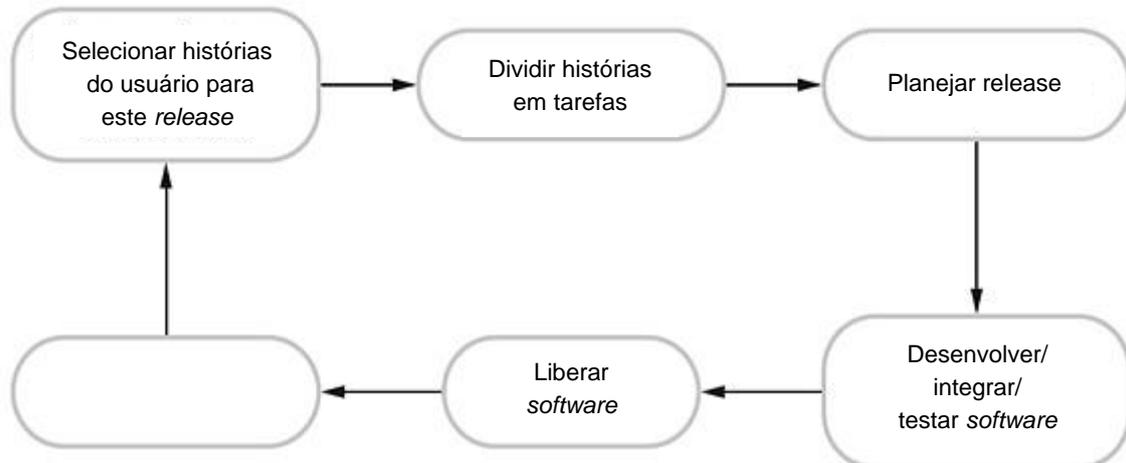


## CONHECIMENTOS ESPECIALIZADOS

31) De acordo com *Sommerville* (2011), o ciclo de um *release* em *Extreme Programming* é composto por diversas etapas, como é mostrado na figura a seguir:



De acordo com a figura anterior, qual é a fase que falta para completar o ciclo?

- a) Avaliar sistema.
- b) Revisar *software*.
- c) Analisar o produto.
- d) Envolvimento do cliente.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

O ciclo de um *release* em *Extreme Programming* é composto por: selecionar histórias de usuário para esta versão; dividir histórias em tarefas; planejar versão; desenvolver/integrar/testar *software*; liberar *software*; e, avaliar sistema.

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

32) De acordo com *Sommerville* (2011), CMMI é “um modelo de maturidade e de capacidade integrado que inclui avaliação das capacidades de engenharia de *software* e de sistemas”. Esse modelo possui duas instâncias: estágios e contínuo. Na primeira, estágios, o modelo é expresso em termos de níveis de maturidade. Em contínuo, uma classificação da maturidade do processo é fornecida em menor granularidade. O modelo CMMI é composto por: áreas de processo, metas e práticas. Acerca das metas, relacionando-as com a área do processo, podem ser classificadas em específicas e genéricas. Assinale a alternativa correta que apresenta uma meta genérica.

- a) O processo é institucionalizado como um processo definido.
- b) O desempenho real e o progresso do projeto são monitorados contra o plano de projeto.
- c) Os requisitos são analisados e validados, e uma definição da funcionalidade requerida é desenvolvida.
- d) Ações corretivas são gerenciadas até a conclusão, quando o desempenho ou os resultados do projeto se desviam significativamente do plano.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

São consideradas metas específicas: ações corretivas são gerenciadas até a conclusão, quando o desempenho ou os resultados do projeto se desviam significativamente do plano; o desempenho real e o progresso do projeto são monitorados contra o plano de projeto; os requisitos são analisados e validados, e uma definição da funcionalidade requerida é desenvolvida; causas-raiz de defeitos e outros problemas são sistematicamente determinados.

Meta genérica: O processo é institucionalizado como um processo definido.

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

**33)** A abordagem de implementação por estágios, do CMMI, é considerada uma evolução do CMM, pois também é baseada em cinco níveis de maturidade: Inicial (1), Gerenciado (2), Definido (3), Gerenciado Quantitativamente (4) e Otimizado (5). Sobre esses níveis de maturidade, assinale a alternativa correta, que apresenta as áreas de processo do nível 3, definido.

