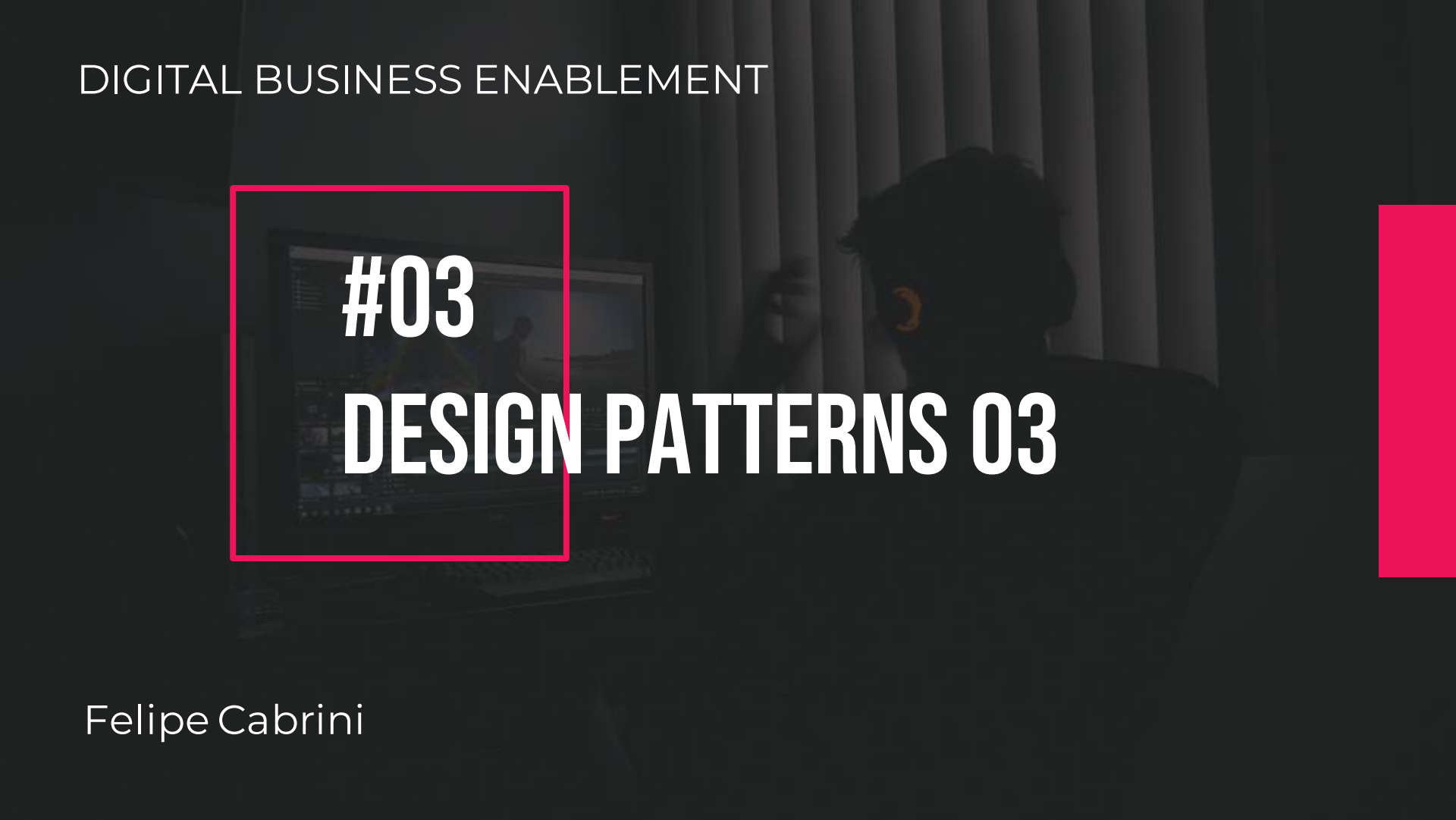
DIGITAL BUSINESS ENABLEMENT

#03
DESIGN PATTERNS 03



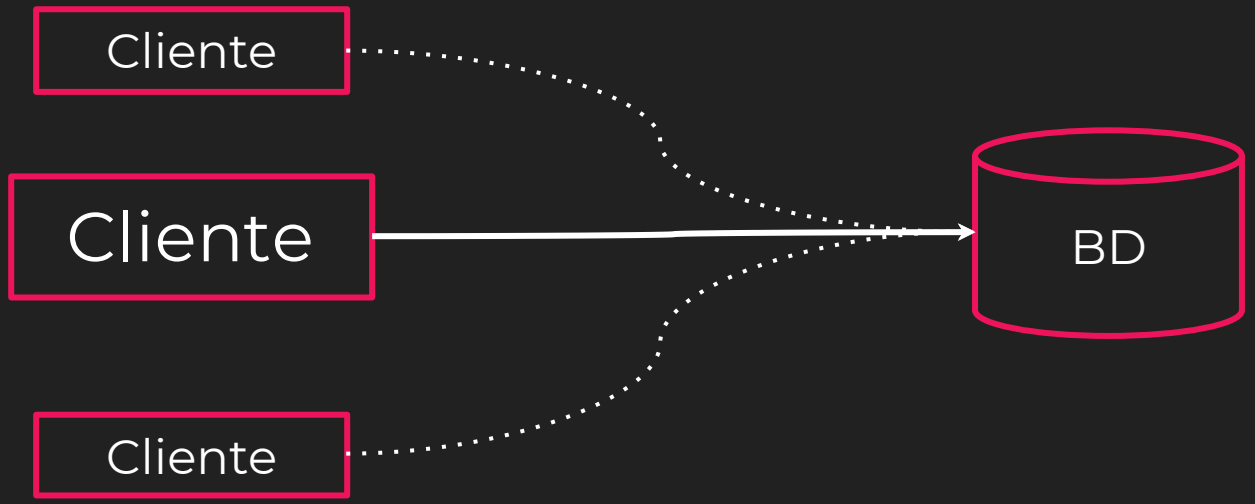
Felipe Cabrini

PROXY

" Padrão de projeto estrutural que permite que você forneça um substituto ou um espaço reservado para outro objeto. Um proxy controla o acesso ao objeto original, permitindo que você faça algo ou antes ou depois do pedido chegar ao objeto original.

