Exercício: Teste de Software

1. No contexto de *Test-Driven Development* (TDD), assinale a alternativa correta sobre o conceito de *baby steps* (passos de bebê).

    a) O uso de valores pequenos como parâmetros das funções nos testes de unidade.
    b) O crescimento gradual dos valores dos parâmetros dos testes de unidade, variando do limite inferior até o limite superior de cada tipo de dados até o teste detectar uma falha.
    c) A ideia de sempre tomar o passo mais simples que resolva o problema naquele momento e faça o teste passar.
    d) Escrever todos os testes de uma vez e depois implementar gradualmente todo o código de produção necessário.
    e) Definir inicialmente *sprints* pequenas que aumentam de tempo à medida que mais testes de unidade vão sendo escritos.

2. Assinale a opção em que é citada a principal vantagem da integração de testes de segurança (SAST e DAST) no processo de implementação de *pipeline* de CI/CD.
    a) melhoria na documentação do código
    b) identificação precoce de vulnerabilidades de segurança.
    c) redução do tempo de desenvolvimento.
    d) aumento da cobertura de testes funcionais.

3. A abordagem que se concentra principalmente em examinar as estruturas internas ou os funcionamentos de uma aplicação de *software* é denominada teste de:
    a) sistema.
    b) caixa preta.
    c) caixa branca.
    d) aceitação.
    e) caixa cinza.

4. A respeito de teste de *software*, é **INCORRETO** afirmar que:

    a) teste de regressão consiste na reaplicação de testes já aplicados, em função de posteriores alterações do *software* testado.
    b) o plano de teste de validação (de requisitos) só pode ser elaborado após a etapa de implementação.
    c) teste de integração avalia a compatibilidade entre módulos de *software* previamente testados.
    d) teste de integração avalia a compatibilidade entre módulos de *software* previamente testados.
    e) teste exaustivo corresponde à exploração de todas as possibilidades de execução de um *software.*

5. O Tribunal solicita o uso de métodos de teste de software para que o Analista de Sistemas possa derivar os casos de teste que:

− garantam que todos os caminhos independentes dentro de um módulo tenham sido exercitados pelo menos uma vez;
− exercitem todas as decisões lógicas para valores falsos ou verdadeiros;
− executem todos os laços (*loops*) em suas fronteiras e dentro de seus limites operacionais;
− exercitem as estruturas de dados internas para garantir sua validade.

Para atender a **todos** os requisitos acima, o Analista de Sistemas usa a abordagem de teste de:

a) particionamento de equivalência.
b) caixa branca.
c) análise de valor limite.
d) técnicas de grafo de causa e efeito.
e) comparação.

6. Com relação a teste de software, é correto afirmar que:

a) Teste de Stress tem caráter destrutivo, sendo utilizado para definir os valores máximos de carga que a aplicação suporta.
b) Ferramentas de acompanhamento de erros (*bug tracking*) são utilizadas para automatizar testes de performance.
c) Teste Unitário é utilizado para validar as interfaces entre os componentes e é baseado no grafo de chamadas entre estes componentes.
d) Teste de Sistema é utilizado para análise do fluxo de dados e de controle, sendo normalmente automatizado por ferramentas xUnit como JUnit e CppUnit.
e) Teste Estrutural é utilizado para medir o comportamento da aplicação em função de seus recursos e da carga gerada por um gerador de transações.

7. Considere as seguintes definições relacionadas a testes.

I. "O _____ concentra o esforço de verificação no menor elemento construtivo do projeto de *software* – o componente ou módulo de *software*. Usando a descrição do projeto em nível de componente como guia, caminhos de controle importantes são testados para descobrir erros dentro do limite do módulo. A relativa complexidade dos testes e os erros que esses testes descobrem são limitados pelo escopo restrito estabelecido. Esse teste se concentra na lógica de processamento interno e nas estruturas de dados dentro dos limites de um componente. Esse tipo de teste pode ser realizado em paralelo para vários componentes."
II. "O _____ é uma técnica sistemática para construir a arquitetura de *software* e, ao mesmo tempo, realizar testes para descobrir erros associados à interface. O objetivo é pegar componentes (unidades construtivas) testados e construir uma estrutura de programa que tenha sido definida pelo projeto."
III. "Cada vez que um novo módulo é adicionado como parte do processo de teste, o *software* muda. Novos caminhos de fluxo de dados são estabelecidos, novas entradas e saídas podem ocorrer, e uma nova lógica de controle é invocada. Os efeitos colaterais associados a essas alterações podem causar problemas com funções que anteriormente funcionavam perfeitamente. No contexto de uma estratégia de teste, o _____ é a reexecução de algum subconjunto de testes que já foram conduzidos para garantir que as alterações não tenham propagado efeitos colaterais indesejados."

