

# DOCUMENTAÇÃO DO SISTEMA - LOJA DE AVIAMENTOS

## SAEP 2025 - Sistema de Controle de Estoque

---

### 1. REQUISITOS FUNCIONAIS

#### RF001 - Autenticação de Usuários

**Descrição:** O sistema deve permitir que usuários façam login com email e senha. **Critérios de Aceitação:** - O usuário deve informar email e senha válidos - Em caso de erro, deve exibir mensagem informativa - Após login bem-sucedido, redirecionar para o dashboard - Deve existir opção de logout

#### RF002 - Gestão de Usuários

**Descrição:** O sistema deve permitir o cadastro e gerenciamento de usuários. **Critérios de Aceitação:** - Cadastrar novos usuários com nome, email, senha e perfil - Listar usuários cadastrados - Editar dados de usuários existentes - Excluir usuários do sistema - Validar unicidade do email

#### RF003 - Cadastro de Categorias

**Descrição:** O sistema deve permitir o cadastro de categorias de produtos. **Critérios de Aceitação:** - Cadastrar categoria com nome e descrição - Listar categorias existentes - Editar categorias - Excluir categorias (se não houver produtos vinculados)

#### RF004 - Cadastro de Produtos

**Descrição:** O sistema deve permitir o cadastro completo de produtos. **Critérios de Aceitação:** - Cadastrar produto com nome, descrição, preço, categoria, estoque atual e estoque mínimo - Listar produtos cadastrados em ordem alfabética - Buscar produtos por nome ou descrição - Editar dados de produtos existentes - Excluir produtos (inativação lógica) - Validar campos obrigatórios

#### RF005 - Controle de Estoque

**Descrição:** O sistema deve permitir o controle de entrada e saída de produtos. **Critérios de Aceitação:** - Registrar movimentações de entrada no estoque - Registrar movimentações de saída do estoque - Informar data da movimentação - Permitir observações nas movimentações - Atualizar automaticamente o estoque do produto - Validar quantidade disponível para saída

### **RF006 - Alertas de Estoque Baixo**

**Descrição:** O sistema deve alertar quando produtos estiverem com estoque abaixo do mínimo. **Critérios de Aceitação:** - Verificar automaticamente estoque após movimentações de saída - Exibir alerta visual quando estoque  $\leq$  estoque mínimo - Listar produtos com estoque baixo no dashboard - Destacar produtos com estoque baixo nas listagens

### **RF007 - Histórico de Movimentações**

**Descrição:** O sistema deve manter histórico completo de todas as movimentações. **Critérios de Aceitação:** - Registrar data/hora, tipo, quantidade, usuário e observações - Exibir histórico por produto - Exibir histórico geral ordenado por data - Permitir consulta de movimentações por período

### **RF008 - Dashboard Principal**

**Descrição:** O sistema deve ter uma tela principal com resumo das informações. **Critérios de Aceitação:** - Exibir nome do usuário logado - Mostrar total de produtos cadastrados - Listar produtos com estoque baixo - Exibir últimas movimentações - Permitir navegação para outras funcionalidades - Opção de logout

### **RF009 - Interface de Busca**

**Descrição:** O sistema deve permitir busca de produtos por termo. **Critérios de Aceitação:** - Campo de busca na listagem de produtos - Buscar por nome ou descrição do produto - Exibir resultados filtrados - Opção para limpar filtro e ver todos os produtos

### **RF010 - Validações e Mensagens**

**Descrição:** O sistema deve validar dados e exibir mensagens apropriadas. **Critérios de Aceitação:** - Validar campos obrigatórios nos formulários - Exibir mensagens de erro específicas - Confirmar ações de exclusão - Exibir mensagens de sucesso após operações - Validar formato de email e valores numéricos

---

## **2. CASOS DE TESTE**

### **CT001 - Login com Credenciais Válidas**

**Pré-condições:** Usuário cadastrado no sistema **Passos:** 1. Acessar tela de login 2. Informar email: admin@admin.com 3. Informar senha: admin123 4. Clicar em “Entrar” **Resultado Esperado:** Redirecionamento para dashboard

### **CT002 - Login com Credenciais Inválidas**

**Pré-condições:** Tela de login aberta **Passos:** 1. Informar email: usuario@inexistente.com 2. Informar senha: senhaerrada 3. Clicar em “Entrar” **Resultado Esperado:** Mensagem de erro e permanência na tela de login

### **CT003 - Cadastro de Produto Válido**

**Pré-condições:** Usuário logado, categoria cadastrada **Passos:** 1. Acessar “Cadastro de Produtos” 2. Clicar em “Novo Produto” 3. Preencher todos os campos obrigatórios 4. Selecionar categoria 5. Clicar em “Salvar” **Resultado Esperado:** Produto cadastrado e listado na tela de produtos

### **CT004 - Busca de Produtos**

**Pré-condições:** Produtos cadastrados no sistema **Passos:** 1. Acessar listagem de produtos 2. Digitar termo de busca no campo 3. Clicar em “Buscar” **Resultado Esperado:** Exibição apenas dos produtos que contenham o termo

