

## CONHECIMENTOS ESPECÍFICOS

Acerca dos conceitos de computadores e sistemas computacionais, julgue os próximos itens.

- 51 A compilação é o processo de análise de um programa escrito em linguagem de alto nível, denominado programa-fonte, e sua conversão em um programa equivalente, escrito em linguagem binária de máquina, denominado programa-objeto.
- 52 Caso um hodômetro — instrumento que mede as distâncias percorridas por um automóvel, com valores expressos em algarismos hexadecimais — apresente, em determinado deslocamento do automóvel, o valor inicial A3FF e o valor final A83C, em quilômetros, é correto afirmar que o veículo percorreu uma distância de 43E quilômetros.
- 53 A unidade de controle é o elemento da unidade central de processamento responsável por emitir os sinais de controle para ativar a realização de cada etapa do ciclo da instrução, sendo iguais esses sinais de controle para todas as instruções.

A respeito dos conceitos e das características dos sistemas operacionais, julgue os itens subsequentes.

- 54 O sistema Windows dá suporte a dois ambientes de execução especiais, ambos chamados de WOW (*Windows-on-Windows*). O WOW32 é usado em sistemas de 32 bits x86 para executar aplicações de 16 bits no Windows 3.x mapeando as chamadas de sistema e os parâmetros entre os ambientes de 16 e 32 bits. De forma similar, o WOW64 permite que as aplicações do Windows de 32 bits sejam executadas em sistemas x64.
- 55 As aplicações desenvolvidas para Android são escritas em Java, não sendo, entretanto, consideradas uma máquina virtual Java.
- 56 No Linux, o comando `ls [ape]*` relaciona todos os arquivos cujos nomes comecem com a sequência `ape`.

Com relação às redes de computadores e ao processamento distribuído, julgue os itens que se seguem.

- 57 O DNS (*domain name system*) organiza o espaço de nomes em uma estrutura hierárquica que permite descentralizar as responsabilidades envolvidas na atribuição de nomes, podendo usar os serviços do UDP ou TCP por meio da porta-padrão 53.
- 58 Um repetidor regenera um sinal, interliga segmentos de uma LAN e não tem nenhum recurso de filtragem.
- 59 O serviço *web* obedece a um conjunto de padrões que lhe permite ser descoberto e acessado pela Internet por aplicações de cliente que também adotam os mesmos padrões.
- 60 No IPv4, o campo de comprimento total define o comprimento total do datagrama, sem considerar o tamanho do cabeçalho, que já é previamente conhecido.

No que diz respeito aos conceitos e fundamentos de lógica de programação, julgue os itens seguintes.

- 61 Em PHP 6, a passagem de variáveis entre páginas, por meio do uso de sessões, está limitada a informações fornecidas pelo usuário em uma página.
- 62 Em Java, para que uma classe interna possa ser instanciada, é necessário haver uma instância de uma classe externa que possa ser usada para se associar à primeira.
- 63 No HTML 5, a tag `<rp>` é usada para definir o que deve ser mostrado na hipótese de determinado *browser* não suportar anotações *ruby*.
- 64 Com o uso de XSLT, é possível fazer a transformação de qualquer dado XML em XHTML para todos os tipos de *browser*, procedendo às transformações no próprio *browser*.
- 65 Por característica, o algoritmo *quicksort* apresenta melhor desempenho que o *merge sort*.

Com relação a testes de sistemas de *software*, julgue os itens a seguir.

