

CONHECIMENTOS ESPECÍFICOS

Julgue os próximos itens a respeito de segurança da informação.

- 51 Na criptografia simétrica, são geradas duas chaves criptográficas, uma privada e outra pública, para que um arquivo seja transferido, entre dois computadores, de forma criptografada.
- 52 O *hash* poderá auxiliar na verificação da integridade de um arquivo transferido de um computador para outro.
- 53 Assinatura digital é um mecanismo capaz de garantir a autenticidade e a integridade de um arquivo transferido via Internet.
- 54 Considere que um arquivo que esteja sendo transferido entre dois usuários tenha sido interceptado e seu conteúdo tenha sido visualizado e encaminhado a outros usuários. Nessa situação, caracterizou-se a ocorrência do comprometimento da integridade do arquivo.
- 55 Se um sítio da web sofrer comprometimento devido a problemas de *hardware* no servidor, impossibilitando a visualização do conteúdo pelos usuários, esse fato poderá ser considerado como comprometimento da disponibilidade do serviço.

Com relação às políticas de segurança da informação e à gerência de riscos, julgue os itens a seguir.

- 56 A política de segurança da informação deverá ser analisada criticamente em intervalos planejados, incluindo-se na análise as tendências relacionadas a ameaças e vulnerabilidades.
- 57 Os riscos devem ser identificados, quantificados e priorizados, com base nos critérios definidos pela área de tecnologia da informação (TI) da organização.
- 58 Os riscos devem ser avaliados antes, durante e após implantações e revisões de políticas de segurança da informação.

A respeito de projetos, julgue os itens que se seguem, considerando a quarta edição do PMBOK.

- 59 O grupo de processos de execução é o responsável pelos resultados das medições do controle de qualidade e por auditar os requisitos de qualidade.
- 60 Coleta e distribuição de informações acerca do desempenho são atividades realizadas pelo grupo de monitoramento e controle.
- 61 O grupo de processos de iniciação é o responsável pelo monitoramento e controle do projeto.

Acerca do MPS.BR, julgue os itens subsequentes.

- 62 O nível E do MPS.BR contempla um processo que resulta no registro dos dados de utilização dos ativos reutilizáveis.
- 63 O processo de gerenciamento de riscos é contemplado, no MPS.BR, a partir do nível F, conhecido como nível gerenciado.

Com relação à norma ISO/IEC 12.207, julgue os itens a seguir.

- 64 A criação, pela organização, da infraestrutura necessária à execução dos outros processos faz parte do processo de gerência, classificado como um processo primário.
- 65 O processo de documentação, essencial durante as atividades de desenvolvimento e manutenção de *software*, é classificado como um dos processos de apoio.

A respeito da utilização do banco de dados relacional em plataforma PostgreSQL, julgue o próximo item.

- 66 Considere que as linhas apresentadas a seguir estejam presentes no arquivo de nome `pg_hba.conf` de instalação do PostgreSQL. Nesse caso, será permitido acesso a todos os computadores, exceto aquele que esteja com o endereço IP 200.217.23.234.

TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD
host	all	all	200.217.23.234	255.255.255.255	reject
host	all	all	0.0.0.0	0.0.0.0	md5

Com relação às metodologias ágeis de desenvolvimento, julgue os itens subsequentes.

- 67 No método XP (*eXtreming programming*), os sistemas são concebidos a partir de uma metáfora e descritos em estórias do usuário. Esse método busca facilitar a comunicação com o cliente, entendendo a realidade deste e guiando o desenvolvimento com o uso de estória simples.
- 68 Na metodologia Scrum, a equipe trabalha nos processos e não há cargos na equipe. Como um dos papéis necessários, o Scrum *master* deve garantir que o processo seja entendido e atuar como facilitador para ajudar a equipe.

Julgue os seguintes itens quanto à arquitetura de banco de dados relacional e hierárquico.

- 69 Para armazenar as informações da estrutura organizacional de uma empresa com suas hierarquias, desde o presidente até o auxiliar administrativo, é necessário utilizar um banco de dados hierárquico.
- 70 Em um banco de dados relacional, a propriedade da integridade referencial é utilizada para garantir que os valores dos campos não extrapolem seus domínios, por exemplo um campo denominado *sexo* pode receber somente os valores *masculino* ou *feminino*.

Acerca das interfaces *web* Ajax e CSS, julgue os itens subsequentes.

- 71 *Media types* servem para direcionar um CSS para determinado tipo de meio de acesso. Alguns tipos de mídias suportadas são *screen*, *print* e *braille*.
- 72 Ajax é o uso metodológico de tecnologias como Javascript e XML. Para a recuperação assíncrona de dados, o Ajax utiliza o objeto XMLHttpRequest; a comunicação deve realizar-se por meio de requisições HTTPS; e o retorno dos dados deve ocorrer em formato XML.

