



UNIVERSIDADE ESTADUAL DA PARAÍBA

**CAMPUS DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**THOMAS FREIRE
ZAYON TAVARES
JOÃO VICTOR BATISTA**

Relatório de Testes e Cobertura de Código

**CAMPINA GRANDE - PB
2025**

1. Introdução

Este relatório apresenta os testes realizados e a análise da cobertura de código para o projeto desenvolvido em Java. Os testes foram implementados utilizando **JUnit** e **Mockito**, enquanto a cobertura de código foi medida por meio da ferramenta integrada do **IntelliJ IDEA**. O objetivo deste relatório é fornecer uma visão detalhada sobre a eficácia dos testes implementados e apontar possíveis melhorias.

2. Ferramentas Utilizadas

- **Linguagem:** Java
- **Framework de Testes:** JUnit, Mockito
- **Ferramenta de Cobertura:** IntelliJ IDEA (integrada)

3. Metodologia

Os testes foram organizados em diferentes classes e camadas do sistema, abrangendo as seguintes categorias:

- **DAO (Data Access Object):** Testes relacionados à persistência de dados.
- **Interface:** Testes de comunicação entre o usuário e o sistema.
- **Sistema:** Testes da lógica de negócio.

Para medir a cobertura, foram considerados os seguintes critérios:

- **Cobertura de métodos:** Quantidade de métodos testados em relação ao total.
- **Cobertura de linhas:** Percentual de código executado durante os testes.

4. Resultados dos Testes

A seguir, apresentamos os resultados dos testes executados:

Módulo	Testes Executados	Testes Passaram	Testes Falharam
--------	-------------------	-----------------	-----------------

AdminDAO	12	12	0
AdminInterface	3	3	0
AdminSystem	8	8	0
ClienteDAO	7	7	0
ClienteSystem	4	4	0
ClienteInterface	5	0	5
LojaDAO	8	8	0
LojaInterface	5	4	1
LojaSystem	5	0	5

4.1 Análise dos Resultados

Os resultados indicam que:

- As classes **DAO** e **System** possuem uma boa taxa de sucesso, com todos os testes passando.
- A classe **ClienteInterface** não teve nenhum teste bem-sucedido, sugerindo problemas na implementação ou nos testes.
- A classe **LojaSystem** apresentou uma taxa de falha de 100%, indicando necessidade de revisão urgente.
- A classe **LojaInterface** teve apenas um teste falho, o que pode indicar um erro específico.

5. Cobertura de Código

A ferramenta do IntelliJ IDEA foi utilizada para medir a cobertura de código, fornecendo insights sobre as áreas do sistema que não estão sendo devidamente testadas. Os principais pontos observados foram:

- **Ótima cobertura** nas classes **DAO** e **AdminSystem**, que tiveram 100% de sucesso nos testes.
- **Baixa cobertura** nas classes **ClienteInterface** e **LojaSystem**, que falharam em todos os testes.
- **Possível falta de testes** em alguns métodos críticos da interface de usuário e do sistema.

6. Conclusão

Os testes realizados demonstram que a camada de persistência de dados (**DAO**) está bem testada e funcionando corretamente. No entanto, há falhas significativas na interface e no sistema, especialmente nas classes **ClienteInterface** e **LojaSystem**.

6.1 Recomendações

- Revisar e corrigir os testes que falharam em **ClienteInterface** e **LojaSystem**.
- Ampliar a cobertura de testes para garantir que todas as funcionalidades essenciais sejam verificadas.
- Utilizar técnicas de **mocking com Mockito** para isolar os componentes durante os testes.
- Executar novas rodadas de testes após as correções para garantir estabilidade.

A realização dessas melhorias contribuirá para um sistema mais confiável e com maior qualidade de código.