- 66 Simulações e *benchmarks* são técnicas de modelagem bastante utilizadas para a avaliação de desempenho de sistemas de *software*.
- 67 Por meio do JUnit, *framework* de teste que viabiliza a documentação e execução automática de testes de unidade em Java, é possível obter informações a respeito da cobertura obtida pelos casos de teste.
- 68 Embora o JUnit apresente limitações, como a de não apoiar a aplicação de um critério de teste, esse *framework* pode ser utilizado ainda que estejam disponíveis apenas o *bytecode* e a especificação do programa.
- 69 O teste de unidade, cujo foco são as menores unidades do programa, como as funções, os procedimentos, os métodos ou as classes, somente pode ser realizado após a implantação de todas as unidades do programa.
- 70 No teste de integração, enfatiza-se a construção da estrutura do sistema: à medida que as diversas partes do *software* são colocadas para trabalhar juntas, é preciso verificar se a interação entre elas funciona de maneira adequada, sem a ocorrência de erros.
- 71 O teste de integração objetiva tanto a verificação da correta implantação dos requisitos funcionais quanto a dos requisitos não funcionais, como desempenho e vulnerabilidade.
- 72 As ferramentas do projeto Selenium são adequadas para a realização de testes funcionais automatizados de aplicações *web*. A API Selenium `WebDriver`, por exemplo, permite a realização de testes de aplicações *web* por meio dos métodos `get()` e `findElement()`.

No que se refere à arquitetura e à programação Java, julgue os itens que se seguem.

- 73 No JSE (Java Standard Edition) versão 8, é possível utilizar recursos inerentes à programação funcional por meio de uma nova característica da linguagem chamada expressões `Lambda`, que permitem o tratamento de funções como argumentos de métodos.
- 74 A programação orientada a aspectos constitui-se como um paradigma de programação para a construção de programas em que os interesses transversais (*crosscutting concerns*) ficam separados dos interesses básicos, e não espalhados pelo sistema.
- 75 O Hibernate permite operações de banco de dados relacionais em ambiente Java e a persistência transparente de classes POJO (*plain old Java objects*), que devem ter um construtor com apenas um argumento, para referenciar o servidor SQL no qual os dados manipulados pela classe serão persistidos.
- 76 Na plataforma JEE (Java Enterprise Edition) versão 6, não é possível encontrar bibliotecas da própria plataforma para o consumo dos serviços REST no formato JSON.
- 77 Utilizando-se a ferramenta de análise estática Findbugs, não será possível identificar, no trecho de código a seguir, o tipo de erro ocorrido. Tal erro só pode ser detectado em tempo de execução, quando a função `foundType()` for invocada.
- ```
public String foundType( ) {
    return this.foundType( );
}
```

Acerca das ferramentas de *software* para suporte às atividades de análise e controle de versão, julgue os próximos itens.

- 78 Os comandos da ferramenta Git são relativamente simples: para adicionar, por exemplo, um arquivo novo ao repositório no Git, basta utilizar o comando `commit` depois de efetuar o comando `add`.
- 79 Para a análise estática de código, a ferramenta Findbugs, que permite a verificação dos problemas relativos a defeitos e convenções de codificação em um *software*, é mais completa que a Checkstyle, que possibilita apenas a detecção dos problemas relativos a convenções de codificação.
- 80 As ferramentas de controle de versão Git e SVN oferecem o mesmo grau de confiabilidade no armazenamento das informações e são ambas implantadas conforme o conceito de sistemas de controle de versão distribuído.

No que se refere a práticas ágeis, julgue os itens de 81 a 85.

- 81 A integração contínua constitui um dos pilares das práticas ágeis: a constante integração das respectivas partes deve ocorrer diariamente e ser realizada concomitantemente à atividade de programação em pares.

- 82 Na atividade de TDD (*test-driven development*), a escrita de teste primeiro define implicitamente tanto uma interface quanto uma especificação do comportamento para a funcionalidade que está sendo desenvolvida, estando, entretanto, a viabilidade do uso dessa abordagem limitada aos processos de desenvolvimento de *software* que seguem as práticas ágeis.
- 83 Por não existir relação direta entre as práticas ágeis de integração contínua e de entrega contínua, pode-se garantir a agilidade do processo pela realização de tais atividades em paralelo, ainda que relacionadas à mesma versão.
- 84 Para se evitar dispêndio de muito tempo na alteração do código e pouco tempo na programação de novas funcionalidades, a prática de *refactoring* deve restringir-se aos casos em que haja grandes porções de código a ser refatorado.
- 85 O fato de cada código ter sua própria estrutura não constitui impedimento para se estabelecerem padrões para a prática de *refactoring*, sendo possível, por exemplo, utilizar padrões de projeto, como o *factory method*, para remover duplicações no código.

