

## CONHECIMENTOS ESPECIALIZADOS

31) Analise as afirmativas abaixo.

- I. O nível interno tem um esquema interno, que descreve a estrutura do armazenamento lógico do banco de dados.
- II. O nível conceitual tem um esquema conceitual, que descreve a estrutura do banco de dados inteiro para uma comunidade de usuários.
- III. O nível externo ou de visão inclui uma série de esquemas externos ou visões do usuário.
- IV. Independência lógica de dados é a capacidade de alterar o esquema lógico sem precisar alterar os esquemas externos ou os programas de aplicações.
- V. Independência física de dados é a capacidade de alterar o esquema interno sem precisar alterar o esquema conceitual.

Estão corretas apenas as afirmativas

- a) I e IV.
- b) II e V.
- c) II e IV.
- d) III e V.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

É correto afirmar que:

- o nível interno tem um esquema interno, que descreve a estrutura do armazenamento físico do banco de dados;
- independência lógica de dados é a capacidade de alterar o esquema conceitual sem ter de alterar os esquemas externos ou os programas de aplicações.

Fonte: ELMASRI, R.; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6ª ed. São Paulo: Pearson, 2011.

32) A SQL tem uma instrução básica para recuperar informações de um banco de dados. Assinale a alternativa que contempla essa instrução.

- a) *Alter.*
- b) *Insert.*
- c) *Select.*
- d) *Update.*

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A SQL tem uma instrução básica para recuperar informações de um banco de dados: a instrução *Select*.

Fonte: ELMASRI, R.; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6ª ed. São Paulo: Pearson, 2011.

33) Com base no Modelo de Camadas OSI (*Open System Interconnection*), assinale a alternativa que contempla a camada relacionada à sintaxe e à semântica das informações transmitidas.

- a) Camada de Sessão.
- b) Camada de Aplicação.
- c) **Camada de Apresentação.**
- d) Camada de Enlace de Dados.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A camada que está relacionada à sintaxe e à semântica das informações transmitidas, no modelo OSI, é a camada de apresentação.

Fonte: TANENBAUM, A. S. Redes de Computadores. 5ª ed. São Paulo: Pearson, 2011.

**34)** Assinale a alternativa que apresenta a Camada do Modelo TCP/IP onde está alocado o protocolo ICMP (*Internet Control Message Protocol*).

- a) Camada de Enlace.
- b) Camada de Aplicação.
- c) Camada de Transporte.
- d) **Camada de Rede (ou Internet).**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

A Camada do Modelo TCP/IP onde está alocado o protocolo ICMP (*Internet Control Message Protocol*) é a camada de rede.

Fonte: TANENBAUM, A. S. Redes de Computadores. 5ª ed. São Paulo: Pearson, 2011.

**35)** Acerca do padrão de rede sem fio baseado na multiplexação por divisão ortogonal de frequência, ou OFDM (*Orthogonal Frequency Division Multiplexing*), assinale a alternativa correta.

- a) **802.11a.**
- b) 802.11b.
- c) 802.11g.
- d) 802.11n.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

O método 802.11a baseia-se na multiplexação por divisão ortogonal de frequência, ou OFDM (*Orthogonal Frequency Division Multiplexing*), pois a OFDM usa o espectro com eficiência e resiste a degradações do sinal sem fios, como o enfraquecimento por múltiplos caminhos.

Fonte: TANENBAUM, A. S. Redes de Computadores. 5ª ed. São Paulo: Pearson, 2011.

**36)** Associe as duas colunas, relacionando os protocolos de rede às respectivas camadas, utilizando como base o Modelo TCP/IP e, em seguida, assinale a alternativa que apresenta a sequência correta. (Alguns números poderão ser usados mais de uma vez.)

<u>Protocolos de rede</u>	<u>Camadas</u>
(1) Aplicação	( ) DSL.
(2) Transporte	( ) RTP.
(3) Rede	( ) TCP.
(4) Enlace	( ) IP.
	( ) DNS.
	( ) UDP.
	( ) <i>Ethernet</i> .

