

# Universidade Tecnológica Federal do Paraná

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

INSTRUTOR: Prof. Dr. Gustavo Santos

Felippe Negrão de Oliveira, 2372517

---

## CCH - Design Patterns Proxy

### Problema

Ao desenvolver uma aplicação que se conecta a um serviço remoto para obter dados, como uma API de terceiros ou um banco de dados remoto cada vez que a aplicação solicita dados, há um custo de tempo e recursos devido à latência da rede e ao processamento do servidor. Frequentemente, os dados solicitados não mudam com muita frequência e podem ser reutilizados sem precisar fazer uma nova solicitação ao servidor.

### Descrição da Solução

Para resolver este problema, podemos usar o padrão de design Proxy. O padrão Proxy fornece um substituto ou um placeholder para outro objeto para controlar o acesso a ele. Podemos criar um proxy que gerencia o acesso ao serviço remoto, armazenando em cache os dados obtidos e servindo-os de maneira eficiente quando solicitados novamente. Dessa forma, podemos reduzir o número de chamadas ao serviço remoto e melhorar o desempenho da aplicação.

---

## Visão Geral

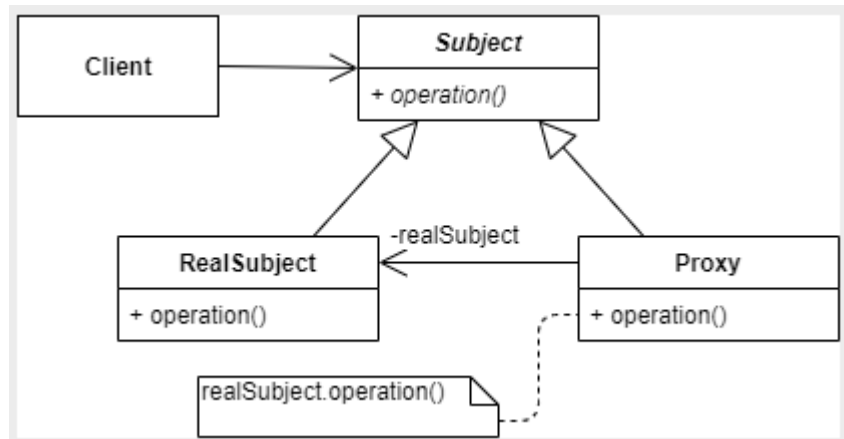


Diagrama de Classe do Proxy

## Exemplo de Código - Dart

O código pode ser encontrado no seguinte link:

[https://github.com/FelippeNO/design\\_patterns\\_utfpr/blob/main/proxy\\_example.dart](https://github.com/FelippeNO/design_patterns_utfpr/blob/main/proxy_example.dart)

## Consequências

- Vantagens
  - Desempenho Melhorado.
  - Controle de Acesso.
  - Encapsulamento.
- Desvantagens
  - Complexidade Adicional.