



**ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA  
SISTEMA DE CONTROLE DE ESTOQUES FARMACÊUTICOS**

**SISTEMA FARM+**

**AUTOR(A): MARIA RAQUEL SILVA LIMA**

**PROFESSOR(ES) ORIENTADOR(ES):  
CARLOS RENAN MOREIRA  
HUGO NATHAN BARBOSA RÉGIS**

**RUSSAS-CE  
2024**

## 1. INTRODUÇÃO

1.1. O sistema proposto, Farm+, oferece um controle de estoques para estabelecimentos farmacêuticos. A ideia central é fazer a organização de controle de produtos e funcionários do estabelecimento. A partir dessas informações será possível controlar também, a inclusão, exclusão, visualização e alteração de produtos e funcionários pelo administrador do sistema.

### 1.2. Visão geral do documento

#### 1.2.1. O que são requisitos funcionais

1.2.1.1. Os requisitos funcionais são aqueles que visam atingir a solução dos problemas do usuário. Um requisito funcional visa atender as necessidades dos usuários, concretizando essas necessidades com funções e serviços.

#### 1.2.2. O que são requisitos não funcionais

1.2.2.1. Os requisitos não funcionais são princípios utilizados para a garantia de eficiência, usabilidade e qualidade de um sistema. Imprimem as limitações e restrições do sistema, não possuem impacto nas funcionalidades do projeto.

#### 1.2.3. Descrição dos usuários

1.2.3.1. **Administrador:** Executarão o controle de produtos (adicionar, editar, excluir e listar produtos) e funcionários (Adicionar, remover, atualizar dados e listar funcionários);

1.2.3.2. **Funcionários:** Executarão o controle de produtos (adicionar, editar, excluir e listar produtos)

## 2. REQUISITOS FUNCIONAIS

### 2.1. Acesso e cadastramento de funcionários[RF01]

2.1.1. O sistema deve permitir o acesso e cadastramento de funcionários por meio das credenciais exigidas.

## **2.2. Diferentes níveis de acesso[RF02]**

2.2.1. O sistema deve permitir acesso, controle e listagem de funcionários e produtos ao administrador;

2.2.2. O sistema deve permitir acesso, controle e listagem de produtos aos funcionários.

## **2.3. Cadastro de produtos[RF03]**

2.3.1. O sistema deve permitir que o administrador e os funcionários adicionem produtos.

## **2.4. Remoção de produtos[RF04]**

2.4.1. O sistema deve permitir que o administrador e os funcionários removam produtos.

## **2.5. Edição de produtos[RF05]**

2.5.1. O sistema deve permitir que o administrador e os funcionários editem os produtos cadastrados;

## **2.6. Cadastro de funcionários[RF06]**

2.6.1. O sistema deve permitir que o administrador adicione funcionários.

## **2.7. Remoção de funcionários[RF07]**

2.7.1. O sistema deve permitir que o administrador remova funcionários.

## **2.8. Atualização de dados dos funcionários[RF08]**

2.8.1. O sistema deve permitir que o administrador atualize os dados de funcionários cadastrados.

## **2.9. O sistema deve permitir a geração de relatório de categoria de produto mais registrado[RF09]**

2.9.1. O sistema deve permitir a geração de relatório contendo a categoria de produto mais registrada.

# **3. REQUISITOS NÃO FUNCIONAIS**

## **3.1. Usabilidade**

### **3.1.1. Responsividade**

3.1.1.1. O sistema deve ser responsivo, deve ajustar automaticamente sua interface e funcionalidades para proporcionar uma experiência de uso otimizada em diversos dispositivos, incluindo desktops, tablets e smartphones;

3.1.1.2. Nesse sistema foi utilizado o framework Bootstrap, com o objetivo de desenvolver um sistema responsivo e moderno.

### **3.1.2. Validação de campos**

3.1.2.1. O sistema deve implementar validação de campos de entrada de dados, garantindo que todos os dados fornecidos pelos usuários sejam precisos, completos e estejam no formato correto antes de serem processados ou armazenados;

3.1.2.2. Nesse sistema foi utilizado as bibliotecas Parsley.js e SweetAlert, com o objetivo de melhorar a experiência do usuário e garantir que os dados enviados sejam válidos.

### **3.1.3. Mensagens de aviso**

3.1.3.1. O sistema fornece mensagens de aviso claras, consistentes e padronizadas para informar os usuários sobre ações que necessitam de atenção, possíveis erros, ou resultados de operações que podem impactar sua experiência.

## **3.2. Banco de dados MySQL**

3.2.1. O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto. Suas principais vantagens são sua organização relacional, sendo possível organizar dados relacionando tabelas entre si através de chaves primárias e estrangeiras, utiliza também a linguagem SQL (Structured Query Language) para gerenciar e manipular dados. Possui um sistema multiusuário, permitindo a gestão de acessos e permissões. Além de alta performance e eficiência, compatibilidade com plataformas. Em resumo, o MySQL é uma solução robusta e confiável para gerenciamento de dados, amplamente utilizada em aplicações web e empresariais.

## **3.3. Linguagem PHP**

**3.3.1.** O PHP (Hypertext Preprocessor) é uma linguagem de programação de código aberto usada principalmente para desenvolvimento web. A linguagem é executada do lado do servidor, gerando um HTML dinâmico, possuindo integração com o mesmo, é possível ser embutido no próprio código HTML. Além da simplicidade e flexibilidade, o PHP suporta diversos bancos de dados, como MySQL, PostgreSQL e SQLite. Em resumo, o PHP é uma escolha popular para desenvolvedores que precisam criar aplicações web rapidamente e de maneira eficiente.

### **3.4. Frameworks**

#### **3.4.1. Bootstrap**

**3.4.1.1.** Como citado anteriormente, esse sistema utiliza o framework bootstrap, em sua versão 5.2.3, com o objetivo de tornar a interface do sistema mais responsiva e moderna. O framework possui componentes pré-estilizados, incluindo botões, formulários, navegação, modais, e outros componentes que são facilmente customizáveis. É uma ferramenta utilizada para desenvolver rapidamente interfaces web.

### **3.5. Bibliotecas**

#### **3.5.1. Parsley.js**

**3.5.1.1.** Parsley.js é uma biblioteca JavaScript utilizada para validação de formulários do lado do cliente. Foi utilizada com o objetivo de validar campos, ajudando o cliente a passar as informações na formatação correta e não mandar informações vazias.

#### **3.5.2. SweetAlert**

**3.5.2.1.** A biblioteca SweetAlert é uma ferramenta de JavaScript utilizada para criar alertas personalizados em aplicações web. Foi utilizado com o objetivo de criar alertas claros e objetivos

para o cliente, deixando explícito o erro cometido e a forma de corrigi-lo.

## **4. REQUISITOS MÍNIMOS PARA USO**

### **4.1. Requisitos mínimos para uso**

#### **4.1.1. Sistema Operacional**

- 4.1.1.1. Windows 10;
- 4.1.1.2. macOS 10.13 ou superior;
- 4.1.1.3. Linux(qualquer versão moderna);

#### **4.1.2. Navegador Web**

- 4.1.2.1. Google Chrome(Última versão);
- 4.1.2.2. Mozilla Firefox(Última versão);
- 4.1.2.3. Microsoft Edge(Última versão);
- 4.1.2.4. Safari(Última versão);