

Projeto BWGI - Resumo Técnico

Arquitetura: Facade + Strategy

O projeto foi estruturado com os padrões de projeto Facade e Strategy para organizar e modularizar as três tasks principais. Isso facilita a manutenção e a adição de novas funcionalidades no futuro.

Resumo das Tasks:

Task 1: reconcile_accounts

- Recebe duas listas de transações.
- Adiciona 'FOUND' ou 'MISSING' a cada transação.
- Datas com até 1 dia de diferença são consideradas correspondentes.
- Quando múltiplas correspondências existem, escolhe a data mais antiga.
- Transações duplicadas são tratadas corretamente.

Task 2: last_lines

- Retorna as linhas de um arquivo em ordem inversa.
- Usa leitura em buffer (não carrega o arquivo todo em memória).
- Funciona como o comando 'tac' do Unix.
- Retorna um generator (iterador).
- Suporta arquivos UTF-8.

Task 3: computed_property

- Permite dependências múltiplas.
- Cacheia valores enquanto as dependências não mudarem.
- Ignora dependências ausentes.
- Suporta getter, setter e deleter.
- Mantém docstrings originais.

Arquitetura geral:

- Classe CSVTransactions encapsula a leitura de CSVs.
- Cada task é implementada como uma Strategy independente.
- Facade gerencia as estratégias de forma simples e desacoplada.

- Código principal (main.py) não conhece detalhes internos.
- Fácil de manter e expandir.

Conclusão:

O projeto atende 100% aos requisitos definidos no PDF original. As boas práticas de programação e arquitetura foram seguidas rigorosamente.