# Padrões de Projetos – GoF

### Introdução

Alguns programadores mais experientes começaram a perceber que os mesmos problemas começaram a aparecer várias e várias vezes e a solução para aqueles problemas eram sempre as mesmas e começaram a catalogar esses padrões. Em 1995 um grupo de pessoas, mas especificamente quatro pessoas escreveram um livro iniciando os Design Patterns mais conhecido de mercado, são eles Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides. Eles foram conhecidos como Gang of four ou GOF.

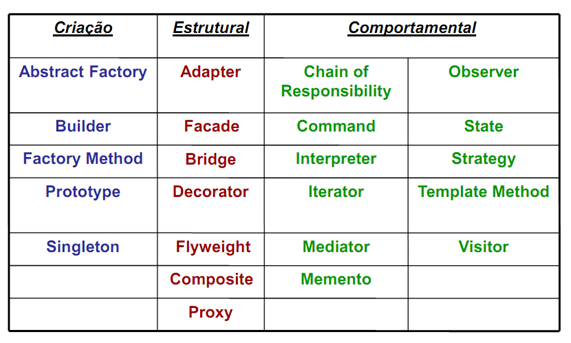


Figura 01 – Resumo das principais catergorias DeisgnPatterns

## Padrão Estrutural

Tratam da alteração da estrutura de um programa, e das associações entre classes e objetos.

**Adapter:** envelopa um objeto e fornece a ele uma interface diferente, permitindo que classes com interfaces incompatíveis possam interagir. O objetivo dessa classe é converter a interface de uma classe em outra interface esperada pelos clientes. Adapte permite a comunicação entre classes que não poderiam trabalhar juntas devido à incompatibilidade de suas interfaces

Diagrama

Descrição gerada automaticamente

Figura 01 – Exemplo Adapter com herança múltipla

**Diagrama

Descrição gerada automaticamente**

Figura 02 – Exemplo Adapter com uso de Composição

**Façade:** Simplifica a interface de um conjunto de classes, Interface unificada para um subsistema, tornando o subsistema mais fácil de usar. Devendo ser utilizada sempre que o cliente precisa saber de muitos detalhes dos subsistemas para utilizá-lo.

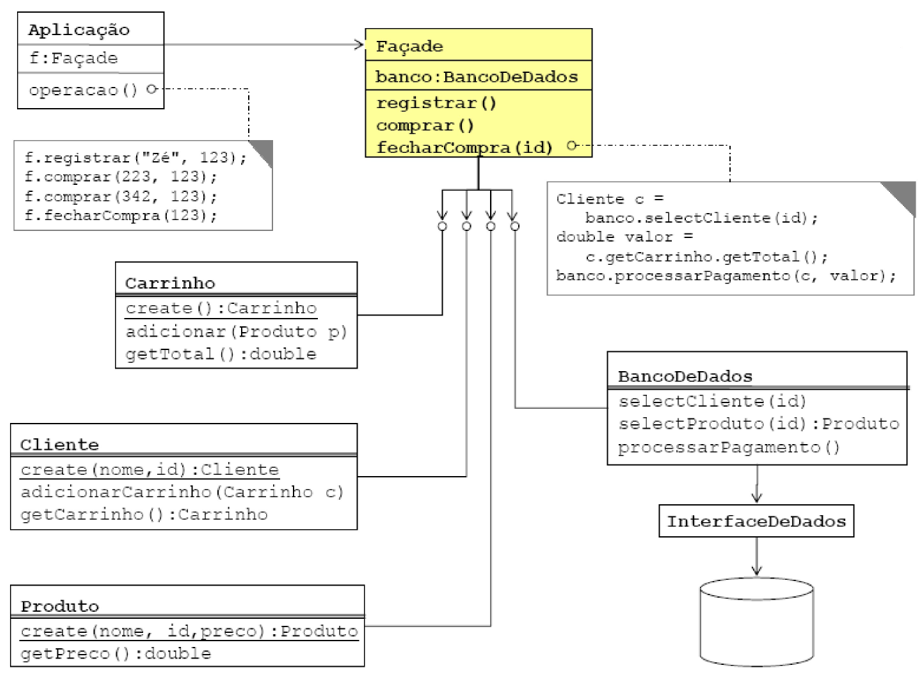


Figura 03 – Exemplo do uso do Façade.