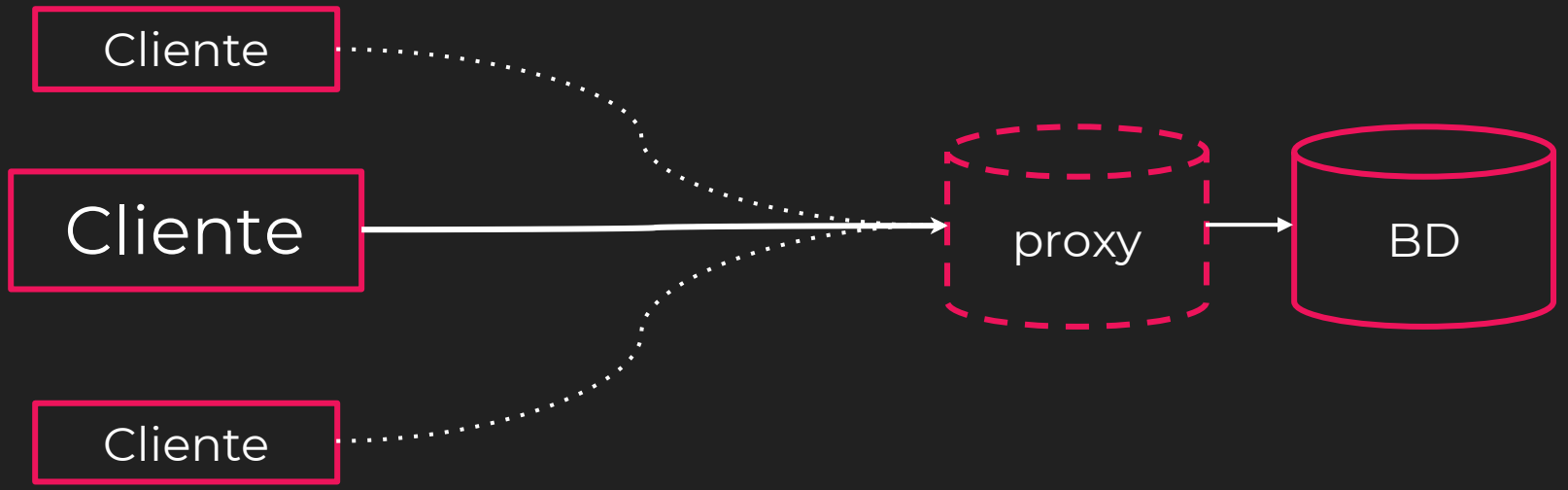
PROXY

O Problema



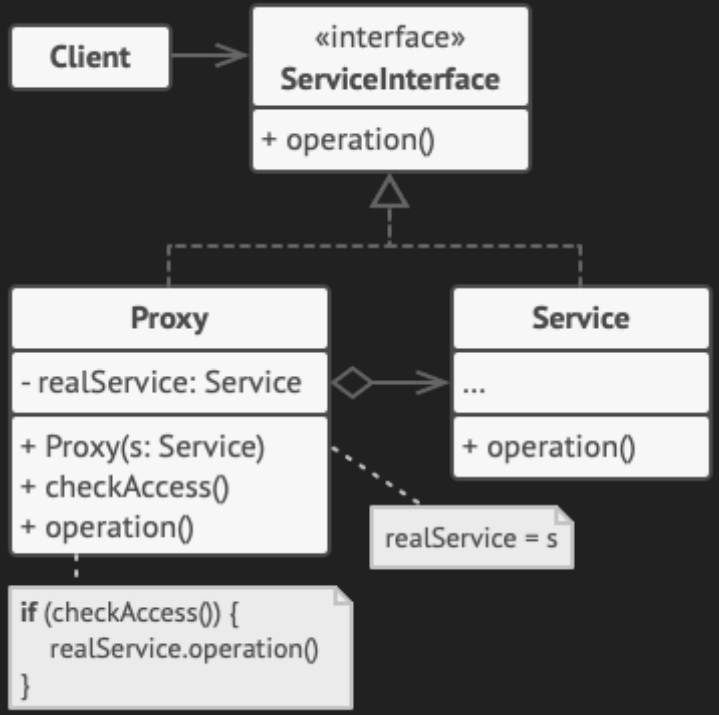
PROXY

A Solução



PROXY

A Solução



PROXY

Vamos ao código

Problema: Em um sistema de arquivos, queremos controlar o acesso a determinados