**Plano de Teste do Sistema de Gerenciamento de Biblioteca (LibraryDesk)**

**1. Introdução**

O objetivo deste plano de teste é validar o sistema LibraryDesk, desenvolvido para gerenciar operações de uma biblioteca, assegurando que as funcionalidades de cadastro de autores, categorias, livros, membros e empréstimos operem corretamente.

**2. Escopo do Teste**

O escopo inclui:

* Cadastro de Autores (Inserção, atualização e exclusão de autores).
* Agendamento de Categorias (Inserção, atualização e exclusão de categorias
* Cadastro de Livros (Inserção, atualização e exclusão de livros).
* Cadastro de membros (Inserção, atualização e exclusão de membros).
* Emprestimo de Livros: Registro e devolução de livros.

**3. Critérios de Aceitação**

O sistema será considerado aprovado se:

* Todos os dados forem cadastrados e exibidos corretamente.
* As funcionalidades de adicionar e listar cadastros forem realizadas sem erros.
* O sistema tratar adequadamente entradas inválidas.

**4. Abordagem de Teste**

Os testes incluirão:

* **Teste de unidade:** Verificar cada classe e método individualmente.
* **Teste de integração:** Avaliar a interação entre componentes (ex: cadastrar cidadão e exibir todos).
* **Teste de validação de entrada:** Confirmar que entradas inválidas são gerenciadas corretamente.
* **Teste de usabilidade:** Avaliar a clareza das mensagens e o fluxo do sistema.

**5. Recursos de Teste**

* Ambiente de desenvolvimento em Java.
* IDEs (IntelliJ, Eclipse).
* Ferramenta de teste JUnit para validação das funções.

**6. Cronograma**

* Preparação do ambiente de testes: 2 dias.
* Execução dos testes de unidade: 2 dias.
* Execução dos testes de integração: 1 dias.
* Validação dos resultados: 1 dia.
* Correção de bugs: 1 dias.

**7. Responsabilidades**

**Testadores:**

* Matheus Rocha
* Marcus Vinicius

**8. Riscos e Problemas**

Riscos e problemas identificados:

* Início de tipo ilegal.
* <identificador> esperado.
* Alcançado o fim do arquivo enquanto analisava.
* Classe, interface, enum ou registro esperado.
* Arquivo com o método main(principal)

**Documento de Caso de Teste – Sistema LibraryDESK**

**Identificação**

* **Versão:** 1.0
* **Data de Criação:** 10/03/2025
* **Autor:** Marcus Vinicius e Matheus Rocha
* **Descrição:** Teste para validar cadastro, agendamento e gerenciamento no sistema.

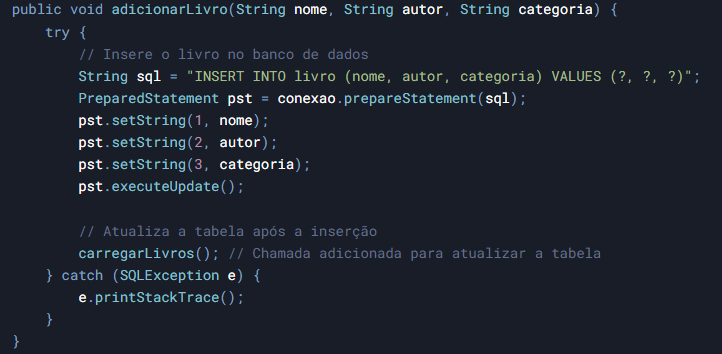
**Todos os testes unitários e de integração foram executados usando JUnit 4.**

* **Teste** testCadastrarLivro: Tabela de Livros não Atualizada após inserção(Após adicionar um novo livro no sistema, a tabela que exibe a lista de livros não é atualizada automaticamente. O livro só aparece na lista após recarregar a tela ou reiniciar o sistema).

**Causa:** O problema ocorre porque, após a inserção do livro no banco de dados, o método que carrega os dados na tabela não é chamado novamente. Isso faz com que a tabela continue exibindo os dados antigos, sem refletir a nova inserção.

**Solução:** Para corrigir o erro, basta chamar o método carregarLivros() após a inserção do livro, garantindo que a tabela seja atualizada com os dados mais recentes.

Codigo Corrigido:



Data de realização do teste:

10/03/2025 – 02:03 horas.

;