- a) Gestão de Requisitos (REQM) e Medição e Análise (MA).
- b) Planejamento do Projeto (PP) e Gestão da Configuração (CM).
- c) **Gestão Integrada do Projeto (IPM) e Análise de Decisão e Resolução (DAR).**
- d) Desempenho do Processo Organizacional (OPP) e Gestão Quantitativa do Projeto (QPM).

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Gestão de Requisitos (REQM); Medição e Análise (MA); Planejamento do Projeto (PP); e, Gestão da Configuração (CM) pertencem ao Nível 2 – Gerenciado.

Desempenho do Processo Organizacional (OPP); e, Gestão Quantitativa do Projeto (QPM) pertencem ao nível 5 – Otimizado.

Nível 3 – Definido são: Gestão Integrada do Projeto (IPM) e Análise de Decisão e Resolução (DAR).

Fonte: FERNANDES, A. A. **Implantando a Governança de TI: da estratégia à gestão dos processos e serviços**. 3. ed. Brasport: Rio de Janeiro, 2012.

**34)** A Norma ISO/IEC 27002 é estruturada em diversas seções, como por exemplo: Política de Segurança da Informação, Organizando a Segurança da Informação, Gestão de Ativos, Segurança em Recursos Humanos, Gerenciamento das Operações e Comunicação, Controle de Acesso, Conformidade, entre outros. No total são 11 seções, de acordo com Fernandes e Abreu (2012). Assinale a alternativa **incorreta** acerca da seção Organizando a Segurança da Informação.

- a) Acordos de confidencialidade.
- b) **Procedimentos para tratamento de informação.**
- c) Identificando os riscos relacionados com partes externas.
- d) Análise crítica independente de segurança da informação.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Procedimentos para tratamento de informação pertencem à seção: gerenciamento das operações e comunicações.

Fonte: FERNANDES, A. A. **Implantando a Governança de TI: da estratégia à gestão dos processos e serviços**. 3. ed. Brasport: Rio de Janeiro, 2012.

**35)** A Norma ISO/IEC 27001 adota o ciclo PDCA (*Plan-Do-Check-Act* – Planejar, Executar, Verificar e Agir) e originariamente foi preparada para prover um modelo para monitorar, rever, operar, implantar, estabelecer e melhorar um ISMS (*Information Security Management System*) ou Sistema de Gestão da Segurança da Informação. A norma é dividida em cinco grandes seções, levando-se em conta termos operacionais. Uma dessas seções é o sistema de gestão da segurança da informação, que está subdividida em algumas subáreas. Uma dessas subáreas é a monitoração e revisão do ISMS, que possui várias seções. Qual é uma das ações dessa subárea?

- a) **Conduzir auditorias internas do ISMS em intervalos planejados.**
- b) Definir a abordagem para a avaliação de riscos da organização.
- c) Definir como medir a eficácia dos controles ou grupos de controle.
- d) Prover recursos suficientes para implementar, operar, monitorar, rever, manter e melhorar o ISMS.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Sobre a Monitoração e revisão do ISMS, a ação correspondente é: conduzir auditorias do ISMS em intervalos planejados.

Fonte: FERNANDES, A. A. **Implantando a Governança de TI: da estratégia à gestão dos processos e serviços**. 3. ed. Brasport: Rio de Janeiro, 2012.

**36)** O COBIT (*Control Objectives for Information and related Technology*), criado em 1994 pela ISACF, é um *framework* que fornece boas práticas para a governança de TI, através de um modelo de domínios e processos. A sua versão 4.1 é pautada em algumas áreas-foco da Governança de TI, como Alinhamento Estratégico, Agregação de Valor, Gerenciamento de Riscos, Gerenciamento de Recursos e Medição de Desempenho. Esta versão conta com quatro domínios e 34 processos. Analise a seguinte afirmativa: “garantia de ligação entre os planos do negócio e de TI, manutenção e validação da proposição de valor da TI e alinhamento das operações da empresa com as de TI.” Essa área-foco encontra-se em:

- a) Agregação de valor.
- b) Alinhamento estratégico.**
- c) Medição de desempenho.
- d) Gerenciamento de recursos.

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

Alinhamento estratégico: garantia de ligação entre os planos do negócio e de TI, manutenção e validação da proposição de valor da TI e alinhamento das operações da empresa com as de TI.

**Fonte:** FERNANDES, A. A. **Implantando a Governança de TI: da estratégia à gestão dos processos e serviços**. 3. ed. Brasport: Rio de Janeiro, 2012.

