

O capítulo 5 do livro *Engenharia de Software Moderna* aborda os princípios fundamentais do projeto de software, destacando sua importância na construção de sistemas eficazes e de fácil manutenção. A introdução enfatiza a necessidade de decompor problemas complexos em partes menores e independentes, permitindo que sejam resolvidos de maneira modular. O autor usa o exemplo de um compilador para ilustrar essa abordagem. Dentre os tópicos discutidos, a *Integridade Conceitual* se destaca como um dos principais aspectos do design de software, garantindo que todas as partes do sistema sigam uma lógica uniforme. A *Ocultação de Informação* também é ressaltada como uma técnica essencial para minimizar o impacto de mudanças no código, protegendo detalhes internos das classes e promovendo o encapsulamento. Outros conceitos fundamentais abordados incluem *Coesão*, que sugere que cada classe deve ter uma única responsabilidade bem definida, e *Acoplamento*, que deve ser minimizado para reduzir dependências excessivas entre módulos. Para reforçar boas práticas, o capítulo apresenta princípios consagrados como SOLID, que auxiliam na organização e manutenção do código. Princípios como *Aberto/Fechado*, *Substituição de Liskov* e *Deméter* são explorados, fornecendo diretrizes claras para um design robusto e extensível. Por fim, o capítulo discute métricas de qualidade de código, destacando sua importância na avaliação objetiva da estrutura do software. Com uma abordagem prática e teórica, o capítulo fornece um guia valioso para desenvolvedores que buscam criar sistemas mais estruturados e fáceis de manter.

Já o capítulo 6 trata dos padrões de projeto, um conceito essencial para o desenvolvimento de software flexível e reutilizável. O texto inicia com uma introdução ao tema, destacando a influência do trabalho de Christopher Alexander na arquitetura e sua adaptação para o mundo do software pelo grupo conhecido como "Gang of Four" (GoF). Os padrões de projeto são apresentados como soluções testadas para problemas recorrentes no design de software. O capítulo analisa

detalhadamente dez padrões amplamente utilizados: Fábrica, Singleton, Proxy, Adaptador, Fachada, Decorador, Strategy, Observador, Template Method e Visitor. Cada padrão é discutido dentro de um contexto específico, seguido da explicação do problema que ele resolve e a solução proposta. Além desses, outros padrões são mencionados brevemente. O texto ressalta que, embora úteis, os padrões de projeto não são uma "bala de prata". Há um alerta contra o uso excessivo desses padrões, um fenômeno chamado "paternite", que pode tornar os sistemas desnecessariamente complexos. O capítulo conclui com exercícios que reforçam o aprendizado, tornando a leitura essencial para desenvolvedores que desejam aprimorar seu conhecimento em engenharia de software.