

CONHECIMENTOS ESPECÍFICOS

Acerca de organização e arquitetura de computadores, julgue os itens a seguir.

- 61 Registradores são memórias internas do processador, entre as quais se incluem o registrador de endereços da memória (REM), que contém o endereço do dado a ser lido ou escrito na memória, e o registrador de dados da memória (RDM), que contém o dado a ser escrito na memória, no caso de uma operação de escrita.
- 62 Na arquitetura de Von Neumann, o caminho único de dados é o barramento físico, que liga a memória diretamente aos dispositivos de entrada e saída (E/S): o objetivo desse barramento é a troca de dados externos com a máquina, enquanto a memória guarda os dados de forma temporária no computador.
- 63 A BIOS gerencia tanto a placa-mãe quanto o *boot* do sistema operacional, além de ser responsável pelo gerenciamento de processamento e pela troca de dados entre o PC (*program counter*) e a memória RAM.
- 64 MBR (*master boot record*) é um *firmware* gravado na placa-mãe e que tem por objetivo guardar, na BIOS, as configurações realizadas pelo usuário por meio do programa SETUP, como, por exemplo, a data/hora do sistema e a ordem do *boot*.

Julgue os itens seguintes, a respeito da aritmética computacional e dos princípios de sistemas operacionais.

- 65 Considerando-se que, nas igualdades a seguir, W, Y e Z receberão o resultado da conversão para a base decimal dos números apresentados, é correto afirmar que $W > Z > Y$.

$W = 137$ (na base octal)

$Y = 1011101$ (na base binária)

$Z = 5E$ (na base hexadecimal)

- 66 Considerando-se que, nas igualdades a seguir, X e T receberão o resultado da conversão para base octal dos números apresentados, é correto afirmar que $X + T = 521$ (na base octal).

$X = 106$ (na base decimal)

$T = E7$ (na base hexadecimal)

- 67 No escalonamento de processos do tipo SJF (*shortest job first*), o tempo de ocupação de CPU é determinado para cada processo; assim, quando a CPU está livre, o processo que irá utilizar a CPU é selecionado pelo critério de menor tempo, formando-se uma fila de processos por ordem crescente de tempo de execução.

Julgue os itens seguintes a respeito da construção de algoritmos, dos conceitos de variáveis e de bloco de comandos e das estruturas de controle.

- 68 Na lógica de programação, um bloco de comando é definido como um conjunto de ações para determinada função e tem como delimitadores as palavras reservadas *INPUT* e *OUTPUT*.
- 69 O laço de repetição na estrutura de repetição *para* será executado pelo menos uma vez.
- 70 A estrutura de controle *seleção* não pode ser utilizada nas situações em que duas alternativas dependam de uma mesma condição — uma de a condição ser verdadeira e outra de a condição ser falsa.
- 71 Uma das vantagens de se construir um algoritmo por meio do pseudocódigo é o fato de que a passagem do algoritmo para uma linguagem de programação qualquer se torna uma atividade quase que instantânea.
- 72 Durante a execução de um programa, o conteúdo de uma variável pode mudar ao longo do tempo, no entanto ela só pode armazenar um valor por vez.

Julgue os itens seguintes, quanto aos conceitos da programação estruturada e da programação orientada a objetos e aos métodos de ordenação, pesquisa e *hashing*.

- 73 Na programação orientada a objetos, uma das características do polimorfismo é a independência do *software* que invoca o comportamento polimórfico em relação aos tipos de objeto para os quais as mensagens são enviadas.
- 74 O método de ordenação conhecido como *quick sort* utiliza o maior elemento, o qual é sempre colocado ao final do vetor, para garantir que a ordenação seja realizada em ordem decrescente.
- 75 Na pesquisa do tipo sequencial, há aumento do desempenho se a tabela estiver ordenada pelo valor da chave.
- 76 Os elementos-chave nas funções de *hashing* são sempre números naturais.
- 77 A programação estruturada é altamente eficiente no desenvolvimento de programas grandes e complexos.
- 78 Na programação orientada a objetos, ao se definir uma classe, não há obrigatoriedade de se criarem objetos pertencentes a essa classe.

Julgue os itens que se seguem, a respeito de arquitetura cliente-servidor e TCP/IP.

