PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: *Teste e Manutenção de Software*

Trabalho Prático - Parte 2

Arthur Henrique de Oliveira Acácio, Bernardo Silva Oliveira, Daniel Henrique Bicalho Dias, Diogo Augusto Magalhães Marques, Gabriel Augusto Lana Vidal, Gustavo Meira Becattini, Hebert Tadeu de Castro Liberato
Professor(a): *Pedro Felipe Alves de Oliveira*

Belo Horizonte – MG
2025

1 Descrição do Software Desenvolvido

O software desenvolvido é um "Sistema de Gerenciamento de Biblioteca Simplificado"implementado em Python. Ele modela as entidades e operações básicas de uma biblioteca, incluindo:

- Entidades Principais: Pessoa, Autor, Usuario, ItemBibliografico (com especializações Livro, Revista, DVD).
- Processos: Emprestimo de itens, gerenciamento do Catalogo da Biblioteca, registro de Usuarios.
- Configuração: ConfiguracaoBiblioteca para definir parâmetros como limite de empréstimos.

O projeto foi estruturado em um único arquivo contendo todas as classes para simplificar a demonstração e o foco nos testes. O objetivo é fornecer uma base coesa para a aplicação de técnicas de teste de software.

2 Apresentação da quantidade de casos de teste criados

Foram desenvolvidos 38 casos de teste utilizando o framework unittest do Python. Os testes foram agrupados em classes de teste correspondentes a cada classe do modelo de domínio do software, cobrindo:

- Criação e inicialização de objetos.
- Validação de entradas e comportamento em casos de erro (lançamento de exceções).
- Funcionalidades chave de cada classe (e.g., empréstimo de livro, cálculo de totais, adições e remoções em coleções).
- Interações entre classes (e.g., um Usuario pegando um Livro emprestado da Biblioteca).

3 Cálculo de cobertura de testes

A cobertura dos testes foi calculada utilizando a ferramenta coverage.py (versão 7.8.2). Os seguintes comandos foram utilizados:

1. Instalação:
    - python -m pip install coverage
2. Execução dos testes com coleta de dados de cobertura:
    - coverage run -m unittest test_biblioteca.py
3. Geração do relatório de cobertura para o módulo principal:
    - coverage report -m biblioteca_models.py

O relatório gerado indicou uma cobertura de 94% das declarações (statements) do código no arquivo biblioteca_models.py.

3.1 Exemplo de saída do Relatório de Cobertura

Veja a tabela abaixo:

Name	Stmts	Miss	Cover	Missing
biblioteca_models.py	185	11	94%	13, 36, 41, 85, 109, 128, 130, 156, 168, 194, 204
TOTAL	185	11	94%	

Tabela 1: Relatório de Cobertura

As 8 linhas não cobertas (Miss) geralmente correspondem a branches de condicionais não totalmente exploradas ou tratamentos de erro muito específicos que não foram alvo de testes dedicados (por exemplo, um else em uma condição onde o if sempre foi verdadeiro nos testes, ou uma exceção específica que não foi provocada). A meta de 70% de cobertura foi amplamente superada.

4 Relação de Classes e Métodos Criados/Testados

- Arquivos do Software Principal: 1 (biblioteca_models.py)
- Classes no Software Principal: 10
    - Pessoa, Autor, Usuario, ItemBibliografico, Livro, Revista, DVD, Emprestimo, Biblioteca, ConfiguracaoBiblioteca.
- Métodos no Software Principal: 38 (incluindo construtores __init__ e métodos __str__)

- Arquivos de Teste: 1 (test_biblioteca.py)
- Classes de Teste: 10
- Métodos de Teste: 38

4.1 Cobertura de Teste por Unidade:

- Classes Testadas: Todas as 10 classes do software principal possuem classes de teste correspondentes e foram exercitadas pelos testes.
- Métodos Testados: A cobertura de 94% indica que a grande maioria dos métodos foi testada. Os métodos não cobertos integralmente geralmente são aqueles com múltiplas branches condicionais onde nem todos os caminhos foram executados, ou métodos muito simples (getters/setters) que são indiretamente cobertos. O esforço de teste foi direcionado para as lógicas mais críticas e complexas de cada classe.

5 Link do GitHub do projeto criado

O projeto está disponível no seguinte link: teste-software-tp2