- a) **4 – 1 – 2 – 3 – 1 – 2 – 4**
- b) 2 – 4 – 1 – 1 – 4 – 3 – 3
- c) 3 – 2 – 3 – 4 – 2 – 1 – 2
- d) 1 – 3 – 4 – 2 – 3 – 4 – 1

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Os respectivos protocolos em suas camadas são:

- Aplicação: RTP; DNS
- Transporte: TCP; UDP
- Rede: IP
- Enlace: Ethernet; DSL

Fonte: TANENBAUM, A. S. Redes de Computadores. 5ª ed. São Paulo: Pearson, 2011.

37) A criptografia simétrica é caracterizada por utilizar o mesmo par de chaves no processo de codificação e decodificação. Acerca dos algoritmos simétricos, assinale a alternativa **incorreta**.

- a) DES.
- b) AES.
- c) **RSA.**
- d) DES Triplo.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

O RSA é um algoritmo de chave pública, ou algoritmo assimétrico.

Fonte: TANENBAUM, A. S. Redes de Computadores. 5ª ed. São Paulo: Pearson, 2011.

38) O padrão de projetos que garante que determinada classe tenha apenas uma instância e fornece um ponto global de acesso a ela é o

- a) *State*.
- b) *Factory*.
- c) ***Singleton***.
- d) *Decorator*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

O padrão *Singleton* garante que uma classe tenha apenas uma instância e fornece um ponto global de acesso a ela.

Fonte: FREEMAN, E. Use a cabeça!: Padrões de projetos (Design Patterns). 1ª ed. Alta Books, 2005

39) Assinale a alternativa que apresenta a função, em PHP, que apenas retorna informações sobre data/hora.

- a) *date*.
- b) ***getdate***.
- c) *gmdate*.
- d) *checkdate*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

As funções do PHP são:

checkdate → valida uma data/hora.

date → formata uma data/hora.

getdate → retorna informações sobre data/hora

gmdate → formata uma data/hora GMT/CUT.

Fonte: NIEDERAUER, J. PHP para quem conhece PHP. 3ª ed. São Paulo: Novatec, 2008.

40) Sobre a linguagem Java, informe se é verdadeiro (V) ou falso (F) o que se afirma abaixo. A seguir, assinale a alternativa que apresenta a sequência correta.

- ( ) O Java não diferencia letras maiúsculas de letras minúsculas, ou seja, para o Java, "Brasil" e "brasil" têm o mesmo significado.
- ( ) Uma sequência de caracteres entre aspas duplas é chamada de *string*, *string* de caracteres, mensagem ou literal de *string*.
- ( ) Uma instrução de declaração de variável especifica o nome e o tipo de uma variável.
- ( ) O método *scanner.nextInt* é utilizado para se obter um número inteiro que será utilizado numa classe do programa.
- ( ) O método *main* é o ponto de partida de cada aplicativo Java e deve iniciar com: *public static void main (String [ ] args)*.
- ( ) As instruções orientam o computador a realizar ações.
- ( ) Uma declaração da classe *public* deve ser salva em um arquivo com o mesmo nome da classe seguido pela extensão do nome do arquivo "Java".

- a) V – F – V – V – V – F – F
- b) **F – V – V – F – F – V – V**
- c) F – V – F – F – F – V – F
- d) V – F – F – V – V – F – V

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

É correto afirmar que:

- o Java diferencia letras maiúsculas de letras minúsculas.
- o método *main* é o ponto de partida de cada aplicativo Java e deve iniciar com: *public static void main (String args [ ])*
- o método *Scanner nextInt* obtém um número inteiro para uso em um programa.

Fonte: DEITEL, H. M.; DEITEL, P. J. Java: Como Programar. 6ª ed. São Paulo: Pearson, 2005.

**41)** Acerca das funções do PHP, leia os trechos abaixo. A seguir, assinale a alternativa que preenche correta e respectivamente as lacunas.

