**DOCUMENTAÇÃO DE REQUISITO**

**Sistema de Gerenciamento**

**Nomes dos membros da equipe:**

**Front-End:** Igor Levandoski Mota & Trinity Domingues do Nascimento

**Back-End:** Igor Levandoski Mota & Trinity Domingues do Nascimento

**Banco de Dados:** Igor Levandoski Mota & Trinity Domingues do Nascimento

**Engenharia de Software/ Documentação:** Igor Levandoski Mota & Trinity Domingues do Nascimento

**Atores:**

**Usuário da interface:** insere, deleta e atualiza produtos.

**Usuário Cadastrado:** dado a solicitação do professor não há necessidade de cadastro.

**Visitante Não Cadastrado:** não definido.

**Sumário**

[1. **INTRODUÇÃO** 3](#_Toc178007592)

[Descrição Geral: 3](#_Toc178007593)

[2. **Requisitos Técnicos:** 3](#_Toc178007594)

[Endpoints Implementados: 3](#_Toc178007595)

[Para Produtos (CRUD completo): 3](#_Toc178007596)

[Para Logs: 4](#_Toc178007597)

[**3.** **Arquivos do Sistema** 4](#_Toc178007598)

[index.html 5](#_Toc178007599)

[produtos.php 5](#_Toc178007600)

[logs.php 5](#_Toc178007601)

[database.db 5](#_Toc178007602)

[setup.php 5](#_Toc178007603)

[**4. Dificuldades Enfrentadas com o XAMPP 5**](#_Toc178007604)

[**5. Testes de CRUD no Postman 6**](#_Toc178007605)

[**6. CRUD realizados no Postman: 6**](#_Toc178007606)

[Teste de Inserção (POST) 6](#_Toc178007607)

[Teste de Listagem (GET) 7](#_Toc178007608)

[Teste de Listagem (GET LOGS) 8](#_Toc178007609)

[Teste de Listagem (GET Por ID) 9](#_Toc178007610)

[Teste de Atualização (PUT) 10](#_Toc178007611)

[Teste de Exclusão (DELETE) 11](#_Toc178007612)

[Teste de ERRO 400 12](#_Toc178007613)

[Teste de ERRO 404 13](#_Toc178007614)

[DataBase (LOGS) 14](#_Toc178007615)

[DataBase (Produtos) 15](#_Toc178007616)

[**7. Conclusão 16**](#_Toc178007617)

# INTRODUÇÃO

## **Descrição Geral:**

Este projeto foi desenvolvido em dupla para atender às necessidades de uma empresa que está criando um sistema de gerenciamento de produtos. O objetivo da aplicação é permitir o cadastro, atualização, listagem e exclusão de produtos no banco de dados, além de registrar todas as operações de exclusão e atualização em uma tabela de logs.

A aplicação conta com dois tipos de endpoints: um para gerenciar produtos e outro para registrar e acessar os logs das operações realizadas. Todos os dados de entrada são validados no momento da inserção ou atualização dos produtos, garantindo que os campos obrigatórios estejam preenchidos e que os valores sejam válidos (por exemplo, o preço deve ser positivo e o estoque deve ser maior ou igual a zero).

# Requisitos Técnicos:

* O banco de dados utilizado é o SQLite, com suporte para a criação das tabelas através do SQLite Database Browser.
* Duas classes principais foram implementadas:
  1. **Produto**: Representa os produtos, com atributos como id, nome, descricao, preco, estoque, userInsert (usuário que realizou a operação) e data\_hora (registro do momento da operação).
  2. **Log**: Registra as operações realizadas nos produtos, com atributos como id, acao, data\_hora, produto\_id e userInsert.

## **Endpoints Implementados:**

### **Para Produtos (CRUD completo):**

GET /produtos: Retorna todos os produtos.

GET /produtos/{id}: Retorna um produto específico com base no ID.

POST /produtos: Cria um novo produto, com validação de nome, preço e estoque.

PUT /produtos/{id}: Atualiza os dados de um produto existente, com validação de campos.

DELETE /produtos/{id}: Exclui um produto específico.

### **Para Logs:**

GET /logs: Retorna todos os logs de operações realizadas nos produtos.

