

## CONHECIMENTOS ESPECIALIZADOS

### Texto para responder às questões 31 e 32.

A máscara padrão de um endereço de rede classe B é 255.255.0.0 e da classe C é 255.255.255.0. Com o esgotamento dos endereços IPv4, um artifício que pode ser utilizado é a criação de sub-redes, ou seja, subdividir uma rede, em redes menores. Para isso pega-se “bit(s) emprestado(s)” do *host* no endereçamento e a máscara de rede é criada, de acordo com a quantidade de *bits* utilizada.

- 31) Utilizando-se 2 *bits* do *host* para criar sub-redes, qual a máscara de rede correta a ser utilizada em um endereço de rede classe C?
- a) 255.255.255.254
  - b) 255.255.255.252
  - c) 255.255.255.248
  - d) 255.255.255.140

#### JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

A solução para o esgotamento é permitir que uma rede seja subdividida em várias redes para uso interno. Como são 2 *bits* de *host* utilizados (e  $2^2 = 4$ ) e só se pode ir a 255 (que, na verdade, são 256 *bits*, pois vai de 0 a 255), então  $256 - 4 = 252$ . A máscara de rede a ser utilizada neste caso é: 255.255.255.252.

Fonte: TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 278-279.

- 32) Utilizando-se 13 *bits* do *host* para criar sub-redes, qual a máscara de rede correta a ser utilizada em um endereço de rede classe B?
- a) 255.255.224.0
  - b) 255.255.192.0
  - c) 255.255.248.0
  - d) 255.255.252.0

#### JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

A solução para o esgotamento é permitir que uma rede seja subdividida em várias redes para uso interno. Num endereço de rede classe B, o cálculo é feito de acordo com o número de *bits* de *hosts* utilizados também. Com o uso de 8 *bits* a máscara fica: 255.255.255.0. Como está sendo emprestado 6 *bits* do segundo octeto ( $2^6 = 64$  ( $256 - 64 = 192$ )), a máscara fica: 255.255.224.0.

Fonte: TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 278-279.

- 33) No cabeçalho do IPv4 existe um campo chamado Opções que foi projetado para permitir que versões posteriores do protocolo pudessem incluir informações inexistentes no projeto original, possibilitando a experimentação de novas ideias e evitando a locação de *bits* de cabeçalho para informações raramente necessárias. Nesse campo existe uma opção de nome *Loose source routing*. Assinale a alternativa que apresenta a descrição correta dessa opção.
- a) Mostra o caminho completo a ser seguido.
  - b) Faz com que cada roteador anexe seu endereço IP.
  - c) Apresenta uma lista de roteadores que não devem ser esquecidos.
  - d) Faz com que cada roteador anexe seu endereço e seu registro de tempo (*flag*).

#### JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Originalmente haviam cinco opções definidas nesse campo: *Security*; *Strict source routing*; *Loose source routing*; *Record route*; e, *Timestamp*. A opção *Loose source routing* significa: apresenta uma lista de roteadores que não devem ser esquecidos.

Fonte: TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 276-277.

**34)** O modelo de camadas TCP/IP foi definido pela primeira vez em 1974 e, posteriormente, considerado um padrão na comunidade da *Internet*. Este modelo possui 5 camadas. Assinale a alternativa **incorreta** acerca das camadas do modelo TCP/IP.

- a) Rede.
- b) Física.
- c) Enlace.
- d) **Apresentação.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

As cinco camadas do modelo TCP/IP são: Transporte; Rede; Enlace; e Física.

Fonte: TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 30.

**35)** Os modelos de referência OSI e TCP/IP têm muita coisa em comum. Ambos apresentam como conceito uma pilha de protocolos independentes. Basicamente, as camadas têm as mesmas funcionalidades. Esses dois modelos de referência possuem algumas diferenças, mas não nos protocolos, que basicamente são os mesmos. O modelo OSI tem três conceitos fundamentais. Assinale a alternativa que apresenta corretamente esses três conceitos.

- a) Segurança, Serviços e Interfaces.
- b) **Serviços, Interfaces e Protocolos.**
- c) Segurança, Serviços e Protocolos.
- d) Protocolos, Interfaces e Segurança.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Os três conceitos fundamentais do modelo de referência OSI são: Serviços, Interfaces e Protocolos.

Fonte: TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 30.

