O **Builder** é um padrão que nos permite construir objetos complexos passo a passo. O padrão permite que você produza diferentes tipos e representações de um objeto usando o mesmo código de construção.

O problema acontece quando temos um objeto complexo que necessite de uma inicialização passo a passo trabalhosa de muitos campos e objetos agrupados. Tal código de inicialização geralmente fica enterrado dentro de um construtor monstruoso com vários parâmetros. Ou pior: espalhado por todo o código cliente.

Por exemplo, vamos pensar em como criar uma classe Casa: Para construir uma classe simples, precisamos de quatro paredes, um piso, instalar uma porta, janelas e um teto. Mas se quisermos uma casa maior e mais iluminada, com um jardim e outras coisas, como sistema de aquecimento e fiação elétrica, essa classe já não irá mais servir.

A solução mais simples seria estender a classe base e criar outras subclasses para cobrir todas as combinações possíveis de parâmetros, porém eventualmente haveriam muitas subclasses, pois cada parâmetro nos forçaria a aumentar essa hierarquia cada vez mais.

Outra solução seria criar um construtor enorme dentro da classe, com todos os possíveis parâmetros, mas como a maioria deles não seria utilizada, as chamadas do construtor seriam algo feio de se ver e difícil de interpretar.

A solução proposta pelo padrão é de extrair o código de construção do objeto para fora da sua classe e colocar ele dentro de objetos separados chamados “builders”. O padrão organiza o processo de construção em uma série de etapas e, para criar um objeto você executa uma série de etapas em um objeto builder. A parte importante é que no builder nós **não precisamos** **chamar todas as etapas.** Chamamos apenas as necessárias para a produção de uma configuração específica do objeto.

Podemos ir além e extrair uma série de chamadas para as etapas do builder que utilizamos para construir um produto em uma classe separada chamada *diretor*. A classe diretor define a ordem na qual as etapas serão executadas, enquanto o builder provê a implementação dessas etapas.

Os pros e contras são semelhantes aos dos outros padrões criacionais.