# Documentação Técnica do Projeto

## 1. Capa

**Título do Projeto:** Loja Ponto Com – Marketplace de Compras Online

**Autor(es):** Equipe KekkaiSensen (Preencher com nomes completos)

**Disciplina:** Desenvolvimento Web 1

**Professor:** (Nome do Professor)

**Semestre / Ano:** 2º Semestre / 2025

**Instituição:** Instituto Federal de São Paulo (IFSP)

---

## 2. Sumário



---

## 3. Introdução

## Contexto do Problema

O comércio eletrônico tem crescido exponencialmente, exigindo plataformas robustas que conectem compradores e vendedores de forma eficiente. A necessidade de um sistema que permita não apenas a compra, mas também a venda de produtos por múltiplos usuários (marketplace), apresenta desafios técnicos interessantes em termos de gestão de dados, sessões e segurança.

## Objetivo do Projeto

Desenvolver uma plataforma de e-commerce completa (Marketplace) como projeto prático da disciplina. O sistema visa permitir que usuários comprem produtos de diversos fornecedores e que os próprios usuários possam se cadastrar como vendedores.

## Escopo

O sistema abrange desde o cadastro de usuários e produtos até a finalização de pedidos, incluindo gestão de carrinho, endereços e histórico de compras.

## O que o sistema faz e o que não faz

**Faz:**

- Cadastro e autenticação de usuários (Compradores e Fornecedores).
- Gestão de produtos (CRUD) com upload de imagens.
- Carrinho de compras persistente.
- Simulação de checkout e pagamento.
- Avaliação de produtos.

**Não faz:**

- Processamento real de pagamentos (integração bancária).
- Rastreamento logístico em tempo real (apenas simulação de status).
- Emissão de nota fiscal eletrônica.

---

# 4. Visão Geral do Sistema

## Descrição Geral da Solução

A **Loja Ponto Com** é uma aplicação web desenvolvida com tecnologias nativas (Vanilla), focando no aprendizado dos fundamentos do desenvolvimento web. A arquitetura segue o padrão de renderização no servidor (SSR) com PHP, utilizando banco de dados relacional MySQL.

## Tecnologias Utilizadas

- **Frontend:** HTML5, CSS3 (Vanilla), JavaScript (ES6+).
- **Backend:** PHP 7.4+ (Vanilla).
- **Banco de Dados:** MySQL 5.7+.
- **Ferramentas:** XAMPP/WAMP, Git, VSCode, Cypress.

## Justificativas de Escolhas Técnicas

A escolha por não utilizar frameworks (como Laravel ou React) foi deliberada para garantir o domínio dos conceitos fundamentais da linguagem PHP e do funcionamento do protocolo HTTP, sessões e manipulação direta do DOM e banco de dados.

---

# 5. Requisitos do Sistema

## 5.1 Requisitos Funcionais

**Para Clientes:**

- **RF01:** O sistema deve permitir o cadastro e login de usuários.
- **RF02:** O sistema deve permitir a busca de produtos por nome.
- **RF03:** O usuário deve poder adicionar, remover e alterar quantidades de produtos no carrinho.
- **RF04:** O usuário deve poder cadastrar e gerenciar endereços de entrega.

- **RF05:** O sistema deve permitir a finalização de pedidos com escolha de forma de pagamento.
- **RF06:** O usuário deve poder visualizar seu histórico de compras.
- **RF07:** O usuário deve poder avaliar produtos comprados.

**Para Fornecedores:**

- **RF08:** O fornecedor deve poder cadastrar, editar e excluir produtos.
- **RF09:** O sistema deve permitir o upload de múltiplas imagens por produto.
- **RF10:** O fornecedor deve poder salvar produtos como rascunho.
- **RF11:** O fornecedor deve poder visualizar suas vendas.

### 5.2 Requisitos Não Funcionais

- **RNF01 - Desempenho:** O carregamento das páginas principais deve ocorrer em menos de 2 segundos.
- **RNF02 - Segurança:** As senhas devem ser armazenadas com hash criptográfico. O sistema deve prevenir SQL Injection.
- **RNF03 - Usabilidade:** A interface deve ser responsiva, adaptando-se a dispositivos móveis e desktops.
- **RNF04 - Manutenibilidade:** O código deve ser organizado em pastas lógicas (src, assets, banco de dados).

---

## 6. Modelagem

Os diagramas UML foram elaborados para guiar o desenvolvimento e documentar a estrutura do sistema.

