

IFNMG - Instituto Federal do Norte de Minas Gerais  
Campus Montes Claros

Filipe Abner Soares Melo

**SEMINÁRIO**  
**Design Patterns – Proxy**

Montes Claros - MG  
10 de junho de 2022

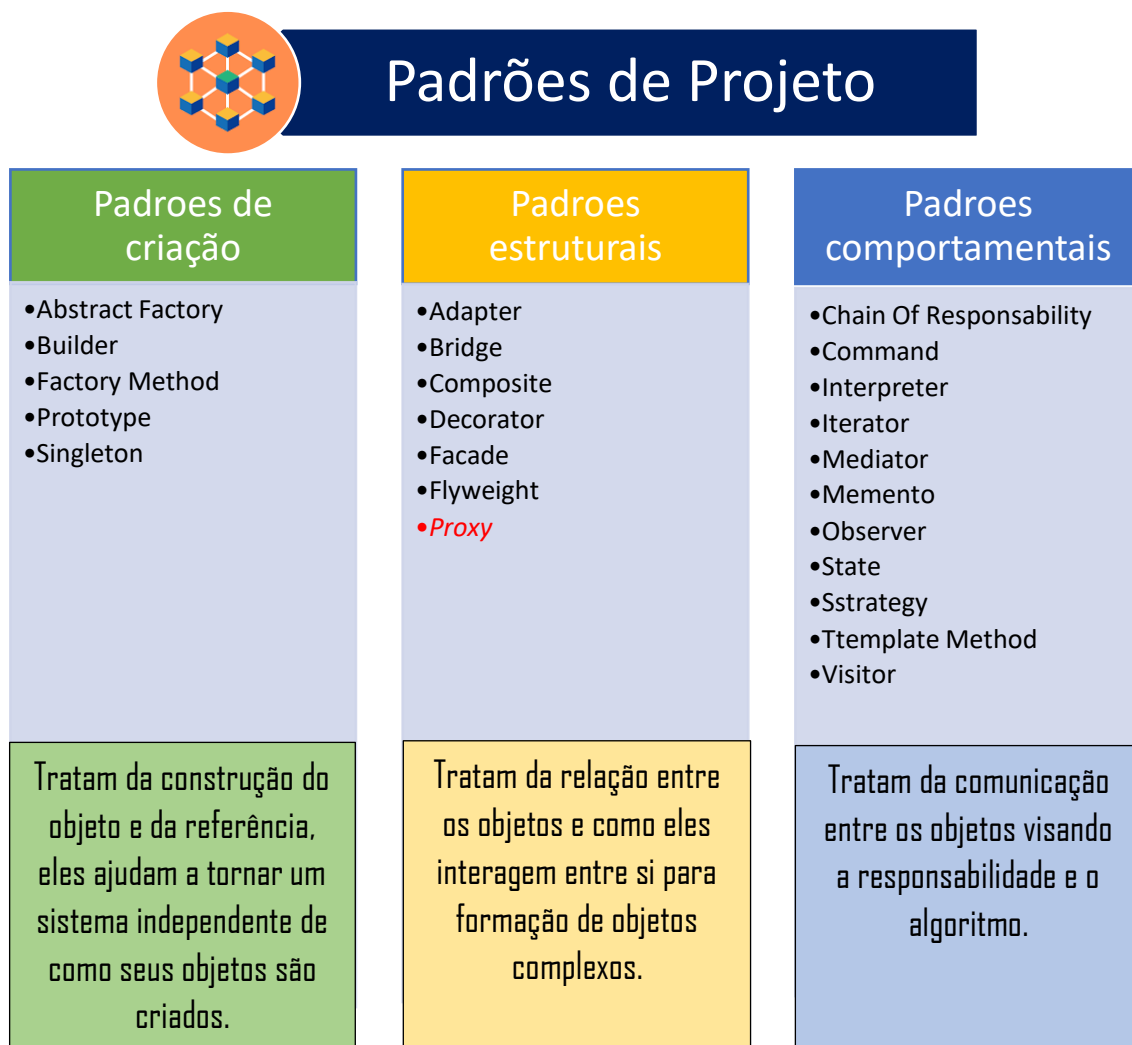
## 1. Introdução

O termo Design Patterns surgiu após os Engenheiros de software Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides escreverem um livro chamado “Design Patterns: Elements of Reusable Object-Oriented Software” esse livro trata de padrões para implementação de diferentes tipos de soluções.