**36)** De acordo com *Pressman*, a engenharia de *software* é uma tecnologia em camadas e as abordagens de engenharia devem se apoiar num compromisso organizacional com a qualidade. As alternativas abaixo apresentam as camadas de engenharia de *software*, **exceto** em:

- a) **produto.**
- b) processo.
- c) ferramentas.
- d) foco na qualidade.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

As quatro camadas da engenharia de *software*, segundo Pressman (2006), são: Ferramentas; Métodos; Processo; e Foco na qualidade.

Fonte: PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

**37)** O CMMI representa um metamodelo de processo de dois modos diferentes: como um modelo contínuo; e, como um modelo em estágios. Cada área de processo do metamodelo contínuo é avaliada com base em metas e práticas específicas, sendo classificada segundo os níveis de aplicação. Em relação aos níveis de aplicação no metamodelo contínuo, associe as colunas.

- |            |                               |
|------------|-------------------------------|
| (1) Zero   | ( ) definido.                 |
| (2) Um     | ( ) incompleto.               |
| (3) Dois   | ( ) quantitativamente gerido. |
| (4) Três   | ( ) gerido.                   |
| (5) Quatro | ( ) otimizado.                |
| (6) Cinco  | ( ) realizado.                |

A sequência correta dessa associação é

- a) 4 – 1 – 5 – 3 – 6 – 2
- b) 6 – 4 – 2 – 5 – 3 – 1
- c) 5 – 3 – 1 – 6 – 2 – 4
- d) 3 – 6 – 4 – 2 – 1 – 5

**JUSTIFICATIVA DA ALTERNATIVA CORRETA:** (LETRA A)

Os níveis de aplicação no metamodelo contínuo são os seguintes: Nível 0 – Incompleto; Nível 1– Realizado; Nível 2 – Gerido; Nível 3 – Definido; Nível 4 – Quantitativamente gerido; e, Nível 5 – Otimizado.

**Fonte:** PRESSMAN, Roger S. **Engenharia de software:** uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

**38)** O SMTP (*Simple Mail Transfer Protocol*) é um protocolo da camada de aplicação, responsável pelo envio de *e-mails* na *web*. O CGI.br (Comitê Gestor da *Internet* no Brasil), há pouco tempo, fez uma recomendação a todos para que se mudasse a porta do protocolo SMTP para a porta 587, como medida de segurança, a fim de se evitar a proliferação de *spams*. Quando da criação do SMTP, originalmente, qual era a porta utilizada por este protocolo?

- a) 21.
- b) 22.
- c) 23.
- d) 25.

**JUSTIFICATIVA DA ALTERNATIVA CORRETA:** (LETRA D)

O SMTP estabelece uma conexão TCP com a porta 256 da máquina de destino. Um servidor de *e-mail* que se comunica em SMTP permanece na escuta nessa porta.

**Fonte:** TANENBAUM, Andrew. S.; WETHERALL, David. **Redes de Computadores.** 5. ed. São Paulo: Pearson Education Brasil, 2011. 600p. p. 30.

**39)** De acordo com *Pressman*, existem sete amplas categorias de *software* de computadores e que apresentam desafios contínuos para os engenheiros de *software*. Uma dessas categorias é o “*Software* para linha de produtos”. Assinale a alternativa correta acerca dessa categoria.

- a) Consiste em programas isolados que resolvem uma necessidade específica do negócio.
- b) Reside dentro de um produto ou sistema e é usado para implementar e controlar características e funções para o usuário final e o próprio sistema.
- c) **Projetado para fornecer uma capacidade específica a ser usada por muitos clientes diferentes, pode focalizar um mercado limitado e especial ou dirigir-se ao mercado de consumo de massa.**
- d) Coleção de programas escritos para servir a outros programas. Alguns, como compiladores, editores e utilitários para gestão de arquivos processam estruturas de informações complexas, mas determinadas.

**JUSTIFICATIVA DA ALTERNATIVA CORRETA:** (LETRA C)

As sete categorias de *software* existentes, segundo Pressman (2006), são: *Software* de sistemas; *Software* de aplicação; *Software* científico e de engenharia; *Software* embutido; *Software* para linha de produtos; Aplicações na *web*; e, *Software* para inteligência artificial. O “*software* para linha de produtos” é projetado para fornecer uma capacidade específica a ser usada por muitos clientes diferentes, podendo focalizar um mercado limitado e especial ou dirigir-se ao mercado de consumo de massa.

