

Plano de Testes - Sistema de Gerenciamento de Inventário

Introdução

O Plano de Testes descreve a estratégia e os procedimentos para testar o Sistema de Gerenciamento de Inventário. O objetivo é garantir que o sistema atenda aos requisitos funcionais e não funcionais e que seja de alta qualidade e confiabilidade.

Objetivo

O objetivo deste plano de testes é definir as atividades de teste necessárias para validar e verificar o funcionamento do Sistema de Gerenciamento de Inventário, garantindo que todas as funcionalidades atendam às expectativas dos usuários.

Escopo

Este plano de testes abrange os testes de unidade, testes de integração e testes de sistema do Sistema de Gerenciamento de Inventário. Ele inclui testes funcionais para validar as principais funcionalidades do sistema.

Estratégia de Teste

- Os testes serão realizados em ambientes de desenvolvimento e teste.
- Serão utilizadas técnicas de teste automatizado para garantir a eficiência e a abrangência dos testes.
- Os casos de teste serão elaborados com base nos requisitos funcionais e não funcionais.
- Serão priorizados os testes de funcionalidades críticas e de alto impacto no sistema.
- Será utilizado um processo de revisão e aprovação para garantir a qualidade dos testes e a conformidade com os requisitos.

Casos de Teste

Testes Unitários

Testes para todas as classes de serviço e repositório.

- Verificação de casos normais, de borda e de exceção.
- Cobertura de código para garantir a qualidade e a robustez do sistema.

Testes de Serviço (*ProdutoService*)

1. **testAddProduto:** Verifica se o método `addProduto` do serviço `ProdutoService` adiciona corretamente um produto.
 - Tipo: Teste Unitário
 - Técnica: Caixa Branca
 - Descrição: Cria um objeto `Produto` com um nome e quantidade específicos, simula a chamada ao método `save` do repositório, e então verifica se o produto retornado é o mesmo que foi adicionado, garantindo que a operação foi bem-sucedida.
2. **testAddProdutoComNomeNulo:** Verifica se o método `addProduto` do serviço lança uma exceção quando o nome do produto é nulo.

- Tipo: Teste Unitário
 - Técnica: Caixa Branca
 - Descrição: Cria um objeto Produto com um nome nulo e uma quantidade válida, e verifica se uma exceção `IllegalArgumentException` é lançada durante a execução do método.
3. `testAddProdutoComQuantidadeNegativa`: Verifica se o método `addProduto` do serviço lança uma exceção quando a quantidade do produto é negativa.
 - Tipo: Teste Unitário
 - Técnica: Caixa Branca
 - Descrição: Cria um objeto Produto com um nome válido e uma quantidade negativa, e verifica se uma exceção `IllegalArgumentException` é lançada durante a execução do método.
 4. `testGetAllProdutos`: Verifica se o método `getAllProdutos` do serviço retorna corretamente uma lista de todos os produtos.
 - Tipo: Teste Unitário
 - Técnica: Caixa Branca
 - Descrição: Cria uma lista de produtos fictícios, simula a chamada ao método `findAll` do repositório para retornar essa lista, e então verifica se a lista retornada pelo serviço corresponde à lista original.
 5. `testAddProdutoNotFound`: Verifica o comportamento do serviço quando o método `save` do repositório retorna null.
 - Tipo: Teste Unitário
 - Técnica: Caixa Branca
 - Descrição: Cria um objeto Produto com um nome e quantidade específicos, simula a chamada ao método `save` do repositório, e verifica se o produto retornado pelo serviço é null, indicando que a operação não foi bem-sucedida.

Testes de Integração

Testes para garantir a integração correta entre os componentes do sistema.

- Verificação da comunicação entre controladores, serviços e acesso ao banco de dados.
- Validação de cenários de uso com dados reais.

Testes de Integração (ProdutoRepositoryIntegrationTest)

1. `testeSalvarProduto`: Verifica se um produto pode ser salvo corretamente no banco de dados.
 - Tipo: Teste de Integração
 - Técnica: Caixa Branca
 - Descrição: Um produto é salvo no banco de dados e o ID gerado é verificado para garantir que a operação de salvamento foi bem-sucedida.
2. `testeEncontrarProdutoPorNome`: Verifica se é possível encontrar um produto no banco de dados por seu nome.
 - Tipo: Teste de Integração
 - Técnica: Caixa Branca

- Descrição: Um produto é salvo no banco de dados e em seguida é feita uma consulta pelo nome do produto. Os resultados da consulta são verificados para garantir que o produto foi encontrado corretamente.

