

---

## 114 – ANALISTA DE TI - TESTE E QUALIDADE

Data e horário da prova: Domingo, 20/01/2013, às 14h.

---

### INSTRUÇÕES GERAIS

- Você receberá do fiscal:
- um **caderno de questões** da prova objetiva contendo 50 (cinquenta) questões de múltipla escolha, com 5 (cinco) alternativas de resposta cada uma e apenas uma alternativa correta;
- um **cartão de respostas** ótico personalizado.
- Verifique se a numeração das questões e a paginação do **caderno de questões** da prova objetiva estão corretas. Quando autorizado pelo **fiscal do IADES**, no momento da identificação, escreva no espaço apropriado do **cartão de respostas**, com a sua caligrafia usual, a seguinte frase:

### A persistência é o caminho do êxito.

- Você dispõe de 4 (quatro) horas para realizar a prova objetiva, devendo controlar o tempo, pois não haverá prorrogação desse prazo. Esse tempo inclui a marcação do **cartão de respostas**.
- Somente será permitido levar o **caderno de questões**, da prova objetiva, após 3 (três) horas e 30 (trinta) minutos do início da prova.
- Somente após decorrida 1 (uma) hora do início da prova, você poderá entregar seu **cartão de respostas** ótico e retirar-se da sala.
- Após o término da prova, entregue ao fiscal o **cartão de respostas**, devidamente assinado.
- Deixe sobre a carteira apenas o documento de identidade e a **caneta esferográfica de tinta preta ou azul, fabricada de material transparente**.
- Não é permitida a utilização de qualquer aparelho eletrônico de comunicação. Desligue e acondicione na embalagem fornecida pelo **fiscal do IADES**, máquina fotográfica; telefone celular; relógio; gravador; *bip*; receptor; *pager*; *notebook*; *tablet* eletrônico; *walkman*; aparelho portátil de armazenamento e de reprodução de músicas, vídeos e outros arquivos digitais; agenda eletrônica; palmtop; régua de cálculo; máquina de calcular e (ou) qualquer outro equipamento similar.
- Não é permitida a consulta a livros, dicionários, apontamentos e apostilas.
- Você somente poderá sair e retornar à sala de aplicação de provas se sua saída for acompanhada por **fiscal do IADES**.
- Não será permitida a utilização de lápis em nenhuma etapa da prova.

### INSTRUÇÕES PARA A PROVA OBJETIVA

- Verifique se os seus dados estão corretos no **cartão de respostas**. Caso haja algum dado incorreto, escreva apenas no(s) campo(s) a ser(em) corrigido(s), conforme instruções no **cartão de respostas**.
- Leia atentamente cada questão e assinale no **cartão de respostas** a alternativa que mais adequadamente a responde.
- O **cartão de respostas** não pode ser dobrado, amassado, rasurado, manchado ou conter qualquer registro, fora dos locais destinados às respostas.
- A maneira correta de assinalar a alternativa no **cartão de respostas** é cobrindo, fortemente, com **caneta esferográfica preta** ou **azul**, o espaço a ela correspondente, conforme modelo a seguir:
- Marque as respostas **assim**: 

**Boa Prova!**

## QUESTÕES ESPECÍFICAS - QUESTÕES 26 A 50

### QUESTÃO 26

Assinale a alternativa que **não** corresponde a um dos testes de *software*, realizados em *softwares* comerciais.

- (A) Teste de sistema.
- (B) Teste de validação.
- (C) Teste de integração.
- (D) Teste de volatilidade de requisitos.
- (E) Teste de unidade.

### QUESTÃO 27

Assinale a alternativa que conceitua corretamente as características de um *software* testável.

- I - A operabilidade é uma característica, cuja documentação técnica é acessível instantaneamente, bem organizada, específica, detalhada e precisa.
- II - A controlabilidade é uma característica, cujos módulos independentes podem ser testados, independentemente.
- III - A estabilidade caracteriza o *software*, quando este se recupera bem de falhas.
- IV - A simplicidade caracteriza o *software*, projetado e implementado com qualidade

A quantidade de itens certos é igual a

- (A) 0. (B) 1. (C) 2. (D) 3. (E) 4.

### QUESTÃO 28

A atividade de teste é composta por elementos essenciais, que auxiliam na formalização desta atividade. Entre

os principais elementos, é correto citar: casos de teste, procedimentos de teste,

- (A) critérios de teste e critérios de cobertura de teste.
- (B) critérios de teste e casos de uso.
- (C) casos de uso e diagrama de teste.
- (D) casos de uso e diagrama de classes.
- (E) critérios de teste e diagrama de classes.

### QUESTÃO 29

Em relação ao teste de *software*, assinale a alternativa correta sobre o teste de integração.

