

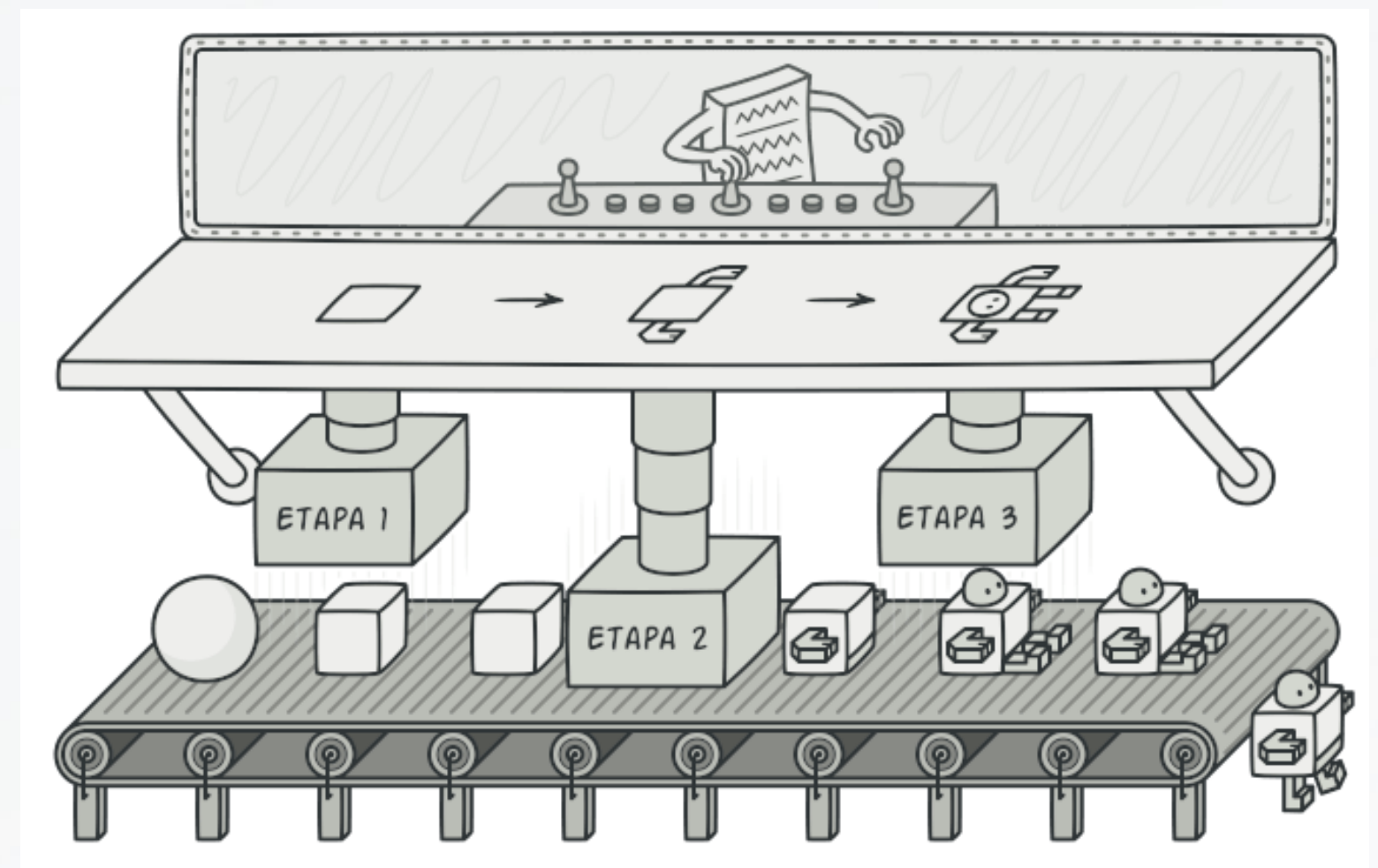


# **PADRÕES DE PROJETO**

## **BUILDER**

# BUILDER

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



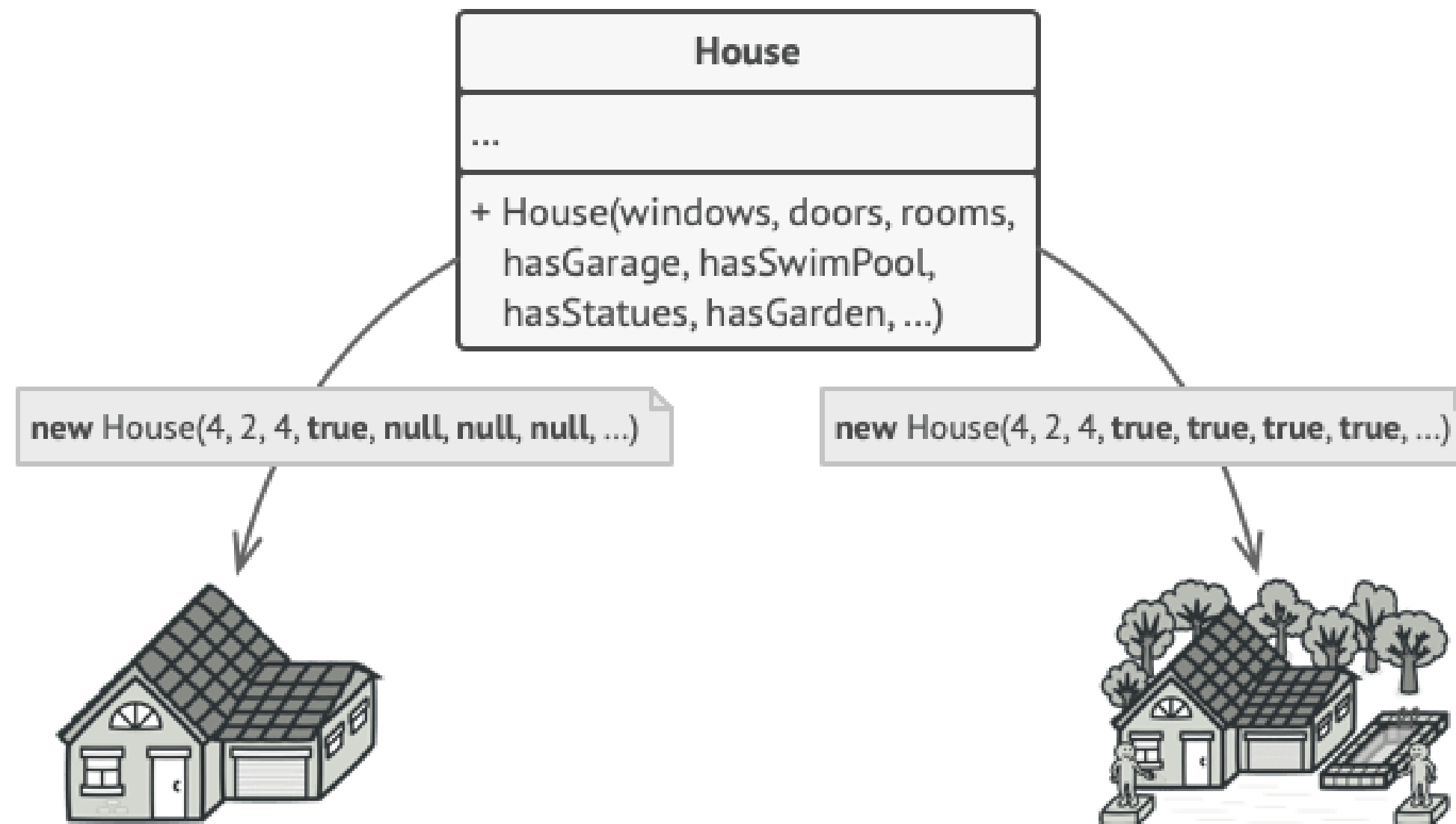
# BUILDER

“Quebrar o processo ou objeto em várias etapas”, criando uma biblioteca de componentes.

- Builders: são os itens para construir o objeto complexo;
- Diretor: Que recebe quais métodos precisa, e dispara a sequência para os builders, que construirão o objeto;