Design Patterns(Padrões de projeto) constitui em padrões para soluções de problemas recorrentes durante a elaboração de programas que utilizam orientação a objetos, padronizando as soluções de maneira que facilite a vida do desenvolvedor.

As grandes vantagens de se usar um padrão de projeto está no ganho de produtividade, pois facilita o entendimento do código pelas demais pessoas que trabalham no projeto além de já apresentar uma solução pronta para um problema, e pela organização e manutenção dos projetos uma vez que há um acoplamento entre as classes e padronização do código.

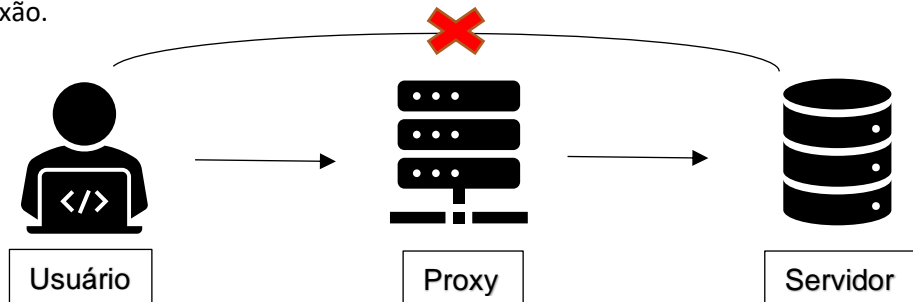
O diagrama a baixo demonstrará a divisão dos Designs Patterns.



Com o passar do tempo inúmeras outras formas de padronização foram criadas, foram listadas aqui as principais

## 2. Design Pattern - Proxy.

De maneira simplificada, o proxy trabalha como um filtro de informações da rede para com o usuário, ele recebe as informações do objeto e simula ser o próprio objeto para se conectar com o banco de dados, mascarando assim as informações que identificam o usuário e sua conexão.



Uma classe proxy dinâmica é uma classe que implementa uma lista de interfaces especificadas em tempo de execução, de modo que uma invocação de método através de uma das interfaces em uma instância da classe será codificada e despachada para outro objeto através de uma interface uniforme.

Quando um cliente se conecta ao servidor proxy, solicitando algum serviço, como um arquivo, conexão, página web ou outros recursos disponíveis de um servidor diferente, o proxy irá avaliar a solicitação visando simplificar e controlar sua complexidade. Os proxies foram inventados para adicionar estrutura e encapsulamento aos sistemas distribuídos. Esses servidores têm diversos usos, como filtrar conteúdo, providenciar anonimato, entre outros.

Os proxies podem agir de diversas maneiras de acordo com as informações recebidas, podendo restringir algum conteúdo da web (muito utilizado em escolas e bibliotecas), podem mascarar o IP para garantir uma navegação segura, armazenam dados em cache o que torna a navegação mais rápida (muito utilizado em grandes empresas que possuem alto tráfego de dados), entre outros.

### Vantagens

- A classe a qual será implementada o proxy nem precisa saber que está usando um proxy (ele finge ser o objeto real)
- É possível adicionar novos proxies sem mudar o código já testado
- O proxy funciona mesmo se o objeto real não estiver operacional ou pronto para uso
- É possível controlar o ciclo de vida de objetos reais dentro do proxy