- (A) Tem como objetivo explorar a menor unidade de um projeto.
- (B) Visa testar as falhas decorrentes da integração dos módulos do sistema.
- (C) Visa descobrir falhas por meio da utilização do mesmo.
- (D) Teste realizado pelos usuários finais do software.
- (E) Tem a mesma aplicação do teste de aceitação.

### QUESTÃO 30

Sobre gerenciamento de riscos de *software*, assinale a alternativa correta.

- (A) O gerenciamento de riscos de *software* não envolve os riscos ambientais, que possam afetar o projeto.
- (B) O controle de riscos é feito pelos planos de casos de uso.
- (C) O controle de riscos é realizado por membros externos ao projeto.
- (D) Gerenciamento de riscos de *software* consiste em avaliar e controlar os riscos, que afetam o projeto, processo ou produto de software.
- (E) O gerenciamento de riscos de *software* consiste, apenas, no gerenciamento dos Testes de Caixa Branca do Software.

### QUESTÃO 31

Quais são os riscos do processo de um gerenciamento de riscos de *software*?

- (A) Procedimentos e interface.
- (B) Gerenciamento e interface.
- (C) Técnicos e gerenciais.
- (D) Técnica e Interface.
- (E) Interface e rede de servidores.

## QUESTÃO 32

Assinale a alternativa que corresponde às três estratégias de depuração de *software*.

- (A) Estratégia da força bruta, estratégia do esforço e estratégia ascendente.
- (B) Estratégia da força bruta, estratégia do rastreamento e estratégia da eliminação da causa.
- (C) Estratégia do rastreamento, estratégia de integração e estratégia ascendente.
- (D) Estratégia do esforço, estratégia de integração e estratégia da eliminação da causa.
- (E) Estratégia do rastreamento, estratégia de integração e estratégia da eliminação da causa.

## QUESTÃO 33

Assinale a alternativa que **não** corresponde a uma das fases do processo de desenvolvimento, dirigido a testes (TDD).

- (A) Executar o teste, com os outros testes implementados, que rodarão e fornecerão o resultado de que o software está sem problemas.
- (B) Escrever o teste para a funcionalidade e implementação.
- (C) Realizar a identificação do incremento de funcionalidade.
- (D) Implementar a funcionalidade e executar novamente o teste.
- (E) Implementar a próxima parte da funcionalidade, após todos os testes terem sido executados, com sucesso.

## QUESTÃO 34

A fase de elaboração dos testes de *software* é uma das partes mais importantes, no desenvolvimento de um *software*. Sobre o teste de caixa branca, assinale a alternativa correta.

- (A) Teste feito pela equipe de testadores de *software*.
- (B) Teste executado pelo usuário final do *software*.
- (C) Teste realizado, na fase de concepção do *software*.
- (D) Teste executado, após a implantação do *software*.
- (E) Teste feito pelo próprio programador que verifica, se o código que foi construído, é funcional.

## QUESTÃO 35

Existem no mercado algumas metodologias de desenvolvimento, que facilitam o processo de produção de *software*. Uma dessas metodologias é o XP (*Extreme Programming*), o qual tem um cuidado especial com os processos de teste de *software*. Como é feito o processo de teste de *software*, utilizando o XP?

- (A) Todos os testes são efetuados, ao fim do desenvolvimento pois, assim, o usuário pode ter uma visão ampla do *software*.
- (B) As etapas de teste são suprimidas do processo.
- (C) Ao final de cada etapa, o usuário é convidado a testar o módulo pronto, evitando, assim, erros muito complexos, ao final do desenvolvimento.
- (D) O processo é efetuado, apenas por profissionais que trabalharam no desenvolvimento do produto, tornando assim, o teste mais eficaz e próximo da realidade do cliente.
- (E) Todos os testes são realizados na etapa de concepção do *software*.

## QUESTÃO 36

A tarefa de efetuar testes, em *software*, foi considerada secundária por muito tempo. Geralmente, era vista como castigo para o programador ou como uma tarefa, onde não se deveria gastar muito tempo e investimentos. O tema esteve relegado a segundo plano e, até alguns anos atrás, não se encontrava muita literatura sobre o assunto. Este é um paradigma que vem mudando no mundo moderno de desenvolvimento de *software*. Um dos testes, que ajudou a mudar este paradigma, é o teste de aceitação que tem como principal característica

- (A) verificar o sistema, em relação aos seus requisitos originais e às necessidades atuais do usuário.
- (B) ser realizado no início do desenvolvimento do *software*, com o objetivo de evitar futuros erros, durante o processo de desenvolvimento.
- (C) ser realizado, pelos analistas de teste, no meio do processo de desenvolvimento do produto.
- (D) ser realizado, com o cliente, apenas após a implantação do *software*.
- (E) ser realizado, com o cliente, apenas após o treinamento de uso do *software*.

