



UnB – Universidade de Brasília  
FGA – Faculdade do Gama  
Curso de Engenharia de Software  
DSW – Desenho de Software  
Professor André Luiz Peron Martins Lanna

### **Lista de exercícios – 2ª. Prova**

Os exercícios desta lista de exercícios cobrem parte do conteúdo da segunda avaliação, cujos tópicos são:

- Conceitos de Desenho de software:
  - Abstração
  - Coesão e acoplamento
  - Decomposição e modularização
  - Encapsulamento
  - Interfaces
- Arquitetura de software
  - Arquitetura lógica e física
  - Arquitetura em camadas
- Projeto de objetos
  - Projeto estático
  - Projeto dinâmico
- Responsabilidades de objetos / padrões GRASP
  - Processo de projeto vs. projeto de software
  - Colaboração entre objetos
  - 9 padrões GRASP

Tal conteúdo pode ser encontrado nas seguintes referências:

- Livro do Larman, capítulos 8 a 18 e 25 a 26.
- Notas de aula disponibilizadas no Moodle.

## Exercícios

- 1) A UML dispõe de vários diagramas para representar as características estáticas e dinâmicas de um projeto de software. Apresente dois exemplos de diagramas estáticos e dois exemplos de diagramas dinâmicos.
- 2) O que é a arquitetura de um software?
- 3) Defina arquitetura lógica e arquitetura física de um projeto. Qual a diferença entre elas?
- 4) O que é uma arquitetura em camadas e quais os seus tipos?
- 5) Em uma arquitetura de camadas, quais as camadas são mais reutilizadas e quais são as menos reutilizadas? Justifique sua resposta.
- 6) Quais os benefícios em decompor uma arquitetura em camadas?
- 7) O que é uma partição em uma arquitetura de software? Qual sua diferença para uma camada de arquitetura de software?
- 8) Qual a relação entre modelo de domínio e arquitetura de software?
- 9) O que é o princípio de separação modelo-visão? O que este princípio determina (como seus elementos se relacionam)?
- 10) Explique a relação existente entre operações de sistema, DSS's e camadas de uma arquitetura. Quem é responsável por receber uma operação de sistema e delegar sua execução e quem é responsável por executá-la?
- 11) A UML serve para modelar diversas características de um software. Quais os tipos principais de modelo que são utilizados e apresente ao menos dois exemplos de diagramas para cada.
- 12) Diferencie *processo de projeto* e *projeto de software*.
- 13) O que é o conceito de abstração em projeto de software e quais são os seus tipos?
- 14) Explique o conceito de independência funcional. Como que ela pode ser atingida e quais suas vantagens?
- 15) O que é coesão e quais são os seus tipos? Apresente-os ordenados da coesão mais baixa para a coesão mais alta.
- 16) O que é acoplamento e quais são os seus tipos? Apresente-os ordenados do acoplamento mais alto para o acoplamento mais baixo.
- 17) O que são software monolíticos e quais suas desvantagens?
- 18) O que é decomposição e modularização e como elas ajudam no desenvolvimento e manutenção de um software? Como a decomposição e a modularização estão relacionadas com os conceitos de acoplamento e coesão? A decomposição e a modularização podem dificultar as tarefas de manutenção de um software. Explique por quê.
- 19) O que é o conceito de interfaces (ocultação de informação) e como eles auxiliam no desenvolvimento do software?
- 20) Apresente o *processo de projeto de software* em termos de suas entradas, atividades e saídas.
- 21) O que são responsabilidades de objetos e quais seus tipos?
- 22) Defina em uma frase a relação entre as execuções de casos de uso, responsabilidades e objetos dos objetos?
- 23) Quais são os nove padrões GRASP? Descreva seus propósitos brevemente.
- 24) Os padrões GRASP são utilizados para distribuir as responsabilidades entre os objetos de um projeto. Verdadeiro ou falso? Justifique sua resposta através de três padrões GRASP.