

Universidade Federal do Rio Grande do Norte

[Edital 061/2020] Processo Seletivo para Residência em Tecnologia da Informação: Área de Concentração 1: Desenvolvimento de Software

Valor da prova: 9,90



Questão 1

Com o passar dos anos, as aplicações corporativas evoluíram em sua arquitetura, saindo de um modelo monolítico executado em computadores de grande porte para um modelo em duas camadas (*two-tier*) cliente-servidor e então para um modelo contendo no mínimo três camadas (*three-tier*). Essas camadas são:

- A) apresentação, negócio e acesso a dados.
- B) visualização, lógica e negócio.
- C) domínio, negócio e acesso a dados.
- D) sistemas, processos e bancos de dados.

Valor: 0,33

Questão 2

Considere a seguinte instrução SQL no PostgreSQL.

```
CREATE VIEW visao AS SELECT 'Hello World';
```

Essa instrução está mal formulada porque:

- A) por padrão, o tipo de dado será considerado indefinido (*unknown*) e a coluna utilizará o nome padrão `?column?`.
- B) a criação de uma visão (*view*) requer a utilização da cláusula `FROM` para o nome da tabela.
- C) a criação de uma visão (*view*) requer a definição de uma função (*function*) correspondente ao nome da coluna.
- D) a criação de uma visão (*view*) requer a utilização da cláusula `WHERE` para restringir os dados.

Valor: 0,33

Questão 3

Considere o seguinte trecho de código em uma página HTML:

```
Mensagens <span class="BOOTSTRAP">{mensagens}</span>
```

Em uma página HTML com *Bootstrap* versão 4.0 ou superior, utilizando as definições de estilo (CSS) padrões do *framework*, para que a *tag span* apresente um componente *badge* em cores de fundo azul e letra branca, deve-se alterar a declaração `BOOTSTRAP` para:

- A) `badge badge-primary`
- B) `badge primary`
- C) `btn badge-primary`
- D) `btn btn-primary`

Valor: 0,33

Questão 4

No SGBD PostgreSQL, uma forma de melhorar o desempenho (*tuning*) quando da inserção de grande quantidade de registros em uma tabela é:

- A) desabilitar o *autocommit* e executar apenas um *commit* ao final das inserções de registros.

- B) criar uma nova tabela com o mesmo nome da tabela original e inserir os registros nessa nova tabela.
- C) realizar, obrigatoriamente, um commit a cada novo registro inserido.
- D) realizar, simultaneamente a cada inserção de novo registro, o *backup* da tabela.

Valor: 0,33

 **Bloco 1** - Questões de 5 a 6 

Considere o seguinte enunciado para responder às duas próximas questões.

Considere o seguinte trecho de código-fonte, implementado no *framework* Angular 2+:

```
import { Component } from '@angular/core';
@Component({
  selector: 'advogados',
  template: `
<h1>TITULO</h1>
<h2>Nome do advogado: ADOVAGADO</h2>
`
})
export class AdvogadoComponent {
  title = 'Advogados cadastrados';
  advogado = 'Nome do advogado';
}
```

 **Questão 5**  

Para que o componente em questão apresente no navegador, na declaração TITULO da tag <h1>, o texto "Advogados cadastrados" e, na declaração ADOVAGADO da tag <h2>, o texto "Nome do advogado", é necessário alterar as declarações indicadas, respectivamente, por:

- A) {{title}} e {{advogado}}
- B) {{this.title}} e {{this.advogado}}
- C) this.title e this.advogado
- D) title e advogado

Valor: 0,33

 **Questão 6**  

Assinale a alternativa que apresenta a *tag* necessária para declarar o uso do componente AdvogadoComponent em uma página HTML considerando um projeto Angular:

- A) <advogados> e </advogados>
- B) <AdvogadosComponent> e </AdvogadosComponent>
- C) <Component> e </Component>
- D) <selector> e </selector>

Valor: 0,33

Questão 7 

Utilizar o *Spring Framework* de forma associada a outras ferramentas torna a codificação em Java mais ágil. O código a seguir demonstra uma injeção de dependência no código de um DAO genérico:

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.springframework.beans.factory.annotation.*;

public class GenericDAO<T> {
  private SessionFactory sessionFactory;

  protected Session getSession() {
    return sessionFactory.getCurrentSession();
  }
}
```

A anotação do *Spring Framework* que o desenvolvedor deverá inserir antes do atributo `sessionFactory`, para que a injeção de dependência funcione corretamente, é:

- A) `@Autowired`
- B) `@Repository`
- C) `@Service`
- D) `@Component`

Valor: 0,33

Questão 8

O projeto do *Spring Framework* cujo objetivo é facilitar a criação de aplicações *stand-alone* é:

- A) *Spring Boot*.
- B) *Spring SA*.
- C) *Spring Data*.
- D) *Spring Security*.

Valor: 0,33

Questão 9

No contexto da UML (*Unified Modeling Language*), um relacionamento é uma ligação entre itens, podendo ser representado graficamente através de diferentes tipos de linhas. Associe os tipos de relacionamentos existentes na UML a suas respectivas descrições:

Relacionamentos:

- A - Associação
- B - Dependência
- C - Generalização

Descrições:

I - É um relacionamento de utilização, determinando que um item usa as informações e/ou serviços de outro item, mas não necessariamente o inverso.

II - É um relacionamento entre itens gerais e tipos mais específicos desses itens.

III - É um relacionamento estrutural que especifica objetos de um item conectados a objetos de outro item. A partir desse relacionamento, é possível navegar de um objeto de uma classe para um objeto de outra classe e vice-versa.

Assinale a alternativa que contém a associação correta:

- A) A-III, B-I, C-II
- B) A-I, B-II, C-III
- C) A-II, B-I, C-III
- D) A-III, B-II, C-I

Valor: 0,33

Questão 10

A respeito das tecnologias Java/Java EE, assinale a alternativa correta:

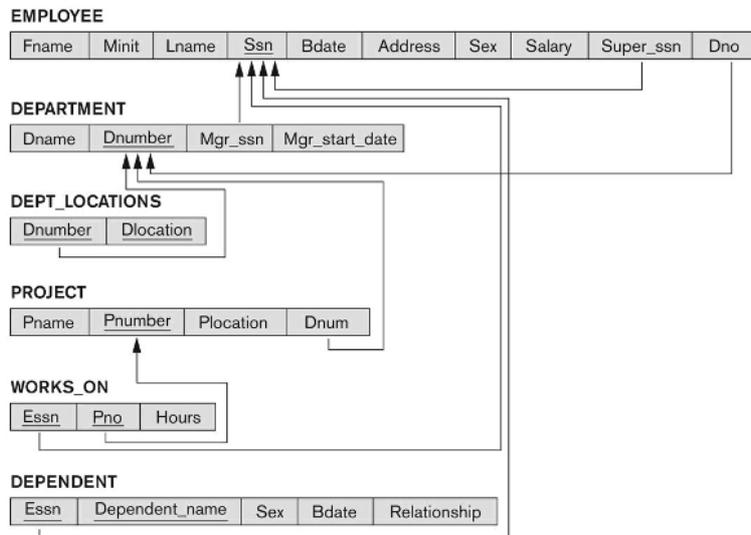
- A) EJB é uma arquitetura de componentes multiplataforma para o desenvolvimento de aplicações distribuídas, escaláveis e orientadas a objetos.
- B) JTA é uma API da linguagem de programação Java que permite a componentes baseados em Java/Java EE criar, enviar, receber e ler mensagens.
- C) JPA é uma especificação que visa padronizar o uso de transações distribuídas feitas por aplicações Java.
- D) JMS é um *framework* para desenvolver aplicações Web de forma ágil, permitindo que o desenvolvedor crie interfaces de usuário através de um conjunto de componentes predefinidos.

Valor: 0,33

 **Bloco 2** - Questões de 11 a 12 

Considere o seguinte enunciado para responder às duas próximas questões.

Considere o seguinte modelo relacional:



 **Questão 11**  

Qual das opções abaixo apresenta a consulta que lista os primeiros nomes de todos os gerentes que não possuem dependentes?

- A) `SELECT fname FROM EMPLOYEE WHERE`
`(ssn IN (SELECT mgr_ssn FROM DEPARTMENT)) AND`
`(ssn NOT IN (SELECT essn FROM DEPENDENT));`
- B) `SELECT fname FROM EMPLOYEE WHERE`
`(ssn IN (SELECT mgr_ssn FROM DEPARTMENT)) OR`
`(ssn NOT IN (SELECT essn FROM DEPENDENT));`
- C) `SELECT fname FROM EMPLOYEE WHERE`
`(ssn IN (SELECT mgr_ssn FROM DEPARTMENT)) OR`
`(ssn IN (NOT SELECT essn FROM DEPENDENT));`
- D) `SELECT fname FROM EMPLOYEE WHERE`
`(ssn NOT IN (SELECT mgr_ssn FROM DEPARTMENT)) AND`
`(ssn NOT IN (SELECT essn FROM DEPENDENT));`

Valor: 0,33

 **Questão 12**  

Qual das opções abaixo apresenta a consulta que lista os primeiros nomes (fname) e endereços (address) de todos os empregados que trabalham em pelo menos um projeto localizado em 'Natal' (plocation) mas cujo departamento do empregado (dno) não é localizado em 'Natal' (dlocation)?

- A) `SELECT fname, address FROM EMPLOYEE`
