

**PROGRAMA DE RESIDÊNCIA EM TECNOLOGIA DA INFORMAÇÃO
TRIBUNAL REGIONAL ELEITORAL DO RIO GRANDE DO NORTE
EDITAL 001/2022 – PROVA DE CONHECIMENTOS ESPECÍFICOS**

ÁREA DE CONCENTRAÇÃO 1 – DESENVOLVIMENTO DE SOFTWARE

Candidato: _____

CPF: _____

Telefone: _____

QUESTÕES

01. 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.

02. Marque a alternativa correta a respeito de padrões de projeto.

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

03. 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.

04. 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

05. 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) 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.
- B) 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.
- 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.

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

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

07. 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 a afirmativa I está correta.

B) apenas a afirmativa II está correta.

C) apenas as afirmativas I e II estão corretas.

D) apenas as afirmativas II e III estão corretas.

08. 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-1, b-3, c-4, d-2, e-5

B) a-5, b-2, c-3, d-4, e-1

C) a-3, b-1, c-2, d-5, e-4

D) a-2, b-4, c-1, d-5, e-3

09. Quanto aos *design patterns*, é correto afirmar que:

- A) São projetos concretos que incluem detalhes de implementação.
- B) São conjuntos de classes abstratas e concretas que podem ser adaptadas e ampliadas para criar sistemas de aplicações.
- C) A UML pode ser utilizada para representar frameworks, mas não é tão ampla ao ponto de oferecer suporte à representação dos design patterns.
- D) Descrevem melhores práticas, bons projetos, e captam a experiência de uma maneira possível de ser usada por outros.

10. 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();
        C2 b = new C2();
        C3 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);
    }
}
```

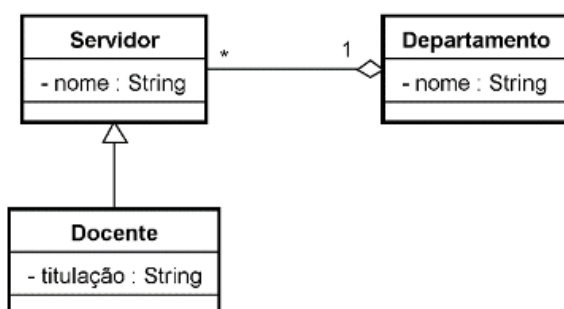
Assinale a alternativa que apresenta corretamente os valores impressos pela execução do programa resultante da compilação desse conjunto de classes:

- 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`

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

12. Considere o seguinte Diagrama UML de Classes:



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

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

13. 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) @Repository
 - B) @Autowired
 - C) @Service
 - D) @Component
-

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

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

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

16. 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-I, b-II, c-III

B) a-II, b-I, c-III

C) a-III, b-II, c-I

D) a-III, b-I, c-II

Considere o seguinte trecho de código-fonte, implementado no *framework* Angular 2+, para responder as questões 17 e 18:

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

17. Para que o componente em questão seja apresentado no navegador, na declaração TITULO da tag <h1>, o texto "Advogados cadastrados" e, na declaração ADVOGADO 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

18. 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>

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

20. 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 a um elemento HTML que tenha seu estilo definido pelo CSS descrito anteriormente?

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

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

- A) JTA é uma API da linguagem de programação Java que permite a componentes baseados em Java/Java EE criar, enviar, receber e ler mensagens.
- B) JPA é uma especificação que visa padronizar o uso de transações distribuídas feitas por aplicações Java.
- C) 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.
- D) EJB é uma arquitetura de componentes multiplataforma para o desenvolvimento de aplicações distribuídas, escaláveis e orientadas a objetos.

22. 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.

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

A) *onCreate*, *onStart*, *onUpdate*, *onPause*, *onResume*

B) *onCreate*, *onStop*, *onSaveInstanceState*, *onService*, *onStop*

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

D) *onCreate*, *onStart*, *onResume*, *onBundle*, *onSaveInstanceState*

24. O padrão de arquitetura MVC é comumente adotado para construção de aplicações *web*. Sobre este padrão considere as seguintes afirmativas:

I. A *View* exibe para o usuário os dados fornecidos pelo *Controller* através de uma interface gráfica produzida por ele mesmo.

II. Em uma aplicação Java EE, arquivos JSP são considerados componentes do tipo *Controller* do MVC, já que estes são convertidos em *servlet* quando a aplicação é compilada e implantada.

III. As classes DAO (que acessam o banco de dados) são representadas no componente *Model*.

IV. O fluxo de comunicação entre os componentes *View* e *Model* no padrão MVC deve passar obrigatoriamente pelo componente *Controller*, pois este último é o responsável pelo controle da comunicação entre os objetos.

As afirmativas corretas são:

A) II e IV

B) II e III

C) I e IV

D) I e III

25. Todas as opções abaixo retornam o mesmo elemento HTML de id "btn-principal" a partir da árvore do DOM, com exceção de:

A) `document.querySelector('#btn-principal');`

B) `document.querySelectorAll('#btn-principal')[0];`

C) `document.getElementById('#btn-principal');`

D) `document.getElementById('btn-principal');`

26. Analise o JSON retornado por um serviço web:

