

-- CONHECIMENTOS ESPECÍFICOS --

Acerca de confiabilidade, integridade e disponibilidade em segurança da informação, julgue os itens subsequentes.

- 51 O acesso à informação com base na necessidade de saber e o gerenciamento de acesso lógico estão diretamente ligados à confiabilidade.
- 52 Um gestor que, por acidente, apague um arquivo de inicialização de um serviço *web* ou insira valores incorretos em uma aplicação de cobrança de um cliente comprometerá a integridade.
- 53 Habilitar *logs* de sistema e de controle de acesso, criptografia de dados de um disco e sistemas de antivírus atualizados são medidas que garantem a integridade.
- 54 Ataque de DoS, política de *backup* não implementada e falha nos discos violam a disponibilidade.
- 55 Oportunidade, continuidade e robustez são características da confiabilidade.

Em relação ao controle de acesso em segurança da informação, julgue os itens a seguir.

- 56 Uma das formas de controle de acesso lógico é o controle de acesso discricionário — DAC, do inglês *discretionary access control* —, em que as permissões derivam de política gerenciada de forma centralizada.
- 57 A apropriada autenticação de usuários externos pode ser alcançada pelo uso de criptografia, *tokens* ou VPN.
- 58 O controle de acesso baseado na função é aquele em que o proprietário da informação ou do sistema define um conjunto de reinvidicações necessárias antes de conceder o acesso.
- 59 Uma política de controle de acesso deve ser estabelecida, documentada e revisada com base nos requisitos de negócio e de segurança da informação.
- 60 Atividades de gestão de direitos de acesso privilegiado e gerência de informações secretas de autenticação são executadas pelo gestor de contas dos usuários.
- 61 Em uma organização em que o *login* nos sistemas é feito por *token*, não existe a necessidade de implementar a política de tela limpa e mesa limpa.

A respeito de ameaça, vulnerabilidade e impacto em segurança da informação, julgue os itens que se seguem.

- 62 No processo de gestão de risco, as ameaças que mais causam impacto na organização devem ser identificadas e as necessidades devem ser implementadas pelos controles da segurança, para que se reduza o risco de ataque.
- 63 Na etapa de avaliação do risco, os especialistas em segurança identificam as ameaças que afetam os ativos, de acordo com seu tipo e sua origem.
- 64 Vulnerabilidade é o grau de suscetibilidade de um objeto a uma ameaça.
- 65 O impacto sobre o negócio, mensurado no momento do tratamento dos riscos, pode ser determinado por abordagens quantitativas ou qualitativas.

Acerca de OWSAP Top 10 e ameaça, julgue os próximos itens.

- 66 OSWAP Top 10 consiste em uma lista com as dez maiores vulnerabilidades relacionadas a aplicações *web*.
- 67 A categoria que ocupa a primeira colocação na lista OWSAP Top 10 é a quebra de controle de acesso.
- 68 SLE é o valor que representa a probabilidade estimada da ocorrência de uma ameaça específica dentro do período de um ano.
- 69 Certas configurações de roteadores e *firewalls* podem reduzir a ameaça de ataques DoS e até impedi-los de acontecer.
- 70 *Firewalls* de aplicação *web* utilizam a lista OWSAP Top 10 como base de suas métricas de segurança e para configuração do equipamento.

Julgue os itens a seguir, a respeito de reúso de componentes e modelos de processo.

- 71 O DFR (*design for reuse*) deve ser considerado quando se inicia a criação de um novo componente.
- 72 O modelo ágil não pode ser aplicado a qualquer processo de *software*, pois, para tanto, é necessário que o processo seja projetado de modo que suas características sejam modeladas como componentes e, em seguida, construídas dentro do contexto da arquitetura do sistema.
- 73 Uma descrição ideal de um componente de *software* reutilizável deve ser feita com base no modelo 3C, que significa composição, conteúdo e contexto.
- 74 Um dos problemas da aplicação do modelo cascata no desenvolvimento de um projeto reside no fato de que uma versão operacional do programa não estará disponível antes que o projeto esteja perto do final, por conta do estado de bloqueio que a natureza linear do ciclo de vida clássico produz.

