

# Resenha dos Capítulos 6 e 7 do livro Engenharia de Software Moderna

Júlia Moreira Nascimento  
Projeto de Software

October 14, 2024

## Introdução

Este documento apresenta uma análise crítica e um resumo dos capítulos 6 e 7, que tratam, respectivamente, dos *Padrões de Projeto* e da *Arquitetura de Software* no contexto da engenharia de software moderna. Ambos os temas são pilares fundamentais no desenvolvimento de sistemas de software escaláveis, flexíveis e manuteníveis. A seguir, serão abordados os principais conceitos de cada capítulo, seguidos de uma análise crítica.

## 1 Capítulo 6: Padrões de Projeto

O capítulo 6 discute os *Padrões de Projeto*, que são soluções recorrentes para problemas comuns no design de software. Eles fornecem uma maneira de reutilizar boas práticas, facilitando a criação de sistemas mais robustos e manuteníveis.

### 1.1 Conceitos Principais

Os padrões de projeto são agrupados em três categorias principais:

1. **Padrões de Criação:** Responsáveis por definir como os objetos são criados. Exemplos incluem o *Singleton* e o *Factory Method*.
2. **Padrões Estruturais:** Focam na organização das classes e objetos. O padrão *Adapter* é um exemplo clássico.
3. **Padrões Comportamentais:** Definem como os objetos interagem entre si. Um exemplo muito utilizado é o *Observer*.

Esses padrões são aplicáveis em diferentes contextos e auxiliam na padronização de soluções, promovendo reuso e eficiência.

## 1.2 Análise Crítica

Embora os padrões de projeto sejam amplamente úteis, o capítulo não aborda suficientemente os desafios associados ao seu uso excessivo ou inadequado. Em equipes menos experientes, há o risco de *overengineering*, o que pode resultar em sistemas desnecessariamente complexos. Seria interessante também discutir os critérios para selecionar o padrão apropriado para cada situação.

# 2 Capítulo 7: Arquitetura de Software

O capítulo 7 aborda a *Arquitetura de Software*, que define a estrutura e o comportamento de um sistema. A arquitetura funciona como um esqueleto para decisões de design e implementação.

## 2.1 Conceitos Principais

Diferentes estilos arquiteturais são discutidos, tais como:

- **Arquitetura em Camadas:** Organiza o sistema em camadas distintas, separando responsabilidades.
- **Cliente-Servidor:** Um modelo clássico onde há uma clara distinção entre clientes e servidores.
- **Microserviços:** Proporcionam maior granularidade e escalabilidade, embora com complexidade de gerenciamento.

Além disso, o capítulo destaca a importância de alinhar a arquitetura aos requisitos do negócio e à qualidade de atributos como desempenho, escalabilidade e manutenibilidade.

## 2.2 Análise Crítica

Este capítulo oferece uma boa visão geral sobre o papel da arquitetura na engenharia de software. Contudo, seria benéfico explorar mais profundamente as arquiteturas emergentes, como as baseadas em eventos ou *serverless*. A discussão sobre o impacto de decisões arquiteturais inflexíveis poderia ser ampliada com exemplos práticos.

# 3 Conclusão

Os capítulos 6 e 7 discutem dois aspectos fundamentais do desenvolvimento de software: os *Padrões de Projeto* e a *Arquitetura de Software*. Ambos desempenham papéis cruciais na criação de sistemas escaláveis, robustos e manuteníveis. A interseção entre padrões e arquitetura é evidente, pois o uso eficaz de padrões depende de uma arquitetura bem definida. Enquanto o capítulo sobre padrões de projeto poderia ter explorado mais os riscos de uso inadequado, o capítulo sobre arquitetura teria se beneficiado de uma análise mais detalhada de tendências arquiteturais modernas.