

Os capítulos 7 e 9 do livro *Engenharia de Software Moderna* abordam temas essenciais para a qualidade do software: arquitetura e refatoração. O capítulo 7 discute a importância da arquitetura de software e seus principais padrões, começando com definições e abordando a organização de componentes e decisões fundamentais. Entre os padrões apresentados, destaca-se a *Arquitetura em Camadas*, que facilita manutenção e modularidade, e a *Arquitetura em Três Camadas*, que separa interface do usuário, lógica de negócios e banco de dados. A *Arquitetura MVC (Model-View-Controller)* promove a separação entre apresentação, controle e dados, sendo amplamente usada no desenvolvimento web moderno. Já a *Arquitetura de Microsserviços* permite desenvolvimento e implantação independentes de partes do sistema, mas apresenta desafios na comunicação entre serviços. O capítulo também aborda a *Arquitetura Orientada a Mensagens*, que utiliza filas para comunicação assíncrona, garantindo o desacoplamento. A *Arquitetura Publish/Subscribe* permite a notificação de múltiplos consumidores a partir de eventos publicados. Outros padrões incluem *Pipes e Filtros* e *Cliente/Servidor*. O capítulo finaliza com o anti-padrão *Big Ball of Mud*, que caracteriza sistemas sem estrutura clara. A escolha da arquitetura depende do contexto do sistema e das necessidades do projeto.

O capítulo 9 aborda o conceito de refactoring, destacando sua importância na manutenção e evolução de sistemas. A refatoração consiste em modificar o código para melhorar sua estrutura e legibilidade sem alterar seu comportamento. O capítulo discute diferentes técnicas, como extração e movimentação de métodos e classes, renomeação de elementos e remoção de código morto. Uma seção relevante explora os code smells, indicadores de baixa qualidade, como métodos longos, classes grandes e código duplicado, apresentando estratégias para resolvê-los. Além disso, enfatiza a necessidade de testes automatizados para garantir o funcionamento correto do código após modificações. Outro ponto abordado é a

automação da refatoração por meio de IDEs, que ajudam a aplicar mudanças de forma segura e eficiente. O capítulo também diferencia refatorações oportunistas, feitas durante o desenvolvimento, de refatorações planejadas, realizadas estrategicamente. Em suma, reforça que refatorar é essencial para manter sistemas organizados, evitando acúmulo de dívida técnica e garantindo a sustentabilidade do software.