**37)** O COBIT 4.1 é um *framework*, composto por quatro domínios e 34 processos. Acerca desses domínios e seus respectivos processos, relacione as colunas e depois marque a sequência correta nas alternativas abaixo. Alguns números poderão ser utilizados mais de uma vez e outros poderão não ser usados.

- |                                |                                       |
|--------------------------------|---------------------------------------|
| (1) Planejamento e organização | ( ) fornecer governança para a TI.    |
| (2) Aquisição e implementação  | ( ) gerenciar o investimento em TI.   |
| (3) Entrega e suporte          | ( ) gerenciar dados.                  |
| (4) Monitoração e avaliação    | ( ) identificar e alocar custos.      |
|                                | ( ) gerenciar mudanças.               |
|                                | ( ) avaliar e gerenciar riscos de TI. |
|                                | ( ) viabilizar operação e utilização. |

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

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

(1) Planejamento e organização	( 4 ) fornecer governança para a TI.
(2) Aquisição e implementação	( 1 ) gerenciar o investimento em TI.
(3) Entrega e suporte	( 3 ) gerenciar dados.
(4) Monitoração e avaliação	( 3 ) identificar e alocar custos.
	( 2 ) gerenciar mudanças.
	( 1 ) avaliar e gerenciar riscos de TI.
	( 2 ) viabilizar operação e utilização.

**Fonte:** FERNANDES, A. A. **Implantando a Governança de TI: da estratégia à gestão dos processos e serviços**. 3. ed. Brasport: Rio de Janeiro, 2012.

**38)** Os processos do COBIT 4.1 estão ligados diretamente com a matriz de objetivos de TI (Tecnologia de Informação). Cada processo, de todos os domínios, tem seus respectivos objetivos nessa matriz. Um desses processos, o PO6, Comunicar as Diretrizes e Expectativas da Diretoria, pertence ao domínio Planejar e Organizar (Planejamento e Organização). Um dos objetivos desse processo, de acordo com essa matriz, é assegurar a

- a) satisfação mútua no relacionamento com terceiros.
- b) conformidade de TI com leis, regulamentos e contratos.
- c) satisfação dos usuários finais com a oferta e níveis de serviços.
- d) transparência e entendimento dos custos, benefícios, estratégias, políticas e níveis de serviços de TI.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

O processo PO6, Comunicar as Diretrizes e Expectativas da Diretoria tem como um dos seus objetivos: assegurar a transparência e o entendimento dos custos, benefícios, estratégias, políticas e níveis de serviços de TI.

Fonte: **COBIT 4.1 em Português**. Acesso em: <<http://www.isaca.org/Knowledge-Center/cobit/Documents/cobit41-portuguese.pdf>>.

**39)** A ITIL (*Information Technology Infrastructure Library* – Biblioteca de Infraestrutura de Tecnologia da Informação), em sua versão 3, é pautada no ciclo de vida do serviço. Seu modelo está estruturado em cinco publicações, cada uma tem os seus respectivos processos e cobre uma etapa do ciclo de vida do serviço. A estratégia de serviço é uma das publicações da ITIL e transforma o Gerenciamento de Serviço em Ativos Estratégicos, atendendo dessa forma aos objetivos estratégicos da empresa. Essa estratégia está baseada em quatro Ps, que são, respectivamente:

- a) perspectiva, posição, plano e padrões.
- b) posição, processos, padrões e pessoas.
- c) produtos, perspectiva, parceiros e planos.
- d) pessoas, parceiros, processos e produtos.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Os quatro Ps da Estratégia de Serviço são: perspectiva, posição, plano e padrões.

Fonte: FREITAS, M. A. dos S. **Fundamentos do Gerenciamento de Serviços de TI: Preparatório para a certificação ITIL® Foundation**. 2. ed. Rio de Janeiro: Brasport, 2011.

**40)** A publicação Operação de Serviço, da ITIL V3, tem como principal objetivo a orientação sobre como alcançar a eficiência e eficácia na entrega e também no suporte dos serviços. Isso é feito para garantir o valor esperado pelo cliente, atendendo dessa forma aos objetivos estratégicos da empresa. A Operação de Serviço está estruturada em Processos e Funções. As alternativas apresentam os Processos da Operação de Serviço, **exceto** o que se apresenta em gerenciamento de

- a) Acesso.
- b) Eventos.
- c) Requisição.
- d) Operações de TI.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Os Processos do ciclo Operação de Serviço são: Gerenciamento de Eventos; Gerenciamento de Incidentes; Cumprimento de Requisição; Gerenciamento de Problemas; e, Gerenciamento de Acesso.