Assinale a alternativa que preencha corretamente as três definições, considerando sua ordem numérica.

a) I – teste de validação; II – teste de integração; III – teste de aceitação.
b) I – teste de sistema; II – teste de validação; III – teste de estresse (*stress test*).
c) I – teste de unidade; II – teste de integração; III – teste de regressão.

d) I – teste de unidade; II – teste de estresse (*stress test*); III – teste de aceitação.
e) I – teste de sistema; II – teste de validação; III – teste de regressão.

8. Conforme o *software* evolui e suas partes são colocadas para trabalhar em conjunto, é necessário verificar se a interação entre elas ocorre da maneira mais correta possível. Os testes responsáveis por isso são chamados de:

a) Testes de integração.
b) Testes de mutação.
c) Testes de unidade.
d) Testes funcionais.
e) Testes unitários

9. Uma categoria de testes comumente utilizada é a de Testes Baseados em Experiência (Experience-based Testing). Nas técnicas dessa categoria, fatores como o histórico de funcionamento da aplicação e erros comuns de utilização das tecnologias empregadas – derivados do conhecimento do testador – são utilizados para antecipar a ocorrência de erros, defeitos e falhas. Testes baseados em experiência não costumam ser empregados como abordagem principal em cenários de alto risco, em função da variabilidade de sua eficiência e cobertura. Considerando um projeto em que o time de desenvolvimento não possua experiência com a tecnologia e o domínio da aplicação, a técnica recomendada para o portfólio de testes desse time é:

a) Teste de Caso de Uso (Use Case Testing);
b) Teste de Hipótese (Statistical Hypothesis Testing);
c) Previsão de Erros (Error Guessing);
d) Teste Exploratório (Exploratory Testing);
e) Teste Baseado em Listas de Verificação (Checklist-based Testing).

10. Dentro do contexto de Teste de Software, o objetivo de _____ é checar se o software atende a seus requisitos funcionais e não funcionais, enquanto o objetivo de _____ é checar que o software atende às expectativas do cliente.

Assinale a alternativa que completa, correta e respectivamente, as lacunas do texto acima.

a) depuração – validação.
b) verificação – depuração
c) teste de componente – depuração
d) verificação – validação
e) validação – verificação

11. Considere as seguintes afirmações sobre Teste de Software.

I - Os testes podem mostrar apenas a presença de erros, mas não sua ausência.
II - Inspeções de software (também chamadas testes de inspeção) são centradas principalmente no código-fonte de um sistema, mas qualquer representação legível do software, como seus requisitos ou modelo de projeto, pode ser inspecionada.
III- Teste unitário é o teste em que alguns ou todos os componentes de um sistema estão integrados e o sistema é testado como um todo.

Quais estão corretas?

a) Apenas I.

b) Apenas I e II.
c) Apenas I e III.
d) Apenas II e III.
e) I, II e III.

12. Abordagem para o desenvolvimento de programas em que se intercalam testes e desenvolvimento de código. Essencialmente, desenvolve-se um código de forma incremental em conjunto com um teste para esse incremento. Não se caminha para o próximo incremento até que o código desenvolvido passe no teste.

Assinale a alternativa que contém o conceito definido acima.

a) Teste de regressão.
b) Teste Unitário .
c) Desenvolvimento orientado a aspectos (Aspect Oriented Development – AOD) .
d) Desenvolvimento dirigido por testes (Test Driven Development – TDD)
e) Desenvolvimento orientado a recursos (Feature Driven Development – FDD)

13. Analise as seguintes afirmações sobre testes de sistemas de *software*.

I - Os testes funcionais são aqueles que abordam funcionalidade, corretude, completude, usabilidade e adequação à tarefa do sistema de *software*.

II - BDD (Desenvolvimento orientado a comportamento – Behavior-Driven Development) e TDD (Desenvolvimento orientado a testes – Test-Driven Development) abordam testes funcionais em nível de abstração caixa-preta e caixa-branca, respectivamente.

III- Testes estáticos não envolvem a execução do sistema em teste, mas podem ser executados automaticamente por ferramentas. Exemplos são ferramentas de análise estática de código e comparação de código com padrões.

Quais estão corretas?