Em relação às metodologias ágeis de desenvolvimento, julgue os itens que se seguem.

- 75 Na metodologia Scrum, cada *sprint* segue um ciclo normal de desenvolvimento, constituído de análise de requisitos, desenvolvimento, testes e documentação.
- 76 O TDD (*test-driven development*) é uma metodologia que, ao longo do tempo, implica que o aplicativo em desenvolvimento tenha um conjunto abrangente de testes que ofereça confiança no que foi desenvolvido até então.
- 77 O método Kanban pode ser utilizado em substituição à metodologia Scrum, mas também ambos podem ser combinados para o alcance de resultados mais eficazes.
- 78 A filosofia da modelagem ágil não admite decisões que levem um projeto a sua rejeição e a sua refatoração, pois os profissionais de tecnologia não possuem todas as respostas e outros *stakeholders* envolvidos no negócio devem ser respeitados e integrados ao processo.

No que se refere a testes ágeis, julgue os próximos itens.

- 79 Entre os métodos de testes ágeis, o BDD (*behavior driven development*) é aquele que, por natureza, é o mais orientado para o cliente.
- 80 Entre os métodos de testes ágeis, o TDD (*test-driven development*) é uma extensão do BDD (*behavior driven development*).

Julgue os itens subsequentes, relativos à qualidade de *software* e a seus atributos e custos.

- 81 Funcionalidade, atributo fundamental de qualidade para *software*, é aquele que avalia o grau com que o *software* satisfaz às necessidades declaradas por seus subatributos, tais quais adequabilidade, exatidão, interoperabilidade, conformidade e segurança.
- 82 Entre as atividades que ajudam uma equipe a atingir o alto padrão de qualidade de *software*, a garantia da qualidade é aquela que engloba um conjunto de ações de engenharia de *software* que contribui para que cada produto resultante atinja suas metas de qualidade.
- 83 O cálculo do custo da qualidade engloba os custos necessários para a execução de atividades relacionadas à qualidade, mas não os custos gerados pela falta de qualidade.

A respeito da arquitetura de banco de dados relacional, julgue os itens seguintes.

- 84 Um modelo relacional de banco de dados não separa as estruturas de armazenamento físicas das estruturas de dados lógicas, assim administradores de banco de dados podem gerenciar tanto o armazenamento de dados físicos quanto o acesso a esses dados.
- 85 A fim de evitar a criação de novos códigos para funções de acesso a dados, os bancos de dados relacionais permitem procedimentos armazenados, os quais são blocos de código que podem ser acessados com uma simples chamada de aplicativos.
- 86 A atomicidade, uma das propriedades essenciais de um modelo relacional de banco de dados, define todos os elementos que compõem uma transação completa do banco de dados.

No que se refere à álgebra relacional e a SQL, julgue os itens a seguir.

- 87 Considerando-se uma tabela nomeada *empregados* que contém, entre outras colunas, uma identificada como *nome*, o comando SQL que retorna o nome de todos os empregados que tenham o nome *luiz* ou *luis* é o seguinte.
SELECT nome FROM empregados WHERE nome like '%lui_%'
- 88 Em álgebra relacional, a operação que produz uma combinação das linhas de uma relação com as linhas correspondentes de outra relação é conhecida como interseção.
- 89 Na álgebra relacional, a operação projeção, representada pela letra grega π (π), é classificada como uma operação unária, por operar sobre apenas um conjunto de entrada.
- 90 Na linguagem SQL, os comandos GRANT e REVOKE são exemplos do subconjunto DDL (linguagem de definição de dados), pois definem o acesso aos dados do banco.

Julgue os itens a seguir, acerca de engenharia de requisitos.

