

CONCURSO PÚBLICO



ARQUITETO DE SOFTWARE Nível Superior

Leia estas instruções:

1. Identifique-se na **parte inferior** desta capa. Caso se identifique em qualquer outro local deste caderno, você será eliminado do Concurso.
2. Este Caderno contém, respectivamente, **duas** questões discursivas, **uma** proposta de redação e **trinta** questões de múltipla escolha. Não destaque nenhuma folha.
3. Nas questões discursivas e na redação, você será avaliado exclusivamente por aquilo que escrever dentro do espaço destinado a cada resposta, não devendo, portanto, ultrapassá-lo.
4. Cada questão de múltipla escolha apresenta **apenas uma opção de resposta correta**.
5. Verifique se o caderno está completo e sem imperfeições gráficas que possam dificultar a leitura. Detectado algum problema comunique-o, imediatamente, ao fiscal.
6. Escreva de modo legível. Dúvida gerada por grafia, sinal ou rasura implicará redução de pontos durante a correção.
7. Você dispõe de, no máximo, **quatro horas** para responder as questões (múltipla escolha e discursivas), elaborar, em caráter definitivo, a redação e preencher a Folha de Respostas.
8. O preenchimento da Folha de Respostas é de sua inteira responsabilidade.
9. Antes de retirar-se definitivamente da sala, **devolva** ao fiscal **este Caderno** e a **Folha de Respostas**.

Identificação do candidato

Nome (em letra de forma)		Nº da Inscrição	
Nº da turma		Assinatura	

Questão Discursiva 1

O *Processo Unificado da Rational*, conhecido como RUP, caracteriza-se por ser dirigido por casos de uso; centrado na arquitetura e iterativo e incremental.

Baseado nessas informações desenvolva as questões propostas a seguir:

- A) explique as características do RUP;
- B) descreva as diferenças entre as fases e atividades e entre iteração e ciclo.

Para rascunho desta questão, utilize o verso da capa.

RESPOSTA

Questão Discursiva 2

Descreva a importância da J2EE para a implementação de aplicações corporativas. Situe o contexto da discussão nos seguintes conceitos: arquitetura de software J2EE, servidor de aplicações J2EE, portabilidade de código J2EE, reusabilidade de componentes J2EE e processo de desenvolvimento de software na tecnologia J2EE.

Para rascunho desta questão, utilize o verso da folha anterior.

RESPOSTA

Redação

O mundo contemporâneo coloca-nos inúmeros desafios, dentre eles o desafio profissional. Hoje, exige-se dos profissionais não apenas um diploma, mas o exercício competente de sua profissão.

Considerando esse fato, construa um texto em que você deverá **dissertar** sobre **o que é ser um profissional competente**. Ao expressar seu ponto de vista, apresente os motivos que o levaram a posicionar-se de tal forma.

Atente para o fato de que seu texto deverá apresentar um título, ser coeso e coerente, estar escrito em língua culta padrão e conter, no mínimo, 20 (vinte) linhas.

Não assine o texto produzido e, para rascunho, utilize o verso da folha anterior.

ESPAÇO DESTINADO AO TEXTO DISSERTATIVO QUE SERÁ AVALIADO

Título	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	

Para rascunho, utilize qualquer espaço em branco, a partir desta folha.

- 01.** Como todo produto industrial, o software tem um ciclo de vida. Em relação ao ciclo de vida do software, é **correto** afirmar que
- A) o ciclo de vida termina quando o software é entregue ao cliente.
 - B) o ciclo de vida inicia quando o software entra em operação.
 - C) a codificação é apenas uma pequena parte do ciclo de vida.
 - D) os testes devem ser realizados no término do ciclo de vida.
- 02.** São exemplos de modelos de ciclo de vida:
- A) espiral, COM e NET.
 - B) cascata, CORBA e COM.
 - C) padrões de projeto, prototipagem evolutiva e CORBA.
 - D) cascata, espiral e prototipagem evolutiva.
- 03.** O Processo Unificado possui 4 fases. Com relação a esse processo, é **correto** afirmar:
- A) a transição é a fase na qual o produto é colocado à disposição de uma comunidade de usuários.
 - B) a construção é a fase que permite verificar os resultados da implementação.
 - C) a elaboração é a fase na qual se justifica a execução de um projeto de desenvolvimento de software do ponto de vista do negócio do cliente.
 - D) a concepção é a fase na qual é produzida uma versão completamente operacional do produto pelos programadores.
- 04.** A técnica de JAD (*Joint Application Development*) é utilizada na engenharia de software para
- A) prototipagem evolutiva.
 - B) manutenção de software.
 - C) codificação em grupo.
 - D) levantamento de requisitos de software.
- 05.** Os elementos de um Documento de Especificação de Requisitos são:
- A) diagramas de contexto, análise de riscos, diagramas de objetos e diagramas de estados.
 - B) planejamento, diagramas de implantação, código fonte e descrição dos requisitos não-funcionais.
 - C) diagrama de contexto, diagramas de casos de uso, descrição das interfaces externas e descrição dos requisitos não-funcionais.
 - D) planejamento, orçamento, cronograma e análise de riscos.