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

14. Assinale as seguintes afirmações sobre teste ágil com **V** (verdadeiro) ou **F** (falso).

( ) A prática do desenvolvimento orientado a testes (TDD, do inglês Test-Driven Development) é voltada a pessoas técnicas, e tem por objetivo atingir uma alta cobertura de código com testes automatizados.
( ) A prática do desenvolvimento orientado a comportamento (BDD, do inglês Behavior-Driven Development) foca na descrição do comportamento do usuário ao usar o sistema, estimulando pessoas técnicas e do negócio a usar uma linguagem de comunicação comum.
( ) O desenvolvimento orientado a testes de aceitação (ATDD, do inglês Acceptance Test-Driven Development) combina BDD e TDD, usando cenários para refinar o entendimento de histórias, guiar o desenvolvimento e automatizar os testes de aceitação.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é

a) V – V – V.

b) F – V – V.
c) F – F – V.
d) V – F – F.
e) F – F – F.

15. Usando o Test Driven Development (TDD), é **INCORRETO** afirmar que o desenvolvedor de um sistema:

a) rapidamente escreve um novo teste para uma pequena parte da funcionalidade que é parte do requisito de usuário.
b) faz pequenas mudanças no código de produção com o objetivo de passar o teste que estava falhando.
c) refatora o código de produção e de testes com o objetivo de aprimorar a estrutura e remover redundâncias, se necessário.
d) integra os códigos de teste e de produção no final do dia.
e) baseia-se em grandes ciclos de repetição, em que em cada funcionalidade é criado um teste posterior ao desenvolvimento.

16. Assinale a alternativa correta que indica o teste relativo à reexecução do mesmo subconjunto de testes já executados anteriormente para assegurar que as alterações não tenham propagado efeitos colaterais indesejados.

a) Desempenho.
b) Repetição.
c) Regressão.
d) Carga.
e) Funcional.

17. Uma prática essencial do Test Driven Development (TDD) é o **teste de unidade**, em que o desenvolvedor cria um ou mais testes para cada unidade do sistema, como uma classe ou uma função ou um método, normalmente usando um pacote feito na mesma linguagem de programação na qual o programa é feito, como JUnit para Java. Esses testes de unidade são, então, necessariamente executados sempre que alguma modificação no código é feita, caracterizando uma outra prática de testes, que ajuda a detectar quando partes do código que estavam funcionando passam a apresentar erros após o código ter sido modificado.

Essa outra prática de testes citada é conhecida como testes de

a) Exaustão
b) Integração
c) Modificação
d) Mutantes
e) Regressão

18. O teste de software compreende um conjunto de ferramentas e técnicas relacionadas à verificação e validação (V&V) de um sistema. Em relação ao tópico de teste de software, analise as assertivas abaixo, assinalando V, se verdadeiras, ou F, se falsas.

( ) O teste beta é conduzido no ambiente de usuários reais, executando tarefas reais, sem a monitoração e interferência próxima dos desenvolvedores.
( ) O teste de aceitação é utilizado para verificar se um sistema de software como um todo é consistente com sua especificação de requisitos, geralmente executado pela equipe de testes

sem o envolvimento do usuário.
( ) Ao corrigir erros de um sistema, é muito fácil introduzir novos erros ou reintroduzir erros que ocorreram anteriormente. Nessa situação, casos de teste aprovados em versões prévias do software podem ser verificados novamente através de testes de sistema.
( ) Testes unitários em sistemas orientados a objetos normalmente realizam verificações de falhas em classes individuais.
A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

a) V – F – F – V.
b) V – V – F – V.
c) V – F – V – F.
d) F – V – F – F.
e) F – F – V – V.

19. Os requisitos de *software* são classificados como requisitos funcionais e não funcionais. De acordo com os requisitos funcionais, determinados testes são projetados para sua validação, não considerando o funcionamento interno de um programa. Assinale, a seguir, tal tipo de teste.

a) Beta.
b) Caixa preta.
c) Caixa branca.
d) Caminho básico.

20. O teste de unidade tem como finalidade testar os componentes mais simples do software

a) de forma isolada.
b) quanto à sua coesão.
c) quanto ao seu acoplamento.
d) quando unificados na versão a ser lançada.
e) quanto à sua capacidade de responder a entradas únicas.

21. Com relação ao framework pytest, da linguagem de programação Python, avalie as afirmativas a seguir:
I. O comando pytest executa os arquivos no formato test_*.py ou *_test.py no diretório corrente e nos subdiretórios.

II. O comando abaixo mostra os 5 testes com maior tempo de duração. >>> pytest -vv --durations=5

III. É possível invocar o framework pytest usando o próprio interpretador do Python por meio do comando abaixo: >>> python -m pytest
Está correto o que se afirmar em

a) I, apenas.
b) II, apenas.
c) I e II, apenas.
d) II e III, apenas.
e) I, II e III.