**Fonte:** PRESSMAN, Roger S. **Engenharia de software:** uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

**40)** Para implementar o modelo de processos, o sistema operacional mantém uma tabela (um arranjo de estruturas) chamada de tabela de processos, com uma entrada para cada processo. É também chamada, por alguns autores, de PCB (*Process Control Blocks* – blocos de controle de processos). Essa entrada contém diversas informações do processo. Assinale a alternativa correta, na qual está apresentada uma dessas informações relativas ao processo.

- a) ID do usuário.
- b) Diretório de trabalho.
- c) **Palavra de estado do programa.**
- d) Ponteiro para informações sobre o segmento de texto.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A alternativa correta é a “C”, visto que palavra de estado do programa refere-se a processo no PCB. As demais alternativas encontram-se incorretas, respectivamente, pois:

- ID do usuário refere-se ao gerenciamento de arquivo;
- Diretório de trabalho refere-se ao gerenciamento de arquivo; e,
- Ponteiro para informações sobre o segmento de texto refere-se a Gerenciamento de memória.

Fonte: TANEMBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

**41)** Os sistemas operacionais mantêm uma lista de processos e seus estados, antes de utilizar a CPU. O escalonamento é utilizado para determinar qual processo vai utilizar a CPU e em que instante. O responsável por fazer essa “escolha” é o escalonador, através de um algoritmo de escalonamento. Um desses algoritmos de escalonamento atribui um intervalo de tempo, em que o processo pode permanecer na CPU, tempo esse que recebe o nome de *quantum*. Assinale a alternativa correta acerca do nome desse algoritmo.

- a) **Round-Robin – RR.**
- b) Tarefa mais curta primeiro.
- c) Com compartilhamento imparcial.
- d) Menor tempo de execução primeiro.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Escalonamento round-robin – RR: a cada processo é atribuído um intervalo de tempo, chamado *quantum*, durante o qual o processo pode ser executado. É um dos algoritmos mais utilizados.

Fonte: TANENBAUM, Andrew S.; WOODHULL, A. S. **Sistemas Operacionais: projeto e implementação**. 3. ed. Porto Alegre: Bookman, 2008. 992p. p. 111.

42) O CMMI define cada área de processo em termos de “metas específicas” e “práticas específicas” necessárias para atingir tais metas. Segundo o CMMI, planejamento de projeto é uma das oito áreas definidas para a categoria “gestão de projetos”. Associe as colunas, relacionando as metas e práticas específicas aos respectivos planejamentos de projeto.

- |                        |   |
|------------------------|---|
| (1) Meta específica    | ( ) define o ciclo de vida do projeto.      |
| (2) Prática específica | ( ) estabelece o orçamento e o cronograma.  |
|                        | ( ) estabelece estimativas.                 |
|                        | ( ) estabelece o plano de projeto.          |
|                        | ( ) desenvolve um plano de projeto.         |
|                        | ( ) obtém comprometimento com o plano.      |
|                        | ( ) planeja o envolvimento dos interessados |

A sequência correta dessa associação é

- a) 1 – 2 – 1 – 1 – 2 – 2 – 1
- b) 2 – 2 – 1 – 2 – 1 – 1 – 2
- c) 1 – 1 – 2 – 1 – 2 – 2 – 1
- d) 2 – 1 – 2 – 2 – 1 – 1 – 2

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

As metas específicas são: estabelecer estimativas; desenvolver um plano de projeto; e, obter comprometimento com o plano.

As práticas específicas são: definir o ciclo de vida do projeto; estabelecer o orçamento e o cronograma; estabelecer o plano de projeto; e, planejar o envolvimento dos interessados.

Fonte: PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

43) Quatro eventos podem fazer com que os processos sejam criados, assim também como quatro eventos podem ser responsáveis pelo término de um processo. Entre a criação e o término, ocorre o processo para pôr estados, que definem a condição de cada processo em relação ao sistema. É **incorreto** afirmar que o processo sai do estado de

- a) pronto para execução.
- b) pronto para bloqueado.
- c) execução para bloqueado.
- d) bloqueado para execução.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Um processo não pode sair do estado bloqueado para o de execução. Primeiramente, deve-se ir para a fila de prontos, para que o escalonador possa alocá-lo à CPU que irá utilizar.

Fonte: TANENBAUM, Andrew S.; WOODHULL, A. S. **Sistemas operacionais: projeto e implementação**. 3. ed. Porto Alegre: Bookman, 2008. 992p. p. 74.