Fonte: FREITAS, M. A. dos S. **Fundamentos do Gerenciamento de Serviços de TI: Preparatório para a certificação ITIL® Foundation**. 2. ed. Rio de Janeiro: Brasport, 2011.

**41)** Em Sistemas Operacionais, quando um processo P1 aguarda por um recurso R2, que está alocado ao processo P2, o qual aguarda o recurso R1, alocado em P1, e P1 não libera o recurso para P2, e vice-versa, essa situação é chamada de *deadlocks*, ou impasse, ou ainda adiamento infinito. Um evento que não irá ocorrer, ou a espera de um recurso que nunca vai estar disponível, caracteriza *deadlock*. Quatro condições são necessárias para que se ocorra um *deadlock*, e cada uma pode ter uma abordagem diferente para evitar que se ocorra o *deadlock*. Exclusão mútua é uma situação de ocorrência de *deadlock*. Assinale a alternativa correta, que apresenta a abordagem da condição Exclusão mútua.

- a) Usar *spool* em tudo.
- b) Retomar os recursos alocados.
- c) Ordenar numericamente os recursos.
- d) Requisitar inicialmente todos os recursos necessários.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Acerca das abordagens das condições de ocorrência de *deadlocks*, as abordagens são: Exclusão mútua (Usar spool em tudo). Posse e espera (Requisitar inicialmente todos os recursos necessários). Não preempção (Retomar os recursos alocados). Espera circular (Ordenar numericamente os recursos).

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

**42)** Todas as instruções de um computador são executadas na Unidade Central de Processamento, ou popularmente CPU (*Central Processing Unit*), que possui um conjunto específico de instruções que pode executar. A CPU contém vários registradores, com a seguinte classificação: registradores de propósito geral e registradores específicos. Um desses registradores “contém os *bits* do código de condições, os quais são alterados pelas instruções de comparação, pelo nível de prioridade da CPU, pelo modo de execução, e também por vários outros *bits* de controle”. Esse registrador denomina-se:

- a) Ponteiro de pilha.
- b) Contador de Programa.
- c) IR (*Instruction Register*).
- d) PSW (*Program Status Word*).

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

O PSW (*Program Status Word* – palavra de estado do programa) contém os *bits* do código de condições, os quais são alterados pelas instruções de comparação, pelo nível de prioridade da CPU, pelo modo de execução, e também por vários outros *bits* de controle.

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

**43)** Com a evolução da computação, todos os modernos computadores podem executar diversas coisas ao mesmo tempo. Cada tarefa/serviço realizado pelo computador tem um processo responsável. Numa definição de processo mais objetiva, pode-se dizer que é um programa em execução. Quatro eventos principais são os responsáveis pela criação dos processos, assim como também existem as razões para que algum processo seja encerrado. Alguns exemplos de encerramento do processo podem ser: a execução de uma instrução ilegal, a referência à memória inexistente ou mesmo uma divisão por zero. Assinale a alternativa correta que apresenta a condição de término de um processo referente às ações apresentadas.

- a) Erro fatal (involuntário).
- b) Saída normal (voluntária).
- c) Saída por erro (involuntária).
- d) Cancelamento por um outro processo (involuntário).

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Das quatro condições apresentadas para o encerramento/término de um processo, Erro fatal involuntário é um erro causado pelo processo, muitas vezes por um erro de programa. Entre os vários exemplos estão a execução de uma instrução ilegal, a referência à memória inexistente ou mesmo uma divisão por zero.

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

**44)** Acerca das operações de álgebra relacional, preencha as lacunas abaixo e, em seguida, assinale a alternativa correta.

- I. Produz uma relação que inclui todas as tuplas em R1 ou R2 ou tanto R1 quanto R2; R1 e R2 precisam ser compatíveis na união. (\_\_\_\_\_)
- II. Produz uma relação que inclui todas as tuplas em R1 e R2; R1 e R2 precisam ser compatíveis na união. (\_\_\_\_\_)
- III. Produz uma relação que inclui todas as tuplas em R1 que não estão em R2; R1 e R2 precisam ser compatíveis na união. (\_\_\_\_\_)

- a) União / Interseção / Diferença
- b) Diferença / União / Interseção
- c) Interseção / União / Diferença
- d) Diferença / Interseção / União

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

UNIÃO – Produz uma relação que inclui todas as tuplas em R1 ou R2 ou tanto R1 quanto R2; R1 e R2 precisam ser compatíveis na união. (UNIÃO).

