

# O Padrão de Projeto **Builder**

Daniel Pereira Poltronieri  
11218029

Universidade Federal da Paraíba

2016.1

# Repositório

## GitHub

- <https://github.com/arashisann/builder>

# Introdução

## Porque?

- *Um objeto precisa de muitos argumentos em seu construtor.*
- *Um objeto precisa de muitos possíveis construtores diferentes, e/ou sofre de construtores telescópicos.*
- *Um problema pode começar como um Factory, mas precisa de mais flexibilidade ou de um código mais enxuto.*
- *Para melhorar a legibilidade.*
- *Semelhante ao Factory e ao Abstract Factory, para facilitar a criação de uma família de objetos.*

# Introdução

## Como?

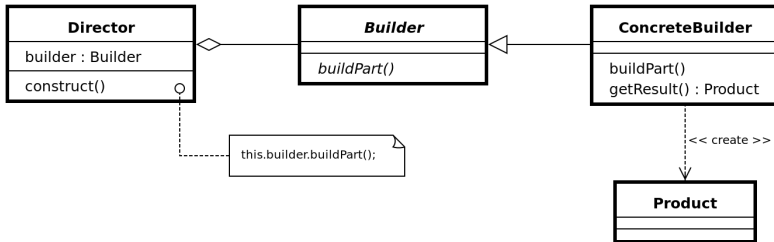
- *Separando a construção de um objeto de sua representação.*
- *Um Diretor orienta um Montador (Builder) a montar objeto uma parte de cada vez.*

# Dificuldades

- *Por ser muito próximo de Factory e Abstract Factory, é difícil escrever um código puramente Builder.*
- *As fontes não concordam entre si, e algumas vezes se contradizem nos exemplos.*

# Diagrama UML

Estrutura do padrão Builder.



# Exemplo

```
Veiculo::Builder mecanico;
```

```
Veiculo moto = mecanico.setRodas(2)  
.setCavalos(15).setCor("Azul").build();  
moto.imprimeDados();
```

```
Veiculo carro = mecanico.setCarroBranco()  
.build();  
carro.imprimeDados();
```

```
Veiculo carreta = Veiculo(12, 800, "Prata");  
carreta.imprimeDados();
```

# O Objeto

```
class Veiculo {  
public:  
    class Builder;  
  
    void setRodas(size_t num);  
    void setCor(std::string cor);  
    void setCavalos(size_t cavalos);  
  
    Veiculo(size_t num,  
            size_t cavalos ,  
            std::string cor);  
  
    void imprimeDados(void);
```



# O Objeto, cont...

```
private:
```

```
    size_t _numero_rodas, _cavalo_vapor;  
    std::string _cor;  
};
```

# Builder

```
class Veiculo::Builder {  
public:  
  
    Builder() :  
        _numero_rodas { 0 },  
        _cavalo_vapor { 0 },  
        _cor { "Indefinido" } {}  
  
    Builder& setRodas(size_t num) {  
        _numero_rodas = num;  
        return *this;  
    }  
}
```

# Concrete Builder

```
Builder& setCarroBranco() {  
    _numero_rodas = 4;  
    _cor          = "Branco";  
    _cavalo_vapor = 150;  
    return *this;  
}
```

# Builder

```
Veiculo build() {  
    return Veiculo(_numero_rodas,  
                   _cavalo_vapor,  
                   _cor);  
}
```

private:

```
    size_t _numero_rodas, _cavalo_vapor;  
    std::string _cor;  
};
```

# Dúvidas?

# Obrigado!