- Função que lê no máximo uma linha: \_\_\_\_\_
- Função que especifica a quantidade de informações a serem lidas: \_\_\_\_\_
- Função que lê apenas um caractere por vez: \_\_\_\_\_
- Função que lê todo o conteúdo de um arquivo e o envia diretamente ao *buffer* de saída: \_\_\_\_\_
- Lê uma linha do arquivo e retira as *tags* HTML e PHP: \_\_\_\_\_
- Função que posiciona o ponteiro em um arquivo: \_\_\_\_\_

a) *fseek* / *readfile* / *fgetss* / *fread* / *fgetss* / *fgetc*

b) *fgetc* / *fgetss* / *readfile* / *fgetss* / *fseek* / *fread*

c) *readfile* / *fgetss* / *fseek* / *fgetc* / *fread* / *fgetss*

d) *fgetss* / *fread* / *fgetc* / *readfile* / *fgetss* / *fseek*

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Ressalta-se que os comandos corretos são:

- função que lê no máximo uma linha: *fgets*;
- função que especifica a quantidade de informações a serem lidas: *fread*;
- função que lê apenas um caractere por vez: *fgetc*;
- função que lê todo o conteúdo de um arquivo e o envia diretamente ao *buffer* de saída: *readfile*;
- lê uma linha do arquivo e retira as *tags* HTML e PHP: *fgetss*;
- função que posiciona o ponteiro em um arquivo: *fseek*.

Fonte: NIEDERAUER, J. PHP para quem conhece PHP. 3ª ed. São Paulo: Novatec, 2008.

**42)** RISC, ou seja, *Reduced Instruction Set Computer* (computador com um conjunto de instruções reduzido) foi o termo cunhado em 1980 por um grupo, em *Berkeley*, liderado por *David Patterson* e *Carlo Séquin*. São características da Arquitetura RISC, **exceto**:

a) arquitetura *pipelining*.

b) instruções com formato fixo.

c) instruções executadas pelo *hardware*.

d) arquitetura com poucos registradores.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

A arquitetura com poucos registradores é uma característica da Arquitetura CISC.

Fonte: MAIA, L. P.; MACHADO, F. B. Arquitetura de Sistemas Operacionais. 4ª ed. Rio de Janeiro: LTC, 2007.

**43)** Acerca das informações sobre o conjunto de instruções da Arquitetura Intel 8051, associe as colunas, relacionando-as, e, em seguida, assinale a alternativa que apresenta a sequência correta.

(1) JNC ( ) salto indireto relativo para DPR + ACC.

(2) AJMP ( ) salto relativo se vai-um não marcado.

(3) JNZ ( ) salto relativo se vai-um marcado.

(4) JMP ( ) salto se ACC for não-zero.

(5) JV ( ) salto relativo (endereço de 11 *bits*).

a) 3 – 5 – 1 – 2 – 4

b) 4 – 1 – 5 – 3 – 2

c) 2 – 4 – 3 – 1 – 5

d) 5 – 3 – 2 – 4 – 1

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

A relação adequada é:

JNC: salte relativo se vai-um não marcado;

AJMP: salto relativo (endereço de 11 bits);

JNZ: salte se ACC for não-zero;

JMP: salto indireto relativo para DPR + ACC;

JV: salte relativo se vai-um marcado.

Fonte: TANEMBAUM, Andrews S. Organização Estruturada de Computadores. 5ª ed. São Paulo: Pearson, 2007.

**44)** Segundo *Tanenbaum* (2007), barramentos são a cola que mantém a integridade dos sistemas de computadores. Assinale a afirmativa que apresenta uma característica do Barramento PCI.