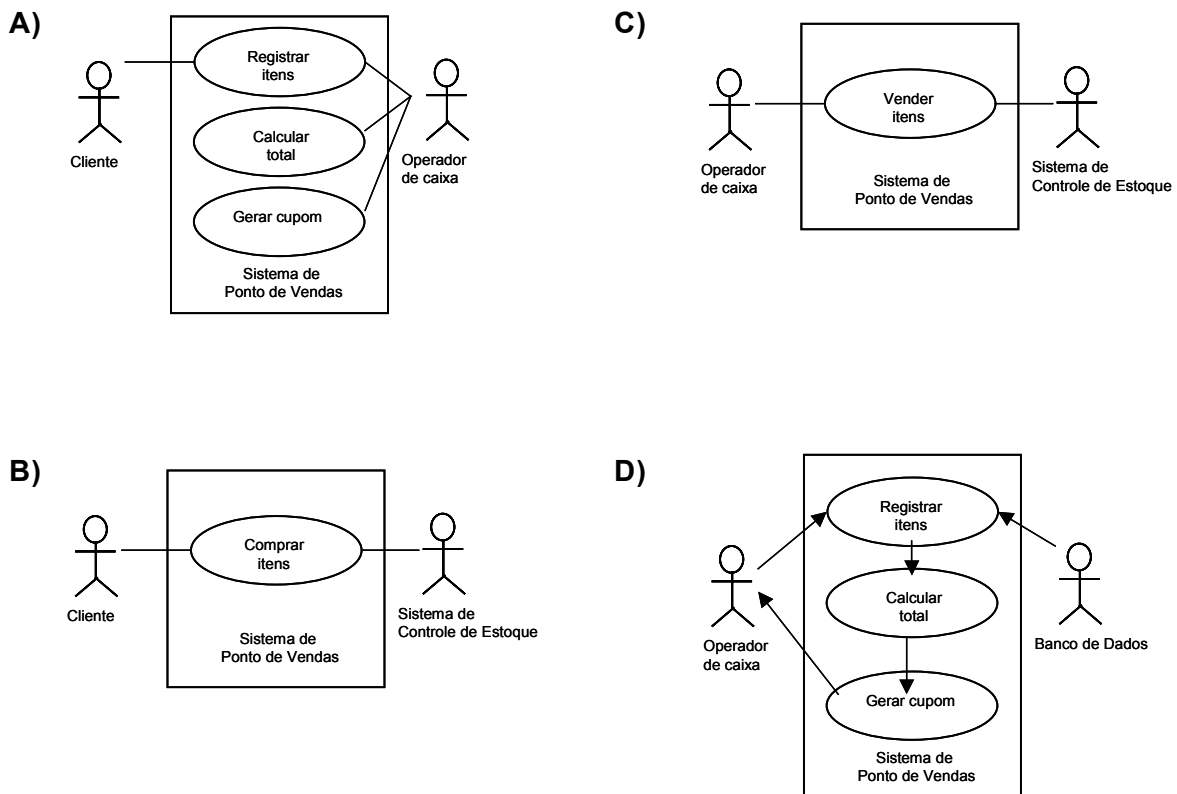
Considere o seguinte texto, que descreve um cenário de sistemas de software, e responda as questões de **06** a **08**.

O cliente está numa loja equipada com um sistema de ponto de vendas (SPV). Ele escolhe vários itens de compra. Dirige-se ao operador de caixa, que abre a transação e registra cada um dos itens no SPV. Se houver mais de um exemplar do item, o operador também pode entrar com a quantidade. O sistema determina o preço do item e acrescenta informação sobre o item à transação de vendas em andamento. A descrição e o preço do item corrente são apresentados. Ao final, o sistema calcula e apresenta o total da venda. O cliente escolhe o tipo de pagamento, que pode ser em dinheiro, com cheque ou com cartão de crédito. Após receber o pagamento, o operador registra no SPV, que atualiza o sistema de controle de estoque. O sistema gera um cupom, que é entregue ao cliente pelo operador, e encerra a transação. O cliente sai com os itens comprados.

