



ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 3

CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 3

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.

Larissa Rodrigues Damiani

Doutora e Mestra em Ciências pelo programa de Engenharia Elétrica – Microeletrônica e Engenheira Elétrica – modalidade Eletrônica (ênfase em Telemática). Professora titular 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.

Questão 1

Questão 1.¹

Uma livraria usa um sistema informatizado para realizar vendas pela Internet. Optou-se por um sistema gerenciador de banco de dados, no qual aplicaram-se regras de corretude e integridade. Cada cliente se cadastra, faz login no sistema e escolhe títulos. Os livros são colocados em um carrinho de compras até que o cliente confirme ou descarte o pedido. As tabelas do sistema são: clientes, livros, carrinho, vendas. Dois clientes acessam o sítio no mesmo horário e escolhem alguns livros. O estado do sistema nesse instante é representado na tabela seguir.

	cliente	
	Benjamim	Alice
	Cálculo I	Estatística básica
	Estatística básica	História geral
livros	Inglês intermediário	

A livraria possui um único exemplar do livro Estatística básica. O computador usado por Alice sofre uma pane de energia antes que ela confirme o pedido. No mesmo instante também ocorre uma pane de energia no computador da loja. Em seguida, o computador de Alice e o computador da loja voltam a ter energia e a funcionar. Ela volta ao sistema e retoma seu carrinho de compras intacto.

Com relação a essa situação, julgue os itens seguintes.

- I. A consistência de uma tabela do banco de dados foi violada temporariamente, para disponibilizar dois exemplares do livro Estatística básica.
- II. Como o pedido de Alice continuou válido apesar da interrupção, o sistema gerenciador de banco de dados não emprega atomicidade.
- III. Todas as transações devem ser fechadas depois do retorno da energia, para que os clientes possam recuperar seus carrinhos de compras.

Assinale a opção correta.

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

¹Questão 35 – Enade 2008.

1. Introdução teórica

Transação e atomicidade

Transação é um programa em execução que forma uma unidade lógica de processamento no banco de dados, podendo incluir uma ou mais operações de acesso a ele.

Atomicidade é uma propriedade das transações que garante que elas sejam indivisíveis, isto é, ou a transação é executada totalmente ou não é executada.

A questão envolve a decisão de quando se deve iniciar uma transação. Enquanto o usuário estiver apenas navegando no site, ou colocando itens no carrinho de compras, não deve iniciar uma transação. Muitas vezes, os clientes adicionam os produtos no carrinho, mas depois o abandonam. Entre os motivos de abandono, temos:

- excesso de etapas no processo de checkout;
- a existência de formulários extensos, que coletam muitas informações de uma só vez;
- navegação confusa, com interfaces não intuitivas;
- falta de opções de pagamento;
- ausência de transparência, como custos adicionais, frete e impostos, revelados apenas na última etapa do checkout.

2. Análise das afirmativas

I – Afirmativa incorreta.

JUSTIFICATIVA. Vários clientes podem colocar um exemplar do livro no carrinho, desde que seu estoque não esteja zerado. Como não é possível saber qual cliente vai finalizar o processo de compra, é permitido que qualquer um possa colocá-lo em seu carrinho. Essa regra de negócio pode fazer com que um livro, com apenas uma unidade disponível em estoque, seja colocado simultaneamente em vários carrinhos de compra, sem que haja qualquer violação de consistência.

II – Afirmativa incorreta.

JUSTIFICATIVA. O fato de Alice não iniciar uma transação para completar o processo de compra (fazer o checkout) não significa que o gerenciador de banco de dados não emprega atomicidade. Uma afirmação não leva à outra.

III – Afirmativa incorreta.

JUSTIFICATIVA. Nenhuma transação foi aberta para a inclusão de itens no carrinho de compras. Portanto, a transação não precisa ser fechada. Para os clientes recuperarem os dados do carrinho de compras, é preciso apenas que o sistema preveja um armazenamento temporário desses dados e sua posterior recuperação. Por exemplo, podem ser usados cookies, sem nenhuma necessidade de acessar o banco de dados.

Alternativa correta: E.

3. Indicações bibliográficas

- E-COMMERCE BRASIL. *Abandono de carrinho do e-commerce atingiu 70,19%*. Disponível em <<https://www.ecommercebrasil.com.br/artigos/abandono-de-carrinho-no-e-commerce-atingiu-7019> >. Acesso em 16 out. 2025.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. São Paulo: McGraw-Hill, 2021.
- SOMMERVILLE, I. *Engenharia de software*. São Paulo: Pearson, 2021.

Questão 2

Questão 2.²

Após realizar uma análise de mercado em busca de soluções para aprimorar o seu negócio, uma empresa adquiriu um sistema de ERP (Enterprise Resource Planning) contendo um conjunto de módulos que integra todos os departamentos existentes. Após um ano de utilização, houve uma mudança na legislação e, para atender às novas exigências, foi necessária uma manutenção no sistema ERP. Considerando essa situação hipotética, é correto afirmar que a empresa irá realizar uma manutenção

- A. corretiva.
- B. adaptativa.
- C. aperfeiçoadora.
- D. preventiva.
- E. perfectiva.

1. Introdução teórica

Manutenção de software

No contexto de engenharia de software, os principais tipos considerados de manutenção de software são os citados abaixo.

- A **manutenção corretiva** é a modificação reativa de um produto de software, executada após a entrega, para corrigir problemas descobertos.
- A **manutenção preventiva** consiste na modificação de um sistema a fim de detectar e de corrigir defeitos latentes, ou seja, defeitos que ainda não se manifestaram durante a execução do software.
- A **manutenção adaptativa** é a modificação de um produto de software, executada após a entrega, a fim de mantê-lo utilizável em um ambiente alterado ou em alteração.
- A **manutenção perfectiva** é a modificação de um produto de software após a entrega, a fim de melhorar o desempenho, a manutenibilidade ou quaisquer outras qualidades do sistema.

²Questão 36 – Enade 2008.

2. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. Não é uma manutenção corretiva porque não foram identificados problemas no funcionamento do ERP.

B – Alternativa correta.

JUSTIFICATIVA. Para que o ERP possa continuar em operação corretamente, é necessário que ele sofra uma adaptação para se adequar à nova legislação que entrou em vigor. Logo, a empresa irá realizar uma manutenção adaptativa no sistema.

C – Alternativa incorreta.

JUSTIFICATIVA. Não existe um tipo de manutenção aperfeiçoadora.

D – Alternativa incorreta.

JUSTIFICATIVA. Não é uma manutenção preventiva porque ela não se propõe a identificar ou corrigir defeitos latentes.

E – Alternativa incorreta.