44) Analise as alternativas sobre a linguagem de programação Java e, em seguida, assinale a alternativa **incorreta**.

- a) Uma classe pode realizar uma tarefa e retornar um resultado.
- b) Uma chamada de método instrui um método a executar sua tarefa.
- c) Uma classe pode ser utilizada para criar uma instância da classe chamada objeto.
- d) Em geral, não se pode chamar um método de outra classe até criar um objeto dessa classe.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

A alternativa **incorreta** é a “A”, visto que um método pode realizar uma tarefa e retornar um resultado, e não uma classe.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 147-148.

45) Informe se é verdadeiro (V) ou falso (F) o que se afirma abaixo. A seguir, assinale a alternativa que apresenta a sequência correta.

- ( ) A instrução *for* pode ser utilizada para implementar qualquer *loop* por controlador.
  - ( ) Em geral, as instruções *while* são utilizadas para repetição controlada por contador e as instruções *for* são utilizadas para repetição controlada por sentinela.
  - ( ) A instrução *switch* consiste em um bloco que contém uma sequência de rótulos *case* e um caso *default* opcional.
  - ( ) Cada *case* pode ter múltiplas instruções, e estas devem ser colocadas entre colchetes.
  - ( ) Listar casos consecutivamente sem instruções entre eles permite aos casos executar o mesmo conjunto de instruções.
  - ( ) A instrução *while* especifica os detalhes da repetição controlada por contador em uma única linha de código.
- a) V – F – F – V – F – V  
b) F – F – V – F – V – F  
c) V – V – F – V – F – V  
d) F – V – V – F – V – F

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Analisando as alternativas, respectivamente, tem-se:

- Falsa – a instrução *while* pode ser utilizada para implementar qualquer *loop* por controlador;
- Falsa – em geral, as instruções *for* são utilizadas para repetição controlada por contador e as instruções *while* são utilizadas para repetição controlada por sentinela;
- Verdadeira – a instrução *switch* consiste em um bloco que contém uma sequência de rótulos *case* e um caso *default* opcional;
- Falsa – cada *case* pode ter múltiplas instruções, e estas não devem ser colocadas entre colchetes.
- Verdadeira – listar casos, consecutivamente, sem instruções entre eles, permite aos casos executar o mesmo conjunto de instruções; e,
- Falsa – a instrução *for* especifica os detalhes da repetição controlada por contador em uma única linha de código.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 147-148.

46) Os Sistemas Operacionais evoluíram muito nos últimos anos e a tendência é a continuidade dessa evolução. Estão divididos por gerações ao longo dos anos. A segunda geração, tradicionalmente, é reconhecida como sendo de 1955 a 1965. Assinale a alternativa correta acerca dessa geração.

- a) CIs e Multiprogramação.
- b) Computadores pessoais.
- c) Transistores e sistemas em lote.
- d) Válvulas e painéis de conectores.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A segunda geração, caracterizada pelos transistores e sistemas em lote, é conhecida como sendo de 1955 a 1965.

Fonte: TANENBAUM, Andrew. S.; WOODHULL, A. S. **Sistemas Operacionais: projeto e implementação**. 3.ed. Porto Alegre: Bookman, 2008. 992p. p. 26.

47) Preencha as lacunas abaixo e, em seguida, assinale a alternativa correta.

Se uma exceção ocorrer em um bloco \_\_\_\_\_, o bloco \_\_\_\_\_ terminará imediatamente e o controle do programa irá passar para o primeiro bloco \_\_\_\_\_ com um tipo de parâmetro que corresponde ao tipo da exceção lançada.

- a) *try / try / catch*
- b) *try / catch / finally*
- c) *catch / finally / try*
- d) *finally / finally / try*

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Se uma exceção ocorrer em um bloco *try*, o bloco *try* terminará imediatamente e o controle do programa irá passar para o primeiro bloco *catch* com um tipo de parâmetro que corresponde ao tipo da exceção lançada.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**: Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 356.

**Gabarito Comentado – EAOAP 2015 – Análise de Sistemas – Versão A**

48) Observe o programa abaixo:

```
package somainteiros;
public class SomalNteiros {
    public static void main(String[] args)
    {
        int total = 0;
        for (int number = 2; number <= 20; number += 2)
            total += number;
        System.out.printf ("SomalNteiros is %d\n", total);
    }
}
```

Assinale a alternativa abaixo onde está apresentado o resultado da operação acima

- a) 20.
- b) 40.
- c) 80.
- d) 110.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Esse programa está somando todos os inteiros, de 2 a 2, até 20 (inclusive):  $2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20$ , utilizando a instrução *for*.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 121-124.