06. Baseado no texto acima, e de acordo com os conceitos de casos de uso UML, os atores do sistema descrito no texto são

- A) o operador de caixa e o sistema de controle de estoque.
- B) o operador de caixa e o cliente.
- C) o cliente e o sistema de controle de estoque.
- D) o cliente e o SPV.

07. Segundo o texto, o diagrama de casos de uso que **melhor** reflete o contexto do sistema descrito no cenário é:

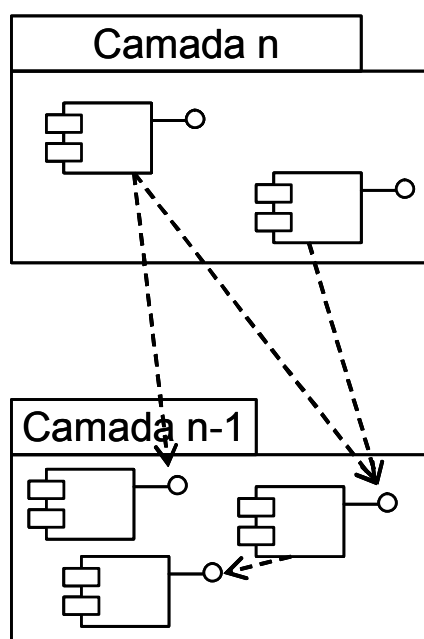


08. Os diagramas UML que descrevem os aspectos comportamentais de um sistema são

- A) atividades, seqüência e colaboração.
- B) classes, seqüência e componentes.
- C) atividades, componentes e pacotes.
- D) classes, pacotes e estados.

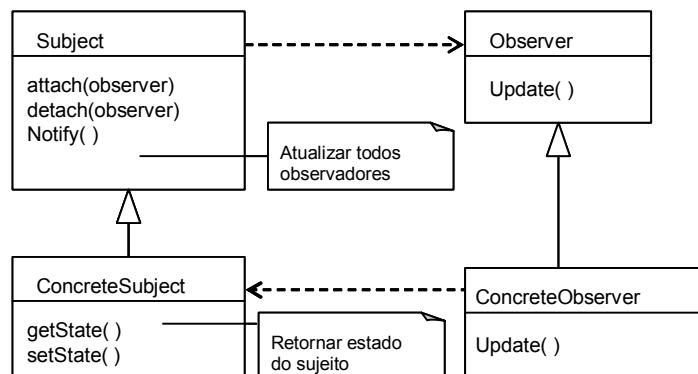
- 09.** Por arquitetura de software, entende-se:
- A)** a descrição, em alto nível, da estrutura de componentes e suas interfaces, conexões e interações de um software.
 - B)** a descrição detalhada dos níveis do fluxo interno dos métodos, funções e procedimento de um software.
 - C)** os desenhos detalhados das telas, das interconexões e dos mecanismos de navegação de um software.
 - D)** a atividade realizada por analistas de sistemas que antecede a atividade de engenharia de software.
- 10.** Os elementos essenciais de um padrão de projeto são
- A)** o problema, o diagrama de estrutura, o diagrama de comportamento e os casos de testes.
 - B)** o nome do padrão, o problema, o contexto, a solução e as conseqüências.
 - C)** o nome do padrão, o diagrama de estrutura e o diagrama de comportamento.
 - D)** a estrutura, o contexto, a solução e os casos de testes.
- 11.** O padrão de projeto MVC - Model-View-Controller (Modelo, Visão, Controlador) deve ser aplicado em
- A)** projeto de sistemas de software distribuídos em ambientes operacionais heterogêneos com intercâmbio de dados.
 - B)** projetos de modelos visuais de software de controle para diferentes aplicações operacionais e de dados.
 - C)** projetos de software de interfaces gráficas que permitam diferentes visões para uma mesma estrutura de dados.
 - D)** projetos de sistemas de software que padronizam vários modelos, vários controladores e uma única visão.

As questões de **12** a **14** tomam como referência a figura abaixo, que descreve um modelo de software.



