

Plano de Teste: Sistema de Gerenciamento de Farmácia Senac

Plano de Teste

Objetivo: Validar o comportamento do ProdutoDAO nas operações de salvar, listar, excluir e editar produtos no banco de dados.

Escopo: Os testes serão realizados nas funções `salvar()`, `listar()`, `excluir()` e `editar()` da classe ProdutoDAO.

Pré-requisitos:

- Banco de dados configurado e acessível.
- Tabelas de Produtos e Fornecedores existentes e com dados de teste.
- A classe ProdutoDAO deve estar implementada corretamente e funcional.

Ambiente de Teste:

- Banco de dados relacional (MySQL, PostgreSQL, etc.).
- Ferramenta de testes unitários: JUnit.

Critérios de Sucesso:

- As operações de salvar, listar, excluir e editar devem ser executadas corretamente sem erros.
- O banco de dados deve ser atualizado conforme esperado após cada operação.
- O comportamento da aplicação deve ser o esperado, como descrito em cada caso de teste.

Casos de Teste

Caso de Teste 1: Salvar Produto

- **Objetivo:** Verificar se um produto é salvo corretamente no banco de dados.
- **Descrição:** O teste verifica se o produto com dados específicos é inserido no banco corretamente.
- **Entrada:**
 - Descrição: "ASPIRINA"
 - Preço: 4.99
 - Quantidade: 12
 - Fornecedor: código 13
- **Processo:**
 - Criar uma instância de Produto com os dados acima.

- Chamar o método `salvar()` do `ProdutoDAO` para persistir o produto.
- **Saída Esperada:**
 - O produto deve ser inserido na tabela `Produtos` com os dados fornecidos.
 - Não deve ocorrer nenhuma exceção (erro SQL ou de conexão).
 - O banco de dados deve refletir a inserção, com o código do produto sendo atribuído automaticamente.

Caso de Teste 2: Listar Produtos

- **Objetivo:** Verificar se a lista de produtos é retornada corretamente.
- **Descrição:** O teste valida a recuperação de todos os produtos cadastrados no banco de dados.
- **Entrada:**
 - Nenhuma (não há dados de entrada diretos).
- **Processo:**
 - Chamar o método `listar()` do `ProdutoDAO`.
 - Iterar sobre a lista retornada e verificar se os produtos estão sendo recuperados corretamente.
- **Saída Esperada:**
 - A lista de produtos deve ser retornada com todos os campos corretamente preenchidos.
 - Para cada produto, deve-se verificar a descrição, preço, quantidade, e os dados do fornecedor (código e descrição).
 - A impressão no console deve ser bem formatada, com todas as informações do produto e fornecedor.

Caso de Teste 3: Excluir Produto

- **Objetivo:** Verificar se um produto é excluído corretamente do banco de dados.
- **Descrição:** O teste verifica a exclusão de um produto existente com código específico.
- **Entrada:**
 - Código do Produto: 3
- **Processo:**
 - Criar uma instância de `Produto` com código 3.
 - Chamar o método `excluir()` do `ProdutoDAO` para deletar o produto.
- **Saída Esperada:**
 - O produto com código 3 deve ser removido da tabela `Produtos`.
 - O banco de dados não deve conter mais o produto com código 3 após a execução.

Caso de Teste 4: Editar Produto

- **Objetivo:** Verificar se um produto é editado corretamente no banco de dados.
- **Descrição:** O teste valida a edição de um produto existente, alterando seus dados.
- **Entrada:**
 - Código do Produto: 5
 - Nova Descrição: "Cataflan"
 - Novo Preço: 15.75
 - Nova Quantidade: 2
 - Novo Fornecedor: código 10
- **Processo:**
 - Criar uma instância de Produto com o código 5 e os novos dados fornecidos.
 - Chamar o método `editar()` do `ProdutoDAO` para atualizar os dados do produto.
- **Saída Esperada:**
 - O produto com código 5 deve ter sua descrição, preço, quantidade e fornecedor atualizados conforme os valores fornecidos.
 - O banco de dados deve refletir essas mudanças na tabela `Produtos`.