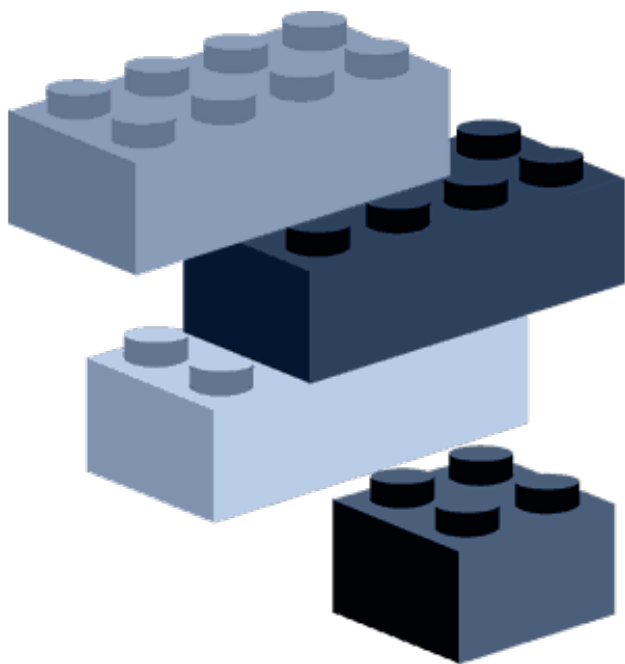




# Design Patterns



# **BUILDER**

## **APRESENTAÇÃO E PRÁTICA DO PP**

# INTRODUÇÃO

O Builder é um padrão de projeto criacional que permite a você **construir objetos complexos passo a passo**.

Ele permite que você produza diferentes tipos de representações de um objeto usando o mesmo código de construção.

A intenção oficial é **separar a construção de um objeto complexo da sua representação** de modo que o mesmo processo de construção possa criar diferentes representações.



# CARACTERÍSTICAS

- Separar o código que cria e o código que usa o objeto;
- Criar objetos complexos:
  - Construtores muito complexos e extensos;
  - Composição de vários objetos;
  - Algoritmo de criação.
- Permite a criação de um objeto em etapas:
  - Pessoa:
    - Etapa 1: crie nome;
    - Etapa 2: crie endereço.
- O objeto final pode variar;
- É padrão complexo.

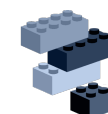
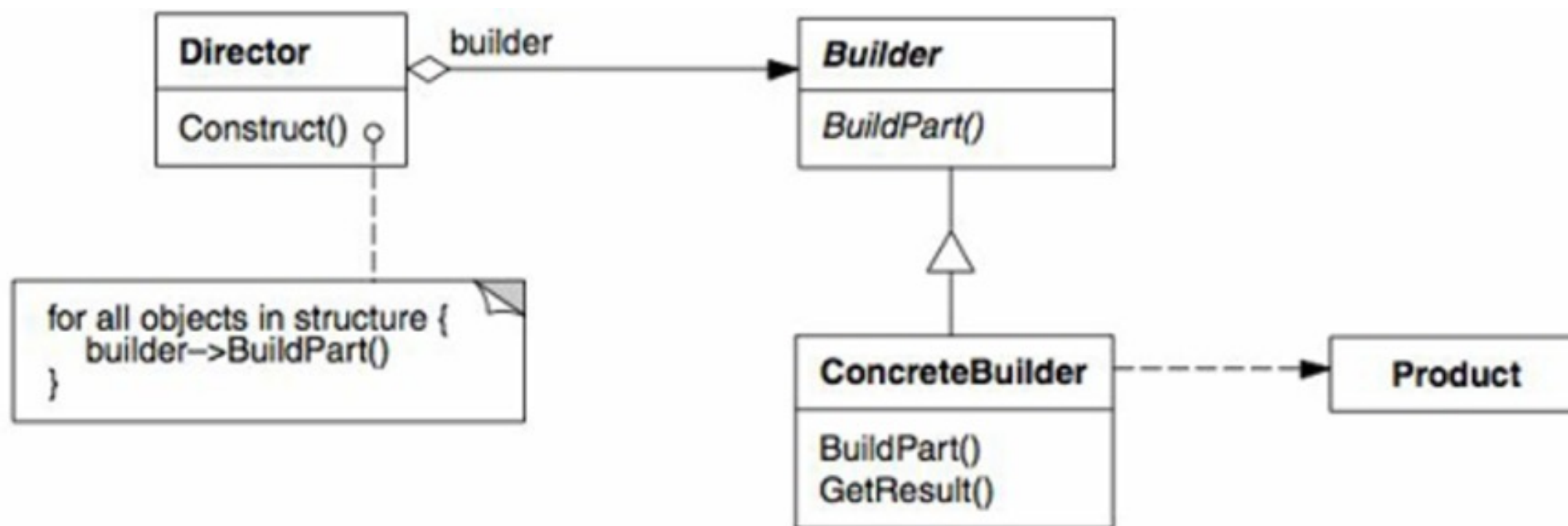


## QUANDO USAR?

- Quando o algoritmo para criação de um objeto complexo deve ser independente das partes que compõem o objeto e de como elas são montadas;
- Quando o processo de construção deve permitir diferentes representações para o objeto que é construído.



# ESTRUTURA



# ESTRUTURA

**Builder:** especifica uma interface abstrata para a criação de partes de um objeto produto.

**ConcreteBuilder:** constrói e monta partes do produto.

**Director:** constrói um objeto usando a interface Builder.

**Product:** objeto complexo em construção.



# CONSEQUÊNCIAS

## Positivas:

Separa criação da utilização;

O cliente não precisa criar objetos diretamente, portanto, o código estará desacoplado;

O mesmo código pode construir objetos diferentes;

Ajuda na aplicação do Single Responsibility Principle;

Ajuda na aplicação do Open Closed Principle.





# CONSEQUÊNCIAS

## Negativas:

🚫 Código final pode se tornar muito complexo.





**Obrigado!**

**Professor Gustavo Dias**  
**luizdias@univas.edu.br**