

# **Funcionamento do padrão de projetos builder**

**Leonardo Baltazar Neves (UTP)**

**Universidade Tuiuti Do Paraná - (UTP)**

Rua Sydnei Antonio Rangel Santos, nº 238 - Santo Inácio - Campos Barigui Curitiba – PR

## **Resumo**

O padrão de projeto builder se utiliza como um facilitador, esse padrão faz com que seja mais fácil se criar objetos, em resumo esse padrão é como ter um construtor externo ao objeto.

Palavras chaves: Facilidade, Padrões.

## **ABSTRACT**

The builder design pattern is used as a facilitator, this pattern makes it easier to create objects, in short, this pattern is like having a constructor external to the object.

Keywords : Ease, Patterns.

## 1 Introdução

O padrão de Projectos builder começa com uma classe que tem o construtor vazio e metodos do tipo set, claro que podemos ter outros metodos mas essa e sua base.

```
class Exemplo {  
private:  
    int _a;  
    string _b;  
public:  
    void setA(int a):_a(a){}  
    void setB(string b): _b(b){}  
};
```

Depois temos uma classe normalmente com o nome da classe mais builder, essa classe tem a função de ser o construtor real dos objetos, dentro dessa classe como atributo privado temos classe a ser construída, também métodos set públicos que retornam o próprio objeto, por fim ele terá um método build que retorna o atributo da classe base por cópia, é muito importante não esquecer de fazer a instanciação do objeto passado como atributo no construtor da classe builder.

```
class ExemploBuilder{  
private:  
    Exemplo _exemplo;  
public:  
    ExemploBuilder(): _exemplo(Exemplo()){}  
    ExemploBuilder *setA(int a){  
        _a = a;  
        return this;  
    }  
    ExemploBuilder *setB(string b){  
        _b = b;  
        return this;  
    }  
    Exemplo build(){  
        return _exemplo;  
    }  
};
```

O Builder cria uma vantagem de facilidade para instancia de um objeto também da um maior controle sobre ele, fazendo com que os programadores não precisem saber a composição da classe ele usa de acordo com a necessidade, pode não parecer muito na imagem acima mas, se tiver mais atributos e objetos internos podemos ter uma ideia melhor de sua função.

Explicando de uma forma melhor é como ter uma classe que pode começar vazia, mas em certos casos ela tem padrões como começar com um ou mais parâmetros setados, como não

a garantias que ele sempre será assim, podemos criar um Builder para fazer o padrão e já adicionar as partes extras.

Quando temos objetos internos nem sempre temos que usalos, então podemos criar padrões builder que instanciem ou não o objeto deixando mais leve e performática já que se economizar o tempo de instancia do objeto interno, conforme o sistema cresce podemos ter com o buider uma manipulação mais precisa do objeto.