JUSTIFICATIVA. Não é uma manutenção perfectiva porque ela não se propõe a aperfeiçoar o sistema, mas sim a adaptá-lo a uma nova realidade.

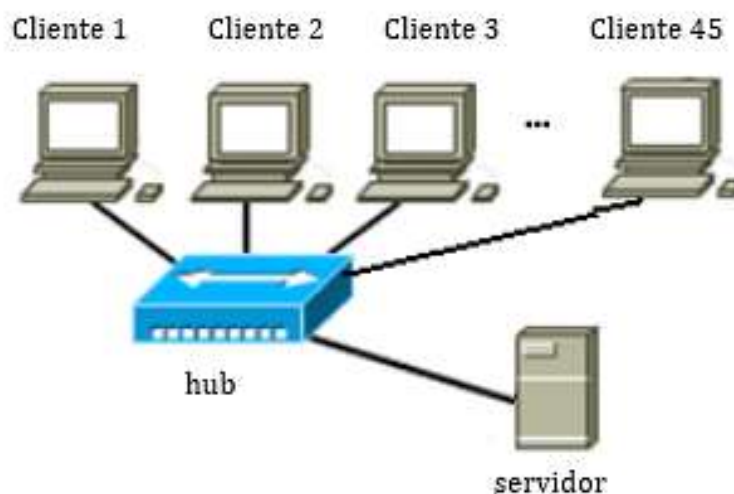
3. Indicações bibliográficas

- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. São Paulo: McGraw-Hill, 2021.
- SOMMERVILLE, I. *Engenharia de software*. São Paulo: Pearson, 2021.

Questão 3

Questão 3.³

As comunicações de um software cliente com o respectivo servidor instalado em um equipamento em uma LAN (local area network) apresentam lentidão. Após várias reclamações dos usuários, o responsável pelo software e o gerente de redes foram convocados para analisar o problema. A partir de uma análise da topologia da rede, foi identificado que o departamento possuía 45 microcomputadores clientes na mesma LAN, interligados ao servidor por meio de um único hub, conforme ilustrado na figura a seguir. Concluiu-se que o problema de lentidão estava relacionado ao elevado número de colisões na LAN.



A solução correta para esse problema hipotético é a substituição do hub por

- A. outros 4 hubs interligados contendo, cada um, menos portas de conexão, o que resultaria na diminuição do volume de colisões na LAN e aumentaria o desempenho da rede.
- B. um switch, que permitiria a diminuição do volume de colisões e uma melhoria no desempenho da rede.
- C. um switch, que traria benefícios relacionados ao desempenho da rede, uma vez que o número de colisões permaneceria inalterado.
- D. um switch, que diminuiria o desempenho da rede, uma vez que aumentaria o volume de colisões.
- E. um switch, que aumentaria o volume de colisões e melhoraria o desempenho da rede.

³Questão 37 – Enade 2008.

1. Introdução teórica

Hubs e switches

No contexto de redes de computadores, um **hub** é um dispositivo concentrador no qual são conectadas estações (PCs) e outros dispositivos de rede por meio de cabos de rede. Os dados enviados por um dispositivo são propagados na rede e recebidos por todos os demais.

Com um hub a propagação de dados ocorre por broadcast. O broadcast acontece quando uma transmissão tem como destino todos os dispositivos conectados na rede.

O **switch** é um dispositivo de conexão que permite que os dados se movimentem entre vários computadores em uma rede ao mesmo tempo. Um switch custa mais do que um hub, mas pode mover dados mais rápido quando muitas pessoas estiverem usando a rede de uma só vez.

A forma de trabalho do hub é mais simples se comparada ao switch. O hub recebe dados vindos de um computador e os transmite às outras máquinas. Quando isso ocorre, nenhum outro computador consegue enviar sinal. Isso acontece somente após o sinal anterior ter sido completamente distribuído. Em um hub, é possível ter várias portas, ou seja, entradas para conectar o cabo de rede de cada computador. Geralmente, há aparelhos com 8, 16, 24 e 32 portas.

Hubs podem ser adequados para redes pequenas e/ou domésticas. Havendo poucos computadores, é muito pouco provável que surja algum problema de desempenho ou colisão. As colisões geralmente acontecem quando duas ou mais estações transmitem simultaneamente dentro de um domínio. Uma colisão simples é uma que foi detectada enquanto se tentava transmitir um quadro, mas que, na tentativa seguinte, o quadro foi transmitido com êxito. Colisões múltiplas indicam que o mesmo quadro colidiu repetidamente antes de ser transmitido com êxito.

O switch é um aparelho muito semelhante ao hub, mas tem uma grande diferença: os dados vindos do computador de origem somente são repassados ao computador de destino. Isso ocorre porque os switches criam uma espécie de canal de comunicação exclusiva entre a origem e o destino. Dessa forma, a rede não fica "presa" a um único computador no envio de informações. Isso aumenta o desempenho da rede, já que a comunicação está disponível, exceto quando dois ou mais computadores tentam enviar dados simultaneamente à mesma máquina. Essa característica também diminui a ocorrência de colisões de pacotes. Além disso,

o switch tem a capacidade de negociar a velocidade e o modo de operação diretamente com a estação. Assim como no hub, é possível ter várias portas em um switch.

Atualmente, os hubs estão caindo em desuso, e estão sendo gradualmente substituídos pelos switches, que custam cada vez menos e apresentam as vantagens citadas anteriormente.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. O aumento na quantidade de hubs não resolve o problema das colisões porque eles não têm inteligência para direcionar o tráfego de máquina para máquina. A conexão vai continuar comum a todos os computadores e dispositivos na rede.

B – Alternativa correta.

JUSTIFICATIVA. O switch tem inteligência para diminuir as colisões na rede e, com isso, aumenta o seu desempenho.

C – Alternativa incorreta.

JUSTIFICATIVA. O número de colisões diminuirá com o switch. Se permanecesse inalterado, não haveria motivo para colocar o switch.

D – Alternativa incorreta.

JUSTIFICATIVA. Se houvesse aumento das colisões, o desempenho diminuiria mais, tornando o problema original ainda pior. O switch, no entanto, faz exatamente o contrário.

E – Alternativa incorreta.

JUSTIFICATIVA. O aumento das colisões não aumenta o desempenho da rede, mas o diminui.

2. Indicações bibliográficas

- BARRET, D. *Redes de computadores*. Rio de Janeiro: LTC, 2010.
- MORAES, A. F. *Redes de computadores: fundamentos*. 6. ed. São Paulo: Érica, 2008.
- TANENBAUM, A. S.; FEAMSTER, N.; WETHERALL, D. J. *Redes de computadores*. 6. ed. Rio de Janeiro: Bookman, 2021.

Questão 4

Questão 4.⁴

Uma montadora de automóveis produz carros de luxo e esportivos. Um carro é formado de várias partes e cada parte pode ser fabricada por diferentes fornecedores. Um gerente ou um operador possui permissão para cadastrar partes do carro, desde que ainda inexistentes no sistema, e consultar a sua disponibilidade para a fabricação dos carros. Se o estoque dessas partes está abaixo do limite mínimo estipulado, o sistema envia um pedido ao respectivo fornecedor.

Considerando a situação acima, faça o que se pede a seguir.

- A) desenhe o diagrama de caso de uso correspondente à situação apresentada.
- B) escolha um caso de uso no diagrama elaborado e descreva-o em termos de ator e fluxo principal.
- C) descreva um tratamento de exceção para cada caso de uso do diagrama elaborado.

1. Introdução teórica

Diagrama de casos de uso

O **diagrama de casos de uso** tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Ele descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. O cliente deve ver no diagrama de casos de uso as principais funcionalidades de seu sistema. É representado por:

- Atores;
- Casos de uso;
- Relacionamentos entre esses elementos.

Casos de uso podem, opcionalmente, estar envolvidos por um retângulo que representa os limites do sistema (fronteira ou boundary). Os participantes de um diagrama de caso de uso estão detalhados a seguir.

• Ator



Um ator é representado por um boneco e por um rótulo com o seu nome. Um ator é um usuário do sistema e pode ser um usuário humano ou outro elemento computacional (software ou hardware).

⁴Questão 38 - Discursiva – Enade 2008.

- **Caso de uso**



Um caso de uso é representado por uma elipse e por um rótulo com o seu nome. Um caso de uso define uma grande função do sistema. A implicação é que uma função pode ser estruturada em outras funções e, portanto, um caso de uso pode ser estruturado.

- **Relacionamentos**

Os relacionamentos representam os vínculos ou dependências entre os atores e os casos de uso, ou entre os próprios casos de uso.

Eles mostram como os elementos interagem ou como um comportamento depende de outro dentro do sistema.

Os principais tipos de relacionamentos são: associação, generalização, include e extend.

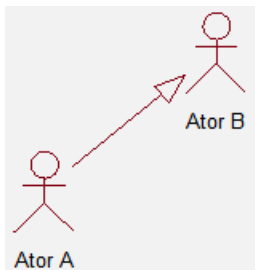
- **Associação**



Define uma funcionalidade do sistema do ponto de vista do usuário.

- Entre ator e caso de uso

- **Generalização**



- O ator A é um tipo do ator B. Por exemplo, A é um gerente e B é um funcionário, significando que um gerente é um funcionário.
- A tem seus próprios casos de uso, além daqueles que herda de B.

- Entre atores ou entre casos de uso

- **Include**

Um relacionamento include, de um caso de uso A para um caso de uso B, indica que B é essencial para o comportamento de A. Pode ser dito também que B é parte de A.

- **Extend**

Um relacionamento extend, de um caso de uso A para um caso de uso B, indica que o caso de uso B pode ser acrescentado para descrever o comportamento de A (não é essencial). A extensão é inserida em um ponto de extensão do caso de uso A.

2. Resolução da questão

A) O diagrama que soluciona a questão está na figura 1.

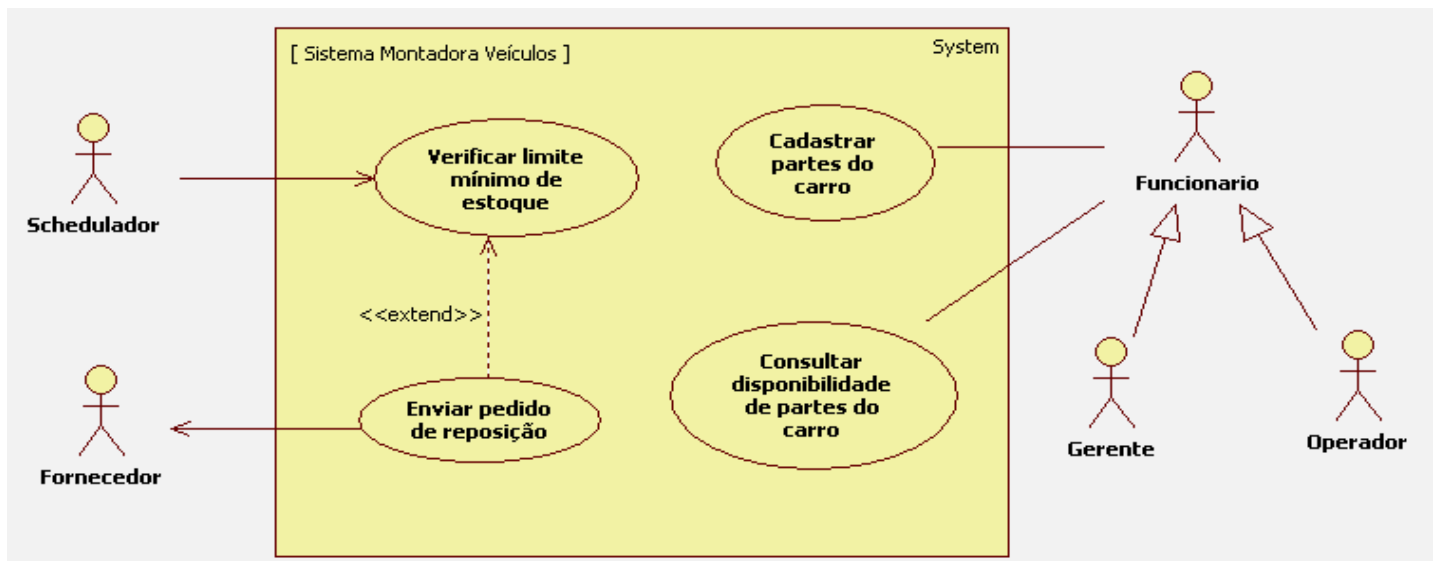


Figura 1. Diagrama de Caso de Uso – montadora de veículo.

A generalização de funcionário simplifica o diagrama porque tanto o gerente quanto o operador podem executar as mesmas funcionalidades.

Os atores são: schedulador (função do sistema que dispara automaticamente), fornecedor e funcionário (que pode ser um gerente ou um operador).

B) Verificar o quadro abaixo.

Caso de Uso:	Verificar limite mínimo de estoque.
Atores:	Schedulador, fornecedor.
Pré-Condições:	Não há.
Pós-Condições:	Não há.
Entradas:	Horário de execução da rotina de verificação de estoque.
Fluxo Principal	
1. O schedulador é disparado no horário programado (por exemplo, 23h).	
2. O sistema verifica se a quantidade de partes em estoque está abaixo do limite mínimo.	
3. O sistema confirma que o limite mínimo foi atingido.	
4. O sistema gera um pedido ao fornecedor.	
5. O sistema envia o pedido gerado ao fornecedor (via e-mail).	
Fluxo Alternativo	
1. No item 3. O sistema verifica que o limite mínimo não foi atingido.	
2. O sistema encerra o processamento sem gerar pedido.	
Ponto de Extensão	
Item 5. Caso de Uso: Enviar Pedido de Reposição.	

C) Os números no tratamento de exceção referenciam o fluxo principal.

Tratamento de Exceção
2. A tabela de estoque não está disponível para consulta. 2.1. O sistema gera uma mensagem de erro e envia por e-mail para o gerente de estoque. 4. Pedido para o Fornecedor não foi gerado por erro no banco de dados. 4.1. O sistema gera uma mensagem de erro e envia por e-mail para o gerente de suporte de banco de dados. 5. O pedido não foi enviado para o fornecedor. 5.1. O sistema gera uma mensagem de erro e envia para o e-mail do comprador responsável pelo fornecedor.

3. Indicações bibliográficas

- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. *UML - Guia do usuário*. 2. ed. Rio de Janeiro: LTC, 2006.
- GUEDES, G. T. A., *UML 2 – uma abordagem prática*. São Paulo: Novatec, 2018.
- LARMAN, C. *Utilizando UML e padrões*. 3. ed. Porto Alegre: Bookman, 2007.

Questão 5

Questão 5.⁵

Uma montadora de automóveis produz carros de luxo e esportivos. Um carro tem marca, modelo, chassi e ano de fabricação. As partes de um carro possuem características como: nome, quantidade, cor e preço. Um fornecedor da montadora tem CNPJ e razão social. O carro de luxo possui sistema GPS; o carro esporte não possui sistema GPS e somente pode ser fabricado na cor vermelha.

Com base nessa situação, faça o que se pede a seguir.

- A) Identifique e escreva o nome das classes correspondentes à situação apresentada.
- B) Desenhe o diagrama de classes, contendo somente os nomes das classes e seus relacionamentos.
- C) Identifique e escreva para as classes seus atributos e métodos, utilizando os símbolos de visibilidade proposto na UML. Os métodos devem estar com sua assinatura completa e obedecerem às regras de encapsulamento da orientação a objetos.

1. Introdução teórica

Diagrama de classes

O **diagrama de classes** representa os vários tipos de objetos no sistema e o relacionamento entre eles. Pode oferecer três perspectivas, cada uma para um tipo de observador diferente. Elas estão descritas a seguir.

- **Conceitual**

- Representa os conceitos do domínio em estudo.
- Apresenta perspectiva destinada ao cliente.

- **Especificação**

- Tem foco nas principais interfaces da arquitetura, nos principais métodos, e não como eles serão implementados.
- É destinada às pessoas que não precisam saber detalhes de desenvolvimento, tais como gerentes de projeto.

⁵Questão 39 - Discursiva – Enade 2008.

- **Implementação (a mais utilizada de todas)**

- Aborda vários detalhes de implementação, tais como navegabilidade, tipo dos atributos etc.
- Perspectiva destinada ao time de desenvolvimento.

O diagrama de classes é composto de associações, atributos e operações. **Associações** representam relações entre as classes. **Atributos** são características das classes, sua visão estrutural. **Operações** (ou métodos) são os comportamentos das classes, ou seja, o que elas fazem, isto é, os processos que a classe executa. O diagrama de classes descreve uma visão estática do sistema, mostrando o relacionamento entre elas. Uma classe é representada por um retângulo dividido em três compartimentos: nome, atributos e operações, conforme mostrado na figura 1.

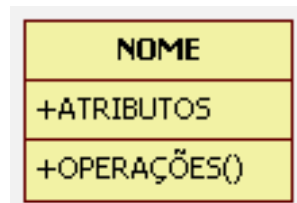


Figura 1. Representação de uma classe.

A visibilidade de um elemento de uma classe indica se ele pode ou não ser visualizado por outras classes, conforme indicado no quadro 1.

Quadro 1. Visibilidade de um elemento.

Pública	Pode ser visto e usado por outras classes
Protegida	Pode ser visto por classes dentro do mesmo pacote e por subclasses
Padrão	Pode ser visto somente por classes dentro do mesmo pacote
Privada	Não pode ser visto por outras classes.

2. Resolução da questão.

A) Nomes das classes:

- Carro.
- ParteDoCarro.
- TipoCarro.
- Fornecedor.
- Cor (opcional).

B) Verificar a figura 2.

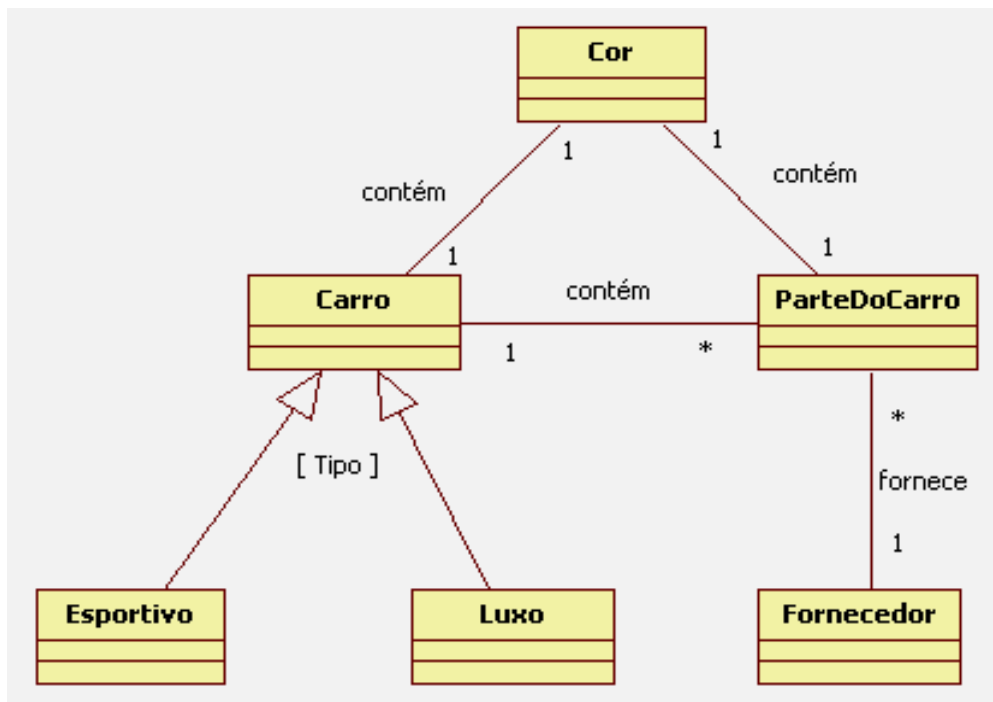


Figura 2. Diagrama de classes.

C) Verificar a figura 3.

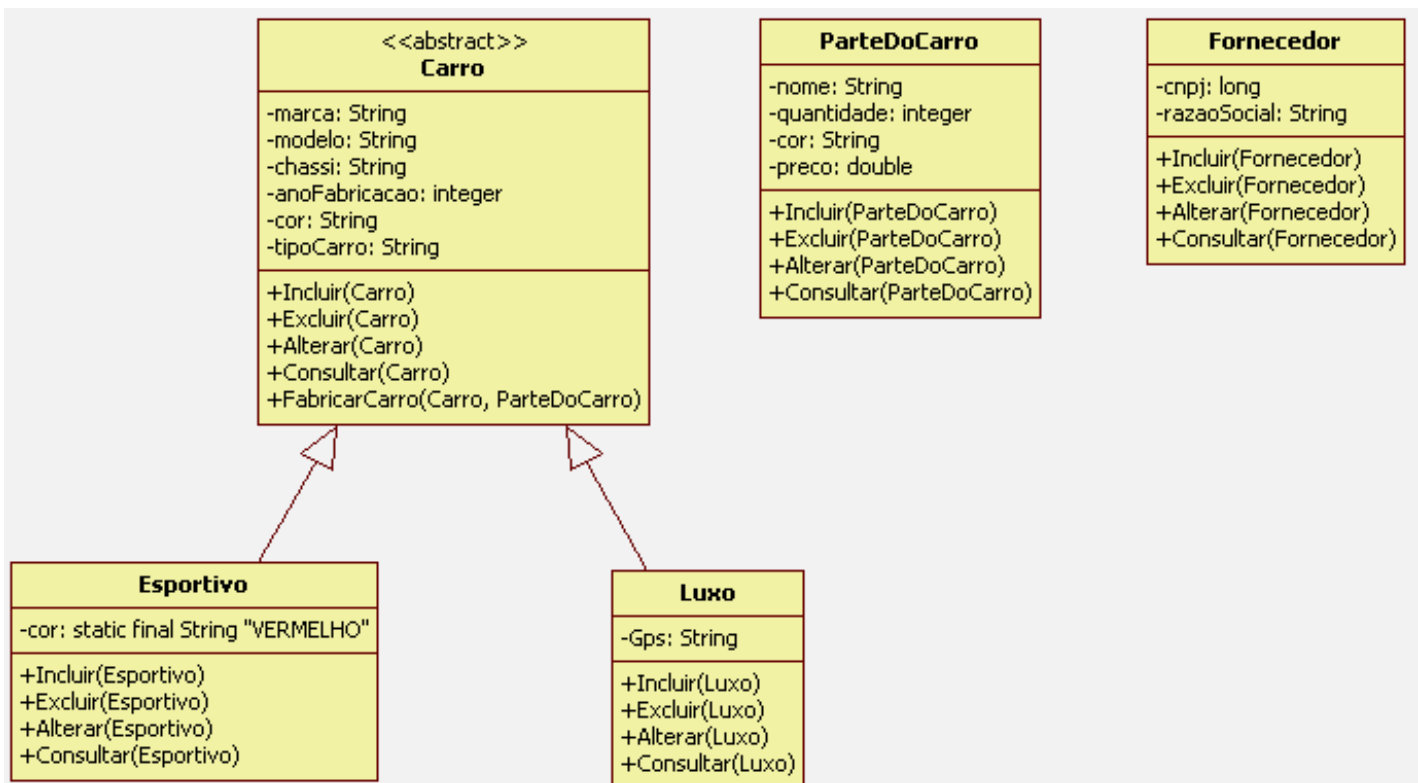


Figura 3. Classe carro.

Os métodos na classe carro são herdados e sobrescritos nas classes Esportivo e Luxo (subclasses) para atenderem às particularidades de cada tipo de fabricação. A classe Carro foi tornada abstrata, de modo que somente é possível executar os métodos nas subclasses.

3. Indicações bibliográficas

- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. *UML - guia do usuário*. 2. ed. Rio de Janeiro: LTC, 2006.
- GUEDES, G. T. A., *UML 2 – uma abordagem prática*. São Paulo: Novatec, 2018.
- LARMAN, C. *Utilizando UML e padrões*. 3. ed. Porto Alegre: Bookman, 2007.

Questão 6**Questão 6.**⁶

Leia o algoritmo a seguir.

```

01. Algoritmo MatrizENADE2008
02. variaveis
03.     M[0..2][0..3], I, J, C : inteiro
04. inicio
05.     C ← 0
06.     para I ← 0 ate 2 passo 1 faca
07.         para J ← 0 ate 3 passo 1 faca
08.             C ← C + 1
09.             M[I][J] ← C
10.         fim para
11.     fim para
12.     para I ← 0 ate 2 passo 1 faca
13.         para J ← 0 ate 3 passo 1 faca
14.             C ← M[2-I][3-J]
15.             M[I][J] ← C
16.         fim para
17.     fim para
18. fim algoritmo

```

Considerando a execução do algoritmo apresentada acima, faça o que se pede a seguir.

A) Apresente os dados da matriz M ao término da execução da linha 10.

B) Apresente os dados da matriz M ao término da execução da linha 15.

1. Introdução teórica**Algoritmos**

As questões envolvendo a interpretação de resultados de algoritmos demandam muita atenção e cuidado. O erro em um passo impossibilita encontrar o resultado correto para toda a questão.

⁶Questão 40 - Discursiva – Enade 2008 (com adaptações).

É fundamental entender o que o algoritmo está fazendo. Ele executa as seguintes macroatividades:

- inicializa as variáveis (linha 3);
- realiza o primeiro loop (laço ou execução) na matriz bidimensional (linhas 5 a 11);
- realiza o segundo loop na matriz bidimensional (linhas 12 a 17).

A questão não solicita a verificação dos resultados ao final de cada loop, e sim em pontos intermediários (linhas 10 e 15). Se tivesse solicitado que fossem mostrados os resultados após as linhas 11 e 17, então poderíamos esperar a completa execução dos dois loops e depois mostrar o resultado. Mas não é o caso, o que demanda cuidado adicional.

Um ponto importante é que as variáveis numéricas são inicializadas com zero. Portanto, a matriz bidimensional, após a linha 3, tem o formato a seguir.

```
0 0 0 0
0 0 0 0
0 0 0 0
```

É uma matriz 3x4, com os índices variando de 0 a 2 (3 valores) e de 0 a 3 (4 valores). O total de membros é 12. A matriz mostrada acima é a entrada do algoritmo. Ela sofrerá o processamento, conforme será explicado a seguir.

2. Resolução da questão

A) A entrada para o processamento é a matriz bidimensional com todos os valores zerados, conforme abaixo.

```
0 0 0 0
0 0 0 0
0 0 0 0
```

É fundamental entender como a matriz bidimensional está sendo processada na questão. O índice I varre as linhas e o índice J varre as colunas. Então, para o índice I começando com 0 (zero), o índice J variará de 0 (zero) até 3 (três). Como a questão solicita a visualização da matriz após a linha 10, o índice I somente conterá o valor 0 (zero). Isso significa que teremos de tirar uma “foto” da matriz quando o índice I for zero e o índice J variar de 0 até 3.

O quadro 1 mostra a evolução do processamento até a linha 10.

Quadro 1. Evolução do processamento até a linha 10.

I	J	C	M[I,J]
0	0	1	1
0	1	2	2
0	2	3	3
0	3	4	4

Portanto, a saída da matriz após a linha 10 é a seguinte:

```

1 2 3 4
0 0 0 0
0 0 0 0

```

B) Para resolver o segundo item da questão, é fundamental entender que a entrada para o item B não é a saída do item A. Ou seja, para que tenhamos os dados corretos que iniciarão o segundo loop, não podemos considerá-los como sendo a resposta ao item A, que solicitou a visualização após a linha 10. A entrada para o item B é a matriz resultante após a linha 11. Portanto, teremos que continuar o processamento do algoritmo após a linha 10, conforme apresentado no quadro 2.

Quadro 2. Evolução do processamento até a linha 11.

I	J	C	M[I,J]
0	0	1	1
0	1	2	2
0	2	3	3
0	3	4	4
1	0	5	5
1	1	6	6
1	2	7	7
1	3	8	8
2	0	9	9
2	1	10	10
2	2	11	11
2	3	12	12

Portanto, a matriz que é a entrada para o processamento do item B é:

```

1 2 3 4
5 6 7 8
9 10 11 12

```

Outro ponto de atenção é o momento em que é pedida a visualização da matriz do item B. Na linha 19, diferentemente do item A, só é processada uma coluna do segundo loop, a posição onde $I=0$ e $J=0$. A atribuição é feita trazendo o dado que se encontra na posição $M[2-I][3-J]$ para a posição $M[I][J]$.

Como tanto I quanto J valem 0 (zero) nesse momento do processamento, a posição é $M[2][3]$, porque $2-0=2$ e $3-0=3$. Essa é a posição de origem. A posição de destino é $M[0][0]$.

O valor que se encontra em $M[2][3]$ é 12. Esse valor será colocado em $M[0][0]$. Com isso, a matriz fica conforme abaixo:

```

12 2 3 4
 5 6 7 8
 9 10 11 12

```

O programa Java abaixo pode ser usado para verificar os resultados apresentados.

```

package enade;
public class Main {
    public static void main(String[] args) {
        int M[][] = new int[3][4];
        int c = 0;
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = c + 1;
                M[I][J] = c;
            }
        }
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = M[2 - I][3 - J];
                M[I][J] = c;
            }
        }
    }
}

```

Para ver o resultado do item A, execute o programa abaixo, que inclui a visualização da matriz no ponto exato após a linha 10.

```

package enade;
public class Main {
    public static void main(String[] args) {
        int M[][] = new int[3][4];
        int c = 0;
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = c + 1;
                M[I][J] = c;
            }
            System.out.println(M[0][0] + " " + M[0][1] + " " + M[0][2] + " " +
M[0][3]);
            System.out.println(M[1][0] + " " + M[1][1] + " " + M[1][2] + " " +
M[1][3]);
            System.out.println(M[2][0] + " " + M[2][1] + " " + M[2][2] + " " +
M[2][3]);
            System.out.println(" ");
            return;
        }
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = M[2 - I][3 - J];
                M[I][J] = c;
            }
        }
    }
}

```

Para ver o resultado do item B, execute o programa a seguir, que inclui a

visualização da matriz no ponto exato após a linha 15.

```
package enade;
public class Main {
    public static void main(String[] args) {
        int M[][] = new int[3][4];
        int c = 0;
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = c + 1;
                M[I][J] = c;
            }
        }
        for (int I = 0; I < 3; I++) {
            for (int J = 0; J < 4; J++) {
                c = M[2 - I][3 - J];
                M[I][J] = c;
                System.out.println(M[0][0] + " " + M[0][1] + " " + M[0][2] + " "
+ M[0][3]);
                System.out.println(M[1][0] + " " + M[1][1] + " " + M[1][2] + " "
+ M[1][3]);
                System.out.println(M[2][0] + " " + M[2][1] + " " + M[2][2] + " "
+ M[2][3]);
                System.out.println(" ");
                return;
            }
        }
    }
}
```

3. Indicações bibliográficas

- BHARGAVA, A. Y. *Entendendo algoritmos*. São Paulo: Novatec, 2017.
- CORMEN, T. H. *Algoritmos – teoria e prática*. São Paulo: LTC, 2012.
- 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 7**Questão 7.⁷**

Leia o algoritmo a seguir.

```

01. funcao busca(V[0..9] : inteiro, K : inteiro) : inteiro
02. variaveis
03.     C, F, K, M : inteiro
04. inicio
05.     F ← 9
06.     [_____]
07.     enquanto((F > C) ou (V[M] <> K))
08.         [_____]
09.         se(K < V[M]) entao
10.             F ← M - 1
11.         senao
12.             [_____]
13.         fim se
14.     fim enquanto
15.     se(V[M] <> K) entao
16.         retorne(0)
17.     senao
18.         retorne(M)
19.     fim se
20. fim

```

O algoritmo representado pelo pseudocódigo acima está incompleto, pois faltam 3 linhas de código. A função busca desse algoritmo recebe um vetor ordenado de forma crescente e um valor a ser pesquisado. A partir disso, essa função verificará se o número armazenado no ponto mediano do vetor é o número procurado. Se for o número procurado, retornará o índice da posição do elemento no vetor e encerrará a busca. Se não for, a função segmentará o vetor em duas partes a partir do ponto mediano, escolherá o segmento no qual o valor procurado está inserido, e o processo se repetirá.

⁷Questão 22 – Enade 2008.

A partir dessas informações, assinale a opção que contém os comandos que completam, respectivamente, as linhas 6, 8 e 12 do algoritmo.

- A. $C \leftarrow 0$
 $M \leftarrow (C + F) / 2$
 $C \leftarrow M + 1$
- B. $C \leftarrow 1$
 $M \leftarrow (C + F) / 2$
 $C \leftarrow M - 1$
- C. $C \leftarrow 0$
 $C \leftarrow M + 1$
 $M \leftarrow (C + F) / 2$
- D. $C \leftarrow 1$
 $C \leftarrow M + 1$
 $M \leftarrow (C + F) / 2$
- E. $C \leftarrow 1$
 $M \leftarrow (C + F) / 2$
 $C \leftarrow M + 1$

1. Introdução teórica

Algoritmos

A resolução de questões envolvendo algoritmos demanda muito cuidado e atenção. Em questões desse tipo, é fundamental entendermos o papel de cada variável no algoritmo (C, F, K e M):

- C – contém o índice da posição inicial do vetor.
- F – contém o índice da posição final do vetor.
- K – contém o valor a ser pesquisado no vetor.
- M – contém o índice no ponto mediano do vetor, que é conseguido pela média entre C e F: $(C + F) / 2$.
- $V[M]$ – contém o valor do vetor no ponto médio em determinado momento.

O algoritmo busca comparar o valor em $V[M]$ com a variável K , que contém o valor a ser buscado. Observe que a sequência (loop ou laço, representado pela palavra enquanto) é executada sob duas condições:

1. o índice final (F) é maior que o índice inicial (C);
2. o Valor mediano $V[M]$ é diferente de K , portanto não é o valor desejado;

Se analisarmos os índices do vetor, veremos que certo valor desejado pode estar acima ou abaixo do índice mediano, conforme indicado a seguir.

0	1	2	3	4	5	6	7	8	9
---	---	---	---	----------	---	---	---	---	---

O índice mediano (M) no vetor acima é 4 (destacado em negrito), resultado de 9 dividido por 2, que dá 4.5, mas, como os números são inteiros, a parte decimal é descartada.

Para achar o número desejado temos 3 possibilidades:

1. o número desejado está no índice 4. Neste caso o algoritmo não entra no loop enquanto e termina retornando o índice mediano, que é o comando `retorne(M)`;
2. o número desejado está acima do índice mediano;
3. o número desejado está abaixo do índice mediano.

Como o algoritmo está incompleto, temos que analisar como ele está se comportando na parte revelada, para entendermos o que a parte faltante deverá fazer.

Na linha 5, o valor de F é inicializado com 9, que é o índice final.

Na linha 6, o valor de C pode ser inicializado tanto com 0 (opções A e C) quanto com 1 (opções B, D e E). Como C representa o início do índice do vetor, variando de 0 a 9, então é correto supormos que o valor inicializado seja 0, para que os índices alcancem todo o vetor.

Portanto, iniciamos os testes com as opções A e C. Somente se elas falharem é que consideramos as demais opções.

Para acharmos o comando correto para a linha 8, devemos analisar a linha imediatamente posterior. Na linha 9, o valor de K (valor de busca desejado) é comparado com o valor mediano ($V[M]$). Se analisarmos o valor de M até este ponto, ele está zerado. Na linha 3, M é definido como inteiro e inicializado com 0 (zero). Na linha 7, $V[M]$ é comparado com K . Se o valor buscado estiver na posição $V[0]$ nesta linha, o algoritmo será interrompido e K encontrado.

Portanto, entre a linha 7 e a linha 9, o valor de M terá que ser modificado, porque M está zerado na linha 7. O índice 0 (zero) é o primeiro índice e não é permitido que K busque

um valor em um índice **menor** que o valor inicial do vetor porque, dessa maneira, o algoritmo estaria errado, pois não estaria procurando K **dentro do vetor**.

Logo, é necessário que o valor de M seja atualizado no ponto entre a linha 7 e 9. A opção C da questão atribui outro valor à variável C na linha 8. A opção A recalcula M na linha 8. Portanto, a opção A é que traz os elementos lógicos para a correta resolução deste algoritmo. Somente se ela estiver errada é que testaremos a opção C, assim como fizemos com as demais opções (B, D e E). Incluindo os dados da opção A no algoritmo, ele fica conforme mostrado a seguir.

```

01. funcao busca(V[0..9] : inteiro, K : inteiro) : inteiro
02. variaveis
03.     C, F, K, M : inteiro
04. inicio
05.     F ← 9
06.     C ← 0
07.     enquanto((F > C) ou (V[M] <> K))
08.         M ← (C + F) / 2
09.         se(K < V[M]) entao
10.             F ← M - 1
11.         senao
12.             C ← M + 1
13.         fim se
14.     fim enquanto
15.     se(V[M] <> K) entao
16.         retorne(0)
17.     senao
18.         retorne(M)
19.     fim se
20. fim

```

Estudando o comportamento dos índices para a solução acima, temos o que segue.

1. Se $K = V[M]$ o algoritmo faz o teste da linha 7 e vai para a linha 16, retornando imediatamente M conforme o comando da linha 17.

2. Se K for **maior** que o valor contido na mediana, o algoritmo se comporta conforme abaixo, incrementando a variável C a cada passo.

Variável C	Variável F	Variável M (C + F) / 2
0	9	4
1	9	5
2	9	5
3	9	6
4	9	6
5	9	7
(assim sucessivamente)		

3. Se K for **menor** que o valor contido na mediana, o algoritmo se comporta conforme segue, decrementando a variável F a cada passo.

Variável C	Variável F	Variável M (C + F) / 2
0	4	4
0	4	4
0	3	3
0	3	3
(assim sucessivamente)		

Assim, a variável K é sempre encontrada, se estiver no Vetor, estando inicialmente abaixo ou acima da mediana. Se K não existir no Vetor, o valor da variável F se igualará ao valor da variável C e o algoritmo terminará com `retorne(0)`.

Outra maneira de resolver o mesmo algoritmo é por meio da “força bruta”, isto é, testar todos os itens de A até E, variando cada alternativa. Recomendamos a abordagem apresentada anteriormente por eliminar as respostas aparentemente incorretas e testar somente as possíveis respostas corretas.

2. Análise das alternativas

A – Alternativa correta.

JUSTIFICATIVA. A variável C é inicializada com 0, que é o índice inicial do Vetor. A mediana é calculada na linha 8. E, por fim, o valor de C é incrementado, fazendo a lógica inversa com o valor F que é decrementado.

B – Alternativa incorreta.

JUSTIFICATIVA. Inicializa o Vetor no índice 1, o que está incorreto, porque a busca ocorrerá somente entre os índices 1 e 9. Além do mais, a variável C é decrementada, comportamento que só faz sentido com a variável F.

C – Alternativa incorreta.

JUSTIFICATIVA. Inicializa corretamente o índice C com 0, mas o próximo comando apenas altera o valor de C. A mediana é calculada num ponto do algoritmo onde não faz mais sentido.

D – Alternativa incorreta.

JUSTIFICATIVA. Inicializa o Vetor no índice 1, o que está incorreto, porque a busca ocorrerá somente entre os índices 1 e 9.

E – Alternativa incorreta.

JUSTIFICATIVA. Inicializa o Vetor no índice 1, o que está incorreto, porque a busca ocorrerá somente entre os índices 1 e 9.

3. Indicações bibliográficas

- BHARGAVA, A. Y. *Entendendo algoritmos*. São Paulo: Novatec, 2017.
- CORMEN, T. H. *Algoritmos – teoria e prática*. São Paulo: LTC, 2012.
- 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 8

Questão 8.⁸

Com relação à forma como o RUP trata a análise de requisitos, assinale a opção correta.

- A. A análise de requisitos ocorre na fase de construção, quando são descritos todos os casos de uso, e em seguida modelados por meio de diagramas de casos de uso UML.
- B. A análise de requisitos ocorre na fase de elaboração, em que são feitas entrevistas com usuários e definição do escopo do projeto.
- C. A maior parte da análise de requisitos ocorre durante a fase de elaboração.
- D. Por se tratar de um processo iterativo e evolutivo, a análise de requisitos ocorre na fase de construção juntamente com a programação, o que permite que os requisitos sejam revistos.
- E. A análise de requisitos deve acontecer antes da programação e testes do sistema, não podendo sofrer alterações a partir do momento que estejam definidos.

1. Introdução teórica

Fases do RUP

O RUP (Rational Unified Process) é um processo de engenharia de software que define um conjunto de práticas e de diretrizes para ajudar as equipes de desenvolvimento a criar produtos de software de alta qualidade, gerenciando os riscos ao longo do processo. O RUP não é uma metodologia rígida, mas sim um processo iterativo e incremental que pode ser adaptado às necessidades específicas de um projeto. As fases do RUP são: iniciação, elaboração, construção e transição.

A análise de requisitos ocorre durante todo o ciclo de vida de software, mas com diferentes ênfases em cada fase. A maior parte do trabalho referente a requisitos ocorre nas fases de iniciação e de elaboração. Mas é na fase de elaboração que ocorre a maior quantidade de trabalho. Na fase de construção e na fase de transição, o trabalho decresce muito.

A meta da fase de iniciação é atingir o consenso dos envolvidos sobre os objetivos do ciclo de vida do projeto. Um de seus objetivos principais é estabelecer o escopo do software do projeto e as condições limite, incluindo uma visão operacional, critérios de aceitação e o que deve ou não estar no produto.

⁸Questão 23 – Enade 2008.

A meta da fase de elaboração é assegurar que os planos de projeto traçados, a arquitetura e os requisitos estejam estáveis o suficiente para autorizar a programação e o desenvolvimento. Uma atividade básica dessa fase é refinar a visão, com base nas informações novas obtidas durante a fase, estabelecendo uma compreensão sólida dos casos de uso mais críticos que conduzem às decisões de arquitetura e de planejamento.

Na fase de elaboração, são feitas várias atividades em requisitos, dentre elas a descrição detalhada dos fluxos de eventos dos casos de uso de maneira que os clientes e usuários possam compreendê-los.

A meta da fase de construção é esclarecer os requisitos restantes e concluir o desenvolvimento do sistema com base na arquitetura baseline escolhida. Um de seus objetivos principais é atingir versões úteis do produto com qualidade, eficiência e rapidez.

Conforme o RUP, as finalidades da disciplina requisitos são

- estabelecer e manter concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer;
- oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos do sistema;
- definir as fronteiras do sistema (ou delimitar o sistema);
- fornecer base para planejar o conteúdo técnico das iterações;
- fornecer base para estimar o custo e o tempo de desenvolvimento do sistema;
- definir uma interface de usuário para o sistema, focando nas necessidades e metas dos usuários.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. A análise de requisitos ocorre em todas as fases do RUP, majoritariamente na elaboração. Alguma análise de requisitos ocorre na fase de construção, mas é muito reduzida se comparada com a fase de elaboração. A descrição de todos os casos de uso é feita na elaboração, bem como a sequente modelagem destes casos de uso no diagrama correspondente.

B – Alternativa incorreta.

JUSTIFICATIVA. A análise de requisitos ocorre principalmente na fase de elaboração. Nessa fase, são conduzidas entrevistas com os usuários para o esclarecimento e o detalhamento dos casos de uso. A definição do escopo do projeto é feita na fase de iniciação.

C – Alternativa correta.

JUSTIFICATIVA. A maior parte da análise de requisitos é realmente feita na fase de elaboração, com algumas alterações e complementos realizados em fases posteriores.

D – Alternativa incorreta.

JUSTIFICATIVA. A análise de requisitos é um processo iterativo e evolutivo, ocorrendo em todas as fases, majoritariamente na elaboração. Alguns requisitos podem ser revistos na fase de construção, mas eles não ocorrem nesta fase em paralelo com a programação.

E – Alternativa incorreta.

JUSTIFICATIVA. A análise de requisitos ocorre antes da programação e de testes do sistema. No entanto, eles podem sofrer alterações em fases posteriores, pois requisitos sofrem modificações tanto por parte dos usuários (modificações do negócio) quanto externamente (alterações de leis e de regulamentos).

3. Indicações bibliográficas

- MARTINS, J. C. C. *Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML*. 5. ed. São Paulo: Brasport, 2011.
- KRUCHTEN, P. *Introdução ao RUP*. São Paulo: Ciência Moderna, 2003.

ÍNDICE REMISSIVO

Questão 1	Sistema informatizado. Sistema gerenciador de banco de dados. Regras de corretude e integridade. Atomicidade.
Questão 2	Sistema ERP. Manutenções adaptativa, corretiva e perfectiva.
Questão 3	Redes locais. Topologia de redes. <i>Hub. Switch</i> . Volume de colisões na LAN.
Questão 4	Sistemas de gestão. Diagrama de casos de uso.
Questão 5	Sistemas de gestão. Diagrama de classes. Associações, atributos e operações.
Questão 6	Algoritmos. Execução.
Questão 7	Algoritmos. Pseudocódigo. Comandos.
Questão 8	Análise de requisitos. RUP (<i>Rational Unified Process</i>).