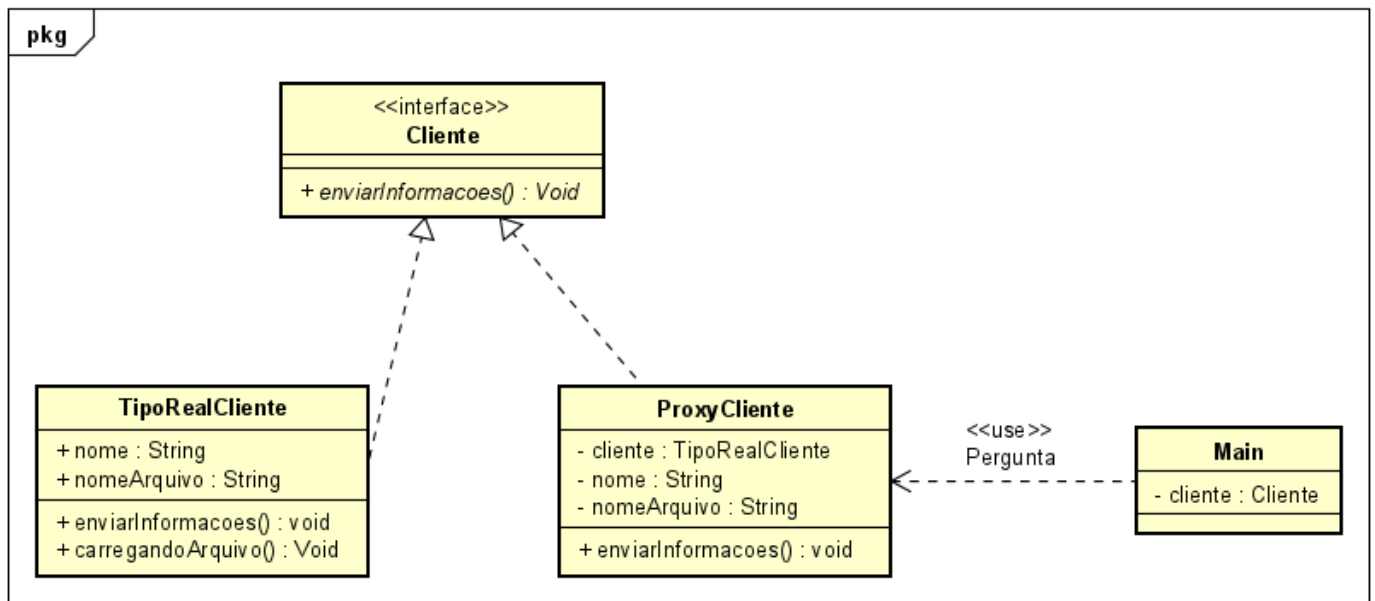
### Desvantagens

- Introduz mais classes ao sistema o que o deixa mais complexo

Portanto para esse trabalho iremos implementar um proxy que agirá como um cliente, recebendo as informações do usuário para enviá-las ao servidor. Abaixo temos o diagrama de classes relacionado ao projeto e o link de referência ao código do mesmo.

**Link Código GitHub:** <https://github.com/FilipeAbner/Seminario-POO>

**Link Vídeo YouTube:** <https://youtu.be/ZfVP3t97kGk>



## Referências

Miranda, Otávio. Proxy Teoria – Padrões de Projeto – Parte 23/45. YouTube, 12 set. 2020. 1 vídeo (15 min). Disponível em: <<https://youtu.be/EsxPylCeBPs>> Acesso em: 12 maio 2022.

Miranda, Otávio. Proxy Prática – Padrões de Projeto – Parte 24/45. YouTube, 13 set. 2020. 1 vídeo (25 min). Disponível em: <<https://youtu.be/xMV22kmnOo8>> Acesso em: 12 maio 2022.

Oracle. Dynamic Proxy Classes. 1993,2022. Disponível em: <<https://docs.oracle.com/javase/8/docs/technotes/guides/reflection/proxy.html>> Acesso em: 22 maio 2022.

Garcia, Nayara. Padrão de Projeto Proxy – implementação java. Youtube, 1 set. 2020. 1 vídeo (6 min). Disponível em: <<https://youtu.be/KMudd2R5Y4c>> Acesso em: 16 maio 2022

Vanessa, Weber. Fróes Gabriel. Design Patterns // Dicionário do Programador. YouTube, 27 ago. 2018. 1 vídeo (8 min). Disponível em: <<https://youtu.be/J-IHpiu-Twk>> Acesso em: 15 maio 2022

Assis, Semíramis. Implementando o Proxy Pattern. YouTube, 30 out 2015. 1 vídeo (15 min). Disponível em: <<https://youtu.be/4YWtUT4MKtw>> Acesso em: 16 maio 2022

Refactoring.Guru. Proxy. Disponível em: <<https://refactoring.guru/pt-br/design-patterns/proxy>> Acesso em: 15 maio 2022.

Opus Software. Design Patters – O que são e quais os benefícios. 26 set. 2018 .Disponível em: <<https://www.opus-software.com.br/design-patterns/>> Acesso em: 15 maio 2022.

Bruna B Barro. Servidor Proxy: O que é, Como Funciona e Para que Serve. 17 maio 2022. Disponível em: <<https://www.hostinger.com.br/tutoriais/servidor-proxy#Usos de um Servidor Proxy>> Acesso em: 15 maio 2022.

Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Design Patterns\_Elements of Reusable Object-Oriented Software. 1994. Disponível em: <<https://www.amazon.com.br/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612>> Acesso em: 12 maio 2022.