- 79 Em uma arquitetura cliente-servidor, a aplicação utiliza comunicação direta entre duplas de servidores (hospedeiros) conectados alternadamente, denominados pares-clientes, tal como ocorre nas aplicações *web*, nas quais um servidor *web* fornece dados para vários pares.
- 80 O TCP, um protocolo da camada de transporte do TCP/IP, oferece à aplicação solicitante um serviço confiável, orientado à conexão, além de controle de congestionamento para evitar que outra conexão TCP encha os enlaces e roteadores entre hospedeiros comunicantes com uma quantidade excessiva de tráfego.
- 81 O IP oferece serviço de entrega de melhor esforço, uma vez que seus *datagramas* não são fragmentados e, como o serviço não é confiável, há necessidade de serem reconstruídos pelos roteadores antes que cheguem à camada de transporte no destino.

- 82 O roteamento é um serviço provido pela camada de rede, por exemplo, pelo algoritmo de vetor de distâncias (DV), que é iterativo — o processo continua até que mais nenhuma informação seja trocada entre vizinhos — e assíncrono — não requer que todos os nós rodem simultaneamente.
- 83 Em uma aplicação *web* tradicional, o usuário requisita uma página *web* constituída de objetos (por exemplo, arquivo HTML, imagem JPEG ou um clipe de vídeo), o navegador envia ao servidor mensagens de requisição HTTP para os objetos da página e, ao receber as requisições, esse servidor responde com mensagens de resposta HTTP por meio do TCP ativado pela máquina que queira receber o arquivo, seja em uma *intranet*, seja na Internet.

Acerca de OAuth e LDAP (Lightweight Directory Access Protocol), julgue os itens seguintes.

- 84 LDAP é um protocolo de diretórios que provê repositórios de informações de recursos de sistemas e serviços dentro de um ambiente centralizado e estritamente relacionado ao servidor. Por questão de limitação do padrão x.500, do qual foi originado, o LDAP não suporta funções de segurança e de acesso de cliente.
- 85 OAuth é um nó de uma árvore LDAP cujo objetivo é prover segurança nos acessos aos dados contidos na árvore, de modo a validar os *tokens* trocados entre as aplicações e(ou) autenticações envolvidas no processo.

Acerca da engenharia de requisitos no desenvolvimento de *software*, julgue os itens a seguir.

- 86 No processo de *elicitação* de requisitos, há atividades relacionadas a identificação, rastreabilidade e mudanças em requisitos.
- 87 O objetivo do processo de desenvolvimento de requisitos, conforme as melhores práticas e processos de mercado, é definir os requisitos do cliente, do projeto e do processo.
- 88 Uma forma de diminuir os riscos resultantes da evolução contínua dos requisitos é a combinação dos processos de gerência e de desenvolvimento de requisitos.
- 89 De acordo com as técnicas facilitadoras de especificação de aplicação, recomenda-se que a descrição de requisitos e regras seja feita diretamente pela equipe técnica, sem a participação do cliente.
- 90 Nas atividades de desenvolvimento de *software*, a validação consiste em assegurar que se está desenvolvendo o produto de acordo com o desejo do cliente; na verificação, busca-se avaliar se o produto está sendo desenvolvido conforme os requisitos.
- 91 Protótipos descartáveis são os que não são utilizados posteriormente; protótipos evolutivos são aqueles que podem ser utilizados como base de parte ou de todo o *software* a ser desenvolvido.

Tendo em vista que o processo de gerência de requisitos controla todos os requisitos recebidos ou gerados, entre os quais estão os funcionais e os não funcionais, bem como os requisitos impostos pela organização, julgue os itens que se seguem, relativos a esses aspectos.

- 92 Para que os requisitos sejam refinados e sejam gerados modelos de análise e projeto para codificação, apenas a avaliação e a aprovação por parte do cliente — mesmo após o entendimento dos requisitos — não são suficientes.
- 93 Definir e manter matriz de rastreabilidade dos requisitos permite controlar e tratar as mudanças em requisitos durante o processo de *elicitação* e especificação do produto.
- 94 Ao se aplicar a rastreabilidade bidirecional, é possível determinar se todos os requisitos-fonte foram completamente tratados e se todos os requisitos do produto atendem aos requisitos do cliente.

Com relação à qualidade de *software*, julgue os itens a seguir.

