Capitulo 5 resumo:

Um sistema é criado para resolver um problema, para isso é necessário decompor ele decompomos eles primeiramente como requisitos funcionais e não funcionais, como será realizado a quebra do problema e como irá implementar o software.

Uma das técnicas mais utilizadas para simplificação é a de abstração, simplificar uma entidade.

Existem os principios do projeto que representam diretrizes para garantir que um projeto atendende determinadas propriedades são elas:

- Integridade Conceitual -> É necessário que um sistema tenha coesão e coerência entre seus artefatos, para que um usuario consiga utiliza-lo por inteiro
- Ocultamento de informação -> As informações que cada usuario deve vizualizar obrigatoriamente e não mostrar informações “inuteis” ou dar mais acesso que o necessário para cada perfil de usuário -> isso ajuda no desenvolvimento paralelo, flexibilidade a mudanças e facilidade de entendimento, pois tem-se classes bem definidas (exemplo disso é o getter e setter que iremos atribuir a uma classe)
- Coesão -> Como dito anteriormente um software deve ser coeso na separação de interesse e em seu codigo
- Acoplamento -> Software deve possuir classes com um acoplamento aceitável
- Codigo demasiadamente extensos devem ser evitados, m

O SOLID estudado no 2 semestre na materia de Programação modular compartilha conceitos com os principios de projeto

- Responsabilidade Única (deve ser facilmente notado o motivo daquela classe) <> Coesão
- Segregação de interface (interfaces podem ter multiplas heranças logo podem ser separadas de diversos modos para melhorar um codigo) <> Coesão
- Inversão de Dependencias (classes não podem ter herança multipla contrario de interfaces) <> Acoplamento
- Prefira composição a Herança <> Acoplamento
- Demeter (Menor conhecimento) <> Ocultamento de informação
- Aberto/Fechado (Acesso a cada atributo)<> Extensibilidade
- Substituição de Liskov (regras para redefinição de métodos de classes base em classes filhas.) <> Extensibilidade