# BUILDER

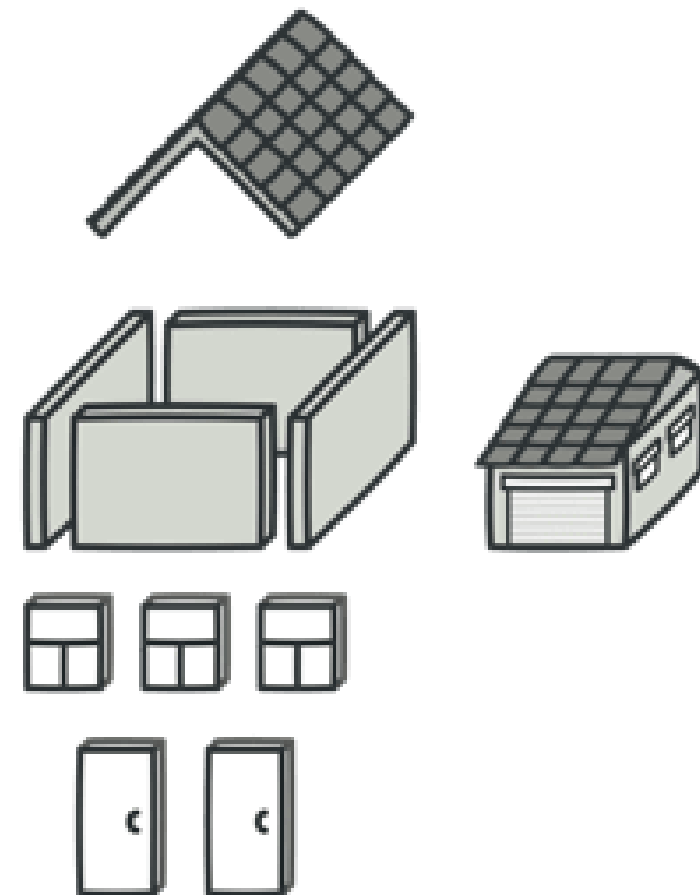
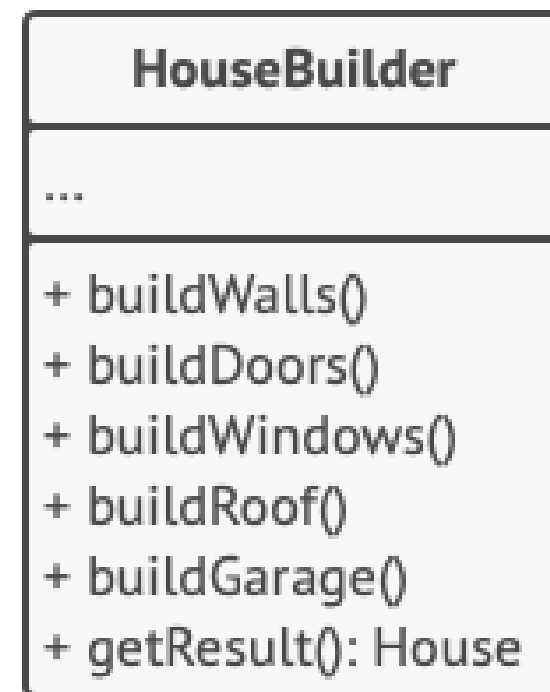
## Problema



*O construtor com vários parâmetros tem um lado ruim: nem todos os parâmetros são necessários todas*

# BUILDER

## Solução



*O padrão Builder permite que você construa objetos complexos passo a passo. O Builder não permite que outros objetos acessem o produto enquanto ele está sendo construído.*

