

Instituto Federal Catarinense (*Campus Blumenau*)

Professor: Ricardo de La Rocha Ladeira

Matéria: Padrões de Projeto

Nomes: Gabrielli Danker

Turma: BCC 2025.1

Data de entrega: 20 de Março de 2025

### Exercícios

- 1) Observando o exemplo 1, quem são Target, Adapter e Adaptee?

**RESPOSTA:**

**Target :** ConfigReader

```
interface ConfigReader {  
    String readConfig();  
}
```

**Adapter:** ConfigAdapter

```
class ConfigAdapter extends ConfigFileLegacy implements ConfigReader {  
    @Override  
    public String readConfig() {  
        return loadOldFormat();  
    }  
}
```

**Adaptee:** ConfigFileLegacy

```
class ConfigFileLegacy {  
    String loadOldFormat() {  
        return "Configuração carregada do formato legado.";  
    }  
}
```

- 2) Crie um outro exemplo de projeto simples que faça uso do padrão *Adapter*. Escolha livremente entre adaptador de classe ou de objeto.

**RESPOSTA:** Adaptador de objeto.

```
interface processadorPag {  
    void processoPagamento(double valorPagamento);  
}  
  
class pagamentoAdapter implements processadorPag {  
    private antigoSistemaPagamento antigoSistemaPag;  
    public pagamentoAdapter(antigoSistemaPagamento antigoSistemaPag) {
```

```

        this.antigoSistemaPag = antigoSistemaPag;
    }
    public void processoPagamento(double valorPagamento){
        antigoSistemaPag.pagar("BRL", valorPagamento);
    }
}

class antigoSistemaPagamento {
    void pagar (String carencia, double valorPagamento){
        System.out.println("Pagamento realizado: " + valorPagamento + "
" + carencia);
    }
}

class adapterExemplo {
    public static void main(String[] args) {
        antigoSistemaPagamento antigoSistema = new
antigoSistemaPagamento();
        processadorPag processadorPagamento = new
pagamentoAdapter(antigoSistema);

        processadorPagamento.processoPagamento(150.00);
    }
}

```