O sistema também garante a integridade dos dados ao validar o nome (mínimo de 3 caracteres), preço (valor positivo) e estoque (inteiro maior ou igual a zero) de cada produto cadastrado ou atualizado. As operações são registradas nos logs, permitindo rastrear quem fez a alteração e quando ela ocorreu.

# Arquivos do Sistema

## **index.html**

O arquivo HTML responsável pela interface inicial do sistema. Ele serve como ponto de entrada para o usuário, permitindo o cadastro e visualização de produtos.

## **produtos.php**

Este arquivo contém as funções PHP responsáveis pelo gerenciamento de produtos. Aqui estão os métodos que lidam com as operações de CRUD (inserção, listagem, atualização e exclusão de produtos) e a interação com o banco de dados SQLite.

## **logs.php**

Responsável pelo registro das operações no sistema, o arquivo logs.php grava cada ação executada (como inserção, atualização e exclusão de produtos) na tabela Log do banco de dados. Cada log contém informações sobre o tipo de operação, o horário e o usuário responsável pela ação.

## **database.db**

O banco de dados SQLite que armazena todas as informações do sistema, incluindo os produtos e os logs de operações. Este arquivo é acessado pelos scripts PHP para manipulação dos dados.

## **setup.php**

O arquivo setup.php é responsável por inicializar o banco de dados. Ele cria as tabelas Produto e Log caso elas ainda não existam. Também define a conexão com o banco de dados, permitindo que o sistema funcione corretamente.

# Dificuldades Enfrentadas com o XAMPP

Durante o desenvolvimento, uma das principais dificuldades encontradas foi a configuração do XAMPP para utilizar o SQLite. Foi necessário habilitar a extensão sqlite3 no PHP, o que gerou alguns desafios, como ajustar o arquivo php.ini corretamente. Outro ponto que demandou tempo foi garantir que as permissões de acesso ao banco de dados estivessem corretamente configuradas, pois isso impactava a escrita e leitura dos dados.

Além disso, houve momentos em que o servidor local apresentava dificuldades de conexão, exigindo reinicializações do Apache para que o sistema voltasse a operar corretamente.

# Testes de CRUD no Postman

Os testes dos endpoints foram realizados utilizando o Postman, garantindo que as operações de **Create**, **Read**, **Update** e **Delete** estivessem funcionando corretamente. Para cada teste realizado, foram verificadas as respostas do servidor, os logs gerados e a consistência dos dados no banco de dados.

Abaixo estão os espaços reservados para a inserção das imagens dos testes de CRUD realizados no Postman e DataBase:

# CRUD realizados no Postman DataBase:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de Inserção (POST)**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## **Teste de Listagem (GET)**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de Listagem (GET LOGS)**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de Listagem (GET Por ID)**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de Atualização (PUT)**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de Exclusão (DELETE)**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de ERRO 400**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## **Teste de ERRO 404**

**Tabela

Descrição gerada automaticamente com confiança média**

## **DataBase (LOGS)**

**Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente**

## **DataBase (Produtos)**

# Conclusão

O desenvolvimento do sistema de gerenciamento de produtos utilizando PHP e SQLite proporcionou um aprendizado valioso em diversas áreas, desde a criação de endpoints para operações CRUD até a integração com o banco de dados. A configuração inicial do ambiente com o XAMPP e a ativação do SQLite foram os maiores desafios enfrentados, especialmente na fase de ajustes do arquivo php.ini e resolução de problemas de conexão. Contudo, uma vez que a configuração foi acertada, o sistema se mostrou funcional, permitindo o cadastro, edição, exclusão e visualização de produtos de forma eficiente.

Os testes realizados no Postman demonstraram que os endpoints funcionam conforme o esperado, com os dados sendo inseridos e manipulados corretamente no banco de dados. O registro das operações nos logs oferece maior controle sobre as ações realizadas no sistema, contribuindo para a rastreabilidade das atividades dos usuários.

Esse projeto não só permitiu o desenvolvimento técnico, mas também reforçou a importância de uma documentação clara e de testes adequados para garantir o bom funcionamento de sistemas web. Assim, o sistema de gerenciamento de produtos está preparado para ser utilizado e expandido conforme as necessidades da empresa, com a possibilidade de melhorias futuras, como a responsividade do layout e a otimização da interface de usuário.