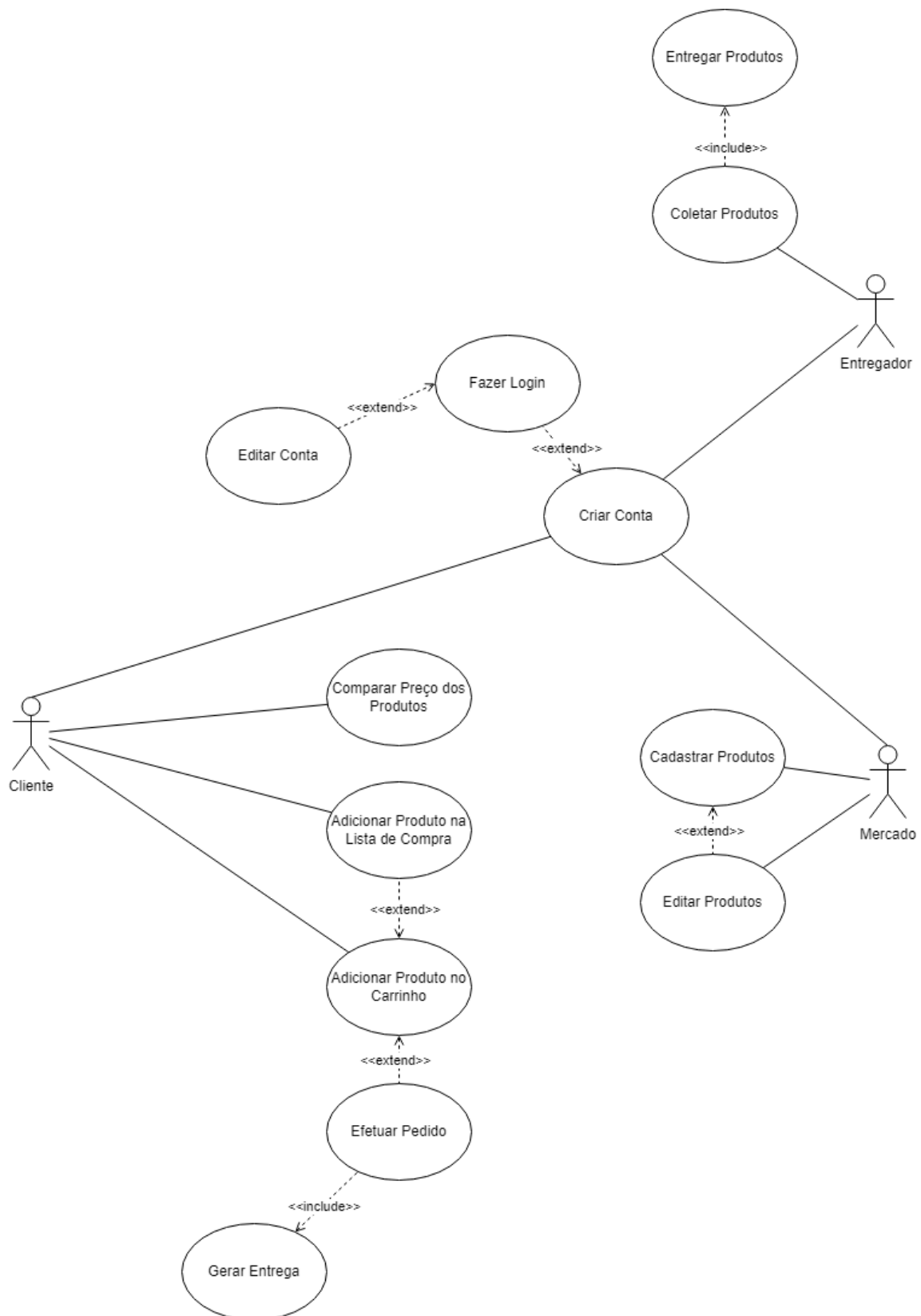
RNF008: O sistema deve respeitar a privacidade dos usuários e seguir as regulamentações da LGPD, garantindo que as informações pessoais sejam tratadas com confidencialidade e não sejam compartilhadas sem consentimento;

RNF009: O sistema deve oferecer suporte técnico eficiente e rápido para resolver quaisquer problemas ou dúvidas dos usuários, garantindo uma experiência positiva e confiável.

## **2.DESCRICÃO DOS OBJETIVOS DOS DIAGRAMA**

### **2.1 DIAGRAMA DE CASO DE USO**

O diagrama de caso de uso é uma ferramenta da Engenharia de Software que ajuda a entender como um sistema será utilizado pelos seus usuários. Ele descreve as interações entre os usuários e o sistema, ajudando na compreensão do sistema, na comunicação entre as partes envolvidas, na identificação de requisitos, na validação das expectativas dos usuários e como base para o design do sistema. Em suma, é uma ferramenta valiosa para garantir que o sistema atenda às necessidades dos usuários finais.

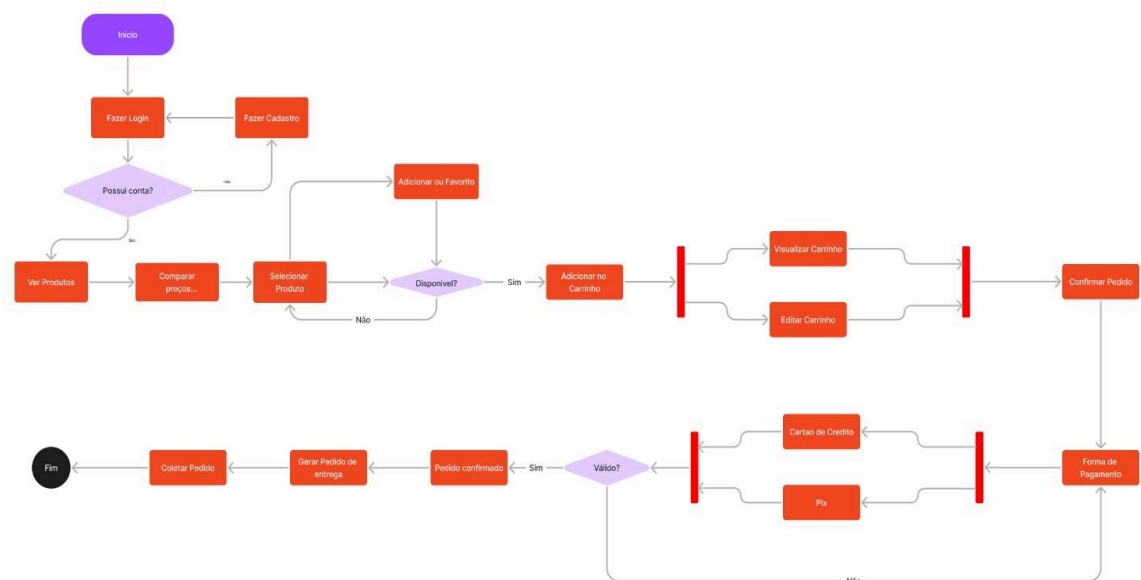




## 2.2 DIAGRAMA DE ATIVIDADES

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.

1. **Visualização de Processos:** Representa graficamente o fluxo de atividades.
2. **Identificação de Decisões:** Mostra pontos de decisão no processo.
3. **Modelagem de Fluxos Paralelos:** Representa atividades paralelas.
4. **Descrição de Sequências:** Mostra a sequência lógica das atividades.
5. **Definição de Início e Fim:** Claramente define onde o processo começa e termina.



## 2.3 DIAGRAMA DE CLASSES

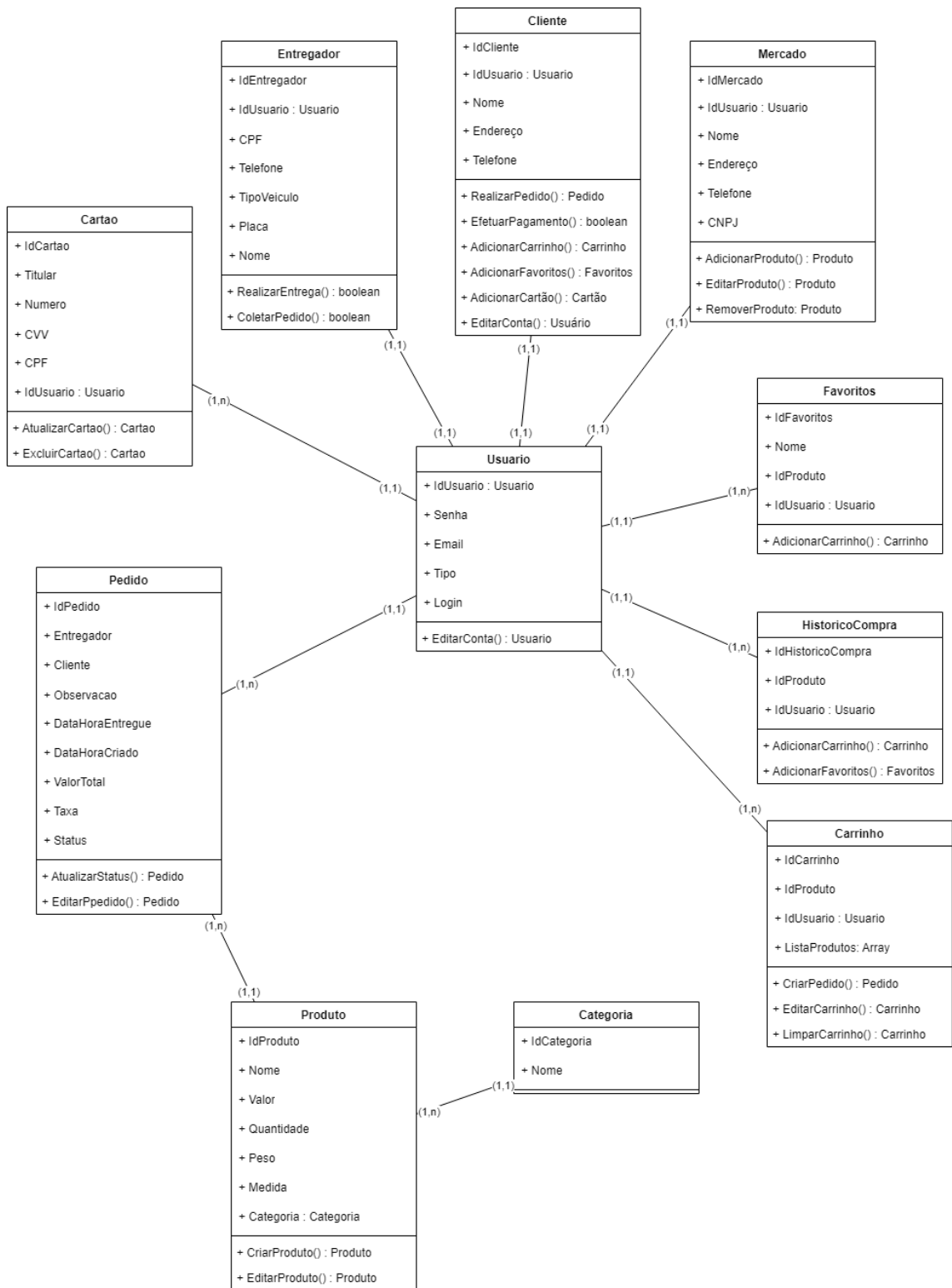
Um diagrama de classes é uma ferramenta essencial na modelagem de sistemas orientados a objetos, particularmente em engenharia de software. Ele serve para representar a estrutura estática de um sistema, mostrando as classes, atributos, métodos e os relacionamentos entre os objetos.

### PRINCIPAIS FUNÇÕES:

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

### PRINCIPAIS COMPONENTES:

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



## 2.4 DIAGRAMA DE MODELO ENTIDADE-RELACIONAMENTO

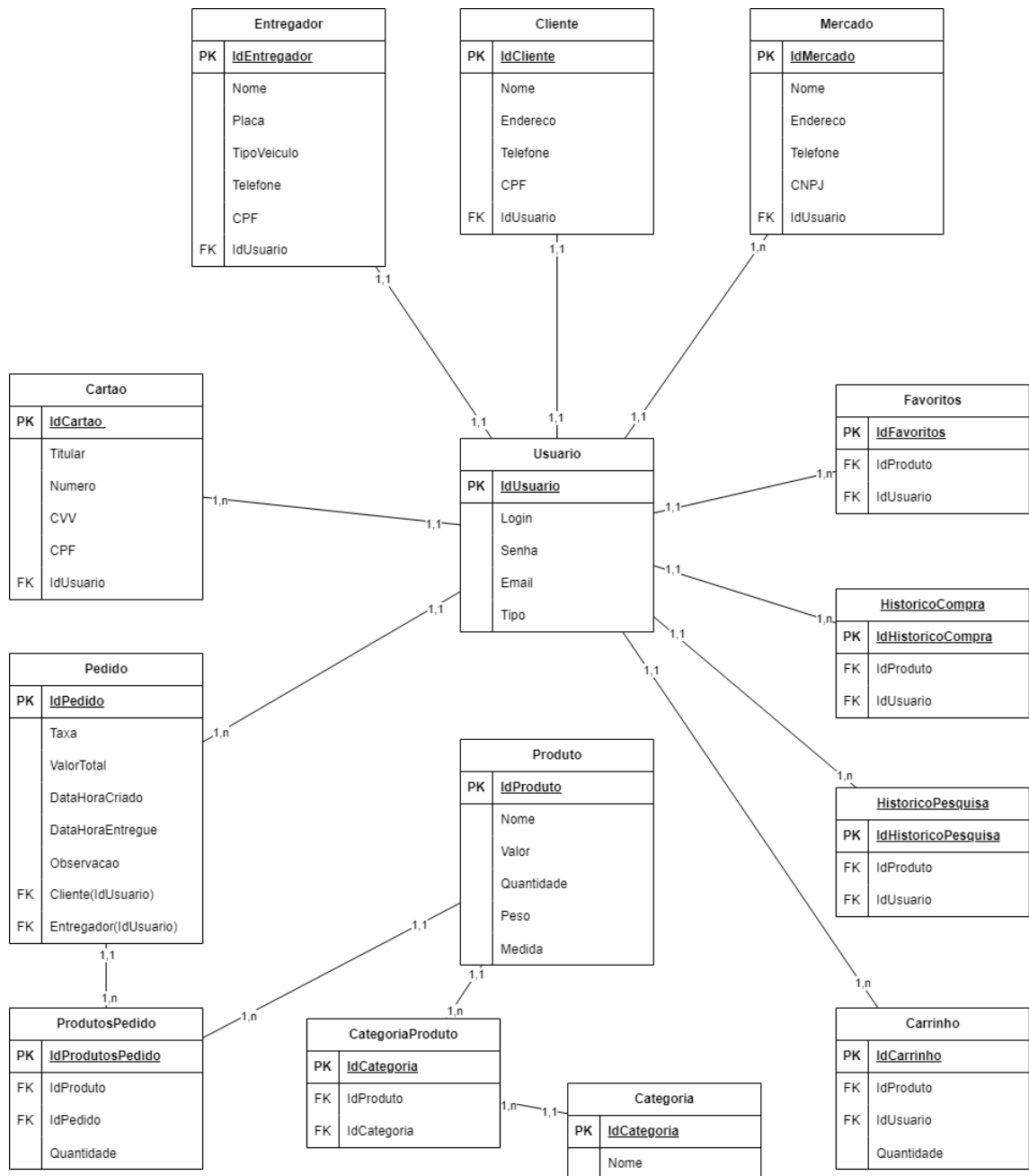
O Diagrama de Modelo Entidade-Relacionamento (MER) é uma ferramenta usada na modelagem de banco de dados para representar a estrutura lógica dos dados. Ele ilustra as entidades relevantes de um sistema, os atributos dessas entidades e os relacionamentos entre elas.

### PRINCIPAIS FUNÇÕES:

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

### PRINCIPAIS COMPONENTES:

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



### **3. HISTÓRIA DO USUÁRIO**

**Usuário 1:** "Como dona de casa gostaria de poder comparar o preço dos produtos sem precisar ir nos mercados toda semana."

**Usuário 2:** "Como uma pessoa que gosta de economizar, gostaria que tivesse um lugar pra não precisar ficar buscando promoções em grupos do Whatsapp e essas coisas."

## **REPOSITÓRIO GITHUB**

Bianca: <https://github.com/Bianca-hue/MMTA3>

Luana: <https://github.com/luanasronchi/A3MMTES202401>

Gustavo: <https://github.com/GusOneOX/A3-F-bio>