### **CT005 - Movimentação de Entrada no Estoque**

**Pré-condições:** Produto cadastrado, usuário logado **Passos:** 1. Acessar “Gestão de Estoque” 2. Clicar em “Movimentar” para um produto 3. Selecionar tipo “Entrada” 4. Informar quantidade 5. Clicar em “Registrar Movimentação” **Resultado Esperado:** Estoque do produto aumentado, movimentação registrada

### **CT006 - Movimentação de Saída com Estoque Insuficiente**

**Pré-condições:** Produto com estoque menor que quantidade solicitada **Passos:** 1. Tentar registrar saída maior que estoque disponível 2. Clicar em “Registrar Movimentação” **Resultado Esperado:** Mensagem de erro informando estoque insuficiente

### **CT007 - Alerta de Estoque Baixo**

**Pré-condições:** Produto com estoque igual ou menor que o mínimo **Passos:** 1. Acessar dashboard ou gestão de estoque **Resultado Esperado:** Produto destacado em vermelho com alerta de estoque baixo

### **CT008 - Validação de Campos Obrigatórios**

**Pré-condições:** Formulário de cadastro de produto aberto **Passos:** 1. Deixar campos obrigatórios em branco 2. Tentar salvar **Resultado Esperado:** Mensagens de erro nos campos obrigatórios

---

### 3. REQUISITOS DE INFRAESTRUTURA

#### 3.1 Sistema Gerenciador de Banco de Dados

- **SGBD:** MySQL
- **Versão:** 8.0 ou superior
- **Justificativa:** Banco robusto, amplamente utilizado, com boa performance para aplicações web

#### 3.2 Linguagem de Programação

- **Linguagem:** Java
- **Versão:** 17 (LTS)
- **Framework:** Spring Boot 3.5.6
- **Justificativa:** Linguagem madura, multiplataforma, com ecossistema robusto

#### 3.3 Sistema Operacional de Desenvolvimento

- **SO:** Windows 11
- **Versão:** 22H2 ou superior
- **Justificativa:** Sistema operacional utilizado no ambiente de desenvolvimento

#### 3.4 Ferramentas Adicionais

- **Build Tool:** Maven 3.8+
- **IDE:** Qualquer IDE compatível com Java (IntelliJ IDEA, Eclipse, VS Code)
- **Servidor Web:** Tomcat Embarcado (Spring Boot)
- **Template Engine:** Thymeleaf 3.1+
- **Framework CSS:** Bootstrap 5.3+

#### 3.5 Requisitos de Hardware Mínimos

- **Processador:** Intel Core i3 ou AMD equivalente
- **Memória RAM:** 4GB (recomendado 8GB)
- **Espaço em Disco:** 2GB livres
- **Conexão de Rede:** Para download de dependências Maven

#### 3.6 Configurações de Banco

- **Porta:** 3306 (padrão MySQL)
- **Charset:** UTF-8
- **Timezone:** UTC
- **Nome do Banco:** saep\_db

## 4. ARQUITETURA DO SISTEMA

### 4.1 Padrão Arquitetural

- **Padrão:** MVC (Model-View-Controller)
- **Camadas:**
  - **Model:** Entidades JPA (Usuario, Produto, Categoria, MovimentoEstoque)
  - **View:** Templates Thymeleaf com Bootstrap
  - **Controller:** Classes Spring MVC para controle de requisições

### 4.2 Estrutura de Pacotes

```
com.loja.lojaaviamento
    config/           # Configurações (Security, DataInitializer)
    controller/      # Controladores MVC
    Domain/          # Entidade Usuario (legado)
    model/           # Entidades do domínio
    repository/     # Interfaces JPA Repository
    service/         # Regras de negócio
    util/            # Classes utilitárias
```

### 4.3 Tecnologias Utilizadas

- **Backend:** Spring Boot, Spring Security, Spring Data JPA
  - **Frontend:** Thymeleaf, Bootstrap, JavaScript
  - **Banco de Dados:** MySQL com Hibernate
  - **Validação:** Bean Validation (JSR-303)
  - **Segurança:** Spring Security com BCrypt
- 

## 5. INSTRUÇÕES DE INSTALAÇÃO E EXECUÇÃO

### 5.1 Pré-requisitos

1. Java 17 instalado
2. MySQL Server instalado e rodando
3. Maven instalado (ou usar wrapper incluído)

### 5.2 Configuração do Banco

1. Criar banco de dados: CREATE DATABASE saep\_db;
2. Executar script SQL fornecido (script\_banco\_saep\_db.sql)

### 5.3 Execução da Aplicação

1. Clonar/extrair o projeto
2. Navegar até a pasta do projeto

3. Executar: `mvn spring-boot:run`
4. Acessar: `http://localhost:8080`

#### 5.4 Usuários Padrão

- **Admin:** admin@admin.com / admin123
  - **Usuário:** user@user.com / user123
- 

**Data:** Janeiro 2025

**Versão:** 1.0

**Autor:** Sistema SAEP 2025