# BUILDER

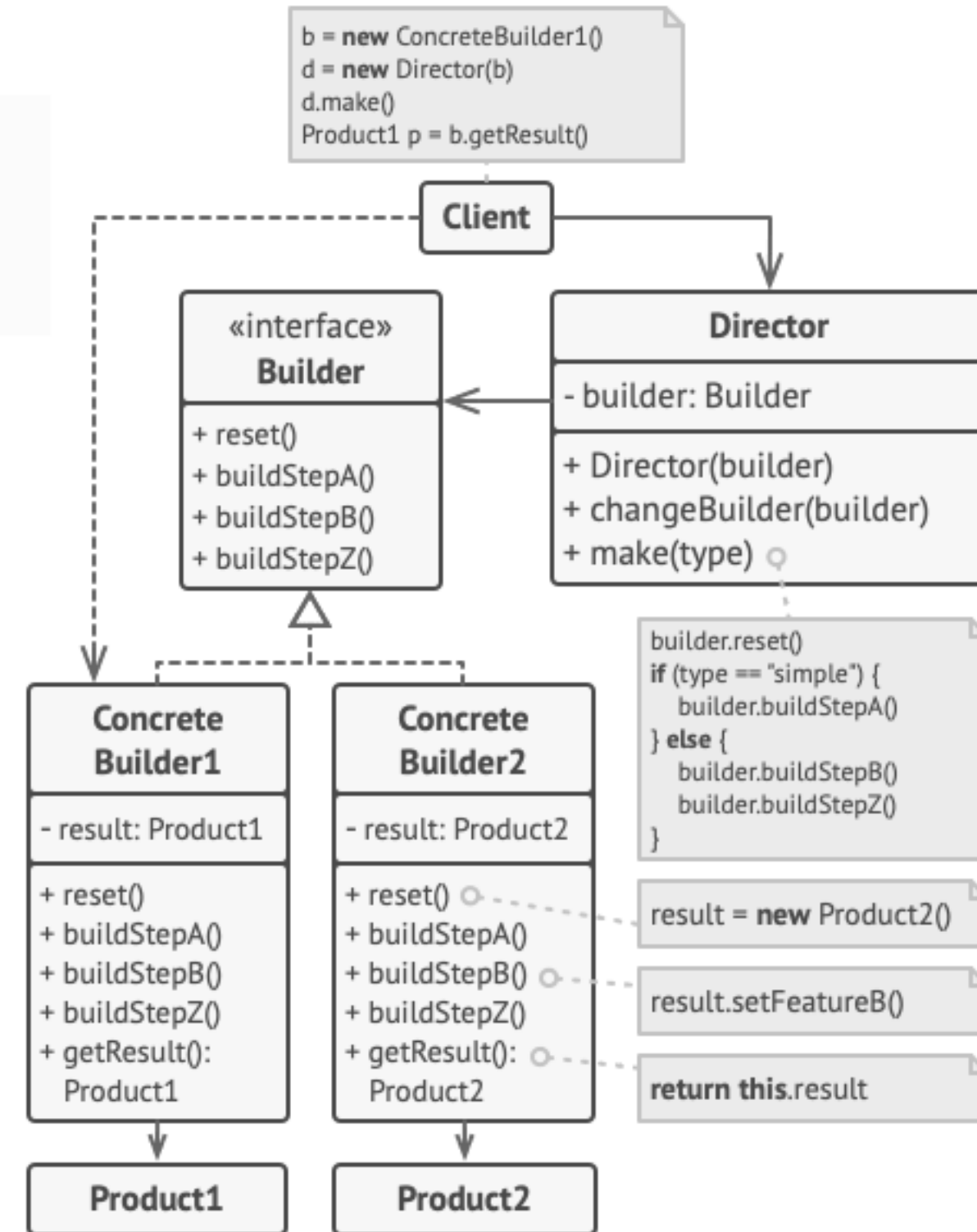
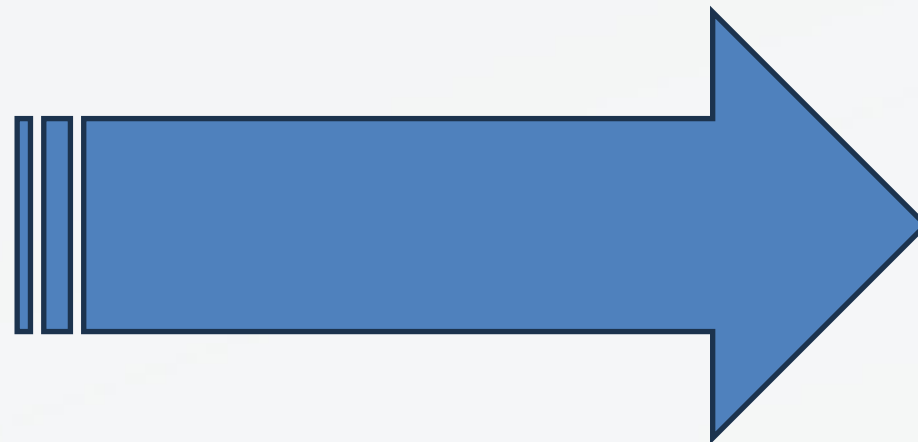
O padrão **Builder** sugere que se extraia o código de construção do objeto para fora de sua própria classe e mova ele para objetos separados chamados *builders*.



Pode-se extrair uma série de chamadas para as etapas do builder em uma classe separada chamada *diretor*. A classe **diretor define a ordem na qual executar as etapas de construção**, enquanto que o **builder provê a implementação dessas etapas**.



# BUILDER



# BUILDER



<https://refactoring.guru/pt-br/design-patterns/builder>



# BUILDER

## Prós e contras

- ✓ Você pode construir objetos passo a passo, adiar as etapas de construção ou rodar etapas recursivamente.
- ✓ Você pode reutilizar o mesmo código de construção quando construindo várias representações de produtos.
- ✓ *Princípio de responsabilidade única.* Você pode isolar um código de construção complexo da lógica de negócio do produto.
- ✗ A complexidade geral do código aumenta uma vez que o padrão exige criar múltiplas classes novas.

# REFERÊNCIAS BIBLIOGRÁFICAS

E. Gamma and R. Helm and R. Johnson and J. Vlissides. Design Patterns - Elements of Reusable Object-Oriented Software. AddisonWesley, 2000.

FREEMAN, E. Use a cabeça! - padrões de projeto (design patterns). 2. ed. Rio de Janeiro: Alta Books, 2007.

<https://refactoring.guru/pt-br/design-patterns>