arquivos. Alguns arquivos devem ser apenas para leitura, enquanto outros podem ser editados.

PROXY

Vamos ao código

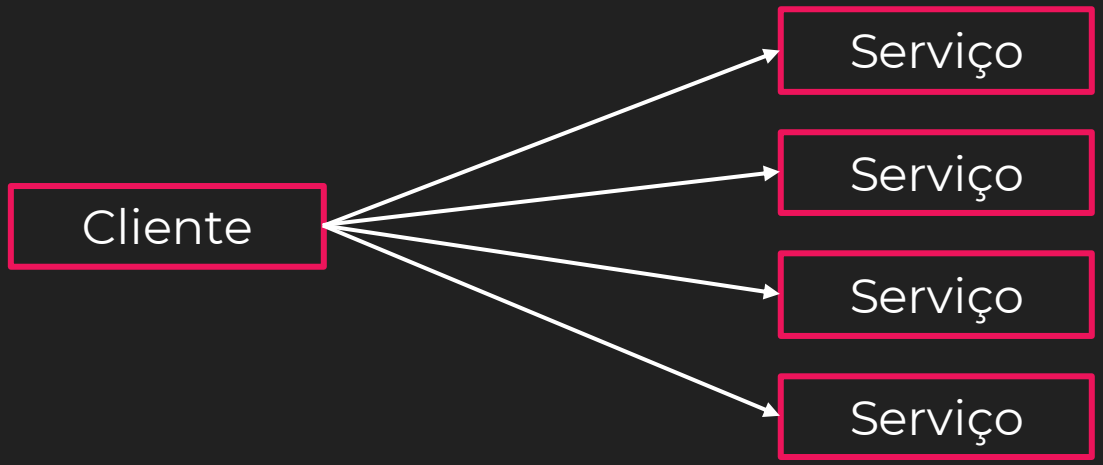
Solução: Usar o padrão Proxy para criar um intermediário que controla o acesso ao arquivo real.