No que diz respeito aos sistemas de suporte à decisão, julgue os itens subsequentes.

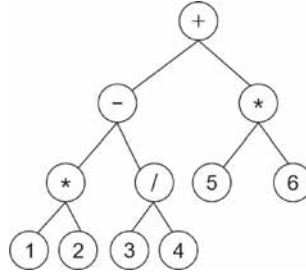
- 86 O processamento analítico *online* (OLAP) é adequado para a descoberta de padrões e relacionamentos em dados corporativos e para a inferência de regras que prevejam o efeito de decisões gerenciais.
- 87 Para viabilizar o processo de inteligência de negócio, é necessário o uso de um *Data Warehouse*.
- 88 Em um *Data Warehouse* (armazém de dados) que apresente a característica de ser não volátil, os dados não são atualizados após a inserção inicial.
- 89 Uma mudança nos valores de dados de um processamento analítico *online* que use a tabulação cruzada para apoio à análise e mineração de dados pode aumentar o número de colunas.

Julgue os itens seguintes, com relação aos sistemas de gestão de conteúdo e à segurança da informação.

- 90 Diferentemente dos portais públicos da Internet, os portais de informações corporativas buscam aumentar a inteligência informacional da organização.
- 91 A disseminação seletiva da informação, por buscar atender aos interesses e às necessidades de grupos específicos da organização, é uma ação que contraria os objetivos da arquitetura da informação.

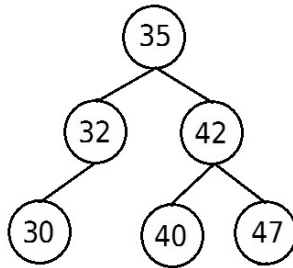
No que diz respeito às estruturas de informação, julgue os itens subsecutivos.

92 Percorrendo a árvore binária abaixo em pós-ordem, obtém-se o seguinte resultado:  $1 * 2 - 3 / 4 + 5 * 6$ .

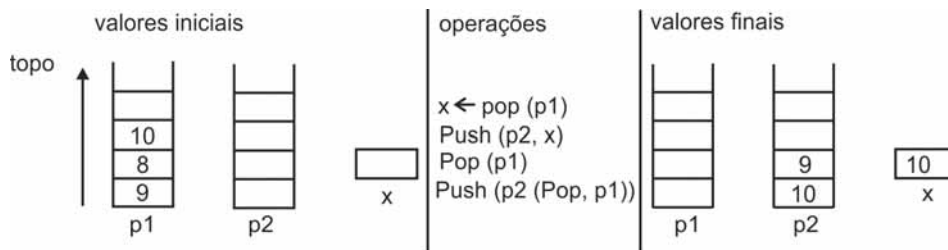


93 Se um grafo simples que represente os usuários de uma rede social tem a soma dos graus de cada vértice igual a 16, então o número de enlaces de comunicação entre os usuários é 8.

94 A inserção dos elementos 47, 35, 40, 32, 42 e 30, nessa ordem, em uma árvore binária balanceada inicialmente vazia apresenta o resultado mostrado na figura abaixo, após a realização de todas as operações necessárias.



95 Considere a figura a seguir, que mostra os valores iniciais das pilhas p1 e p2, nas quais foram realizadas operações de empilhar (push) e desempilhar (pop), além de ter sido atribuído um valor para a variável x (x ← valor). Nesse caso, realizando-se as operações indicadas, na ordem listada, obtém-se os valores finais das pilhas e da variável x mostrados.



Nos comandos em linguagem de consulta estruturada (SQL) apresentados a seguir, as chaves primárias estão sublinhadas e apenas horas\_gastas é do tipo numérico, os demais campos são do tipo caractere. Em uma tarefa, com a utilização de um mesmo veículo, pode haver a participação de mais de um motorista na função de auxiliar.

```
MOTORISTA (cod_mot, nome, cnh)
TAREFA (cod_tarefa, cod_mot, cod_mot_auxiliar, placa, descricao, horas_gastas)
VEÍCULO (placa, ano, modelo)
```

Tendo como base as informações acima, julgue os itens a seguir.