- 95 Na *extreme programming*, como não há especificação de sistema que possa ser usada por equipe de teste externa, a característica de *test-first* exige que os implementadores de tarefa compreendam detalhadamente a especificação de comportamento da funcionalidade em desenvolvimento, a fim de que possam escrever o teste para o sistema.
- 96 A análise por pontos de função é métrica que dimensiona o *software* com base em características funcionais e fluxo de dados, diferenciando-se de outras métricas que utilizem remuneração por hora trabalhada ou por posto de serviço.
- 97 No Scrum, as iterações (*sprints*) e funcionalidades a serem implementadas (*product backlog*) são estabelecidas pelo *scrum master*, planejadas e priorizadas em reuniões chamadas *review meetings*, quando a prática de programação em pares auxilia na realização de processos informais de revisão de código.
- 98 Na contagem de pontos por função com a utilização de métodos ágeis, mudanças em funcionalidades decorrem de alteração de escopo, regras de negócio, refinamentos de requisitos ou mudanças legais e(ou) regulamentares e serão consideradas funcionalidades novas relativas a uma *sprint* específica para que seja possível realizar a referida contagem.
- 99 No processo unificado de desenvolvimento de *software*, a fase de construção está relacionada à entrada do sistema em funcionamento, o que requer entendimento do domínio do problema e um sistema de *software* documentado que funcione corretamente no ambiente operacional.

Acerca de CMMI, julgue os próximos itens.

- 100 De acordo com o conceito de institucionalização (*institutionalization*) para o CMMI-DEV v.1.3, se houver mudança de objetivos para o processo, as implementações desse processo também poderão ser alteradas para garantir que ele continue efetivo. Os objetivos genéricos incorporam os graus de institucionalização e expressam o nome dos processos associados a cada objetivo.
- 101 Em sua representação contínua, o CMMI-DEV possibilita à organização escolher conjuntos de áreas de processos inter-relacionadas como, por exemplo, gestão de configuração, medição e análise, integração de produtos e planejamento de projetos.

Julgue os itens a seguir, a respeito de MPS-BR.

- 102 No nível A – em otimização do MPS-BR, a implementação de pelo menos um dos processos selecionados para análise de desempenho produz como resultado processos alinhados a objetivos quantitativos de negócio e à definição da medição do processo a partir das necessidades de informação.
- 103 No MPS-BR, o nível de maturidade C – definido é composto pelos processos dos níveis de maturidade anteriores (G ao D) acrescidos dos processos de desenvolvimento de requisitos, integração do produto, projeto e construção do produto, validação e verificação. Nos processos desse nível, um dos resultados esperados é a avaliação dos domínios de aplicação quanto aos potenciais de reutilização.

Julgue os itens a seguir, relativos a projetos orientados a objeto.

- 104 Padrões de projeto devem conter uma especificação detalhada somente dos problemas e da solução, para permitir o reúso de componentes executáveis.
- 105 Na abordagem orientada a objeto da UML, o processo de desenvolvimento possui as fases de análise, projeto, programação, teste e implantação.
- 106 A modelagem orientada a objetos demonstra como os objetos podem ser classificados, a maneira como herdamos atributos e operações, como são compostos e de que forma interagem entre si.

Com relação à UML, julgue os itens a seguir.

- 107 O diagrama de estrutura composta é utilizado para descrever a estrutura interna de um classificador, com detalhamento das partes internas que o compõem e da comunicação e da colaboração entre tais partes.
- 108 O diagrama de tempo é utilizado para descrever a mudança no estado ou na condição de uma instância de uma classe.
- 109 O diagrama de caso de uso deve ser formal, pouco abstrato e aplicado apenas no início da modelagem do sistema, servindo de referência aos demais diagramas.
- 110 Devido à necessidade de se obterem visões múltiplas do sistema a ser modelado, a UML fornece diagramas que se complementam e que devem ser aplicados na sua totalidade.

Acerca de conceitos básicos e tratamentos de exceção em programação orientada a objetos, julgue os itens subsequentes.

- 111 Se o sistema de informação tiver uma consulta sobre o estado de conservação do automóvel, será considerado que o estado é um atributo do objeto carro.
- 112 Objetos podem ter mais de uma identidade; para isso, é necessário que a sua posição esteja explícita em um *array* de dados.
- 113 Considere que, em um sistema de informações, um objeto possua dados de uma pessoa, tais como: nome, endereço, data de aniversário e número do cartão de crédito. Considere, ainda, que esse sistema exponha, de forma pública, informações sobre o nome e a data de aniversário e deixe os dados do cartão de crédito protegidos em formato privado. Nesse caso, o sistema estará usando o recurso de interface.
- 114 Em um sistema de informação que contenha dados pessoais de clientes de uma empresa de cartões de créditos, caso sejam criadas classes cliente e pessoas, então a classe pessoas é uma classe de polimorfismo de sobrecarga.

