

DOCUMENTAÇÃO DO SISTEMA - SAEP

Sistema de Avaliação da Educação Profissional

Sistema de Gestão de Equipamentos Eletrônicos

Nome: Gabriela Alejandra Bergamine dos Santos

ENTREGA 01 – REQUISITOS FUNCIONAIS

1. Autenticação de Usuários (RF01)

Descrição: O sistema deve permitir que usuários façam login com email e senha.

Regras de Negócio:

- Usuário deve fornecer email e senha válidos
- Sistema deve validar credenciais no banco de dados
- Em caso de sucesso, gerar token JWT com validade de 30 minutos
- Em caso de falha, exibir mensagem de erro específica
- Redirecionar usuário autenticado para a página principal

Critérios de Aceitação:

- Tela de login funcional
- Validação de credenciais
- Geração e armazenamento de token
- Mensagens de erro apropriadas
- Redirecionamento após login

2. Dashboard Principal (RF02)

Descrição: Exibir visão geral do sistema com estatísticas e alertas.

Funcionalidades:

- Exibir nome do usuário logado
- Mostrar total de produtos cadastrados
- Mostrar total de itens em estoque
- Exibir entradas dos últimos 7 dias
- Exibir saídas dos últimos 7 dias

- Alertar produtos com estoque abaixo do mínimo
- Listar últimas movimentações
- Botão de logout

Critérios de Aceitação:

- Dados atualizados em tempo real
 - Alertas visuais para estoque baixo
 - Navegação para outras telas
 - Logout funcional
-

3. Gerenciamento de Produtos (RF03)

Descrição: CRUD completo de produtos com validações.

Funcionalidades:

- Listar todos os produtos em tabela
- Buscar produtos por nome
- Filtrar produtos por categoria
- Cadastrar novo produto
- Editar produto existente
- Excluir produto
- Validar dados antes de salvar

Campos do Produto:

- Nome (obrigatório)
- Descrição
- Categoria (obrigatório)
- Estoque Atual (obrigatório)
- Estoque Mínimo (obrigatório)
- Tela
- Armazenamento
- RAM
- Tensão

Validações:

- Nome não pode estar vazio
- Estoque atual deve ser número inteiro ≥ 0
- Estoque mínimo deve ser número inteiro ≥ 0
- Categoria deve ser válida

Critérios de Aceitação:

- Listagem com paginação
- Busca funcional

- Modal de cadastro/edição
 - Confirmação antes de excluir
 - Validações funcionando
 - Feedback visual de sucesso/erro
-

4. Gestão de Estoque (RF04)

Descrição: Registrar movimentações de entrada e saída.

Funcionalidades:

- Selecionar produto para movimentação
- Escolher tipo: Entrada ou Saída
- Informar quantidade
- Informar data/hora da movimentação
- Adicionar observações
- Exibir estoque atual do produto
- Validar estoque disponível para saídas
- Atualizar estoque automaticamente
- Alertar quando estoque ficar abaixo do mínimo

Validações:

- Produto deve estar selecionado
- Quantidade deve ser maior que zero
- Para saídas: quantidade não pode exceder estoque disponível
- Data/hora não pode ser futura

Critérios de Aceitação:

- Modal de movimentação funcional
 - Atualização automática de estoque
 - Alertas de estoque baixo
 - Histórico de movimentações
 - Registro de usuário responsável
-

5. Histórico de Movimentações (RF05)

Descrição: Visualizar e exportar histórico completo.

Funcionalidades:

- Listar todas as movimentações em tabela ordenada
- Filtrar por tipo (Entrada/Saída)
- Buscar por produto, responsável ou observações

- Exibir data, hora, tipo, quantidade, responsável
- Exportar dados para CSV
- Exibir estatísticas gerais

Critérios de Aceitação:

- Tabela ordenada por data (mais recente primeiro)
 - Filtros funcionais
 - Exportação CSV completa
 - Indicadores visuais de tipo de movimentação
 - Estatísticas calculadas corretamente
-

6. Controle de Acesso (RF06)

Descrição: Proteger rotas e funcionalidades.