96 O comando a seguir produz o seguinte resultado: mostra o código do motorista auxiliar, a média das horas gastas na execução das tarefas de que esse motorista participou, excluindo as horas realizadas na tarefa AB e os motoristas auxiliares que participaram de apenas uma tarefa. A resposta é classificada em ordem descendente da média das horas gastas.

```
Select cod_mot_auxiliar, avg(horas_gastas) from tarefa where cod_tarefa != 'AB' group by cod_mot_auxiliar having count(*) > 1 order by 2 desc;
```

97 O comando a seguir exclui a chave primária, cod\_tarefa, da tabela TAREFA.

```
Alter table tarefa drop primary key cod_tarefa;
```

98 O comando a seguir lista o nome dos motoristas que executaram uma tarefa seguido do nome do seu auxiliar. O nome do motorista é apresentado mesmo que ele não tenha tido, na tarefa, um auxiliar.

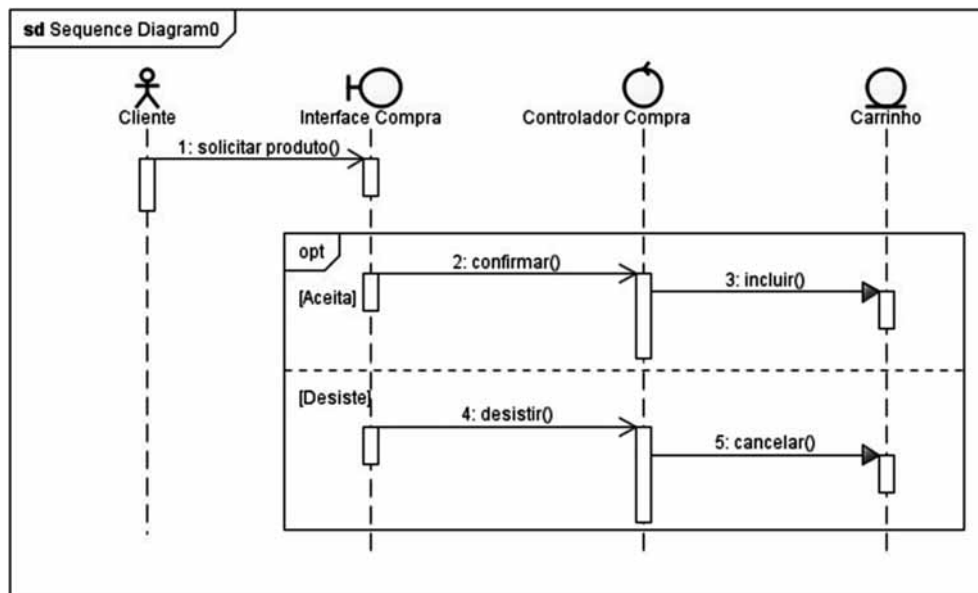
```
select m.nome as motorista, a.nome as auxiliar
from motorista a
right join tarefa t on t.cod_mot_auxiliar = a.cod_mot
right join motorista m on t.cod_mot = m.cod_mot;
```

Julgue os itens que se seguem, no que se refere à administração de dados.

- 99 No contexto da qualidade dos dados, o mapeamento distorcido do mundo real para um estado errado no sistema de informação implica a falta de acurácia.
- 100 Informações sobre o perfil de quem pode acessar a informação, os direitos de reprodução, a autenticação e os dados de segurança são exemplos de metadados de preservação.
- 101 Na elaboração e implantação de um projeto de banco de dados, a pesquisa em um índice existente para os atributos com a restrição de serem únicos é suficiente para o apoio à verificação dessa restrição de exclusividade.

Com base nos conceitos gerais e nas disciplinas de engenharia de *software*, julgue os próximos itens.