- a) Originalmente transferia 32 bits por ciclo e funcionava em 33 MHz (tempo de ciclo de 30 nanossegundos) para uma largura de banda total de 133 MB/s.
- b) Livrar-se do barramento paralelo com seus muitos mestres e escravos e passar para um projeto baseado em conexões seriais ponto a ponto de alta velocidade.
- c) Tinha 62 linhas de sinais, incluindo 20 para endereço de memória, 8 para dados e 1 linha de cada para ativar leitura de memória, escrita na memória, leitura de E/S e escrita de E/S.
- d) Era o padrão *de facto* dos sistemas que usavam o 8088, porque, praticamente, todos os fabricantes de clones do PC o copiaram para permitir que as muitas placas de E/S existentes, fabricadas por terceiros, fossem usadas com seus sistemas.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Barramento ISA: Era o padrão *de facto* dos sistemas que usavam o 8088 porque praticamente todos os fabricantes de clones do PC o copiaram para permitir que as muitas placas de E/S existentes fabricadas por terceiros fossem usadas com seus sistemas.

Barramento PCI-Express: Se livrar do barramento paralelo com seus muitos mestres e escravos e passar para um projeto baseado em conexões seriais ponto a ponto de alta velocidade.

Barramento ISA: Tinha 62 linhas de sinais, incluindo 20 para endereço de memória, 8 para dados e uma linha de cada para ativar leitura de memória, escrita na memória, leitura de E/S e escrita de E/S.

Fonte: TANEMBAUM, Andrews S. Organização Estruturada de Computadores. 5ª ed. São Paulo: Pearson, 2007.

**45)** Em relação ao tratamento de exceções na linguagem Java, leia o trecho abaixo. A seguir, assinale a alternativa que preenche correta e respectivamente as lacunas.

O Java garante que um bloco \_\_\_\_\_ fornecido executará se uma exceção for lançada no bloco \_\_\_\_\_ correspondente ou em qualquer um de seus blocos \_\_\_\_\_ correspondentes.

- a) *finally / try / catch*
- b) *throw / try / throwable*
- c) *exception / catch / try*
- d) *throwable / catch / try*

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

O Java garante que um bloco *finally* fornecido executará se uma exceção for lançada no bloco *try* correspondente ou em qualquer um de seus blocos *catch* correspondentes.

Fonte: DEITEL, H. M.; DEITEL, P. J. Java: Como Programar. 6ª ed. São Paulo: Pearson, 2005.

**46)** Além dos registradores de propósito geral, usados para conter variáveis e resultados temporários, a maioria dos computadores têm vários registradores especiais visíveis ao programador. Assinale a afirmativa correta acerca do registrador contador de programa.

- a) Aponta para o topo da pilha atual de memória.
- b) Contém o endereço de memória da próxima instrução a ser buscada.
- c) Desempenha um papel importante nas chamadas de sistema e em E/S.
- d) Contém os *bits* do código de condições, os quais são alterados pelas instruções de comparação, pelo nível de prioridade da CPE, pelo modo de execução (usuário ou núcleo) e por vários outros *bits* de controle.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Registrador ponteiro da pilha: aponta para o topo da pilha atual de memória;

Registrador PSW (Program Status Word): contém os *bits* do código de condições, os quais são alterados pelas instruções de comparação, pelo nível de prioridade da CPE, pelo modo de execução (usuário ou núcleo) e por vários outros bits de controle;

Registrador PSW (Program Status Word): desempenha um papel importante nas chamadas de sistema e em E/S.

Fonte: TANENBAUM, Andrews S. Sistemas Operacionais Modernos. 3ª ed. São Paulo: Pearson, 2010.

**47)** Os sistemas operacionais precisam de mecanismos para criar processos. Há quatro eventos principais que fazem com que processos sejam criados. (Tanenbaum, 2009). Estes eventos são, **exceto**:

- a) início do sistema.
- b) início de uma tarefa em lote (*batch job*).
- c) **requisição do sistema para criar um novo processo.**
- d) execução de uma chamada de sistema de criação de processo por um processo em execução.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A afirmativa adequada é que uma requisição do usuário para criar um novo processo.

Fonte: TANENBAUM, Andrews S. Sistemas Operacionais Modernos. 3ª ed. São Paulo: Pearson, 2010.