- 91 Requisitos organizacionais são requisitos de sistema amplos, derivados das políticas e dos procedimentos nas organizações do cliente e do desenvolvedor, cujas funções incluem definir como o sistema será utilizado e especificar a linguagem de programação.
- 92 Requisitos não funcionais de um sistema descrevem seu objetivo e dependem do tipo de *software* a ser desenvolvido, dos usuários esperados para o *software* e da abordagem geral adotada pela organização ao escrever os requisitos.

A respeito dos conceitos de teste de integração e de teste unitário, julgue os itens subsequentes.

- 93 Os testes unitários são realizados com o objetivo de isolar cada parte do sistema para garantir que elas estejam funcionando conforme especificado.
- 94 O objetivo do teste de integração é verificar se os requisitos atendem a especificação e se as funcionalidades do sistema foram implementadas corretamente, sendo todo o sistema testado de modo a simular um ambiente de execução real.

Julgue o item a seguir, com relação aos testes de RNF (carga, estresse).

- 95 Os testes de estresse devem verificar o uso da memória ao longo do tempo para garantir que não existam perdas acumulativas.

Acerca do conceito de revisão por pares, julgue o item subsequente.

- 96 A revisão por pares é uma forma de análise da causa-raiz, na qual a equipe define uma meta ou efeito arquitetural e, então, enuncia as ações relacionadas para o alcance da meta.

Julgue os itens a seguir, relativos aos conceitos de TDD e BDD.

- 97 No processo de TDD, o código é desenvolvido em grandes blocos de requisitos do usuário. Cada iteração resulta em um novo teste, que faz parte um conjunto de testes de regressão executado no final do processo de integração.
- 98 Desenvolvedores que se beneficiam das vantagens do BDD escrevem os testes em sua língua nativa, em combinação com a linguagem ubíqua.

No que concerne a qualidade de *software*, julgue os itens seguintes.

- 99 Define-se confiabilidade de *software* como a probabilidade de operação, sem falhas, de um programa de computador em dado ambiente por determinado tempo.
- 100 De acordo com a SQA (*software quality assurance*), correção, completude e consistência do modelo de requisitos são características da qualidade do código que influenciam a qualidade de todos os produtos.

Acerca dos conceitos de gestão de configuração, julgue os itens que se seguem.

- 101 Os principais recursos de um sistema de controle de versão incluem um repositório de dados que armazena todos os objetos de configuração relevantes e um recurso de gestão de versão que armazena todas as versões de um objeto de configuração.
- 102 A prática DevOps visa à entrega contínua e à automação de tarefas repetitivas e tediosas, o que possibilita dedicar mais tempo à interação humana.

Julgue o item a seguir, a respeito dos padrões GRASP (*general responsibility assignment software patterns*).

- 103 Entre os padrões definidos pelo GRASP, destacam-se baixa coesão e alto acoplamento.

No que se refere aos conceitos de orientação a objetos, julgue o item a seguir.

- 104 Nas classes abstratas, que servem de modelo para outras classes, é obrigatória a existência de pelo menos um método abstrato, sem corpo.

Julgue o item a seguir, com relação aos conceitos de SOLID.

- 105 Os princípios de programação orientada a objetos que correspondem aos princípios SOLID são: criador (*creator*), especialista na informação (*information expert*), controlador (*controller*), polimorfismo (*polymorphism*), fabricação pura (*pure fabrication*).

Julgue os próximos itens, relativos a conceitos, processos e ciclo de vida dos testes de *software*.

- 106** O teste estático é uma técnica de verificação de *software* que revisa o código do programa para identificar se ele representa as especificações produzidas para o *software*.
- 107** É possível que um defeito que resida em código sem causar uma falha não seja encontrado em um teste dinâmico.
- 108** A complexidade ciclomática, usada no contexto do método de teste de caminho básico, encontra o número de rotas de execução de um programa para percorrer os comandos elementares pelo menos uma vez.
- 109** No teste de fumaça (*smoke test*), os códigos do *software* são integrados em componentes bloqueadores de erros com módulos reutilizáveis necessários para implementar as funções do *software*.