- 102 O princípio da extensibilidade, adotado pelo metamodelo da UML 2.3, permite que sejam usados perfis na customização da linguagem para se modelarem cenários específicos de uma organização.
- 103 No ciclo de vida clássico, ou modelo em cascata, as fases concluídas são congeladas para se evitar o retrabalho e maximizarem-se os custos de produção e a aprovação de documentos.
- 104 Em um diagrama de sequência, como o mostrado na figura abaixo, deve-se usar um fragmento combinado com o operador de `opt` (opção) para representar uma escolha entre dois ou mais comportamentos.



- 105 Ao adquirir licença de uso de uma planilha eletrônica como ferramenta para automatização de escritório da empresa, está-se adotando uma engenharia de *software* orientada a reúso.

Com relação à área de abrangência do Padrão de Interoperabilidade do Governo Eletrônico (e-Ping), julgue o item abaixo.

- 106 O e-Ping é definido por um conjunto mínimo de políticas e especificações técnicas que regulamentam a utilização da tecnologia da informação e comunicação e abrange as seguintes áreas: segurança; meios de acesso; interconexão; organização e intercâmbio de informações; e áreas de integração para governo eletrônico.

Acerca dos processos de desenvolvimento de *software*, julgue os itens subsequentes.

- 107 A etapa de planejamento do Extreme Programming (XP) inicia-se com a escrita de *UserStories* (história do usuário). Por meio dessa ferramenta, aqueles que conhecem a técnica de construção de uma solução poderão guiar quem necessita dessa solução no exercício de descrevê-la de forma simples e concisa.
- 108 O *Scrum* é um conjunto simples e eficaz de regras e ferramentas que são utilizadas para maximizar resultados. O *ScrumMaster* exerce o papel de facilitador e motivador da equipe, além de garantir que as regras e as ferramentas sejam utilizadas com vistas à criatividade do trabalho e ao retorno do investimento.

A respeito de acessibilidade eMAG, julgue o próximo item.

**109** Na validação automática de conteúdos eletrônicos, utilizam-se leitores de tela com tecnologia assistiva, em que o validador automático, simulando um teclado, percorre toda a página, a fim de verificar os comportamentos de atalhos e folhas de contraste.

A respeito de usabilidade, julgue os itens subsecutivos.

**110** De acordo com a diretriz de consistência e familiaridade do governo eletrônico, as seguintes convenções devem ser utilizadas nas páginas *web*: *links* azuis, *links* sublinhados, botões desenhados como botões (elementos clicáveis), caixa de busca no canto superior direito, nome (ou logotipo) do sítio no canto superior esquerdo e logotipo do sítio com função de *link* de volta à página inicial.

**111** Usabilidade de programas e conteúdo refere-se ao acesso a locais, produtos, serviços e informações disponíveis ao maior número e variedade possível de pessoas, independentemente da capacidade motora, cultural e social de cada uma delas.

Com relação às soluções de tecnologia da informação e comunicação, julgue o item a seguir.

**112** REST é uma técnica de engenharia de *software* para sistemas hipermídia distribuídos. De acordo com essa técnica, o estado da informação deve ser mantido no cliente, e o servidor não deve guardar o estado da comunicação de nenhum cliente que se comunique com o servidor, além de uma única requisição.

Julgue o item abaixo a respeito de geoprocessamento.

**113** As coordenadas do Sistema de Referência Geocêntrico para a América do Sul (SIRGAS) acompanham todos os movimentos e variações dos pontos em relação a um sistema de referência fixa, servindo, assim, para os estudos científicos da figura da Terra.

Julgue o seguinte item, acerca da configuração servidor de aplicações *web* Apache.

**114** Pelo módulo Expires (`mod_expires.c`) é possível configurar o servidor de aplicações *web* Apache para que o navegador do cliente faça *cache* de determinados tipos de arquivos, o que reduz a quantidade de requisições feitas ao servidor.

Julgue o próximo item, acerca de arquitetura em camadas.

