PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Bacharelado em Sistemas de Informação

TRABALHO PRÁTICO 2

Teste e Manutenção de Software

Locadora de Filmes

Autores:

Ana Luíza Gonçalves Leite

Gabriel Lucas

João Victor Vasconcellos

Letícia Gomes S. Costa

Lucas Silva Tanure

Marcos Vinicius Nunes

Murilo Silva

Sumário

1. Descrição do Software

O sistema **Locadora de Filmes** é uma aplicação Java desenvolvida para informatizar o gerenciamento de uma locadora tradicional. O objetivo é automatizar processos como cadastro de filmes e clientes, controle de locações e devoluções, envio de notificações e geração de relatórios, proporcionando eficiência, segurança e praticidade.

A arquitetura do sistema segue um modelo modular, dividido entre as camadas de modelo de dados, regras de negócio e interface com o usuário. Isso garante clareza no código, facilidade de manutenção e excelente testabilidade.

A persistência dos dados é realizada por meio do banco de dados **SQLite**, o que permite armazenar informações como filmes cadastrados, clientes, locações, pagamentos e dados de funcionários. O sistema cria e manipula tabelas como filmes, clientes, locações, pagamentos e funcionários. Todos os dados são salvos de forma permanente, garantindo integridade mesmo após o fechamento da aplicação.

Entre as funcionalidades implementadas, destacam-se:

- Cadastro e edição de filmes com informações como título, categoria, duração, diretor, ano de lançamento, data de estreia e quantidade em estoque.
- Cadastro de clientes com dados pessoais e de contato.
- Realização de locações com controle de estoque e data prevista para devolução.
- Registro de devoluções e atualização automática do estoque.
- Envio de notificações para clientes em caso de atraso, promoções e avisos internos para funcionários.
- Geração de relatórios de locações por período e lista dos filmes mais alugados.

2. Casos de Teste Implementados

Foram desenvolvidos 52 **testes unitários**, distribuídos nas seguintes áreas:

- A classe ServiceTest, com 26 testes, cobre a lógica de negócio principal, como locações, devoluções, controle de estoque, notificações e geração de relatórios.
- A classe UiTest, com 16 testes, valida o comportamento da interface por linha de comando, especialmente entrada de dados do usuário.
- A classe ModelTest, com 10 testes, testa os atributos e comportamentos das entidades como Cliente, Filme, Locacao, Estoque, Pagamento e Funcionario.

Os testes abrangem tanto cenários positivos quanto negativos, além de exceções esperadas, garantindo maior confiabilidade ao sistema.

3. Cobertura de Testes

A cobertura de testes foi medida com a ferramenta **JaCoCo**, que gerou o relatório detalhado com os seguintes resultados:

- Aproximadamente **80% das instruções** do código (operações executáveis em bytecode) foram cobertas, com 446 instruções não cobertas de um total de 2.299.
- Cerca de **78% das linhas de código executável** foram exercitadas nos testes, com 485 linhas cobertas de um total de 617.
- **65% das ramificações** (condições lógicas como if, else, switch) foram testadas, com 44 ramificações não cobertas de um total de 129.
- **91% dos métodos** do sistema foram executados durante os testes, com 12 métodos não cobertos de um total de 127.
- **100% das classes** foram cobertas, ou seja, todas as 13 classes principais do sistema foram instanciadas e utilizadas ao menos uma vez durante os testes.

Esses resultados demonstram uma cobertura ampla e consistente, reforçando a confiabilidade do sistema e a efetividade da estratégia de testes adotada.

.

Locadora Filmes

Locadora Filmes

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.seuprojeto.locadora.ui		53%		63%	16	35	68	147	1	11	0	2
com.seuprojeto.locadora.service		89%		86%	8	57	34	312	1	31	0	4
com.seuprojeto.locadora.model		82%		38%	22	103	22	158	10	85	0	7
Total	446 of 2.299	80%	44 of 129	65%	46	195	124	617	12	127	0	13

4. Software com Testes Implementados

O projeto foi desenvolvido em **Java**, com testes estruturados usando a biblioteca **JUnit 5**. Para simulações de componentes e verificação de chamadas, foi utilizado o **Mockito**. A interface em console foi testada por meio de simulação de entrada (Scanner) e captura da saída do console (System.out), permitindo testar mensagens e interações com o usuário.

Os testes estão organizados em pacotes específicos para os módulos do sistema: model, service e ui.

5. Ferramenta de Medição da Cobertura

Foi utilizada a ferramenta **JaCoCo (Java Code Coverage Library)** para calcular a cobertura de testes. Os relatórios gerados no formato HTML oferecem uma visão detalhada da cobertura por pacote, classe, método e linha de código.

6. Arquivos, Classes e Métodos Testados

O sistema é composto por 13 classes principais, todas elas sendo exercitadas pelos testes unitários, garantindo assim **cobertura completa das classes** do projeto. Dentro dessas classes, existem 127 métodos, dos quais aproximadamente 87% foram efetivamente testados, o que demonstra uma ampla verificação do comportamento funcional da aplicação.

No que diz respeito ao código executável, o relatório identificou 617 linhas, das quais cerca de 78% foram cobertas pelos testes, evidenciando que a maior parte do código foi validada durante o processo de teste.

Além disso, das 129 estruturas de decisão presentes no código, como condicionais if, else e switch, aproximadamente 65% foram exercitadas, indicando uma boa cobertura dos diferentes fluxos lógicos do sistema, embora haja espaço para ampliar os testes em cenários alternativos.

Essa distribuição mostra que o projeto possui uma base sólida de testes, com foco significativo tanto nas estruturas principais quanto nos detalhes do comportamento da aplicação, contribuindo para a qualidade, confiabilidade e manutenção futura do sistema.

Conclusão

O sistema **Locadora de Filmes** apresenta uma cobertura robusta de testes, com validações de entrada, manipulação de banco de dados SQLite, regras de negócio consolidadas e uma arquitetura modular que facilita a manutenção. A medição de cobertura confirma a qualidade da implementação e a confiabilidade da aplicação, que está pronta para uso real ou continuidade acadêmica.

Link repositório: https://github.com/LeticiaCosta31/Locadora-filmes.git