



Design Patterns

(Padrões de Projeto)

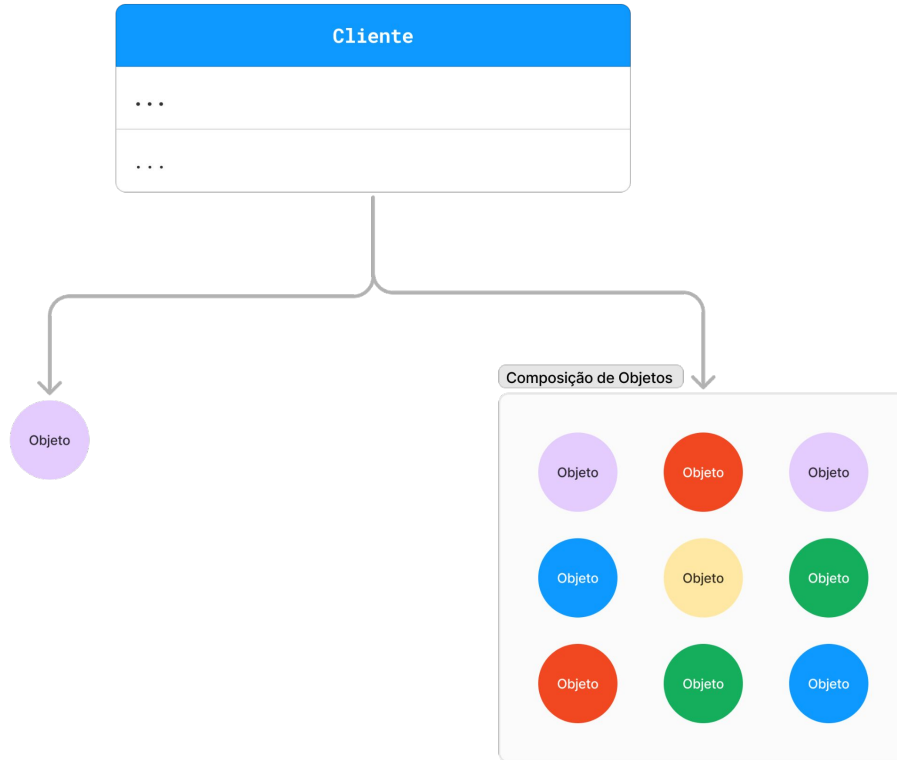


Composite

Definição

O padrão Composite permite a composição de objetos em *estruturas de árvore* para representar hierarquias parte-todo. Com esse padrão, os *clientes podem tratar objetos individuais ou composições de objetos de maneira transparente e uniforme*.