Testes de Repositório (ProdutoRepositoryCRUDTest)

Testes para validar as operações CRUD no repositório.

- Tipo: Teste de Integração
 - Técnica: Caixa Branca
1. testCreateProduto: Verifica se um produto pode ser criado corretamente no banco de dados.
 - Descrição: Um produto é criado e salvo no banco de dados e o ID gerado é verificado para garantir que a operação de criação foi bem-sucedida.
 2. testReadProduto: Verifica se um produto pode ser lido corretamente do banco de dados.
 - Descrição: Um produto é salvo no banco de dados e em seguida é feita uma consulta pelo ID do produto. Os resultados da consulta são verificados para garantir que o produto foi lido corretamente.
 3. testUpdateProduto: Verifica se um produto pode ser atualizado corretamente no banco de dados.
 - Descrição: Um produto é salvo no banco de dados, em seguida é atualizado e novamente salvo. Os resultados da consulta são verificados para garantir que o produto foi atualizado corretamente.
 4. testDeleteProduto: Verifica se um produto pode ser excluído corretamente do banco de dados.
 - Descrição: Um produto é salvo no banco de dados, em seguida é excluído e uma nova consulta é feita pelo ID do produto. Os resultados da consulta são verificados para garantir que o produto foi excluído corretamente.

Testes de Sistema

Testes para validar as principais funcionalidades do sistema, como cadastro, atualização e exclusão de produtos.

- Tipo: Teste de Sistema
 - Técnica: Caixa Preta
1. testCadastroProduto: Verifica se é possível cadastrar um novo produto no sistema.
 - Descrição: Realiza a ação de cadastrar um novo produto e verifica se ele aparece corretamente na lista de produtos cadastrados.
 2. testAtualizacaoProduto: Verifica se é possível atualizar as informações de um produto existente.
 - Descrição: Realiza a ação de atualizar as informações de um produto específico e verifica se as alterações são refletidas corretamente no sistema.
 3. testExclusaoProduto: Verifica se é possível excluir um produto do sistema.
 - Descrição: Realiza a ação de excluir um produto específico e verifica se ele é removido corretamente da lista de produtos.
 4. testConsultaProduto: Verifica se é possível consultar as informações de um produto específico.
 - Descrição: Realiza a ação de consultar as informações de um produto específico e verifica se os dados retornados estão corretos.

5. `testPerformanceCadastroProduto`: Verifica o desempenho do sistema ao cadastrar um novo produto.
 - Descrição: Mede o tempo de resposta do sistema ao realizar a ação de cadastrar um novo produto, garantindo que o sistema funcione adequadamente sob diferentes cargas de trabalho.
6. `testEscalabilidadeCadastroProduto`: Verifica a escalabilidade do sistema ao cadastrar múltiplos produtos simultaneamente.
 - Descrição: Realiza a ação de cadastrar vários produtos ao mesmo tempo e verifica se o sistema consegue lidar com a carga de trabalho sem degradação significativa no desempenho.

Recursos de Teste

- Ambientes de desenvolvimento e teste configurados com as dependências necessárias.
- Ferramentas de automação de teste, como JUnit, Mockito e Spring Boot Test.

Cronograma de Testes

Os testes serão realizados ao longo do ciclo de desenvolvimento, com foco em cada etapa do processo de desenvolvimento e entrega do sistema.

Responsabilidades

- Equipe de Desenvolvimento: Elaboração dos casos de teste.
- Equipe de Testes: Execução dos testes e reporte de problemas.
- Equipe de Gerenciamento: Acompanhamento do progresso dos testes e tomada de medidas corretivas, se necessário.

Riscos e Considerações

- Alterações nos requisitos podem impactar os casos de teste e exigir revisões adicionais.
- Problemas técnicos podem afetar a execução dos testes e a qualidade dos resultados.

Métricas de Teste

- Cobertura de código alcançada pelos testes.
- Número de casos de teste executados e bem-sucedidos.
- Tempo médio de execução dos testes.

Revisão e Aprovação

Este plano de testes será revisado pela equipe de desenvolvimento, testes e gerenciamento para garantir a adequação e a eficácia das atividades de teste.

Esses testes ajudarão a garantir a qualidade e a confiabilidade do seu sistema de gerenciamento de inventário. Certifique-se de executá-los regularmente durante o desenvolvimento para identificar e corrigir quaisquer problemas o mais cedo possível.