## QUESTÃO 37

A atividade de teste é composta por alguns elementos essenciais, que auxiliam na formalização desta atividade. A afirmação “serve para selecionar e avaliar casos de teste, de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto”, refere-se a qual elemento da atividade de teste?

- (A) Caso de teste.
- (B) Procedimento de teste.
- (C) Critério de cobertura dos testes.
- (D) Critério de teste.
- (E) Critério de geração de casos de Teste.

SOMMERVILLE, Ian. **Engenharia de software**: 9ª Edição. São Paulo: Pearson 2011.

## QUESTÃO 38

Segundo Pressman (2011), a definição de defeito de *software* é um problema de qualidade encontrado,

- (A) somente após a liberação de uso do *software* para os usuários finais.
- (B) antes de o *software* ser liberado aos usuários finais.
- (C) na fase de revisão.
- (D) na fase de levantamento de requisitos.
- (E) na fase de prototipação.

PRESMAN, R. S. **Engenharia de Software**. São Paulo Bookman.

## QUESTÃO 39

Os testes de *software* são executados, usando os procedimentos e documentos de *script* de teste. Para que a fase de execução de teste, seja realizada com sucesso deve(m) ser executado(s)

- (A) os casos de uso.
- (B) os diagramas de atividade.
- (C) os casos de teste.
- (D) os testes de *Turing*.
- (E) o teste de COMA.

## QUESTÃO 40

Sempre que é encontrado um erro, no processo de teste de *software*, é necessário relatar o incidente. Para isso, deve-se definir os relatórios necessários, para acompanhar o progresso do projeto de teste, segundo a norma IEEE 829-1998. Os relatórios de teste que a IEEE sugere são

- (A) teste unitário, teste de sistema e teste de aceitação.
- (B) *log* de teste, incidente de teste e sumário de teste.
- (C) documento de teste, caso de teste e teste de sistema.
- (D) teste de sistema, teste de integração e casos de teste.
- (E) casos de teste, casos de usos e teste de sistema.

## QUESTÃO 41

Assinale a alternativa correta, sobre automação de teste de *software*.

- (A) É a tarefa executada, pelos analistas de teste, tendo como objetivo descrever os fluxos dos UCs do Sistema.
- (B) Tem como principal tarefa, ajudar na concepção do *Software*.
- (C) É um questionário, aplicado para os usuários finais do Sistema.

- (D) É ferramenta de instalação de *Software*.
- (E) É a utilização de um sistema, para controlar a execução dos testes de um *Software*.

## QUESTÃO 42

A automatização de *software* é um processo, em qual, de alto custo financeiro e que envolve várias etapas de teste. Alguns dos testes de *software*, aplicados nesse processo, são os testes de caixa branca e os testes de caixa preta. Assinale a alternativa correta sobre o teste de caixa preta.

- (A) É um teste de *software*, que visa aprimorar a interface homem – máquina.
- (B) É um teste de interação com o banco de dados.
- (C) É um teste de verificação, segundo o *Designer Pattern*.
- (D) Teste que verifica as saídas de dados, usando diversos tipos de entradas de dados.
- (E) Modelo de teste, adotado pelos modelos orientados a objeto.

## QUESTÃO 43

Assinale a alternativa que **não** corresponde a risco de um projeto de teste de *software*.

- (A) Ausência de cronograma detalhado do projeto de desenvolvimento, o que torna impossível elaborar um cronograma de testes.
- (B) As datas finais dependentes da execução dos testes, haja vista o risco das pressões caírem sobre a equipe de testes, causando sérios riscos na qualidade.
- (C) Escopo do projeto bem delimitado, possibilitando que se estabeleça um cronograma de testes.
- (D) Baixa qualidade da base dos testes.
- (E) A base dos testes não estar disponível, nas datas programadas.

## QUESTÃO 44

As estratégias de gerenciamento de risco são divididas em três categorias. Uma dessas categorias é a estratégia

- (A) organizacional.
- (B) de minimização.
- (C) de análise de riscos.
- (D) de monitoramento de riscos.
- (E) da automatização.

## QUESTÃO 45

Métricas de *software* são parâmetros utilizados, para mensurar ou medir algo que se queira estimar, do ponto de vista da produção de *software*. Em relação às métricas de qualidade de *software*, assinale a alternativa correta.