Composite



Composite



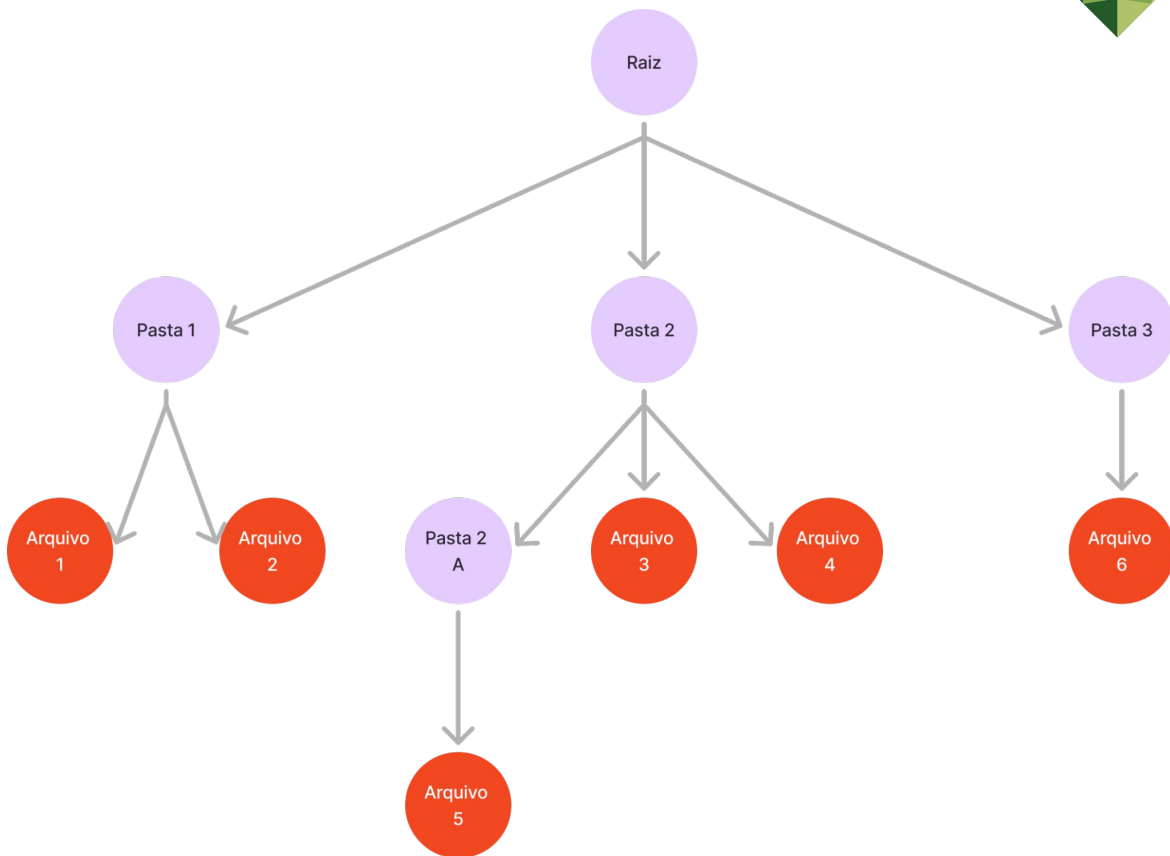
Árvore

Pasta

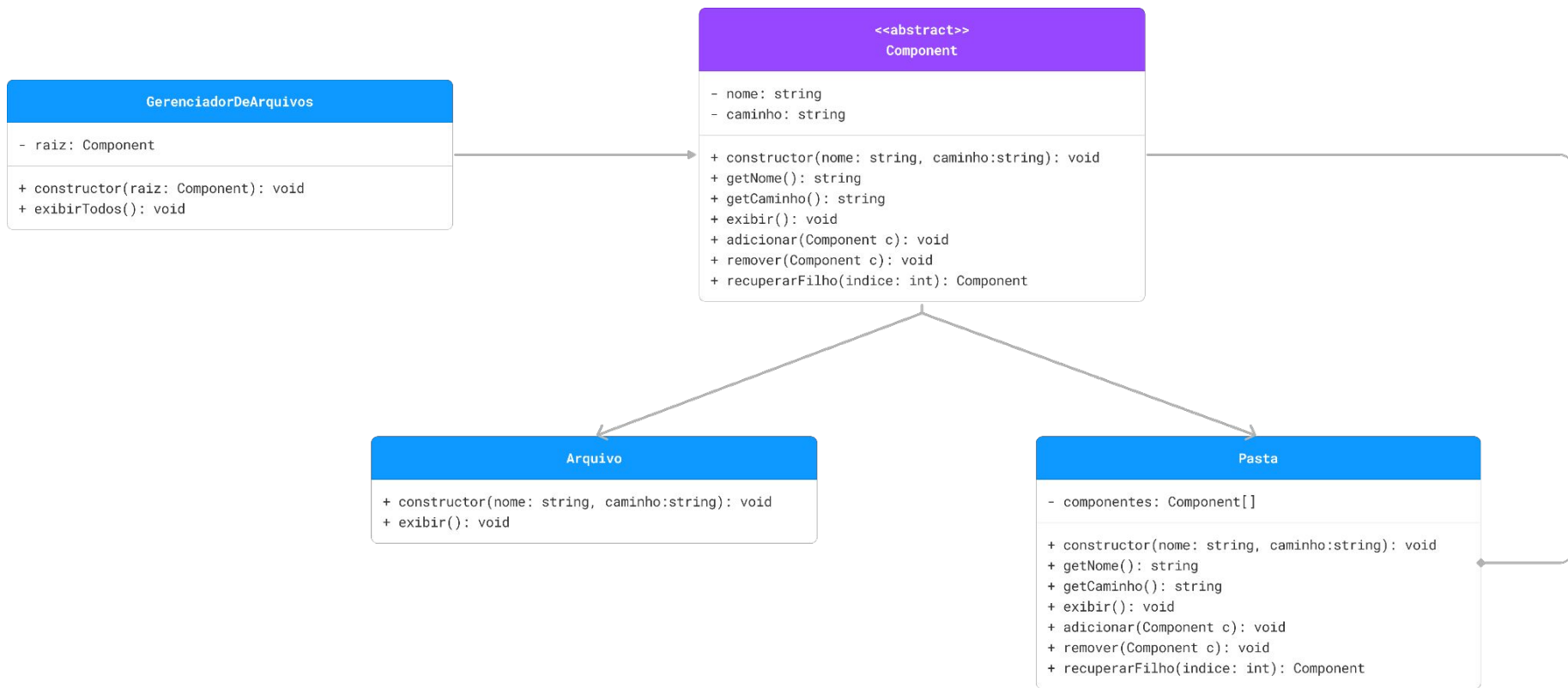
- Nome;
- Caminho;
- Pode ser exibida;
- Pode ter filhos adicionados;
- Pode ter filhos removidos;
- Pode ter filhos recuperados.

Arquivo

- Nome;
- Caminho;
- Pode ser exibido.



Composite





Composite

Aplicabilidade

- Quando é necessário representar uma hierarquia de objetos.
- Quando é preciso que os clientes possam ignorar a diferença entre composições de objetos e objetos individuais.



Composite

Consequências

- Definição de hierarquias de classes que consistem em objetos primitivos(folhas) e objetos compostos(nós).
- Os clientes podem tratar estruturas compostas e objetos individuais de maneira uniforme.
- Fácil adição de novos tipos de componentes.
- Pode tornar a arquitetura do código excessivamente genérica.



Composite

Exercício

Utilizando o padrão Composite, implemente uma estrutura que represente uma organização. Nessa organização, há **departamentos** e **funcionários**. Um departamento pode conter outros departamentos ou funcionários. Os departamentos e funcionários devem ser capazes de calcular seus custos.

Requisitos:

1. Crie uma interface **CustoComponent** que será implementada tanto por departamentos quanto por funcionários.
2. Um **Departamento** pode conter outros departamentos ou funcionários.
3. Um **Funcionário** tem um salário, e esse será o custo dele.
4. O método **calcularCusto** deve retornar o custo total de um departamento, incluindo seus subcomponentes.