

Universidade Tecnológica Federal do Paraná

Engenharia de Software - Curso de Arquitetura de Software (AS27S)

INSTRUTOR: Prof. Dr. Gustavo Santos

Diego Domingos Contini de Lima, 2617897

CCH - Design Patterns [Builder]

Problema

No contexto do desenvolvimento mobile, utilizando Flutter especificamente, precisamos componentizar nossos Widgets (componentes) e com isso, trazendo vários parâmetros opcionais. O que, por muitas vezes, dificulta a manutenção, e até mesmo reaproveitamento desta componentização.

Descrição da Solução

Com o padrão Builder, trazemos uma maior clareza na criação do componente. Onde poderemos definir o que de fato importa na criação do objeto, e possibilitando um entendimento maior do que está acontecendo no código.

Exemplo de Código [linguagem de programação da sua preferência]

<https://github.com/diegocontini/builder-pattern-flutter>

Código disponível na pasta /lib.

Com o padrão proposto, fica nítido como o componente está sendo criado.

Com um nome, com uma imagem, com uma descrição na bio, com uma cor de background diferente, e uma cor de foreground diferente também.

```
child: ProfileCardBuilder()  
  .withName('John Doe')  
  .withImageUrl('https://picsum.photos/200')  
  .withBioDescription('Flutter Developer')  
  .withBackgroundColor(Colors.blueAccent)  
  .withForegroundColor(Colors.white)  
  .build(),
```

Consequências [vantagens e desvantagens]

- Vantagens
 1. Facilidade de leitura, está claro o que está acontecendo
 2. Evitar múltiplos construtores (ex: Card.WithNameAndImageAndBio..., Card.WithNameAndImage..)
 3. Extensão. Caso seja necessário algum atributo novo, basta criar um novo método com a propriedade desejada. Sem impactar diretamente código antigo
- Desvantagens
 1. Maior complexidade
 2. Muito boilerplate para componentes simples