INTERSEÇÃO – Produz uma relação que inclui todas as tuplas em R1 e R2; R1 e R2 precisam ser compatíveis na união. (INTERSEÇÃO).

DIFERENÇA – Produz uma relação que inclui todas as tuplas em R1 que não estão em R2; R1 e R2 precisam ser compatíveis na união. (DIFERENÇA).

Fonte: ELMASRI, Ramez, NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

**45)** Segundo *Tanenbaum* (2009), “algo que permite ao sistema distribuído conseguir uniformidade na presença de diferentes *hardwares* e sistemas operacionais é ter uma camada de *software* no topo do sistema operacional”. A citação anterior trata-se da camada:

- a) Aplicação.
- b) Executiva.
- c) *Middleware*.
- d) Abstração do *hardware*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Algo que permite ao sistema distribuído conseguir uniformidade na presença de diferentes *hardwares* e sistemas operacionais é ter uma camada de *software* no topo do sistema operacional. Essa camada é chamada *Middleware*.

Fontes:

- TANEMBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2009.
- TANEMBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

**46)** Nos sistemas *Linux*, a biblioteca de chamadas de sistema e o próprio sistema operacional, em todas as versões, é fornecida uma série de programas-padrão, sendo que alguns desses são especificados pelo padrão POSIX 1003.2 e outros diferem entre as várias versões *Linux*. Entre esses programas estão o interpretador de comandos (*shell*), editores, compiladores, programas de processamento de textos e os utilitários de manipulação de arquivos. Acerca de alguns programas utilitários requeridos pelo POSIX, preencha as lacunas abaixo e, em seguida, assinale a alternativa correta.

- (\_\_\_\_) – Concatena vários arquivos para a saída-padrão.
- (\_\_\_\_) – Procura certo padrão dentro de um arquivo.
- (\_\_\_\_) – Extrai as últimas linhas de um arquivo.

- a) *cat / grep / tail*
- b) *make / grep / cat*
- c) *head / make / cat*
- d) *tail / head / make*

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

*cat* – Concatena vários arquivos para a saída-padrão.

*grep* – Procura certo padrão dentro de um arquivo.

*tail* – Extrai as últimas linhas de um arquivo.

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

**47)** O IPv4 (*Internet Protocol version 4*), ou Protocolo de *Internet* versão 4, é um endereço que toda máquina conectada a uma rede de computadores possui, que pode ser atribuído dinamicamente, através do servidor, com o uso do protocolo DHCP, ou ser configurado estaticamente, por um usuário ou mesmo alguém com conhecimento para essa atividade, e que tenha autorização para fazê-lo. É formado por quatro octetos, possui 32 *bytes* de tamanho e é dividido em classes. Assinale a alternativa correta referente a um IP da classe D.

- a) **11100000.1010.1100100.1011**
- b) 11011111.10110000.10111.11
- c) 11011101.11011110.10110.1101111
- d) 11011110.11110001.11001000.1100110

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

O IP de classe D é: 11100000.1010.1100100.1011 (224.10.100.11). A classe D vai de 224.0.0.0 a 239.255.255.255.

Fonte: TANENBAUM, Andrew S.; WETHERALL, D. - **Redes de Computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

**48)** O padrão 802.11, da subcamada MAC, é bem diferente do protocolo da Ethernet, e isso acontece por causa da complexidade que é inerente às comunicações sem fio. Acerca dos espaçamentos entre os quadros do 802.11, analise as afirmativas a seguir:

- I. O intervalo entre quadro de dados regulares é chamado de espaçamento entre quadros DCF, ou DIFS (*DCF InterFrame Spacing*).
- II. O menor intervalo é o espaçamento estendido entre quadros, ou EIFS (*Extended InterFrame Spacing*).
- III. O último intervalo, o espaçamento curto entre quadros, ou SIFS (*Short InterFrame Spacing*), só é usado por uma estação que tenha acabado de receber um quadro defeituoso ou desconhecido, a fim de informar sobre o problema.

Está(ão) correta(s) apenas a(s) afirmativa(s)

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

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

II. O menor intervalo é o espaçamento curto entre quadros, ou SIFS (*Short InterFrame Spacing*).

III. O último intervalo, o espaçamento estendido entre quadros, ou EIFS (*Extended InterFrame Spacing*) só é usado por uma estação que tenha acabado de receber um quadro defeituoso ou desconhecido, a fim de informar sobre o problema.