`WHERE ((ssn IN (SELECT essn FROM WORKS_ON JOIN PROJECT ON pnumber=pno`
`WHERE plocation='Natal'))`

AND

(ssn NOT IN

(SELECT ssn FROM EMPLOYEE WHERE dno IN

(SELECT dnumber FROM (DEPARTMENT NATURAL JOIN DEPT_LOCATIONS)

WHERE dlocation='Natal'))));

B) SELECT fname, address FROM EMPLOYEE

WHERE ((ssn NOT IN (SELECT essn FROM WORKS_ON JOIN PROJECT ON pnumber=pno

WHERE plocation='Natal'))

AND

(ssn NOT IN

(SELECT ssn FROM EMPLOYEE WHERE dno IN

(SELECT dnumber FROM (DEPARTMENT NATURAL JOIN DEPT_LOCATIONS)

WHERE dlocation='Natal'))));

C) SELECT fname, address FROM EMPLOYEE

WHERE((ssn IN (SELECT essn FROM WORKS_ON JOIN PROJECT ON pnumber=pno

WHERE plocation='Natal'))

AND

(ssn NOT IN

(SELECT ssn FROM EMPLOYEE WHERE dno NOT IN

(SELECT dnumber FROM (DEPARTMENT NATURAL JOIN DEPT_LOCATIONS)

WHERE dlocation='Natal'))));

D) SELECT fname, address FROM EMPLOYEE

WHERE((ssn IN (SELECT essn FROM WORKS_ON JOIN PROJECT ON pnumber=pno

WHERE plocation='Natal'))

AND

(ssn IN (SELECT ssn FROM EMPLOYEE WHERE dno IN

(SELECT dnumber FROM (DEPARTMENT NATURAL JOIN DEPT_LOCATIONS)

WHERE dlocation='Natal'))));

Valor: 0,33

Questão 13

São métodos do ciclo de vida de uma *Activity* da plataforma Android:

A) *onCreate*, *onStart*, *onResume*, *onPause*, *onStop*.

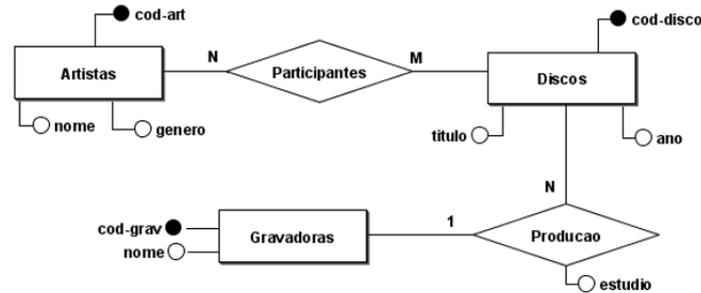
B) *onCreate*, *onStart*, *onInit*, *onPause*, *onResume*.

- B) onCreate, onStart, onStop, onPause, onResume
 C) onCreate, onStop, onSaveInstanceState, onService, onStop
 D) onCreate, onStart, onResume, onBundle, onSaveInstanceState

Valor: 0,33

Questão 14 ■

Considere o seguinte Diagrama Entidade-Relacionamento (ER) que representa um conjunto de entidades e seus relacionamentos no contexto do projeto de um banco de dados:



Um mapeamento desse diagrama para um esquema relacional descrevendo possíveis tabelas e seus respectivos campos (estando os campos chave sublinhados) seria:

- A) ARTISTAS (cod-art, nome, genero)
 PARTICIPANTES (cod-art, cod-disco)
 DISCOS (cod-disco, titulo, ano, cod-grav, estudio)
 GRAVADORAS (cod-grav, nome)
- B) ARTISTAS (cod-art, nome, genero)
 PARTICIPANTES (cod-art, cod-disco)
 DISCOS (cod-disco, titulo, ano)
 PRODUCAO (cod-grav, cod-disco, estudio)
 GRAVADORAS (cod-grav, nome)
- C) ARTISTAS (cod-art, nome, genero)
 PARTICIPANTES (cod-art, cod-disco)
 DISCOS (cod-disco, titulo, ano, cod-grav, nome, estudio)
 PRODUCAO (cod-grav, cod-disco, estudio)
 GRAVADORAS (cod-grav, nome)
- D) ARTISTAS (cod-art, nome, genero)
 DISCOS (cod-disco, titulo, ano, cod-art)
 PRODUCAO (cod-grav, cod-disco, estudio)
 GRAVADORAS (cod-grav, nome)

Valor: 0,33

Questão 15 ■

Analise as seguintes afirmativas:

I - O encapsulamento permite que uma classe defina métodos com o mesmo nome de métodos presentes em sua superclasse desde que esses métodos tenham argumentos diferentes.

II - Na linguagem de programação Java, uma instância de uma classe *C* que implementa uma interface *I* é objeto tanto do tipo definido pela interface *I* quanto do tipo definido pela classe *C*.

III - Na linguagem de programação Java, classes abstratas não precisam ser completamente abstratas, ao contrário das interfaces. Classes abstratas podem ter métodos implementados que serão herdados por suas subclasses.

A análise das afirmativas permite concluir que:

- A) apenas as afirmativas II e III estão corretas.
- B) apenas as afirmativas I e II estão corretas.
- C) apenas a afirmativa II está correta.
- D) apenas a afirmativa I está correta.

Valor: 0,33

Questão 16

Considere os seguintes trechos de código fonte implementado na linguagem de programação Java, referentes a três classes pertencentes a um mesmo pacote:

```
public abstract class C1 {
    public abstract Object criar();

    public void exibir() {
        System.out.println("Olá, mundo");
    }
}

public class C2 extends C1 {
    static int i = 0;
    int j;

    public Object criar() {
        i++;
        j = i;
        return j;
    }

    public void exibir() {
        System.out.println("j = " + j);
    }
}

public class C3 extends C1 {
    double d = 3.14;
    float f;

    public Object criar() {
        d = d + 1.0;
        f = (float) d;
        return f;
    }

    public void exibir() {
        System.out.println("f = " + f);
    }
}

public class Main {
    public static void main(String args[]) {
        C1 a = new C2();
        C1 b = new C2();
        C1 c = new C3();
        Object o1 = a.criar();
        o1 = a.criar();
        Object o2 = b.criar();
        Object o3 = c.criar();
        o3 = c.criar();
        a.exibir();
        b.exibir();
        c.exibir();
        System.out.print(" " + o1);
        System.out.print(" " + o2);
        System.out.print(" " + o3);
    }
}
```

Assinale a alternativa

que apresenta corretamente os valores impressos pela execução do programa resultante da compilação desse conjunto de classes:

A) j = 2

j = 3

f = 5.14

2 3 5.14

B) j = 2

j = 1

f = 5.14

2 1 5.14

C) Olá, mundo

Olá, mundo

Olá, mundo

2 1 5.14

D) Olá, mundo

Olá, mundo

Olá, mundo

2 3 5.14

Valor: 0,33

Questão 17

A ideia do "modelo em caixa" (*box model*) do CSS (*Cascading Style Sheet*) é de representar qualquer elemento de uma página Web com uma área retangular à qual se pode adicionar bordas e fazer ajustes em termos de tamanho e espaçamento. Uma ilustração representando esse modelo seria:



Nessa representação, as letras A, B, C e D referem-se, respectivamente, a:

- A) *border, padding, margin* e *content*.
- B) *content, margin, border* e *padding*
- C) *content, padding, border* e *margin*
- D) *padding, border, margin* e *content*

Valor: 0,33

Questão 18

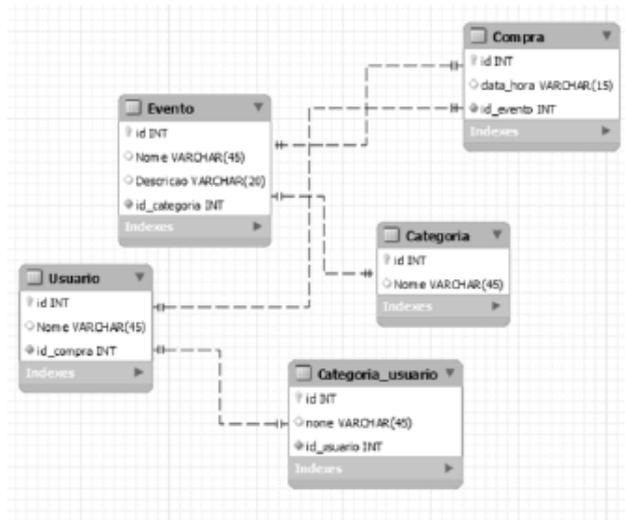
Com relação à normalização de um banco de dados é correto afirmar que:

- A) A normalização de um modelo relacional visa, principalmente, reduzir a redundância de dados aumentando a sua integridade.
- B) A normalização aplica-se a um modelo entidade-relacionamento e tem como principal função a remoção de ambiguidades.
- C) A maioria dos SGBDs atuais aplica automaticamente a normalização.
- D) A normalização de banco de dados é necessária apenas quando se busca eficiência nas consultas aos bancos de dados relacionais.

Valor: 0,33

Considere o seguinte enunciado para responder às duas próximas questões.

Considere a seguinte modelagem, referente ao banco de dados de um sistema de gerenciamento de eventos classificados por categoria. Para tais eventos, os ingressos podem ser comprados individualmente e nominalmente apenas por usuários cadastrados no sistema, restringindo-se a um único ingresso por usuário para cada evento cadastrado:



Questão 19 ✓

Considerando a modelagem do banco de dados anteriormente apresentada, analise as seguintes afirmativas sobre problemas existentes ou falta de boas práticas aplicadas nessa modelagem:

- I - O tipo utilizado na coluna data_hora da tabela Compra não é o mais apropriado.
- II - A tabela Usuario não deveria ter a coluna id_compra.
- III - A coluna Nome da tabela Categoria deveria ser uma coluna da tabela Evento.

Estão corretas as afirmativas:

- A) I, II e IV.
- B) I, II e III.
- C) I, III, IV.
- D) II, III e IV.

Valor: 0,33

Questão 20 ✓

Considerando o modelo anteriormente apresentado, analise as seguintes afirmativas:

- I - Para se visualizar para quais eventos cada usuário comprou ingresso, usa-se o seguinte comando SQL:
`SELECT * FROM evento, categoria, compra, usuario.`
- II - Para se visualizar os nomes dos usuários que compraram ingresso para o evento de ID igual a 2, usa-se o seguinte comando SQL:
`SELECT usuario.nome FROM usuario WHERE usuario.id_compra=compra.id AND compra.id_evento=2`
- III - Para se visualizar os eventos que não foram relacionados a nenhuma categoria, usamos o seguinte comando SQL:
`SELECT evento.nome FROM evento WHERE evento.id_categoria=0`
- IV - Para se visualizar a quantidade de eventos cadastrados no sistema, usamos o seguinte comando SQL:
`SELECT SUM(id) FROM evento`

Quanto às afirmativas anteriores, são **falsas**:

- A) todas as afirmativas.
- B) apenas uma afirmativa.
- C) apenas duas afirmativas.
- D) apenas três afirmativas.

Questão 21 

Padrões arquiteturais expressam formas de organizar os elementos que podem compor a arquitetura de um software, inclusive podendo auxiliar na definição dessa arquitetura pelo fato de exporem quando podem ser utilizados e documentarem suas respectivas vantagens e desvantagens. Associe os seguintes padrões arquiteturais aos cenários em que podem ser utilizados:

Padrões

- A - Cliente-servidor
- B - Tubos e filtros
- C - Camadas
- D - Model-View-Control
- E - Repositorio

Cenários

- 1) Quando há necessidade de manter uma gerência centralizada de todos os dados, de modo que estes sejam acessíveis a todos os componentes do sistema.
- 2) Em aplicações que envolvem a entrada de dados que são processados em etapas separadas, nas quais os dados fluem de um componente para outro para processamento.
- 3) Quando há possibilidade de incorporar novos requisitos não funcionais (tais como distribuição, segurança, persistência, etc.) de modo a minimizar modificações no restante do sistema em razão da integração desse novo requisito.
- 4) Quando há necessidade que os dados sejam mantidos de maneira independente de sua apresentação, de modo que possam existir diversas maneiras de visualizar e interagir com os dados.
- 5) Quando os dados compartilhados precisam ser acessados a partir de vários locais.

As associações corretas são:

- A) A-5, B-2, C-3, D-4, E-1
- B) A-1, B-3, C-4, D-2, E-5
- C) A-3, B-1, C-2, D-5, E-4
- D) A-2, B-4, C-1, D-5, E-3

Valor: 0,33

Questão 22 

Na definição de estilos com CSS (*Cascading Style Sheet*), é possível utilizar elementos, classes e identificadores. Nesse sentido, considere o seguinte trecho de código CSS com a definição de alguns estilos aplicáveis a elementos de uma página Web:

```
h1 .center {
    text-align: center;
    font-weight: bold;
}

#p1 {
    text-align: center;
    font-weight: bold;
}
```

Qual alternativa corresponde à utilização correta de estilos sobre os elementos de uma página Web?

- A) <h1 id="p1">Olá, mundo!</h1>
- B) <h1 id="center">Olá, mundo!</h1>
- C) <p class="center">Olá, mundo!</h1>
- D) <p class="p1">Olá, mundo!</h1>

Valor: 0,33

Questão 23 

A UML (*Unified Modeling Language*) é uma linguagem visual de modelagem que pode ser utilizada para visualizar, especificar, construir e documentar artefatos relacionados a um software. Em relação aos diferentes diagramas que essa notação provê, é correto afirmar:

- A) O Diagrama de Casos de Uso apresenta as funcionalidades externamente observáveis do sistema e os elementos externos com os quais

ele interage. Nesse diagrama, um elemento externo que interage com o sistema é chamado de ator, que pode representar, por exemplo, pessoas, outros sistemas e equipamentos.

- B) A UML 2.0 divide os diagramas em duas categorias básicas, a saber, diagramas estruturais e diagramas comportamentais. O Diagrama de Componentes é um diagrama comportamental que representa a topologia física do sistema, bem como os vários componentes de software de um sistema e suas dependências.
- C) O Diagrama de Máquina de Estados permite visualizar um fluxo ou processo de negócio. Ele é especialmente útil para detalhar um caso de uso que descreve um fluxo complexo envolvendo muitas partes e ações concorrentes.
- D) Um Modelo de Domínio, ilustrado como um conjunto de Diagramas de Classes, é uma representação de classes conceituais do mundo real e as restrições inerentes à tecnologia a ser utilizada na solução. É importante constarem nesse modelo os atributos e operações de cada classe.

Valor: 0,33

Questão 24

Um código fonte na linguagem de programação JavaScript é interpretado a partir de arquivos HTML que são carregadas em um navegador. Isso significa que, para que seja possível utilizar JavaScript em páginas Web, é necessário integrar o código fonte em JavaScript com o restante do código fonte em HTML. Nesse contexto, considere as seguintes afirmações:

I - Há duas formas de incluir código fonte em JavaScript em um documento HTML: embutindo o código no documento ou carregando o código a partir de um arquivo separado.

II - Para embutir um código fonte em JavaScript em um arquivo HTML, é necessário utilizar a *tag* `<script>`, colocar o código fonte e fechar a *tag* com `</script>`.

III - Outra forma de carregar código fonte em JavaScript em uma página Web é carregando um arquivo que contém esse código.

IV - Existe apenas uma forma de incluir código fonte JavaScript em um documento HTML, que é embutindo o código nele.

V - Para incluir código fonte em JavaScript em um documento HTML, basta substituir a *tag* `<html>` pela *tag* `<script>` e mudar o nome do arquivo para a extensão `.js`.

A ordem correta de julgamento das afirmações anteriores é:

- A) I - verdadeira, II - verdadeira, III - verdadeira, IV - falsa, V - falsa
B) I - verdadeira, II - falsa, III - verdadeira, IV - verdadeira, V - falsa
C) I - falsa, II - verdadeira, III - falsa, IV - verdadeira, V - verdadeira
D) I - falsa, II - verdadeira, III - falsa, IV - falsa, V - verdadeira

Valor: 0,33

Questão 25

Analise as seguintes afirmativas acerca da estrutura de aplicações na plataforma Android:

I - Uma *Task* corresponde a uma unidade de interação do usuário ou uma unidade de execução.

II - Um *Intent* representa uma descrição abstrata de uma operação que uma atividade requer que outra desempenhe.

