



**PUC Minas**



# **Resenha do Capítulo 7**

## ***Arquitetura***

*Resenha do capítulo 7*

*do livro Engenharia de Software Moderna de Marco Tulio Valente*

Júlia Medeiros Silva

Belo Horizonte, 2025

## **7.1 Introdução**

O Capítulo 7 do livro "Engenharia de Software Moderna" aborda a arquitetura de software, que pode ser definida como a estrutura de alto nível de um sistema, envolvendo decisões críticas que impactam sua evolução e manutenção. São apresentados diversos padrões arquiteturais e seus benefícios e desafios.

## **7.2 Arquitetura em Camadas**

A Arquitetura em Camadas é um dos padrões arquiteturais mais utilizados e propõe a organização do sistema em camadas hierárquicas. Cada camada só pode interagir com a imediatamente inferior, garantindo separação de responsabilidades e facilitando a manutenção. Um exemplo é a Arquitetura em Três Camadas, composta por:

- Interface com o Usuário (apresentação);
- Lógica de Negócio (processamento);
- Banco de Dados (armazenamento).

## **7.3 Arquitetura MVC**

O padrão MVC separa a interface gráfica da lógica do sistema, dividindo-o em:

- Modelo (dados e regras de negócio);
- Visão (apresentação e interface gráfica);
- Controlador (captura interações do usuário e atualiza Modelo e Visão).

## **7.4 Microserviços**

A arquitetura de microserviços fragmenta um sistema monolítico em serviços independentes, cada um executando em um processo separado. Benefícios incluem:

- Evolução independente: Times de desenvolvimento podem trabalhar de forma autônoma;
- Escalabilidade: Possibilidade de escalar apenas os serviços mais exigidos;
- Falhas parciais: falhas afetam apenas serviços individuais, não o sistema inteiro.

## **7.5 Arquiteturas Orientadas a Mensagens**

Nessa arquitetura, clientes e servidores se comunicam por meio de filas de mensagens, permitindo desacoplamento no espaço e no tempo. As vantagens incluem:

- Assincronicidade: Clientes podem continuar processando sem esperar resposta imediata;
- Escalabilidade: Possibilita balanceamento de carga distribuindo mensagens entre vários servidores.

## **7.6 Arquiteturas Publish/Subscribe**

Esse padrão permite que múltiplos assinantes sejam notificados quando um evento é publicado. A comunicação ocorre por meio de um broker de eventos, garantindo comunicação em grupo e notificação assíncrona.

## **7.7 Outros Padrões Arquiteturais**

- Pipes e Filtros: Baseado em processamento sequencial de dados, como no uso de pipes em Unix.
- Cliente/Servidor: Estrutura comum para sistemas distribuídos, como servidores de banco de dados e web.
- Peer-to-Peer (P2P): Arquitetura descentralizada onde nós atuam tanto como clientes quanto servidores, como em redes BitTorrent.

## **7.8 Anti-padrões Arquiteturais**

O anti-padrão Big Ball of Mud ocorre quando um sistema cresce de forma desorganizada, resultando em dependências excessivas e um código-fonte caótico. Isso dificulta a manutenção e a evolução do sistema. Um estudo de caso citado mostra um sistema bancário de 25 milhões de linhas de código que sofreu com esse problema, tornando a evolução lenta e burocrática.

## **Conclusão**

Este capítulo fornece uma visão abrangente sobre padrões arquiteturais, suas vantagens e desafios. A escolha de uma arquitetura apropriada impacta diretamente na escalabilidade, manutenção e desempenho do sistema. Além disso, a adoção de boas práticas evita que um software se transforme em uma big ball of mud, garantindo longevidade e qualidade ao sistema.