



Teste de Conhecimento

avalie sua aprendizagem

ARQUITETURA DE SISTEMAS
CCT0168_A7_2023110137_V1

Lupa Calc.
🔍 🧮

Aluno: DOUGLAS MATOS DA SILVA
Disc.: ARQ. SISTEMAS

Matr.: 20231110137
2022.3 EAO IGJ / EX

Prezado (a) Aluno(a),

Você terá agora seu **TESTE DE CONHECIMENTO**. Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responder cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

1. Na Arquitetura Orientada a Serviço - SOA, é **INCORRETO** afirmar que o serviço

- ☐ responde às requisições encapsulando todo o detalhe do seu processamento.
- ☐ é uma unidade de trabalho oferecida pelo provedor de serviço para atender à demanda requerida por um consumidor de serviço.
- ☐ é invocada por meio de protocolos de comunicação independentes da localização e do suporte tecnológico.
- ☒ é um componente fortemente acoplado e altamente coeso que implementa uma função reutilizável de negócio.
- ☐ não depende do estado de outros componentes externos para executar um ciclo completo de trabalho.

Explicação: é um componente fracamente acoplado e não fortemente acoplado.

2. Em relação ao padrão de arquitetura MVC (Model View Controller), assinale a afirmativa correta.

- ☐ O componente View interpreta as ações realizadas pelos usuários e as envia para o Model.
- ☐ O padrão de arquitetura MVC não permite gerenciar múltiplas visões utilizando somente um modelo.
- ☐ O componente Controller é a camada de apresentação responsável por gerar uma visualização dos dados e por recebê-los como dados de entrada dos usuários.
- ☒ O componente Model é a camada lógica da aplicação, representa os dados da aplicação e as suas regras de negócio.
- ☐ Os componentes de uma arquitetura de sistemas estão divididos em camadas não estruturadas, que, durante o desenvolvimento, devem ser consideradas de forma fundamental pelo arquiteto.

Explicação:

Os componentes de uma arquitetura de sistemas estão divididos em camadas estruturais.

A camada de apresentação é o componente View.

O padrão de arquitetura MVC permite gerenciar múltiplas visões utilizando somente um modelo.

O componente Controller é quem interpreta as ações realizadas pelos usuários e as envia para o Model

3. O componente Controller do MVC:

- ☐ Responde às solicitações de queries e encapsula o estado da aplicação.
- ☐ E onde são concentradas todas as regras de negócio da aplicação e o acesso aos dados.
- ☐ Envia requisições do usuário para o controlador e recebe dados atualizados dos componentes de acesso a dados.
- ☐ Notifica os componentes de apresentação das mudanças efetuadas nos dados e expõe a funcionalidade da aplicação.
- ☒ Define o comportamento da aplicação, as ações do usuário para atualizar os componentes de dados e seleciona os componentes para exibir respostas de requisições.

Explicação: Controller: Gerencia a interação da apresentação visual do sistema (View) com os comportamentos da aplicação (Model).

4. Analise as afirmações a seguir e escolha a alternativa correta.

- I. Os dados estarem presentes em um único lugar é uma característica da arquitetura em camadas
 - II. Estilos arquiteturais podem ser classificados em situações em que o seu uso é apropriado, com a classificação das vantagens e desvantagens do uso nessas situações
 - III. A arquitetura de repositório (blackboard) é fortemente baseada no uso de dados centralizados
 - IV. Uma das características da arquitetura MVC é a sua aplicação em sistemas interativos
- ☐ I. Somente a alternativa III é verdadeira
 - ☐ II. Somente as alternativas II e III são verdadeiras.
 - ☐ III. Somente a alternativa I é verdadeira.
 - ☐ IV. Somente as alternativas I e II são verdadeiras.
 - ☒ V. Somente as alternativas II, III e IV são verdadeiras.

Explicação:

As camadas mais "baixas", que interagem a nível de byte e bit, conversam com periféricos, lidam com pilhas de comunicação mais robustas são chamadas de camadas mais especializadas, enquanto as camadas acima, que conversam mais com aplicações, são chamadas camadas menos especializadas. Observe a figura 1, a qual ilustra um exemplo de divisão de camadas da comunicação TCP/IP.

