



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ  
CAMPUS PEDRO II. TECNÓLOGO EM ANÁLISE EM DESENVOLVIMENTO DE  
SISTEMAS

DISCIPLINA: ENGENHARIA DE SOFTWARE III E IV  
PROFESSOR: WILLYAMS MOREIRA SARAIVA

JOSÉ JACKSON LIMA DE SOUZA JÚNIOR  
JOÃO MARCOS FERREIRA ARAÚJO  
DANIEL TEIXEIRA DA SILVA

“Sistema de E-commerce MEGAMART: Plataforma Especializada na Venda de diversas variações de Produtos”

# 1. Introdução

O aumento das compras online tornou o e-commerce uma plataforma essencial para negócios e consumidores. Este projeto visa desenvolver um sistema de e-commerce que facilite a compra e venda de diversos produtos, inspirado no modelo da Amazon. O documento tem como objetivo registrar os aspectos do desenvolvimento, incluindo um contexto geral, requisitos, arquitetura, design de sistema e tecnologias utilizadas. O sistema abrangerá funcionalidades como cadastro de usuários, listagem de produtos, carrinho de compras, processamento de pagamentos etc.

## 1.1. Visão Geral

O sistema de E-commerce Megamart é uma plataforma digital robusta e intuitiva, projetada para facilitar a compra e venda de diversos produtos online, como eletrônicos, roupas, itens de cafeteria e muito mais. Seu principal objetivo é oferecer uma solução integrada e acessível, onde consumidores e vendedores possam se conectar em um ambiente virtual prático e eficiente. O sistema ainda permite que os usuários naveguem por uma ampla gama de produtos, compare preços, adicione itens ao carrinho e finalize suas compras de forma rápida e segura. A importância deste software reside em sua capacidade de democratizar o acesso ao comércio eletrônico, facilitando a conexão entre pequenos vendedores e um público global, promovendo uma experiência de compra moderna e segura para todos os envolvidos.

## 1.2. Objetivos

Definir os principais objetivos do software, tanto de curto quanto de longo prazo:

- **Facilitar a Compra e Venda de Produtos:** Proporcionar uma plataforma intuitiva que simplifique o processo de compra e venda, conectando consumidores e vendedores de maneira eficiente.
- **Oferecer Navegação Eficiente:** Permitir que usuários naveguem por uma ampla variedade de produtos com facilidade, utilizando filtros e categorias que melhorem a experiência de busca.
- **Garantir Compras Seguras:** Implementar métodos de pagamento seguros e confiáveis, garantindo a proteção das informações dos usuários durante as transações.

## 1.3. Escopo

Dentro do escopo:

- **Cadastro e autenticação de usuários:** Implementar a funcionalidade para que clientes e gerentes se registrem e façam login no sistema.

- **Gestão de produtos e clientes:** O sistema deve permitir que gerentes adicionem, editem e excluam produtos e clientes.
- **Catálogo de produtos:** Exibir uma lista completa de produtos eletrônicos, com opções de filtragem por categoria, preço e nome.
- **Carrinho de compras:** Permitir que os clientes adicionem, removam e atualizem a quantidade de itens no carrinho de compras, com o cálculo do valor total.
- **Finalização de compra e pagamento:** Integração com sistemas de pagamento, permitindo que os clientes finalizem suas compras com métodos de pagamento seguros.
- **Sessão segura:** Manter a sessão dos usuários ativa durante a navegação e permitir logout seguro.

Fora do escopo:

- **Relatórios financeiros detalhados:** O sistema não incluirá funcionalidades avançadas de geração de relatórios financeiros, como balanço contábil, controle de lucros e despesas.
- **Integração completa com redes sociais:** Embora o sistema possa ter notificações, ele não terá uma integração avançada com plataformas de redes sociais para login ou compartilhamento de produtos.
- **Gestão de devoluções e trocas de produtos:** O processo de devolução e troca de produtos não será coberto nesta versão do sistema.
- **Suporte multilíngue e multi-moeda:** O sistema será projetado para suportar apenas o idioma português e o real como moeda para as transações

## 2. Requisitos do Sistema

### 2.1. Requisitos Funcionais

**RF 01: Autenticação de usuários:** O sistema deve permitir que clientes e gerentes façam login utilizando nome de usuário e senha. Deve ter uma rota de autenticação para clientes e outra para gerentes.

- Rota de autenticação para clientes: (/autenticar\_cliente) e para gerentes: (/autenticar.).

**RF 02: Gerenciamento de Sessão:** O sistema deve gerenciar sessões de usuários autenticados, permitindo que o usuário permaneça logado durante a navegação. Deve ter uma funcionalidade de logout para clientes e gerentes:

- Rota de logout para clientes: (/logout\_cliente) e para gerentes: (/logout).

**RF 03: Catálogo de Produtos:** O sistema deve exibir uma lista de produtos disponíveis para compra, com opções de filtragem por categoria e ordenação por nome ou preço.

