



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina de Arquitetura e Projeto de Sistemas II**  
**AP3 – 2º semestre de 2010**

**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)

Em relação a modelos de classe segundo a UML, explique com suas palavras e dê um exemplo **diferente do visto em aula** para:

(a) Associação simples

**Resposta:** Associação simples indica que duas classes colaboram, ambas com o mesmo nível de importância. Por exemplo, a classe Cliente tem uma associação simples com a classe Locação em um sistema de videolocadora.

(b) Agregação

**Resposta:** Agregação é uma associação onde uma das partes exerce o papel de “todo” e a outra parte exerce o papel de “parte”. Por exemplo, Loja agrega Filmes em um sistema de videolocadora.

(c) Composição

**Resposta:** Composição é uma variação de agregação que introduz mais força na relação todo-parte, onde a “parte” pertence a somente um “todo” e tem o seu ciclo de vida atrelado a ele. Por exemplo, a classe Filme é composta de Item de Conteúdo (abertura, comerciais, filme propriamente dito e extras) num sistema de videolocadora.

(d) Generalização (Herança)

**Resposta:** Generalização é um relacionamento entre tipos que pertencem a uma mesma hierarquia de herança, onde um é mais genérico e outro é mais específico. Por exemplo, Mídia é uma generalização de VHS, DVD e Blu-ray em um sistema de videolocadora.

Questão 2 (3 pontos)

Em relação a arquiteturas de software, responda com suas palavras:

(a) Descreva as responsabilidades das camadas de uma arquitetura em duas e em três camadas.

**Resposta:** Arquiteturas em duas camadas têm no servidor uma camada de apresentação e uma camada de armazenamento. A camada de apresentação tem a responsabilidade de tratar a interação com o usuário, e a camada de armazenamento tem a responsabilidade de persistir dados. Já arquiteturas em três camadas têm no servidor uma camada de apresentação, uma camada de aplicação e uma camada de armazenamento. Onde a camada de aplicação tem a responsabilidade de processar as regras de negócio.

(b) Quais as vantagens da adição da terceira camada em arquitetura de três camadas?

**Resposta:** Em arquiteturas em duas camadas, as regras de negócio usualmente se situam no servidor, mas espalhadas entre essas duas camadas. Já arquiteturas em três camadas têm uma camada específica para o tratamento das regras de negócio (camada de aplicação). Desta forma, profissionais especializados em interface com usuário podem se concentrar na camada de apresentação. Profissionais especializados em gerência de dados podem se concentrar na camada de armazenamento. Finalmente, profissionais especializados em desenvolvimento de software podem se concentrar na camada de aplicação.

(c) Exemplifique a funcionalidade de locação de filmes em uma arquitetura web de 3 camadas.

**Resposta:** A funcionalidade de locação de filmes inicia com uma tela que permita a sua solicitação. O código HTML dessa tela é composto na camada de apresentação. Ao solicitar a locação do filme, a camada de aplicação será acionada. Uma classe que trata o caso de uso de locação (eg.: ControleLocação) trata o pedido, acessando tanto para leitura quanto para escrita a camada de armazenamento. Por fim, o fluxo retorna para a camada de apresentação, onde uma página de retorno é composta e fornecida ao usuário.

Questão 3 (2 pontos)

Explique e exemplifique cada um dos seguintes tipos de *frameworks*:

(a) Caixa-preta

**Resposta:** Nos *frameworks* caixa-preta, o *framework* recebe um conjunto de parâmetros previamente estabelecidos que representa o comportamento específico da aplicação. Este fornecimento de parâmetros é feito por protocolo, através da interface de cada classe do *framework* e, por isso, esta forma de especialização requer que seja conhecida apenas esta interface. Como exemplo, podemos pensar em um sistema de leitor de código de barras, que necessita ser parametrizado com valores como o número de faixas a serem capturadas para que, após essa configuração, o sistema gere um *drive* para determinado tipo de hardware. Dessa forma, não é necessário conhecer o funcionamento interno do sistema para integrá-lo a outros sistemas, mas apenas configurar seus parâmetros para que ele possa ser executado.

(b) Caixa-branca

**Resposta:** Por outro lado, uma segunda forma de especialização é representada pelos *frameworks* caixa-branca, onde o comportamento específico da aplicação é inserido na arquitetura genérica, somando-se métodos às subclasses de uma ou mais classes do *framework*. Cada método somado a uma subclasse deve estar de acordo com as convenções da respectiva superclasse. Os *frameworks* que fazem uso deste processo de especialização obrigam quem os utiliza a conhecer os seus dispositivos internos. Como exemplo, podemos pensar em um sistema básico que informatize uma rede de supermercados. Este sistema, para ser utilizado, deve ser especializado para a rede de supermercados que deseja utilizá-lo, com a implementação de funcionalidades mais específicas (e.g., módulo de vendas ser especializado para efetuar um balanço geral semanal e comunicar os resultados via e-mail ao gerente de cada supermercado, além da comunicação tradicional já realizada para o gerente de rede) e com a configuração de quaisquer opções ou parâmetros (taxas de juros para pagamentos de compras à prazo, bonificações incorporadas ao salário dos funcionários, etc.).

Questão 4 (3 pontos)

Em relação à disciplina de arquitetura de software,

- (a) Quais são os benefícios de se considerar a arquitetura de software durante o desenvolvimento de um sistema?

**Resposta:** Dentre os benefícios de se considerar a arquitetura de software durante o desenvolvimento de um sistema, estão: (i) favorecer a gerência da complexidade; (ii) facilitar a comunicação entre os stakeholders envolvidos no desenvolvimento de software (desenvolvedores, clientes, gerentes, etc.); (iii) criar possibilidades de reutilização; (iv) viabilizar a evolução de sistemas; (v) reconhecer estruturas comuns entre sistemas; (vi) manter o controle da produção de software, já que os detalhes relativos a esforços e desenvolvimento e impacto de mudanças tornam-se mais claros; e (vii) permitir novas oportunidades para análise (ex. consistência, atributos de qualidade, atendimento a estilos arquiteturais).

(b) O que é um estilo arquitetural? Cite ao menos dois exemplos.

**Resposta:** Um estilo arquitetural caracteriza uma família de sistemas em termos de seus padrões de organização estruturais e é definido por uma coleção de componentes, descrição de suas interações e (conectores) e suas restrições. Exemplos são: tubos e filtros (*pipe and filters*) e camadas.

(c) O que é uma Arquitetura Específica de Domínio (DSSA)?

**Resposta:** “Uma DSSA é uma coleção de componentes de software especializados para um determinado tipo de tarefa (domínio), generalizados para que seja possível seu uso efetivo através do domínio, e compostos em uma estrutura padronizada (topologia) efetiva para a construção de aplicações” (HAYES, 1994) ou “Uma DSSA compreende um modelo de domínio, requisitos de referência, uma arquitetura de referência, a infraestrutura ou ambiente de suporte e um processo de desenvolvimento.” (TAYLOR, 1995)