UNIP
EDUCAÇÃO, POR SUA ESCOLHA

SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 1

CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP

SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 1

Christiane Mazur Doi

Doutora em Engenharia Metalúrgica e de Materiais, Mestra em Ciências - Tecnologia Nuclear, Especialista em Cálculo e Matemática Aplicada, Especialista em Língua Portuguesa e Literatura, Especialista em Recursos Gramaticais para Revisão Textual, Engenheira Química e Licenciada em Matemática, com Aperfeiçoamento em Estatística e MBA em Engenharia Econômica. Professora titular da Universidade Paulista.

José Carlos Morilla

Doutor e Mestre em Engenharia de Materiais, Mestre em Engenharia de Produção, Especialista em Engenharia Metalúrgica, Especialista em Física e Engenheiro Mecânico, com MBA em Gestão de Empresas. Professor adjunto da Universidade Paulista.

Tiago Guglielmeti Correale

Doutor em Engenharia Elétrica, Mestre em Engenharia Elétrica e Engenheiro Elétrico (ênfase em Telecomunicações). Professor titular da Universidade Paulista.

Questão 1

Questão 1.[1]

A tendência atual de desenvolver aplicações que possam ser utilizadas em nuvem otimiza o desempenho de máquinas locais, munidas de poucos recursos computacionais, em uma rede e garante a possibilidade de esses computadores utilizarem outros recursos computacionais de maior desempenho, bem como a inserção de novos hosts nesse ambiente computacional, sem comprometimento do sistema como um todo.

Esse cenário pressupõe a utilização de sistema operacional com

A. arquitetura monolítica.

B. arquiteturas em camadas.

C. arquitetura monocamadas.

D. arquiteturas cliente-servidor.

E. arquitetura de máquinas virtuais.

1. Introdução teórica

Arquitetura de sistemas operacionais: arquitetura cliente-servidor

Para compreendermos a estratégia de criar um sistema operacional com uma arquitetura cliente-servidor, é importante termos um panorama geral desse padrão arquitetural. Adicionalmente, devemos entender como a evolução do software e do hardware motivou a utilização desse tipo de arquitetura.

A evolução do hardware na computação, a elevação das necessidades de uso e o crescimento da complexidade dos sistemas computacionais levaram ao aumento da complexidade dos sistemas operacionais. Máquinas com hardware mais sofisticado requerem sistemas operacionais mais sofisticados para seu gerenciamento. Além disso, o desenvolvimento de softwares mais sofisticados, que rodam sobre dado sistema operacional, impõe a requisição de novas funcionalidades por parte dos desenvolvedores de software, como, por exemplo, as relativas à segurança. Esse ciclo leva ao debate do software "puxando" o hardware versus o hardware "puxando" o desenvolvimento do software.

Um exemplo clássico desse problema ocorreu com o surgimento das redes de computadores. Durante muito tempo, os computadores funcionavam como "ilhas", isolados uns dos outros. A transmissão de informação de uma máquina para outra, nessa fase, tinha

[1]Questão 22 – Enade 2017.

de ser feita de forma física, por alguma mídia e pela cópia manual de arquivos. Esse processo é ruim e penoso, com ampla possibilidade para erros, além de ser frustrante para o usuário. As redes de computadores permitem que a cópia de arquivos seja feita de forma muito mais simples, rápida e segura.

A possibilidade de interligação de computadores levou à possibilidade do consumo remoto de recursos computacionais. Por exemplo, é possível que uma pessoa remotamente se conecte a uma máquina, execute programas e grave arquivos, mesmo que esse usuário esteja a quilômetros de distância.

No início (durante as décadas de 1960 e 1970), esse processo foi chamado de teleprocessamento. Isso foi estimulado pelas empresas da época, pelo fato de que as máquinas eram muito caras, e a melhor forma de evitar a sua ociosidade (tempo no qual um computador fica parado sem executar nenhum programa) era distribuir o seu uso por diversos usuários, utilizando os "terminais burros", equipamentos baratos (em comparação ao computador) que permitiam o acesso remoto a um computador e que não dispunham de capacidade local de processamento.

Ao mesmo tempo, a elevação da capacidade de processamento dos computadores e o aumento da sofisticação dos sistemas operacionais tornaram possível fazer com que um computador execute vários processos em paralelo (sistemas operacionais multitarefa) e de vários usuários diferentes (sistemas operacionais multiusuário).

Nesse processo de evolução de hardware e software, surge mais um novo elemento: o microcomputador. Com o barateamento dos custos de produção de um computador, empresas e pessoas começaram a ter a sua própria máquina pessoal. Assim, as redes de computadores passaram a conectar não apenas máquinas poderosas em grandes centros de processamento, mas também máquinas simples, distribuídas por vários usuários.

Esses elementos históricos da evolução da computação explicam a popularização da arquitetura cliente-servidor. Nessa arquitetura, dois tipos de programas trabalham em conjunto no processamento da informação. Clientes fazem requisições a um processo servidor, que normalmente está sendo executado em outra máquina. Nesse caso, as requisições são transmitidas pela rede.

O processo-servidor recebe várias requisições dos clientes e deve atendê-las de acordo com as regras de negócio. Um exemplo bastante comum de utilização dessa abordagem é baseado no protocolo TCP/IP. Um processo-servidor fica "escutando" em uma porta específica de uma máquina conectada a uma rede. Cada máquina conectada na rede tem um endereço IP próprio, inclusive o servidor. Os clientes, que rodam em outras

máquinas também conectadas na rede, devem tentar conectar-se ao servidor utilizando o endereço IP da máquina e na porta específica em que o processo-servidor fica "escutando".

O processo-cliente deve, então, enviar requisições para o processo-servidor, normalmente utilizando algum tipo de protocolo de comunicação associado ao serviço que está sendo provido. Frequentemente, antes do envio das primeiras requisições, deve existir uma etapa de autenticação para evitar que ocorram abusos no uso do servidor.

Após a abertura da conexão entre o processo-servidor e o processo-cliente, muitas requisições podem ser enviadas, seguidas das suas respectivas respostas por parte do processo-servidor. Quando o cliente não precisar mais enviar novas requisições, ele pode solicitar o fim da comunicação e fechar a conexão com o servidor.

Uma das grandes vantagens desse tipo de arquitetura é que uma mesma máquina com o processo-servidor pode atender a várias máquinas-clientes diferentes. Isso pode ocorrer de forma simultânea, como o que acontece com um servidor web: um mesmo servidor web serve páginas para inúmeros clientes simultâneos (navegadores web como o Google Chrome ou o Mozilla Firefox, por exemplo).

Contudo, existe um problema que devemos ter mente: conforme o número de conexões dos diversos clientes aumenta, o consumo de recursos computacionais do processo servidor também tende a aumentar. Isso varia em função do tipo de aplicação, mas, se a comunicação for intensa e o número de requisições aumentar de forma considerável, a máquina com o processo-servidor pode chegar ao limite da sua capacidade de processamento.

Nesse momento, entramos na questão do escalonamento: como aumentar a capacidade de atender a diversos clientes simultâneos? Existem, essencialmente, duas abordagens para a solução desse problema: a vertical e a horizontal (ENRIQUEZ e SALAZAR, 2018). Essas duas estratégias de escalonamento estão ligadas à forma como o servidor foi implementado.

Quando a arquitetura de um servidor (do software, não da máquina) permitir que o processo seja rodado simultaneamente em várias máquinas em paralelo e algum algoritmo distribuir as conexões dos clientes entre as diversas máquinas servidores, podemos utilizar o escalonamento horizontal (ENRIQUEZ e SALAZAR, 2018). A ideia é que as conexões sejam distribuídas de forma homogênea entre diversas máquinas e que o aumento no número de conexões possa ser compensado pelo aumento no número de máquinas servidoras.

O escalonamento vertical é um pouco mais simples: ele envolve, em geral, simplesmente fazer um *upgrade* no hardware da máquina que roda o processo-servidor,

buscando aumentar a sua capacidade de processamento e, consequentemente, aumentar a performance do processo.

Essas alterações na arquitetura de software tiveram impacto significativo na arquitetura dos sistemas operacionais e levaram ao desenvolvimento dos sistemas operacionais com arquitetura cliente-servidor. Muitas das funcionalidades que anteriormente eram providas por um kernel monolítico passaram a ser providas por processos em separados do kernel (algumas vezes, chamados de serviços).

Nesse tipo de arquitetura, as diversas funcionalidades do sistema operacional, como gerenciamento de arquivos e gerenciamento da memória, são divididas em processos-servidores diferentes, cada um com uma responsabilidade específica. Os programas que queiram utilizar essas funcionalidades devem ser escritos como programas-clientes desses servidores, e o kernel atua como uma camada na qual os objetivos são:

- proporcionar a comunicação entre os clientes e os servidores;
- garantir a segurança da máquina.

Esse padrão de arquitetura de software apresenta uma série de vantagens para o desenvolvimento de sistemas operacionais. Em primeiro lugar, torna possível separar o desenvolvimento dos diversos servidores, que podem ser feitos por equipes diferentes. Isso também facilita os testes, pois cada servidor tem uma responsabilidade específica que pode ser testada de forma isolada dos demais. Além disso, também permite que o usuário do sistema operacional desabilite funcionalidades que não lhe sejam necessárias, desabilitando ou parando a execução dos processos servidores.

2. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Os primeiros sistemas operacionais dos computadores pessoais eram "monolíticos", no sentido de que o kernel do sistema operacional tinha todas as chamadas de sistema para as suas diversas funcionalidades. Neles, o kernel provia e executava tudo, diferentemente do que ocorre nos sistemas operacionais mais modernos, que oferecem diversos serviços para prover as diversas funcionalidades. Ainda que essa abordagem seja simples de ser implementada, ela limita a complexidade do sistema operacional. Muitos desses sistemas nem podiam ser conectados a uma rede e necessitavam de produtos de terceiros para poder oferecer algum tipo de conectividade. Devido a tais limitações do ponto

de vista de arquitetura, esse não é o melhor tipo de sistema operacional para ser executado em uma "nuvem", como sugere o enunciado.

B e C – Alternativas incorretas.
JUSTIFICATIVA. A arquitetura em camadas foi uma evolução dos sistemas monolíticos e criou ao menos alguma estrutura em sistemas que, muitas vezes, não apresentavam nenhuma. Alguns aspectos dessa arquitetura ainda permanecem nos sistemas operacionais modernos, mas esse padrão, de forma pura, não é mais tão utilizado.

D – Alternativa correta.
JUSTIFICATIVA. A grande vantagem da arquitetura cliente-servidor, nesse caso, é podermos fazer com que clientes com pouca capacidade computacional (como um telefone celular, por exemplo) façam uso do processamento de máquinas com elevado poder computacional, os servidores. Esses servidores também podem ser escalonados, o que aumenta o poder de processamento global oferecido aos clientes.

E – Alternativa incorreta.
JUSTIFICATIVA. A arquitetura de máquinas virtuais poderia ser utilizada nos servidores, mas não necessariamente nos clientes. De qualquer forma, mesmo que se empreguem máquinas virtuais, é bastante provável que o sistema operacional usado apresente uma arquitetura do tipo cliente-servidor. Contudo, a existência de linguagens como a linguagem Java, em que os aplicativos são executados em máquinas virtuais (a JVM, ou *Java Virtual Machine*), em sistemas operacionais com arquitetura cliente-servidor, complica um pouco a definição da fronteira entre os tipos de arquiteturas.

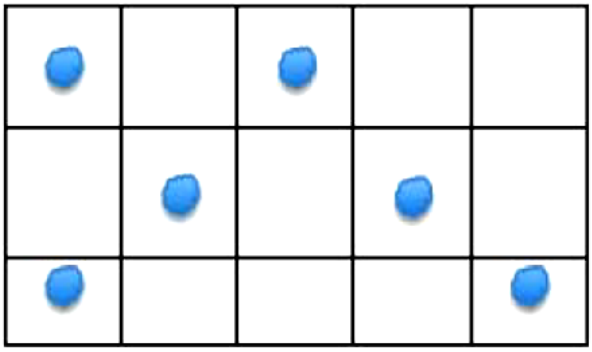
3. Indicações bibliográficas

- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice.* Boston: Pearson, 2003.
- ENRIQUEZ, R.; SALAZAR, A. *Software architecture with Spring 5.0.* Birmingham: Packt Publishing, 2018.
- TANENBAUM, A. S. *Sistemas operacionais modernos.* 6. ed. São Paulo: Prentice-Hall do Brasil, 2021.