FACADE

> Padrão de projeto estrutural que fornece uma interface simplificada para uma biblioteca, um framework, ou qualquer conjunto complexo de classes.

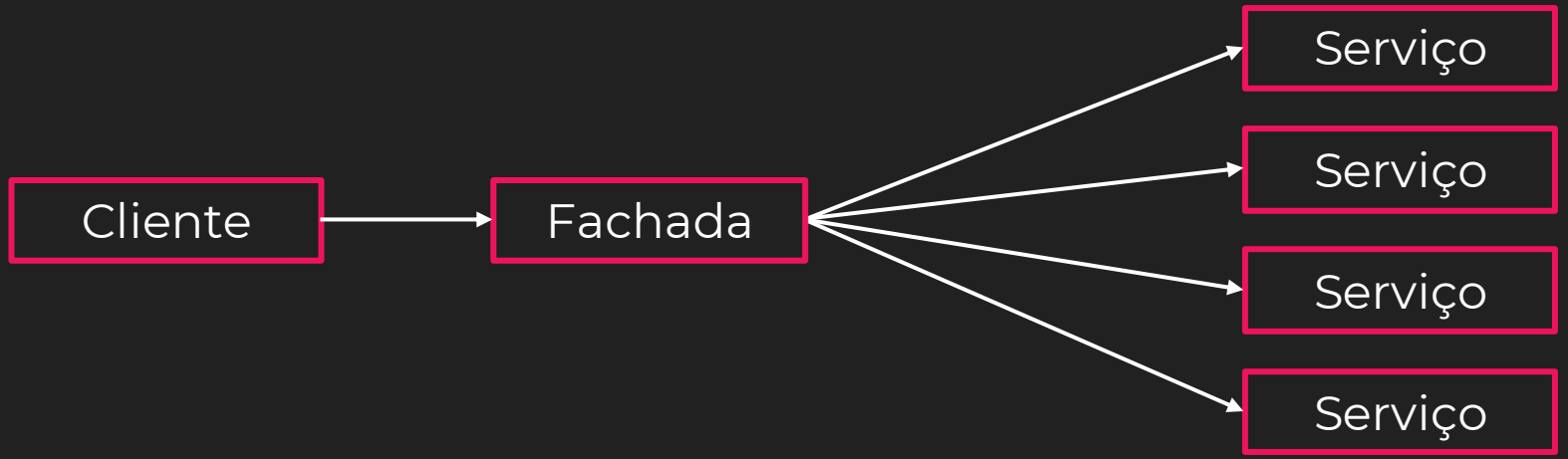
FACADE

O Problema



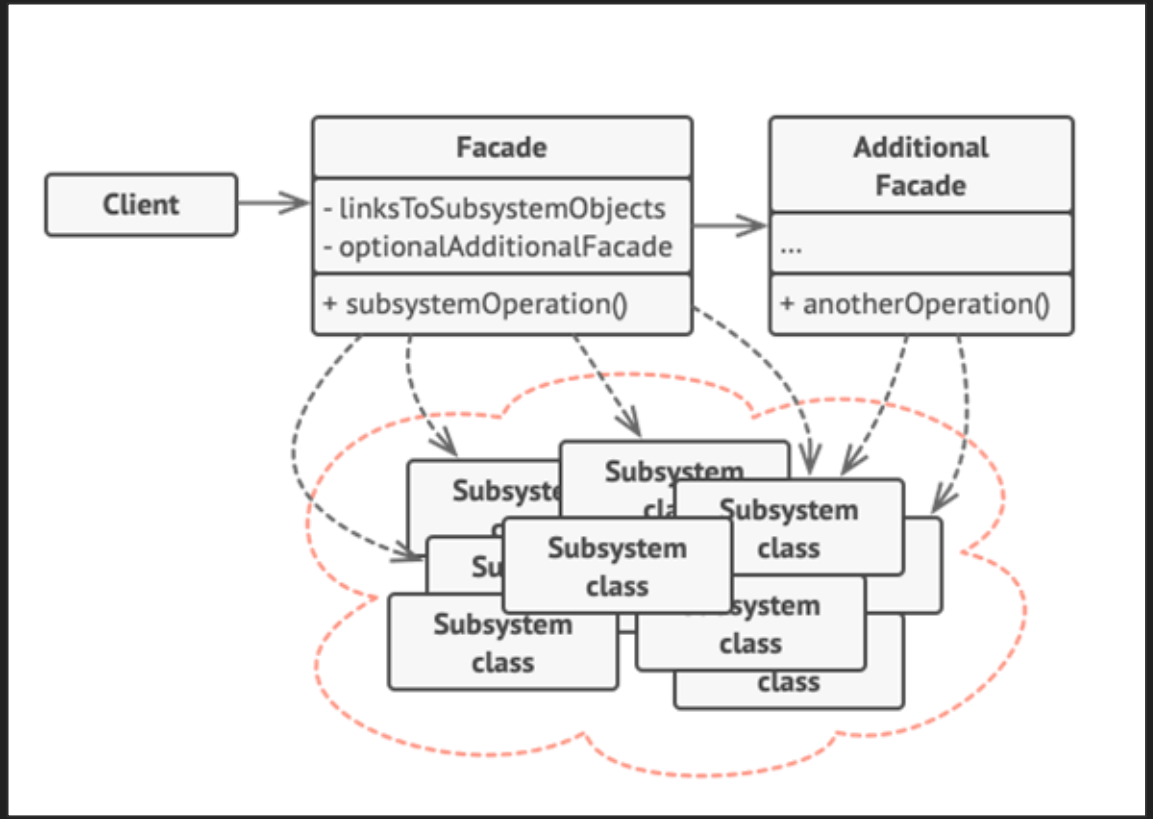
FACADE

A Solução



FACADE

A Solução



FACADE

Vamos ao código

