

Universidade de Vassouras
Curso de Engenharia de Software

Trabalho de Manutenção e Gestão de Códigos

Turma: A
Período: 7º - 2025
24/03/2025 - Maricá/RJ

Trabalho Acadêmico Apresentado à Disciplina de Gestão da Manutenção e Evolução de Software

Curso de Engenharia de Software - Universidade de Vassouras

Aluno: João Vitor Figueiredo França(202211931) & Gabriel de Souza Lima(202211061)

Professor: Sidney Loyola

SUMÁRIO

1. Introdução - pág 4
2. Justificativas das Alterações Feitas - pág 4
3. Dificuldades Encontradas - pág 5
4. Sugestões de Melhoria para o Sistema - pág 5
5. Conclusão - pág 5
6. Bibliografia - pág 5

1. INTRODUÇÃO

Este relatório tem como objetivo analisar, manter e propor melhorias para um sistema simples de contas bancárias desenvolvido em Python. A atividade foi realizada na disciplina de Manutenção e Gestão de Códigos, com foco nos conceitos de encapsulamento, documentação, boas práticas de programação e testes automatizados.

2. JUSTIFICATIVAS DAS ALTERAÇÕES FEITAS

Durante a análise e execução do código original, algumas modificações foram necessárias para garantir o correto funcionamento e boas práticas de desenvolvimento:

a) Correção de métodos especiais do Python

Erro: os métodos `__init__` e `__str__` estavam como `_init_` e `_str_`.

Correção: ajustes para os nomes corretos.

b) Acesso a atributos privados

Erro: tentativas de acessar atributos privados com apenas um underscore.

Correção: uso de `self.__titular` e `self.__saldo` conforme padrão de encapsulamento.

c) Formatação de saída

Melhoria: uso de formatação monetária nas exibições de saldo.

d) Proteção de execução

Melhoria: inclusão de `if __name__ == "__main__"` para evitar execução indevida ao importar o arquivo.

e) Encapsulamento

Implementado: `@property` e `@setter` para atributos privados.

f) Docstrings

Implementado: documentação clara em cada método e classe.

g) Testes automatizados

Implementado: módulo `unittest` com mais de 5 casos cobrindo funcionalidades essenciais.

3. DIFICULDADES ENCONTRADAS

- Corrigir métodos especiais mal escritos sem mensagens de erro claras.
- Compreensão de conceitos de encapsulamento e name mangling.
- Execução de testes manuais não reaproveitáveis.

4. SUGESTÕES DE MELHORIA PARA O SISTEMA

- Criar um método próprio de transferência entre contas.
- Impedir titulares repetidos.
- Adicionar persistência em arquivos JSON/SQLite.
- Criar interface com Tkinter ou Flask.
- Expandir cobertura dos testes com `pytest`.

5. CONCLUSÃO

As modificações realizadas corrigem falhas críticas do sistema, melhoram a legibilidade e abrem caminho para expansão futura. A manutenção preventiva e corretiva se mostrou essencial para garantir a qualidade e confiabilidade do código. O uso de boas práticas como encapsulamento, documentação e testes automatizados fortaleceu a base do projeto e torna sua evolução mais segura e estruturada.

6. BIBLIOGRAFIA

- Documentação oficial `unittest`: <https://docs.python.org/3/library/unittest.html>