**115** O MVC (*Model View Controller*) é um modelo de arquitetura de *software* que implementa uma operação de clonagem em cada uma das classes concretas do modelo. Cada modelo é uma instância de um controle especializado em construir visões iguais a si mesmo. Esse padrão é utilizado frequentemente em linguagens estaticamente tipadas, como C++ e Java.

A respeito de computação nas nuvens, julgue o item a seguir.

**116** A DaaS (*Database as a Service*), uma das formas de disponibilizar computação nas nuvens, oferece uma solução de comunicação unificada, hospedada em uma central de dados do provedor ou fabricante, entre fornecedores e clientes.

A respeito de qualidade de *software*, julgue o item abaixo.

**117** Em engenharia de *software*, acoplamento refere-se a uma medida de um componente, ao passo que, em casos de sistemas, cada componente pode ser a sua medida de acoplamento. O acoplamento de um componente é alto quando os seus integrantes estão relacionados a um tema comum, isto é, têm o mesmo objetivo, fazem uma única tarefa.

Com relação às técnicas de avaliação de sistemas e tratamento de exceções, julgue o item seguinte.

**118** A linguagem Java possui *checked exceptions* que estendem `java.lang.Exception`, em que o compilador força o programador a capturar tais exceções. Essas exceções devem ser tratadas com um bloco *try-catch* ou com um *throws*.

Julgue o item abaixo a respeito de arquiteturas para desenvolvimento de aplicativos em dispositivos móveis.

**119** A arquitetura do Android disponibiliza o módulo denominado OOM *handling*, que permite às aplicações o gerenciamento do mecanismo de baixo consumo de energia do aparelho; por exemplo, se um processo necessitar ser executado em segundo plano, o referido módulo possibilitará a desativação temporária desse mecanismo até a finalização do processo em execução.

Acerca da consistência dos dados, julgue o item a seguir.

**120** Os processos que optam por utilizar a consistência eventual dos dados devem garantir que as informações serão atualizadas imediatamente após sua modificação e que serão consistentes para todos os usuários da entidade.

## PROVA DISCURSIVA

- Nesta prova, tanto na dissertação como nas questões, faça o que se pede, usando os espaços para rascunho indicados no presente caderno. Em seguida, transcreva os textos para o **CADERNO DE TEXTOS DEFINITIVOS DA PROVA DISCURSIVA**, nos locais apropriados, pois **não serão avaliados fragmentos de texto escritos em locais indevidos**.
- Tanto na dissertação como nas questões, qualquer fragmento de texto além da extensão máxima de linhas disponibilizadas será desconsiderado.
- No **caderno de textos definitivos**, identifique-se apenas no cabeçalho da primeira página, pois **não será avaliado** texto que tenha qualquer assinatura ou marca identificadora fora do local apropriado.
- Em cada parte dessa prova, ao domínio do conteúdo serão atribuídos até **10,00 pontos**, dos quais até **0,50 ponto** será atribuído ao quesito apresentação (legibilidade, respeito às margens e indicação de parágrafos) e estrutura textual (organização das ideias em texto estruturado).

### DISSERTAÇÃO

O Linux difere de outras variantes UNIX pelo fato de o projeto do *kernel* básico definir apenas um *kernel* de sistema operacional. O *kernel* tem de ser empacotado junto com comandos, *daemons* e outros *softwares* para formar um sistema operacional completo e utilizável — em termos do Linux, uma “distribuição”. Todas as distribuições do Linux compartilham da mesma linhagem do *kernel*, porém os materiais auxiliares (uma vasta coleção de *software* desenvolvida nos últimos trinta anos por milhares de pessoas), que vêm junto com esse *kernel*, podem variar significativamente entre essas distribuições.

Evi Nemeth, Garth Snyder e Trent R. Hein. **Manual completo do Linux**. São Paulo: Pearson Prentice Hall, 2007, p. 5 (com adaptações).