A propósito de linguagens de implementação de regras de negócio orientadas a objeto, julgue os itens que se seguem.

73 Considere o código PHP mostrado a seguir.

```
<?php
$a = 10;
$b = "10";
if ($a == $b)
    print "a Igual b";
else
    print "a diferente b";
print "\n";
if ($a === $b)
    print "a Igual b";
else
    print "a diferente b";
?>
```

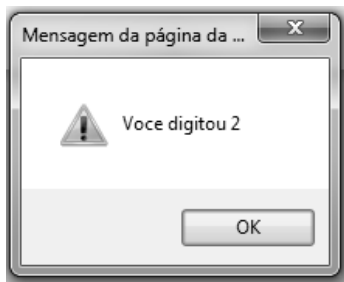
Esse código, caso executado, apresentará a resposta abaixo.

a igual b
a diferente b

74 Considere o código JavaScript mostrado a seguir.

```
<script>
suaEscolha=window.prompt('Entre com um valor
entre 1 e 3');
switch(SuaEscolha){
    case '1':alert('Voce digitou 1');break;
    case '2':alert('Voce digitou 2');break;
    case '3':alert('Voce digitou 3');break;
    default:alert('Permitido somente 1, 2 ou
3');
}
</script>
```

Se o usuário digitar o valor 2, esse código apresentará um resultado como o mostrado abaixo.



Com referência aos servidores de *web* e de aplicação, julgue os itens a seguir.

75 Em um servidor *web* Apache, a diretiva `Timeout` define o tempo máximo, em segundos, que o servidor irá esperar, mantendo uma conexão aberta com o cliente.

76 Na configuração do Tomcat, o arquivo `WEB-INF/server.xml` contém o descrito do contexto da aplicação *web*, segundo a especificação Java Servlet. As informações contidas nesse arquivo são as configurações específicas da aplicação.

Julgue os itens subsequentes com relação a linguagens SQL e a mecanismos de persistência.

77 Para ajustar o comportamento do Hibernate, pode ser utilizada a propriedade denominada `hibernate.xml2ddl.auto`, que executa a linguagem de manipulação de dados e pode assumir os valores `update-valid`, `create-update` ou `create-drop`.

78 Considere, em uma base de dados relacional, as seguintes tabelas, que contêm informações acerca de empregados, departamentos e a vinculação entre eles.

```
Departamento (CodDepto (PK), NomeDepto)
Empregado (CodEmp (PK), NomeEmp, CodDepto(FK))
```

Para serem obtidos os nomes dos departamentos que não estejam vinculados a nenhum empregado, é correto utilizar a consulta em SQL mostrada a seguir.

```
(SELECT NomeDepto
FROM Departamento D)
EXCEPT
(SELECT NomeDepto
FROM Departamento D, Empregado E
WHERE D.CodDepto=E.CodDepto)
```

No que concerne à UML, julgue o item abaixo.

79 A UML foi projetada primariamente para o processamento de dados simbólicos. Ela é uma linguagem formal matemática, tendo sido pioneira em aplicações como administração automática de armazenamento, linguagens interpretadas e programação funcional.

Com relação à tecnologia de desenvolvimento móvel Android, julgue os próximos itens.

80 No Android, os componentes `Intents` são criados a partir de ações do usuário e representam a intenção de se realizar alguma atividade, como iniciar o aplicativo de correio eletrônico ou abrir uma página, utilizando-se do navegador que acompanha o Android. O código mostrado a seguir apresenta corretamente um exemplo de utilização desse componente.

```
Uri uri = Uri.parse("http://www.cespe.unb.br");
Intent intent = new Intent(Intent.ACTION_VIEW,
uri);
startActivity(intent);
```

81 Para se apresentar uma lista de itens que permita ao usuário interagir com esses itens, clicando-se sobre eles para a realização de operações, deve-se utilizar a classe `ListItems`.

Em relação à implementação de banco de dados, julgue os itens subsecutivos.

82 O projeto físico é direcionado para um SGBD específico como, por exemplo, Oracle, Postgresql ou SQLServer. Decisões tomadas durante o projeto físico para melhorar o desempenho podem afetar a estrutura do esquema lógico.

83 Um SGBD garante que haja repetição de informação para que seja possível aplicar as regras de normalização.

```
CREATE TABLE sede
(
  codigo numeric(7,0),
  nome varchar(50),
  local numeric(12,0),
  apelido varchar(10),
  CONSTRAINT pk_sede PRIMARY KEY (codigo),
  CONSTRAINT uq_sede UNIQUE (apelido)
);

CREATE TABLE salas
(
  id numeric(7,0) NOT NULL,
  local varchar(10),
  descricao varchar(50),
  area numeric(10,2),
  CONSTRAINT pk_salas PRIMARY KEY (id),
  CONSTRAINT fk_sede_sala FOREIGN KEY (local)
REFERENCES sede (apelido)
);
```

Considerando as tabelas acima, criadas em SQL, julgue os itens que se seguem.