- 115 Se usuários, ao consultar um objeto específico do sistema, gerarem um erro conhecido no sistema, que emite uma mensagem em inglês, a qual é modificada no código-fonte do sistema e emitida em português para ser visualizada pelo usuário final, estará sendo gerado um tipo de ocorrência caracterizada como tratamento de exceção.
- 116 O armazenamento de informações referentes às classes de um sistema é feito em uma metaclassa.
- 117 Se, em tempo de execução de um sistema, ocorrer associação entre uma entidade e um atributo, então essa associação será considerada um acoplamento dinâmico.

Acerca de microsserviços e práticas ágeis, julgue os itens a seguir.

- 118 **Situação hipotética:** Uma empresa possui um grande sistema com todas as suas funcionalidades em uma aplicação que acessa um banco de dados. A aplicação foi desmembrada em várias outras, em formatos de contêineres que podem ser provisionados, iniciados e parados sob demanda em ambientes de homologação e desenvolvimento, porém, em produção, o *deploy* é feito manualmente. **Assertiva:** Nessa situação, configura-se um ambiente que possui práticas de entrega contínua.
- 119 Em uma arquitetura de microsserviço, caso sejam criados vários sistemas, a falha de algum deles não afetará os demais sistemas.

A respeito de Scrum, julgue os itens que se seguem.

- 120 Como prática, deve-se sempre priorizar o que seja considerado mais importante para o projeto, sendo de responsabilidade do *product backlog* a priorização do que deve ser feito em cada *sprint*.
- 121 O Scrum *master* tem a responsabilidade de ajudar todos os envolvidos no projeto a manter as práticas e os princípios do Scrum.

Acerca de PHP e SOA (*service oriented architecture*), julgue os itens a seguir.

- 122 No caso de um novo serviço de informação necessitar de obter dados de uma aplicação legada via *web*, sem acesso direto a base de dados, tal demanda pode ser atendida no padrão SOA, por meio de uma API (*application programming interface*) que utilize o método `POST`.
- 123 Se uma aplicação é executada em um servidor *web* Apache usando a linguagem PHP na versão 4.X, então ela deve ter a configuração `register_globals=off` para diminuir riscos de segurança relacionados a essa versão e também evitar que os desenvolvedores escrevam códigos inseguros.

Acerca da linguagem Java, julgue os próximos itens.

- 124 O método no código
`System.out.println(valor.charAt(0));`
 retornará todos os caracteres referentes à *string* avaliada.
- 125 O método no código
`System.out.println(valor.codePointAt(0));`
 retornará a comparação entre duas *strings* em formato unicode.

A respeito da metodologia XP (*Extreme Programming*), julgue os itens que se seguem.

- 126 Para apoiar a equipe de desenvolvimento, é uma prática o uso do cliente *on-site* em tempo integral.
- 127 O ritmo ágil de desenvolvimento de *softwares* é uma prática usada para favorecer a entrega das *releases* quando grandes volumes de horas extras são tolerados.

Com relação ao processo de testes de *software*, julgue os itens a seguir.

- 128 Para a estimativa do esforço do teste, são adotadas duas abordagens: estimativa de horas baseada em valores típicos; e estimativa de esforço baseada em especialistas.
- 129 As atividades de planejamento de teste incluem programar a implementação, a execução e a validade dos testes e configurar o nível de detalhe para os procedimentos de teste e reprodução do incidente.
- 130 No teste de integração, o foco é a comunicação entre os módulos do *software*, não as suas funcionalidades; portanto, nessa fase, testes funcionais não podem ser utilizados.
- 131 No teste funcional, que é uma das fases do processo de validação, não é necessário o conhecimento das estruturas internas do *software*.
- 132 O planejamento de testes deve ser documentado em um plano de teste mestre, no qual sejam agrupados os diferentes níveis de aplicação e descritos os documentos a serem utilizados bem como as atividades de teste.

No que se refere a ferramentas de testes de *software*, julgue os itens seguintes.

- 133 Uma ferramenta de medição de cobertura, intrusiva ou não intrusiva, é utilizada para se calcular a porcentagem de estruturas de código que são exercitadas por um dado conjunto de testes.
- 134 As ferramentas de execução do teste são classificadas como ferramentas de suporte para execução e registro e têm, como vantagem, o fato de não requererem um grande esforço para a obtenção de resultados expressivos.
- 135 A ferramenta de análise dinâmica é aplicada para se encontrarem defeitos que só são evidenciados quando o *software* está em execução, como, por exemplo, vazamento de memória.