O algoritmo a seguir representa uma parte de um sistema de avaliação de alunos. O algoritmo exhibe, com exatidão, as regras do negócio e os desejos do usuário. A linguagem usada para implementar esse algoritmo preservará a estrutura e a lógica exatamente como representadas.

Algoritmo "decide a lista de aprovados"

```

Var
faltas: inteiro
limite: inteiro
média: real

Início
// programa deve analisar até 8 alunos
faltas <- 0
média <- 0
limite <- 1

enquanto(limite < 9) faça
  escreva("informe a média das notas")
  leia(faltas)
  escreva("informe a média das notas")
  leia(média)
  // reprovado com mais de 20 faltas
  se(faltas > 20) então
    escreva("por faltas, não vai para
lista")
  senão
    se(média > 6) então
      escreva("pode ir para lista")
    fimse
  fimse
  limite <- limite - 1
fimenquanto
Fimalgoritmo

```

Com base nessas informações, julgue os itens a seguir, acerca da previsão de testes a serem realizados no trecho de código apresentado.

- 110** No teste de caixa branca, conforme indicação do algoritmo, estima-se que o número mínimo de casos de teste de comando para 100% da cobertura de declaração do código será de dois casos.
- 111** No teste de caixa branca, conforme indicação do algoritmo, estima-se que o número mínimo de casos de teste para 100% da cobertura de decisão do código será de quatro casos.
- 112** Submetendo-se apenas os seguintes casos de teste, a cobertura de decisão alcançada seria de 25%.
- Caso 1 --> limite (enquanto) = 10;
faltas = 0; menção = 0
- Caso 2 --> limite (enquanto) = 1;
faltas = 25; menção = 0
- 113** Submetendo-se apenas o seguinte caso de teste, a cobertura de declaração alcançada seria de 50%.
- limite (enquanto) = 1; faltas = 18;
menção = 8

Em um sistema de controle escolar, o *software* tem de selecionar os estudantes que deverão realizar uma prova de recuperação. Para prestar a prova, o estudante deve, no mínimo, ter obtido uma nota 5,0 durante o ano letivo, mas não ter conseguido nota 7,0 ou superior. As notas têm precisão de 0,1.

A partir dessa situação hipotética, julgue os itens a seguir, relativos aos testes de caixa preta.

- 114** O conjunto de valores de entrada $S = \{4,9; 5,0; 6,9; 7,0; 7,1\}$ cobre os valores-limite para um aluno estar em recuperação.
- 115** O conjunto $S = \{4,9; 5,0; 6,9; 7,0\}$ representa o mínimo de quantidades desejadas que cobrem as classes de equivalência dos critérios da prova de recuperação.

Acerca da análise, do controle e da gestão de testes e casos de testes de *software*, julgue os itens a seguir.

- 116** Antes de utilizar o Selenium em um projeto de automação de testes, é preciso instalar as bibliotecas de linguagem para a linguagem escolhida e os *drivers* para os navegadores definidos.
- 117** No desenvolvimento orientado a comportamento (BDD), os ciclos iniciam-se com a criação de testes de unidade e integração.
- 118** Na programação de um *script* Selenium, a sincronização do código ao estado atual do navegador pode ser realizada com uma espera explícita, durante a qual o código interrompe a execução do programa até que se resolva a condição definida ou decorra o limite de espera da chamada.
- 119** Na gestão de defeitos, o princípio de teste da regra 10 de Myers estabelece que as atividades de teste estático e dinâmico devem ser planejadas muito antes de serem iniciadas.
- 120** O Appium foi projetado para, entre outras finalidades, automatizar testes sem a necessidade de recompilar códigos ou estruturas específicas do próprio Appium ou de terceiros.

Espaço livre