- Rota para exibir o catálogo de produtos: (/catalogo).

**RF 04: Carrinho de Compras:** O sistema deve permitir que os clientes adicionem produtos ao carrinho, ajustem a quantidade desejada e removam itens. O carrinho deve exibir o valor total dos produtos adicionados.

- Rota para adicionar itens ao carrinho: /adicionar\_carrinho. e remover: (/remover\_item\_carrinho)..
- Rota para exibir o carrinho: (/carrinho).

**RF 05: Finalização de Compra:** O sistema deve permitir a finalização da compra, atualizando o estoque com base nas quantidades adquiridas. O cliente deve poder selecionar o método de pagamento e finalizar a compra com sucesso.

- Rota para finalizar o carrinho: (/finalizar\_carrinho) e concluir a compra: (/finalizar\_compra).
- Rota para selecionar pagamento: (/pagamento).

**RF 06: Gestão de Clientes e Produtos pelo o Gerente:** O gerente deve poder adicionar, excluir e editar clientes e produtos por meio de interfaces de gerenciamento.

- Rota para adicionar: (/adicionar) e para excluir clientes: (/excluir/<int:idcliente>).
- Rota para editar clientes: /editar\_cliente/<int:idcliente>..

**RF 07: Pesquisa de Clientes e Produtos:** O sistema deve permitir a busca de clientes e produtos por meio de campos de pesquisa.

- Rota para pesquisa de clientes: (/pesquisar\_cliente) e para pesquisa de produtos: (/pesquisar\_produto).

## 2.2. Requisitos Não Funcionais

**RNF 01: Segurança:** O sistema deve usar uma ( secret\_key ) para gerenciar as sessões de forma segura.

- A autenticação de usuários deve ser implementada com proteção adequada para garantir a segurança dos dados de login.

**RNF 02 Usabilidade:** O sistema deve fornecer feedback visual e mensagens flash para os usuários, indicando sucesso ou falha nas operações, como login, logout, e manipulação do carrinho de compras.

**RNF 03: Otimização de Acesso ao Banco de Dados:** As operações de leitura e gravação no banco de dados (SQLite) devem ser otimizadas para garantir um desempenho eficiente.

- O sistema deve utilizar consultas específicas e otimizadas para evitar acessos desnecessários e reduzir o tempo de resposta nas interações com o banco de dados.

**RNF 04: Consistência em Tempo Real:** As atualizações de estoque e carrinho de compras devem ser processadas em tempo real, garantindo a consistência dos dados durante todo o processo de compra.

- O sistema deve assegurar que qualquer modificação de estoque, baseada em compras efetuadas, seja imediatamente refletida para evitar inconsistência.

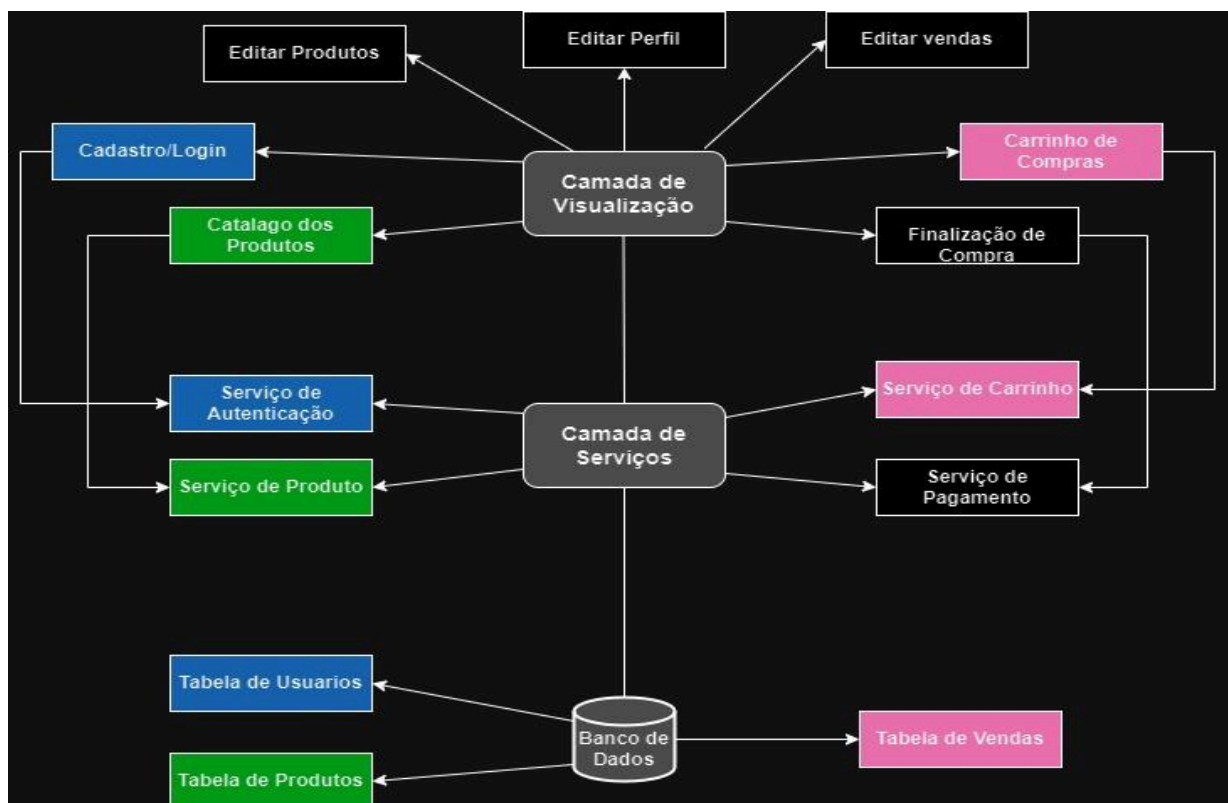
**RNF 05: Escalabilidade:** A arquitetura do sistema deve permitir fácil expansão, com separação clara entre frontend (HTML) e backend (Flask).