- 84 A expressão de consulta mostrada a seguir apresenta como resultado o local da sede, a área da sala e a descrição da(s) sala(s) com a maior área.
- ```
select c1.local, c1.nome, c2.descricao
from sede as c1,
(
 select local, descricao, area from salas as c1
 where area = (select max(area) from salas as
c2 where area>0)
) as c2 where c2.local=c1.apelido;
```
- 85 A expressão de consulta mostrada abaixo permite visualizar o nome da sede, com a área da sala e sua descrição.
- ```
select nome, area, descricao from salas, sede
where sede.local=salas.local
```

Com relação à engenharia de *software*, julgue os itens seguintes.

- 86 A abordagem iterativa e a incremental compõem o desenvolvimento em fases. Na primeira, o sistema é dividido em subsistemas por funcionalidades, adicionando-se mais funcionalidades a cada versão; na segunda, o sistema é entregue completo e muda a funcionalidade a cada nova versão.
- 87 A engenharia de *software* define quatro fases para o ciclo de vida de um sistema: iniciação, elaboração, construção e transição, sendo essa última responsável pela homologação dos artefatos junto ao cliente.
- 88 Engenharia de *software* não está relacionada somente aos processos técnicos de desenvolvimento de *softwares*, mas também a atividades como gerenciamento de projeto e desenvolvimento de ferramentas, métodos e teorias que apoiem a produção de *softwares*.
- 89 Controle, planejamento e garantia de qualidade são atividades do gerenciamento de qualidade; o controle de qualidade estabelece procedimentos e padrões que objetivam o desenvolvimento de *software* com qualidade.
- 90 Sistemas que incluem *software* são classificados em duas categorias: sistemas técnicos embasados em computadores e sistemas sociotécnicos. Os primeiros incluem componentes de *hardware*, *software*, pessoas, procedimentos e processos; os segundos são regidos pelas políticas e regras organizacionais.

Com referência à engenharia e gestão de requisitos de *software*, julgue os próximos itens.

- 91 De acordo com a evolução dos requisitos, estes podem ser classificados em permanentes, que são gerados nas fases iniciais do desenvolvimento, e voláteis, que surgem ao longo do processo de construção do *software*.
- 92 A gerência de requisitos deve manter a matriz de rastreabilidade atualizada para, caso o cliente solicite uma mudança, o item de configuração correspondente seja implementado.
- 93 Após a identificação, os requisitos devem ser modelados para se obter uma melhor compreensão do produto a ser desenvolvido. Os principais paradigmas de modelagem de requisitos são análise estruturada e análise orientada a objetos.
- 94 O desenvolvimento de requisitos é constituído por processos de elicitação de requisitos, análise e negociação de requisitos, especificação e modelagem dos requisitos e validação de requisitos.
- 95 Na análise estruturada, o modelo criado representa o fluxo e o conteúdo da informação, dividido em partições funcionais e comportamentais. Na análise orientada a objetos, o objetivo é modelar os objetos do domínio do produto, seus relacionamentos e comportamentos.
- 96 A rastreabilidade bidirecional deve ocorrer tanto de forma horizontal quanto vertical. A horizontal estabelece a dependência de um requisito-fonte até o nível de decomposição mais baixo do produto, enquanto a rastreabilidade vertical estabelece a dependência dos requisitos entre si.

A respeito de teste de *software*, julgue os itens subsecutivos.

- 97 Os testes de *software* buscam garantir que as funcionalidades definidas para atender as necessidades do cliente foram implementadas. Portanto, só podem ser realizadas ao término da construção de um *software*.
- 98 No teste de *software*, defeitos em um produto podem provocar falhas, gerando erros, que são comportamentos inesperados em um *software*.
- 99 Os principais níveis de teste de *software* são os de caixa branca, os de caixa preta, os de sistema e os de aceitação.
- 100 No *V-model*, que integra o ciclo de vida de desenvolvimento de *software* ao ciclo de teste, a validação refere-se ao desenvolvimento, enquanto a verificação se refere ao teste.

Com relação à gerência de configuração de *software*, julgue os itens que se seguem.

- 101 Na perspectiva de desenvolvimento, a gerência de configuração pode ser dividida em controle de modificações, controle de versões e gerenciamento de construção.
- 102 Quando a *baseline*, gerada na função de controle da configuração, é selecionada para ser liberada para o cliente, ocorre a função de avaliação e revisão da configuração.

No que concerne à análise e ao projeto de *software*, julgue os próximos itens.

