



ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 11

CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 11

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 e Mestre em Engenharia Elétrica e Engenheiro Elétrico - ênfase em Telecomunicações. Professor titular da Universidade Paulista.

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

Questões 1 e 2

Questão 1.¹

Leia o texto a seguir.

Os requisitos de um sistema consistem nas descrições daquilo que o sistema deve fazer, dos serviços que o sistema oferece e das restrições ao seu funcionamento. Esses requisitos refletem as necessidades específicas dos clientes do sistema, como controlar um dispositivo, realizar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é denominado engenharia de requisitos.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011 (com adaptações).

Tendo o texto como referência, avalie as atribuições listadas abaixo, no que se refere à etapa de especificação de requisitos em projetos de software.

- I. Identificar as expectativas e necessidades dos stakeholders com relação ao software a ser desenvolvido.
- II. Distribuir os requisitos em categorias, explorar as relações entre eles e classificar sua importância para os stakeholders.
- III. Produzir um documento de especificação de requisitos, de forma que todos os stakeholders possam entendê-lo.
- IV. Examinar a especificação do software para assegurar que todos os requisitos foram definidos sem inconsistências.

São atribuições na etapa de especificação de requisitos os itens

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

Questão 2.²

Leia o texto a seguir.

À medida que o projeto de uma interface do usuário evolui, quatro questões de projeto comuns quase sempre vêm à tona: tempo de resposta do sistema, recursos de ajuda do usuário, informações de tratamento de erros e atribuição de nomes a comandos.

PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 8. ed. São Paulo: AMGH, 2016 (com adaptações).

¹Questão 28 – Enade 2017.

²Questão 31 – Enade 2017.

Nesse contexto, avalie as afirmativas abaixo, a respeito do projeto de interface e dos requisitos de software.

- I. A análise de interfaces tem por objetivo identificar o perfil dos usuários e auxiliar na especificação do requisito não funcional de segurança pertinente ao perfil de acesso dos usuários.
- II. O requisito não funcional de segurança descreve as características relacionadas a inteligibilidade, modo de usar ou operar o sistema.
- III. O requisito não funcional de confiabilidade descreve as características relacionadas ao tempo de resposta para o processamento de dados ou informações.
- IV. O requisito não funcional de usabilidade descreve diretrizes de acessibilidade para pessoas com deficiência.

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.

1. Análise da questão

Engenharia de requisitos

A análise de requisitos é uma etapa fundamental da engenharia de software. Segundo Pressman (2016), “a análise de requisitos funciona como uma ponte entre a engenharia de sistemas e o projeto do software”.

O processo inicial de busca e identificação dos requisitos começa na etapa de elicitação de requisitos, na qual os usuários e os analistas de sistemas devem se reunir para estabelecer:

- as necessidades básicas do cliente;
- as suas expectativas em relação ao sistema a ser desenvolvido.

A engenharia de requisitos tem, dessa forma, pelo menos dois “públicos-alvo” no que se refere aos seus artefatos:

- os técnicos que irão desenvolver o sistema;

- os usuários e os interessados no sistema, que podem não ser conhecimentos técnicos avançados (ao menos em relação ao desenvolvimento de software).

Um dos desafios dos profissionais que trabalham na engenharia de requisitos é serem capazes de se comunicar de forma clara e precisa com ambos os grupos mencionados anteriormente.

Na literatura da área de engenharia de software e, em especial, literatura da engenharia de requisitos, esse desafio é claramente identificado. Por exemplo, no livro “Engenharia de Software”, Sommerville (2007) diferencia os “requisitos de usuário” dos “requisitos de sistema”. Essa diferenciação está baseada no fato de que as descrições dos requisitos são feitas para grupos diferentes de pessoas, com finalidades diferentes e com níveis diferentes de profundidade técnica.

Para Sommerville (2007):

- os “requisitos de usuário” devem ser descritos sem muitos detalhes técnicos, uma vez que seu público-alvo não é composto apenas por desenvolvedores de software, mas por várias pessoas envolvidas ou interessadas no projeto;
- os “requisitos de sistema” devem ser descritos com muitos detalhes técnicos, uma vez que tais requisitos orientam a equipe de desenvolvimento e, em alguns casos, formam um documento que pode ter caráter contratual.

Ainda podemos classificar os requisitos de sistema de acordo com os seus tipos. Sommerville (2007) identifica três categorias:

- requisitos funcionais, que identificam as funcionalidades que o sistema deve oferecer e como deve ser o comportamento do sistema;
- requisitos não funcionais, que, normalmente, são restrições sobre as funcionalidades do sistema (por exemplo, restrições de performance);
- requisitos de domínio, que são originados diretamente do contexto no qual a aplicação vai ser utilizada, e não diretamente dos usuários (por exemplo, uma aplicação que será desenvolvida para uma empresa do setor financeiro pode ter requisitos que vêm diretamente do seu domínio, como cálculos específicos de tarifas).

É importante termos em mente que essas categorias não são mutuamente exclusivas, uma vez que requisitos de domínio podem ser requisitos funcionais, requisitos não funcionais ou restrições a outros requisitos, como, por exemplo, restrições em requisitos funcionais.

Ambas as questões em estudo abordam esses temas, mas sob perspectivas um pouco diferentes:

- a primeira questão descreve, basicamente, as diversas atividades que devem ser feitas, de forma geral, em qualquer projeto de software na especificação dos requisitos do projeto;
- a segunda questão é um pouco mais específica do que a primeira e explora a diferenciação entre requisitos funcionais e não funcionais no que tange o projeto de interfaces.

2. Análise das afirmativas

Questão 1

I – Afirmativa correta.

JUSTIFICATIVA. É fundamental que, no processo de elicitação de requisitos, as necessidades dos *stakeholders* sejam identificadas para que o sistema entregue não provoque decepções em pessoas fundamentais para o projeto.

II – Afirmativa correta.

JUSTIFICATIVA. A classificação e o entendimento dos requisitos são fundamentais para a execução do projeto. A identificação da importância desses requisitos para os diversos envolvidos no projeto faz com que se dê a devida prioridade ao desenvolvimento do sistema.

III – Afirmativa correta.

JUSTIFICATIVA. O aspecto abordado na afirmativa está intimamente relacionado ao que Sommerville (2007) chama de requisitos de usuário, ou seja, à especificação dos requisitos em um grau de abstração que permita que outras pessoas, além dos técnicos diretamente envolvidos na implementação, entendam o que deve ser feito.

IV – Afirmativa correta.

JUSTIFICATIVA. Inconsistências nos requisitos podem levar a vários problemas, erros e atrasos em projetos. Logo, é fundamental que se busque identificar e eliminar essas inconsistências o mais rapidamente possível.

Alternativa correta: E.

Questão 2

I – Afirmativa correta.

JUSTIFICATIVA. Requisitos não funcionais, como os requisitos de segurança, podem ser extremamente importantes, ou até mesmo fundamentais, em muitos sistemas. Em diversas empresas (como bancos), é fundamental que esses requisitos sejam claramente identificados.

II – Afirmativa incorreta.

JUSTIFICATIVA. A inteligibilidade está relacionada ao requisito não funcional de usabilidade e não ao requisito de segurança.

III – Afirmativa incorreta.

JUSTIFICATIVA. O tempo de resposta está relacionado ao requisito não funcional de desempenho, e não ao requisito de confiabilidade.

IV – Afirmativa correta.

JUSTIFICATIVA. A usabilidade envolve diversos aspectos; entre eles, a questão da acessibilidade para diversos grupos de usuários.

Alternativa correta: B.

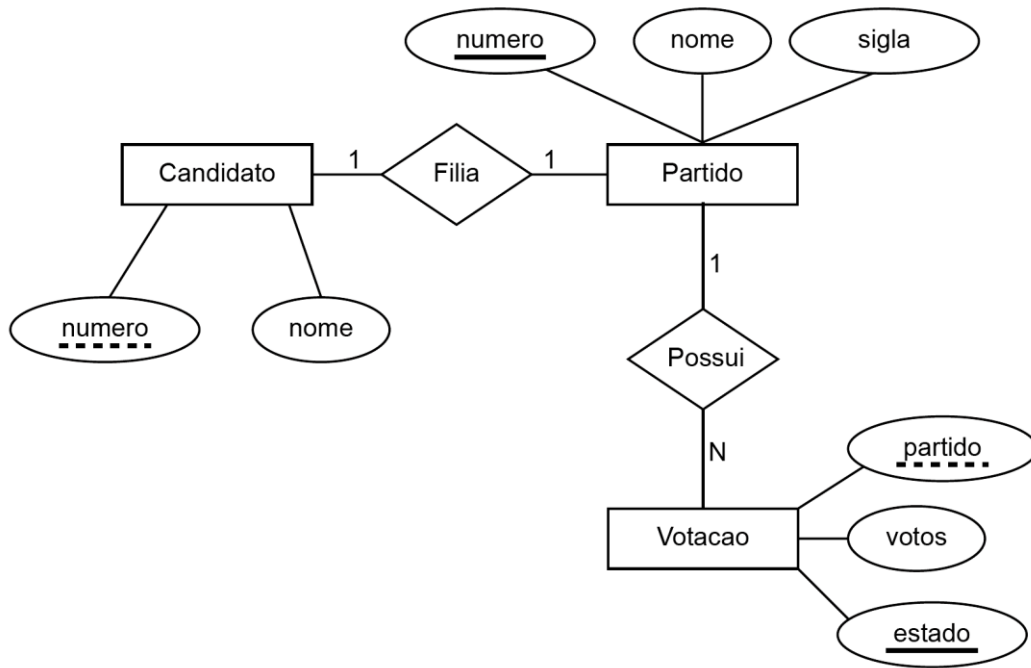
3. Indicações bibliográficas

- MACHADO, F. N. R. *Análise e gestão de requisitos de software: onde nascem os sistemas*. 3. ed. São Paulo: Érica, 2016.
- PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 8. ed. São Paulo: AMGH, 2016.
- SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.
- WIEGERS, K.; BEATTY, J. *Software requirements*. 3. ed. Redmond: Microsoft Press, 2013.

Questões 3 e 4

Questão 3.³

Um cliente solicitou a uma empresa a criação de um banco de dados para armazenar o resultado de uma eleição presidencial, com dados sobre os partidos políticos, os candidatos e a votação obtida por cada candidato em cada unidade da federação. O seguinte diagrama de Entidade-Relacionamento foi elaborado como representação dos requisitos obtidos com o cliente.



As tabelas a seguir contêm os dados registrados a partir do resultado dessa eleição.

Partido		
Número	Nome	Sigla
91	Partido 1	P1
92	Partido 2	P2
93	Partido 3	P3
94	Partido 4	P4
95	Partido 5	P5

Candidato	
Número	Nome
91	Candidato 1
92	Candidato 2
93	Candidato 3
94	Candidato 4
95	Candidato 5

³Questão 32 – Enade 2017.

Votação		
Partido	Votos	Estado
91	12345	AC
91	98323	PE
91	1726453	SP
92	98463	AC
92	192837	PE
92	4283747	SP
93	16253	AC
93	293845	PE
93	6253745	SP
94	98372	AC
94	598333	PE
94	8271347	SP
95	46837	AC
95	327264	PE
95	5938374	SP

Com base nas informações e na situação apresentada, qual o comando SQL que seleciona corretamente os nomes dos candidatos, seus partidos e o total de votos de cada partido nessa eleição?

- A. SELECT c.nome, p.nome, SUM(v.votos) FROM Partido p, Candidato c, Votacao v WHERE c.numero = p.numero and v.partido = c.numero;
- B. SELECT c.nome, p.nome, COUNT(v.votos) FROM Partido p, Candidato c, Votacao v WHERE c.numero = p.numero and v.partido = c.numero GROUP BY c.nome, p.nome;
- C. SELECT c.nome, p.nome, SUM(v.votos) FROM Partido p, Candidato c, Votacao v WHERE c.numero = p.numero and v.partido = c.numero GROUP BY c.nome, p.nome;
- D. SELECT c.nome, p.nome, v.votos FROM Partido p, Candidato c, Votacao v WHERE c.numero = p.numero and v.partido = c.numero GROUP BY c.nome, p.nome, SUM(v.votos);
- E. SELECT c.nome, p.nome, COUNT(v.votos) FROM Partido p, Candidato c, Votacao v WHERE c.numero = p.numero and v.partido = c.numero GROUP BY c.nome, p.nome, v.votos;

Questão 4.⁴

Leia o texto a seguir.

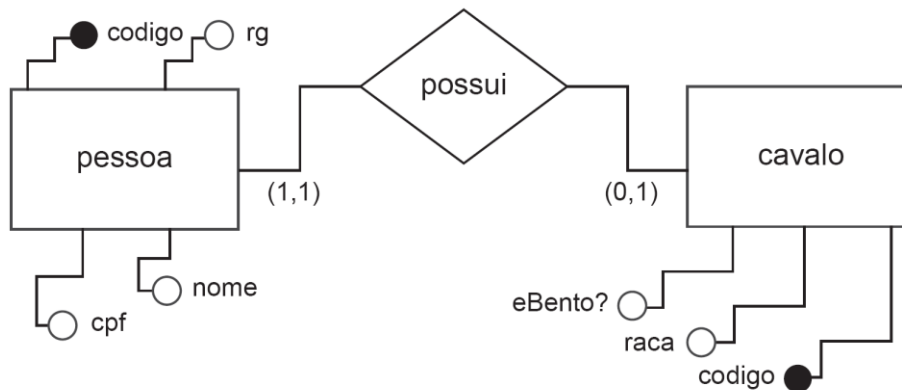
JOÃO GRILO: - Isso é coisa de seca. Acaba nisso, essa fome: ninguém pode ter menino e haja cavalo no mundo. A comida é mais barata e é coisa que se pode vender. Mas seu cavalo, como foi?

CHICÓ: - Foi uma velha que me vendeu barato, porque ia se mudar, mas recomendou todo cuidado, porque o cavalo era bento. E só podia ser mesmo, porque cavalo bom como aquele eu nunca tinha visto.

SUASSUNA, A. *Auto da compadecida*. Rio de Janeiro: Agir, 2000 (com adaptações).

⁴Questão 34 – Enade 2017.

A seguir apresenta-se um modelo de dados elaborado a partir do diálogo entre Chicó e João Grilo.



Com base no diálogo e no diagrama apresentados, avalie as afirmativas.

- I. O Chicó e a velha poderão ser cadastrados na entidade pessoa.
- II. O Chicó e a velha poderão ter mais que um cavalo cadastrado.
- III. O atributo rg da entidade pessoa pode ter a função de chave primária nessa entidade.
- IV. O cavalo deverá ter no mínimo uma pessoa e uma pessoa poderá ser cadastrada sem a necessidade de ter um cavalo.

É correto apenas o que se afirma em

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

1. Análise da questão

Modelagem de dados

As duas questões abordam modelagem de dados, particularmente o modelo de entidade-relacionamento. A primeira questão apresenta não apenas um diagrama de entidade-relacionamento, mas vai um pouco além disso, com a apresentação de tabelas. Adicionalmente, é proposta uma pergunta voltada à escrita de uma consulta feita na linguagem SQL.

É importante termos em mente que o modelo de entidade-relacionamento e o modelo relacional são diferentes, feitos em momentos diferentes e com objetivos diferentes.

Na abordagem tradicional de modelagem de dados, existem três etapas de modelagem:

- modelo conceitual;
- modelo lógico;
- modelo físico.

O modelo conceitual é construído com base nos requisitos do usuário e, frequentemente, utiliza o diagrama de entidade-relacionamento. Assim, nessa etapa, o projetista deve buscar identificar:

- as entidades relevantes para o projeto (e que, provavelmente, também serão relevantes para o banco de dados);
- os relacionamentos existentes entre as entidades.

O modelo de entidade-relacionamento ainda é bastante abstrato e não está diretamente ligado às tecnologias que serão utilizadas na implementação do projeto. Por esse motivo, deve ser feita uma “ponte” entre o modelo de entidade-relacionamento e o projeto do banco de dados. O modelo lógico faz esse papel. Normalmente, utilizamos o modelo relacional para a construção do modelo lógico. Essa construção é baseada em um mapeamento do modelo de entidade-relacionamento e do modelo relacional.

Finalmente, é necessário estabelecermos o modelo físico do banco de dados, que foca nos detalhes técnicos da implementação do modelo em um banco de dados específico. Várias questões devem ser observadas, como, por exemplo, índices e estruturas de armazenamento. Assim, de modo análogo ao caso anterior, o modelo físico deve ser obtido a partir do modelo lógico pelo processo de mapeamento. Esse processo depende tanto das técnicas utilizadas no projeto quanto das especificidades do sistema de gerenciamento de banco de dados escolhido.

Nesse sentido, como dissemos no início desta abordagem:

- a primeira questão fornece, além do diagrama de entidade-relacionamento, as tabelas do banco e o seu conteúdo, com o objetivo de que o aluno identifique a consulta que deve ser feita para retornar um conjunto de informações específicas;
- a segunda questão apresenta uma proposta mais conceitual, que foca nas características do diagrama de entidade-relacionamento.

Análise das afirmativas e das alternativas

Questão 3

A – Alternativa incorreta.

JUSTIFICATIVA. A consulta não mostra a cláusula GROUP BY, que é necessária para apresentar os dados agrupados por candidato.

B e E – Alternativas incorretas.

JUSTIFICATIVA. As consultas apresentadas nas alternativas utilizam a função COUNT, mas, na realidade, deveriam utilizar a função SUM, pois o objetivo é somar o total de votos, e não contar o número de registros.

C – Alternativa correta.

JUSTIFICATIVA. A consulta apresenta os resultados solicitados, utiliza a cláusula GROUP BY para agrupar os dados por candidato e emprega a função SUM para calcular o total de votos.

D – Alternativa incorreta.

JUSTIFICATIVA. A cláusula SUM está utilizado de maneira incorreta.

Questão 4

I – Afirmativa correta.

JUSTIFICATIVA. Supondo que seja possível associar um código tanto para Chicó quanto para a “velha”, ambos poderiam ser cadastrados na entidade pessoa.

II – Afirmativa incorreta.

JUSTIFICATIVA. De acordo com o diagrama apresentado, uma pessoa pode ter no máximo um cavalo.

III – Afirmativa incorreta.

JUSTIFICATIVA. De acordo com o diagrama apresentado, a chave primária deve ser o atributo “codigo”, e não o atributo “rg”.

IV – Afirmativa correta.

JUSTIFICATIVA. De acordo com o diagrama apresentado, uma pessoa pode não ter nenhum cavalo cadastrado ou apenas um cavalo cadastrado. Todo cavalo deve ter apenas uma pessoa associada.

Alternativa correta: B.

3. Indicações bibliográficas

- DATE, C. J. *Introdução a sistemas de banco de dados*. 8. ed. Rio de Janeiro: Elsevier, 2003.
- DATE, C. J. *SQL e teoria relacional: como escrever códigos SQL precisos*. São Paulo: Novatec, 2015.
- DOMANSKI, P.; IRVINE, P. *A practical guide to relational database design*. Herefordshire: Diaxon, 2000.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAM, S. *Database System Concepts*. 7. ed. New York: McGrawHill, 2020.
- SIMSION, G.; WITT, G. *Data modeling essentials*. 3. ed. Oxford: Elsevier, 2004.

Questão 5

Questão 5.⁵

Leia o texto a seguir.

A ordenação por inserção funciona de modo semelhante à forma como algumas pessoas ordenam cartas de baralho. Inicia-se com a mão esquerda vazia e as cartas empilhadas na mesa. Remove-se da pilha uma carta de cada vez, inserindo-a na posição correta na mão esquerda. Para se identificar a posição correta de uma carta, deve-se compará-la com as cartas presentes na mão esquerda, no sentido da direita para a esquerda. Em todos os momentos, as cartas na mão esquerda estão ordenadas, tendo sido obtidas no topo da pilha da mesa.

CORMEN, T. H. et al. *Introduction to algorithms*. 3. ed. Cambridge: The MIT Press, 2009 (com adaptações).

Um programador implementou um algoritmo de ordenação semelhante à forma de ordenação de cartas descrita no texto. Ao realizar um teste com um vetor de nove posições (`vetor[1..9]`), verificou que o algoritmo não funcionava corretamente.

```
01 para i <- 2 até 9 faça
02     valor <- vetor[i]
03     j <- i - 1
04     enquanto ((j >= 1) e (valor < vetor[j])) faça
05         vetor[i] <- vetor[j]
06         j <- j - 1
07         se (j = 0) então
08             interrompa
09         fim se
10     fim enquanto
11     vetor[j + 1] <- valor
12 fim para
```

Com base nessas informações, assinale a opção em que se apresentam a linha e o respectivo comando a ser substituído, para que o algoritmo ordene corretamente um vetor de inteiros de forma crescente.

- A. Linha 01; para i <- 1 até 9 faça
- B. Linha 03; j <- i
- C. Linha 04; enquanto ((j >= 1) ou (valor < vetor[j])) faça
- D. Linha 05; vetor[j+1] <- vetor[j]
- E. Linha 11; vetor[j] <- valor

⁵Questão 33 – Enade 2017.

1. Análise da questão

Ordenação por inserção

A ordenação por inserção é um dos algoritmos mais simples de ordenação e, por consequência, é de fácil entendimento. O algoritmo faz a ordenação local, o que significa que a quantidade de memória utilizada e o vetor fornecido para ser ordenado são constantes.

O algoritmo trabalha dividindo o vetor de entrada em duas partes. A parte inicial “começa” pequena e vai sendo ordenada e expandida conforme avançamos no processo de ordenação. Assim, inicia-se o processo de ordenação em uma das extremidades (normalmente a extremidade esquerda), a partir da segunda posição do vetor de entrada. Essa posição de referência costuma ser chamada de chave. É importante observarmos que a chave vai mudando durante a execução do algoritmo, indo em direção ao fim do vetor.

Os elementos anteriores à chave vão sendo comparados com ela, do elemento imediatamente mais próximo até o primeiro elemento do vetor (ou seja, trabalha-se “de trás para frente”). Os elementos são, então, empurrados para a frente, caso sejam maiores do que a chave. Quando chegamos a um elemento inferior à chave (ou ao início do vetor), encontramos a posição correta da chave atual no vetor. Assim, devemos escolher uma nova chave, que é o elemento à direita da chave atual.

Dessa forma, o vetor vai sendo ordenado da esquerda para direita, de forma que a última chave é o último elemento do vetor. Ao terminarmos de identificar a posição correta dessa última chave no vetor, terminamos a execução do algoritmo e temos um vetor com elementos organizados do menor para o maior.

2. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. A linha 1 do programa não deve ser modificada, uma vez que a chave deve ser associada a partir da segunda posição.

B – Alternativa incorreta.

JUSTIFICATIVA. A linha 3 não deve ser modificada, pois as comparações devem ser feitas com o elemento imediatamente à esquerda do elemento que é a chave.

C – Alternativa incorreta.

JUSTIFICATIVA. A linha 4 não deve ser modificada, pois o laço mais interno deve ter duas condições simultâneas:

- a primeira é o limite da posição inicial do vetor (a posição que corresponde a $j=1$);
- a segunda é a comparação entre o valor da chave e o elemento j do vetor.

Ambas as condições devem ser avaliadas simultaneamente e, por isso, é necessário o uso de um “E” lógico.

D – Alternativa correta.

JUSTIFICATIVA. A linha 5 deve ser modificada, pois é nessa linha que um elemento do vetor deve ser “empurrado” para frente, ou seja, deve ser modificado para a sua posição relativa de ordenação em dada iteração do algoritmo, como ilustrado na alternativa.

E – Alternativa incorreta.

JUSTIFICATIVA. A linha 11 não deve ser mudada e corresponde ao posicionamento da chave após o laço de comparação mais interno do algoritmo.

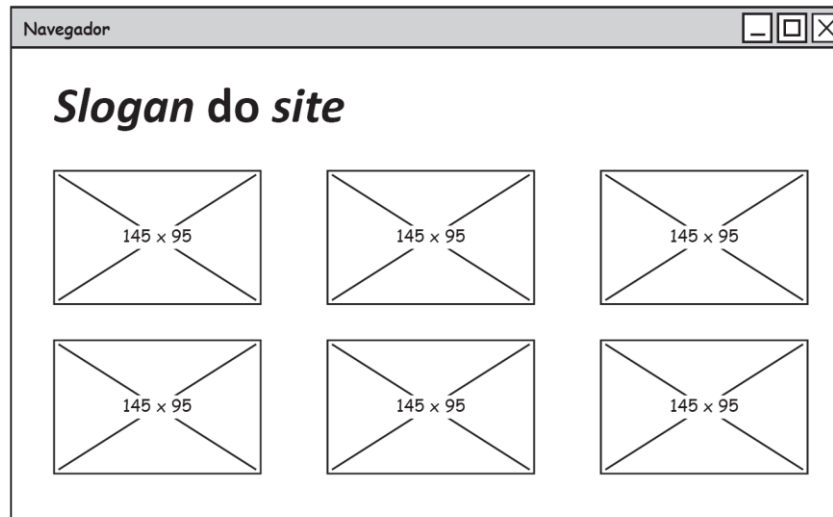
3. Indicações bibliográficas

- CORMEN, T. et al. *Algoritmos: teoria e prática*. 2. ed. Rio de Janeiro: Elsevier, 2002.
- DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U. *Algoritmos*. São Paulo: McGraw-Hill, 2006.
- MEDINA, M.; FERTIG, C. *Algoritmos e programação: teoria e prática*. São Paulo: Novatec, 2005.

Questão 6

Questão 6.⁶

Em um sistema web para venda de ingressos para shows musicais, a equipe de marketing solicitou a um web designer que, na página inicial, os links para a área de venda de ingressos de cada show fossem imagens, para tornar a navegação pelos conteúdos da página mais atrativa e para que o próprio link já servisse como material de divulgação.



A partir do protótipo da página inicial apresentado e usando HTML, o web designer optou por implementar cada link de forma que a imagem forneça um texto informativo (por meio do atributo alt) para usuários que usem leitores de tela, conforme a seguir.

```
<a href="link-area-venda-ingresso">

</a>
```

Considerando a situação apresentada, o uso do atributo alt e aspectos da interação humano-computador, pode-se afirmar que o web designer atendeu, em sua implementação, o requisito de

A. segurança no uso, pois se uma imagem tiver baixa resolução e impossibilitar a correta visualização, o texto informativo associado à imagem possibilita a identificação do conteúdo do respectivo link.

⁶Questão 35 – Enade 2017.

B. facilidade de aprendizado, pois a comunicação visual do sistema web, por meio do uso de imagens, aliada à comunicação textual, por meio do uso de textos informativos para cada imagem, ajuda o usuário a aprender os caminhos de navegação.

C. acessibilidade, pois a possibilidade de acesso à página por pessoas com deficiência visual, para que interajam com os conteúdos, oferece condições de igualdade às pessoas na interação com o sistema web.

D. ergonomia, pois o uso de imagens em links causa um cansaço visual quando o sistema web é aberto em um smartphone ou em um tablet com baixa resolução, e o atributo alt substitui a imagem por texto.

E. comunicabilidade, pois se uma imagem não comunica adequadamente a informação desejada, um texto alternativo deve complementar a respectiva informação, de forma que a interação com o sistema web não seja prejudicada.

1. Análise da questão

Acessibilidade e desenvolvimento web

A questão aborda o tema acessibilidade, relacionado ao desenvolvimento de sites. Quando desenvolvemos um sistema ou um site, queremos que o maior grupo de pessoas seja capaz de interagir com o resultado do nosso trabalho. Devemos ter em mente que, mesmo pessoas que possuam algum tipo de limitação ou manifestem dificuldades devem ser capazes de interagir com o nosso sistema.

Um exemplo de limitação que pode ocorrer com relação ao uso de um sistema é o caso de pessoas com deficiências visuais. Em alguns casos, essas deficiências podem ser apenas uma restrição na visão (como, por exemplo, pessoas que apresentam elevado grau de miopia); em outros casos, o indivíduo pode apresentar cegueira completa.

Muitos sistemas operacionais dispõem dos chamados leitores de tela, ferramentas que apresentam um sintetizador de voz e leem o conteúdo mostrado na tela para o usuário. Dessa forma, mesmo um usuário cego é capaz de interagir com o sistema por meio de uma descrição textual. É comum que esses usuários prefiram interagir com os programas usando atalhos de teclado ou escrevendo comando textuais.

Contudo, para que a leitura de tela seja possível, programas devem fornecer uma descrição textual de elementos visuais, como figuras, para que eles possam ser lidos pelo

sintetizador de voz. No caso de um site, quando adicionamos uma figura, é possível adicionar esse texto descritivo com o atributo *alt*. A figura original continua sendo mostrada normalmente na tela, mas, de maneira adicional, uma ferramenta de leitura de tela é capaz de ler essa descrição para um usuário com dificuldades visuais.

Essa funcionalidade também pode ser útil em outras situações, como, por exemplo, no caso de a figura não poder ser mostrada na tela em virtude de algum erro ou de problemas de conexão. Nesses casos, o texto do atributo *alt* é mostrado no lugar da figura. É importante ter em mente que isso não é o mesmo que a “dica” (*tooltip*) para uma imagem, a qual deve ser atribuída, separadamente, pelo atributo *title*.

2. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Se for utilizada uma imagem de baixa resolução, o texto fornecido não será mostrado (mas poderá ser lido por um leitor de tela). Assim, nesse caso, o comportamento não é o descrito na alternativa.

B – Alternativa incorreta.

JUSTIFICATIVA. No caso descrito, o uso do atributo *alt* não teria o efeito pretendido. Para isso, o site deveria fazer uso de outros elementos textuais.

C – Alternativa correta.

JUSTIFICATIVA. A descrição da figura realizada por meio do atributo *alt* faz com que pessoas portadoras de dificuldades visuais possam utilizar um sistema da mesma forma que um usuário com visão normal.

D – Alternativa incorreta.

JUSTIFICATIVA. O comportamento descrito na alternativa não seria obtido simplesmente com o uso do atributo *alt*.

E – Alternativa incorreta.

JUSTIFICATIVA. Além de o comportamento descrito na alternativa não corresponder ao uso do atributo *alt*, devemos questionar a utilização de imagens que não comunicam adequadamente sua mensagem.

3. Indicações bibliográficas

- BARBOSA, S.; SILVA, B. da. *Interação humano-computador*. Rio de Janeiro: Elsevier, 2010.
- BENYON, D. *Interação humano-computador*. 2. ed. São Paulo: Pearson Prentice Hall, 2011.
- CYBIS, W.; BETIOL, A. H.; FAUST, R. *Ergonomia e usabilidade: conhecimentos, métodos e aplicações*. São Paulo: Novatec, 2007.
- KALBACH, J. *Design de navegação web: otimizando a experiência do usuário*. Porto Alegre: Bookman, 2009.
- NORMAN, D. *The design of everyday things: revised and expanded edition*. New York: Basic Books, 2013.
- ROGERS, Y.; PREECE, J.; SHARP, H. *Design de interação: além da interação humano-computador*. 3. ed. Porto Alegre: Bookman, 2013.
- W3SCHOOLS. *HTML alt Attribute*. Disponível em <https://www.w3schools.com/tags/att_alt.asp>. Acesso em 11 jul. 2025.

Questão 7

Questão 7.⁷

Uma empresa de cosméticos comercializa cinco diferentes tipos de produtos e os armazena em uma estante de 40x40 posições. Em cada posição da estante pode ficar armazenada apenas uma caixa com um desses produtos. Para facilitar sua identificação, os produtos foram codificados da forma a seguir:

- 1: xampu;
- 2: condicionador;
- 3: hidratante;
- 4: tintura;
- 5: demaquilante;
- 0: vazio;

Nessa situação e considerando o desenvolvimento de um sistema para gerenciar a organização dos produtos na estante, estabeleceu-se a declaração de variáveis a seguir.

Var

Estante: matriz [1..40][1..40] de inteiro

Produtos: vetor [0..5] de texto = {"vazio", "xampu", "condicionador", "hidratante", "tintura", "demaquilante"}

Contador: vetor [0..5] de inteiro = {0,0,0,0,0,0}

i, j: inteiro

Com base nessa declaração e considerando a codificação dos produtos exposta, faça o que se pede nos itens a seguir, apresentando as soluções em pseudocódigo ou em linguagem de programação.

- a) Escreva um trecho de código para ler os códigos dos produtos e armazená-los na matriz Estante.
- b) Escreva um trecho de código para contar e imprimir a quantidade de caixas de cada tipo de produto na estante.

⁷Questão Discursiva 3 – Enade 2017.

1. Análise da questão

Vetores e matrizes

A questão aborda a manipulação de estruturas de dados estáticas, como matrizes e vetores.

O objetivo é que sejam escritos algoritmos simples para:

- a entrada de dados em uma matriz de armazenamento;
- a sumarização de informações a partir dos dados armazenados na matriz.

Como o enunciado já apresenta as estruturas de dados a serem utilizadas, alcançar o objetivo proposto requer, simplesmente, que sejam escritos os laços de manipulação das estruturas.

2. Resolução da questão

a) A solução sugerida pelo Inep (2017), feita em pseudocódigo, é mostrada a seguir:

```

01  /* Colocação dos produtos na estante */
02  Para i de 1 até 40 passo 1 Faça
03      Para j de 1 até 40 passo 1 Faça
04          imprima "Digite o código do produto da posição", i , " e ", j, ":"
05          leia Estante[i][j]
06          /* o Estudante pode ou não fazer a validação da entrada de dados
    aqui */
07          Se (Estante[i][j]<0 ou Estante[i][j]>5)
08              imprima "Codigo de produto invalido"
09              j <- j-1
10          Fim_Se
11      Fim Para
12  Fim Para
  
```

As linhas 2 e 3 correspondem a dois laços, e o objetivo é percorrer cada uma das posições disponíveis na matriz de 40 linhas por 40 colunas, que corresponde à matriz Estante. Para cada uma das posições, um texto é mostrado na tela (linha 4); nele, é solicitado que o usuário entre com o código do produto. Na linha 7, é feita uma validação do valor entrado pelo usuário (que deve estar entre 0 e 5). Se o código for inválido, o valor de j é decrementado de uma unidade (linha 9), ou seja, o usuário terá de entrar novamente com o código para o mesmo ponto da matriz (observe que os dois laços incrementam automaticamente os valores de i e j).

b) Para a segunda parte da questão, dois casos devem ser considerados:

- a solução apresentada no item a) inclui alguma forma de validação;
- a solução apresentada no item a) não inclui alguma forma de validação.

Além disso, a solução apresentada é dividida em duas partes:

- a parte que faz a contagem;
- a parte que mostra o resultado na tela.

Caso a solução apresentada no item anterior incluía a validação, a seguinte solução, em pseudocódigo, foi sugerida na resolução padrão do Inep (2017):

```

01   Para i de 1 até 40 passo 1 Faça
02       Para j de 1 até 40 passo 1 Faça
03           Contador[Estante[i][j]]=Contador[Estante[i][j]]+1
04       Fim Para
05   Fim Para

```

Como todos os códigos presentes na matriz Estante são válidos, basta iterarmos por todos os seus elementos e incrementarmos de uma unidade o total de produtos do vetor Contador, como mostrado na linha 3.

Se a solução proposta pelo aluno na parte a) não inclui a validação da entrada, é necessário verificarmos se os códigos apresentados na matriz Estante são válidos ou não e fazermos a contabilização apenas no caso em que os códigos sejam válidos, como indicado no código a seguir, sugerido na resolução padrão do Inep (2017).

```

01   Para i de 1 até 40 passo 1 Faça
02       Para j de 1 até 40 passo 1 Faça
03           Se (Estante[i][j]>=0 e Estante[i][j]<=5)
04               Contador[Estante[i][j]]=Contador[Estante[i][j]]+1
05           Fim_Se
06       Fim Para
07   Fim Para

```

Nessa solução, a linha 3 protege o programa para que não tentemos contabilizar algum código inválido, o que, além de resultar em valores errados, também implicaria uma invasão de memória no vetor Contador. Assim, são contabilizados apenas valores entre 0 e 5, válidos no vetor Contador. Observe que o código não protege contra qualquer tipo de erro: um usuário poderia cadastrar o código 1 em vez do código 2, o que estaria incorreto, mas não geraria um código inválido.

O vetor Contador é simplesmente um vetor de inteiros no qual, em cada uma de suas posições, é contabilizado o total relacionado aos seus produtos. Por exemplo, a posição 0 corresponde

às posições vazias na estante, a posição 1 corresponde ao total de xampu, a posição 2 corresponde ao total condicionadores etc. No código mostrado anteriormente, a contabilização é feita na linha 4.

Finalmente, uma vez que a quantidade de itens tenha sido contabilizada, é necessário mostrarmos os resultados para o usuário. No gabarito de resposta do Inep (2017), isso é feito por meio do código a seguir.

```
01      Para i de 1 até 5 passo 1 Faça
02          imprima "Produto ", Produtos[i], " possui ", Contador[i], " caixas na Estante"
03      Fim Para
```

O código é simplesmente um laço que imprime cada um dos nomes dos produtos, armazenados no vetor `Produtos`, e o seu total correspondente, armazenado no vetor `Contador`. Observe que o código não mostra o total de posições vazias na estante, pois o laço começa a partir da posição 1, e não da posição 0 dos vetores `Produtos` e `Contador`.

3. Indicações bibliográficas

- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. *Introdução às estruturas de dados*. Rio de Janeiro: Campus, 2004.
- INEP. *Padrão de Resposta da Prova de Tecnologia em Análise e Desenvolvimento de Sistemas de 2017*. Disponível em <<http://portal.inep.gov.br/educacao-superior/enade/provas-e-gabaritos>>. Acesso em 21 jan. 2025.
- SENNE, E. L. F. *Primeiro curso de programação em C*. 3. ed. Florianópolis: Visual Books, 2006.

Questão 8

Questão 8.⁸

Uma empresa trabalha na produção de concreto e terceiriza o serviço de transporte do produto. Os caminhoneiros telefonam para a empresa e registram seu interesse pelo trabalho. Todas as manhãs, os caminhoneiros estacionam o caminhão no pátio da empresa e aguardam sua vez. O atendimento segue o critério de ordem de chegada. Esse processo é, atualmente, controlado pela secretária, que utiliza sua agenda para gerenciar os motoristas diariamente. A empresa, que carrega no máximo 10 caminhões por dia, pretende informatizar esse processo. Para a solução do problema, apresenta-se, a seguir, um pseudocódigo que utiliza o conceito de fila, mantendo os elementos sempre nas primeiras posições do vetor.

```

Algoritmo Fila_Caminhoneiros
início
    var
    caminhoneiros : vetor[1..10] de texto
    total : inteiro
    procedimento inicializa()
    início
        total <- 0
    fim
    função estaVazia() : lógico
    início
        se (total = 0) então
            retorna verdadeiro
        senão
            retorna falso
        fim-se
    fim
    função estaCheia() : lógico
    início
        se (total >= 10) então
            retorna verdadeiro
        senão
            retorna falso
        fim-se
    fim
    procedimento enfileirar(caminhoneiro : texto)
    início
        se (estaCheia() = falso) então
            total <- total + 1
            caminhoneiros[total] <- caminhoneiro
        senão
            imprima("Fila cheia")
        fim-se
    fim
fim

```

⁸Questão Discursiva 4 – Enade 2017.

Com base nas informações apresentadas, faça o que se pede nos itens a seguir, expondo cada solução em pseudocódigo ou em linguagem de programação.

- a) Implemente a função `desenfileirar`, que deve remover e retornar um elemento representado por um caminhoneiro da fila ou a mensagem "Fila vazia" se não houver elementos.
- b) Implemente o procedimento `mostrarFila`, que deve apresentar a lista de elementos, ou seja, os caminhoneiros que estão na fila.

1. Análise da questão

Estrutura de dados: filas

A fila é uma estrutura de dados linear do tipo *FIFO – First In, First Out*. Isso significa que os primeiros dados inseridos na fila devem ser os primeiros a ser retirados. Esse processo é equivalente a uma fila de pessoas em um banco: a primeira pessoa que entrou na fila é a primeira que vai ser atendida (saindo da fila), seguida das demais pessoas, na ordem de chegada. Dessa forma, a estrutura de dados fila garante uma ordem de acesso aos elementos, além de servir como mecanismo de armazenamento das informações.

No enunciado é apresentado o pseudocódigo de um programa em que é construída uma fila utilizando-se um vetor como base da implementação. A implementação é feita usando uma estrutura de dados estática, de forma que a fila tem uma capacidade máxima de armazenamento (no caso do programa do enunciado, esse valor é de 10 posições). Esse é um vetor de texto, chamado "caminhoneiros", declarado no início do programa.

A implementação do enunciado apresenta algumas das funções necessárias para se trabalhar com a estrutura de dados fila. Uma variável inteira "total" registra o número de itens correntemente armazenado na estrutura de dados. O procedimento *inicializa*, que deve ser chamado no início do programa, atribui zero ao valor da variável total.

O enunciado também provê duas funções, `estaVazia()` e `estaCheia()`, para que seja possível identificar as situações limites de uso da estrutura de dados. A função `estaCheia()` é necessária pois a fila é internamente baseada em uma estrutura de dados estática.

A função `enfileirar()` também é fornecida, sendo responsável por verificar os limites da estrutura de dados e inserir novos elementos caso isso seja possível. O enunciado da questão solicita a implementação de duas funções: a função `desenfileirar()`, para retirar

elementos da fila, e a função `mostrarFila()`, para mostrar os elementos atualmente armazenados na fila.

2. Resolução da questão

a) O Inep (2017) indica a sugestão a seguir de implementação da função `desenfileirar()`.

```

01      Função desenfileirar() : texto
02          Var
03              retirado : texto
04              x : inteiro
05          Inicio
06              Se(total = 0) então
07                  retirado <- "Fila vazia"
08              Senão
09                  retirado <- caminhoneiros[1]
10                  total <- total - 1
11                  Para(x <- 1 até total passo 1) faça
12                      caminhoneiros[x] <- caminhoneiros[x+1]
13                  fim-para
14              Fim-se
15              retorna retirado
16      Fim

```

A função `desenfileirar()` não deve receber nenhum argumento e deve retornar o elemento da fila de acordo com o esquema FIFO. Observe que o elemento é um texto e que, caso a fila esteja vazia, o texto retornado pela função é "Fila vazia".

Na linha 3 da função, declara-se uma variável "retirado", do tipo texto. Essa variável contém o elemento a ser retornado pela função. Se a fila não estiver vazia, esse elemento corresponde ao primeiro elemento do vetor "caminhoneiros", como feito na linha 9. Se a fila estiver vazia (a checagem é feita na linha 6), associa-se o texto "Fila vazia" à variável "retirado" (linha 7). Na linha 10, diminui-se a variável que armazena o total de itens armazenados na fila, caso um item seja retirado.

As linhas 11, 12 e 13 correspondem a um laço no qual temos de reorganizar os elementos contidos no vetor "caminhoneiros". Devemos observar que esse vetor armazena os caminhoneiros na ordem em que eles chegaram. Assim, o primeiro elemento do vetor é o primeiro elemento a ser retirado. Contudo, ao retirarmos o primeiro elemento do vetor, não podemos manter os demais elementos nas mesmas posições: devemos mover todos os

elementos anteriores uma posição para a esquerda (ou para a direção de início do vetor). Isso é feito na linha 12.

Finalmente, após o término dos laços e das condições anteriores, a função retorna o valor contido na variável retirado, como indicado na linha 15.

b) O gabarito indica duas soluções possíveis:

a primeira solução apresenta o texto “Fila vazia” caso não exista nenhum elemento armazenado na fila (o que é controlado pela variável global total);

a segunda solução apenas mostra os itens armazenados no vetor caminhoneiros (de acordo com a variável total), potencialmente não imprimindo nenhuma informação, caso total seja nulo.

A primeira solução apresentada é a que segue (Inep, 2017):

```

01  Procedimento mostrarFila()
02      Var
03          x : inteiro
04      Início
05          Se(total = 0) então
06              imprima("Fila vazia")
07          Senão
08              Para(x <- 1 até total passo 1) faça
09                  imprima(caminhoneiros[x])
10              fim-para
11          fim-se
12      Fim

```

A linha 5 é utilizada para verificar se a fila está vazia ou não. Para isso, usamos a variável total. Em caso positivo, a função imprime na tela a mensagem “Fila vazia”. Caso contrário (linha 7), a função deve percorrer o vetor caminhoneiros, do primeiro elemento até o último (o número de elementos é armazenado na variável total), pelo laço na linha 8. Uma variável inteira auxiliar, x, é utilizada para percorrer o vetor. Cada elemento é impresso na linha 9.

A segunda solução apresentada é a que segue (Inep, 2017):

```

01  Procedimento mostrarFila()
02      Var
03          x : inteiro
04      Início
05          Para(x <- 1 até total passo 1) faça

```

```

06         imprima(caminhoneiros[x])
07         fim-para
08     Fim

```

Nesse caso, a função trabalha apenas com um laço (linha 5), que percorre o vetor caminhoneiros do início até o fim (armazenado na variável total) e também utiliza uma variável auxiliar inteira x. Essa função não mostra nada caso não exista nenhum elemento na fila.

3. Indicações bibliográficas

- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. *Introdução às estruturas de dados*. Rio de Janeiro: Campus, 2004.
- INEP. *Padrão de Resposta da Prova de Tecnologia em Análise e Desenvolvimento de Sistemas de 2017*. <<http://portal.inep.gov.br/educacao-superior/enade/provas-e-gabaritos>>. Acesso em 07 abr. 2020.
- SENNE, E. L. F. *Primeiro curso de programação em C*. 3 ed. Florianópolis: Visual Books, 2006.

Questão 9

Questão 9.⁹

A coordenação de gestão de pessoas (CGP) de uma empresa solicitou à equipe de TI que desenvolvesse um aplicativo (*app*) para criação de uma rede social *on-line* que incentivasse os empregados a idealizar e executar projetos. A fim de realizar o levantamento de requisitos, a equipe de desenvolvimento se reuniu com a CGP e alguns empregados e, para melhorar a comunicação entre os envolvidos, os analistas de sistemas da empresa optaram pela prototipação de telas do app, conforme os exemplos a seguir.

Protótipo 1

Protótipo 2

Protótipo 3

Tendo-se usado a abordagem de prototipação, foram documentados os seguintes requisitos:

- todo colaborador deve informar número de matrícula, nome, endereço de e-mail e senha de login para se cadastrar no app e poder registrar projetos e enviar comentários; a coordenação de um projeto é atribuída automaticamente ao colaborador que o registrar;
- cada projeto pode ser classificado em uma de duas possíveis categorias: "Melhoria de Processo" ou "Social"; além dos dados em comum para ambas as categorias, um projeto de "Melhoria de Processo" deve ser alocado ao departamento (selecionado de uma lista) em que

⁹Questão Discursiva 5 – Enade 2017.

ele será aplicado e os resultados esperados devem ser informados (conforme o protótipo 1); para um projeto "Social", deve-se indicar o público beneficiário (conforme o protótipo 2);

- uma vez registrado o projeto, qualquer colaborador pode visualizá-lo no app e registrar um ou mais comentários para contribuir com o refinamento da ideia (conforme o protótipo 3); os comentários são visualizados em ordem cronológica, por data e horário;
- o coordenador do projeto pode incluir até três colaboradores para participarem da equipe.

Considerando a situação e as informações apresentadas, construa um diagrama de classe que esteja em conformidade com os requisitos, indicando os atributos de cada classe e as associações entre as classes.

1. Análise da questão

Diagrama de classe

A questão solicita ao aluno que apresente um diagrama de classes para o cenário do enunciado. O enunciado apresenta, além dos requisitos, os protótipos de tela do aplicativo para celular.

Segundo Booch, Rumbaugh e Jacobson (2005), uma classe é “uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamento e semântica”. Na UML, especificamente no diagrama de classes, as classes são representadas como retângulos e há espaços reservados para representar seu nome, seus atributos e suas operações (seus métodos), como ilustrado na figura 1.

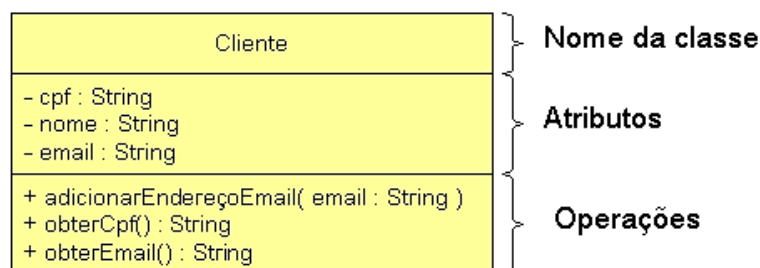


Figura 1. Diagrama de classes proposto.
BOOCH, RUMBAUGH e JACOBSON, 2005 (com adaptações).

O diagrama de classes apresenta, além das classes propriamente ditas, a maneira como as diversas classes se relacionam umas com as outras (de um ponto de vista estático). A UML

prevê diversos tipos de relacionamentos entre as classes, cada uma com uma notação específica: associações, herança, realização, dependência, agregação e composição.

Devemos ter em mente que o diagrama de classes é apenas um dos muitos diagramas propostos na UML. Os vários diagramas têm a finalidade de ilustrar os diversos aspectos de um sistema que está sendo modelado. Os diagramas podem focar nos aspectos estáticos ou nos aspectos dinâmicos do modelo e devem ser utilizados de forma a complementar.

Os diagramas listados a seguir focam nos aspectos estáticos de um sistema.

- Diagrama de classes.
- Diagrama de componentes.
- Diagrama de estrutura composta.
- Diagrama de objetos.
- Diagrama de implementação.
- Diagrama de artefatos.

Os diagramas listados a seguir capturam os aspectos dinâmicos de um sistema.

- Diagrama de caso de uso.
- Diagrama de sequência.
- Diagrama de comunicação.
- Diagrama de gráfico de estados.
- Diagrama de atividades.

Contudo, a questão foca apenas no diagrama de classes e, como vamos perceber, o gabarito requer que o aluno desenhe somente os aspectos principais do diagrama, uma vez que a solução sugerida não apresenta muitos detalhes das classes utilizadas.

Na resolução da questão, mantivemos a notação utilizada no gabarito publicado pelo Inep, no qual não se utilizam acentuação nem caracteres como o cedilha no texto do diagrama. A maioria das ferramentas de modelagem atual não apresenta mais problemas com o uso de caracteres diferentes dos caracteres da língua inglesa, mas, aqui, decidimos permanecer com o padrão adotado pelo Inep.

2. Resolução da questão

O aluno deve apresentar um diagrama que contenha as classes mostradas a seguir.

- Comentário.
- Colaborador.

- Projeto.
- Melhoria.
- Social.
- Departamento.

A resposta sugerida é apresentada na figura 2.

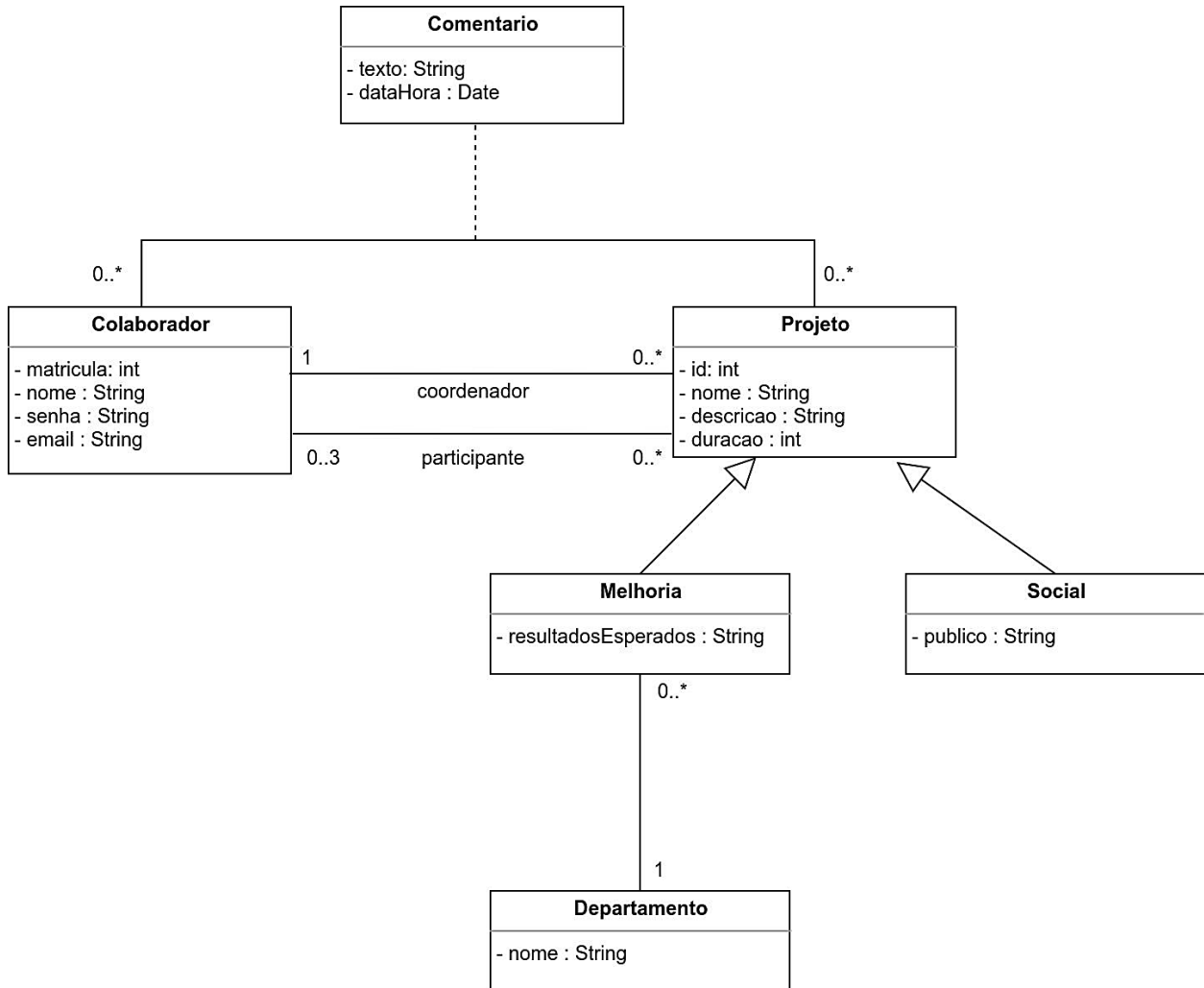


Figura 2. Diagrama de classes da questão.
Inep, 2017 (com adaptações).

Podemos observar que todas as classes apresentadas na solução têm apenas seus atributos listados, não os seus métodos. Além disso, todos os atributos listados são privados, o que significa que sua visibilidade é restrita apenas à própria classe.

A classe "Colaborador" apresenta os atributos "matricula", "nome", "senha" e "email". Essa classe tem três relacionamentos de associação com a classe Projeto; um deles é feito por uma classe associativa (a classe "Comentario"). A associação "coordenador" apresenta multiplicidade "um para muitos", de forma que um colaborador pode ser coordenador de zero ou mais projetos, mas um projeto deve ter apenas um colaborador como coordenador. A

associação “participante” deve permitir que um colaborador participe de zero ou fr mais projetos, mas um projeto pode ter de zero até três colaboradores como participantes.

A classe associativa (ou classe de associação) “Comentario” é necessária, visto que a informação contida em cada comentário (o texto e sua data de criação) não deve ser armazenada na classe “Projeto” nem na classe “Colaborador”. Assim, essa classe associativa deve apresentar os atributos “texto” (uma string) e “dataHora” (tipo date) para armazenar essas informações.

A classe “Projeto” deve dispor dos seguintes atributos: “id” (número inteiro), “nome” (string), “descricao” (string) e “duração” (número inteiro). Além dos relacionamentos de associação com a classe “Colaborador”, existe um relacionamento de herança com as classes “Melhoria” e “Social”. Essas duas classes herdam a classe “Projeto”. Assim, a ideia por trás dessa decisão de projeto é utilizar o mecanismo de herança para modelar os dois tipos de projeto: os de caráter de melhoria e os sociais.

A classe “Melhoria” tem um atributo adicional, dado pelos resultados esperados (modelados como uma string). A classe “Social” tem o atributo “publico”. Além disso, devemos ter mais uma classe: a classe “Departamento”. Essa classe apresenta o atributo “nome” (string) e está associada à classe “Melhoria”. A multiplicidade é tal que um departamento pode estar associado a zero ou a mais projetos de melhoria.

O gabarito ainda apresenta os critérios para a correção da questão (INEP, 2017), mostrados a seguir.

- *A utilização da herança para a distinção entre os tipos de projetos.*
- *A utilização das duas associações “coordenador” e “participante” entre as classes “Colaborador” e “Projeto”. O gabarito também adiciona a possibilidade do aluno utilizar uma agregação, ao invés de uma associação.*
- *O aluno pode utilizar uma classe de associação para o comentário (como mostrado na solução proposta) ou utilizar uma classe convencional, mas, nesse caso, as multiplicidades devem ser ajustadas.*
- *Deve ser criada também uma classe “Departamento”, para que se possa escolher o departamento no cadastro de projetos.*

Disponível em <<http://portal.inep.gov.br/educacao-superior/enade/provas-e-gabaritos>>. Acesso em 11 jul. 2025.

3. Indicações bibliográficas

- BELL, D. *The class diagram*. Disponível em <<https://developer.ibm.com/articles/the-class-diagram/>>. Acesso em 12 fev. 2020.
- BLAHA, M.; RUMBAUGH, J. *Modelagem e projetos baseados em objetos com UML 2*. 2. ed. Rio de Janeiro: Elsevier, 2006.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do usuário*. 2. ed. Rio de Janeiro: Elsevier, 2005.
- GAMMA, E. et al. *Design patterns: elements of reusable object-oriented software*. Boston: Addison-Wesley, 1995.
- INEP. *Padrão de Resposta da Prova de Tecnologia em Análise e Desenvolvimento de Sistemas de 2017*. Disponível em <<http://portal.inep.gov.br/educacao-superior/enade/provas-e-gabaritos>>. Acesso em 07 abr. 2020.
- LARMAN, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*. Porto Alegre: Bookman, 2007.
- PAGE-JONES, M. *Fundamentos do desenho orientado a objeto com UML*. São Paulo: Pearson, 2001.
- WIRFS-BROCK, R.; WILKERSON, B.; WIENER, L. *Designing object-oriented software*. Englewoods Cliffs: Prentice-Hall, 1990.

ÍNDICE REMISSIVO

Questão 1	Engenharia de requisitos. Requisitos funcionais. Requisitos não funcionais.
Questão 2	Engenharia de requisitos. Requisitos funcionais. Requisitos não funcionais.
Questão 3	Modelagem de dados. Modelo de entidade-relacionamento. Linguagem SQL.
Questão 4	Modelagem de dados. Modelo de entidade-relacionamento. Chave primária.
Questão 5	Algoritmos. Ordenação por inserção. Programação.
Questão 6	Interface Humano-computador. Acessibilidade. Programação web.
Questão 7	Estrutura de dados. Vetores. Matrizes. Programação.
Questão 8	Estrutura de dados. Filas. Programação.
Questão 9	Diagrama de classe. UML. Orientação a objetos.