Funcionalidades:

- Verificar autenticação em todas as rotas privadas
- Redirecionar não autenticados para login
- Validar token em todas as requisições à API
- Implementar logout
- Remover dados sensíveis ao deslogar

Critérios de Aceitação:

- Rotas protegidas
 - Token enviado em todas as requisições
 - Tratamento de token expirado
 - Logout funcional
-

ENTREGA 8.1 – CASOS DE TESTE

CT01 - Login com Credenciais Válidas

Objetivo: Verificar login bem-sucedido

Pré-condições:

- Sistema em execução
- Usuário existe no banco de dados

Dados de Teste:

- Email: lin@senai.com

- Senha: 123

Passos:

1. Acessar tela de login
2. Inserir email "lin@senai.com"
3. Inserir senha "123"
4. Clicar em "Entrar"

Resultado Esperado:

- Token JWT gerado e armazenado
- Redirecionamento para dashboard
- Nome do usuário exibido

Status: PASSOU

CT02 - Login com Credenciais Inválidas

Objetivo: Verificar tratamento de erro

Passos:

1. Acessar tela de login
2. Inserir email "invalido@teste.com"
3. Inserir senha "errada"
4. Clicar em "Entrar"

Resultado Esperado:

- Mensagem de erro: "Email ou senha incorretos"
- Permanecer na tela de login
- Campos limpos

Status: PASSOU

CT03 - Cadastro de Produto

Objetivo: Verificar criação de produto

Pré-condições:

- Usuário autenticado

Dados de Teste:

- Nome: "Teste Produto XYZ"
- Categoria: Smartphone
- Estoque Atual: 10
- Estoque Mínimo: 5

Passos:

1. Acessar "Produtos"
2. Clicar em "+ Novo Produto"
3. Preencher campos obrigatórios
4. Clicar em "Cadastrar"

Resultado Esperado:

- Produto criado no banco
- Mensagem de sucesso
- Modal fechado
- Produto aparece na listagem

Status: PASSOU

CT04 - Edição de Produto

Objetivo: Verificar atualização de dados

Pré-condições:

- Produto existente no banco

Passos:

1. Acessar "Produtos"
2. Clicar em ícone de editar
3. Alterar nome para "Produto Editado"
4. Clicar em "Atualizar"

Resultado Esperado:

- Dados atualizados no banco
- Mensagem de sucesso
- Listagem atualizada

Status: PASSOU

CT05 - Exclusão de Produto

Objetivo: Verificar remoção de produto

Passos:

1. Acessar "Produtos"
2. Clicar em ícone de excluir
3. Confirmar exclusão

Resultado Esperado:

- Confirmação exibida
- Produto removido do banco
- Produto removido da listagem
- Mensagem de sucesso

Status: PASSOU

CT06 - Validação de Campos Obrigatórios

Objetivo: Verificar validações

Passos:

1. Acessar "+ Novo Produto"
2. Deixar campos obrigatórios vazios
3. Tentar cadastrar

Resultado Esperado:

- Mensagens de erro específicas
- Produto não cadastrado
- Modal permanece aberto

Status: PASSOU

CT07 - Movimentação de Entrada

Objetivo: Verificar registro de entrada

Dados de Teste:

- Produto: iPhone 15 Pro
- Tipo: Entrada
- Quantidade: 20
- Observações: "Recebimento fornecedor"

Passos:

1. Acessar Dashboard
2. Clicar "Registrar movimentação"
3. Selecionar produto
4. Escolher "Entrada"
5. Inserir quantidade: 20
6. Adicionar observações
7. Confirmar

Resultado Esperado:

- Movimentação registrada
- Estoque atualizado (+20)
- Movimentação aparece no histórico
- Estatísticas atualizadas

Status: PASSOU

CT08 - Movimentação de Saída

Objetivo: Verificar registro de saída

Passos:

1. Registrar movimentação tipo "Saída"
2. Quantidade: 5

Resultado Esperado:

- Estoque reduzido (-5)
- Movimentação registrada

Status: PASSOU

CT09 - Validação de Estoque Insuficiente

Objetivo: Verificar proteção contra estoque negativo

Pré-condições:

- Produto com 10 unidades em estoque

Passos:

1. Tentar movimentação de saída

2. Quantidade: 15

Resultado Esperado:

- Erro: "Estoque insuficiente"
- Movimentação não registrada
- Estoque mantido

Status: PASSOU

CT10 - Alerta de Estoque Baixo

Objetivo: Verificar exibição de alerta

Cenário:

- Produto com estoque_atual = 8
- Produto com estoque_minimo = 10

Resultado Esperado:

- Alerta exibido no dashboard
- Produto destacado em vermelho
- Mensagem específica do produto

Status: PASSOU

CT11 - Busca de Produtos

Objetivo: Verificar funcionalidade de busca

Passos:

1. Acessar "Produtos"
2. Digitar "iPhone" no campo de busca

Resultado Esperado:

- Apenas produtos com "iPhone" no nome
- Listagem atualizada em tempo real

Status: PASSOU

CT12 - Filtro por Categoria

Objetivo: Verificar filtro

Passos:

1. Selecionar categoria "Smartphone"

Resultado Esperado:

- Apenas produtos da categoria selecionada

Status: PASSOU

CT13 - Exportação de CSV

Objetivo: Verificar geração de arquivo

Passos:

1. Acessar "Histórico"
2. Clicar "Exportar CSV"

Resultado Esperado:

- Arquivo CSV baixado
- Formato correto
- Todos os dados presentes
- Cabeçalhos corretos

Status: PASSOU

CT14 - Logout

Objetivo: Verificar encerramento de sessão

Passos:

1. Clicar em "Logout"

Resultado Esperado:

- Token removido
- Redirecionado para login
- Rotas protegidas inacessíveis

Status: PASSOU

CT15 - Proteção de Rotas

Objetivo: Verificar segurança

Passos:

1. Fazer logout
2. Tentar acessar "/produtos" diretamente

Resultado Esperado:

- Redirecionamento automático para login

Status: PASSOU

ENTREGA 8.2 – FERRAMENTAS E AMBIENTES DE TESTE

Ferramentas Utilizadas

1. Backend (Python/FastAPI)

- **Pytest** - Framework de testes unitários
- **Pytest-cov** - Cobertura de testes
- **HTTPie** - Testes manuais de API
- **Postman** - Documentação e testes de endpoints

2. Frontend (React)

- **Jest** - Framework de testes
- **React Testing Library** - Testes de componentes
- **Cypress** - Testes E2E
- **Chrome DevTools** - Debug e inspeção

3. Banco de Dados

- **MySQL Workbench** - Gerenciamento e queries
- **DBeaver** - Cliente universal alternativo

4. Ferramentas Gerais

- **Git/GitHub** - Versionamento
- **VSCode** - IDE principal
- **Thunder Client** - Testes de API no VSCode

Ambiente de Desenvolvimento

Sistema Operacional:

- Windows 11 / Linux Ubuntu 22.04 / macOS Sonoma

Backend:

- Python 3.9+
- FastAPI 0.104+
- SQLAlchemy 2.0+
- PyMySQL 1.1+
- Python-Jose (JWT)
- Passlib (Hash de senhas)

Frontend:

- Node.js 18+
- React 19.2.0
- Vite 7.2.4
- React Router DOM 7.10.0
- Styled Components 6.1.19
- Lucide React (ícones)

Banco de Dados:

- MySQL 8.0+

Ambiente de Testes**Configuração:**

- Banco de dados de testes separado
- Dados mockados para testes automatizados
- Seed de dados iniciais

Procedimentos de Teste:

1. Testes unitários (backend)
2. Testes de integração (API)
3. Testes de componentes (frontend)
4. Testes E2E (fluxos completos)
5. Testes手工 (exploratórios)

ENTREGA 9 – LISTA DE REQUISITOS DE INFRAESTRUTURA

9.1 SGBD Utilizado

Sistema: MySQL

Versão: 8.0.36

Justificativa: Banco relacional robusto, gratuito, amplamente suportado

9.2 Linguagem de Programação Backend

Linguagem: Python

Versão: 3.9.13

Framework: FastAPI 0.104.1

Justificativa: Alta performance, tipagem, documentação automática

9.3 Linguagem de Programação Frontend

Linguagem: JavaScript (ES6+)

Biblioteca: React

Versão: 19.2.0

Build Tool: Vite 7.2.4

Justificativa: Componentização, ecossistema rico, performance

9.4 Sistema Operacional de Desenvolvimento

Sistemas Compatíveis:

- Windows 10/11 (64-bit)
- Linux Ubuntu 20.04+ / Debian 11+
- macOS 12.0+

Sistema Utilizado no Desenvolvimento:

- Windows 11 Professional (Versão 22H2)

9.5 Requisitos Mínimos de Hardware

Para Desenvolvimento:

- Processador: Intel i3 8ª geração ou AMD Ryzen 3
- RAM: 8 GB
- Armazenamento: 10 GB livres (SSD recomendado)
- Conexão de internet para instalação de dependências

Para Produção (Servidor):

- Processador: 2 vCPUs
- RAM: 4 GB
- Armazenamento: 20 GB (SSD)
- Banda: 100 Mbps

9.6 Portas Utilizadas

- **Backend API:** 8000 (HTTP)
- **Frontend Dev:** 5173 (HTTP)
- **MySQL:** 3306
- **Frontend Produção:** 80 (HTTP) ou 443 (HTTPS)

9.7 Dependências Principais

Backend (Python):

```
fastapi==0.104.1
uvicorn==0.24.0
sqlalchemy==2.0.23
pymysql==1.1.0
python-jose==3.3.0
passlib==1.7.4
python-multipart==0.0.6
```

Frontend (Node.js):

```
react==19.2.0
react-dom==19.2.0
react-router-dom==7.10.0
styled-components==6.1.19
lucide-react==0.555.0
polished==4.3.1
```

