Relatório Técnico – Torre de Hanói com Estrutura de Dados Pilha

1. Introdução

Este projeto tem como objetivo a implementação da clássica Torre de Hanói utilizando estruturas de dados do tipo Pilha. Foram desenvolvidos dois módulos principais: `classe_Pilha.py`, responsável pela abstração da pilha, e `classe_TorreDeHanoi.py`, que modela a lógica do jogo. O código é desenvolvido em Python e emprega boas práticas de modularização, encapsulamento e tratamento de erros.

---

2. Estrutura dos Módulos

2.1. Módulo classe_Pilha.py

Estrutura de Dados Utilizada

- Utiliza a estrutura `array` do módulo nativo `array` do Python.
- Tipos suportados: `int` (padrão) e `str` (com um caractere por elemento).

Componentes Principais

- **Exceções Personalizadas**:
  - `PilhaCheiaErro`
  - `PilhaVaziaErro`
- **Métodos Importantes**:
  - `empilha(dado)` – Insere elemento no topo.
  - `desempilha()` – Remove o elemento do topo.
  - `ver_topo()` – Consulta o topo da pilha sem removê-lo.
  - `troca()` – Troca os dois elementos do topo.
  - `pilha_esta_vazia()` e `pilha_esta_cheia()`.
  - `__str__()` – Representação string da pilha.

Complexidade de Tempo: Operações de empilhar/desempilhar: O(1); Troca de topo: O(1)

Complexidade de Espaço: O(n), onde *n* é o número máximo de elementos definidos na inicialização.

2.2. Módulo classe_TorreDeHanoi.py

Arquitetura

- Define a classe `Torre_de_Hanoi` com três pilhas nomeadas A, B e C.
- O número de discos é parametrizável.
- O método `resolver()` utiliza recursão para resolver o problema da Torre de Hanoi.

Principais Componentes

- `__init__()` – Inicializa as pilhas e insere os discos na haste A.
- `mover_disco(origem, final)` – Realiza as validações e movimenta discos.
- `visualizar()` – Imprime o estado atual das torres.
- `obter_discos(torre)` – Recupera os discos de forma não destrutiva.
- `resolver()` – Implementação recursiva da solução.

Complexidade de Tempo: O(2^n), onde *n* é o número de discos (complexidade clássica do problema).

Complexidade de Espaço: O(n) para a pilha de chamadas recursivas; O(n) por torre, totalizando 3 pilhas com espaço O(n).

4. Conclusão

O projeto atende satisfatoriamente à proposta de implementar o jogo da Torre de Hanoi com uso de estruturas de dados personalizadas. A abstração da pilha foi corretamente feita e utilizada com eficiência. A lógica da Torre de Hanói está correta e funcional, com um bom equilíbrio entre clareza e robustez. Para melhorias futuras, sugere-se automatizar mais a execução, aplicar testes unitários e considerar generalizar a estrutura da pilha para suportar outros tipos.