III - Uma *Activity* representa uma cadeia de atividades que podem se estender por mais de um aplicativo.

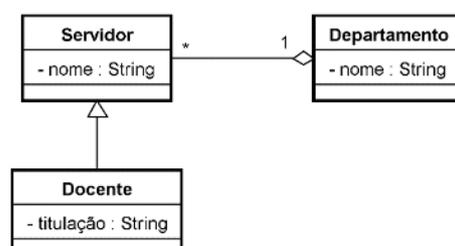
Está(ão) correta(s) a(s) afirmativa(s):

- A) II, apenas.
B) I e II, apenas.
C) I e III, apenas.
D) I, II e III.

Valor: 0,33

Questão 26

Considere o seguinte Diagrama UML de Classes:



Com base no que está representado nesse diagrama, assinale a alternativa **incorreta**:

- A) Todo departamento tem ao menos um servidor.
- B) Todo docente está associado a um departamento.
- C) Um departamento pode ter nenhum servidor associado.
- D) Um departamento pode ter nenhum docente associado.

Valor: 0,33

Questão 27

Marque a alternativa correta a respeito de padrões de projeto.

- A) Cada padrão de projeto busca resolver um problema recorrente de projeto de software, oferecendo uma solução concreta dentro de um contexto específico.
- B) Cada padrão de projeto oferece código-fonte que mostra como utilizar bibliotecas de classes diretamente na aplicação
- C) Todo padrão de projeto permite flexibilizar um determinado aspecto da aplicação.
- D) Padrões de projeto promovem o reuso de código de *frameworks* existentes.

Valor: 0,33

Questão 28

Importante conceito do paradigma de Programação Orientada a Objetos, o encapsulamento de dados tem por objetivo ocultar detalhes de implementação de um determinado módulo. Em linguagens de programação que seguem esse paradigma, tais como Java, esse ocultamento é obtido fazendo com que todos os membros (atributos e métodos) em uma classe tenham um nível particular de visibilidade com relação às suas subclasses e às classes que acessam esses membros. No que se refere aos níveis de visibilidade, assinale a alternativa correta:

- A) Um membro público é visível a qualquer classe que acessa esse membro bem como à subclasse da classe à qual ele pertence.
- B) Um membro protegido é visível somente à classe à qual ele pertence, mas não às suas subclasses ou às classes que o acessam.
- C) Um membro privado é visível somente às subclasses da classe à qual ele pertence.
- D) Um método público pode acessar somente atributos públicos declarados na classe à qual ele pertence.

Valor: 0,33

Questão 29

No que se refere à linguagem HTML:

- I - O elemento `<head>` é usado para a descrição do título do documento e pode ser subdividido em `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`.
- II - O elemento `<head>` é usado para a descrição de informações complementares sobre o documento, como, por exemplo, código de caracteres do documento e língua na qual ele está escrito.
- III - O elemento `<head>` é o local para colocar o cabeçalho de uma página no layout em HTML 5.

Dentre essas afirmações:

- A) apenas II está correta.
- B) apenas III está correta.
- C) I e II estão corretas.
- D) I e III estão corretas.

Valor: 0,33

Questão 30

Com relação aos relacionamentos possíveis em Diagramas de Classes na UML (Unified Modeling Language), considere as afirmativas a seguir:

- I - Uma agregação é uma forma mais forte de relacionamento de composição.
- II - Uma parte pode pertencer a somente um todo de cada vez.
- III - Somente uma classe no relacionamento pode representar o todo.
- IV - As partes no relacionamento de composição só existem enquanto o todo existir.

Assinale a alternativa correta.

- A) Somente as alternativas II, III e IV são corretas.
- B) Somente as afirmativas I e IV são corretas.

- C) Somente as afirmativas I e II são corretas.
- D) Somente as alternativas I, II e III são corretas.

Valor: 0,33

[← Voltar](#) [→ Próximo](#)