**48)** Escalonamento é uma técnica dos Sistemas Operacionais que equaciona a utilização do processador e emprega algoritmos de escalonamento para esse fim. O escalonador é responsável por isso. Existem o escalonamento em sistemas em lote, interativos e de tempo real. Assinale a alternativa que apresenta o algoritmo de escalonamento em sistemas de lote.

- a) **Tarefa mais curta primeiro.**
- b) Próximo processo mais curto.
- c) Escalonamento por fração justa.
- d) Escalonamento por chaveamento circular.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Escalonamento por chaveamento circular; próximo processo mais curto; escalonamento por fração justa pertencem ao escalonamento em sistemas interativos.

Fonte: TANENBAUM, Andrews S. Sistemas Operacionais Modernos. 3ª ed. São Paulo: Pearson, 2010.

**49)** “Processos que ficam em *background* com a finalidade de lidar com alguma atividade, como mensagem eletrônica, páginas da *web*, notícias, impressão, entre outros.” A descrição anterior refere-se a

- a) pai.
- b) filho.
- c) **daemons.**
- d) sequenciais.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Processos que ficam em *background* com a finalidade de lidar com alguma atividade como mensagem eletrônica, páginas da *web*, notícias, impressão, entre outros são chamados de *daemons*.

Fonte: TANENBAUM, Andrews S. Sistemas Operacionais Modernos. 3ª ed. São Paulo: Pearson, 2010.

50) “Para que o *hardware* tenha utilidade prática deve existir um conjunto de programas utilizado como interface entre as necessidades do usuário e as capacidades do *hardware*.” (Maia, L.P. / Machado, F. B., 2007)

De acordo com essa afirmativa, assinale o utilitário responsável por carregar na memória principal um programa a ser executado.

- a) *Linker*.
- b) *Loader*.
- c) Depurador.
- d) Interpretador.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

É correto afirmar que:

Interpretador: considerado um tradutor que não gera módulo-objeto.

Linker: utilitário responsável por gerar, a partir de um ou mais módulos-objeto, um único programa executável.

Depurador: utilitário que permite ao usuário acompanhar toda a execução de um programa, a fim de detectar erros na sua lógica.

Fonte: MAIA, L. P.; MACHADO, F. B. Arquitetura de Sistemas Operacionais. 4ª ed. Rio de Janeiro: LTC, 2007.

51) “Servem para imitar a funcionalidade dos *threads* de núcleo – porém com melhor desempenho e maior flexibilidade – em geral associados aos pacotes de *threads* de usuário.” A afirmativa anterior refere-se a

- a) *upcall*.
- b) *threads pop-up*.
- c) implementações híbridas.
- d) *ativações do escalonador*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

As ativações do escalonador servem para imitar a funcionalidade dos *threads* de núcleo – porém com melhor desempenho e maior flexibilidade –, em geral associados aos pacotes de *threads* de usuário.

Fonte: TANENBAUM, Andrews S. Sistemas Operacionais Modernos. 3ª ed. São Paulo: Pearson, 2010.

52) Os sistemas operacionais implementam, basicamente, três estratégias para determinar em qual área livre um programa será carregado para execução. Assinale a técnica que **não** pertence a essas estratégias.

- a) *Last-fit*.
- b) *First-fit*.
- c) *Best-fit*.
- d) *Worst-fit*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

A única técnica que não pertence às estratégias é *Last-fit*, pois não pertence a nenhuma classe citada.

Fonte: MAIA, L. P.; MACHADO, F. B. Arquitetura de Sistemas Operacionais. 4ª ed. Rio de Janeiro: LTC, 2007.

53) Em Engenharia de *Software*, o teste cuja finalidade é mostrar ao cliente o *software* desejado que atende ao seus requisitos é o

- a) Teste Alfa.
- b) Teste Beta.
- c) Teste de Inspeção.
- d) *Teste de Validação*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

O teste de validação tem a finalidade de mostrar que o *software* é o que o cliente deseja – aquele que atende a seus requisitos.