12. Os elementos UML presentes no diagrama são:
- A) classes, atividades, estados e dependências.
 - B) pacotes, classes, camadas e objetos.
 - C) componentes, objetos, camadas e interfaces.
 - D) pacotes, componentes, interfaces e dependências.
13. Assinale a afirmativa **verdadeira**, de acordo com a figura.
- A) A retirada de um elemento da Camada $n-1$ não afeta a Camada n .
 - B) Os relacionamentos entre as camadas indicam que a Camada $n-1$ depende da Camada n .
 - C) Os relacionamentos entre as camadas indicam que a Camada n depende da Camada $n-1$.
 - D) O acréscimo de um novo elemento na Camada n afeta a Camada $n-1$.
14. O diagrama da figura descreve a aplicação
- A) do padrão arquitetural Layer e do padrão de projeto Factory.
 - B) do padrão arquitetural Layer.
 - C) do padrão de projeto ChainOfResponsibility.
 - D) do padrão arquitetural Layer e do padrão de projeto ChainOfResponsibility.

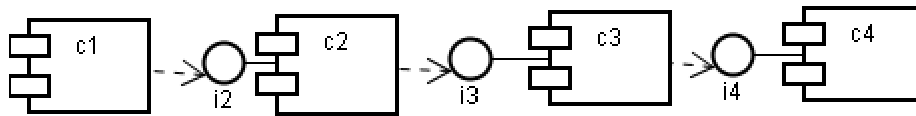
As questões de 15 a 17 referem-se à figura seguinte, que apresenta um diagrama de classe UML.



15. Os nomes dos relacionamentos entre as classes Subject e ConcreteSubject; Subject e Observer; Observer e ConcreteObserver; ConcreteObserver e ConcreteSubject são, **respectivamente**,
- A) associação, agregação, associação, agregação.
 - B) generalização, agregação, generalização, agregação.
 - C) agregação, dependência, agregação, dependência.
 - D) generalização, dependência, generalização, dependência.

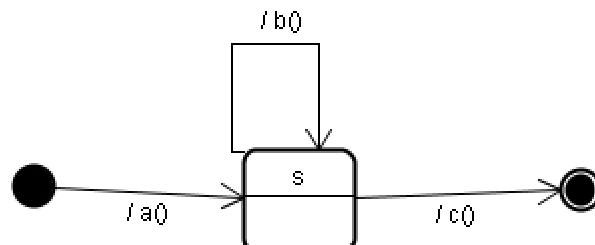
16. O relacionamento entre `ConcreteSubject` e `ConcreteObserver` indica que
- A) `ConcreteObserver` herda o método `Update()` da classe `ConcreteSubject`.
 - B) `ConcreteObserver` envia seus resultados para a classe `ConcreteSubject`.
 - C) `ConcreteSubject` herda o método `Update()` da classe `ConcreteObserver`.
 - D) `ConcreteObserver` utiliza algum método da classe `ConcreteSubject`.
17. O diagrama da figura representa o padrão de projeto Observador (observer), também conhecido como Publish-Subscribe. Esse padrão deve ser aplicado
- A) quando um objeto tem dois aspectos independentes um do outro. Encapsulando-se esses aspectos num único objeto, fica possível variá-los e reutilizá-los independentemente.
 - B) quando um objeto deveria ser capaz de notificar outros objetos sem fazer hipóteses ou usar informações de quem são esses objetos. Em outras palavras, não se quer que esses objetos sejam fortemente acoplados.
 - C) quando uma mudança em um objeto não exige mudanças em outros e não é necessário determinar quantos objetos precisam ser notificados.
 - D) quando uma mudança em um objeto afeta todos os outros objetos independentes, de maneira que não é possível notificá-los e, dessa forma, não é possível também reutilizá-los.
18. As diretrizes de usabilidade a serem aplicadas na prototipagem de interfaces gráficas de usuário são
- A) coesão, simplicidade, portabilidade e modularidade.
 - B) modularidade, reusabilidade, simplicidade e portabilidade.
 - C) coesão, acoplamento, independência funcional e complexidade.
 - D) consistência, simplicidade, realimentação e reversibilidade.
19. Os estilos de interação típicos em interfaces gráficas de usuário são
- A) menus, SQL, linguagens de comando e caixas de diálogo.
 - B) hipertexto, linguagem natural, linguagem de comandos e funções.
 - C) janelas, menus, formulários e caixas de diálogo.
 - D) janelas, linguagem natural, funções e SQL.
20. Em relação ao desenvolvimento de protótipos de interfaces gráficas, é **correto** afirmar que
- A) a prototipagem deve ser realizada após o sistema ter sido completamente implementado, para que ela não interfira na sua essência funcional.
 - B) o protótipo pode ser incrementado e evoluir para um produto de software final, possibilitando maior produtividade.
 - C) a prototipagem não deve ser realizada durante a especificação de requisitos, de forma a não influenciar clientes e usuários.
 - D) o protótipo não deve ser utilizado para testes de usabilidade, pois os testes com protótipos nunca são confiáveis.

