

Builder

Padrão de Projeto de Criação

INTEGRANTES: Arthur Monteiro, Brenda Stefany, Bruna Letícia, Elenice Florentina,
Eduarda Faria e Marco Aurélio de Faria

01

O que é:

Builder é um padrão de projeto criacional que permite criar objetos complexos passo a passo, separando a construção da representação.

Problema:

Construtores com muitos parâmetros (muitos opcionais) tornam o código confuso e inflexível.

Solução:

Usar um Builder com métodos encadeados para montar o objeto de forma clara e flexível.

// Sem Builder - difícil de ler

```
new Veiculo("Carro", "Tesla", "Model 3", "Branco", 2025, true, true, true);
```

// Com Builder - legível e flexível

```
new Veiculo.Builder("Carro", "Tesla", "Model 3")
    .comCor("Branco")
    .doAno(2025)
    .eletrico()
    .comSom()
    .build();
```

03

Estrutura

- **Product (Produto):** objeto complexo final.
- **Builder (Construtor):** define os passos de construção.
- **Concrete Builder (Construtor Concreto):** implementa os passos do Builder.
- **Director (Opcional):** organiza a sequência de construção.
- **Client (Cliente):** usa o Builder para criar o objeto passo a passo.

O Client inicia a construção, o Concrete Builder implementa os passos do Builder, o Director (opcional) organiza a ordem, e ao final obtemos o Product completo, passo a passo e de forma flexível.

04

Vantagens

- Clareza e Legibilidade
- Flexibilidade
- Controle
- Manutenibilidade

Desvantagens

- Complexidade

Conclusão

- O Builder simplifica a criação de objetos complexos, eliminando construtores longos e confusos.
- É ideal quando há muitos atributos opcionais e várias combinações possíveis.
- Não é recomendado para objetos simples, pois adiciona complexidade desnecessária.