Outras linguagens/plataformas também possuem frameworks que aderem ao padrão arquitetural MVC. Isso não inviabiliza que uma equipe crie o seu próprio framework, mas é preciso lembrar que um desenvolvedor não precisa de tempo para aprender a desenvolver em determinada arquitetura e caso a empresa/projeto já utilize um framework bastante popular a curva de aprendizado será bem menor ou praticamente nula. Isso inclusive ajuda na contratação de novos funcionários, onde a empresa já pode exigir como pré-requisito conhecimentos neste framework.

Na arquitetura de repositórios, os dados compartilhados podem ser mantidos em um banco de dados central e acessados por todos os subsistemas. Cada subsistema mantém seu próprio banco de dados e passa dados para outros subsistemas

5. O padrão de arquitetura MVC é um modelo de camadas que divide a aplicação em três componentes: **Model**(modelo), **View** (visualizador) e **Controller** (controlador). As funções de cada um destes três componentes são apresentadas abaixo:

- I. interpreta eventos de entrada e envia requisições para o modelo de dados; em seguida, processa os dados carregados a partir do modelo e envia para o visualizador.
- II. encapsula o acesso aos dados e funções básicas da aplicação, fornecendo ao usuário procedimentos que executam tarefas específicas.
- III. exibe para o usuário os dados fornecidos pelo controle e estabelece uma interface para interação entre o usuário e a aplicação.

A associação correta do componente do padrão MVC com sua função está expressa respectivamente em:

- ☐ Controller, View e Model
- ☐ View, Model e Controller
- ☐ Model, Controller e View
- ☒ Controller, Model e View
- ☐ Nenhuma das opções

Explicação:

Explicando cada um dos objetos do **padrão MVC** tem-se primeiramente o controlador (Controller) que interpreta as entradas do mouse ou do teclado enviado pelo usuário e mapeia essas ações do usuário em comandos que são enviados para o modelo (Model) e/ou para a janela de visualização (View) para efetuar a alteração apropriada. Por sua vez o modelo (Model) gerencia um ou mais elementos de dados, responde a perguntas sobre o seu estado e responde a instruções para mudar de estado.

6. Dentre as afirmativas abaixo, correto afirmar que: I. Componentes de Software consiste no termo utilizado para descrever o elemento de software que tem a função de encapsular funcionalidades. II. Um componente de software é uma unidade independente, e pode ser utilizado com vários outros componentes a fim de formar um sistema mais complexo. III. Em programação orientada a objetos um componente de software implementa somente a interface e não é autônomo em relação aos demais componentes do sistema. IV. Um sistema de software pode ser constituído inteiramente por componentes, já que estes podem se interligar através de suas interfaces.

- ☐ IV
- ☐ I, II
- ☐ III, IV
- ☒ I,II,IV
- ☐ I, III

Explicação:

1. Componentes de Software consiste no termo utilizado para descrever o elemento de software que tem a função de encapsular funcionalidades. CORRETO

2. Um componente de software é uma unidade independente, e pode ser utilizado com vários outros componentes a fim de formar um sistema mais complexo. CORRETO

3. Em programação orientada a objetos um componente de software implementa somente a interface e não é autônomo em relação aos demais componentes do sistema. **ERRADO**

A programação orientada a objetos traz outros pontos que acabam sendo mais interessantes no contexto de aplicações modernas. Como o desempenho das aplicações não é uma das grandes preocupações na maioria das aplicações (devido ao poder de processamento dos computadores atuais), a programação orientada a objetos se tornou muito difundida. Essa difusão se dá muito pela questão da reutilização de código e pela capacidade de representação do sistema muito mais perto do que veríamos no mundo real. Outra grande vantagem que o desenvolvimento orientado a objetos traz diz respeito a leitura e manutenção de código. Como a representação do sistema se aproxima muito do que vemos na vida real, o entendimento do sistema como um todo e de cada parte individualmente fica muito mais simples. Isso permite que a equipe de desenvolvimento não fique dependente de uma pessoa apenas, como acontecia com frequência em linguagens estruturadas como o C, por exemplo.

4. Um sistema de software pode ser constituído inteiramente por componentes, já que estes podem se interligar através de suas interfaces. **CORRETO**

Col@bore

Sugira! Sinalize! Construa!

Antes de finalizar, clique aqui para dar a sua opinião sobre as questões deste exercício.

☐ Não Respondida

☐ Não Gravada

☒ Gravada

Exercício iniciado em 25/09/2022 20:05:49.