49) A tecnologia *JavaServer Pages* (JSP) é uma extensão da tecnologia de *servlet*. Cada JSP é um documento convertido pelo contêiner JSP em um *servlet*. Há quatro componentes-chave para JSP. Assinale a alternativa correta que apresenta esses quatro componentes.

- a) Cabeçalhos, ações, método *string* e método *service*.
- b) Método *service*, ações, elementos de *script* e aplicativos.
- c) Diretivas, ações, elementos de *script* e bibliotecas de *tags*.
- d) Diretivas, método *service*, aplicativos e bibliotecas de *tags*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Os quatro componentes-chaves para JSPs são: diretivas; ações; elementos de *script*, e, bibliotecas de *tags*.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 951.

50) Listas, pilhas e filas são estruturas de dados lineares, isto é, sequências. Uma árvore é considerada uma estrutura de dados

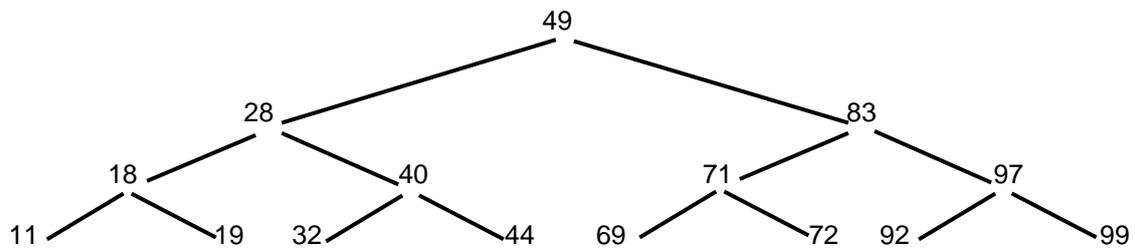
- a) linear, sem propriedades especiais.
- b) linear, com propriedades especiais.
- c) bidimensional, não linear, sem propriedades especiais.
- d) bidimensional, não linear, com propriedades especiais.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Como listas, pilhas e filas são estruturas de dados lineares, ou seja, sequências, uma árvore é considerada uma estrutura de dados, não linear, com propriedades especiais.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 709.

51) Observe a seguinte estrutura de uma árvore.



Assinale a alternativa correta que apresenta o caminho percorrido em pré-ordem na árvore acima.

- a) 11 – 19 – 18 – 32 – 44 – 40 – 69 – 72 – 71 – 92 – 99 – 97 – 28 – 83 – 49
- b) 11 – 19 – 32 – 44 – 69 – 72 – 92 – 99 – 18 – 40 – 71 – 97 – 28 – 83 – 49
- c) 49 – 28 – 18 – 11 – 19 – 40 – 32 – 44 – 83 – 71 – 69 – 72 – 97 – 92 – 99
- d) 49 – 28 – 83 – 18 – 40 – 71 – 97 – 11 – 19 – 32 – 44 – 69 – 72 – 92 – 99

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Em um percurso na pré-ordem, o valor de cada nó é processado quando o nó é percorrido. Então, os valores na subárvore esquerda são processados e, em seguida, os valores da subárvore direita.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 709-710.

52) Centenas de métricas têm sido propostas para *software* de computador, mas nem todas fornecem apoio prático ao engenheiro de *software*. Algumas exigem medições muito complexas, outras são tão restritas que poucos profissionais tentam entendê-las. Um conjunto de atributos deve ser abrangido para métricas de *software* efetivas. Associe as colunas, relacionando as medidas para as métricas de *software* às respectivas métricas derivadas.