Considerando que o trecho acima tem caráter unicamente motivador, fundamente, em texto dissertativo, a escolha de uma distribuição Linux para configurar tanto servidores de rede local quanto servidores dedicados. Ao elaborar seu texto, aborde, necessariamente, os seguintes aspectos:

- ▶ capacidade de fornecimento de suportes específicos; [valor: 3,00 pontos]
- ▶ ferramentas administrativas existentes; [valor: 3,00 pontos]
- ▶ oferecimento de escalabilidade. [valor: 3,50 pontos]

**RASCUNHO – DISSERTAÇÃO**

|    |  |
|----|--|
| 1  |  |
| 2  |  |
| 3  |  |
| 4  |  |
| 5  |  |
| 6  |  |
| 7  |  |
| 8  |  |
| 9  |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |
| 21 |  |
| 22 |  |
| 23 |  |
| 24 |  |
| 25 |  |
| 26 |  |
| 27 |  |
| 28 |  |
| 29 |  |
| 30 |  |

**QUESTÃO 1**

Em engenharia de *software*, as técnicas de análise estática de código e os testes de unidade são reconhecidos como formas complementares e bastante eficientes para a identificação de defeitos no código. Entre as práticas ágeis, a técnica de *Test-driven Development* é apresentada no contexto de teste de unidade realizado de forma automática. No entanto, não há recomendação explícita do uso automatizado de técnicas de análise estática entre as práticas ágeis.

Considerando que o texto acima tem caráter unicamente motivador, discorra acerca da eficiência de construção de *software* sem defeitos em metodologias ágeis, sob a perspectiva de análise estática e dinâmica do código. Ao elaborar seu texto, aborde, necessariamente, os seguintes aspectos:

- ▶ contemplação de técnicas de análise estática em metodologias ágeis; [valor:3,00 pontos]
- ▶ formas automáticas de se realizar análise estática; [valor:3,00 pontos]
- ▶ eficiência da construção de *software* sem defeitos em metodologias ágeis sob a perspectiva de análise estática de código. [valor:3,50 pontos]

**RASCUNHO – QUESTÃO 1**

|    |  |
|----|--|
| 1  |  |
| 2  |  |
| 3  |  |
| 4  |  |
| 5  |  |
| 6  |  |
| 7  |  |
| 8  |  |
| 9  |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |



**QUESTÃO 2**

Com base nos diagramas da UML versão 2.4, redija um texto dissertativo a respeito do uso dos diagramas de interação na análise e no projeto de sistemas orientados a objeto. Ao elaborar seu texto, aborde, necessariamente, os seguintes aspectos:

- ▶ característica comum dos diagramas de interação; [valor: 1,50 ponto]
- ▶ lista dos diagramas de interação; [valor: 2,00 pontos]
- ▶ função de cada um dos diagramas de interação e critérios que definem a sua escolha na modelagem de um *software*.  
[valor: 6,00 pontos]

**RASCUNHO – QUESTÃO 2**

|    |  |
|----|--|
| 1  |  |
| 2  |  |
| 3  |  |
| 4  |  |
| 5  |  |
| 6  |  |
| 7  |  |
| 8  |  |
| 9  |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |

**QUESTÃO 3**

Considere que a direção de uma organização, pretendendo implantar metodologias de desenvolvimento de *softwares* ágeis para atender a transformações implantadas na empresa, percebeu que os processos não estão suficientemente maduros, o que requer mudanças frequentes durante o desenvolvimento dos *softwares*. Entre as metodologias ágeis, a administração do órgão escolheu o *Scrum*.

Tendo como referência essa situação, discorra sobre os fundamentos dos processos de desenvolvimento de *software*. Ao elaborar seu texto, aborde, necessariamente, os seguintes aspectos:

- ▶ processos e práticas ágeis de desenvolvimento de *softwares*; [valor: 4,50 pontos]
- ▶ os papéis, artefatos e ritos no *Scrum*. [valor: 5,00 pontos]

**RASCUNHO – QUESTÃO 3**

|    |  |
|----|--|
| 1  |  |
| 2  |  |
| 3  |  |
| 4  |  |
| 5  |  |
| 6  |  |
| 7  |  |
| 8  |  |
| 9  |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |