

# DESIGN PATTERNS

Flyweight

Gabriel Gomes de Oliveira

# REFERÊNCIAS



REFACTORING  
• GURU •

**Marcos Brizeno**

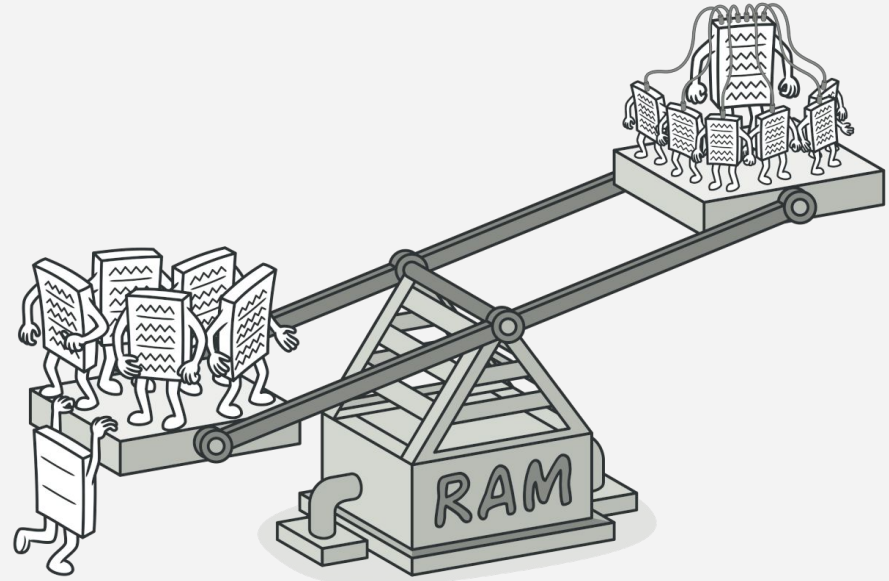
DESENVOLVIMENTO DE SOFTWARE #SHOWMETHECODE



# Flyweight

## O que é?

Padrão de design estrutural que permite ajustar mais objetos à quantidade disponível de RAM, compartilhando partes comuns do estado entre vários objetos, em vez de manter todos os dados em cada objeto.



# Flyweight

## O que é?

Cada objeto "flyweight" é dividido em duas partes: a parte dependente do estado (extrínseca) e a parte independente do estado (intrínseca). O estado intrínseco é armazenado (compartilhado) no objeto Flyweight. O estado extrínseco é armazenado ou calculado pelos objetos do cliente e passado para o Flyweight quando suas operações são invocadas.

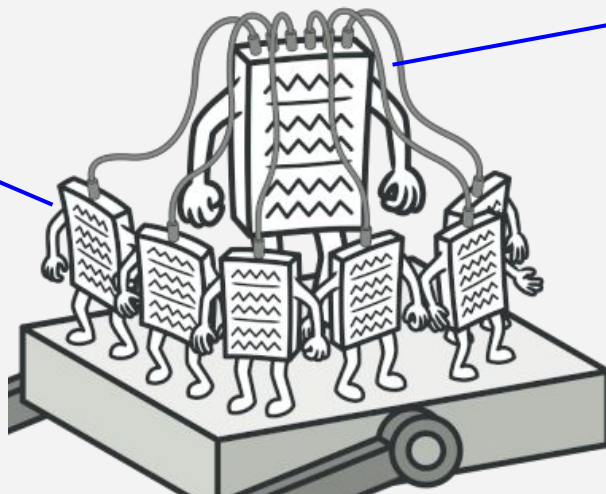
# Flyweight

## O que é?

Cada objeto "flyweight" é dividido em duas partes

extrínseca

intrínseca



# Flyweight

## O que é?

**Intrínseca:** Parte independente do estado (Este é armazenado -Compartilhado- no objeto Flyweight).

**Extrínseca:** Parte dependente do estado (Este é armazenado ou calculado pelos objetos do cliente e passado para o Flyweight quando suas operações são invocadas.)

## Flyweight

Lista de controle:

Garanta que a sobrecarga do objeto seja um problema que precise de atenção.

Divida o estado da classe de destino em estado compartilhável (intrínseco) e estado não compartilhável (extrínseco).

# Flyweight

Lista de controle:

Factory que pode armazenar em cache e reutilizar instâncias de classe existentes.

O cliente deve usar o Factory em vez do novo operador para solicitar objetos.



# Flyweight

Ponto de destaque:

Enquanto o Flyweight mostra como criar muitos objetos pequenos, o Facade mostra como criar um único objeto para representar um subsistema inteiro.