- |   |  |
|---|--|
| (1) Simples e computáveis                                 | ( ) a métrica deve produzir sempre resultados que não sejam ambíguos.  |
| (2) Empíricas e intuitivamente persuasivas                | ( ) métricas devem ser baseadas no modelo de análise, modelo de projeto ou na estrutura do programa propriamente dita.       |
| (3) Consistentes e objetivas                              | ( ) deve ser relativamente fácil aprender como derivar a métrica e o seu cálculo não deve exigir esforço ou tempo exagerado. |
| (4) Consistentes no uso de unidades e dimensões           | ( ) isto é, a métrica deve levar a um produto final da mais alta qualidade.  |
| (5) Independentes da linguagem de programação             | ( ) a métrica deve satisfazer as noções intuitivas do engenheiro sobre o atributo do produto que está sendo considerado.     |
| (6) Mecanismo efetivo por realimentação de alta qualidade | ( ) o cálculo matemático da métrica deve usar medidas que não levam a combinações de unidades bizarras.                      |

A sequência correta dessa associação é

- a) 5 – 3 – 2 – 4 – 6 – 1
- b) 6 – 2 – 4 – 1 – 3 – 5
- c) 3 – 5 – 1 – 6 – 2 – 4
- d) 5 – 1 – 3 – 2 – 4 – 6

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Analisando as medidas para as métricas de *software*, tem-se:

- Simples e computáveis – deve ser relativamente fácil aprender como derivar a métrica e seu cálculo não deve exigir esforço ou tempo exagerado;
- Empíricas e intuitivamente persuasivas – a métrica deve satisfazer as noções intuitivas do engenheiro sobre o atributo do produto que está sendo considerado;
- Consistentes e objetivas – a métrica deve produzir sempre resultados que não sejam ambíguos;
- Consistentes no uso de unidades e dimensões – o cálculo matemático da métrica deve usar medidas que não levam a combinações de unidades bizarras;
- Independentes da linguagem de programação – métricas devem ser baseadas no modelo de análise, modelo de projeto ou na estrutura do programa propriamente dita; e,
- Mecanismo efetivo par realimentação de alta qualidade – métricas devem ser baseadas no modelo de análise, modelo de projeto ou na estrutura do programa propriamente dita.

Fonte: PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

53) Observe o código a seguir.

```
public class Calculate1 {
    public static void main(String[] args) {
        int sum=0;
        for (int i=1; i<=10; ++i){
            sum +=i;
        }
        System.out.printf("A soma é: %d\n", sum);
    }
}
```

Qual a saída que o código acima vai gerar?

- a) 45.
- b) 46.
- c) 54.
- d) 55.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

O programa está contando todos os números inteiros, de 1 a 10 (inclusive). Portanto, o resultado é 55.

Fonte: DEITEL, H. M.; DEITEL, P. J. **Java**. Como programar. 8. ed. São Paulo: Prentice Hall Brasil, 2010. p. 121-124.

54) Como são necessárias quatro condições para que ocorra um *deadlock*, também são conhecidas quatro estratégias usadas para lidar com os *deadlocks*. Todas as alternativas abaixo apresentam essas estratégias, **exceto**:

- a) detecção e recuperação.
- b) ignorar por completo o problema.
- c) **anulação estática por meio de uma alocação dos recursos.**
- d) anulação dinâmica por meio de uma alocação cuidadosa de recursos.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A estratégia correta para lidar com um *deadlock* (impasse) é a anulação dinâmica por meio de uma alocação cuidadosa de recursos.

Fonte: TANEMBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

55) Observe a seguinte expressão em álgebra relacional:

$((P \text{ WHERE } COR = \text{'Vermelho'}) \text{ JOIN FP } \{F\# \} \text{ JOIN F}) \{FNOME\}$

Assinale a alternativa correta que apresenta a saída gerada por essa expressão.

- a) A expressão não vai produzir resultado, pois há um erro.
- b) Obter nomes de fornecedores correspondentes a fornecedores que fornecem só peças em vermelho.
- c) **Obter objetos de fornecedores correspondentes a outros que fornecem pelo menos uma peça vermelha.**
- d) Obter nomes de fornecedores correspondentes a outros que fornecem peças em vermelho e que começam com a letra F.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

A expressão apresentada irá produzir o seguinte resultado: obter objetos de fornecedores correspondentes a outros que fornecem pelo menos uma peça vermelha.

Fonte: ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Education Brasil, 2011. 808p.

**56)** Muitas das atividades de análise de risco têm como única meta ajudar a equipe de projeto a desenvolver uma estratégia para lidar com o risco. Uma estratégia efetiva deve considerar três pontos. Todas as alternativas apresentam esses pontos, **exceto**:

- a) evitar o risco.
- b) monitorar o risco.
- c) **estimativa de risco.**
- d) gerenciar o risco e planejar para a contingência.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Os três pontos a serem considerados para uma estratégia efetiva são: evitar o risco; monitorar o risco; e, gerenciar o risco e planejar para a contingência.