As questões 21 e 22 referem-se à figura abaixo, que representa um diagrama típico de níveis (tiers) de uma aplicação J2EE, do qual foram removidas algumas informações.



21. Assinale a seqüência **correta** de nomes associados aos níveis apresentados no diagrama, na ordem c1, c2, c3 e c4.
- A) midlet, container servlet, sgbd e container ejb.
 - B) applet, ejb, servlet e sgbd.
 - C) browser, container web, container ejb e sgbd.
 - D) sgbd, browser, container servlet e container ejb.
22. Assinale a seqüência **correta** de protocolos associados às interfaces apresentadas na ordem i2, i3 e i4.
- A) HTTP, RMI-IIOP, HTML.
 - B) XML, FTP, SMTP.
 - C) HTML, SQL, RMI-IIOP.
 - D) HTTP, RMI-IIOP, JDBC.
23. Durante a instalação e operação real de um sistema J2EE com alto volume de transações e usuários, um dos níveis apresentados no diagrama possuirá a **maior** quantidade de instâncias de componentes. Esse nível é
- A) c1.
 - B) c2.
 - C) c3.
 - D) c4.

As questões de 24 a 26 referem-se ao diagrama abaixo. Esse diagrama representa o ciclo de vida de um servlet que serve pedidos através do método GET. Foram substituídos os nomes e os métodos que são usados para delimitar suas transições de estados.



24. A associação correta entre os nomes de métodos na ordem a, b e c é
- A) service, doGet, service.
 - B) init, service, destroy.
 - C) newInstance, ejbCreate, ejbRemove.
 - D) init, destroy, run.

25. O nome correto associado ao estado `s` do servlet é
- A) `started` (iniciado).
 - B) `ready` (pronto).
 - C) `destroyed` (destruído).
 - D) `running` (executando).
26. Considerando o framework Servlet, o método que lê os parâmetros de um pedido HTTP e preenche as informações de uma resposta HTTP é
- A) `run()`.
 - B) `init()`.
 - C) `service()`.
 - D) `destroy()`.
27. São padrões de arquitetura da camada EJB:
- A) `ChainOfResponsibility`, `Proxy`, `Iterator`, `Observer`.
 - B) `DataTransferObject`, `DomainDataTransferObject`, `CustomDataTransferObject`, `DataTransferHashMap`, `DataTransferRowSet`.
 - C) `VersionNumber`, `JDBCForReading`, `DataAccessCommandBean`, `DualPersistenceEntityBean`.
 - D) `SessionFaçade`, `MessageFaçade`, `EJBCommand`, `DataTransferObjectFactory`, `GenericAttributeAccess`, `BusinessInterface`.
28. São padrões de transferência de dados entre camadas J2EE:
- A) `SessionFaçade`, `MessageFaçade`, `EJBCommand`, `DataTransferObjectFactory`, `GenericAttributeAccess`, `BusinessInterface`.
 - B) `DataTransferObject`, `DomainDataTransferObject`, `CustomDataTransferObject`, `DataTransferHashMap`, `DataTransferRowSet`.
 - C) `VersionNumber`, `JDBCForReading`, `DataAccessCommandBean`, `DualPersistenceEntityBean`.
 - D) `ChainOfResponsibility`, `Proxy`, `Iterator`, `Observer`.
29. São padrões de design GoF (Gang-of-Four):
- A) `Adapter`, `Proxy`, `Iterator`, `Observer`.
 - B) `Façade`, `Command`, `DataTransferObjectFactory`, `BusinessInterface`.
 - C) `Strategy`, `Bridge`, `Adapter`, `DataTransferHashMap`.
 - D) `AbstractFactory`, `Singleton`, `Iterator`, `DualPersistenceEntityBean`.
30. Com relação às tecnologias HTML, JSP, Servlet e EJB, é **correto** afirmar que
- A) o código de um EJB é compilado para gerar o código de um Servlet.
 - B) o código de um JSP é compilado para gerar o código de um Servlet.
 - C) um componente JSP depende de um componente EJB para funcionar.
 - D) um componente Servlet depende de um componente JSP para funcionar.