Questões 2 e 3

Questão 2.[2]

A figura a seguir representa uma plantação de café com x colunas e y linhas. Nessa figura, as regiões com os cafeeiros plantados são as que estão com os espaços preenchidos, e as regiões com falhas de plantios são as que apresentam seus espaços vazios.



Considerando uma solução algorítmica para contar a quantidade de falhas de plantio do cafezal representado na figura, faça o que se pede nos itens a seguir.

a) Indique o tipo de estrutura de dados utilizada na solução, justificando a sua resposta.

b) Escreva o algoritmo da solução em pseudocódigo ou em linguagem de programação.

Questão 3.[3]

O coordenador geral de um comitê olímpico solicitou a implementação de um aplicativo que permita o registro dos recordes dos atletas à medida que forem sendo quebrados, mantendo a ordem cronológica dos acontecimentos e possibilitando a leitura dos dados a partir dos mais recentes.

Considerando os requisitos do aplicativo, a estrutura de dados mais adequada para a solução a ser implementadas é

A. o deque: tipo especial de lista encadeada, que permite a inserção e a remoção em qualquer das duas extremidades da fila e que deve possuir um nó com a informação (recorde) e dois apontadores, respectivamente, para os nós próximo e anterior.

B. a fila: tipo especial de lista encadeada, tal que o primeiro objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura FIFO (*First In - First Out*), a inserção e a remoção são feitas em extremidades contrárias e a estrutura deve

[2]Questão Discursiva 04 – Enade 2017.
[3]Questão 35 – Enade 2017.

possuir um nó com a informação (recorde) e um apontador, respectivamente, para o próximo nó.

C. a pilha: tipo especial de lista encadeada, na qual o último objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura LIFO (*Last In - First Out*), a inserção e a remoção são feitas na mesma extremidade e a estrutura deve possuir um nó com a informação (recorde) e um apontador para o próximo nó.
D. a fila invertida: tipo especial de lista encadeada, tal que o primeiro objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura FIFO (*First In - First Out*), a inserção e a remoção são feitas em extremidades contrárias e a estrutura deve possuir um nó com a informação (recorde) e um apontador, respectivamente, para o nó anterior.
E. a lista circular: tipo especial de lista encadeada, na qual o último elemento tem como próximo o primeiro elemento da lista, formando um ciclo, não havendo diferença entre o primeiro e último, e a estrutura deve possuir um nó com a informação (recorde) e um apontador, respectivamente, para o próximo nó.

1. Introdução teórica

1.1. Estruturas de dados

Segundo Cormen et al. (2012), uma estrutura de dados é "uma forma de armazenar e organizar dados com o objetivo de facilitar o seu acesso e modificação". Diversos livros, como os de AHO, HOPCROFT e ULLMAN (1983) e de CELES, CERQUEIRA e RANGEL (2004), por exemplo, conectam a ideia de estrutura de dados ao conceito de "Tipo Abstrato de Dados" (TAD), ou, em inglês, "Abstract Data Types" (ADT). Nessa técnica, procura-se definir as operações que o programador deve utilizar para trabalhar com a informação, sem que ele saiba fisicamente como a informação vai ser armazenada. Em outras palavras, define-se inicialmente uma interface para a manipulação da informação, sem que o usuário dessa interface saiba como ela será armazenada. Esse princípio é chamado de encapsulamento.

Contudo, em algum momento, o programador vai ter de implementar a maneira como os dados vão ser armazenados. No momento em que o programador vai implementar a interface do tipo abstrato de dados e definir a forma como os dados vão ser armazenados fisicamente na memória, o programador está implementando uma estrutura de dados.

Um dos aspectos fundamentais para a construção de um programa é a escolha de boas estruturas de dados para a resolução de dado problema. Uma boa escolha de estrutura de dados, associada a um bom algoritmo, leva a um programa que não só apresenta um bom desempenho, mas também tende a ter uma manutenção mais fácil. Uma escolha ruim, por sua vez, leva à situação oposta: desempenho ruim e surgimento de artifícios para se "adaptar" uma estrutura errada à solução de dado problema.

Felizmente, já existe uma série de estrutura de dados bem definidas e com características conhecidas que pode ser utilizada nas mais diferentes situações. Vamos explorar algumas dessas estruturas de dados básicas.

1.2. Lista ligada

A lista ligada é uma das estruturas de dados mais simples e uma das quais podemos utilizar como base para a implementação de diversas outras estruturas. Ela apresenta algumas similaridades com estruturas como o vetor, mas tem uma diferença importante: seus itens não são diretamente acessados por um índice, mas estão interligados por uma referência (ou um ponteiro) para outro elemento da lista (figura 1).

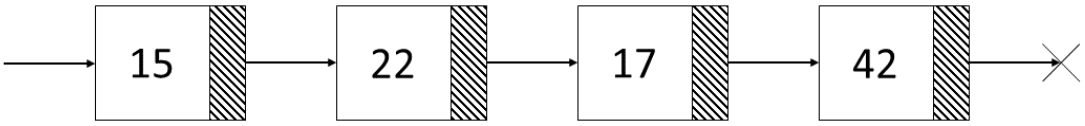


Figura 1. Exemplo de uma lista-ligada de números inteiros (não ordenada).

Em algumas linguagens, como na linguagem C, a diferença entre listas ligadas e vetores tem um significado especial: vetores são alocados de forma que todos os seus elementos são "vizinhos" de memória, ou seja, de forma que, dada a posição do primeiro elemento, somos capazes de calcular a posição em memória dos demais elementos, sabendo o tamanho do tipo ou a estrutura do vetor (considerando uma estrutura de dados homogênea, em que todos os elementos são do mesmo tipo de dados, como ocorre com os vetores na linguagem C).

Em uma lista ligada, essa obrigação de "vizinhança" na memória não é necessária. Isso significa que, ao alocarmos dinamicamente uma lista, não é necessário que os diversos elementos sejam vizinhos imediatos. O ponteiro para o próximo elemento garante que sejamos capazes de acessar o próximo elemento, mesmo que ele esteja em uma posição distante e não contínua de memória. Isso pode ser uma vantagem em situações nas quais

existe fragmentação da memória ou se quisermos evitar a alocação contínua de objetos de grandes dimensões na memória RAM.

Isso implica limitação das ligas ligadas simples: não podemos acessar seus elementos diretamente por um índice, como fazemos no caso de um vetor. Normalmente, para acessar um elemento que está "no meio", é necessário ir percorrendo a lista elemento por elemento, até encontrarmos o elemento desejado.

As listas ligadas (ou listas encadeadas) também apresentam algumas vantagens com relação à inserção e à remoção de elementos quando comparadas aos vetores. No caso de um vetor, a remoção ou a inserção de um item implica grande quantidade de movimentação de dados.

Para ilustrar a complexidade da retirada de um elemento de vetor, observe, na figura 2, as diversas operações que devem ser feitas. Se quisermos retirar o elemento 42, é necessário movermos os elementos seguintes (25, 33, 19 e 21) para as posições anteriores e diminuirmos o tamanho do vetor original (na linguagem C, isso pode ser feito com o comando realloc, mas deve-se tomar cuidado com questões de eficiência computacional).

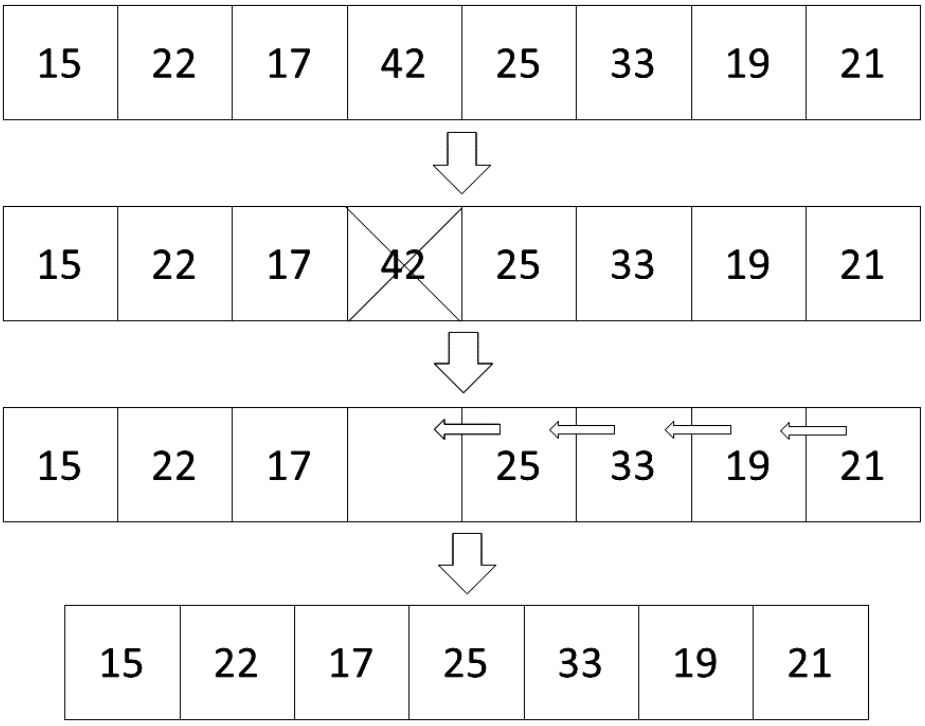


Figura 2. Retirada de um elemento de um vetor.

No caso de uma lista ligada, a operação de remoção de um elemento é muito mais simples, pois não requer a movimentação de outros elementos, apenas o ajuste de um ponteiro (ou referência) para um novo elemento, como ilustrado na figura 3.

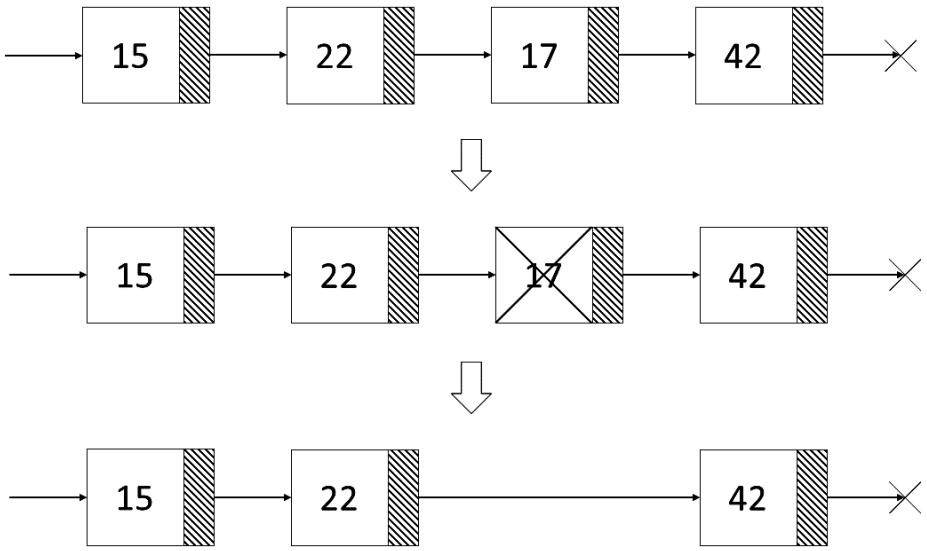


Figura 3. Retirada de um elemento de uma lista ligada simples.

A operação de inserção de um elemento em uma lista ligada apresenta o mesmo tipo de vantagem que a da remoção, em relação ao vetor. As vantagens e as desvantagens do uso de listas e de vetores devem ser cuidadosamente avaliadas pelo programador durante a implementação de um programa.

Existem vários outros tipos de listas, como as listas ligadas duplas, nas quais um elemento tem um ponteiro (ou referência) para o próximo e, também, um ponteiro para o elemento anterior. Uma característica desse tipo de lista é a possibilidade de inserção de elementos no seu início e no seu fim, uma vez que existem ponteiros em duas direções. Nesses casos, essas estruturas costumam ser denominadas deques. Outro tipo de estrutura de dado que é uma variação da lista ligada é a chamada lista ligada circular. Elas são similares às listas ligadas simples, mas o último elemento aponta para o primeiro, formando uma estrutura cíclica, por isso o nome "circular".

1.3. Pilha

A ideia da estrutura de dados do tipo pilha é similar ao conceito intuitivo de pilhas que encontramos no dia a dia. Normalmente, quando vamos a um restaurante e pegamos um prato em uma pilha, pegamos o prato que está no topo. Na versão computacional, o elemento que pode ser retirado da pilha sempre é o do topo, e tal disciplina no acesso é fundamental no uso dessa estrutura (na vida real, podemos retirar um prato que não está no topo da pilha e violar a ideia básica da estrutura de dados, mas não devemos fazer isso no mundo computacional).

Quando vamos adicionar um novo prato à pilha, ele é adicionado no topo. Se alguém retirar um prato da pilha logo depois da última adição, o último prato que foi colocado é o primeiro prato que vai ser retirado, e isso é sumarizado no princípio *LIFO – Last In, First Out*, ou, em português, "o último que entra é o primeiro que sai".

Assim como outras estruturas de dados, a pilha também tem um conjunto de operações típicas. As duas operações mais típicas são a de empilhar (*push*) e a de desempilhar (*pop*). Na operação de empilhar, um item é adicionado ao topo da pilha. Na operação de desempilhar, um item é retirado do topo da pilha. Observe que essas duas operações são fundamentais para garantir a disciplina de acesso aos elementos da pilha.

Outras operações comuns são as de verificação se a pilha está vazia, de verificação se a pilha alcançou algum tipo de limite máximo (dependendo da forma como a pilha for implementada, isso pode ser necessário) e de operação para a remoção automática de todos os elementos e a liberação da memória associada à pilha (o que também depende da linguagem na qual a pilha for implementada).

1.4. Fila

Assim como a pilha, a fila é uma outra estrutura de dados na qual o conceito fundamental é bastante comum no dia a dia. A ideia da fila, em programação, é similar à fila de pessoas para ser atendida em algum tipo de serviço (como um banco, por exemplo). As pessoas organizam-se de forma que o primeiro que chegou ao lugar torna-se o primeiro elemento da fila, o segundo que chegou torna-se o segundo elemento da fila, e assim por diante. Quando alguém puder ser atendido, o primeiro elemento da fila vai ser chamado, seguido pelo segundo etc. A ordem das pessoas na fila é a mesma ordem na qual vão ser atendidas. Por isso, dizemos que uma fila é uma estrutura do tipo *FIFO – First In, First Out*, ou, "o primeiro que entra é o primeiro que sai".

A fila é útil em situações nas quais temos um conjunto de requisições que podem chegar em velocidade maior do que a que podemos atender, sendo que queremos garantir disciplina no atendimento a essas requisições. Assim como na pilha, é necessária a existência de uma interface de funções (ou de métodos) que garantam a disciplina de acesso, ou seja, a inserção e a remoção de elementos, seguindo o princípio FIFO. Por exemplo, devem existir funções para inserir (ou enfileirar), que inserem um elemento sempre no fim da fila, e para desenfileirar (ou retirar) elementos, que retiram um elemento sempre no início da fila.

2. Resposta da questão e análise das alternativas

Questão 2

a) A estrutura do problema sugere uma solução no formato de uma matriz: cada pé de café ocupa um espaço quadrado e poderia ser interpretado como um elemento da matriz. A matriz poderia ser formada por números inteiros, sendo atribuídos o valor um para cada local em que existe um pé e o valor zero para cada local em que não há um pé plantado. Essa não é a melhor solução em termos de ocupação do espaço em memória RAM, mas atende aos requisitos do problema.

Outra possibilidade é utilizar uma lista ligada. Nessa abordagem, perdemos a informações sobre a geometria do problema (que é bidimensional), mas podemos ganhar eficiência de armazenamento. Por exemplo, podemos armazenar apenas a informação do pé que foi plantado e economizar espaço em memória dos locais em que não há plantas.

b) Podemos criar o código em C ilustrado na listagem 1.

```
1.    #include <stdio.h>
2.    #include <stdlib.h>
3.
4.    int main()
5.    {
6.        int MAT[3][5];
7.        Int lin = 0, col = 0, falha = 0;
8.        // inicializando a matriz
9.        For(lin = 0; lin < 3; lin++ )
10.           for(col = 0; col < 5; col++)
11.               MAT[lin][col] = 0;
12.       // colcar a informacao dos locais no qual existe uma
      planta
13.       MAT[0][0] = 1;
14.       MAT[0][2] = 1;
15.       MAT[1][1] = 1;
16.       MAT[1][3] = 1;
17.       MAT[2][0] = 1;
18.       MAT[2][4] = 1;
19.       // algoritmo para contar o numero de falhas de plantio
20.       For(lin = 0; lin < 3; lin++ )
21.           for(col = 0; col < 5; col++)
22.               if ( MAT[lin][col] == 0 )
23.                   falha++;
24.       printf("Total de falhas: %d \n", falha);
25.
26.       return 0;
27.   }
```

Listagem 1. Exemplo de programa que conta o número de falhas de plantio.

Na listagem 1, declaramos as variáveis nas linhas 6 e 7. Foi utilizada uma matriz de inteiros, com 3 linhas e 5 colunas, para armazenar a informação de onde foi feito o plantio. Depois, nas linhas entre 9 e 11, a matriz foi inicializada com zeros, o que representa as regiões vazias. Da linha 13 até a 18, foi atribuído um a todas as posições nas quais existe um pé de café plantado. Da linha 20 até a 13, foi feita uma estrutura com dois laços aninhados, que percorre cada elemento da matriz e compara-o a zero. Se essa comparação for verdadeira, incrementamos a variável falha, que conta o total de falhas de plantio. Observe que essa variável é inicializada com zero. Na linha 24, o conteúdo da variável falha é mostrado na tela, após termos computado o total de falhas.

Questão 3

A – Alternativa incorreta.
JUSTIFICATIVA. Como a estrutura deve manter a ordem cronológica e permitir a leitura dos dados a partir dos mais recentes, a estrutura do tipo deque não é a mais adequada, pois ela permite que os dados sejam lidos, inseridos e retirados tanto no seu início quanto no seu fim, o que violaria a ordenação dos dados.

B – Alternativa incorreta.
JUSTIFICATIVA. Como o problema pede que os dados sejam lidos a partir dos mais recentes, o mecanismo de leitura não é o da fila, em que o primeiro a ser lido foi o primeiro a ser inserido, e não o último (FIFO).

C – Alternativa correta.
JUSTIFICATIVA. A pilha é a estrutura mais adequada para o problema, pois ela garante que os dados vão ser lidos na ordem inversa na qual foram inseridos (LIFO), de forma que o último elemento que foi inserido vai ser o primeiro a ser retirado, como solicitado no enunciado.

D – Alternativa incorreta.
JUSTIFICATIVA. A estrutura descrita não resolve o problema, pois é uma estrutura do tipo FIFO, com inserção e remoção em extremidades contrárias.

E – Alternativa incorreta.

JUSTIFICATIVA. O enunciado deixa claro que existe a necessidade de ordenação dos dados na estrutura: os recordes devem ser registrados na ordem em que forem quebrados, e a leitura deve ser feita a partir dos dados mais recentes. Em uma lista circular, a possibilidade de leitura "cíclica" dos dados violaria essa regra.

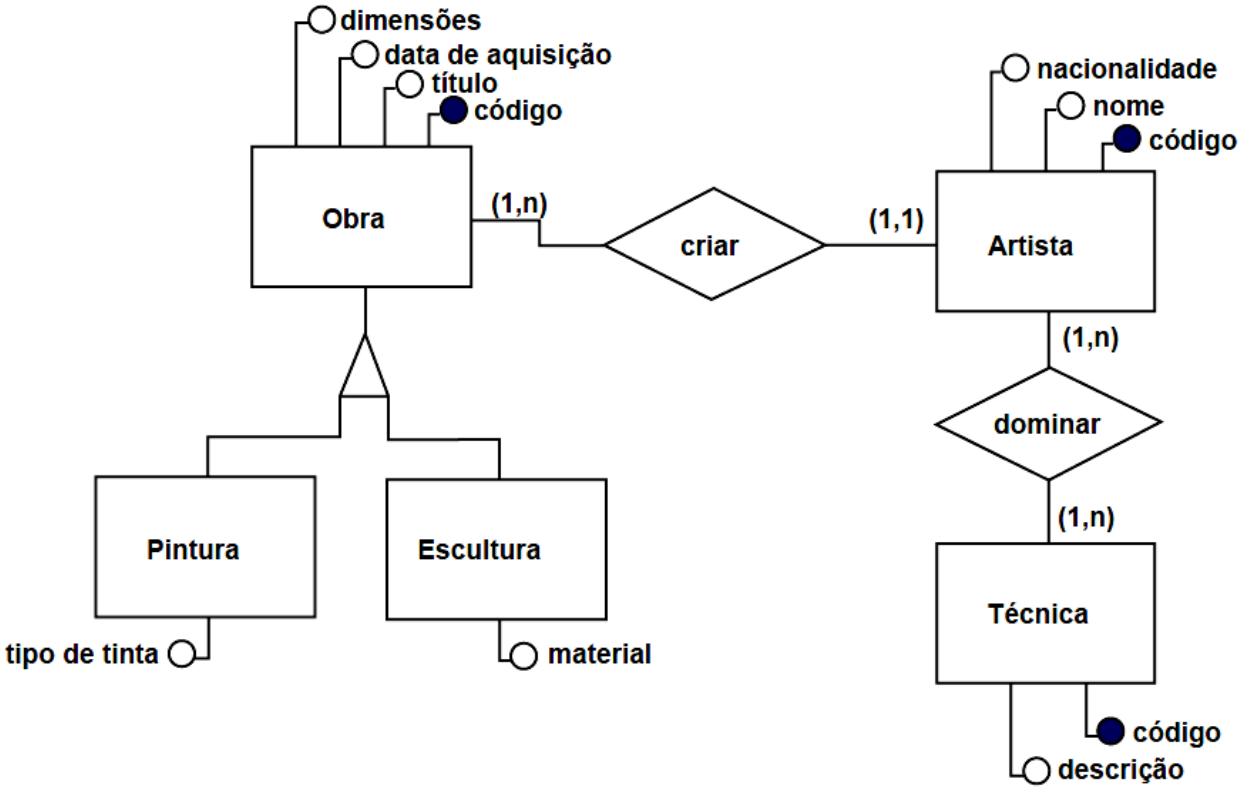
2. Indicações bibliográficas

- AHO, A.; HOPCROFT, J.; ULLMAN, J. *Data structures and algorithms*. London: Addison-Wesley, 1983.
- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. *Introdução à estrutura de dados com técnicas de programação em C.* Rio de Janeiro: Elsevier, 2004.
- CORMEN, T. et al. *Algoritmos: teoria e prática*. 3. ed. Rio de Janeiro: Campus-Elsevier, 2012.
- GOODRICH, M. T.; TAMASSIA, R.; MOUNT, D. *Data structures & algorithms in C++*. 2. ed. Hoboken: John Wiley & Sons, 2011.

Questões 4 e 5

Questão 4.[4]

O diagrama Entidade-Relacionamento (DER) a seguir apresenta a modelagem conceitual de dados de um sistema de informação para um museu. A partir dessa modelagem, observa-se o seguinte: uma Obra é criada por um único Artista e um Artista pode criar no mínimo uma Obra e no máximo várias Obras; as entidades Pintura e Escultura são especializações da entidade Obra; um Artista tem o domínio de várias Técnicas, assim como uma Técnica é dominada por diversos Artistas.



Com base nas regras de mapeamento que transformam o Modelo Conceitual em um Modelo Lógico Relacional, avalie as afirmações a seguir, a respeito do Esquema Lógico Relacional gerado a partir do DER apresentado.

I. No Esquema Lógico Relacional, haverá uma tabela associativa, criada em função do relacionamento muitos para muitos entre as entidades Artista e Técnica, que terá uma chave primária composta pelo código do artista e o código da técnica.

II. No Esquema Lógico Relacional, haverá uma tabela Artista na qual o atributo código do artista será a chave primária da tabela, e o código da obra será uma chave estrangeira que fará referência a uma obra existente na tabela Obra.

[4]Questão 11 – Enade 2017.

III. No Esquema Lógico Relacional, haverá, em função da generalização/especialização, uma tabela Obra com apenas os atributos código da obra, título, data de aquisição e dimensões, e duas outras tabelas: a tabela Pintura, com apenas o atributo tipo de tinta, e a tabela Escultura com apenas o atributo material.

É correto o que se afirma em

A. I, apenas.

B. II, apenas.

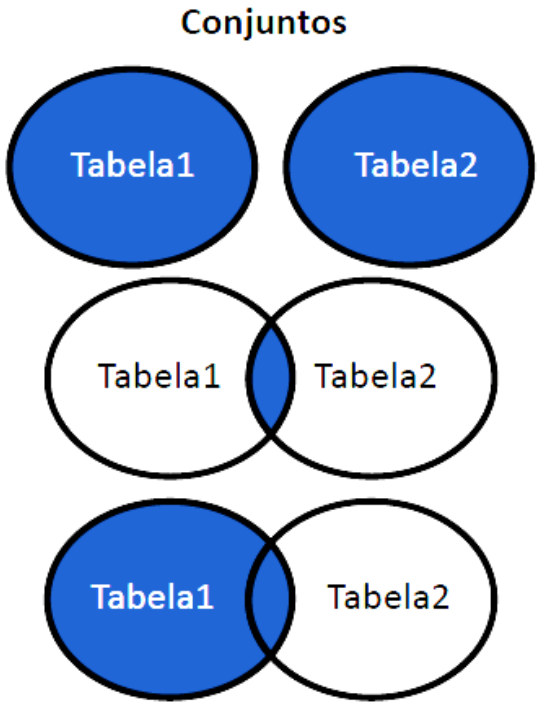
C. I e III, apenas.

D. II e III, apenas.

E. I, II e III.

Questão 5.[5]

No modelo de dados relacional, uma junção entre tabelas cria uma pseudotabela derivada de duas ou mais tabelas de acordo com os critérios especificados, similar às regras da teoria dos conjuntos. Nos conjuntos a seguir, as áreas hachuradas representam os resultados das consultas SQL (padrão) em que, id é uma chave primária e fk uma chave estrangeira.



Consultas SQL equivalentes

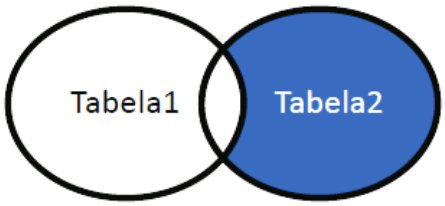
```
SELECT * FROM Tabela1;
SELECT * FROM Tabela2;
```

```
SELECT * FROM Tabela1 t1;
INNER JOIN Tabela2 t2
ON t1.id = t2.fk;
```

```
SELECT * FROM Tabela1 t1;
LEFT JOIN Tabela2 t2
ON t1.id = t2.fk;
```

Assinale a opção em que a declaração SQL é equivalente ao conjunto apresentado a seguir.

[5]Questão 23 – Enade 2017.



A. SELECT * FROM Tabela1 t1 JOIN Tabela2 t2 ON t1.id = t2.fk;
B. SELECT * FROM Tabela1 t1 RIGHT JOIN Tabela2 ON t1.id = t2.fk WHERE t2.fk <> NULL;
C. SELECT * FROM Tabela1 t1 WHERE EXISTS (SELECT 1 FROM Tabela2 t2 WHERE t2.id = t1.fk);
D. SELECT * FROM Tabela1 t1 RIGHT JOIN Tabela2 t2 ON t1.id = t2.fk WHERE t1.id is NULL;
E. SELECT * FROM Tabela1 t1 WHERE NOT EXISTS (SELECT 1 FROM Tabela2 t2 WHERE t2.id = t1.fk);

1. Introdução teórica

1.1. Projetos de banco de dados e seus modelos

A construção de um sistema que utiliza banco de dados relacionais abarca diversas etapas. As primeiras etapas envolvem o entendimento do problema de uma forma mais abstrata, na busca de compreender o contexto no qual o usuário vive. Isso leva à construção de modelos que estão mais voltados à captura do domínio do problema do que à captura dos detalhes de implementação da solução. Na computação, existem diversas abordagens que foram desenvolvidas com esse objetivo, cada uma com um foco um pouco diferente.

Em sistemas cujo projeto é focado no banco de dados, é comum que se comece o processo de modelagem pela construção de um modelo conceitual. Esse modelo apresenta caráter abstrato, no sentido de que ele procura descrever a estrutura dos dados do problema de forma independente da tecnologia (o sistema de gerenciamento de banco de dados específico) que vai ser utilizada para construir a solução. Um dos modelos mais comuns utilizados com essa finalidade é o modelo de entidade-relacionamento (MER).

No modelo de entidade-relacionamento, busca-se descrever não apenas quais os dados que devem ser armazenados no sistema, mas qual é a relação entre esses dados. Um

modelo de entidade-relacionamento é composto de três elementos básicos: entidades, atributos e relacionamentos.

Quando o projetista inicia o trabalho de modelagem, ele começa escolhendo quais elementos do domínio do usuário são relevantes para o sistema, as entidades. Frequentemente, as entidades são definidas como objetos relacionados ao mundo que está sendo modelado, mas, aqui, deve-se tomar cuidado com essa palavra. Em primeiro lugar, porque não estamos nos restringindo necessariamente a objetos físicos, e sim a um aspecto do problema que é relevante para o modelo do sistema. Em segundo lugar, o nome objeto também é utilizado em outra abordagem, a chamada orientação a objetos, com um significado um pouco diferente. Heuser (2009) define entidade como "um conjunto de objetos da realidade modelada".

Como exemplo de entidades, na modelagem de um sistema para uma universidade, alunos e professores podem ser entidades que têm um significado físico direto, assim como cursos e matérias, que têm um significado mais abstrato. Essas entidades são os elementos dos quais queremos manter informações armazenadas no banco de dados.

Entidades podem conter atributos, que caracterizam a entidade e representam quais aspectos dela devem ser armazenados e utilizados pelo sistema. Por exemplo, em dado sistema, pode existir uma entidade cliente que tem como atributos um nome, um CPF e um endereço, além de outros atributos relevantes ao problema em questão.

Finalmente, é bastante provável que, na maioria dos sistemas, as entidades estejam, de alguma forma, relacionadas entre si. Assim, um modelo adequado do domínio do problema não deve conter apenas as entidades "soltas", mas também deve conter informações sobre como essas entidades devem relacionar-se. Essas informações são incorporadas pelos relacionamentos. Além de dizer quais entidades se relacionam, os relacionamentos também armazenam informações sobre a "cardinalidade" do relacionamento: eles quantificam como uma entidade se relaciona com outras entidades. Por exemplo, para relações binárias entre dois conjuntos, podemos encontrar as seguintes cardinalidades: um-para-um, um-para-muitos, muitos-para-um e muitos-para-muitos.

O diagrama de entidade e relacionamento foi inicialmente proposto por Peter Chen em 1976 (LIU, 2012) e apresenta as convenções gráficas a seguir.

- Entidades são desenhadas como retângulos.
- Um diamante (ou losango) representa um relacionamento entre entidades.
- Atributos são desenhados como formais ovais ou como pequenas circunferências ao redor de entidades ou ao redor dos atributos.

Uma vez que o modelo conceitual tenha sido feito, capturando os principais aspectos do domínio do problema de forma independente do banco de dados, segue-se para a construção do modelo lógico. Nesse caso, o objetivo é criar um novo modelo baseado no modelo conceitual, mas que também incorpore as informações sobre a tecnologia que vai ser escolhida para a construção do sistema. No caso dos sistemas de banco de dados relacionais (SGBDs), esse modelo foca na criação das tabelas, das suas colunas e de outros aspectos relacionados a um banco de dados relacional específico.

1.2. Consultas a um banco de dados relacional

As informações armazenadas no banco de dados podem ser consultadas por meio de programas escritos na linguagem SQL. Utiliza-se o comando select para isso, juntamente com as cláusulas from e where.

As ideias gerais desse comando são:

- a cláusula select informa quais colunas devem ser selecionadas;
- a cláusula from informa de quais tabelas as informações vão ser retiradas;
- a cláusula where impõe diversas condições (tecnicamente chamadas de predicados) que devem ser satisfeitas pelos dados selecionados.

Normalmente as consultas são feitas envolvendo não apenas uma única tabela, mas um conjunto de tabelas. Contudo, o objetivo do comando select é retornar dados no formato de uma única "tabela virtual", ou seja, o comando select seleciona os dados da forma especificada na consulta e apresenta-os como se fossem uma tabela do banco de dados. Por esse motivo, quando o comando select opera em várias tabelas simultâneas, ele deve ser capaz de sintetizar uma única tabela a partir dos dados vindos de diversas tabelas diferentes.

A forma como essa tabela sintética, resultado do comando select, é feita depende do tipo de "junção" (*join*) especificado na consulta. Sem a utilização de um tipo específico de junção (*join*) e sem nenhuma restrição na cláusula where, o comando select simplesmente faz o produto cartesiano dos dados presentes em todas as tabelas, ou seja, gera uma tabela que é o conjunto de todas as combinações possíveis dos dados presentes nas tabelas definidas na consulta.

Infelizmente, o produto cartesiano cria uma quantidade grande de combinações, muitas das quais sem nenhum significado para o negócio. A utilização de junções permite

restringir o resultado com a utilização de chaves ou de outros campos das tabelas envolvidas e do relacionamento entre essas tabelas.

A junção natural (*natural join*) é um tipo de junção que seleciona apenas os dados correspondentes aos campos (colunas) em comum às várias tabelas envolvidas na consulta. Linhas que não tenham correspondências em qualquer uma das tabelas são eliminadas do resultado. Dessa forma, muita informação pode ser "eliminada" desse tipo de consulta.

Para aliviar esse problema, quando trabalhamos com duas tabelas, podemos utilizar o "outer join". Ao utilizarmos esse tipo de junção, controlamos qual informação vai ser adicionada e qual informação vai ser retirada do resultado.

No "left outer join", todos os registros da tabela esquerda são adicionados ao resultado, independentemente de terem correspondentes na tabela da direita. Se essas informações não existirem, as colunas correspondentes vão ser impressas com o valor NULL.

O "right outer join" opera de forma equivalente, porém, nesse caso, são os registros da tabela da direita que devem ser sempre apresentados, e é na coluna correspondente à tabela da esquerda que os resultados não existentes serão impressos como NULL.

Finalmente, o "full outer join" opera pela incorporação de todos os resultados das tabelas da esquerda e da direita, mostrando NULL sempre que o registro não tiver equivalência em ambas as tabelas.

2. Análise das afirmativas e solução da questão

Questão 4

I – Afirmativa correta.

JUSTIFICATIVA. Sabemos que um artista pode dominar várias técnicas diferentes (por exemplo, um mesmo artista pode ser um pintor de quadros e um escultor). Contudo, dada técnica pode ser dominada por vários artistas (existem vários escultores e vários pintores diferentes). Aqui, é importante termos em mente um aspecto da modelagem do enunciado: a técnica não está diretamente associada ao artista como um atributo. Uma modelagem ingênua poderia colocar a técnica como um atributo do artista, mas isso geraria muitos problemas: por exemplo, o que fazer com artistas que dominam várias técnicas diferentes? Ao criarmos duas entidades diferentes (Artista e Técnica), que se tornam tabelas diferentes referenciadas por uma tabela associativa, somos capazes de modelar muitos mais casos de forma mais natural (em vez de adicionarmos "mais colunas" na tabela Artista, por exemplo).

II – Afirmativa incorreta.

JUSTIFICATIVA. Podemos perceber, no modelo apresentado, que um artista pode criar uma ou mais obras. Cada obra tem uma chave estrangeira para a tabela Artista, o que pode gerar um problema: o modelo não admite obras criadas por vários artistas. No entanto, a tabela Artista não deve conter uma chave estrangeira para a tabela Obra, pois um artista pode criar várias obras diferentes.

III – Afirmativa incorreta.

JUSTIFICATIVA. Existem duas abordagens para fazermos o mapeamento no modelo relacional:

- uma na qual criamos tabelas tanto para as entidades mais genéricas quanto para as entidades mais específicas;
- outra na qual criamos tabelas apenas para as mais específicas.

De qualquer forma, em ambas as abordagens, devemos incluir, pelo menos, a chave primária nas tabelas de Escultura e Pintura (código da obra). Observe que isso não é mencionado como atributo na afirmativa, e, portanto, a afirmativa está errada.

Alternativa correta: A.

Questão 5

Queremos selecionar apenas os registros que estão na Tabela2, excluindo os dados que tenham algum tipo de relacionamento com a Tabela1. Observe que isso é diferente de um RIGHT OUTER JOIN, no qual todos os dados da Tabela2 seriam retornados, ou, em uma situação análoga, se fizéssemos apenas um select sem condições na Tabela2. De acordo com o diagrama do enunciado, somos obrigados a eliminar os registros que tenham algum relacionamento com a Tabela1. Esses registros são aqueles que têm uma chave estrangeira fk válida, ou seja, diferente de NULL. Dessa forma, sabemos o que segue.

- Precisamos fazer uma consulta que utilize um RIGHT JOIN entre as tabelas Tabela1 e Tabela2.
- Todos os registros selecionados da Tabela2 devem ter a chave estrangeira fk igual a NULL.
- Devemos selecionar todas as colunas disponíveis, o que é feito pelo símbolo *.

Assim, a consulta obtida é: "SELECT * FROM Tabela1 t1 RIGHT JOIN Tabela2 t2 ON t1.id=t2.fk WHERE t1.td is NULL". Nessa consulta, "*" significa que todas as colunas disponíveis são retornadas, "FROM Tabela1 t1" indica que a tabela da esquerda é a Tabela1, e que essa vai ser temporariamente chamada de t1 (apenas dentro da consulta). O "RIGHT JOIN Tabela t2" estabelece qual vai ser a tabela da direita (Tabela2) e seu nome temporário na consulta (t2). Já "ON t1.id=t2.fk" indica o relacionamento entre as duas tabelas, enquanto "WHERE t1.td is NULL" indica a condição que vai excluir os registros da Tabela2 que apresentam relacionamento com a tabela Tabela1.

Alternativa correta: D.

3. Indicações bibliográficas

- HEUSER, C. A. *Projeto de banco de dados.* 6. ed. Porto Alegre: Bookman, 2009.
- LIU, H. H. *Oracle Database Performance and Scalability: A Quantitative Approach*. Hoboken: Wiley, 2012.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Campus-Elsevier, 2012.

Questões 6 e 7

Questão 6.[6]

O Guia do Conhecimento em Gerenciamento de Projetos (*Project Management Body of Knowledge* – PMBOK) inclui o gerenciamento de riscos como uma das áreas de conhecimento que devem ser observadas para o sucesso do projeto. Os riscos identificados são analisados em função da probabilidade de sua ocorrência e do impacto que causarão ao projeto, caso ocorram.

A respeito da definição de risco em um projeto, de acordo com as orientações do PMBOK, avalie as afirmativas.

I. A maioria dos projetos conduzidos pela gerência não apresenta nenhum risco associado.
II. O impacto do risco, caso ocorra, pode ser positivo ou negativo.
III. Devem ser desconsiderados os riscos que não afetam o objetivo do projeto.
IV. O valor estimado referente ao impacto deve ser diferente de zero.

É correto apenas o que se afirma em

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

Questão 7.[7]

O Guia PMBOK (*Project Management Body of Knowledge*) orienta a gestão de projetos considerando que processos devem ser executados nas diversas fases do gerenciamento, a saber: iniciação, planejamento, execução e encerramento.

Em determinado projeto, foi desenvolvido um sistema segundo o ciclo de vida em cascata, cujas atividades são realizadas sequencialmente e sem sobreposição. Na atividade de homologação do sistema, às vésperas da data planejada de sua implantação, um usuário identificou a inviabilidade de uma funcionalidade testada, pois alguns requisitos básicos não haviam sido considerados durante o levantamento dos requisitos. A alteração que se mostrou necessária atrasou o projeto em relação à linha de base do plano do projeto.

Segundo o PMBOK, pode-se considerar que a falha foi originada na fase de

A. iniciação ou planejamento, em um processo da área de conhecimento gestão do tempo.
B. planejamento ou execução, em um processo da área de conhecimento gestão do tempo.
C. execução ou encerramento, em um processo da área de conhecimento gestão do tempo.
D. iniciação ou encerramento, em um processo da área de conhecimento gestão do escopo.
E. planejamento ou execução, em um processo da área de conhecimento gestão do escopo.

[6]Questão 10 - Enade 2017 (com adaptações).
[7]Questão 28 - Enade 2017.

1. Introdução teórica

1.1. Áreas de conhecimento do guia PMBOK

Na sexta edição do guia PMBOK® (PMI, 2017), são encontradas dez áreas de conhecimento e descrições de seus respectivos gerenciamentos.

Essas áreas são: integração, escopo, tempo, custos, qualidade, recursos, comunicação, risco, aquisições e partes interessadas.

Devemos destacar que, à época da aplicação do Enade 2017, ainda não havia a sétima edição do guia PMBOK®, lançado em 2021.

1.1.1. Gerenciamento do escopo do projeto

Os processos ligados ao gerenciamento do escopo têm dois objetivos básicos:

- o primeiro é definir todo trabalho necessário para o sucesso do projeto;
- o segundo é garantir que seja executado apenas o trabalho necessário para o sucesso do trabalho, sem a execução de nenhum trabalho adicional.

Dessa forma, o primeiro objetivo busca assegurar que o projeto não exclua tarefas necessárias, enquanto o segundo objetivo tenta limitar essas tarefas apenas ao mínimo necessário, a fim de que não exista desperdício de tempo e recursos. Queremos identificar o que é essencial e garantir que apenas isso seja feito.

O PMBOK define seis processos necessários para o gerenciamento de escopo, como veremos a seguir.

O primeiro processo é chamado de planejamento do escopo e envolve identificar o que é necessário para que o escopo seja definido e gerenciado da forma mais correta possível. Como a definição do escopo tem impacto muito grande no projeto como um todo, é fundamental garantir que todas as atividades necessárias à sua definição e ao seu gerenciamento sejam feitas de maneira adequada.

O segundo processo envolve descobrir "o que" deve ser feito. As pessoas, os grupos ou as instituições que contrataram o projeto (as partes interessadas) têm necessidades específicas. Além disso, algumas dessas necessidades podem ser fruto de aspectos regulatórios, vindos de leis e normas, por exemplo. Descobrir o que deve ser feito engloba identificar e registrar as necessidades, ou seja, identificar os requisitos do projeto.

O terceiro processo está ligado à definição do escopo propriamente dito. Observe que o primeiro processo envolve o planejamento do que é neccessário para a definição do escopo, enquanto o terceiro processo envolve a própria definição do escopo.

Projetos reais podem ser bastante complexos, incluindo muitas tarefas e atividades diferentes, com graus variados de complexidade. Para facilitar o gerenciamento, é interessante dividir um projeto em partes menores, com a intenção de determinar o que deve ser feito e o que deve ser entregue. A ideia é que essas partes menores sejam mais fáceis de serem gerenciadas. Esse processo de subdivisão é feito por meio do conhecemos como EAP ("Estrutura Analítica do Projeto"): o quarto processo é aquele no qual se cria a EAP.

O quinto processo e o sexto processo são aqueles em que é fazemos a verificação (ou a validação) do escopo e em que realizamos o seu controle. Na verificação, podem ser feitas inspeções ou revisões das entregas, com o objetivo de decidir se elas estão de acordo com critérios de aceitação para assim, finalmente, formalizarmos as aceitações. O controle do escopo lida com um problema complexo, mas, muitas vezes, inevitável em muitos projetos: gerenciar as mudanças do escopo, que podem ocorrer por muitos motivos durante a sua execução.

1.1.2. Gerenciamento do tempo do projeto

Além de definirmos e gerenciarmos o que deve ser feito em um projeto, também devemos nos preocupar com os prazos. O guia PMBOK® estabelece uma série de processos com relação ao gerenciamento do tempo de um projeto. Assim como é feito de maneira geral, primeiramente se define o planejamento do gerenciamento do cronograma. Isso é necessário porque, em projetos reais, o cronograma, além de ser complexo, é atualizado e mantido ao longo do projeto. Precisamos planejar e definir como serão feitas a criação e a manutenção do cronograma ao longo do projeto.

Para que seja possível fazer e manter um cronograma, é necessário identificar quais atividades devem ser realizadas (processo de definição das atividades). Também necessitamos identificar a relação entre as diversas atividades e a sua ordenação (processo de sequenciamento das atividades).

Além da definição das atividades e do seu sequenciamento, é importante identificar quanto tempo elas devem levar para serem executadas, o que é feito no processo de estimação das durações das atividades. Todos esses processos dão origem a diversas

informações (atividades, sequenciamento e duração) fundamentais para o processo do desenvolvimento do cronograma. Finalmente, o processo de controle do cronograma, de forma similar ao processo de controle do escopo, tem como objetivo controlar e influenciar o resultado das mudanças ocorridas ao longo do projeto.

1.1.3. Gerenciamento de riscos em um projeto.

Segundo a sexta edição do guia PMBOK® (PMI, 2017), o risco do projeto é um evento ou uma condição que, se ocorrer, terá efeito sobre um ou mais objetivos do projeto. Esse efeito pode ser positivo (oportunidade) ou negativo (ameaça) e pode incidir no cronograma, no custo, no escopo ou na qualidade. Um risco pode ter uma ou mais causas e, se ocorrer, um ou mais impactos. Observa-se que os eventos que não impactam no projeto não são considerados riscos.

Os riscos podem ser conhecidos (foram identificados, analisados e considerados no planejamento do projeto) ou desconhecidos (não foram considerados no projeto inicial).

No caso dos riscos desconhecidos, quando o evento ocorre, eles são considerados problemas ou questões para o projeto e devem ser resolvidos rapidamente. O gerente do projeto deve tomar as devidas ações corretivas, identificar as causas e tomar medidas preventivas para que o problema não ocorra novamente. Além disso, as decisões tomadas devem ser documentadas, e os responsáveis devem ser notificados.

Com relação ao gerenciamento de riscos, de acordo com a sexta edição do Guia PMBOK®, são necessárias as etapas a seguir.

- Planejar o gerenciamento dos riscos.
- Identificar os riscos.
- Realizar a análise qualitativa dos riscos.
- Realizar a análise quantitativa dos riscos.
- Planejar respostas aos riscos.

No que concerne à análise quantitativa dos riscos, o objetivo é calcular o seu valor monetário. Para tal, são utilizadas árvores de decisão e diagramas de impacto e são feitas simulações para ajudar na análise dos seus impactos. Nesse aspecto, os efeitos (impactos) são analisados numericamente. De qualquer maneira, se há risco, existe um valor que deve ser considerado no projeto, referente ao impacto desse risco.

Na sexta edição, foi adicionada uma nova estratégia de resposta aos riscos, a "escalonar". Isso significa que os riscos devem ser escalonados em nível de programa ou de portfólio.

2. Análise das afirmativas e das alternativas

Questão 6

I – Afirmativa incorreta.
JUSTIFICATIVA. Qualquer projeto está sujeito a certo risco, pois nem todos os elementos podem estar sob controle da gerência do projeto.

II – Afirmativa correta.
JUSTIFICATIVA. Os riscos em um projeto podem ter efeito positivo (oportunidade) ou efeito negativo (ameaça).

III – Afirmativa correta.
JUSTIFICATIVA. Se não há interferência no objetivo do projeto, a probabilidade de ocorrer um evento não pode ser considerada um risco.

IV – Afirmativa correta.
JUSTIFICATIVA. Se existe a chance de o evento ser risco, deve ser feita a análise quantitativa do seu impacto. Caso o impacto financeiro resultante dessa análise seja zero, o evento não deve ser considerado um risco. Contudo, é interessante não nos restringirmos apenas aos impactos financeiros de um projeto, especialmente se considerarmos as incertezas referentes aos valores vindos da análise.

Alternativa correta: E.

Questão 7

A, B e C – Alternativas incorretas.

JUSTIFICATIVA. O problema não está ligado à área de conhecimento da gestão do tempo, mas à área da gestão do escopo, uma vez que alguns requisitos não foram identificados, e essa falha levou ao atraso.

D – Alternativa incorreta.

JUSTIFICATIVA. A falha na identificação de um requisito não corresponde a um problema nas fases de iniciação ou de encerramento, ainda que esteja relacionada com a área de conhecimento da gestão do escopo do projeto.

E – Alternativa correta.

JUSTIFICATIVA. A falha pode ter sido originada na fase planejamento e não ter sido detectada (ou comunicada) na fase de execução. Como a falha envolve a não identificação de um requisito, sabemos que ela está relacionada com a área de conhecimento da gestão do escopo de um projeto.

3. Indicações bibliográficas

- MONTES, E. *Introdução ao gerenciamento de projetos*. São Paulo: Create Space, 2017.
- MONTES, E. *Gerenciamento dos riscos do projeto.* Disponível em <https://escritoriodeprojetos.com.br/gerenciamento-dos-riscos-do-projeto>. Acesso em 11 jul. 2025.
- OLIVEIRA, G. 6ª *Edição do Guia PMBOK – O que mudou?* Disponível em <https://blog.softexpert.com/pmbok-6a-edicao/>. Acesso em 11 nov. 2019.
- PINHEIRO, F. *Resumo com pontos mais importantes para os exames PMP® e CAPM®.* Disponível em <http://www.tiexames.com.br/PMP_PREP5/anexos/PMP_CAPM_Resumo.pdf>. Acesso em 11 nov. 2019.
- PROJECT MANAGEMENT INSTITUTE – PMI. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK®).* 6. ed. Atlanta: PMI, 2017.

Questão 8

Questão 8.[8]

Leia o texto a seguir.

Nos últimos anos, a melhoria do desempenho organizacional, por meio da identificação, da avaliação e da melhoria dos processos de negócios organizacionais, tem-se tornado uma prática padrão nas organizações ao redor do mundo e tem sido reconhecida como uma disciplina denominada Gestão de Processos de Negócios.
O primeiro passo da gestão de processos é identificar que processo de negócio se deseja melhorar. Sem clarificar exatamente qual é o processo, torna-se difícil controlar o escopo de um projeto de melhoria de processos, pois, em qualquer organização, todas as atividades estão, de alguma forma, relacionadas.

SHARP, A.; MCDERMOTT, P. *Wokflow modeling.* 2. ed. Boston: Artech House, 2009 (com adaptações).

Considerando as informações apresentadas, assinale a opção que corresponde a um processo de negócio que deve ser selecionado para análise pelo projeto de melhoria de processos de uma organização.

A. Criar contas de clientes.
B. Prospectar novos clientes.
C. Comércio eletrônico via *web*.
D. Relacionamento com clientes.
E. Calcular a faixa de crédito de clientes.

1. Introdução teórica

Processo de negócio

Um processo de negócio (processo organizacional ou método de negócio) é uma sequência de atividades iniciadas a partir de uma demanda, com o objetivo de entregar algum resultado.

O processo de negócio é um conjunto de atividades ou de tarefas que são estruturadas e giram em torno da produção de um resultado de valor para o cliente, por meio da entrega de um serviço ou de um produto. Um processo de negócio mostra o que, como e quem é o responsável por realizar alguma tarefa.

Isso significa que o processo de negócio determina:

- o trabalho a ser executado;
- o modo de o trabalho ser feito na organização;
- a sequência lógica das atividades a serem executas;

[8]Questão 25 - Enade 2017.

- o valor entregue ao cliente ou a outros processos.

No processo de negócio, os insumos (materiais, conhecimento e outros) são transformados em resultados (produtos e serviços).

É importante saber a distinção entre tarefa, atividade e processo. A tarefa (nível operacional) é o que é executado, a atividade (nível tático) é um conjunto de tarefas que são executadas para determinado fim, e o processo (nível estratégico) é uma combinação de atividades cujo objetivo é a entrega de um resultado.

As organizações são sistemas compostos por uma coleção de processos de negócio. Esses processos são classificados em:

- primários;
- de suporte;
- de gerenciamento.

São exemplos de processos primários: desenvolvimento de produtos, marketing, produção, logística e serviços de pós-vendas.

São exemplos de processos de suporte: gestão de recursos humanos, gestão de infraestrutura de TI e gestão de estoques.

São exemplos de processos de gerenciamento: governança corporativa, gestão estratégica e gestão de desempenho.

Partindo do princípio de que todos os processos podem ser melhorados, a melhoria de processos pode ser definida como a análise para determinação das ineficiências existentes em uma empresa, a indicação das causas da ineficiência e a sugestão dos meios para correção.

A melhoria de processos é fundamental para a competitividade da empresa. Para que as empresas se mantenham competitivas, é imprescindível que elas façam investimento na melhoria de seus processos.

2. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Criar a conta do cliente é uma tarefa executada em um processo que está estabelecido na empresa. Para criar a conta do cliente, é necessário que processo de inserção dos dados tenha sido estabelecido.

B – Alternativa correta.
JUSTIFICATIVA. Para que seja feita a prospecção de novos clientes, devemos estabelecer como esse processo será executado.

C – Alternativa incorreta.
JUSTIFICATIVA. O comércio eletrônico é uma forma de oferecer os produtos (ou serviços) de uma empresa. O comércio eletrônico é uma atividade.

D – Alternativa incorreta.
JUSTIFICATIVA. O relacionamento com os clientes é a forma com que a empresa interage com seus clientes. Essa atividade deve fazer parte do processo de gerenciamento de relacionamento com os clientes (CRM).

E – Alternativa incorreta.
JUSTIFICATIVA. Calcular a faixa de crédito de um cliente é uma atividade cujo objetivo é determinar o quanto a empresa está disposta a fornecer de crédito ao cliente.

3. Indicações bibliográficas

- ALMEIDA, V. N. *O que é processo de negócio: entenda a classificação de processos em uma organização.* Disponível em <https://www.euax.com.br/2018/08/processo-de-negocio/>. Acesso em 11 jul. 2025.
- CEOLIN, G. *Tarefa, atividade ou processo?* 16 dez. 2016. Disponível em <https://www.linkedin.com/pulse/tarefa-atividade-ou-processo-glauber-ceolin/>. Acesso em 11 nov. 2019.
- MARQUEZ, D. *Ferramentas de melhoria de processos: como aperfeiçoar os seus negócios.* Disponível em <https://nfe.io/blog/gestao-empresarial/ferramentas-de-melhoria-de-processos/>. Acesso em 11 jul. 2025.
- PARREIRAS, P. *Modelagem de processos de negócios: O que é, para que serve?* Disponível em <https://fluxoconsultoria.poli.ufrj.br/blog/gestao-empresarial/modelagem-de-processos-de-negocios//>. Acesso em 11 jul. 2025.
- SILVA, D. C. *Metodologia para implantação da gestão por processos no setor operacional de uma empresa detentora de usinas hidrelétricas.* 2010. Trabalho de conclusão de

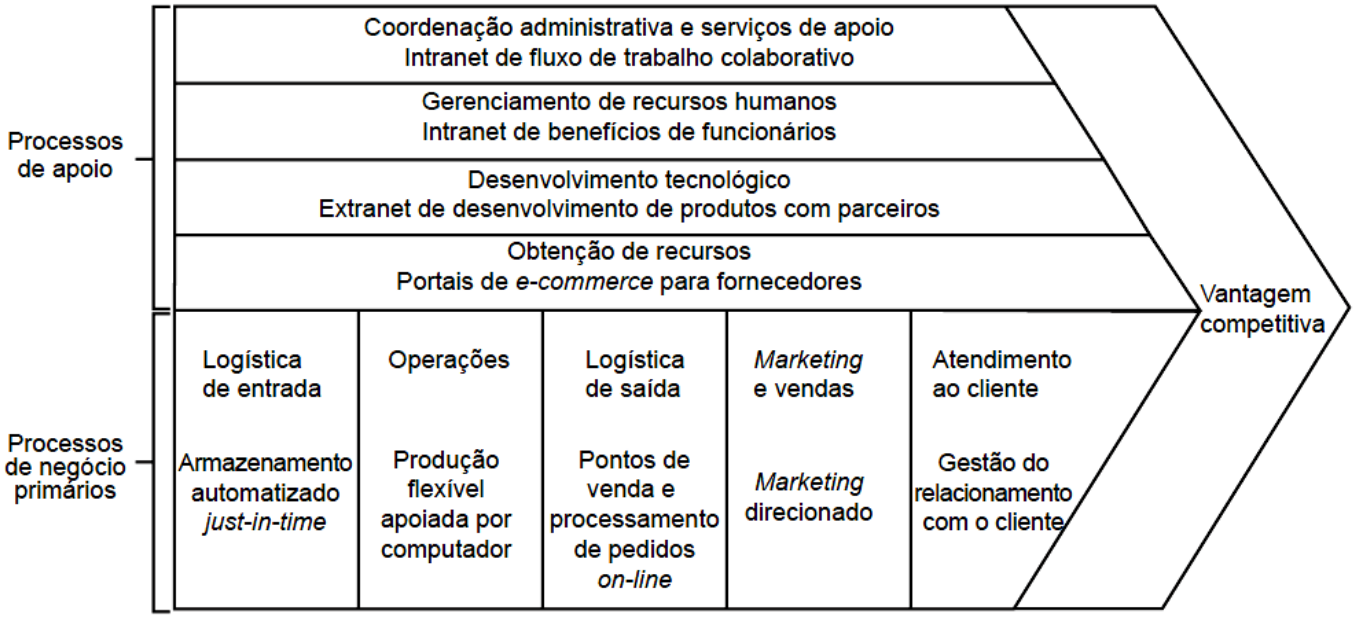
curso. Faculdade de Engenharia da Universidade Federal de Juiz de Fora. Juiz de Fora: 2010.

- SILVA, L. C. da. *Gestão e melhoria de processos: conceitos, técnicas e ferramentas*. Rio de Janeiro: Brasport, 2015.

Questão 9

Questão 9.[9]

Analise a figura a seguir.



O'BRIEN, J. A.; MARAKAS, G. M. *Administração de sistemas de informação.* 15. ed. Porto alegre: AMGH, 2012 (com adaptações).

A partir da análise da figura, avalie as afirmativas.

I. É apresentado um exemplo de como e onde as tecnologias da informação podem ser aplicadas nos processos de negócios por meio da estrutura da cadeia de valor.

II. É sugerido que um fluxo de trabalho colaborativo pela internet pode aumentar as comunicações e colaborações necessárias à significativa melhoria da coordenação administrativa e dos serviços de apoio.

III. É sugerido que os portais de *e-commerce* podem melhorar a economia de recursos ao oferecer mercados *online* para os fornecedores.

IV. É demonstrado que a vantagem competitiva é ampliada quando a área de Tecnologia da Informação (TI) está alinhada ao negócio.

É correto o que se afirma em

A. I, II e III, apenas.

B. I, II e IV, apenas.

C. I, III e IV, apenas.

D. II, III e IV, apenas.

E. I, II, III e IV.

[9]Questão 13 – Enade 2017.

1. Introdução teórica

Administração de sistemas de informação

A disputa cada vez mais acirrada por clientes tem provocado o aumento da automatização de tarefas nas empresas, e isso tem como aliada a tecnologia de informação (TI). A TI é fator essencial para ajudar a organização a operar com sucesso em ambientes competitivos.

A TI é caracterizada por um conjunto de recursos tecnológicos cujos objetivos são a captura, o processamento, o armazenamento, a utilização e a transmissão da informação. Isso permite que a integração do setor de TI com os demais setores de uma empresa ofereça inúmeros benefícios ao negócio, como aumento da produtividade, redução de custos, segurança na produção, garantia na entrega de produtos ou serviços dentro dos prazos, aumento da lucratividade, otimização no setor de atendimento, melhora nos resultados etc.

Segundo Westcon (2019), a TI tem se consolidado no ambiente de trabalho das empresas, deixando de ser apenas uma área de suporte técnico para se tornar parte fundamental nas estratégias dos negócios. Sua contribuição é grande diferencial na geração de valor para as empresas.

Para Correa (2019),

> *a gestão da tecnologia da informação é a administração dos recursos tecnológicos utilizados no processo de tratamento da informação de uma entidade, instituição ou organização. Esse processo envolve coleta, armazenamento, processamento, seleção, comparação, distribuição e avaliação de dados, que serão convertidos em informações úteis para a tomada de decisão.*

Os processos de negócio que caracterizam a atuação da empresa são cada vez mais amparados pelos sistemas de informação baseados em TI. A TI permite a integração entre organizações e a integração das partes de uma organização para que seja entregue ao cliente final um valor que é maior do que a soma das partes isoladas.

A TI pode ser utilizada de várias maneiras, e seus vários usos podem provocar benefícios que variam com a intensidade da aplicação e com o tipo de negócio.

A figura 1 apresenta exemplos desses benefícios e formas de mensurá-los.

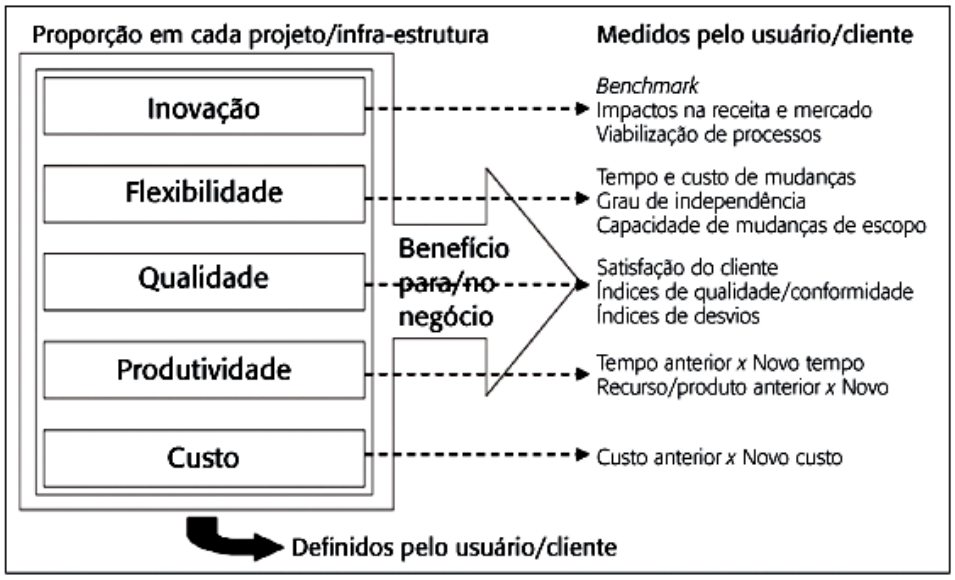


Figura 1. Benefícios oferecidos pelo uso da TI.
ALBERTIN e ALBERTIN, 2008.

Para que a TI ofereça um diferencial competitivo, ela deve comunicar-se de maneira constante com todos os setores do negócio, garantindo que a integração na empresa ocorra de forma fluida e focada em resultados.

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. A parte inferior da figura do enunciado mostra os setores nos quais a TI pode ser aplicada e criar vantagem competitiva para a empresa. Na figura 2, esses setores estão dentro de um quadro com bordas em vermelho.

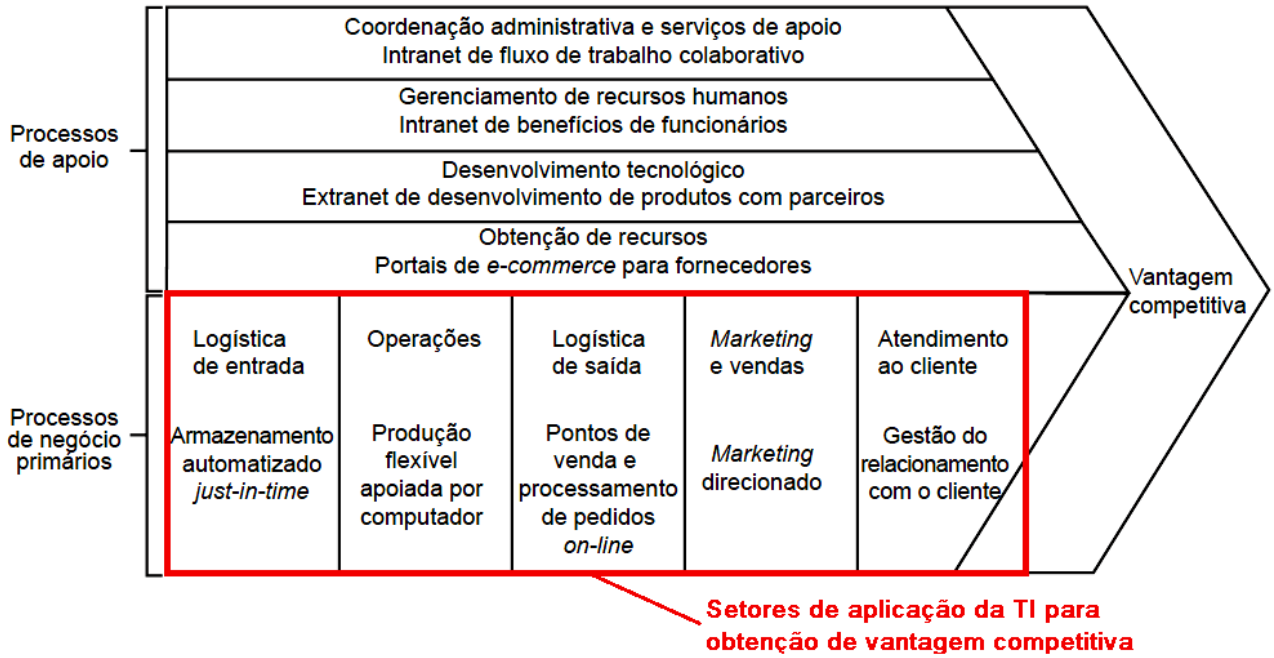


Figura 2. Setores onde pode ser aplicada a TI.

II – Afirmativa correta.

JUSTIFICATIVA. Na figura 3, dentro do quadro com bordas em azul, está destacado serviço de intranet para um fluxo de trabalho colaborativo.

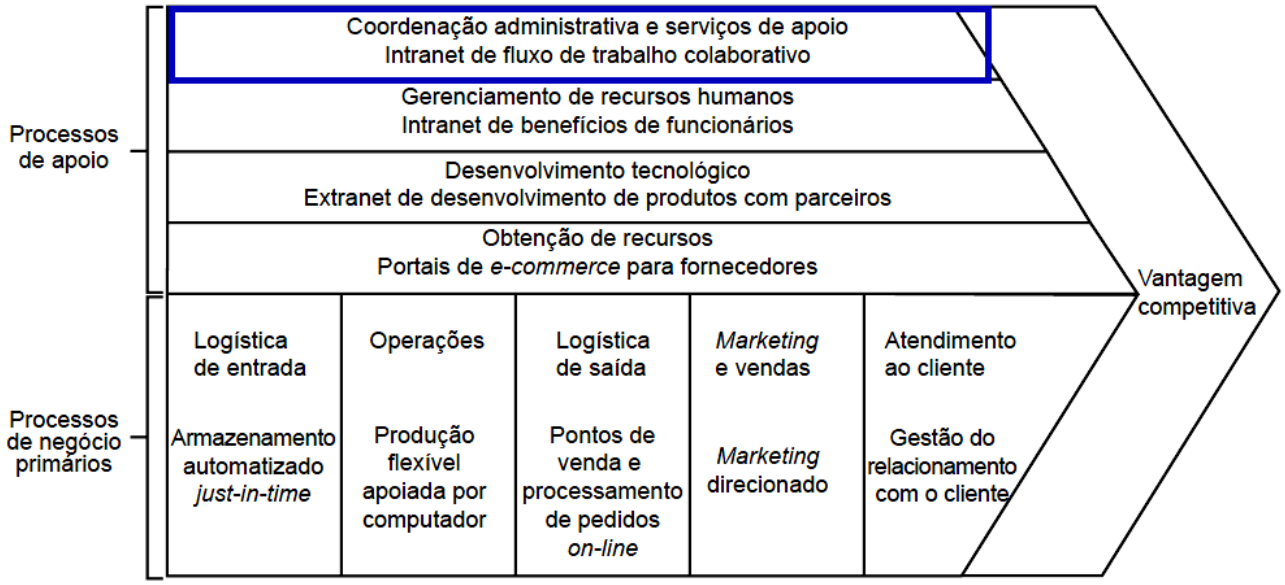


Figura 3. Uso da Intranet.

III – Afirmativa correta.

JUSTIFICATIVA. Na figura 4, dentro do quadro com bordas em laranja, está destacado o serviço de *e-commerce*.

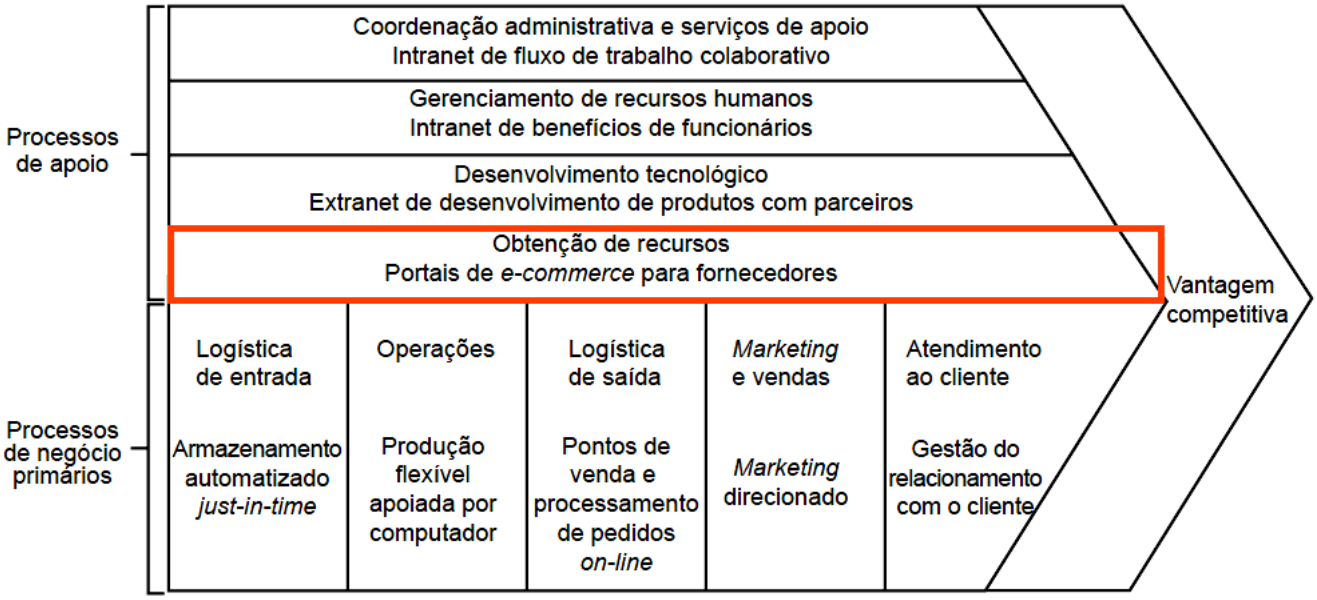


Figura 4. *E-commerce* destacado

IV – Afirmativa correta.

JUSTIFICATIVA. A figura do enunciado sugere que, quando há a integração dos processos de apoio com os processos de negócio por meio da TI, a empresa obtém vantagem competitiva. Isso está evidenciado na área sombreada em amarelo da figura 5.

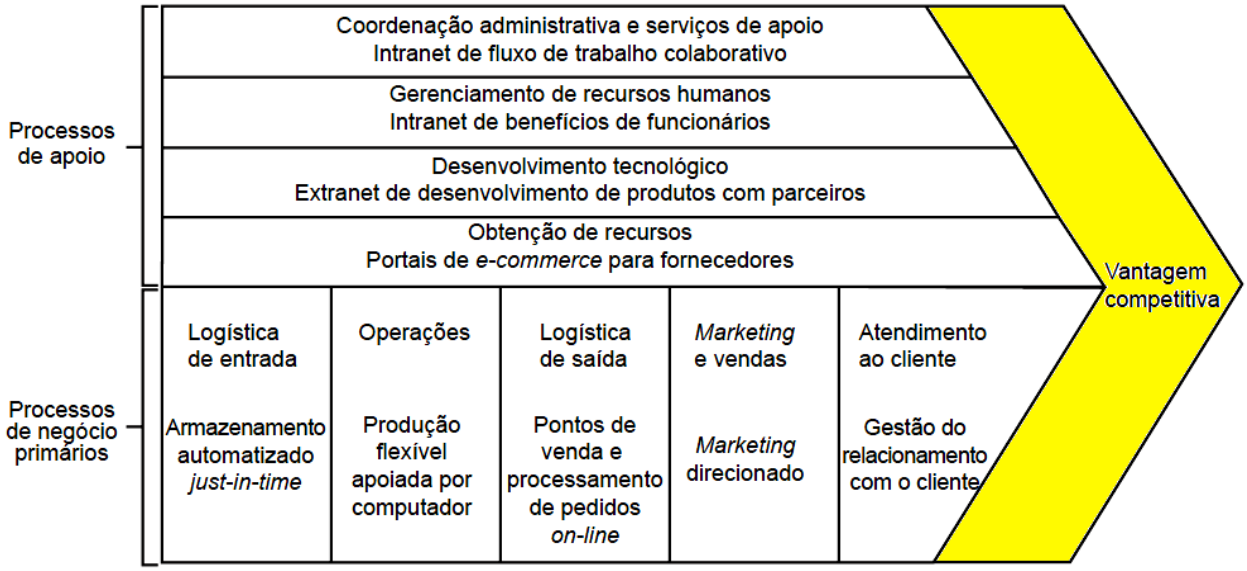


Figura 5. Vantagem competitiva obtida pela integração dos processos por meio da TI.

Alternativa correta: E.

3. Indicações bibliográficas

- ALBERTIN, A. L.; ALBERTIN, R. M. M. *Benefícios do uso de tecnologia de informação para o desempenho empresarial.* Rio de Janeiro. Revista de Administração Pública. v.42 n. 2, mar./abr. 2008.
- BALTZAN, P.; PHILLIPS, A. *Sistemas de informação*. Porto Alegre: AMGH, 2012.
- BALTZAN, P. *Tecnologia orientada para gestão.* Porto Alegre: AMGH, 2016.
- CORREA, R. M. *O que é Gestão da Tecnologia da Informação? Entenda como a TI pode ser uma aliada estratégica do negócio.* Disponível em <https://www.euax.com.br/2018/08/gestao-da-tecnologia-da-informacao/>. Acesso em 12 nov. 2019.
- SCHIO, A. *A importância da TI para o sucesso da empresa.* Disponível em <http://blog.loupen.com.br/a-importancia-da-ti-para-o-sucesso-da-empresa/>. Acesso em 11 nov. 2019.
- WESTCON. *Como a governança de TI pode ajudar nos negócios?* Disponível em <https://blogbrasil.westcon.com/como-a-governanca-de-ti-pode-ajudar-nos-negocios>. Acesso em 11 nov. 2019.

Questão 10

Questão 10.[10]

Em uma pesquisa de satisfação com notas de 0 a 10, sendo 0 muito insatisfeito e 10 muito satisfeito, uma empresa construiu um quadro, mostrado a seguir, com o resumo das notas atribuídas pelos seus clientes aos serviços recebidos.

NOTA	QUANTIDADE DE CLIENTES
0	0
1	0
2	0
3	1
4	2
5	2
6	2
7	2
8	2
9	1
10	0

Considerando essa situação e as informações apresentadas, avalie as afirmativas.

I. A média das notas dos clientes é igual a 6,0.

II. A mediana das notas dos clientes é igual a 6,0.

III. A variância populacional é menor do que 3,0.

IV. O conjunto de dados é amodal.

V. Um cliente que atribuiu nota 3,0 encontra-se no 1° quartil.

É correto apenas o que se afirma em

A. I, II e III.

B. I, II e V.

C. I, III e IV.

D. II, IV e V.

E. III, IV e V.

[10]Questão 24 – Enade 2017 (com adaptações).

1. Introdução teórica

1. Medidas de posição e medidas de dispersão de um conjunto de dados

1.1. Medidas de posição de um conjunto de dados

As medidas de posição fornecem estatísticas que podem caracterizar o comportamento dos elementos de um conjunto de dados. Em outras palavras, essas medidas descrevem a característica de tendência central dos valores numéricos de um conjunto de observações.

As principais medidas de tendência central são:

- média;
- mediana;
- moda.

Considere uma série de N medidas de certa grandeza, cada uma delas representada por x_i. A média $\overline{x}$ do conjunto de medidas é definida como a soma de todas as medidas dividida pelo número de medidas. Ou seja:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

A mediana é o elemento que ocupa a posição central de uma série de dados. Para determiná-la, os dados devem estar dispostos em ordem crescente. Temos, então, 50% dos dados com valores inferiores à mediana e 50% dos dados com valores superiores à mediana. A mediana também é conhecida como o segundo quartil, indicado como Q_2.

Temos outros quartis. O primeiro quartil (ou Q_1) é aquele superior ou igual a 25% dos dados de uma distribuição, o terceiro quartil (ou Q_3) é aquele superior ou igual a 75% dos dados de uma distribuição, e o quarto quartil (Q_4) é aquele que limita superiormente 100% dos dados.

A moda é o valor que ocorre com maior frequência em uma série de dados. Quando a distribuição tem apenas uma moda, é classificada como unimodal; quando tem duas modas, é classificada como bimodal; quando tem 3 ou mais modas, é classificada como multimodal; e quando não apresenta moda, é classificada como amodal.

1.2. Medidas de dispersão de um conjunto de dados

As medidas de dispersão informam a respeito do grau de variação ou de dispersão dos valores observados.

As principais medidas de dispersão são:

- amplitude total;
- variância;
- desvio padrão.

A amplitude total é a diferença entre o maior e o menor valor das observações de um conjunto de dados.

A variância de um conjunto de dados é a soma dos quadrados dos erros (isto é, das diferenças entre cada valor e a média) dividida pelo número de observações de um conjunto de dados. A variância σ^2 de um conjunto de dados com N medidas, cada uma delas representada por x_i, é dada por:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$$

O desvio padrão é uma medida que fornece a dispersão dos dados observados em torno do valor médio. O desvio padrão σ de um conjunto de dados com N medidas, cada uma delas representada por x_i, é dado por:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

Vale notar que podemos calcular a dispersão relativa como a razão entre o desvio padrão e a média do conjunto de dados em análise.

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. A média das notas é obtida somando-se o produto de cada nota por sua frequência f (quantidade de clientes) e dividindo-se esse resultado pelo total de notas dadas N. Logo:

$$\overline{nota} = \frac{1}{N}\sum_{i=1}^{N} nota_i.f$$

$$\overline{nota} = \frac{1}{12}.\left(0.0 + 1.0 + 2.0 + 3.1 + 4.2 + 5.2 + 6.2 + 7.2 + 8.2 + 9.1 + 10.0\right) = \frac{72}{12} = 6,0$$

A nota média é, portanto, igual a 6,0.

II – Afirmativa correta.

JUSTIFICATIVA. A mediana é o valor que "divide os dados ordenados de uma distribuição ao meio". Então, devemos ter 50% dos valores acima da mediana e 50% dos valores abaixo dela. Para os dados do enunciado, vemos que a mediana é 6,0.

III – Afirmativa incorreta.

JUSTIFICATIVA. O desvio padrão populacional σ para uma série de N dados x_i, de média $\bar{x}$, é dado por:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Quando os valores estão divididos em faixas de frequência, no caso de N valores, temos:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^2.f_i}$$

Para os dados apresentados no enunciado, ficamos com:

$$\sigma = \sqrt{\frac{1}{12}.\left[(3-6)^2 + 2.(4-6)^2 + 2.(5-6)^2 + 2.(6-6)^2 + 2.(7-6)^2 + 2.(8-6)^2 + (9-6)^2\right]}$$

$$\sigma = \sqrt{\frac{1}{12}.\left[(-3)^2 + 2.(-2)^2 + 2.(-1)^2 + 2.(0)^2 + 2.(1)^2 + 2.(2)^2 + (3)^2\right]} = \sqrt{\frac{38}{12}} = \sqrt{\frac{19}{6}}$$

Mesmo sem efetuarmos o cálculo, podemos estimar um limite superior para esse valor:

$$\sigma = \sqrt{\frac{19}{6}} < \sqrt{\frac{24}{6}} \Leftrightarrow \sigma < \sqrt{4} \Leftrightarrow \sigma < 2$$

Logo, o desvio padrão populacional dos dados é inferior a 2.

Sobre a variância da população, que é o quadrado do desvio padrão, temos:

$$\sigma^2 = \frac{19}{6} > \frac{18}{6} \Leftrightarrow \sigma^2 > 3$$

Logo, a variância populacional dos dados é superior a 3.

IV – Afirmativa incorreta.
JUSTIFICATIVA. Uma distribuição amodal é aquela que não tem moda. A moda é o valor mais frequente de uma distribuição. Na distribuição do enunciado, vemos que as notas mais frequentes são 4, 5, 6, 7 e 8, todas com frequência 2.

V – Afirmativa correta.
JUSTIFICATIVA. Os quartis dividem os dados em 4 partes iguais. Logo, o primeiro quartil é aquele em que 25% dos dados são menores ou iguais a ele. Sobre os dados do enunciado, temos 12 clientes que atribuíram notas. Portanto, as três primeiras notas (12/4) estão abaixo do primeiro quartil ou são iguais ao primeiro quartil. Logo, o primeiro quartil é dado pela nota 4 (temos três clientes que atribuíram nota menor ou igual a esse valor), e a nota 3 encontra-se no primeiro quartil.

Alternativa correta: B.

3. Indicações bibliográficas

- CRESPO, A. A. *Estatística fácil*. 19. ed. São Paulo: Saraiva, 2009.
- MAGALHÃES, M. N.; LIMA, A. C. P. *Noções de probabilidade e estatística*. 7. ed. São Paulo: Edusp, 2009.
- MORETTIN, P. A.; BUSSAB, W. O. *Estatística básica*. 6. ed. São Paulo: Saraiva, 2010.

ÍNDICE REMISSIVO

Questão 1	Sistemas operacionais. Arquitetura de sistemas operacionais. Escolha de sistemas operacionais.
Questão 2	Estrutura de dados. Algoritmos. Matrizes. Listas ligadas.
Questão 3	Estrutura de dados. Algoritmos. Deque. Fila. Pilha. Fila invertida. Lista circular.
Questão 4	Banco de dados. Diagrama de Entidade-Relacionamento (DER). Modelo conceitual. Modelo Lógico.
Questão 5	Banco de Dados. Consultas. SQL. Comando SELECT.
Questão 6	Gerência de Projetos. Guia PMBOK®. Análise de riscos.
Questão 7	Gerência de Projetos. Guia PMBOK®. Gerenciamento de escopo. Gerenciamento de tempo.
Questão 8	Gestão de processos de negócios. Gestão organizacional. Processos de negócios.
Questão 9	Gestão de processos de negócios. Importância da tecnologia da informação nos processos das empresas.
Questão 10	Estatística. Medidas de posição central. Medidas de dispersão.