```
{
  "id": "0001",
  "type": "donut",
  "name": "Cake",
  "ppu": 0.55,
  "batters": {
    "batter": [
      { "id": "1001", "type": "Regular" },
      { "id": "1002", "type": "Chocolate" },
      { "id": "1003", "type": "Blueberry" },
      { "id": "1004", "type": "Devil's Food" }
    ]
  },
  "topping": [
    { "id": "5001", "type": "None" },
    { "id": "5002", "type": "Glazed" },
    { "id": "5005", "type": "Sugar" },
    { "id": "5007", "type": "Powdered Sugar" },
    { "id": "5006", "type": "Chocolate with Sprinkles" },
    { "id": "5003", "type": "Chocolate" },
    { "id": "5004", "type": "Maple" }
  ]
}
```

Após converter para um objeto JavaScript chamado **objeto**, com qual chamada retornaríamos o valor armazenado com valor "Chocolate"?

- A) objeto.batters[0].batter[1].type
- B) objeto.batters.batter[1].type
- C) objeto.batters.batter.id["1002"]
- D) objeto[4].batter[1].type

27. 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:

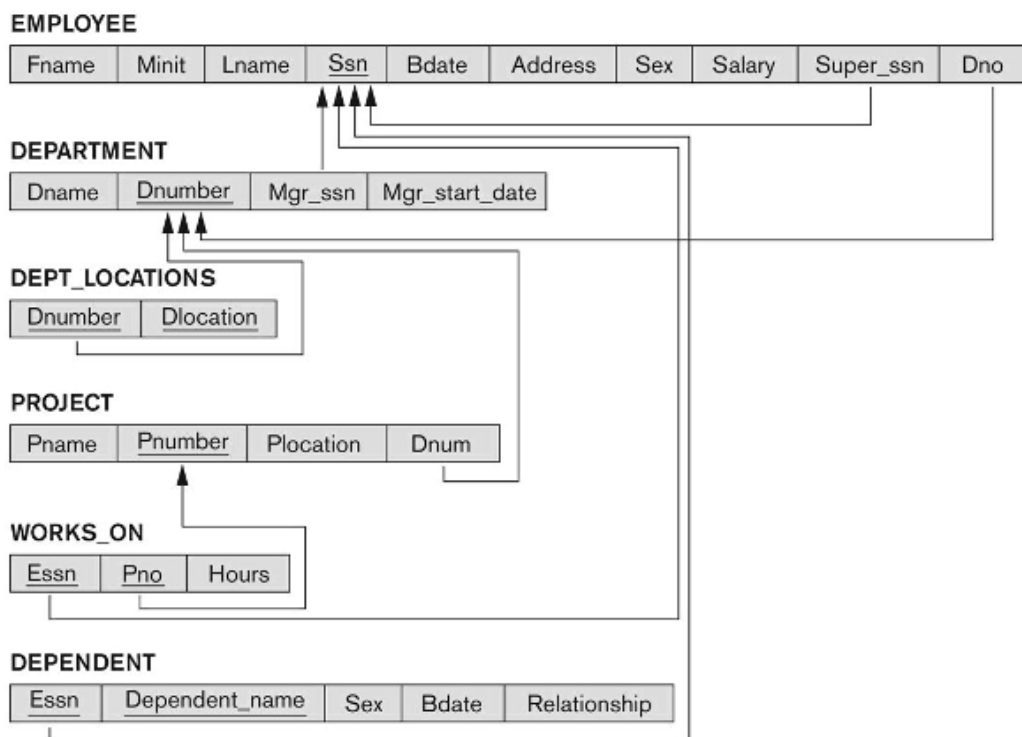
- A) apenas II está correta.
- B) apenas III está correta.
- C) I e II estão corretas.
- D) I e III estão corretas.

28. Modificações são inevitáveis quando um *software* é construído. Um objetivo primordial da Engenharia de *Software* é otimizar a forma como as modificações podem ser acomodadas e reduzir a quantidade de esforço despendido quando elas estiverem sendo feitas. Uma das ferramentas existentes para dar suporte ao processo de Gestão de Modificações é a ferramenta GIT. Acerca dessa ferramenta e considerando o comando abaixo, assinale a alternativa correta.

```
git push -u origin master
```

- A) O comando acima cria uma cópia de trabalho/repositório na máquina onde esse comando foi executado.
- B) O comando acima envia cópia para o repositório local das alterações feitas em um repositório remoto.
- C) O comando acima envia revisões locais para outro repositório remoto.
- D) O comando acima envia alterações para o repositório local, criando uma revisão.

Considere o seguinte modelo relacional para as questões 29 e 30:



29. 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)) OR
(ssn NOT IN (SELECT essn FROM DEPENDENT));
```
- B)

```
SELECT fname FROM EMPLOYEE WHERE
(ssn IN (SELECT mgr_ssn FROM DEPARTMENT)) AND
(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));

30. 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 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'))));

B) 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'))));

C) 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'))));

D) 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'))));