

Este repositório apresenta uma resenha dos conceitos abordados nos capítulos 5 e 6 do livro "Engenharia de Software Moderna", de Marco Túlio Valente.

No Capítulo 5, o foco está nos princípios de projeto, que são fundamentais para garantir a manutenção e a extensibilidade dos sistemas. Um dos conceitos-chave é a Integridade Conceitual, que busca manter o design coeso e consistente, evitando decisões contraditórias ou desorganizadas. Outro ponto importante é o Ocultamento de Informação, que visa reduzir a exposição de detalhes internos, facilitando a manutenção e evitando que mudanças em uma parte do sistema afetem outras.

A Coesão também é destacada, garantindo que cada módulo tenha uma única responsabilidade, o que torna o código mais claro e fácil de entender. Já o Acoplamento trata de minimizar as dependências entre módulos, aumentando a modularidade e a flexibilidade do sistema. Por fim, os Princípios SOLID são apresentados como um conjunto de diretrizes para criar software bem estruturado, com foco em responsabilidades claras e baixo acoplamento.

No Capítulo 6, são abordados os padrões de projeto, que são soluções reutilizáveis para problemas comuns no desenvolvimento de software. Um dos padrões mais conhecidos é o Factory Method, que permite a criação flexível de objetos sem expor a lógica de instantaneidade. Outro é o Singleton, que garante que uma classe tenha apenas uma instância, útil para recursos compartilhados.

O Proxy atua como intermediário, controlando o acesso a um objeto, enquanto o Adapter permite que interfaces incompatíveis funcionem juntas. Já o Facade simplifica o acesso a subsistemas complexos, oferecendo uma interface mais amigável. O Decorator é outro padrão interessante, pois permite adicionar funcionalidades a objetos de forma dinâmica, sem alterar sua estrutura.

O Strategy possibilita a troca de algoritmos em tempo de execução, oferecendo flexibilidade. O Observer implementa comunicação entre objetos de forma desacoplada, ideal para sistemas que precisam reagir a mudanças de estado. O Template Method define o esqueleto de um algoritmo, permitindo que partes específicas sejam customizadas. Por fim, o Visitor separa algoritmos da estrutura de objetos, facilitando a adição de novas operações sem modificar as classes existentes.