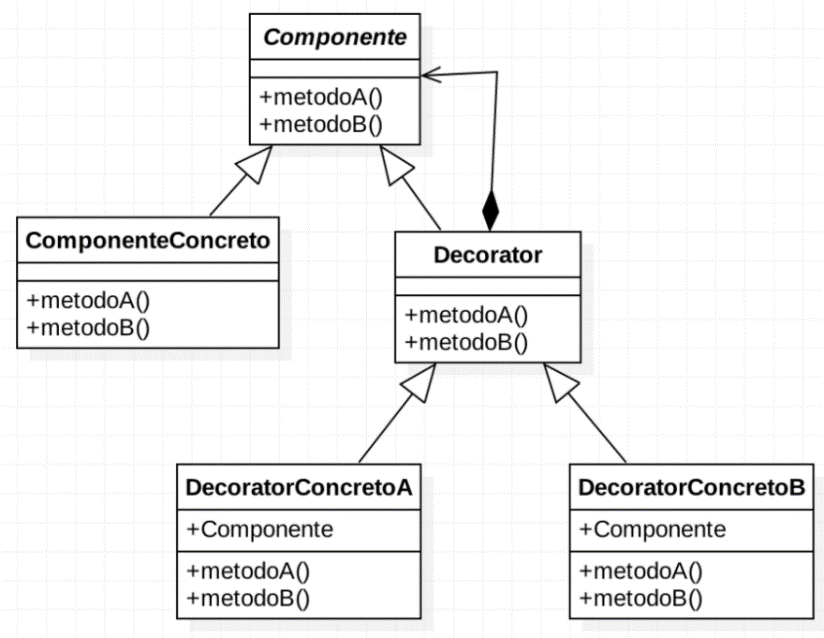
Diagrama

Descrição gerada automaticamente**Bridge:** Permite criar uma ponte para variar não apenas a sua implementação, como também as suas abstrações, desacoplando a interface da implementação e ocultando detalhes dos clientes/aplicações

Diagrama

Descrição gerada automaticamente

**Decorator:** Envelopa um objeto para fornecer novos comportamentos, sendo que o objeto que é constituído pela composição de objetos similares a ele

****

**FlyWeigth:** uma instancia de uma classe pode ser usada para fornecer muitas “instancias virtuais”;

Forma

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

**Composite:** Objeto que é constituído pela composição de objetos similares a ele, tratando as coleções de objetos e os objetos individuais de maneira uniforme, o que compõe objetos em estruturas de árvore para representar hierarquias todo-parte, permitindo que clientes tratem objetos individuais e composições de objetos de maneira uniforme.

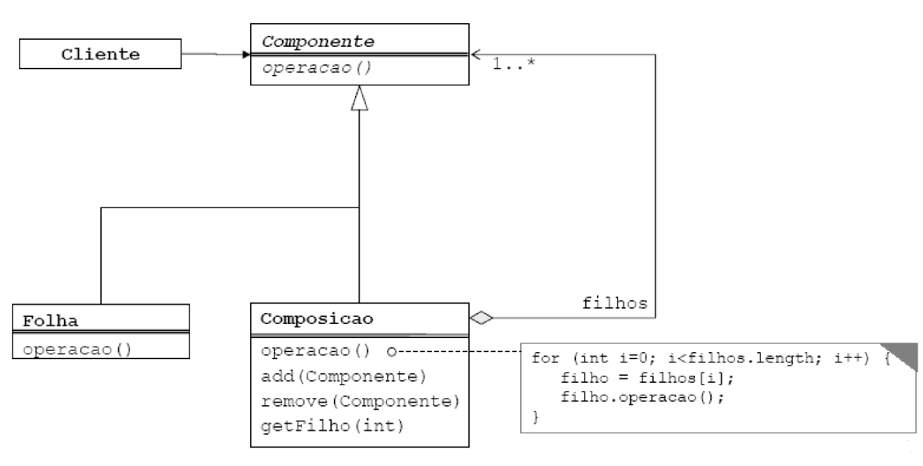


Figura 04 – Exemplo do uso do Composite.

**Proxy:** envelopa um objeto para controlar o acesso a ele.

Diagrama

Descrição gerada automaticamenteNo Proxy o cliente usa uma classe intermediária em vez de da classe real, a classe intermediária suporta a mesma interface da real, para isso a classe intermediária contém uma referência para a classe real e repassa chamadas, acrescentando informações ou filtrando dos no processo.

Figura 05 – Exemplo do uso do Proxy.