Fonte: TANENBAUM, Andrew S.; WETHERALL, D. - **Redes de Computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

**49)** O WIMAX (*Worldwide Interoperability for Microwave Access*) é considerado um padrão de rede sem fio metropolitana, com velocidade da casa dos megabits/s, se aproximando do cabo ou DSL. WIMAX foi projetado para transportar pacotes IP pelo ar, dessa forma se conectando a uma rede com fios, baseada em IP, sem muito alarde. Nesse padrão de rede, um esquema flexível divide o canal entre as estações, de modo que diferentes conjuntos de subportadoras podem ser atribuídos a diferentes estações, e assim mais de uma estação pode enviar ou receber ao mesmo tempo. O referido esquema está apresentado em:

- a) TDD – *Time Division Duplex*.
- b) FDD – *Frequency Division Duplex*.
- c) OFDM – *Orthogonal Frequency Division Multiple*.
- d) OFDMA – *Orthogonal Frequency Division Multiple Access*.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

OFDMA – *Orthogonal Frequency Division Multiple Access* – esquema flexível para dividir o canal entre estações. Com OFDMA diferentes conjuntos de subportadoras podem ser atribuídos a diferentes estações, e dessa forma mais de uma estação pode enviar ou receber ao mesmo tempo.

Fonte: TANENBAUM, Andrew S.; WETHERALL, D. - **Redes de Computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

**50)** O SMTP (*Simple Mail Transfer Protocol*) é um protocolo de envio de *e-mail*, que originariamente usava a porta 25, mas, em virtude do alto número de *spams* que estava circulando no Brasil, o CGI (Comitê Gestor da Internet) do Brasil emitiu uma recomendação, para que os provedores de *e-mail* trocassem a porta padrão para 587. Para resolver diversos problemas, relacionados ao processamento da mensagem, o SMTP foi revisado para ter um mecanismo de extensão. ESMTP (*Extended SMTP*) é o nome que se dá ao uso do SMTP com extensões. Uma dessas extensões permite ao servidor aceitar mensagens grandes em pedaços. Trata-se de:

- a) AUTH.
- b) STARTTLS.
- c) **CHUNKING**.
- d) BINARYMIME.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

CHUNKING – Servidor aceita mensagens grandes em pedaços.

Fonte: TANENBAUM, Andrew S.; WETHERALL, D. - **Redes de Computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

51) Em Java, as sequências de escape podem ser utilizadas para impressão de texto numa mesma linha utilizando-se de várias instruções, ou mesmo realizar a impressão do texto em várias linhas utilizando-se de uma única instrução. Uma dessas sequências é utilizada para posicionar o cursor de tela no início da linha atual – não avançando para a próxima linha. Acerca dessa sequência de caracteres, assinale a alternativa correta.

- a) \n
- b) \r**
- c) \s
- d) \t

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

A sequência de escape \r significa retorno de carro. Posicionar o cursor de tela no início da linha atual – não avança para a próxima linha. Qualquer saída de caracteres depois do retorno de carro sobrescreve a saída de caracteres anteriormente gerados na linha atual.

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

52) *Applets* são considerados programas em Java, que são tipicamente incorporados a documentos XHTML, que também podem ser chamadas páginas *web*. Acerca dos *applets* em Java, analise as afirmativas a seguir:

- I. O método *start* é chamado uma vez pelo contêiner de *applets* para inicializar um *applet* quando ele é carregado.
- II. O método *init* é chamado pelo contêiner de *applets* depois dos métodos *paint* e *stop*. O método *init* também é chamado quando o *applet* precisa ser repintado.
- III. O método *destroy* é chamado pelo contêiner de *applets* quando o *applet* é removido da memória.
- IV. Cinco métodos do ciclo de vida de um *applet* são chamados pelo contêiner de *applets* entre o momento em que o *applet* é carregado no navegador e o momento em que ele é fechado pelo navegador.

Estão corretas apenas as afirmativas

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

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

- I. O método *init* é chamado uma vez pelo contêiner de *applets* para inicializar um *applet* quando ele é carregado.
- II. O método *paint* é chamado pelo contêiner de *applets* depois dos métodos *init* e *start*. O método *paint* também é chamado quando o *applet* precisa ser repintado.

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

53) Em Java, uma indicação de que ocorre um problema, durante a execução de um programa, é chamada de exceção. Existem vários tipos de exceções, como divisão por zero, esgotamento de memória, entre outros. Sobre o tratamento de exceções, preencha as lacunas das afirmativas abaixo e, em seguida, assinale a alternativa correta.