Acerca do desenvolvimento seguro de *software*, julgue os itens seguintes.

- 136 São exemplos de falhas de XSS (*cross-site scripting*) a injeção de SQL e o LDAP (*lightweight directory access protocol*), criado como alternativa ao DAP.
- 137 A fim de mitigar riscos à segurança computacional, as infraestruturas críticas devem ser protegidas por meio de um processo de desenvolvimento em que a arquitetura, o *design* e a implementação do *software* resistam a ataques, para que se protejam o próprio *software* e as informações por ele processadas.
- 138 Aplicações *web* que não protegem informações sensíveis, tais como dados de cartões de crédito, recorrem à criptografia no armazenamento ou no trânsito dos dados e a precauções especiais quando as informações trafegam pelo navegador.

- 139 **Situação hipotética:** Ao realizar uma sessão em um navegador, um usuário enviou, para o sítio de um atacante, a seguinte requisição HTTP forjada, cujo parâmetro *AB* foi alterado com a diretiva `document.location` apontando para esse sítio, tal que o ID da sessão do usuário foi enviado para o atacante, que, assim, teve condições de acessar a sessão atual em questão.

```
(String)page+="input name = 'senha' type = 'TEXT'
value = '"+request.getParameter("AB")+"'>";
```

Assertiva: A situação apresentada configura uma falha de XSS.

- 140 No caso em que contas-padrão não são alteradas em razão de o console de administração do servidor de aplicação ter sido instalado automaticamente e não ter sido removido, recomenda-se utilizar um processo de *hardening* recorrente ou desenvolver arquitetura de aplicação que separe os componentes.
- 141 Mesmo em sistemas com componentes e bibliotecas mantidos atualizados, porém não versionados, pode ocorrer falhas do tipo utilização de componentes vulneráveis conhecidos, em que o atacante identifica um componente vulnerável de tais bibliotecas, personaliza o *exploit* por meio de varredura ou análise manual, conforme necessário, e executa o ataque.
- 142 **Situação hipotética:** Com a finalidade de realizar testes de segurança, foi desenvolvido um sistema de inferência *fuzzy* com a definição das propriedades das entradas para observar a saída desse sistema, a fim de que fossem indicados erros (*bugs*) e arquivos malformados que pudessem causar falhas no aplicativo. **Assertiva:** Nessa situação, uma boa prática será manter os testes referidos, para que seja possível reproduzir o erro e verificar se as regressões de código não foram reintroduzidas no erro.
- 143 Em ataques do tipo *man-in-the-middle*, embora o atacante possa interceptar a conexão do atacado e, dessa forma, produzir uma falha de injeção (conhecida como *project mogul*), o protocolo de *change cipher spec*, por meio do método de compressão *mogul cipher*, determina o algoritmo de criptografia e o algoritmo de MAC a ser utilizados, além de definir o tamanho de *hash*.

```
SELECT nome
FROM funcionario
WHERE area = 'INTELEGENCIA'
AND endereco LIKE '%BRASILIA,DF%';
```

Tendo como referência o código SQL precedente, julgue os itens a seguir.

- 144 Na cláusula *WHERE*, a condição de seleção `area = 'INTELEGENCIA'` escolhe a tupla de interesse em particular na tabela *funcionario*, pois *area* é um atributo de *funcionario*.
- 145 O código em apreço realiza uma consulta que mostra o nome dos funcionários da área de *INTELEGENCIA* e que têm, como parte do endereço, a cidade de *BRASILIA, DF*.
- 146 A palavra *INTELEGENCIA* está entre aspas simples por pertencer a um atributo, *area*, o qual tem o tipo de dados definido como caractere.
- 147 Em `LIKE '%BRASILIA,DF%'`, o recurso `LIKE` foi definido de forma incorreta, uma vez que a utilização da vírgula (*,*), sem a inclusão da palavra-chave `ESCAPE`, impedirá que o código seja executado.

Julgue os próximos itens, a respeito de conceitos e comandos PostgreSQL e MySQL.

- 148** Por se tratar de um sistema gerenciador de banco de dados de código aberto, o MySQL não oferece suporte a conexões criptografadas entre clientes e o servidor.
- 149** No programa psql do PostgreSQL, a instrução \h permite mostrar o histórico de comandos SQL na sessão atual.
- 150** No MySQL, a instrução CALL é usada para chamar os procedimentos armazenados.
-

Espaço livre
