



MINISTÉRIO DA EDUCAÇÃO

INSTITUTO FEDERAL DO ESPÍRITO SANTO

REITORIA

Avenida Rio Branco, 50 – Santa Lúcia – 29056-255 – Vitória – ES

27 3357-7500

CONCURSO PÚBLICO

Edital nº 2/2015

Professor de Ensino Básico, Técnico e Tecnológico

Caderno de Provas Questões Objetivas

Ciência da Computação – Índice 205

Instruções

- 1 - Aguarde autorização para abrir o CADERNO DE PROVAS.
- 2 – Confira se o seu CADERNO DE PROVAS contém todas as questões. Caso o caderno esteja incompleto, tenha qualquer defeito ou apresente divergência, comunique ao fiscal da sala para que seja substituído.
- 3 - A prova terá duração máxima de 4 (quatro) horas, não podendo o candidato retirar-se com a prova antes que transcorram 2 (duas) horas do seu início.
- 4 - A prova é composta de 50 (cinquenta) questões objetivas.
- 5 - Verifique se os seus dados estão corretos no CARTÃO RESPOSTA, caso haja alguma divergência, comunique-a imediatamente ao fiscal da sala. Após a conferência, escreva e assine seu nome nos espaços próprios do CARTÃO RESPOSTA.
- 6 - As respostas às questões objetivas deverão ser assinaladas no CARTÃO RESPOSTA. Lembre-se de que para cada questão objetiva há APENAS UMA resposta.
- 7 - O CARTÃO RESPOSTA deverá ser marcado, obrigatoriamente, com caneta esferográfica (tinta azul ou preta).
- 8 - Não dobre, não amasse e nem rasure o CARTÃO RESPOSTA, pois ele não poderá ser substituído.
- 9 - A interpretação dos enunciados faz parte da aferição de conhecimentos. Não cabem, portanto, esclarecimentos.
- 10 – Não será permitido o uso de materiais impressos ou quaisquer equipamentos eletrônicos, tais como telefones celulares, notebooks, calculadoras ou similares, no período destinado à prova.
- 11 - O Candidato deverá devolver ao Fiscal o CARTÃO RESPOSTA, ao término de sua prova.

PROVA DE CONHECIMENTOS ESPECÍFICOS

01 Com relação às linguagens de programação, pode-se afirmar que aquela que dominou a computação científica nos últimos 50 anos foi:

- a) Lisp
- b) C++
- c) Python
- d) Fortran
- e) Pascal

02 Em 1964, na Universidade de Dartmouth, foi criada a linguagem de programação BASIC. Entre os requisitos propostos no projeto dessa linguagem encontram-se, **EXCETO**:

- a) ser uma linguagem agradável e amigável;
- b) permitir acesso público e privado;
- c) priorizar o tempo do computador em detrimento do tempo do programador;
- d) ser de fácil aprendizado, principalmente por estudantes de áreas não científicas;
- e) ser rápida para resolver exercícios.

03 Dentre as características de uma variável em um programa, aquela que define o intervalo de possíveis valores que uma variável pode armazenar é denominada:

- a) endereço;
- b) escopo;
- c) valor;
- d) tempo de Vida;
- e) tipo.

04 Ao se projetar uma linguagem de programação, é necessário definir de antemão como ela irá manipular cadeias de caracteres (*strings*). Portanto, uma das questões de projeto de uma linguagem é sobre o tamanho das *strings* (tamanho estático ou tamanho dinâmico). Algumas linguagens permitem a utilização de *strings* de tamanho dinâmico, mas com um tamanho máximo definido. Nesses casos, cada *string* possui um descritor em tempo de execução que contém os seguintes campos, **EXCETO**:

- a) nome do tipo;
- b) escopo do tipo;
- c) tamanho atual da *string*;
- d) tamanho máximo da *string*;
- e) endereço do primeiro caractere da *string*.

05 Suponha a existência de um programa escrito na linguagem Java que possua uma expressão lógica. Pode-se afirmar que essa expressão terá uma avaliação curto-circuito quando:

- a) A avaliação da expressão contiver erros semânticos.
- b) O resultado não for do tipo esperado.
- c) O resultado puder ser determinado sem que todos os operandos e/ou operadores sejam avaliados.
- d) A avaliação da expressão não corresponder ao desejado.
- e) O resultado puder ser determinado sem considerar a ordem de precedência dos operadores.

06 Nas linguagens C, C++ e Java a estrutura de controle de seleção múltipla é o:

- a) *switch*
- b) *elsif*
- c) *case*
- d) *break*
- e) *goto*

07 Uma questão importante no projeto de uma linguagem de programação é a definição de subprogramas, que por sua vez demandam a definição do método de passagem de parâmetros. Os métodos de passagem de parâmetros conhecidos são, **EXCETO**:

- a) passagem de parâmetro por valor
- b) passagem de parâmetro por valor-resultado
- c) passagem de parâmetro por nome
- d) passagem de parâmetro por tipo
- e) passagem de parâmetro por referência

08 A linguagem de programação Java permite a definição de tipos abstratos de dados, porém, diferentemente de outras linguagens que também permitem tal definição, o Java não tem métodos destrutores. Assinale a característica que lhe permite não ter tais métodos.

- a) paradigma de programação orientada a objetos
- b) alocação dinâmica de variáveis
- c) sobrecarga de operadores
- d) passagem de parâmetro por referência
- e) coleta de lixo implícita

09 Considerando os diferentes paradigmas de programação, assinale a alternativa que contém as linguagens que representam, respectivamente, os paradigmas orientado a objetos, funcional e lógico.

- a) Java, Pascal, Prolog
- b) Java, Haskell, Pascal
- c) Pascal, Prolog, Java
- d) Java, Haskell, Prolog
- e) Pascal, Prolog, Haskell

10 Certo programador deseja criar um programa simples no qual é informado o valor total de vendas da mercearia do Sr. Joaquim. O programa exibe o valor dos impostos a serem pagos, que corresponde a 37% do valor total de vendas. Antes disso, o programador deseja armazenar o valor calculado em uma variável e está em dúvida sobre a escolha do nome. Abaixo estão as opções. A que **NÃO** representa um nome possível de ser utilizado é:

- a) p
- b) p37
- c) porcentagem_imposto
- d) calculodoimpostodosrjoaquim
- e) p%37

11 Considere a seguinte expressão aritmética que faz parte de um programa, onde \square representa o comando de atribuição do valor da expressão à variável x.

$$x \square A+B-C/(A+B);$$

Supondo que as variáveis numéricas A, B e C possuam respectivamente os valores 2, 3 e 10, pode-se afirmar que, após a avaliação da expressão, a variável x terá o seguinte valor:

- a) 2
- b) 3
- c) -1
- d) 10
- e) 5

12 João precisa criar um programa na linguagem C que utilize uma estrutura de repetição. No entanto, João está em dúvida sobre qual estrutura deve usar. Ele sabe que o número de repetições é constante e, para não ter maiores problemas, deseja que a variável de controle seja incrementada ou decrementada automaticamente. A estrutura que João deve utilizar é:

- a) *if-else*
- b) *while*
- c) *for*
- d) *switch*
- e) *do-while*

13 Considere a seguinte expressão, que faz parte de um programa na linguagem Pascal.

$$Y:= X > A+B/C;$$

Supondo que as variáveis numéricas X, A, B e C possuam respectivamente os valores 1.8, 2, 3 e 10 e que avaliação da expressão não retornou qualquer tipo de erro, pode-se afirmar que a variável Y é do tipo:

- a) *integer*
- b) *boolean*
- c) *char*
- d) *real*
- e) *string*

14 Uma das maiores vantagens de se utilizar linguagens de programação estruturadas é a possibilidade de utilização de variáveis compostas. Dentre estas, as variáveis compostas homogêneas se destacam por serem bastante utilizadas em diversas aplicações. Sobre sua utilização pode-se afirmar, **EXCETO**:

- a) Os dados armazenados nessas variáveis são do mesmo tipo.
- b) Cada elemento ocupa uma posição definida.
- c) Na linguagem Pascal, as variáveis do tipo *array* são variáveis compostas homogêneas.
- d) Os elementos são identificados pelo mesmo nome, mas possuem índices diferentes.
- e) Somente é possível acessar um elemento após ter acessado todos os elementos antecessores.

15 Quais valores são impressos quando o seguinte algoritmo, escrito em Pascal, é executado?

```
Program P;  
var a, b : integer;  
Procedure Proc(x:integer; var y:integer);  
begin  
    x:=y+b+1;  
    y:=x+a+1;  
end;  
begin  
    a := 1; b := 2;  
    Proc(a,b);  
    write(a,b);  
end.
```

- a) 5 7
- b) 1 7
- c) 3 5
- d) 1 4
- e) 1 5

16 Um estudante desenvolveu o algoritmo a seguir em Pascal. A partir de sua análise pode-se afirmar que o comando *writeln* será executado quantas vezes?

```
Program P;  
Procedure H(x:integer; a, b, c:char);  
begin  
    if (x > 0)  
    begin  
        H(x-1,a,c,b);  
        writeln(a, ' --> ', c);  
        H(x-1,b,a,c);  
    end;  
end;  
begin  
    H(4, 'A', 'B', 'C');  
end.
```

- a) 2
- b) 8
- c) 15
- d) 7
- e) 4

17 Considere o seguinte trecho de algoritmo em Pascal.

```
...
const
  n = 5;
type
  vectortype = array[1..n] of integer;
Procedure Ordena(var a:vectortype);
var i, j, t : integer;
begin
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if(a[i] < a[j]) then
        begin
          t := a[i];
          a[i] := a[j];
          a[j] := t;
        end;
    end;
end;
...
```

Qual será o valor, respectivamente, do primeiro e do último elemento do *array* abaixo após a execução do procedimento Ordena?

| | | | | |
|---|---|---|---|---|
| 3 | 5 | 2 | 1 | 4 |
|---|---|---|---|---|

- a) 1 4
- b) 5 1
- c) 4 3
- d) 3 5
- e) 1 5

18 Quais valores são impressos quando o seguinte algoritmo, escrito em Pascal, é executado?

```
Program P;
var y : integer;
Procedure Proc(var z:integer);
var x : integer;
begin
  x:=y+z;
  z:=x+y+1;
end;
var x : integer;
begin
  x := 2; y := 3;
  Proc(y);
  write(x,y);
end.
```

- a) 2 10
- b) 6 3
- c) 2 3
- d) 2 6
- e) 6 10

19 No paradigma de programação orientada a objetos existe a possibilidade de reutilização de *software*, pois uma nova classe poderá absorver membros de uma classe existente e ser aprimorada com capacidades novas ou modificadas. O nome desse mecanismo é:

- a) polimorfismo
- b) *thread*
- c) encapsulamento
- d) multiplicidade
- e) herança

20 Os membros de uma classe (atributos e operações), em programação orientada a objetos, podem ser protegidos, privados ou públicos. Suponha que exista um método de determinada classe que só deve ser executado por instâncias dessa mesma classe. Assinale a alternativa que **MELHOR** descreve a visibilidade desse método.

- a) privado ou protegido
- b) privado ou público
- c) somente público
- d) somente privado
- e) somente protegido

21 O conceito de multiplicidade em programação orientada a objetos é considerado de extrema importância na modelagem de classes. Por isso, é altamente recomendável especificar a multiplicidade dos relacionamentos (associações). Um dos tipos mais comuns de multiplicidade é a “um-para-muitos” (1:n). Entre as alternativas, assinale a que representa uma situação desse tipo de multiplicidade. Considere a notação “associação(classe1, classe2)”.

- a) Visitar(Turista, País)
- b) Votar(Presidente, Eleitor)
- c) Casar(Marido, Esposa)
- d) Estudar(Aluno, Disciplinas)
- e) Compor(Música, Compositor)

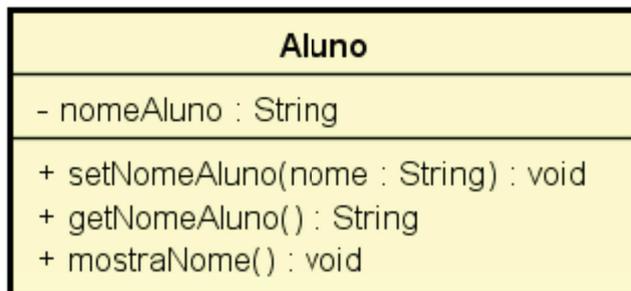
22 Na linguagem de programação Java é possível implementar alguns benefícios de herança múltipla através da definição de uma classe abstrata, também denominada:

- a) sobrescrita;
- b) acoplamento;
- c) interface;
- d) superclasse;
- e) final.

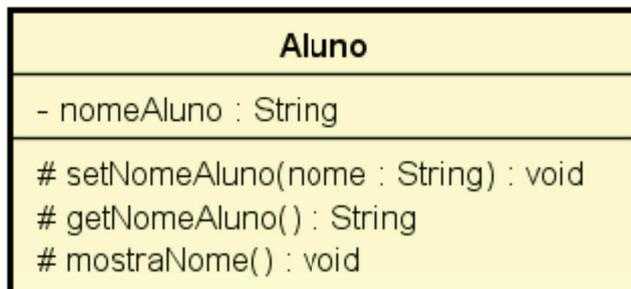
23 Observe o código abaixo escrito em linguagem Java e marque a alternativa que contém o diagrama de classes da UML que o representa corretamente.

```
public class Aluno {
    private String nomeAluno;
    public void setNomeAluno (String nome) {
        nomeAluno = nome;
    }
    public String getNomeAluno () {
        return nomeAluno;
    }
    public void mostraNome() {
        System.out.printf("Nome do Aluno\n%s!\n ", getNomeAluno());
    }
}
```

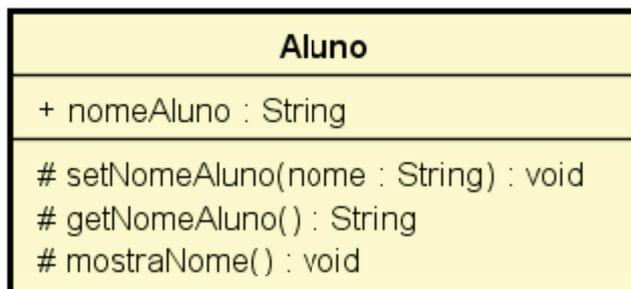
a)



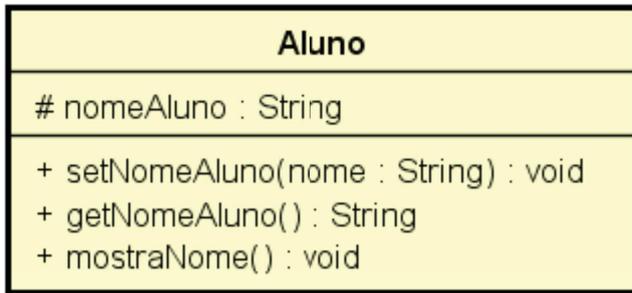
b)



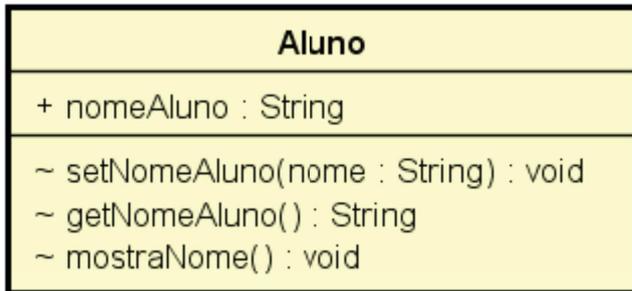
c)



d)



e)



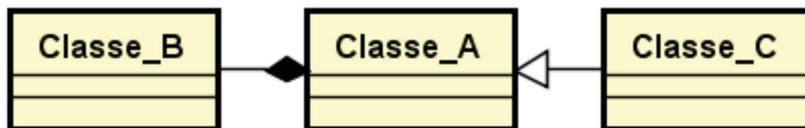
24 Sobre o conceito de herança em programação orientada a objetos, analise as afirmativas a seguir marque a opção **INCORRETA**.

- a) Uma das vantagens da implementação de orientação a objetos em linguagem de programação Java é o suporte à herança múltipla.
- b) Herança é uma forma de reutilização de *software* na qual uma nova classe é criada observando membros de uma classe existente e aprimorada com capacidades novas ou modificadas.
- c) A notação *@override* indica que um método de uma subclasse sobrescreve o método da superclasse.
- d) A notação *extends* indica que a subclasse herda as características da superclasse.
- e) Os membros da superclasse *private* não são acessíveis fora da própria classe. Em vez disso, eles permanecem ocultos nas suas subclasses e só podem ser acessados pelos métodos *public* ou *protected* herdados das superclasses.

25 O paradigma de orientação a objetos busca, no mundo real, os seus elementos-chave: classes e objetos. As classes definem as características, comportamentos e relacionamentos de seus objetos que, por sua vez, são elementos existentes no mundo real. Marque a afirmativa **CORRETA**.

- a) A orientação a objetos define dois tipos de classe: as classes concretas, que definem objetos concretos do mundo real, como, por exemplo, um produto que é vendido em uma loja; e as classes abstratas, que representam os objetos abstratos do mundo real, como, por exemplo, um registro de vendas.
- b) A orientação a objetos modela a comunicação entre os objetos. Diz-se, nesse paradigma, que dois objetos devem se comunicar por meio da alteração do estado de um deles diretamente pelo outro.
- c) O projeto orientado a objetos encapsula atributos e operações em objetos e a comunicação é feita por meio de interfaces. Os detalhes de implementação são ocultados dentro do próprio objeto.
- d) Em linguagens de programação orientadas a objetos, como Java e C, a unidade de programação é a classe, a partir da qual os objetos são instanciados.
- e) Empacotar *software* como classes possibilita que sistemas futuros herdem as classes. A eficiência da herança entre classes em projetos orientados a objetos depende diretamente da qualidade do projeto de pacotes do sistema.

26 Observe o Diagrama de Classes da UML abaixo e marque a afirmativa que o descreve corretamente.



- a) A Classe_B é uma parte que compõe a Classe_A, enquanto a Classe_C é uma generalização da Classe_A.
- b) A Classe_B é uma especialização da Classe_A, enquanto a Classe_A é uma parte que compõe a Classe_C.
- c) A Classe_B é uma generalização da Classe_A, enquanto a Classe_A é uma parte que compõe a Classe_C.
- d) A Classe_B é uma parte que compõe a Classe_A, enquanto a Classe_C é uma especialização da Classe_A.
- e) A Classe_B é uma generalização da Classe_A, enquanto a Classe_C é uma especialização da Classe_A.

27 Observe o código abaixo escrito em linguagem Java e marque a alternativa que o descreve corretamente.

```
import java.util.Random;
public class ImplementaThread implements Runnable {
    private final int tempoAdormece;
    private final String nomeTarefa;
    private final static Random randomico = new Random();
    public ImplementaThread(String nome)
    {
        nomeTarefa = nome;
        tempoAdormece = randomico.nextInt(1000);
    }
    @Override
    public void run()
    {
        try
        {
            Thread.sleep(tempoAdormece);
        }
        catch (InterruptedException exception)
        {
            System.out.printf("%s %s\n", nomeTarefa, "interrompida");
        }
        System.out.printf("%s tarefa acordada\n", nomeTarefa);
    }
}
```

- a) A capacidade de implementação de *multithreading* não está disponível na linguagem Java e deve ser realizada por meio de bibliotecas que forneçam essa funcionalidade.
- b) A diretiva *Thread.sleep(tempoAdormece)* fará com que a *thread* entre no estado de espera sincronizada até completar o *tempoAdormece* ou até que haja um processador disponível para executá-la.
- c) A implementação de *multithreading* só acontece de fato em computadores com múltiplos processadores, pois, em computadores com um processador, apenas uma tarefa é executada em um determinado instante.
- d) A implementação de *multithreading* permite que várias instâncias da classe *ImplementaThread* sejam criadas, entretanto apenas uma estará em execução em um determinado instante.
- e) Os objetos da classe *ImplementaThread* poderão ser executados concorrentemente com outras tarefas, pois a classe implementa a interface *Runnable*.

28 Todas as empresas precisam tomar decisões eficazes e, na maioria dos casos, o planejamento estratégico e os objetivos gerais da organização estabelecem o curso para essa ação. Com frequência, sistemas de informação servem de apoio na tomada de decisão. Sobre esse tipo de sistema, analise as afirmativas abaixo e marque a opção **INCORRETA**.

- a) Um modelo otimizador de Sistema de Apoio a Decisão encontra a melhor solução, normalmente baseado em um parâmetro limitador, como, por exemplo, tempo, espaço ou orçamento.
- b) Um modelo satisfatório de Sistema de Apoio a Decisão encontra um conjunto de soluções viáveis após analisar computacionalmente todas as possíveis soluções.
- c) Sistemas de Apoio a Decisão podem ser implementados com heurísticas. Nesse caso, são estabelecidas orientações ou procedimentos que encontram uma boa solução por proximidade a um objetivo ou a um conjunto de regras.
- d) Conceitualmente, Sistemas de Apoio a Decisão e Sistemas de Informação Gerencial se diferenciam, dentre outros fatores, pelo tipo de problema que objetivam resolver: enquanto o primeiro lida com problemas não estruturados, o outro normalmente é utilizado para a solução de problemas estruturados.
- e) Um Sistema de Apoio a Decisão baseado em modelo realiza principalmente análise matemática ou quantitativa, enquanto o Sistema de Apoio a Decisão baseado em dados permite que os decisores realizem uma análise qualitativa no depósito de dados disponível.

29 O modo como um Sistema de Informação é projetado, implantado e mantido afeta profundamente o funcionamento diário de uma organização e visa alcançar as metas organizacionais, como redução de custo, aumento do lucro e da qualidade do serviço prestado. Sobre as etapas do processo de desenvolvimento de sistemas, analise as afirmativas abaixo e marque a opção **CORRETA**.

- a) O objetivo geral da etapa de projeto é determinar as necessidades funcionais do usuário em relação ao novo sistema.
- b) Durante a etapa de projeto da abordagem orientada a objetos de desenvolvimento de sistemas, a sequência de eventos de um determinado cenário pode ser modelada utilizando o diagrama de classes da UML.
- c) O objetivo da etapa de coleta de dados é buscar informações sobre o problema ou a necessidade, a começar pela identificação e localização das várias fontes de dados internas ou externas.
- d) A etapa de análise deve responder à questão sobre como um sistema de informação resolverá um problema.
- e) A observação direta do sistema existente caracteriza-se como um vício de desenvolvimento e deve ser evitada a fim de que os erros do sistema atual não sejam transferidos para o novo sistema.

30 Um sistema empresarial é central para uma organização e garante que as informações possam ser compartilhadas por todas as funções da empresa e por todos os níveis de gerência, para apoiar a administração e o gerenciamento de um negócio. Sobre os sistemas de informação empresariais, analise as afirmativas abaixo e marque a opção **INCORRETA**.

- a) Além de servir como ferramenta das atividades operacionais das empresas, os Sistemas de Processamento de Transações servem como principal fonte de informação para os seus Sistemas de Informações Gerenciais.
- b) Uma forma de Sistemas de Processamento de Transações operarem é por meio de processamento de transações *on-line* (OLTP), no qual as transações são operadas remotamente através da internet.
- c) Em um Sistema de Transações típico os dados passam por um ciclo de processamento de transações que incluem coleta, edição, correção, manipulação e armazenamento dos dados e produção de resultados.
- d) Uma forma de Sistemas de Processamento de Transações operarem é por meio de processamento em lote, no qual as transações de negócio são acumuladas por um período de tempo e preparadas para processamento como uma unidade.
- e) Sistemas de Processamento de Transação são caracterizados pelo alto volume de entrada e saída e baixo volume de processamento de dados.

31 O *software* é indispensável a qualquer sistema computacional e a seus usuários. Há software de sistema, que coordena as atividades do *hardware*, e *software* de aplicação, que auxilia os usuários a resolver problemas específicos de computação. Analise as afirmativas abaixo sobre conceitos de *software* e marque a opção **CORRETA**.

- a) A maioria dos sistemas operacionais mantém memória virtual, que aloca espaço em disco rígido para suplementar a capacidade funcional da memória RAM. A memória virtual trabalha com o conceito de paginação e cada página virtual corresponde a um programa em execução na memória real.
- b) A utilização de um provedor de serviço de aplicação (ASP - *application service provider*) é uma solução voltada para grandes empresas, fisicamente distribuídas, que demandam serviço de rede eficiente, entretanto é pouco recomendada para pequenas e médias empresas devido ao alto custo de manutenção.
- c) O componente gerenciador de arquivos do sistema operacional é o responsável por controlar a alocação de espaços na memória principal para os aplicativos que manipulam arquivos, como, por exemplo, um editor de textos.
- d) A arquitetura orientada a objetos é especialmente indicada para projetar sistemas reutilizáveis, enquanto a arquitetura orientada a serviço é especialmente indicada para projetar sistemas ad hoc.
- e) Uma arquitetura orientada a serviço utiliza serviços de aplicação modular a fim de permitir que os usuários interajam com os sistemas e que os sistemas interajam entre si.

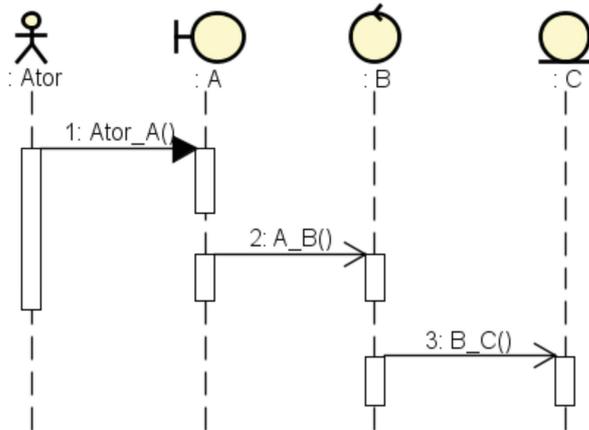
32 O Modelo Entidade-Relacionamento (ER) é um modelo de dados conceitual utilizado no projeto de bancos de dados tradicionais. Com a introdução de novas tecnologias dos bancos de dados, fez-se necessária sua representação por meio de um modelo que reflita as propriedades de dados e restrições de forma mais precisa, surgindo, assim, o modelo Entidade-Relacionamento Estendido (EER). Analise as afirmativas abaixo sobre o modelo EER e marque a opção **INCORRETA**.

- a) No modelo EER é possível modelar um tipo de entidade que possui diversos subtipos significativos. O conjunto de entidades de um subtipo pertence também ao conjunto de entidades do tipo descrito.
- b) Em alguns casos, os subtipos são definidos pelo valor de algum atributo da superclasse, denominado atributo de definição.
- c) Uma especialização $Z = \{S_1, S_2, \dots, S_n\}$ é um conjunto de subclasses que têm a mesma superclasse G , ou seja, G/S_i é um relacionamento superclasse/subclasse para $i = 1, 2, \dots, n$.
- d) Cada tipo pode possuir um ou mais subtipos e cada subtipo está associado a um e somente um supertipo.
- e) Uma categoria é uma subclasse com mais de uma superclasse que representa diferentes tipos de entidade.

33 O ciclo de vida de um sistema inclui uma série de fases distribuídas ao longo do tempo, conforme o paradigma de desenvolvimento adotado pela equipe. As afirmativas abaixo descrevem algumas dessas fases. Marque a **INCORRETA**.

- a) A fase operacional inicia-se quando todas as funções do sistema estão em funcionamento e foram validadas.
- b) A fase de projeto possui dois aspectos fortemente relacionados: o projeto do sistema de banco de dados e o projeto do sistema de aplicação, que usará e processará os dados armazenados no banco de dados.
- c) A análise da viabilidade econômica e o cálculo de custo-benefício devem ser feitos anteriormente à análise dos requisitos, mesmo que os requisitos sofram mudanças nessa fase.
- d) Novos requisitos podem surgir ao longo do desenvolvimento do sistema e, à medida que vão surgindo, todas as fases anteriores devem ser refeitas, levando em consideração o impacto do novo requisito.
- e) Na fase de validação e teste de aceitação é feito o teste de unidade e de integração do sistema. Implementação de código, integração e desempenho são exemplos de elementos testados nessa fase.

34 Com base no diagrama da UML abaixo, analise as afirmativas que seguem e marque a opção que o descreve **CORRETAMENTE**.



- É um diagrama de objetos da UML e descreve a sequência de envio de mensagem de um objeto para outro.
- É um diagrama de sequência da UML e descreve a sequência temporal do envio de mensagem de um objeto para outro.
- É um diagrama de objetos da UML e descreve a sequência de ativação de um objeto por outro.
- É um diagrama de atividades da UML e descreve a sequência temporal do envio de mensagem de um objeto para outro.
- É um diagrama de sequência da UML e descreve a sequência de ativação de um objeto por outro.

35 O processo de normalização de banco de dados leva um esquema de relação por uma série de testes para verificar se ele satisfaz certa forma normal. Sobre o processo de normalização, leia as afirmativas abaixo e marque a opção **INCORRETA**.

- A primeira forma normal (1FN) preconiza que o domínio de um atributo está na 1FN se seu conteúdo for atômico. Caso um atributo seja multivalorado e o número máximo de valores seja conhecido, pode-se desdobrar o atributo multivalorado em tantos atributos quantos necessário para atender ao número máximo de valores possíveis.
- Um esquema de relação R está na segunda forma normal (2FN) se ele satisfizer a 1FN e cada atributo não principal A em R for total e funcionalmente dependente da chave primária R .
- Um esquema de relação R está na terceira forma normal (3FN) se ele satisfizer a 2FN e nenhum atributo não principal de R for transitivamente dependente da chave primária.
- Um esquema de relação R está na forma normal de *Boyce-Codd* (FNBC) se toda vez que uma dependência funcional não trivial $X \rightarrow A$ se mantiver em R , então X é uma superchave de R .
- Na prática, a maioria dos esquemas de relação que estão na 3FN também estão na FNBC. A exceção se dá somente se $X \rightarrow A$ se mantiver em um esquema de relação R com X não sendo uma superchave e A sendo um atributo principal.

36 Observe o *script* de criação de um banco de dados *MySQL* abaixo e em seguida marque a opção que contém o diagrama que o representa corretamente.

```

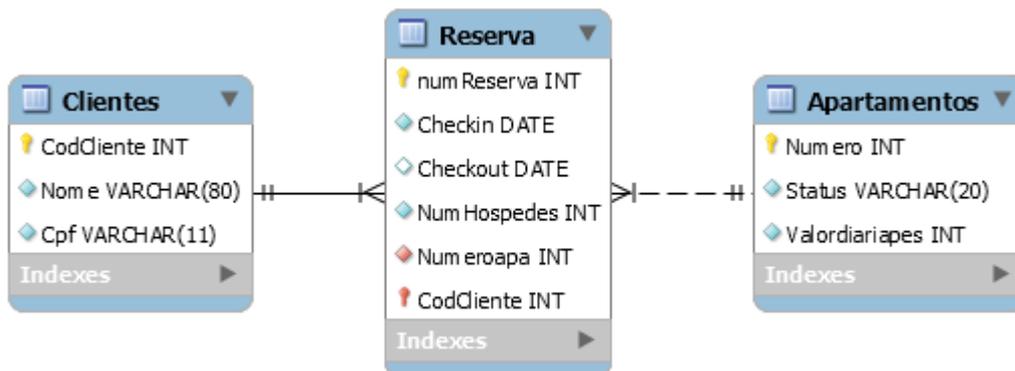
CREATE SCHEMA IF NOT EXISTS `bdhotel` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `bdhotel` ;
CREATE TABLE IF NOT EXISTS `bdhotel`.`Clientes` (
  `CodCliente` INT NOT NULL AUTO INCREMENT COMMENT '',
  `Nome` VARCHAR(80) NOT NULL COMMENT '',
  `Cpf` VARCHAR(11) NOT NULL COMMENT '',
  PRIMARY KEY (`CodCliente`) COMMENT ''
)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `bdhotel`.`Apartamentos` (
  `Numero` INT NOT NULL AUTO INCREMENT COMMENT '',
  `Status` VARCHAR(20) NOT NULL COMMENT '',
  `Valordiariapes` INT NOT NULL COMMENT '',
  PRIMARY KEY (`Numero`) COMMENT ''
)
ENGINE = InnoDB;

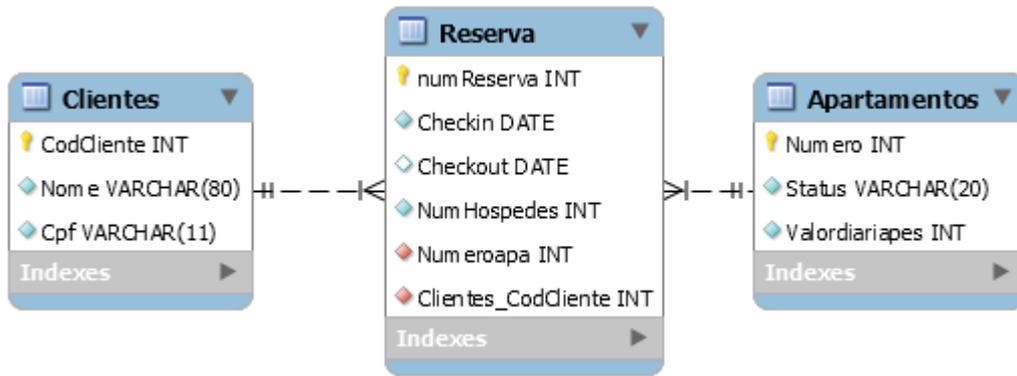
CREATE TABLE IF NOT EXISTS `bdhotel`.`Reserva` (
  `numReserva` INT NOT NULL AUTO INCREMENT COMMENT '',
  `Checkin` DATE NOT NULL COMMENT '',
  `Checkout` DATE NULL COMMENT '',
  `NumHospedes` INT NOT NULL COMMENT '',
  `Numerooapa` INT NOT NULL COMMENT '',
  `CodCliente` INT NOT NULL COMMENT '',
  PRIMARY KEY (`numReserva`, `CodCliente`) COMMENT '',
  INDEX `Naparta_idx` (`Numerooapa` ASC) COMMENT '',
  INDEX `fk Reserva Clientes1_idx` (`CodCliente` ASC) COMMENT '',
  CONSTRAINT `Naparta`
    FOREIGN KEY (`Numerooapa`)
    REFERENCES `bdhotel`.`Apartamentos` (`Numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk Reserva Clientes1`
    FOREIGN KEY (`CodCliente`)
    REFERENCES `bdhotel`.`Clientes` (`CodCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;

```

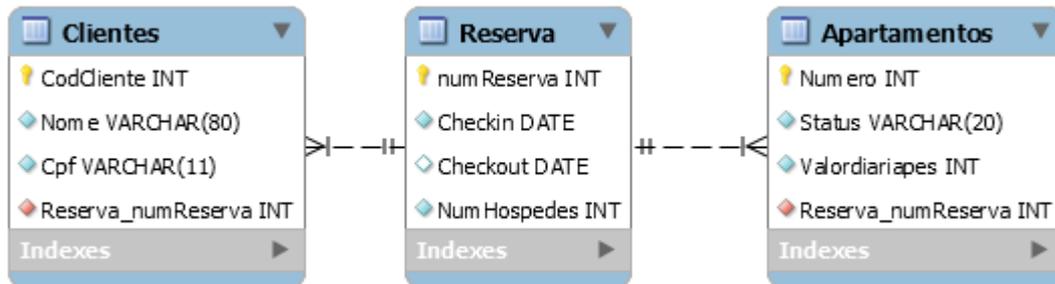
a)



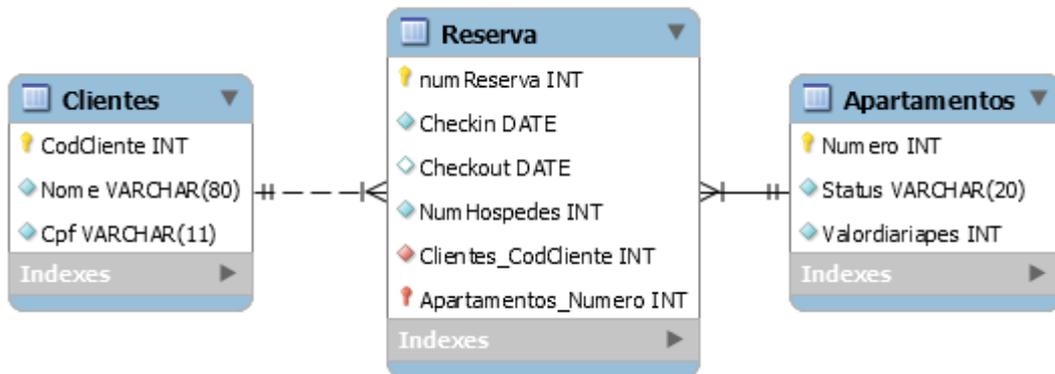
b)



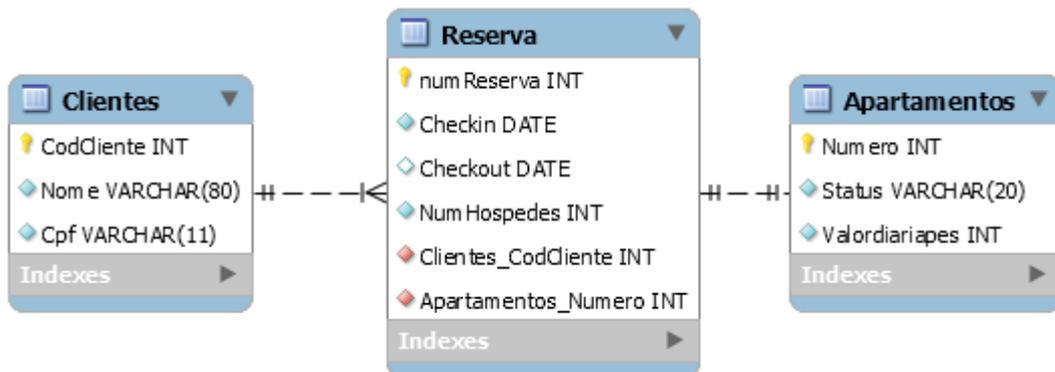
c)



d)



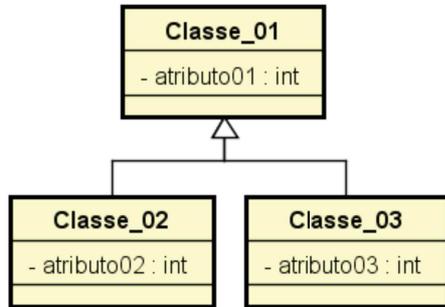
e)



37 O conjunto básico de operações para modelo relacional de banco de dados é a álgebra relacional, enquanto o cálculo relacional oferece uma linguagem declarativa de nível mais alto para especificar consultas relacionais. Sobre esses dois conceitos, analise as afirmativas abaixo e marque a opção **INCORRETA**.

- a) A operação *seleção* realiza uma partição horizontal da relação em dois conjuntos de tuplas: aquelas que satisfazem a condição e são selecionadas e aquelas que não satisfazem a condição e, portanto, não são selecionadas.
- b) A operação *projeção* realiza uma partição vertical da relação em duas relações: uma com as colunas selecionadas e outra com as colunas descartadas.
- c) Na operação *união* $R \cup S$, o resultado é uma relação que inclui todas as tuplas que estão em R e em S , inclusive as tuplas duplicadas.
- d) Na operação *intersecção* o resultado da operação indicada por $R \cap S$, é uma relação que inclui todas as tuplas que estão tanto em R quanto em S .
- e) O resultado da operação *diferença de conjunto*, indicada por $R - S$, é uma relação que inclui todas as tuplas que estão em R , mas não estão em S .

38 O mapeamento objeto-relacional permite a exportação de um modelo de classes da UML para um modelo Entidade-Relacionamento, porém esses dois modelos apresentam algumas especificidades que necessitam de adaptação para que se concretize a migração. Dentre essas especificidades está a herança, não suportada em bancos de dados relacionais tradicionais. Tendo em vista os objetivos básicos de projeto de bancos de dados, de permitir bom desempenho nas operações de consulta e alteração bem como a simplificação do desenvolvimento e manutenção da aplicação, observe o diagrama abaixo e marque a opção com a alternativa mais indicada.



- a) *Classe_01* migrará para uma tabela tendo como atributo *atributo01* e a *Classe_02* e a *Classe_03* migrarão para uma segunda tabela contendo *atributo02*, *atributo03* e *atributo01* como chave estrangeira. Essa solução deve ser usada com o objetivo de aumentar o desempenho nas operações de consulta.
- b) Cada classe do modelo de classes migrará para uma tabela no banco de dados e as tabelas *Classe_02* e *Classe_03* deverão ter *atributo01* como chave estrangeira para garantir a integridade do relacionamento 1:n intrínseco na herança.
- c) Por se tratar de uma classe abstrata, *Classe_01* não pode migrar para uma tabela de banco de dados, pois os bancos de dados relacionais não suportam tabelas abstratas. Nesse caso, criam-se duas tabelas: *Classe_02* e *Classe_03*, ambas com *atributo01*, além de seus respectivos atributos.
- d) *Classe_01*, *Classe_02* e *Classe_03* serão fundidas em uma única tabela contendo *atributo01*, *atributo02* e *atributo03* além de um atributo identificador. Essa solução deve ser usada em hierarquias mais simples e com poucos atributos.
- e) *Classe_01*, *Classe_02* e *Classe_03* serão fundidas em uma única tabela contendo *atributo01*, *atributo02* e *atributo03*. Essa solução somente pode ser implementada quando todos os objetos pertencerem exclusivamente à *Classe_02* ou à *Classe_03*.

39 Sistemas de gestão de conhecimento tentam ajudar as organizações a alcançar seus objetivos e podem envolver diferentes tipos de conhecimento. A gestão do conhecimento é uma área da computação de desenvolvimento de *softwares* os quais simulam a inteligência natural. Sobre os conceitos que envolvem a gestão do conhecimento, analise as afirmativas e marque a opção **CORRETA**.

- a) Uma máquina de inferência é especialmente utilizada quando não se tem uma base de dados que permita uma análise histórica. Por meio dessa técnica fazem-se inferências denominadas encadeamento para frente com retroalimentação.
- b) Conhecimento tácito é aquele objetivo, que pode ser medido e documentado em relatórios, artigos e regras.
- c) Uma heurística consiste em um conjunto de *software* e ferramentas usadas para desenvolver um sistema especialista para reduzir o tempo e o custo do desenvolvimento.
- d) Sistemas de aprendizagem são caracterizados pela realimentação dos resultados indesejáveis a fim de alterar as ações futuras. Como não trabalha com retroalimentação dos resultados desejáveis, essa técnica é uma solução mais simples, porém eficiente.
- e) Uma rede neural é um sistema computacional que pode simular o funcionamento do cérebro humano. Uma habilidade das redes neurais é analisar tendências detalhadas, uma vez que são capazes de aprender padrões e comportamentos e predizerem o que acontecerá em seguida.

40 Observe a tabela a seguir, que pertence a um banco de dados relacional. Marque a alternativa que contém o resultado esperado após a execução da instrução SQL ***SELECT * FROM aluno where nota > all (select nota from aluno where turma='B')***.

| matricula | Nome | Nota | Turma |
|-----------|----------|------|-------|
| 1 | Maria | 9.0 | A |
| 2 | Ana | 10.0 | A |
| 3 | Paulo | 7.0 | A |
| 4 | Tania | 6.0 | A |
| 5 | Jose | 8.0 | A |
| 6 | Neuza | 7.0 | A |
| 7 | Carlos | 10.0 | A |
| 8 | Simone | 7.0 | A |
| 9 | Cristina | 7.0 | B |
| 10 | Nubia | 6.0 | B |
| 11 | Matilde | 8.0 | B |
| 12 | Tere sa | 7.0 | B |
| 13 | Marcela | 6.0 | B |
| 14 | Cecilia | 7.0 | B |

a)

| matricula | Nome | Nota | Turma |
|-----------|--------|------|-------|
| 1 | Maria | 9.0 | A |
| 2 | Ana | 10.0 | A |
| 7 | Carlos | 10.0 | A |

| matricula | Nome | Nota | Turma |
|-----------|--------|------|-------|
| 1 | Maria | 9.0 | A |
| 2 | Ana | 10.0 | A |
| 3 | Paulo | 7.0 | A |
| 4 | Tania | 6.0 | A |
| 5 | Jose | 8.0 | A |
| 6 | Neuza | 7.0 | A |
| 7 | Carlos | 10.0 | A |
| 8 | Simone | 7.0 | A |

c)

| matricula | Nome | Nota | Turma |
|-----------|----------|------|-------|
| 9 | Cristina | 7.0 | B |
| 10 | Nubia | 6.0 | B |
| 11 | Matilde | 8.0 | B |
| 12 | Teresa | 7.0 | B |
| 13 | Marcela | 6.0 | B |
| 14 | Cecilia | 7.0 | B |

| matricula | Nome | Nota | Turma |
|-----------|--------|------|-------|
| 2 | Ana | 10.0 | A |
| 7 | Carlos | 10.0 | A |

| matricula | Nome | Nota | Turma |
|-----------|---------|------|-------|
| 1 | Maria | 9.0 | A |
| 2 | Ana | 10.0 | A |
| 5 | Jose | 8.0 | A |
| 7 | Carlos | 10.0 | A |
| 11 | Matilde | 8.0 | B |

41 Sobre a implementação de visões (*views*) em bancos de dados relacionais, é **INCORRETO** afirmar.

- a) Em SQL, o comando para especificar uma *view* é *Create View*.
- b) É responsabilidade do SGBD, e não do usuário, cuidar para que a *view* mantenha-se atualizada.
- c) Uma consulta em uma *view* é mais simples do que em tabelas de base, pois a leitura é especificada como se fosse em apenas uma tabela em vez de envolver a junção de duas ou mais tabelas.
- d) Assim como uma tabela de base, uma *view* é uma tabela física no banco de dados cujas tuplas sempre estão armazenadas fisicamente.
- e) Uma *view* é uma única tabela derivada de outras tabelas. Essas tabelas podem ser de base ou outras *views*.

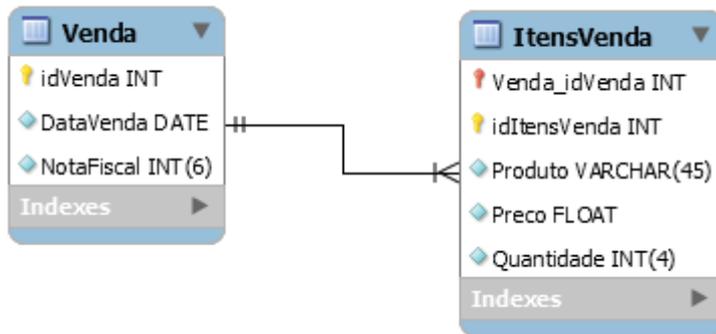
42 O PHP é uma linguagem de *script* de uso geral com código-fonte aberto especialmente utilizado para a implementação de recursos dinâmicos nas páginas *Web*. Sobre a linguagem PHP, marque a afirmativa **CORRETA**.

- a) Ao se delimitar uma *string* com aspas simples, os nomes das variáveis que aparecem dentro da *string* são substituídos pelos valores que estão atualmente armazenados nessas variáveis.
- b) Na linguagem PHP, os índices de um vetor são sempre valores numéricos inteiros iniciados por zero.
- c) Às variáveis do tipo booleana (*boolean*) podem ser atribuídos dois valores distintos: 0, que representa falso e 1, que representa verdadeiro.
- d) Os dados de um formulário HTML podem ser enviados para o servidor pelo método GET, que utiliza a própria URL. Nesse caso, o navegador coloca as informações do formulário junto à URL, separando o endereço da URL dos dados do formulário por um ponto de interrogação.
- e) A concatenação de *strings* é feita pelo operador +, assim, a expressão \$a='Ifes '+'2015' retornaria como resultado a *string* 'Ifes 2015'.

43 Sobre conceitos de sistemas de informação, marque a afirmativa **CORRETA**.

- a) A mineração de dados é uma ferramenta de análise de informação que envolve a descoberta automatizada de padrões e relações em um banco de dados.
- b) Processamento de folha de pagamento e processamento de pedidos são exemplos de sistemas de informação gerencial computadorizados.
- c) Software de aplicação direta caracteriza-se por manipular o *hardware* diretamente e, assim, aumentar o desempenho do processamento.
- d) Planejamento de recursos empresariais (ERP) é um tipo de *software* clássico para apoio a decisão, cujos dados de entrada são os dados produzidos por sistemas transacionais existentes na empresa.
- e) *Customer Relationship Management* (CRM) é um tipo de sistema de informação cujo objetivo principal é prover um correto planejamento de produção baseado no conhecimento do histórico de comportamento do cliente.

44 Uma empresa de comércio eletrônico possui em seu banco de dados uma tabela de vendas e outra de itens da venda, conforme esquema abaixo. Sempre que uma venda é excluída, os itens correspondentes àquela venda também devem ser excluídos, a fim de evitar inconsistência nos dados do sistema. Analise as afirmativas abaixo e marque a opção que melhor garanta a integridade referencial dos dados das duas tabelas.



- a) Deve-se criar uma *trigger* associada à tabela *Venda* com a cláusula *after delete*, que será disparada automaticamente quando a venda for excluída.
- b) Deve-se criar uma *trigger* associada à tabela *ItensVenda* com a cláusula *after delete*, que será disparada automaticamente quando a venda for excluída.
- c) Deve-se criar uma *stored procedure* associada à tabela *Venda*, que será disparada automaticamente quando a venda for excluída.
- d) Deve-se criar uma *stored procedure* associada à tabela *ItensVenda*, que será disparada automaticamente quando a venda for excluída.
- e) Deve-se criar uma *view* com a junção das tabelas *Venda* e *ItensVenda* e associar uma *trigger* a essa *view* que faça a exclusão em cascata.

45 Observe a tabela abaixo, que pertence a um banco de dados relacional.

| matricula | Nome | Nota | Turma |
|-----------|----------|------|-------|
| 1 | Maria | 9.0 | A |
| 2 | Ana | 10.0 | A |
| 3 | Paulo | 7.0 | A |
| 4 | Tania | 6.0 | A |
| 5 | Jose | 8.0 | A |
| 6 | Neuza | 7.0 | A |
| 7 | Carlos | 10.0 | A |
| 8 | Simone | 7.0 | A |
| 9 | Cristina | 7.0 | B |
| 10 | Nubia | 6.0 | B |
| 11 | Matilde | 8.0 | B |
| 12 | Teresa | 7.0 | B |
| 13 | Marcela | 6.0 | B |
| 14 | Cecilia | 7.0 | B |

Sobre essa tabela, foi executada uma instrução SQL que gerou o resultado a seguir com as colunas: turma, número total de alunos da turma e média das notas dos alunos da turma.

| turma | total | media |
|-------|-------|-------|
| A | 8 | 8.0 |

Analise as afirmativas e marque a opção que apresenta a instrução SQL que **NÃO** gera o resultado esperado.

- a) `select turma, count(nota) as total, avg(nota) as media from aluno group by turma having turma='A'`.
- b) `select turma, count(nota) as total, avg(nota) as media from aluno group by turma having turma <> 'B'`.
- c) `select turma, count(nota) as total, avg(nota) as media from aluno group by turma where turma='A'`.
- d) `select turma, count(nota) as total, avg(nota) as media from aluno where turma <> 'B'`.
- e) `select turma, count(nota) as total, avg(nota) as media from aluno where turma = (SELECT turma from aluno where matricula=1)`.

PROVA DE LEGISLAÇÃO DO SERVIÇO PÚBLICO FEDERAL

46 Ana Clara é ocupante do cargo efetivo de professora de Ensino Básico, Técnico e Tecnológico no IFES – Instituto Federal de Educação do Espírito Santo, lotada no Campus Vitória. Após três anos, passa a exercer suas funções, em caráter permanente, no Campus Montanha.

Com fundamento na Lei no 8.112/1990, considerando que Ana Clara foi deslocada no mesmo quadro e no mesmo cargo, analise o caso apresentado e responda, com base nos itens a seguir:

- I. A remoção é forma de provimento originário de cargo público.
- II. A remoção a pedido de Ana Clara sujeita-se a critério da Administração.
- III. A remoção somente pode ocorrer se houver necessariamente mudança de sede.
- IV. A remoção de Ana Clara não pode ser feita de ofício.

A partir da situação hipotética e dos itens acima, é correto afirmar que:

- a) Apenas os itens I e II estão corretos.
- b) Apenas os itens II e IV estão incorretos.
- c) Apenas o item II está incorreto.
- d) Apenas os itens I, III e IV estão incorretos.
- e) Apenas o item IV está correto.

47 Mauro, ocupante de cargo efetivo de Técnico Administrativo no IFES – Instituto Federal de Educação do Espírito Santo, saiu 2 (duas) horas antecipadamente do serviço, nos dias 04 e 05 de junho de 2015, com a justificativa de realização de exames médicos. Nos termos do que dispõe a Lei no 8.112/1990, em relação à compensação:

- a) Mauro não perderá a parcela de remuneração diária, proporcional às saídas antecipadas, uma vez que cumpriu mais da metade de sua carga horária diária de trabalho.
- b) Mauro não perderá a parcela de remuneração diária, proporcional às saídas antecipadas, se houver compensação de horário, a ser estabelecida pela chefia imediata, desde que compense a totalidade de horas até o último dia útil de junho de 2015.
- c) Mauro não perderá a parcela de remuneração diária, proporcional às saídas antecipadas, se houver compensação de horário, a ser estabelecida pela chefia imediata, desde que compense a totalidade de horas até o último dia útil de julho de 2015.
- d) Mauro não perderá a parcela de remuneração diária, proporcional às saídas antecipadas, se houver compensação de horário, a ser estabelecida pela chefia imediata, desde que compense a totalidade de horas até o último dia útil do ano de 2015.
- e) Mauro não perderá a parcela de remuneração diária, proporcional às saídas antecipadas, se houver autorização da chefia imediata quanto à dispensa da compensação, por ser mera liberalidade do gestor.

48 De acordo com a Lei nº 8.112/90, o servidor em estágio probatório faz jus às licenças e aos afastamentos, exceto:

- a) Serviço militar
- b) Tratamento da saúde do próprio servidor
- c) Tratamento de interesses particulares
- d) Tratamento de saúde de pessoa da família
- e) Exercício de mandato eletivo

49 Quanto ao estágio probatório, é correto afirmar (Lei nº 8.112/90):

- a) A 4 (quatro) meses antes de findo o período do estágio probatório, será submetida à homologação da autoridade competente a avaliação do desempenho do servidor, realizada por comissão constituída para essa finalidade.
- b) O servidor não aprovado no estágio probatório ficará em cadastro de disposição de outra instituição pública federal ou será reconduzido a cargo anteriormente ocupado.
- c) O servidor em estágio probatório não poderá exercer quaisquer cargos em comissão ou funções de direção, chefia ou assessoramento no órgão ou entidade de lotação.
- d) São os únicos fatores de avaliação para o desempenho do cargo durante o estágio probatório assiduidade, disciplina e pontualidade.
- e) O servidor em estágio probatório não sofre qualquer interrupção quando ocorrem licenças ou afastamentos, bem como na participação em curso de formação.

50 São formas de provimento de cargo público previstas na Lei nº 8.112/90, exceto:

- a) Remoção
- b) Nomeação
- c) Promoção
- d) Reversão
- e) Reintegração