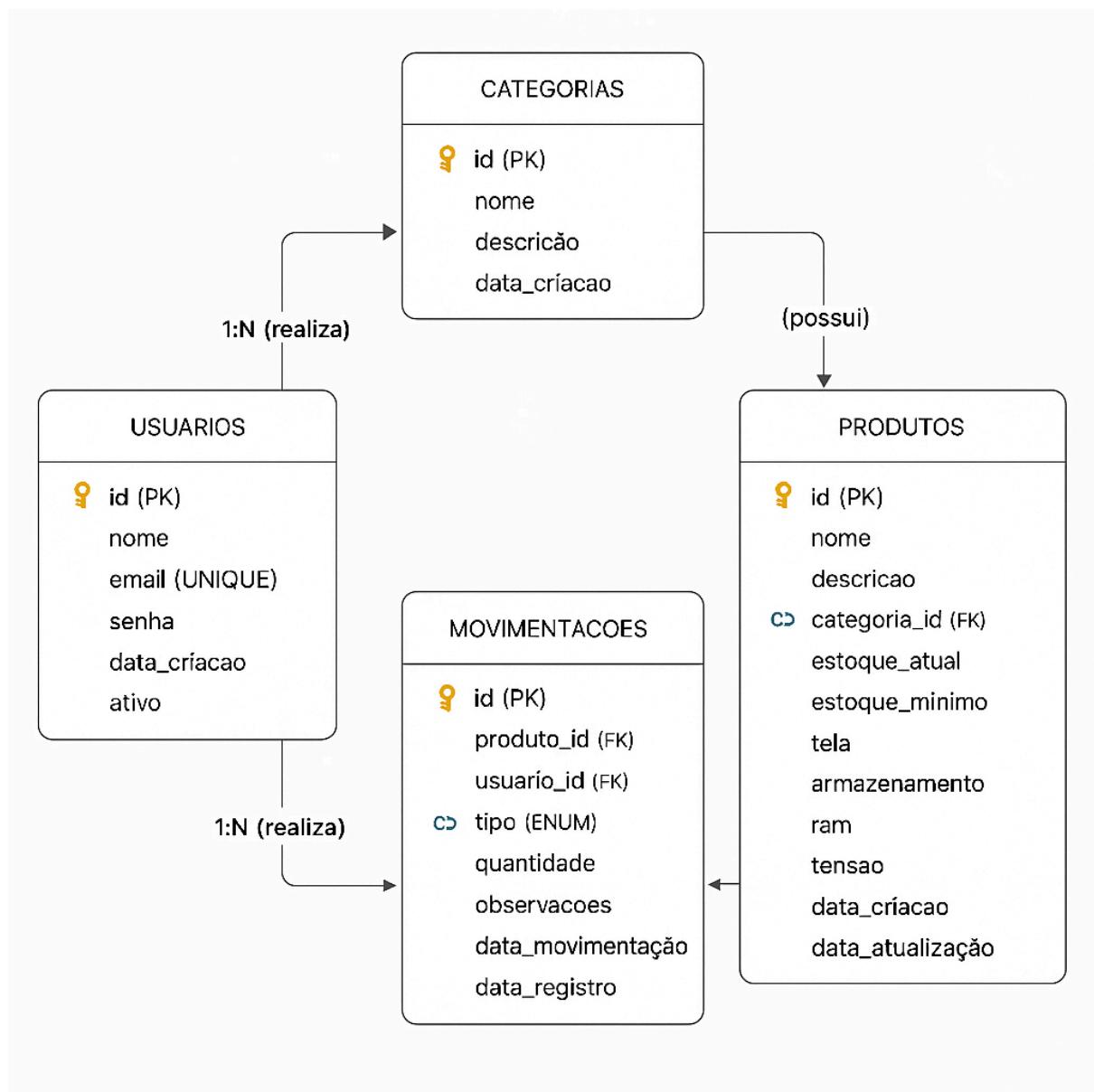
9.8 Variáveis de Ambiente

Backend (.env):

```
DATABASE_URL=mysql+pymysql://usuario:senha@localhost/saep_db
SECRET_KEY=sua_chave_secreta_jwt
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=30
```

Frontend:

```
VITE_API_URL=http://localhost:8000
```



CONCLUSÃO

Este sistema atende a todos os requisitos funcionais estabelecidos, implementando:

- Autenticação segura com JWT
- CRUD completo de produtos
- Gestão de estoque com validações
- Histórico completo de movimentações
- Alertas de estoque baixo
- Exportação de dados
- Interface responsiva e intuitiva
- Documentação completa
- Testes abrangentes