- 103 Requisitos não funcionais do sistema podem influenciar o estilo e a estrutura escolhida para uma aplicação, pois a arquitetura de sistema afeta seu desempenho, sua distribuição e manutenção.
- 104 A fase de análise define os requisitos do cliente, conforme as necessidades de negócio, e as considerações técnicas envolvidas, que se agrupam em uma solução tecnológica, compõem a fase de projeto de *software*.
- 105 Em projeto de *software*, a independência funcional pode ser medida pela coesão, isto é, pela interdependência relativa entre os módulos, e pelo acoplamento, ou seja, pela força funcional relativa de um módulo.

Acerca de *service-oriented architecture*, julgue os itens subsequentes, considerando que SOAP corresponde a *simple object access protocol* e WSDL corresponde a *web services description language*.

- 106 As mensagens SOAP são transmitidas em formato XML, cujo envelope é composto de cabeçalho e corpo. Apenas o cabeçalho é opcional, e, quando preenchido, é necessariamente composto de rota, dados de segurança, origem e destino.
- 107 O SOAP permite a troca de mensagens estruturadas em ambiente distribuído e descentralizado, com o uso de tecnologias XML. Essas mensagens podem ser trocadas por uma variedade de protocolos subjacentes como, por exemplo, o HTTP.
- 108 O WSDL descreve um *web service* utilizando a sintaxe XML. Está correta a sintaxe mostrada a seguir para o elemento `<portType>`, que define o protocolo e a porta de comunicação para troca de mensagens do *web services*.

```
<portType>
  <portType protocol = "HTTP">
  <portType port = "80">
</portType>
```

- 109 Os quatro principais elementos de um arquivo WSDL são `<message>`, `<portType>`, `<types>` e `<binding>`. A sintaxe mostrada a seguir, para o elemento `<message>`, que define os dados que estão sendo transmitidos, está correta.

```
<message name="getTesteRequest">
  <part name="nome" type="xs:string"/>
</message>
<message name="getTesteResponse">
  <part name="valor" type="xs:string"/>
</message>
```

Acerca de componentes de *softwares*, julgue os itens seguintes.

- 110 No contexto de desenvolvimento de uma solução de *software*, as etapas do ciclo de vida de um componente podem ser descritas como: análise e *design* da solução, implementação do componente, realização de testes unitários e sua integração com os demais componentes em *builds*.
- 111 Considerando o desenvolvimento de uma solução de *software* seguindo sequencialmente as fases de iniciação, elaboração, construção e transição, é possível criar os componentes na fase de iniciação, embora, sob o ponto de vista da arquitetura, esses componentes sejam criados na fase de elaboração e os demais na fase de construção, sendo atualizados durante a fase de transição à medida que defeitos são localizados e corrigidos.
- 112 Na implementação de uma solução de *software*, o componente representa um trecho de um código de *software*, um arquivo contendo a ajuda desse *software* ou a agregação de outros componentes como, por exemplo, um aplicativo composto de vários executáveis.

Acerca dos padrões GRASP, julgue os itens a seguir.

- 113 O padrão Pure Fabrication objetiva designar a responsabilidade unívoca pela criação de uma nova instância de uma classe.
- 114 Nos casos em que a solução oferecida pelo padrão Expert violar a alta coesão e o baixo acoplamento, o padrão adequado a ser aplicado será o Creator, que atribui um conjunto altamente coesivo de responsabilidades a uma classe artificial que não representa um conceito do domínio do problema.
- 115 O padrão Indirection é utilizado para atribuir responsabilidades à classe que tiver a informação necessária para satisfazer a responsabilidade.
- 116 O padrão Don't Talk to Strangers é utilizado para fortalecer o polimorfismo, realizado pelo padrão Polymorphism. O objetivo de ambos os padrões é substituir um componente sem afetar outro componente, embora o primeiro implemente o polimorfismo em nível de classe e o segundo lide com alternativas embasadas no tipo de componente.

Julgue os itens subsequentes a respeito dos padrões de projeto.

- 117 Uma das aplicabilidades do padrão Iterator é a representação de hierarquias do tipo todo-parte de objetos, de modo que a aplicação seja capaz de ignorar a diferença entre composições de objetos e objetos individuais, haja vista que todos os objetos tratados no padrão têm comportamento uniforme.
- 118 O padrão Adapter será mais apropriado que o Façade quando for necessário fornecer uma interface unificada para um conjunto de interfaces em um subsistema.
- 119 O uso do padrão Builder tem a vantagem de permitir acesso controlado à instância de uma classe, uma vez que ele encapsula a classe, criando um ponto global único de acesso.
- 120 Os padrões estruturais, tais como o Bridge e o Proxy, abstraem o processo de instanciação, ajudando a tornar um sistema independente de como os seus objetos são criados. Já os padrões de criação, como Prototype e o Builder, se preocupam com a forma como as classes e os objetos são criados para formar estruturas compostas maiores.