Fonte: SOMMERVILLE, I. Engenharia de Software. 9ª ed. São Paulo: Addison Wesley Brasil, 2011.

54) Sommerville (2011) apresenta, em seu livro Engenharia de *Software*, uma estrutura de um plano de teste de *software*. Associe as colunas, relacionando a estrutura às respectivas características e, em seguida, assinale a alternativa que apresenta a sequência correta.

<u>Estrutura</u>	<u>Características</u>
(1) Processo de teste	( ) cronograma global de testes e alocação de recursos para esse cronograma, obviamente vinculado ao cronograma mais geral de desenvolvimento de projeto.
(2) Rastreabilidade de requisito	( ) restrições que afetam o processo de teste, como falta de pessoal, devem ser antecipadas nesta seção.
(3) Itens testados	( ) descrição das fases principais do processo de teste.
(4) Cronograma de testes	( ) usuários são os mais interessados em que o sistema atenda a seus requisitos e que os testes sejam planejados de maneira que todos os requisitos sejam individualmente testados.
(5) Procedimento de registro de testes	( ) produtos do processo de <i>software</i> a serem testados devem ser especificados.
(6) Requisitos de <i>hardware</i> e <i>software</i>	( ) não é suficiente realizar simplesmente os testes, os resultados dos testes devem ser sistematicamente registrados.
(7) Restrições	( ) esta seção deve estabelecer as ferramentas de <i>software</i> necessárias e a utilização estimada de <i>hardware</i> .

a) 6 – 1 – 4 – 7 – 2 – 3 – 5

b) 4 – 7 – 1 – 2 – 3 – 5 – 6

c) 3 – 5 – 2 – 4 – 7 – 6 – 1

d) 5 – 2 – 6 – 3 – 1 – 4 – 7

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Segundo Sommerville, a estrutura de um plano de teste de *software* é:

Processo de teste: descrição das fases principais do processo de teste.

Rastreabilidade de requisito: os usuários são os mais interessados em que o sistema atenda a seus requisitos e em que os testes sejam planejados de maneira que todos os requisitos sejam individualmente testados.

Itens testados: os produtos do processo de *software* a serem testados devem ser especificados.

Cronograma de testes: um cronograma global de testes e alocação de recursos para esse cronograma estão, obviamente, vinculados ao cronograma mais geral de desenvolvimento de projeto.

Procedimento de registro de testes: não é suficiente realizar simplesmente os testes. Os resultados dos testes devem ser sistematicamente registrados.

Requisitos de *hardware* e *software*: esta seção deve estabelecer as ferramentas de *software* necessárias e a utilização estimada de *hardware*.

Restrições: as restrições que afetam o processo de teste, como falta de pessoal, devem ser antecipadas nesta seção.

Fonte: SOMMERVILLE, I. Engenharia de Software. 9ª ed. São Paulo: Addison Wesley Brasil, 2011.

55) Os requisitos de sistema são descrições dos serviços fornecidos pelo sistema e suas restrições operacionais. Os requisitos organizacionais, os de segurança e os do produto estão classificados como requisitos:

a) funcionais.

b) de usuário.

c) de sistema.

d) não funcionais.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Requisitos organizacionais, segurança e produto são classificados como requisitos não funcionais.

Fonte: SOMMERVILLE, I. Engenharia de Software. 9ª ed. São Paulo: Addison Wesley Brasil, 2011.

**56)** O objetivo da prototipação é permitir que os usuários ganhem experiência direta com a interface. Nessa abordagem, cria-se telas com elementos visuais, como botões e menus, e associa-se um *script* a esses elementos. Assinale a alternativa em que essa abordagem é apresentada.

- a) Prototipação de sistema.
- b) Abordagem dirigida a *scripts*.**
- c) Prototipação baseada na *Internet*.
- d) Linguagens de programação visual.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Abordagem dirigida a *scripts*: Se é necessário simplesmente explorar ideias com os usuários, pode-se usar uma abordagem dirigida a *scripts*, conforme é encontrado no *Macromedia Director*. Nessa abordagem, cria-se telas com elementos visuais, como botões e menus, e associa-se um *script* a esses elementos. Quando o usuário interage com essas telas, o *script* é executado e a próxima tela é apresentada, mostrando nessas telas os resultados de suas ações. Não existe lógica de aplicação envolvida.