- (A) É uma abordagem utilizada para definir o tempo gasto, em cada ponto de função.
- (B) Oferece meios de mensurar o tempo gasto, para o desenvolvimento do *software*.
- (C) Fornece informações sobre a quantidade de linhas de código.
- (D) É uma abordagem utilizada, para reforçar os testes de aceitação.
- (E) Oferece uma estimativa de quanto o *software* se adequa às exigências implícitas e explícitas do cliente.

## QUESTÃO 46

À qual modalidade de métrica pertencem a funcionalidade, a modularidade e a manutenibilidade?

- (A) De qualidade.
- (B) Técnicas.
- (C) Produtividade.
- (D) De estabilidade
- (E) De vulnerabilidade.

## QUESTÃO 47

Durante a realização de um *checklist* de inspeção de *software*, a equipe de teste de qualidade de software, de um órgão público, deparou com uma classe de defeito de interface. Qual das afirmativas, a seguir, corresponde a verificação que deverá ser feita, para checar a existência desta classe de defeito?

- (A) Verificar se todas as constantes foram nomeadas.
- (B) Verificar se todas as variáveis de entrada são usadas.
- (C) Verificar se, para cada instrução condicional, a condição está correta.
- (D) Verificar se cada *loop* do código irá terminar.
- (E) Verificar se, todas as chamadas de função e métodos, têm o número correto de parâmetros.

## QUESTÃO 48

De acordo com o padrão de qualidade ISO 9126, são identificados seis atributos fundamentais da qualidade. Sobre o tema, assinale a alternativa correta.

- (A) A usabilidade diz respeito à quantidade de tempo, que o *software* fica disponível para uso.

- (B) A eficiência é o grau com que o *software* satisfaz às necessidades declaradas.
- (C) A disponibilidade é o grau de tempo em que o *software* permanece no ar para utilização.
- (D) A portabilidade é a facilidade com a qual um *software* pode ser transportado de um ambiente para outro.
- (E) A confidencialidade é a capacidade de manter partes do *software*, em sigilo, só sendo permitido o conhecimento, por parte de pessoas autorizadas.

## QUESTÃO 49

Gerência de Configuração de Software é um conjunto de atividades de apoio, que permite a absorção controlada das mudanças, inerentes ao desenvolvimento de software, mantendo a estabilidade na evolução do projeto. Podemos dividir o Gerenciamento de Configuração de Software em três níveis:

Gerência de Configuração		
Controle de Versão	Controle de Mudanças	Integração Contínua

(Fonte: [http://www.pronus.eng.br/artigos\\_tutoriais/gerencia\\_configuracao/gerencia\\_configuracao.php?pagNum=0](http://www.pronus.eng.br/artigos_tutoriais/gerencia_configuracao/gerencia_configuracao.php?pagNum=0). Acesso em 10/12/2012.)

Tomando por base a figura anterior, assinale a alternativa que define, corretamente, cada nível do Gerenciamento de Configuração de *Software*.

- (A) **Controle de versão:** identifica em qual versão está o *software* e quais as características de cada versão.  
**Controle de mudanças:** identifica quais foram as mudanças, efetuadas na versão.  
**Integração contínua:** tem como características testar as mudanças, assim que são realizadas.
- (B) **Controle de versão:** identifica em qual versão está o *software* e quais as características de cada versão.  
**Controle de mudanças:** tem como característica testar as mudanças, assim que são realizadas.  
**Integração contínua:** identifica quais foram as mudanças, efetuadas na versão.
- (C) **Controle de versão:** identifica quais foram as mudanças, efetuadas na versão.  
**Controle de mudanças:** tem como característica testar as mudanças, assim que são realizadas.  
**Integração contínua:** identifica em qual versão está o *software* e quais as características de cada versão.
- (D) **Controle de versão:** identifica os usuários dos sistemas.  
**Controle de mudanças:** tem como característica testar as mudanças, assim que são realizadas.  
**Integração contínua:** identifica em qual versão está o *software* e quais as características de cada versão.
- (E) **Controle de versão:** identifica os usuários dos sistemas.  
**Controle de mudanças:** tem como característica testar as mudanças, assim que são realizadas.  
**Integração contínua:** identifica as aplicações do *software*.

## QUESTÃO 50

O gerenciamento de Configuração de *Software* trabalha diretamente ligado com os *Baselines*. De acordo com essa informação, assinale alternativa correta.

- (A) É uma configuração informal, aprovada para servir de referência ao desenvolvimento posterior do sistema.
- (B) É uma configuração informal, aprovada para servir de referência ao desenvolvimento posterior do sistema, após a entrega do *software*.
- (C) São testes executados no sistema.
- (D) É um aplicativo de interação, que reforça interface com o servidor de aplicação.
- (E) É uma configuração formalmente aprovada, para servir de referência ao desenvolvimento posterior do sistema.

**RASCUNHO**