Fonte: PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

**57)** Um método para a identificação de riscos é a criação de uma *checklist* de itens de risco. Essa *checklist* pode ser usada para identificação de risco e concentra-se em algum subconjunto de riscos conhecidos e previsíveis em algumas subcategorias genéricas. Uma dessas categorias é o ambiente de desenvolvimento, cujos riscos estão associados com

- a) restrições impostas pela gerência ou pelo mercado.
- b) **a disponibilidade e a qualidade das ferramentas a serem usadas para construir o produto.**
- c) o grau em que o processo de *software* foi definido e é seguido pela organização de desenvolvimento.
- d) a complexidade do sistema a ser construído e com a “novidade” da tecnologia que é incorporada ao sistema.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Sobre o ambiente de desenvolvimento é correto afirmar que os riscos estão associados à disponibilidade e à qualidade das ferramentas a serem usadas para construir o produto.

Fonte: PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

**58)** Alguns multiprocessadores apresentam propriedades adicionais, como a possibilidade de que cada palavra de memória possa ser lida tão rapidamente quanto qualquer outra palavra de memória. Essas máquinas são chamadas de multiprocessadores UMA (*Uniform Memory Access* – acesso uniforme à memória). Em contraste, os multiprocessadores NUMA (*Non-Uniform Memory Access* – acesso não uniforme à memória) não apresentam essa propriedade. Quando o tempo de acesso à memória é explícito (porque não existe nenhuma cache), o sistema recebe um nome. Assinale a alternativa correta que apresenta esse nome.

- a) NC-UMA.
- b) CC-UMA.
- c) **NC-NUMA.**
- d) CC-NUMA.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

O correto é NC-NUMA – significa *no cache* NUMA – NUMA sem cache, que é a resposta correta.

Fonte: TANEMBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

**59)** Alguns fatores afetam a qualidade do *software*. Esses fatores concentram-se em três aspectos importantes de um produto de *software*: características operacionais, habilidade de passar por modificações e adaptabilidade a novos ambientes. Um desses fatores é chamado de correção. Sobre a correção, é correto afirmar que

- a) é o esforço necessário para modificar um programa operacional.
- b) é o esforço necessário para localizar e consertar um erro em um programa.
- c) o acesso ao *software* ou a dados por pessoas não autorizadas pode ser controlado.
- d) **o programa satisfaz sua especificação e preenche os objetivos da missão do cliente.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Sobre o fator correção é correto afirmar que ocorre quando um programa satisfaz sua especificação e preenche os objetivos da missão do cliente.

Fonte: PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: Mc Graw Hill – Bookman, 2011.

60) Segundo *Tanenbaum*, Impasse (*deadlock*) pode ser formalmente definido como: “Um conjunto de processos estará em situação de impasse se todo processo pertencente ao conjunto estiver esperando por um evento que somente outro processo desse mesmo conjunto poderá acontecer”. Associe as colunas, relacionando as quatro condições existentes para que ocorra um impasse (*deadlock*) às respectivas definições.

- |                     |   |
|---------------------|---|
| (1) Exclusão mútua  | ( ) processos que, em um determinado instante, retêm recursos concedidos anteriormente podem requisitar novos recursos.   |
| (2) Posse e espera  | ( ) deve existir um encadeamento de dois ou mais processos; cada um deles encontra-se à espera de um recurso que está sendo usado pelo membro seguinte dessa cadeia.      |
| (3) Não preempção   | ( ) em um determinado instante, cada recurso estará em uma de duas situações ou associado a um único processo ou, ainda disponível.                                       |
| (4) Espera circular | ( ) recursos concedidos previamente a um processo não podem ser forçosamente tomados desse processo – eles devem ser explicitamente liberados pelo processo que os retêm. |

A sequência correta dessa associação é

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

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

As condições para *deadlock*, ou impasse, são as seguintes:

- Exclusão mútua – em ou um determinado instante, cada recurso está em uma de duas situações: ou associado a um único processo ou disponível;
- Posse e espera – processos, que em um determinado instante, retêm recursos concedidos anteriormente podem requisitar novos recursos;
- Não preempção – recursos concedidos previamente a um processo não podem ser forçosamente tomados desse processo – eles devem ser explicitamente liberado pelo processo que os retêm; e,
- Espera circular – deve existir um encadeamento de dois ou mais processos; cada um deles encontra-se à espera de um recurso que está sendo usado pelo membro seguinte dessa cadeia.

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