



TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTITUCIONAL ESPECÍFICO

TOMO 2

CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO

**TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 2

Material instrucional específico, cujo conteúdo integral ou parcial não pode ser reproduzido ou utilizado sem autorização expressa, por escrito, da CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP – UNIVERSIDADE PAULISTA.

Questão 1

Questão 1.¹

Durante as eleições o eleitor deve comparecer à sua seção e zona, munido de um documento válido. Ao chegar ao local, apresenta o documento ao mesário, que verifica se o eleitor está apto a votar. Caso afirmativo, o mesário informa ao sistema o número do título de eleitor. O sistema valida o título e habilita o voto eletrônico para o eleitor. O eleitor informa os números de seus candidatos, podendo anular ou confirmar seu voto. Ao final do dia, termina o processo eleitoral da seção, o mesário finaliza o sistema, que gera os dados em tela ou em papel do resultado da urna, listando os votos para cada candidato. A totalização das urnas ocorre em um processo distinto em que o resultado final da eleição é apresentado à população.

Partindo dessa descrição, assinale a opção correta que corresponde à modelagem conceitual, utilizando diagrama de caso de uso com UML.

- A. Verificar o Documento do eleitor e Habilitar o Voto Eletrônico são casos de uso.
- B. No processo eleitoral da seção, os atores são: Eleitor, Mesário e População.
- C. O caso de uso Informar Título tem uma associação do tipo <<extends>> com o caso de uso Validar Título.
- D. O caso de uso Informar Número Candidato tem uma associação do tipo <<extends>> com os casos de uso Anular Voto e Confirmar Voto.
- E. Gerar Dados em Tela e Gerar Dados em Papel têm uma associação do tipo <<implements>> com o caso de uso Gerar Dados.

1. Introdução teórica

Caso de Uso

Um Caso de Uso é uma sequência de ações realizada pelo sistema que gera um resultado observável de valor para determinado Ator (RUP). Portanto, é a descrição de uma sequência de eventos de um Ator que usa o sistema para completar um processo. É uma técnica utilizada para descrever o que um sistema deve fazer. O Caso de Uso mapeia o problema, não a solução. Trata-se da visão do usuário a respeito do sistema.

Os Casos de Uso têm por objetivo:

- decidir e descrever os requisitos funcionais do sistema;

¹Questão 25 – Enade 2008.

- fornecer uma descrição clara e consistente do que o sistema deve fazer.

Um Ator é um papel que solicita ações ou eventos do sistema e recebe resultados. Cada Ator pode participar de vários Casos de Uso. Cada Caso de Uso é um objetivo de um Ator, e não tarefas do sistema.

Um exemplo de Atores e Casos de Usos está mostrado na figura 1.

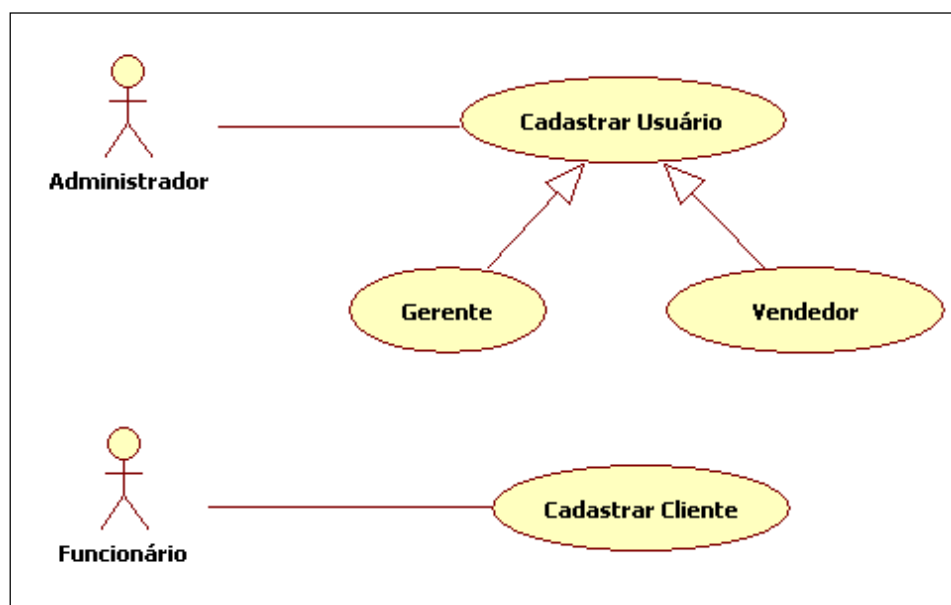


Figura 1. Exemplo de Atores e Casos de Usos.

Um relacionamento <<extends>> entre os Casos de Uso A e B significa que o Caso de Uso B pode ser acrescentado para complementar a funcionalidade do Caso de Uso A, mas essa adição não é obrigatória. Quando o Caso de Uso A for invocado, ele verificará se suas extensões devem ou não ser invocadas.

Na figura 2, temos um exemplo de Caso de Uso *Editar Documento*, que pode utilizar-se ou não das funcionalidades *Substituir Texto* ou *Corrigir Ortografia*, as quais são Casos de Uso independentes do primeiro.

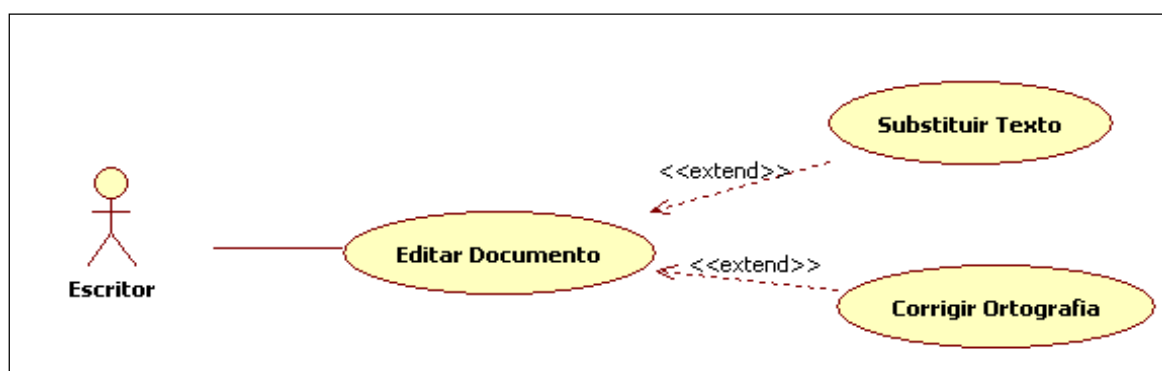


Figura 2. Exemplo de Caso de Uso *Editar Documento*.

A palavra-chave *implements* é utilizada para indicar que uma classe implementa determinada interface, conforme o exemplo dado na figura 3.

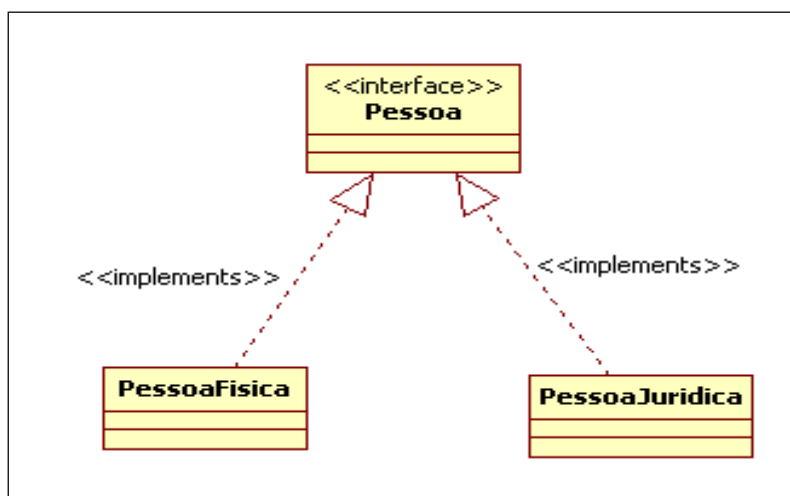


Figura 3. Palavra-chave *implements*.

2. Indicações bibliográficas

- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. *UML - guia do usuário*. 2. ed. Rio de Janeiro: Campus, 2005.
- GUEDES, G. T. A. *UML 2 – uma abordagem prática*. São Paulo: Novatec, 2009.
- LARMAN, C. *Utilizando UML e padrões*. 3. ed. Porto Alegre: Bookman, 2007.
- Processo Unificado do RUP (RUP - *Rational Unified Process*, versão 2002.05.00 (em Português). Disponível em <<http://www.wthreex.com/rup/>>. Acesso em 18 set. 2023.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. *Habilitar o voto eletrônico* é um Caso de Uso porque é uma ação desenvolvida pelo sistema para um retorno de valor para o Ator. *Verificar o documento do eleitor* não é um Caso de Uso porque nada acontece no sistema com relação a essa ação.

B – Alternativa incorreta.

JUSTIFICATIVA. O mesário informa ao sistema o número do título de eleitor, portanto é um Ator. O eleitor informa os números de seus candidatos e confirma ou anula o voto no sistema, portanto também é um Ator. Mas a população é apenas uma generalização que, no sistema apresentado, não tem significado, não sendo, portanto, Ator.

C – Alternativa incorreta.

JUSTIFICATIVA. *Informar Título* e *Validar Título* são passos do Caso de Uso *Habilitar o Voto Eletrônico*, e não Casos de Uso separados.

D – Alternativa correta.

JUSTIFICATIVA. O eleitor, ao informar o número do candidato, pode anular ou confirmar seu voto. Como são ações opcionais, a associação <<extends>> é a adequada.

E – Alternativa incorreta.

JUSTIFICATIVA. A associação do tipo <<implements>> refere-se a classes. Os Casos de Uso não são implementados, são realizados. Uma realização de Casos de Uso (não implementação) descreve como determinado Caso de Uso é realizado no modelo de projeto em termos de objetos de colaboração.

Questão 2

Questão 2.²

Os métodos de projeto e análise de algoritmos são necessários para o desenvolvimento de algoritmos eficientes, pois eles permitem que se resolvam problemas computacionais, reduzindo complexidade e tempo de execução. Acerca desses métodos, assinale a opção incorreta.

- A. A abordagem de Divisão e Conquista propõe dividir o problema em vários subproblemas, resolvendo-os e combinando suas soluções para criar a solução final do problema original.
- B. A Programação Dinâmica é uma técnica tipicamente aplicada a problemas de otimização em que pode haver várias soluções possíveis.
- C. O método Guloso nem sempre encontra a solução ótima, mas faz sempre a melhor escolha momentânea.
- D. A Programação Dinâmica e o Método Guloso têm em comum o fato de que se aplicam a problemas em que se observa a existência de sobreposição de subproblemas, ou seja, subproblemas que se repetem.
- E. Os métodos de Divisão e Conquista e Programação Dinâmica apresentam a mesma eficiência quando resolvem problemas combinando soluções de subproblemas dependentes uns dos outros.

1. Introdução teórica

Método de Divisão e Conquista

O método de Divisão e Conquista divide o problema em duas ou mais partes, criando subproblemas menores. Os subproblemas são resolvidos recursivamente, usando novas divisões e conquistas. Caso os subproblemas sejam suficientemente pequenos, serão resolvidos de forma direta.

A Programação Dinâmica é uma técnica computacional apoiada no princípio de otimização. As soluções são obtidas frequentemente por um processo regressivo, trabalhando no problema do final para o começo. A dificuldade é reduzida ao decompor o problema numa sequência de problemas inter-relacionados mais simples. Essa técnica baseia-se no princípio de que o problema completo pode ser resolvido se os valores das melhores soluções de certos subproblemas forem determinados.

²Questão 26 – Enade 2008.

O método Guloso baseia-se em decisões tomadas de forma isolada em cada passo. A estratégia é pegar a solução ótima local. Ao término do algoritmo, espera-se que a solução ótima tenha sido encontrada. É a técnica mais simples que pode ser aplicada a uma grande variedade de problemas. O problema possui n entradas e é requerido que satisfaça uma restrição chamada de solução viável. Uma solução viável que satisfaça a função objetivo é chamada de solução ótima.

É fácil determinar uma solução viável, mas não necessariamente uma solução ótima. O método Guloso sugere que podemos construir um algoritmo que trabalhe em estágios, considerando uma entrada por vez. Em cada estágio, uma decisão é tomada considerando se uma entrada particular é uma solução ótima. Isso é conseguido considerando as entradas em uma ordem determinada por algum processo de seleção. Se a inclusão da próxima entrada, na solução ótima construída parcialmente, resultará em uma solução inviável, então essa entrada não será adicionada à solução parcial. O processo de seleção em si é baseado em alguma medida de otimização, que pode ou não ser a função objetivo. Na verdade, várias medidas de otimização diferentes podem ser aplicáveis para determinado problema. Muitas delas, entretanto, resultarão em algoritmos que gerarão solução subótimas.

2. Indicações bibliográficas

- CORMEN, T. H. *Algoritmos - teoria e prática*. São Paulo: Campus, 2002.
- DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U. *Algoritmos*. São Paulo: McGraw-Hill, 2006.
- GOODRICH, M. T.; TAMASSIA, R. *Projeto de algoritmo*. Porto Alegre: Bookman, 2004.
- MEDINA, M.; FERTIG, C. *Algoritmos e programação - teoria e prática*. São Paulo: Novatec, 2005.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Trata-se da definição clássica do método *Divisão e Conquista*.

B – Alternativa incorreta.

JUSTIFICATIVA. A *Programação Dinâmica* aplica-se, principalmente, a problemas de otimização combinatória.

C – Alternativa correta.

JUSTIFICATIVA. O método *Guloso* nem sempre faz a melhor opção momentânea. Ele escolhe um subconjunto do problema que satisfaça determinada restrição. Essa função viável é chamada de *Solução Ótima*, mas o algoritmo nem sempre tende para uma solução ótima. Somente alguns problemas, que possuem a propriedade de escolha gulosa, têm uma solução global ótima, alcançada a partir de um conjunto de soluções locais ótimas.

D – Alternativa incorreta.

JUSTIFICATIVA. Os dois métodos trabalham em subproblemas que se repetem.

E – Alternativa correta.

JUSTIFICATIVA. Embora o método *Divisão e Conquista* resolva, recursivamente, problemas combinando subproblemas independentes, a *Programação Dinâmica* trata de subproblemas que não são necessariamente diferentes. Portanto, eles não apresentam a mesma eficiência.

Como há duas alternativas corretas (C e E), a questão foi **anulada**.

Questão 3

Questão 3.³

Os termos da sequência de Fibonacci são definidos por:

$$\text{Fibonacci}(0) = 0$$

$$\text{Fibonacci}(1) = 1$$

$$\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$$

Uma solução recursiva para o cálculo do i -ésimo termo da sequência é dada pela função a seguir.

```

1 funcao fibonacci(inteiro longo n)
2   se((n=0) OU (n=1)) entao
3     retorne n
4   senao
5     retorne fibonacci(n-1) + fibonacci(n-2)
6   fim se
7 fim
```

Acerca da execução recursiva dessa função, assinale a opção incorreta.

- A. À medida que o valor de n cresce, há um aumento no número de chamadas recursivas.
- B. Na linha 4, a ordem de execução é calcular o valor para $\text{fibonacci}(n-1)$ e somente depois calcular o valor para $\text{fibonacci}(n-2)$.
- C. O método recursivo é o mais eficiente para o cálculo do i -ésimo termo da sequência de Fibonacci, pois realiza duas chamadas por passo da recursão, cada uma mais simples do que a chamada original.
- D. As condições de parada da recursão são: o valor de n é 0 ou o valor de n é 1.
- E. O uso da recursão para o problema da série de Fibonacci não é indicado, pois ele gera rapidamente uma explosão de chamadas do método.

1. Introdução teórica

Solução recursiva de problemas

Uma solução recursiva de problemas é uma estratégia que utiliza a técnica de dividir para conquistar. A ideia fundamental é dividir um problema em um conjunto de subproblemas menores, resolvidos independentemente, para depois serem combinados a fim de gerar uma solução final.

³Questão 27 – Enade 2008.

Uma definição recursiva é constituída geralmente de uma parte base e uma parte recursiva. Muitas linguagens de programação permitem que funções chamem a si mesmas. As principais vantagens do uso da recursão são a clareza na interpretação do código e a simplicidade e elegância na implementação. Suas principais desvantagens são a dificuldade para depurar erros e a ineficiência.

A chamada de uma função requer espaço para parâmetros, variáveis locais e endereço de retorno. Com as chamadas recursivas, todas essas informações são armazenadas em uma pilha e depois retiradas. A quantidade de informação armazenada pode facilmente se tornar desproporcional ao número de chamadas.

O processo de alocar e liberar memória e de copiar informações envolve tempo computacional. Quanto mais chamadas, mais tempo gasto.

Por fim, é fundamental definir corretamente os critérios de parada, pois, se forem definidos de maneira equivocada, podem provocar uma infinidade de chamadas e consumir toda memória disponível.

2. Indicações bibliográficas

- CORMEN, T. H. *Algoritmos - teoria e prática*. São Paulo: Campus, 2002.
- DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U. *Algoritmos*. São Paulo: McGraw-Hill, 2006.
- GOODRICH, M. T.; TAMASSIA, R. *Projeto de algoritmo*. Porto Alegre: Bookman, 2004.
- MEDINA, M.; FERTIG, C. *Algoritmos e programação - teoria e prática*. São Paulo: Novatec, 2005.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. À medida que o valor de n cresce, há efetivamente um aumento no número de chamadas recursivas ($n-1$ e $n-2$).

B – Alternativa incorreta.

JUSTIFICATIVA. A execução de $fibonacci(n-1)$ também é recursiva e, somente após concluída, chama-se $fibonacci(n-2)$.

C – Alternativa correta.

JUSTIFICATIVA. O método recursivo torna-se mais ineficiente com o aumento de ***n***. Dependendo do valor de ***n***, pode estourar toda memória do computador.

D – Alternativa incorreta.

JUSTIFICATIVA. A condição de parada é sempre $n=0$ ou $n=1$.

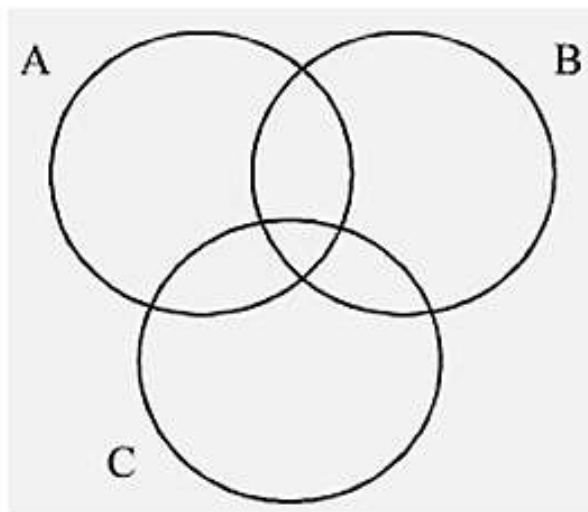
E – Alternativa incorreta.

JUSTIFICATIVA. Se ***n*** for muito grande, haverá uma explosão de chamadas de métodos e um *StackOverflow*. Um exemplo do cálculo de fibonacci para qualquer ***n*** dado, sem recursão, está mostrado abaixo. Esse programa é muito mais eficiente (embora menos elegante) que o proposto na questão. Para testes em casa, programe as duas soluções, execute para $n=30.000$ e verifique o tempo gasto por cada um dos programas.

```
1.  public long fibonacci ( long n ) {  
2.      long n1, n2, total;  
3.  
4.      total = 1;  
5.      n1 = 0;  
6.      n2 = 1;  
7.  
8.      for ( int i = 2; i <= n; i++ ) {  
9.          total = ( n1 + n2 );  
10.         n1 = n2;  
11.         n2 = total;  
12.     }  
13.  
14.     return total;  
15.  
16. }
```

Questão 4**Questão 4.⁴**

Leia o diagrama a seguir.



A figura acima mostra 3 conjuntos — A, B e C — em que cada conjunto é representado, no diagrama de Venn, por um círculo no plano. Com relação aos conjuntos A, B e C, julgue os seguintes itens.

- I $A \cup B = \emptyset$
- II $A - (B \cap C) = (A - B) \cap (A - C)$
- III $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- IV $A \cap A = \emptyset$

Assinale a opção correta.

- A. Apenas um item está certo.
- B. Apenas os itens I e II estão certos.
- C. Apenas os itens II e III estão certos.
- D. Apenas os itens III e IV estão certos.
- E. Apenas os itens II, III e IV estão certos.

1. Introdução teórica**Conjuntos**

Dados os conjuntos A e B, a sua união ($A \cup B$) é formada por todos os elementos pertencentes a A e B, conforme mostrado na figura 1.

⁴Questão 28 – Enade 2008.

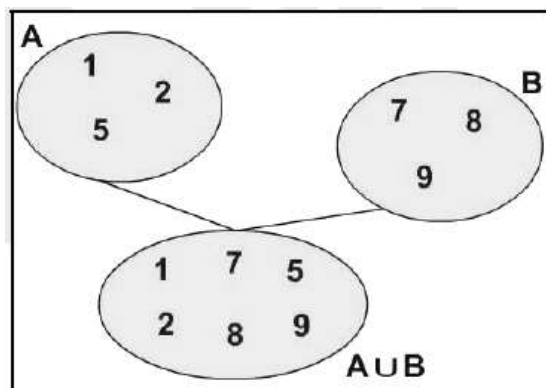


Figura 1. União de conjuntos.

Dados os conjuntos A e B, a sua interseção ($A \cap B$) é formada por todos os elementos pertencentes a A e B simultaneamente, conforme mostrado na figura 2.

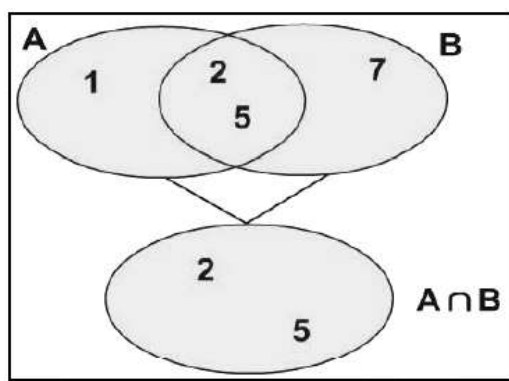


Figura 2. Interseção de conjuntos.

Dados os conjuntos A e B, a sua diferença ($A - B$) é formada pelos elementos que pertencem a A e não pertencem a B, conforme mostrado na figura 3.

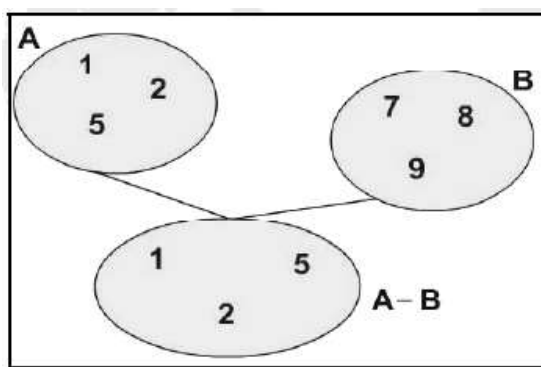


Figura 3. Diferença de conjuntos.

Vejam os exemplos a seguir.

1. Se $A = \{1, 2, 5\}$ e $B = \{7, 8, 9\}$ então:

$$A \cup B = \{1, 2, 5, 7, 8, 9\}$$

$$A \cap B = \emptyset \text{ (conjunto vazio)}$$

$$A - B = \{1, 2, 5\}$$

2. Se $A = \{1, 2, 5\}$ e $B = \{2, 5, 7, 8\}$ então:

$$A \cup B = \{1, 2, 5, 7, 8\}$$

$$A \cap B = \{2, 5\}$$

$$A - B = \{1\}$$

O conjunto vazio (\emptyset) é o conjunto sem elementos. Todo conjunto o possui como subconjunto. A união de um conjunto A com um conjunto vazio resulta sempre em A. A interseção do conjunto vazio com qualquer conjunto é vazio.

2. Indicações bibliográficas

- GERSTING, J. L. *Fundamentos matemáticos para a Ciência da Computação*. 5. ed. São Paulo: LTC, 2004.
- HAZZAN, S. *Fundamentos de Matemática Elementar – conjuntos e funções*. 8. ed. São Paulo: Atual, 2004.
- KNUTH, D. E. *Matemática Concreta: fundamentos para a Ciência da Computação*. 2. ed. São Paulo: LTC, 1995.

3. Análise dos itens

Para facilitar a visualização e o entendimento, colocamos números em cada um dos envolvidos nas interações entre os conjuntos, conforme mostrado na figura 4.

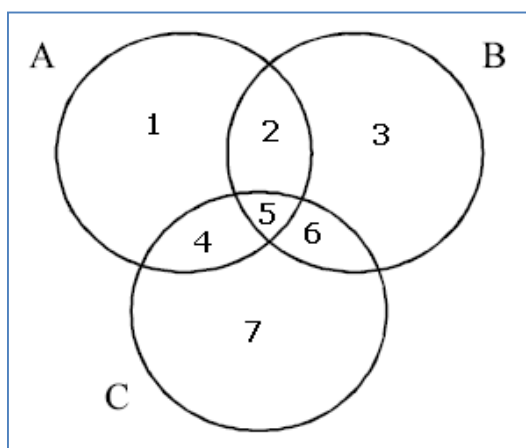


Figura 4. União dos conjuntos.

I - Item incorreto.

JUSTIFICATIVA. A união de A com B somente daria um conjunto vazio se tanto A quanto B fossem conjuntos vazios.

II – Item incorreto.

JUSTIFICATIVA. Vamos resolver o item “passo a passo”.

- $B \cap C = \{5, 6\}$
- $A - (B \cap C) = \{1, 2, 4, 5\} - \{5, 6\} = \{1, 2, 4\}$
- $(A - B) = \{1, 4\}$
- $(A - C) = \{1, 2\}$
- $(A - B) \cap (A - C) = \{1, 4\} \cap \{1, 2\} = \{1\}$

Portanto $\{1, 2, 4\}$ é diferente de $\{1\}$ e o item está incorreto.

III – Item correto.

JUSTIFICATIVA. Vamos resolver o item “passo a passo”.

- $(B \cup C) = \{2, 3, 4, 5, 6, 7\}$
- $A \cap (B \cup C) = \{2, 4, 5\}$
- $(A \cap B) = \{2, 5\}$
- $(A \cap C) = \{4, 5\}$
- $(A \cap B) \cup (A \cap C) = \{2, 4, 5\}$

Portanto $\{2, 4, 5\}$ é igual a $\{2, 4, 5\}$ e o item está correto.

IV – Item incorreto.

JUSTIFICATIVA. A interseção de A com A somente daria um conjunto vazio se A fizesse interseção com um conjunto vazio.

A regra é: $A \cap A = A$ e $A \cap \emptyset = \emptyset$.

Alternativa correta: A.

Questão 5

Questão 5.⁵

Considere a sentença a seguir.

Se Maria for ao aniversário, João irá e ficará feliz, mas Maria ficará infeliz, ou, se João não for ao aniversário, Maria irá e ficará feliz, mas João ficará infeliz.

Considere as seguintes proposições:

P: João vai ao aniversário;

Q: Maria vai ao aniversário;

R: João feliz; e

S: Maria feliz.

Assinale a opção que contém fórmula de lógica proposicional com uma representação válida para a sentença proposta. Quanto à notação dos operadores, considere: junção = \wedge ; disjunção = \vee ; negação = \neg ; implica = \rightarrow .

- A $((Q \rightarrow (P \wedge R)) \rightarrow \neg S) \vee ((\neg P \rightarrow (Q \wedge S)) \rightarrow R)$
- B $((\neg Q \rightarrow (P \wedge R)) \rightarrow S) \vee ((P \rightarrow (Q \wedge S)) \rightarrow \neg R)$
- C $((Q \rightarrow (P \wedge R)) \rightarrow \neg S) \vee ((\neg P \rightarrow (Q \wedge S)) \rightarrow \neg R)$
- D $((\neg Q \rightarrow (P \wedge R)) \rightarrow \neg S) \vee ((\neg P \rightarrow (Q \wedge S)) \rightarrow \neg R)$
- E $((Q \rightarrow (P \wedge R)) \rightarrow S) \vee ((\neg P \rightarrow (Q \wedge S)) \rightarrow R)$

1. Introdução teórica

Junção lógica (E) e disjunção lógica (OU)

A junção lógica (E) implica que todas as proposições têm que ser verdadeiras.

A disjunção lógica (OU) implica que pelo menos uma das proposições tem que ser verdadeira. Para a correta compreensão da questão, é necessário substituir cada item na sentença e comparar com a questão proposta.

2. Indicações bibliográficas

- FAVARO, S.; FILHO, O. K. *Noções de Lógica e Matemática Básica*. São Paulo: Ciência Moderna, 2005.

⁵Questão 29 – Enade 2008.

- GERSTING, J. L. *Fundamentos matemáticos para a Ciência da Computação*. 5. ed. São Paulo: LTC, 2004.
- KNUTH, D. E. *Matemática Concreta: fundamentos para a Ciência da Computação*. 2. ed. São Paulo: LTC, 1995.
- QUILELLI, P. *Raciocínio lógico-matemático*. 2. ed. São Paulo: Ferreira, 2010.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Não reflete o proposto na questão, conforme segue.

((Maria vai ao aniversário **implica que** (João vai ao aniversário **E** João feliz)) **implica que** Maria não feliz).

OU

((João não vai ao aniversário **implica que** (Maria vai ao aniversário **E** Maria fica feliz) **implica que** João feliz.)

Está incorreto porque, no segundo caso, João está feliz enquanto que, na proposta da questão, ele fica infeliz.

B – Alternativa incorreta.

JUSTIFICATIVA. Não reflete o proposto na questão, conforme segue.

((Maria não vai ao aniversário **implica que** (João vai ao aniversário **E** João feliz)) **implica que** Maria feliz).

OU

((João vai ao aniversário **implica que** (Maria vai ao aniversário **E** Maria fica feliz) **implica que** João não feliz.)

Está incorreto porque Maria não vai ao aniversário, enquanto que, nas duas situações da questão proposta, ela vai ao aniversário. Em uma, sente-se feliz (se João não for), e na outra, sente-se infeliz (se João for).

C – Alternativa correta.

JUSTIFICATIVA. Reflete o proposto na questão, conforme detalhado a seguir.

((Maria vai ao aniversário **implica que** (João vai ao aniversário **E** João feliz)) **implica que** Maria não feliz).

OU

((João não vai ao aniversário **implica que** (Maria vai ao aniversário **E** Maria fica feliz) **implica que** João não feliz.)

Está correto porque reflete exatamente o comportamento apresentado na questão, tanto da parte de João quanto da parte de Maria.

D – Alternativa incorreta.

JUSTIFICATIVA. Não reflete o proposto na questão, conforme segue.

((Maria não vai ao aniversário **implica que** (João vai ao aniversário **E** João feliz)) **implica que** Maria não feliz).

OU

((João não vai ao aniversário **implica que** (Maria vai ao aniversário **E** Maria fica feliz) **implica que** João não feliz.)

A segunda proposição está correta, mas a primeira não representa o comportamento proposto pela questão.

E – Alternativa incorreta.

JUSTIFICATIVA. Não reflete o proposto na questão, conforme segue.

((Maria vai ao aniversário **implica que** (João vai ao aniversário **E** João feliz)) **implica que** Maria feliz).

OU

((João não vai ao aniversário **implica que** (Maria vai ao aniversário **E** Maria fica feliz) **implica que** João feliz.)

Está incorreto nos dois casos porque, no primeiro, Maria fica feliz com a ida de João, mas na questão proposta ela fica infeliz se João for. No segundo caso, João não vai e Maria fica feliz, mas João também fica feliz, o que contraria o fato de que João fica infeliz.

Questão 6

Questão 6.⁶

Leia o *script* SQL a seguir.

```
CREATE TABLE JOGADOR (PSEUDONIMO VARCHAR(10) NOT NULL,
NOME VARCHAR(25) NOT NULL,
SENHA VARCHAR(6) NOT NULL,
PRIMARY KEY (PSEUDONIMO));

CREATE TABLE NIVEL(NIVEL NUMERIC(3) NOT NULL,
NOMEPSEUD VARCHAR(10) NOT NULL,
PONTOS NUMERIC(5) NOT NULL,
BONUS NUMERIC(5) NOT NULL,
PRIMARY KEY(NIVEL, NOMEPSEUD),
FOREING KEY (NOMEPSEUD) REFERENCES PSEUDONIMO);
```

A partir do *script* SQL de criação de um banco de dados acima, assinale a opção que apresenta comando SQL que permite obter uma lista em ordem decrescente de quantidade de bônus e que contenha somente o pseudônimo do jogador e seu bônus.

- A** SELECT NOME BONUS, FROM JOGADOR, NIVEL ORDER BY NIVEL.BONUS DESC
- B** SELECT * FROM JOGADOR, NIVEL WHERE JOGADOR.PSEUDONIMO = NIVEL.NOMEPSEUD ORDER BY NIVEL.BONUS DESC
- C** SELECT NOME, BONUS FROM JOGADOR, NIVEL WHERE JOGADOR.PSEUDONIMO = NIVEL.NOMEPSEUD ORDER BY NIVEL.BONUS DESC
- D** SELECT NOME, PONTOS FROM JOGADOR, NIVEL WHERE JOGADOR.PSEUDONIMO = NIVEL.NOMEPSEUD ORDEY BY NIVEL.BONUS DESC
- E** SELECT DISTINCT NOME, PONTOS FROM JOGADOR, NIVEL WHERE JOGADOR.PSEUDONIMO = NIVEL.NOMEJOGADOR ORDER BY NIVEL.BONUS DESC

1. Introdução teórica

CREATE TABLE

A questão está formulada de maneira errada porque o *CREATE TABLE* apresentado para a tabela NIVEL (segundo *create*) não funciona. O comando correto está mostrado abaixo, para o banco de dados MySQL.

```
create table nivel (nivel numeric(3) not null,
nomepseud varchar(10) not null,
pontos numeric(5) not null,
bonus numeric(5) not null,
primary key(nivel, nomepseud),
constraint fk_nomepseud foreign key (nomepseud) references
jogador(pseudonimo));
```

⁶Questão 30 – Enade 2008.

Para o banco de dados *SQL Server*, o comando é:

```
add constraint foreign key (nomepseud) references  
    dbo.jogador (pseudonimo)
```

Na formulação do comando, faltou informar a tabela destino do relacionamento. Portanto, a questão foi **anulada**.

2. Indicações bibliográficas

- DATE, C. J. *Introdução a sistemas de bancos de dados*. 8. ed. São Paulo: Campus, 2004.
- NAVATHE, S. *Sistemas de bancos de dados*. 4. ed. São Paulo: Addison-Wesley, 2005.
- SILBERSCHATZ, A. *Sistema de bancos de dados*. 5. ed. São Paulo: Campus, 2006.
- TEORY, T. *Projeto e modelagem de bancos de dados*. São Paulo: Campus, 2006.

3. Análise da questão

A questão foi formulada de maneira errada e foi **anulada**, conforme explicação dada.

Adicionalmente, mesmo que o comando estivesse correto, nenhuma pesquisa fornecida nas cinco opções traz uma lista decrescente de quantidade de bônus, contendo somente o pseudônimo do jogador. Todas as pesquisas trazem o nome do jogador. Não existe, portanto, uma resposta correta, ainda que o *create* estivesse certo.

Questão 7

Questão 7.⁷

O plano de negócios, mais do que um documento de elaboração das ações de implementação de um novo empreendimento, serve como documento que estabelece o relacionamento entre empreendedores e investidores. O conhecimento de características dos atores envolvidos nessa relação interfere diretamente na elaboração do plano de negócios. Considerando os papéis do empreendedor, do investidor e de conceitos de fatores envolvidos na elaboração do plano de negócios, assinale a opção correta.

- A. O verdadeiro empreendedor cria um negócio diante de uma oportunidade e procura, o mais breve possível, vendê-lo para um grupo de investidores.
- B. Investidores inteligentes consideram, ao analisar onde investir, que projeções financeiras mês a mês para um período maior que um ano constituem um dos fatores que garantem o sucesso de um novo empreendimento.
- C. O empreendedor é uma pessoa à procura de riscos, que diante de uma nova oportunidade de empreendimento transfere todos os riscos para si.
- D. As pessoas, as oportunidades, o contexto e as possibilidades de riscos e recompensas são quatro fatores fundamentais, que devem ser considerados para o sucesso de um novo empreendimento.
- E. Um plano de negócios deve ser criado seguindo uma fórmula de sucesso preestabelecida apresentada em livros da área de administração e implementada em aplicativos.

1. Introdução teórica

Empreendedorismo e plano de negócios

Uma empresa é criada para sobreviver no longo prazo. Para isso, algumas ações são fundamentais, como as citadas a seguir.

- Inovar para crescer.
- Investir em qualificação.
- Fazer um planejamento estratégico.
- Manter a qualidade de produtos e serviços.
- Gerenciar o negócio.
- Controlar os custos.

⁷Questão 31 – Enade 2008.

- Manter a base de clientes.

O plano de negócios é o documento que estrutura as ideias e as opções que o empreendedor analisará para decidir sobre a viabilidade do seu negócio. Deve conter informações sobre o negócio, as previsões e as projeções financeiras, a análise do mercado, a previsão de fluxo de caixa e as necessidades de capital.

O empreendedor é aquele que inicia novos negócios e consegue escolher entre várias alternativas. Ele apresenta determinadas habilidades e competências para criar, abrir e gerir um negócio, gerando resultados positivos.

É impossível afastar o elemento risco do negócio, por isso ele tem que ser gerenciado e acompanhado, com o objetivo de minimizá-lo. Assumir riscos é a principal característica do empreendedor.

Não existem fórmulas mágicas para ser um empreendedor de sucesso. Há exemplos de inúmeros negócios que prometiam sucesso e fracassaram. Igualmente, temos, também, inúmeros negócios que nasceram por acaso e hoje são completos sucessos.

2. Indicações bibliográficas

- CARVALHO, H. G. *Empreendedorismo*. São Paulo: Ferreira, 2009.
- GOLEMAN, D. *Os grandes empreendedores*. São Paulo: Elsevier, 2007.
- JUSTUS, R. L. *O empreendedor*. São Paulo: Larousse, 2007.
- PETERS, M. *Empreendedorismo*. 7. ed. São Paulo: Artmed, 2009.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. A criação do negócio é voltada para o longo prazo e para a permanência da empresa.

B – Alternativa incorreta.

JUSTIFICATIVA. As projeções financeiras devem abranger vários períodos de análise (mensais, semestrais, anuais e acima de um ano). Quanto mais distante o prazo, menos precisas serão. Ao projetar um investimento, o investidor deve levar em conta o período necessário para que o capital seja remunerado, o que pode levar vários anos.

C – Alternativa incorreta.

JUSTIFICATIVA. Embora seja correto que o empreendedor assume riscos, ele não os procura (os riscos são inerentes ao negócio). Muito menos transfere todos para si mesmo.

D – Alternativa correta.

JUSTIFICATIVA. Os fatores listados estão corretos. O negócio tem que ser avaliado em várias dimensões para que não sejam cometidos erros graves que o inviabilizem.

E – Alternativa incorreta.

JUSTIFICATIVA. Não existe uma fórmula de sucesso pré-estabelecida. Sucessos passados não garantem a viabilidade futura de um negócio similar. Cada um tem que ser analisado como um evento singular, mesmo que sejam comparados com negócios ou mercados similares.

Questão 8

Questão 8.⁸

Com relação a conceitos de orientação a objetos, julgue os itens.

- I. As variáveis ou métodos declarados com modificador de acesso *private* só são acessíveis a métodos da classe em que são declarados.
- II. Uma classe deve possuir uma única declaração de método construtor.
- III. Uma instância de uma classe abstrata herda atributos e métodos de sua superclasse direta.
- IV. O polimorfismo permite substituir a lógica condicional múltipla (lógica *switch* ou faça caso).

São corretos apenas os itens

- A. I e II.
- B. I e III.
- C. I e IV.
- D. II e III.
- E. II e IV.

1. Introdução teórica

Construtor, classe e polimorfismo

Um membro (método ou variável) declarado com o modificador *private* somente pode ser acessado por uma instância da classe que declara o método ou variável. Veja o exemplo abaixo.

```
// Classe Cliente
class Cliente{
    private String nome = "Osmar";
}

// Classe de teste
public class Estudos{
    public static void main(String args[]){
        Cliente cliente = new Cliente();
        System.out.println(cliente.nome);

        System.exit(0);
    }
}
```

⁸Questão 32 – Enade 2008.

A tentativa de compilação do código anterior dará erro, porque a classe Estudos não tem acesso aos membros privados da classe Cliente. Do mesmo modo, uma subclasse não tem acesso aos membros privados da superclasse.

Um construtor é o conjunto de códigos utilizado para iniciar uma classe ou para construí-la. Sempre que uma classe é criada, um construtor padrão é vinculado a ela. É o método chamado sempre que uma nova instância da classe é criada, responsável pela alocação de recursos necessários ao funcionamento do objeto, além da definição inicial dos atributos. As instruções contidas no construtor serão executadas sempre que for instanciado um objeto da classe.

No exemplo abaixo, a classe TV é construída com tamanho de 21 polegadas, nenhum canal e desligada. Além do construtor padrão, uma classe pode ter inúmeros construtores.

```
class TV {
    int tamanho;
    int canal;
    boolean ligada;

    TV() {
        tamanho=21;
        canal=0;
        ligada=false;
    }
}
```

Uma classe abstrata é um tipo de classe que somente pode ser herdada, nunca instanciada. É uma classe conceitual que pode definir funcionalidades para suas subclasses implementarem. Vejamos um exemplo:

```
public abstract class ClasseAbstrata {
    public void umaImplementacaoDefault() {
        // chama o método abstrato
        fazAlgumaCoisa();
    }

    public abstract void fazAlgumaCoisa();
}
```

Observemos a implementação abaixo.

```
class Filha extends ClasseAbstrata {
    public void fazAlgumaCoisa() {
        System.out.println("executou método fazAlgumaCoisa");
    }
}
```

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura), mas

comportamentos distintos, especializados para cada classe derivada. A decisão de qual método usar é tomada em tempo de execução, de acordo com o tipo de classe derivada.

Veja o que segue.

- **Classe abstrata.**

```
public abstract class OperacaoMatematica {
    public abstract double calcular(double x, double y);
}
```

- **Implementações.**

```
public class Soma extends OperacaoMatematica {
    public double calcular(double x, double y) {
        return x+y;
    }
}
```

```
public class Subtracao extends OperacaoMatematica {
    public double calcular(double x, double y) {
        return x-y;
    }
}
```

Observe que o método *calcular* retorna resultados diferentes dependendo da operação. A sobrecarga de métodos é um tipo de polimorfismo estático.

A instrução *switch* é usada quando precisamos escolher entre múltiplos caminhos alternativos. Vejamos um exemplo.

```
public class Estudos{
    public static void main(String[] args){
        int valor = 4;

        switch(valor){
            case 1:
                System.out.println("Valor é 1");
                break;
            case 2:
                System.out.println("Valor é 2");
                break;
            case 3:
                System.out.println("Valor é 3");
                break;
            default:
                System.out.println("Valor diferente de 1, 2 e 3");
                break;
        }
    }
}
```

Praticamente, qualquer *switch* pode ser substituído por polimorfismo, dando maior clareza ao código. O *switch* também poderia ser substituído, em alguns casos, por uma *hash table* (dicionário, mapa etc.).

Os exemplos abaixo, para as classes *Soma* e *Subtração*, mostram a diferença de implementação com *switch* e com polimorfismo.

- Classe de Teste.

```
public class Contas {
    public static void mostrarCalculo(OperacaoMatematica operacao, double x, double y) {
        System.out.println("O resultado é: " + operacao.calcular(x, y));
    }

    public static void main(String args[]) {
        //Primeiro calculamos uma soma
        Contas.mostrarCalculo(new Soma(), 5, 5); //Imprime o resultado é: 10
        //Depois uma subtração
        Contas.mostrarCalculo(new Subtracao(), 5, 5); //Imprime o resultado é: 0
    }
}
```

- Classe que implementa o tipo de cálculo.

```
public static void mostrarCalculo(Soma soma, double x, double y) {
    System.out.println("O resultado é: " + soma.calcular(x, y));
}

public static void mostrarCalculo(Subtracao subtracao, double x, double y) {
    System.out.println("O resultado é: " + subtracao.calcular(x, y));
}
```

O mesmo resultado utilizando o switch está indicado abaixo.

```
public void mostrarCalculo (int operacao, double x, double y) {
    System.out.print("O resultado é: ");
    switch (operacao) {
        case SOMA:
            System.out.print(""+(x+y));
            break;
        case SUBTRACAO:
            System.out.print(""+(x-y));
            break;
        //... outras operacoes
        default: throw new UnsupportedOperationException()
    }
}
```

2. Indicações bibliográficas

- ARAÚJO, E. C. *Orientação a objetos com Java*. São Paulo: Visual Books, 2008.
- CAMARA, Fo. *Orientação a objeto com.net*. 2. ed. Florianópolis: Visual Books, 2006.
- CARDOSO, C. *Orientação a objeto na prática*. São Paulo: Ciência Moderna, 2006.
- WEST, D. *Use a cabeça! Análise e projeto orientado ao objeto*. Porto Alegre: Alta Books, 2007.

3. Análise dos itens

I – Item correto.

JUSTIFICATIVA. As variáveis *private* somente são acessíveis a métodos da classe em que são declarados. É a definição de variáveis *private*.

II – Item incorreto.

JUSTIFICATIVA. Uma classe deve possuir uma única declaração de método construtor.

III – Item incorreto.

JUSTIFICATIVA. Uma instância de uma classe abstrata herda atributos e métodos de sua superclasse direta. Classes abstratas não são instanciadas.

IV - Item correto.

JUSTIFICATIVA. Polimorfismo permite substituir lógica condicional múltipla.

Alternativa correta: C.

Questão 9

Questão 9.⁹

Após atuar como programador em uma empresa de desenvolvimento de software por aproximadamente 10 anos, um funcionário que se destacou por nunca atrasar um cronograma foi nomeado gerente de projetos. Ao assumir o primeiro projeto, o funcionário foi informado que sua principal responsabilidade era a realização da entrega conforme o cronograma estabelecido no contrato. Para o gerenciamento de tempo, o gerente de projetos irá utilizar o PMBOK.

Considerando essa situação, é correto afirmar que o gerente de projetos deverá coordenar processos de

- A. planejamento, garantia e controle da qualidade.
- B. definição e sequenciamento de atividades, estimativa de recursos e duração da atividade, desenvolvimento e controle do cronograma.
- C. planejamento, definição, verificação e controle do escopo.
- D. estimativa de custos, realização do orçamento e controle de custos.
- E. planejamento de compra e contratações, seleção de fornecedores e encerramento do contrato.

1. Introdução teórica

Processos

O PMBOK é o *Project Management Body of Knowledge*, isto é, Corpo de Conhecimento em Gerência de Projetos do PMI (*Project Management Institute*, Instituto de Gerência de Projetos).

Os processos requeridos para entregar o projeto no prazo acordado estão na Gerência de Tempo. Esses processos são os citados a seguir.

- Definição das atividades.
- Sequenciamento das atividades.
- Estimativa de recursos das atividades.
- Estimativa de duração das atividades.
- Desenvolvimento do cronograma.
- Controle do cronograma.

⁹Questão 33 – Enade 2008.

Os demais processos apresentados na questão pertencem a outras áreas, como descrito a seguir.

- Planejamento, garantia e controle da qualidade pertencem à Gerência de Qualidade.
- Planejamento, definição, verificação e controle do escopo pertencem à Gerência de Escopo.
- Estimativa de custos, realização do orçamento e controle de custos pertencem à Gerência de Custos.
- Planejamento de compra e contratações, seleção de fornecedores e encerramento do contrato pertencem à Gerência de Aquisições (*Procurement*).

2. Indicações bibliográficas

- DINSMORE, P. C. *Gerenciamento de projetos*. São Paulo: QualityMark, 2004.
- *Guia de Conhecimento em Gerenciamento de Projetos - PMBOK*. 3. ed. PMI, 2005.
- IRELAND, L. R. *Gerenciamento de projetos*. 2. ed. São Paulo: Lab, 2007.
- PMBOK – *A Guide to the Project Management Body of Knowledge*. 3. ed. PMI, 2004.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Processos da Gerência de Qualidade.

B – Alternativa correta.

JUSTIFICATIVA. Processos da Gerência de Tempo.

C – Alternativa incorreta.

JUSTIFICATIVA. Processos da Gerência de Escopo.

D – Alternativa incorreta.

JUSTIFICATIVA. Processos da Gerência de Custos.

E – Alternativa incorreta.

JUSTIFICATIVA. Processos da Gerência de Aquisições (*Procurement*).

Questão 10

Questão 10.¹⁰

Um mercado que comercializa alimentos hortifrutigranjeiros faz compras diárias de diversas fazendas e enfrenta prejuízos decorrentes da falta de controle relacionada ao prazo de validade de seus produtos. Para resolver esse problema, o proprietário resolve investir em informatização, que proporcionará o controle do prazo de validade a partir da data da compra do produto. A modelagem de dados proposta pelo profissional contratado apresenta três tabelas, ilustradas a seguir, sendo que o atributo Código nas tabelas Produto e Fazenda são unívocos.

Produto	Estoque	Fazenda
Código	Produto	Código
Tipo de Produto	Fazenda	Nome
Descrição	Data da compra	Endereço
	Validade do Produto	Telefone
	Quantidade	

A partir das informações acima, é correto concluir que

- A. o relacionamento entre as tabelas Produto e Estoque é do tipo um-para-muitos.
- B. o campo Produto na tabela Estoque não pode fazer parte da chave nesta tabela e corresponde ao campo Descrição na tabela Produto.
- C. o campo Fazenda na tabela Estoque deverá ser a chave primária nesta tabela e corresponde ao campo Código na tabela Fazenda.
- D. o campo Código é chave primária na tabela Produto e identifica a fazenda fornecedora do produto.
- E. a tupla {produto, fazenda} deverá ser usada como a chave primária da tabela Estoque.

1. Introdução teórica

Atributos, modelagem e relacionamento

Um atributo ou conjunto de atributos designa-se por chave quando permite identificar de modo único ou unívoco cada entidade concreta ou registro da tabela.

¹⁰Questão 34 – Enade 2008.

A modelagem apresenta a resolução de um relacionamento “muitos-para-muitos”, entre *Produto* e *Fazenda*, com a criação da tabela associativa *Estoque*. O relacionamento surge quando vários produtos são comprados de várias fazendas.

A resolução de um relacionamento “muitos-para-muitos” ($M * N$) é feita pela transformação em dois relacionamentos “1-para-muitos”. Assim, teremos um relacionamento “1-para-muitos” entre *Produto* e *Estoque* e outro relacionamento “1-para-muitos” entre *Fazenda* e *Estoque*.

Conforme mostrado na figura 1, a entidade *Produto* tem uma chave primária *Codigo*, a entidade *Fazenda* também tem uma chave primária *Codigo* e a Entidade *Estoque* tem as chaves *Produto* e *Fazenda* ligando cada *Codigo* respectivo.



Figura 1. Resolução do relacionamento NxM.

A figura 2 mostra a ligação dos campos em cada tabela (*Codigo* com *Produto* e *Fazenda* com *Codigo*).



Figura 2. Ligação entre campos.

É necessária a chave *DataCompra* na entidade *Estoque* porque, senão, os produtos poderiam somente ser comprados em uma data específica (seja ela qual for), não permitindo registrar as datas de dois dias de compra diferentes, por exemplo.

2. Indicações bibliográficas

- DATE, C. J. *Introdução a sistemas de bancos de dados*. 8. ed. São Paulo: Campus, 2004.
- NAVATHE, S. *Sistemas de bancos de dados*. 4. ed. São Paulo: Addison-Wesley, 2005.

- SILBERSCHATZ, A. *Sistema de bancos de dados*. 5. ed. São Paulo: Campus, 2006.
- TEORY, T. *Projeto e modelagem de bancos de dados*. São Paulo: Campus, 2006.

3. Análise das alternativas

A – Alternativa correta.

JUSTIFICATIVA. O relacionamento entre *Produto* e *Estoque* é “1-para-muitos”, resultante de um lado da resolução da relação “muitos-para-muitos” entre *Produto* e *Fazenda*.

B – Alternativa incorreta.

JUSTIFICATIVA. O campo *Produto* na tabela *Estoque* corresponde ao campo *Codigo* na tabela *Produto* e tem que fazer parte da chave primária para resolver corretamente o relacionamento “muitos-para-muitos”.

C – Alternativa incorreta.

JUSTIFICATIVA. O campo *Fazenda* na tabela *Estoque* também tem que ser parte da chave primária, mas não toda a chave primária. Ele corretamente corresponde ao campo *Codigo* na tabela *Fazenda*.

D – Alternativa incorreta.

JUSTIFICATIVA. O campo *Codigo* ocorre nas tabelas *Produto* e *Fazenda*. Na tabela *Produto*, identifica o *Produto* fornecido, e, na tabela, *Fazenda* a *Fazenda* fornecedora.

E – Alternativa incorreta.

JUSTIFICATIVA. A chave primária na tabela *Estoque* tem que corresponder à tupla {*produto*, *fazenda*, ***data da compra***} para que seja permitido identificar e controlar as compras feitas em dias diferentes. Sem esse controle, não é possível resolver o problema principal da questão, que é a falta de controle relacionado ao prazo de validade dos produtos.

ÍNDICE REMISSIVO

Questão 1	Caso de Uso.
Questão 2	Método de Divisão e Conquista.
Questão 3	Solução recursiva de problemas.
Questão 4	Conjuntos.
Questão 5	Junção lógica (E) e disjunção lógica (OU).
Questão 6	CREATE TABLE.
Questão 7	Empreendedorismo e plano de negócios.
Questão 8	Construtor, classe e polimorfismo.
Questão 9	Processos.
Questão 10	Atributos, modelagem e relacionamento.