

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CAMPUS PAU DOS FERROS
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA

ALAN ALMEIDA DA SILVA - 2024011434
DALTON FIRMINO CAMPOS - 2024011414
IVERTON EMIQUISON RIBEIRO DE BESSA - 2024011408
PAULINA JÚLIA COSTA DE OLIVEIRA - 2023023644

SISTEMA DE GERENCIAMENTO DE MERCADO

PAU DOS FERROS – RN

2024

1. INTRODUÇÃO

Este documento apresenta o plano de testes para o sistema de gerenciamento de mercado. O objetivo é garantir que o sistema atenda aos requisitos funcionais e não funcionais, detectando possíveis falhas e garantindo que todos os componentes funcionem corretamente, tanto isoladamente quanto em conjunto.

2. OBJETIVOS

O principal objetivo dos testes é verificar:

- A garantia do comportamento correto das funcionalidades do sistema.
- O funcionamento adequado das unidades de código (testes unitários).
- A integração correta entre diferentes componentes (testes de integração).
- A funcionalidade do sistema sob cenários reais de uso (testes funcionais).

3. ESCOPO

Os testes cobrirão:

1. **Testes Unitários:** Foco em models, forms e views do sistema.
2. **Testes de Integração:** Foco na interação entre diferentes componentes como Funcionários, vendas, produtos e estoque.
3. **Testes Funcionais:** Foco em simular fluxos de uso do sistema, como cadastro de produtos, realizar compras e geração de relatórios financeiros.

4. ABORDAGEM DOS TESTES

4.1 Testes Unitários

Os testes unitários visam testar cada componente do sistema de forma isolada. As seguintes áreas serão cobertas:

- **Modelos:** Testar regras de negócios, como cálculo de total de vendas, atualização de estoque, entre outros.

Exemplo de teste: Verificar se o total de uma venda é calculado corretamente somando os preços dos produtos.

- **Validações:** Garantir que formulários e validações funcionam conforme esperado, como impedir o cadastro de produtos sem nome ou preço.

Exemplo de teste: Testar se um formulário de cadastro de produto falha ao tentar salvar um produto sem preço.

- **Views:** Garantir que as views da API retornem o código de status HTTP correto e realizem a ação apropriada.

Exemplo de teste: Verificar se uma requisição POST para criar um novo produto retorna um status “201 created” e se o produto é salvo no banco de dados.

4.2 Testes de Integração

Os testes de integração garantem que diferentes partes do sistema funcionam corretamente juntas. A integração entre modelos e views será testada em cenários reais de uso:

- **Fluxo de Venda:** Verificar se o processo de adicionar produtos a uma venda e finalizar a venda atualiza corretamente o estoque e o caixa.

Exemplo de teste: Verificar se, após realizar uma venda, o estoque é devidamente atualizado para refletir a quantidade vendida.

- **Relatório Financeiro:** Testar se o sistema gera corretamente relatórios financeiros baseados nas vendas e despesas registradas.

Exemplo de teste: Verificar se o relatório reflete o total de vendas realizadas em um período.

- **Controle de Permissões:** Garantir que diferentes tipos de usuários (ex: caixa, gerente) têm as permissões corretas para acessar funcionalidades.

Exemplo de teste: Verificar se um usuário com permissão de caixa não pode acessar o gerenciamento de estoque.

4.3 Testes Funcionais

Os testes funcionais simulam fluxos completos que um usuário final poderia realizar no sistema. Esses testes validam se o sistema, como um todo, atende às expectativas e aos requisitos funcionais.

- **Cadastro de Produtos:** Verificar o fluxo completo de adicionar, editar e remover produtos no sistema.

Exemplo de teste: Verificar se é possível cadastrar um novo produto, atualizar informações do produto e excluí-lo do sistema.

- **Fechamento de Caixa:** Testar se o sistema permite o fechamento correto do caixa, calculando o saldo final e considerando as vendas realizadas.

Exemplo de teste: Verificar se, ao fechar o caixa, o saldo final é a soma correta do saldo inicial e das vendas realizadas.

- **Pagamentos e Troco:** Verificar se o sistema calcula corretamente o troco ao registrar uma venda.

Exemplo de teste: Verificar se, ao registrar um pagamento de R\$100 para uma venda de R\$50, o troco calculado é de R\$50.

5. CRITÉRIOS DE ACEITAÇÃO

- **Testes Unitários:**

Todos os métodos testados devem retornar os resultados esperados;
O código deve passar por validações sem erros em todos os testes de model, form e view.

- **Testes de Integração:**

Todos os fluxos de negócio devem funcionar corretamente, como a atualização de estoque, a criação de relatórios e a geração de caixas;
Não deve haver falhas ao integrar diferentes partes do sistema.

- **Testes Funcionais:**

O sistema deve ser capaz de executar todos os fluxos principais, como cadastrar produtos, registrar vendas, calcular troco e gerar relatórios sem erros;

As permissões de acesso devem estar corretamente implementadas, restringindo ou permitindo o acesso conforme o perfil do usuário.

6. FERRAMENTAS

Para a execução dos testes, serão utilizadas as seguintes ferramentas:

- **Django Test Framework:** Para escrever e rodar testes unitários, de integração e funcionais.
- **Pytest:** Como alternativa ao sistema de testes padrão do Django.

7. CONCLUSÃO

O plano de testes descrito visa garantir que o sistema de gerenciamento de mercado funcione corretamente em todos os níveis. Com a implementação de testes unitários, de integração e funcionais, será possível assegurar a qualidade do código e a funcionalidade do sistema, além de detectar e corrigir falhas durante o desenvolvimento.