Problema: Em um sistema de automação residencial, há diversos sistemas separados: iluminação, climatização e sistema de som. O usuário quer uma maneira fácil de inicializar todos esses sistemas ao chegar em casa.

FACADE

Vamos ao código

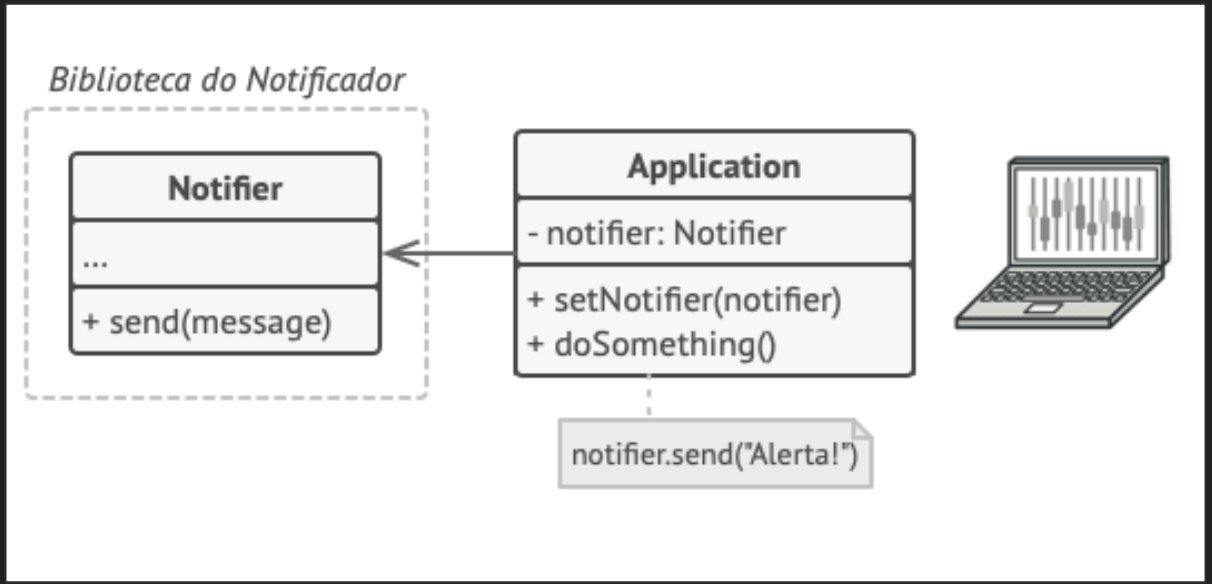
Solução: Usar o padrão Facade para fornecer uma interface unificada para estes sistemas.

