

Projeto Individual de Modelagem:

Árvore de Decisão (Mock)

Objetivo do Trabalho:

Você deverá modelar, sem implementar o algoritmo real, uma árvore de decisão simplificada, utilizando quatro padrões de projeto fundamentais: Composite, Iterator, Visitor e State.

Descrição do Problema:

Uma árvore de decisão contém nós internos e folhas. A construção da árvore envolve transições entre estados (por exemplo: estado de divisão, estado de parada, estado de poda). A navegação pela árvore deve ser implementada usando Iterator. Operações independentes (como calcular profundidade ou contar folhas) devem ser implementadas com Visitor. A estrutura hierárquica da árvore deve ser modelada com Composite.

Você deverá entregar dois arquivos Python:

1. `tree_design.py`: Estrutura dos padrões de projeto.
2. `tree_demo.py`: Arquivo que demonstra o uso do design e executa operações via prints.

Elementos obrigatórios:

- Composite: Estruturas mínimas: Node, DecisionNode, LeafNode.
- State: Classe TreeBuilder e estados como SplittingState, StoppingState e PruningState.
- Iterator: Um iterador próprio, como PreOrderIterator ou BFSIterator.
- Visitor: Pelo menos dois visitantes independentes, por exemplo: DepthVisitor, CountLeavesVisitor.

Restrições do trabalho:

- Todo o código deve ser *mockado* – nenhum cálculo real. Apenas prints que representem comportamentos.
- Não utilize bibliotecas externas. O foco está no design, não no algoritmo.