- I. Pelo menos um bloco \_\_\_\_\_ ou um bloco \_\_\_\_\_ deve seguir imediatamente o bloco \_\_\_\_\_.
- II. Se um bloco \_\_\_\_\_ lançar uma exceção, o bloco \_\_\_\_\_ ainda executará. A exceção é então transferida para o próximo bloco \_\_\_\_\_ externo.

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

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

- I. Pelo menos um bloco *catch* ou um bloco *finally* deve seguir imediatamente o bloco *try*.  
II. Se um bloco *catch* lançar uma exceção, o bloco *finally* ainda executará. A exceção é então transferida para o próximo bloco *try* externo.

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

- 54)** Assim como em Java existem as exceções, há também os tratamentos para elas, permitindo remover o código de tratamento de erro da linha principal de execução do programa. Isso aprimora a clareza do programa e destaca a sua capacidade de modificação. Sobre o tratamento de exceções, em Java, assinale a alternativa **incorreta**.
- a) Uma instrução *throw* pode lançar qualquer objeto *Throwable*.
  - b) O Java se distingue entre duas categorias de exceções: verificadas e não verificadas.
  - c) O tratamento de exceções processa erros síncronos, que ocorrem quando uma instrução é executada.
  - d) **Se houver múltiplos blocos *try* correspondentes quando uma exceção ocorrer, somente o primeiro é executado.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Se houver múltiplos blocos *catch* correspondentes quando uma exceção ocorrer, somente o primeiro é executado.

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

- 55)** Uma coleção organizada de dados recebe o nome de banco de dados, e um SGBD (Sistema de Gerenciamento de Banco de Dados) fornece mecanismos para armazenar, recuperar, organizar e modificar dados para diversos usuários. Em Java, a JDBC (*Java Database Connectivity*) permite que os programas comuniquem-se com banco de dados e possam manipular seus dados. Acerca da manipulação de dados, com a JDBC, assinale a alternativa correta.
- a) O método *setAutoCommit* determina o estado de confirmação automática para a *Connection*.
  - b) **A interface *RowSet* configura a conexão de banco de dados e executa a consulta automaticamente.**
  - c) *JdbcRowSet*, um *RowSet* conectado, empacota um objeto *JdbcRowSet* e permite que se role e atualizem as linhas.
  - d) O método *getAutocommit* especifica se cada instrução SQL é confirmada depois de ser completada (um argumento *true*) ou se várias instruções SQL devem ser agrupadas como uma transação.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

A informação correta é: A interface *RowSet* configura a conexão de banco de dados e executa a consulta automaticamente. O método *getAutoCommit* determina o estado de confirmação automática para a *Connection*. O método *setAutocommit* especifica se cada instrução SQL confirma depois que ela for completada (um argumento *true*) ou se várias instruções SQL devem ser agrupadas como uma transação. *JdbcRowSet*, um *RowSet* conectado, empacota um objeto *ResultSet* e permite que você role e atualize as linhas. Ao contrário de um objeto *ResultSet*, um objeto *JdbcRowSet* é rolável e atualizável por padrão.

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

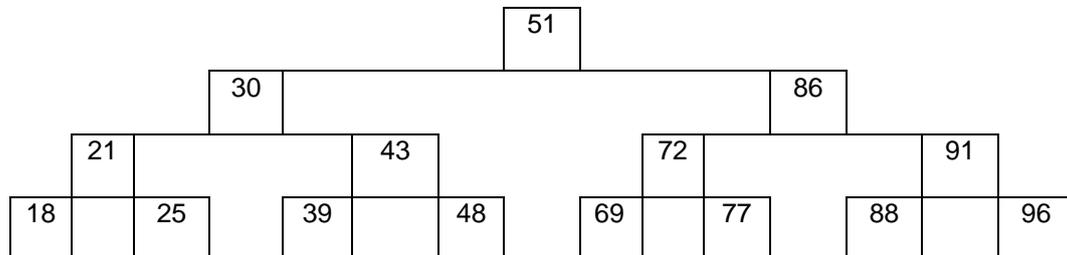
- 56)** Informe se são verdadeiras (V) ou falsas (F) as afirmativas sobre *Threads* em Java e, em seguida, marque a opção com a sequência correta.
- ( ) Um objeto *Runnable* representa uma “tarefa” que pode executar concorrentemente com outras tarefas.
  - ( ) Pode-se prever a ordem em que as *threads* serão agendadas, sabendo a ordem em que elas foram criadas e iniciadas.
  - ( ) O método *ExecutorService* (do pacote *java.util.concurrent*) estende a interface *Executor* e declara outros métodos para gerenciar o ciclo de vida de um *Executor*.
  - ( ) Uma *thread* adormecida permanece no estado de espera sincronizada por um período de tempo designado, depois do qual ela retorna ao estado executável.
  - ( ) Um objeto que implementa a interface *Executor* pode ser criado com os métodos *static* declarados na classe *Executors* (do pacote *java.util.concurrent*).
  - ( ) A sincronização de *threads* coordena o acesso a dados compartilhados por múltiplas *threads* concorrentes.
- a) F – V – V – F – F – V
  - b) **V – F – F – V – F – V**
  - c) F – F – V – V – V – F
  - d) V – V – F – F – V – F

