

Os capítulos 6 e 7 do livro Engenharia de Software Moderna abordam conceitos fundamentais para o desenvolvimento de software, com foco em padrões de projeto e arquitetura de software, respectivamente. Juntos, eles fornecem uma visão abrangente de como projetar sistemas robustos, flexíveis e escaláveis, enfatizando tanto o design detalhado quanto a organização estrutural de alto nível.

No capítulo 6, são apresentados diversos padrões de projeto, que são soluções reutilizáveis para problemas comuns no desenvolvimento. Inspirados pela obra da "Gangue dos Quatro" (GoF), esses padrões são classificados em três categorias: criacionais, estruturais e comportamentais. Entre os padrões discutidos estão o Singleton, que garante a existência de uma única instância de uma classe, o Strategy, que permite a troca dinâmica de algoritmos, e o Observer, útil para sistemas que precisam de notificação de mudanças em tempo real. O capítulo destaca que esses padrões não são apenas soluções técnicas, mas um vocabulário compartilhado entre desenvolvedores, facilitando a comunicação e a documentação. No entanto, alerta que o uso excessivo ou inadequado dos padrões pode gerar complexidade desnecessária, indo contra o princípio de simplicidade do design.

Já o capítulo 7 amplia o escopo para tratar da arquitetura de software, que é vista como um conjunto de decisões críticas e difíceis de reverter, influenciando diretamente a evolução e manutenção dos sistemas. São explorados vários padrões arquiteturais, como a Arquitetura em Camadas, que separa a aplicação em níveis hierárquicos, e o MVC (Model-View-Controller), que organiza o sistema em três componentes — modelo, visão e controlador — para facilitar a manutenção e especialização do trabalho. O capítulo também destaca as vantagens e desafios dos microsserviços, uma abordagem que permite que pequenos serviços independentes sejam desenvolvidos e implantados separadamente, promovendo escalabilidade e flexibilidade. No entanto, essa arquitetura traz complexidades, como maior latência e dificuldades no gerenciamento de transações distribuídas.

Ambos os capítulos convergem na ideia de que, para construir software de qualidade, é necessário pensar em mudança e evolução desde o início. Padrões de projeto oferecem soluções que aumentam a flexibilidade no nível do código, enquanto padrões arquiteturais ajudam a garantir que o sistema como um todo seja modular e preparado para crescer. A obra reforça que, embora existam boas práticas amplamente aceitas, não há uma solução única para todos os problemas. Cada decisão de design e arquitetura deve considerar o contexto do projeto e seus requisitos específicos, equilibrando simplicidade e funcionalidade. Com isso, o livro transmite uma mensagem clara: o sucesso no desenvolvimento de software depende tanto de uma boa estrutura de alto nível quanto de decisões precisas no detalhe do código.