*(Insira aqui as imagens dos diagramas localizados na pasta `Artefatos/Diagramas`)*

- **Diagrama de Caso de Uso:** `Artefatos/Diagramas/Casos de Uso/UC - Imagem.png`
- **Diagrama de Classes/ER:** `Artefatos/Diagramas/Banco de dados/BD - imagem.png`
- **Diagrama de Sequência:** `Artefatos/Diagramas/Sequência/Sequência - Imagem.png`
- **Diagrama de Estado:** `Artefatos/Diagramas/Estado/Estado - Imagem.png`

# 7. Arquitetura e Organização do Código

## 7.1 Estrutura de Pastas

- `Telas/src/` : Contém o código-fonte das páginas PHP e HTML (Views e Controllers misturados, padrão Page Controller).
- `Telas/assets/` : Armazena recursos estáticos como folhas de estilo (CSS), imagens e scripts JavaScript globais.
- `Telas/Banco de dados/` : Contém scripts de conexão, lógica de processamento (Models/Services) e o dump SQL.
- `Telas/Artefatos/` : Documentação do projeto, diagramas e slides.
- `Telas/Testes/` : Testes automatizados E2E com Cypress.

## 7.2 Descrição dos Módulos

- **Conexão ( `conexao.php` ):** Responsável por estabelecer a comunicação com o banco de dados MySQL usando PDO.
- **Autenticação ( `processa_login.php` , `logout.php` ):** Gerencia o ciclo de vida da sessão do usuário.
- **Catálogo ( `index.php` , `buscar.php` ):** Exibe os produtos e processa filtros de busca.
- **Carrinho ( `tela_carrinho.php` , `sincronizar_carrinho.php` ):** Gerencia o estado do carrinho de compras.

## 7.3 Fluxo Interno do Programa

O sistema funciona baseado em requisições HTTP. O usuário acessa uma página `.php` (ex: `index.php` ), o servidor processa a lógica (consulta banco, verifica sessão) e devolve o HTML renderizado. Ações de formulário (POST) são enviadas para scripts de processamento (ex: `processa_login.php` ) que executam a ação e redirecionam o usuário.

---

# 8. Guia de Instalação e Execução

### Dependências

- PHP 7.4 ou superior.
- MySQL 5.7 ou superior.
- Servidor Web (Apache/Nginx).

### Passo a Passo

1. **Clone o repositório:**

   ```
   git clone https://github.com/KekkaiSensen/Projeto-de-desenvolvimento-web-.
   ```

2. **Configure o Banco de Dados:**
   - Importe o arquivo `Banco de dados/bancodadosteste.sql` no seu SGBD.
   - Ajuste as credenciais em `Banco de dados/conexao.php`.
3. **Execute o Servidor:**
   - Mova a pasta para o diretório do servidor web (ex: `htdocs`) ou use o servidor embutido do PHP:

     ```
     cd src
     php -S localhost:8000
     ```

4. **Acesse:** Abra `http://localhost:8000/index.php` no navegador.

---

## 9. Detalhamento Técnico do Código

### Conexão com Banco de Dados ( `conexao.php` )

Utilizamos a biblioteca **PDO** para garantir segurança e portabilidade. O uso de `PDO::ERRMODE_EXCEPTION` facilita a depuração de erros SQL.

```
try {
    $pdo = new PDO($dsn, $username, $password, $options);
} catch (\PDOException $e) {
```

```
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
```

### Listagem de Produtos ( `index.php` )

A consulta SQL utiliza `LEFT JOIN` para trazer a média de avaliações junto com os dados do produto, otimizando a performance ao evitar múltiplas consultas (N+1 problem).

```
SELECT p.*, AVG(a.nota) as media_avaliacoes
FROM produtos p
LEFT JOIN avaliacoes a ON p.id = a.produto_id
WHERE p.status = 'ativo'
GROUP BY p.id
```

---

## 10. Testes

### Estratégia de Testes

Foram utilizados testes **End-to-End (E2E)** com a ferramenta **Cypress** para simular a interação real do usuário com o sistema. Isso garante que os fluxos críticos (compra, login, cadastro) estejam funcionando conforme o esperado.

### Casos de Teste Principais

- Fluxo de Cadastro de Usuário.
- Fluxo de Login e Logout.
- Adição de produtos ao carrinho e finalização de compra.
- Cadastro de produto pelo fornecedor.

*(Insira aqui prints do Cypress rodando ou o resultado do terminal)*

---

## 11. Resultados

O sistema atendeu aos objetivos propostos, entregando uma plataforma funcional e responsiva.

*(Insira aqui capturas de tela do sistema funcionando, como a Home Page, Carrinho e Painel do Fornecedor. Você pode usar as imagens da pasta `Artefatos/Readme/Fotos do readme/` )*

---

## 12. Conclusão

O desenvolvimento da **Loja Ponto Com** permitiu a aplicação prática dos conceitos de desenvolvimento web. As principais dificuldades encontradas foram relacionadas ao gerenciamento de estado (sessões) e à modelagem complexa do banco de dados para um marketplace. Futuras melhorias incluem a implementação de uma API REST para desacoplar o frontend e a integração com gateways de pagamento reais.

---

## 13. Referências


1. **PHP Documentation**. Disponível em: https://www.php.net/docs.php.
2. **MySQL Reference Manual**. Disponível em: https://dev.mysql.com/doc/.
3. **MDN Web Docs (HTML, CSS, JS)**. Disponível em: https://developer.mozilla.org/.
4. **Cypress Documentation**. Disponível em: https://docs.cypress.io/.


---