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Pode-se prever a ordem em que as *threads* serão agendadas, mesmo se souber a ordem em que elas foram criadas e iniciadas. A interface *ExecutorService* (do pacote *java.util.concurrent*) estende a interface *Executor* e declara outros métodos para gerenciar o ciclo de vida de um *Executor*. Um objeto que implementa a interface *ExecutorService* pode ser criado com os métodos *static* declarados na classe *Executors* (do pacote *java.util.concurrent*).

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

57) Analise a árvore de pesquisa binária apresentada:



Assinale a alternativa correta que apresenta o percurso na pós-ordem.

- a) 18 – 25 – 21 – 39 – 48 – 43 – 30 – 69 – 77 – 72 – 88 – 96 – 91 – 86 – 51
- b) 18 – 21 – 25 – 30 – 39 – 43 – 48 – 69 – 72 – 77 – 86 – 88 – 91 – 96 – 51
- c) 18 – 25 – 21 – 39 – 48 – 43 – 69 – 77 – 72 – 88 – 96 – 91 – 30 – 86 – 51
- d) 18 – 25 – 39 – 48 – 69 – 77 – 88 – 96 – 21 – 43 – 72 – 01 – 30 – 86 – 51

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Em um percurso na pós-ordem, o valor em cada nó é processado depois dos valores de seus filhos. 18 – 25 – 21 – 39 – 48 – 43 – 30 – 69 – 77 – 72 – 88 – 96 – 91 – 86 – 51

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

58) Uma consulta em um Banco de Dados Distribuídos pode ser processada em diversos estágios. Um desses estágios “[...] consiste em selecionar uma estratégia com base em uma lista de candidatas que está mais próxima do ideal. Uma lista de consultas de candidatas pode ser obtida ao permutar a ordenação das operações em uma consulta de fragmento gerada pelo estágio anterior.” Esse estágio apresenta-se em:

- a) Localização.
- b) Mapeamento de consulta.
- c) Otimização de consulta local.
- d) Otimização global da consulta.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Otimização global da consulta: A otimização consiste em selecionar uma estratégia com base em uma lista de candidatas que está mais próxima do ideal. Uma lista de consultas candidatas pode ser obtida ao permutar a ordenação das operações em uma consulta de fragmento gerada pelo estágio anterior.

Fonte: ELMASRI, Ramez, NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

59) Analise o seguinte fragmento de um código em Java:

```
{
    int total = 0;
    for (int number = 2; number <= 25; number += 2)
        total += number;
    System.out.printf ("Resultado is %d\n", total);
    //
}
```

O resultado dessa instrução é:

- a) 154
- b) 156**
- c) 180
- d) 324

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Essa instrução pede para serem somados todos os inteiros, pares, de 2 até 25. Serão somados: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24 e o resultado é: 156

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

**60)** Acerca do Sistema Operacional *Windows*, preencha as lacunas abaixo e, em seguida, assinale a alternativa correta.

“\_\_\_\_\_ são uma característica do sistema operacional OS/2 implementada no *Windows* para compatibilidade.”

“\_\_\_\_\_ são como pipes, só que em geral conectam processos de máquinas diferentes.”

- a) Eventos / Mailslots
- b) Mailslots / Soquetes**
- c) Seções críticas / Eventos
- d) Soquetes / Seções críticas

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

Mailslots são uma característica do sistema operacional OS/2 implementada no *Windows* para compatibilidade. Soquetes são como pipes, só que em geral conectam processos de máquinas diferentes.

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