

Capítulo 5 - Princípios de Projeto

O capítulo 5 aborda de forma bem completa o projeto de software, destacando sua importância e os principais aspectos que devem ser considerados durante essa atividade. Ele apresenta desde conceitos fundamentais até princípios práticos e métricas que ajudam a avaliar a qualidade do código.

Um ponto que achei interessante foi a ênfase na importância da coesão, do baixo acoplamento e da integridade conceitual, já que esses fatores fazem toda a diferença na construção de sistemas mais robustos e fáceis de manter. Outro destaque é a explicação sobre os princípios SOLID e outras diretrizes de projeto. Isso é essencial para quem quer desenvolver software de forma mais eficiente, pensando não só na funcionalidade, mas também na facilidade de manutenção e evolução do sistema ao longo do tempo. Além disso, a parte sobre métricas de código-fonte é bem útil para quem quer avaliar a qualidade do projeto de forma mais objetiva.

A conclusão do capítulo reforça a importância de aplicar esses conceitos e princípios para garantir a qualidade do projeto de software. Além de abordar técnicas essenciais, o texto também traz uma visão prática de como melhorar a organização e a manutenibilidade do código.

Capítulo 6 - Padrões de Projeto

No sexto capítulo do livro Engenharia de Software Moderna, são abordados os Padrões de Projeto, que oferecem soluções eficientes para desafios recorrentes no desenvolvimento de software. Esses padrões são classificados em três categorias principais: criacionais, estruturais e comportamentais.

Os padrões criacionais, como Factory e Singleton, visam otimizar a criação de objetos, garantindo maior controle e eficiência na geração de instâncias. Já os padrões estruturais, exemplificados por Adapter e Decorator, são aplicados para organizar e interligar classes e objetos de maneira flexível. Por fim, os padrões comportamentais, como Strategy e Observer, têm o propósito de aprimorar a comunicação entre os componentes do sistema, tornando a interação mais dinâmica e eficiente.

O capítulo destaca que o uso adequado desses padrões não apenas evita soluções improvisadas, mas também contribui para um código mais organizado, de fácil manutenção e reutilização. Essa abordagem é especialmente benéfica em projetos complexos e de grande porte, onde a robustez e a adaptabilidade do software são essenciais.