- O banco de dados SQLite pode ser substituído por uma solução mais robusta, como PostgreSQL, à medida que o sistema crescer em número de usuários.

**RNF 06: Manutenibilidade:** O sistema deve ser desenvolvido de forma que facilite a manutenção, permitindo a adição e modificação de funcionalidades sem comprometer a estrutura existente.

### 3. Arquitetura do Sistema

- Camada de Apresentação: Lida com rotas para o catálogo de produtos, login, carrinho de compras e finalização de compra.
- Camada de Serviço: Gerência autenticação, operações do carrinho, processamento de pagamentos..
- Camada de Persistência: Contém tabelas de produtos, usuários e vendas.



### 3.1. Visão Geral da Arquitetura

O sistema será desenvolvido com uma arquitetura baseada em camadas, demonstrando a interação entre front-end, back-end e banco de dados. O front-end será o sistema web responsivo desenvolvido em Python, enquanto o back-end será uma API RESTful construída em Flask. O MySQL será utilizado como banco de dados para armazenamento de dados, e a comunicação entre os serviços será realizada por meio de APIs.

- Front-end: Aplicativo web responsivo desenvolvido em Python(HTML).
- Back-end: API RESTful construída em Flask.
- Banco de dados: Banco para armazenamento de dados

## 4. Design do Sistema

### 4.1. Modelagem de Dados

Aqui está a estrutura do banco de dados do projeto do commerce, incluindo os modelos de dados e suas relações.

- **Tabela Usuários:** Armazena informações dos usuários, como ID, nome, e-mail e tipo de cliente (consumidor ou vendedor).
  - **id\_cliente:** Identificador único do cliente.
  - **nome:** Nome completo do cliente.
  - **email:** Endereço de e-mail do cliente.
  - **senha:** Senha do cliente
  - **endereço:** Endereço de entrega do cliente.
- **Tabela Produtos:** Armazena dados sobre os produtos disponíveis, como ID, título, descrição, preço, categoria e quantidade.
  - **id\_produto:** Identificador único do produto.
  - **nome:** Nome do produto.
  - **descricao:** Descrição do produto.
  - **preco:** Preço do produto.
  - **categoria:** Categoria a que o produto pertence.
  - **quantidade\_estoque:** Quantidade disponível em estoque.
- **Tabela Vendas:** Armazena informações sobre pedidos realizados, incluindo ID do pedido, ID do usuário, lista de produtos, e status do pedido.
  - **id\_venda:** Identificador único da venda.
  - **quantidade:** Quantidade de produtos vendidos.
  - **data\_venda:** Data em que a venda foi realizada.
  - **valor\_total:** Valor total da venda.

## 5. Tecnologias Utilizadas

Principais tecnologias e frameworks utilizados para o desenvolvimento do sistema:

- Linguagem de Programação: Python
- Frameworks: Flask (backend), HTML (frontend)
- Banco de Dados: MySQL
- Ferramenta: para ambiente de Desenvolvimento: Visual Studio Code, Git para controle de versão

## 6. Conclusão

O sistema de e-commerce Megamart representa uma solução inovadora e abrangente para o comércio eletrônico, facilitando a compra e venda de uma ampla variedade de produtos, desde eletrônicos até roupas e itens de cafeteria. Através de uma interface intuitiva e funcionalidades robustas, o Megamart não apenas atende às necessidades dos consumidores, mas também empodera pequenos vendedores, democratizando o acesso ao comércio eletrônico. Com a implementação de requisitos funcionais e não funcionais, como autenticação de usuários, segurança nas transações e um catálogo diversificado de produtos, o Megamart se destaca por proporcionar uma experiência de compra moderna, segura