



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina de Arquitetura e Projeto de Sistemas II**  
**Gabarito da AP2 – 1º semestre de 2012**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
- 

Questão 1 (2 pontos)

Relacione cada problema da coluna da esquerda com um e somente um padrão da coluna da direita que é o principal indicado para tratar o problema.

- |   |                               |
|---|-------------------------------|
| (a) Quem deveria ser responsável por tratar um evento de sistema?   | (1) Padrão Information Expert |
| (b) Em um sistema com centenas de classes, como selecionamos quais responsabilidades devem estar em quais classes?      | (2) Padrão Creator            |
| (c) Como posso evitar o acoplamento direto?   | (3) Padrão Controller         |
| (d) Como tratar alternativas em função do tipo da classe?   | (4) Padrão Low Coupling       |
| (e) A quem atribuir uma responsabilidade quando todas as opções ferem os princípios de acoplamento baixo e coesão alta? | (5) Padrão Polymorphism       |
|   | (6) Padrão Pure Fabrication   |
|   | (7) Padrão Indirection        |
|   | (8) Padrão High Cohesion      |

**Resposta: a → 3; b → 1; c → 7; d → 5; e → 6**

Questão 2 (3 pontos)

Em relação a Componentes, responda com suas palavras:

(a) Qual a diferença entre interfaces providas e interfaces de configuração?

Resposta: Ambas podem ser acessadas por outros componentes ou pela aplicação. Contudo, as interfaces providas disponibilizam as funcionalidades do componente para os demais componentes da aplicação, enquanto as interfaces de configuração servem para que a aplicação possa customizar o comportamento do componente em tempo de execução.

(b) O que deve ser feito quando o componente encontrado na etapa de busca por componentes não é exatamente o que se esperava?

Resposta: Neste caso, é necessário adaptar a arquitetura e customizar o componente a fim de torná-los compatíveis.

(c) Quando um componente precisa ser customizado, o ideal é customizar via variabilidade ou adaptação? Justifique a sua resposta.

Resposta: O ideal é customizar via variabilidade. A customização via variabilidade utiliza mecanismos que foram concebidos para esse fim, como, por exemplo, geração, parametrização e interface de configuração. Assim, é menos propensa a erros e tem mais facilidade de acolher mudanças no futuro.

### Questão 3 (2 pontos)

Responda com Verdadeiro (V) ou Falso (F) as afirmações abaixo:

- a) ☐ Reutilização melhora os índices de produtividade.
- b) ☐ Uma DSSA representa elementos genéricos de um domínio.
- c) ☐ Na especialização de Frameworks Caixa-Branca o framework recebe um conjunto de parâmetros que representa o comportamento específico da aplicação.
- d) ☐ Reutilização não depende da qualidade dos artefatos reutilizáveis.
- e) ☐ Uma Arquitetura de software é uma coleção de componentes computacionais e suas interações.

Resposta: a → V; b → V; c → F; d → F; e → V

### Questão 4 (3 pontos)

a) Defina o que são Estilos Arquiteturais e;

Resposta: Caracterizam famílias de sistemas em termos dos seus padrões de organização estrutural. São tipicamente definidos como uma coleção de Componentes, Conectores e suas Restrições.

- b) descreva os elementos (descrição, componentes, conectores, restrições, ...) de dois estilos arquiteturais.

Resposta:

Camadas

- Descrição – um sistema em camadas é organizado hierarquicamente, once cada camada se comunica apenas com as camadas adjacentes.
- Componentes - Camadas
- Conectores – Protocolos entre Camadas
- Restrições – a comunicação é feita apenas entre as camadas adjacentes
- Exemplos – o Modelo OSI

Pipers & Filters

- Descrição – cada componente tem um conjunto de entradas e saídas
- Componentes - Filtros
- Conectores – Tubos (Pipes)
- Restrições – Filtros são unidades independentes e não sabem das identidades dos outros
- Exemplos - Compiladores