DECORATOR

" Padrão de projeto estrutural que permite que você acople novos comportamentos para objetos ao colocá-los dentro de invólucros de objetos que contém os comportamentos.

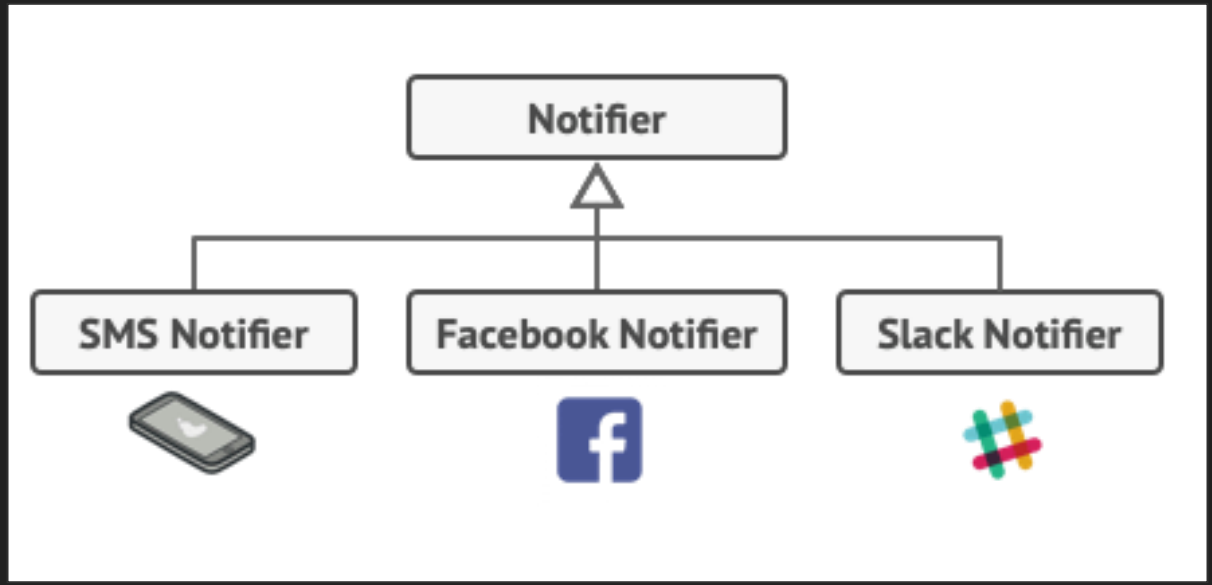
DECORATOR

O Problema



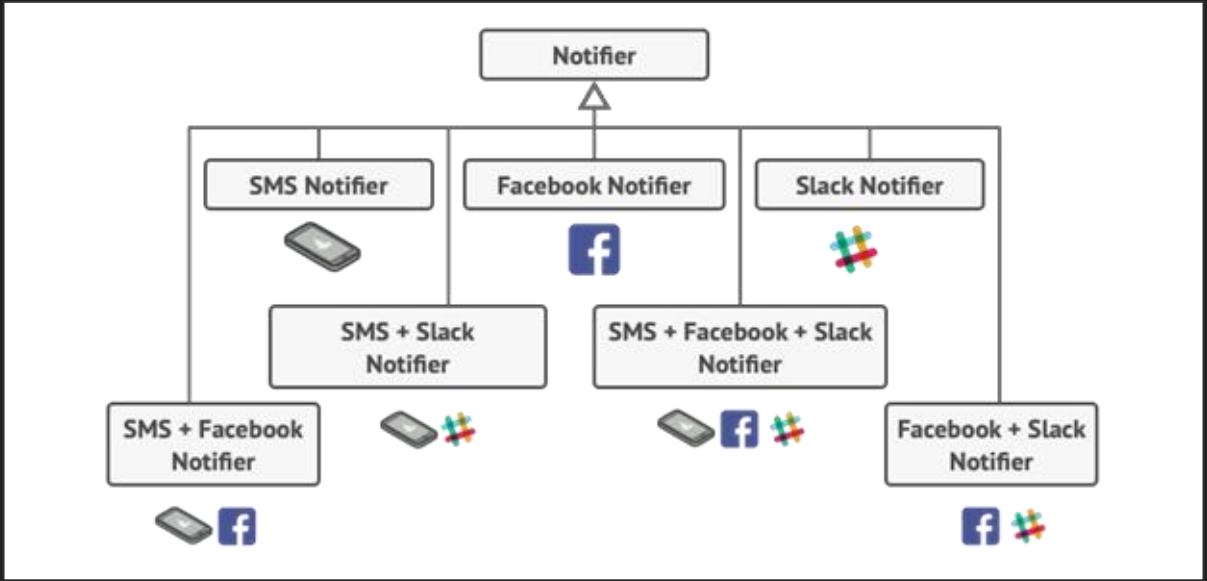
DECORATOR

O Problema



DECORATOR

O Problema



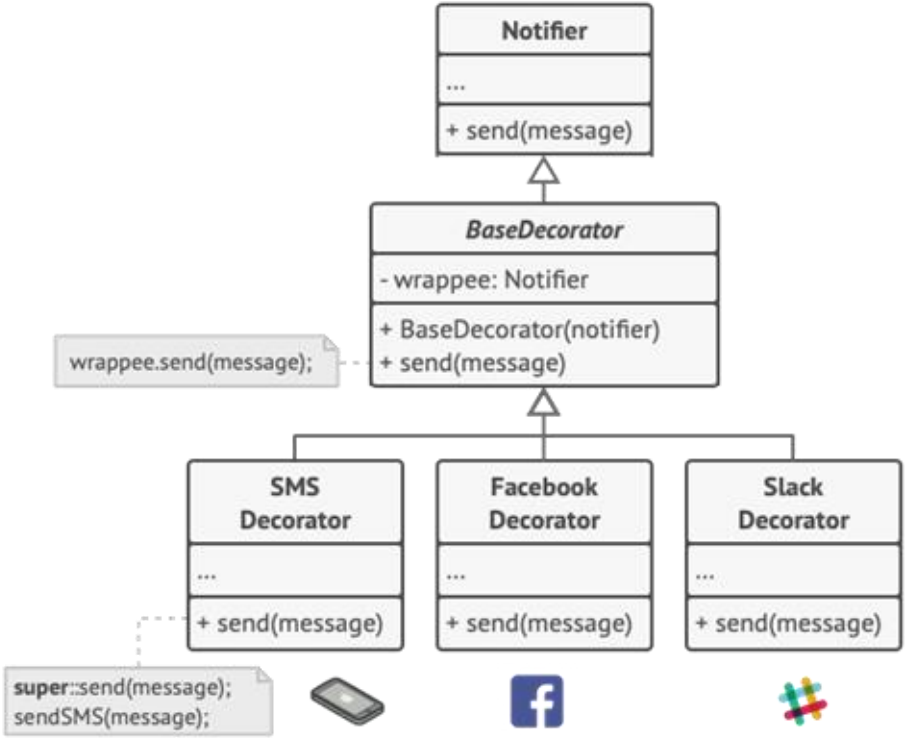
DECORATOR

A Solução

```
stack = new Notifier()
if (facebookEnabled)
    stack = new FacebookDecorator(stack)
if (slackEnabled)
    stack = new SlackDecorator(stack)

app.setNotifier(stack)
```

```
notifier.send("Alerta!")
// Email → Facebook → Slack
```



DECORATOR

Vamos ao código

Problema: Em um sistema de café, os clientes podem escolher diversos adicionais (como leite, chocolate, caramelo). Precisamos calcular o preço com base nos adicionais selecionados.

DECORATOR

Vamos ao código

Solução: Usar o padrão Decorator para adicionar comportamentos em tempo de execução.

VAMOS REFLETIR

VAMOS REFLETIR


Decorator


Facade


Proxy


Chain of Responsibility


Observer


State


Template Method


Strategy

VAMOS REFLETIR


Decorator


Facade


Proxy


Chain of Responsibility


Observer


State


Template Method


Strategy

VAMOS REFLETIR


Decorator


Facade


Proxy


Chain of Responsibility


Observer


State


Template Method


Strategy

VAMOS REFLETIR

 Decorator

 Facade

 Proxy

 Chain of Responsibility

 Observer

 State

 Template Method

 Strategy