Fonte: SOMMERVILLE, I. Engenharia de Software. 9ª ed. São Paulo: Addison Wesley Brasil, 2011.

**57)** Em gerenciamento de projeto, a fase que acontece paralelamente ao planejamento operacional e à execução do projeto é a de

- a) iniciação.
- b) execução.
- c) planejamento.
- d) monitoramento e controle.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

É correto afirmar que:

Fase de execução: é a fase que materializa tudo aquilo que foi planejado anteriormente.

Fase de planejamento: é a fase responsável por detalhar tudo aquilo que será realizado pelo projeto, incluindo cronogramas, interdependências entre atividades, alocação dos recursos envolvidos, análises de custos etc., para que no final dessa fase, ele esteja suficientemente detalhado para ser executado sem dificuldades e imprevistos.

Fase de iniciação: é a fase inicial do projeto, quando uma determinada necessidade é identificada e transformada em um problema estruturado a ser resolvido por ele.

Fonte: VARGAS, Ricardo. Gerenciamento de Projetos. 7ª ed. Rio de Janeiro: Brasport, 2009.

**58)** O interrelacionamento entre as atividades do projeto compõe um todo organizado, denominado “diagrama de rede”, ou vulgarmente conhecido como rede PERT (*Program Evaluation and Review Technique*). Sobre o “diagrama de rede”, é **incorreto** afirmar que

- a) é de fácil manipulação.**
- b) é de simples entendimento.
- c) apresenta relatórios muito extensos.
- d) não mostra uma relação visual entre as durações das atividades.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Sobre o diagrama de rede há vantagens e desvantagens.

Vantagens: simples entendimento

Desvantagens: apresenta relatórios muito extensos; não mostra uma relação visual entre as durações das atividades; É de difícil manipulação.

Fonte: VARGAS, Ricardo. Gerenciamento de Projetos. 7ª ed. Rio de Janeiro: Brasport, 2009.

**59)** O PMBOK subdivide o gerenciamento de escopo em cinco processos, a saber: planejamento do escopo, definição do escopo, criar EAP, verificação do escopo e controle do escopo. Assinale a alternativa que apresenta as fases do projeto em que esses processos estão distribuídos.

- a) execução e controle.
- b) iniciação e execução.
- c) planejamento e controle.**
- d) planejamento e execução.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Os 5 processos de gerenciamento de escopo estão distribuídos ao longo das seguintes fases do projeto:

Planejamento: planejamento do escopo; definição do escopo; criar EAP;

Controle: verificação do escopo; controle do escopo.

Fonte: VARGAS, Ricardo. Gerenciamento de Projetos. 7ª ed. Rio de Janeiro: Brasport, 2009.

**60)** Sobre SQL, analise as afirmativas e assinale a **incorreta**.

- a) Um tipo booleano tem os valores tradicionais *TRUE* (verdadeiro) ou *FALSE* (falso).
- b) O formato de *DATE*, *TIME* e *TIMESTAMP* pode ser considerado um tipo especial de cadeia.
- c) O tipo de dados *DATE* possui dez posições e seus componentes são *DAY* (dia), *MONTH* (mês) e *YEAR* (ano) na forma DD-MM-YYYY.
- d) Um tipo de dado *TIMESTAMP* inclui os campos *DATE* e *TIME*, mais um mínimo de 8 posições para frações decimais de segundos e um qualificador obrigatório *WITH TIME ZONE*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Um tipo de dado timestamp (*TIMESTAMP*) inclui os campos *DATE* e *TIME*, mais um mínimo de 6 posições para frações decimais de segundos e um qualificador opcional *WITH TIME ZONE*.

Fonte: ELMASRI, R.; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6ª ed. São Paulo: Pearson, 2011.