



ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 5

CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 5

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.

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

Leia o texto a seguir.

Várias técnicas relacionadas à programação extrema (XP) são diretamente ligadas ao código, incluindo a refatoração, programação em pares e integração contínua. A programação em pares é a prática preferida dos desenvolvedores XP trabalhando em pares em um computador.

SCOTT, K. *O Processo Unificado Explicado*. Porto Alegre: Artmed, 2002. BECK, K. *Programação Extrema (XP) Explicada*. Porto Alegre: Artmed, 2000.

A programação em pares auxilia no desenvolvimento de código de melhor qualidade quando os pares

- A. elaboram e utilizam padrões de codificação conjuntamente, os quais, quando utilizados corretamente e apropriadamente, reduzem problemas individuais.
- B. estão acostumados ao desenvolvimento e à propriedade coletiva, limitando-se a fazer pequenas mudanças na ocorrência de erros em tempo de execução.
- C. minimizam os riscos de insucesso no projeto através da utilização de ferramentas para a geração automática de testes funcionais e protótipos de interface.
- D. escrevem testes em separado e discutem os resultados posteriormente, o que dá a eles a chance de se sintonizarem antes de começarem a implementação.
- E. trabalham em projetos complexos onde a codificação seja desenvolvida de forma conjunta, minimizando erros e agregando valor onde quer que o sistema necessite.

1. Introdução teórica

Programação extrema (XP)

Programação extrema (XP) é um processo de desenvolvimento de software que tem, como princípios, a comunicação (com uso mínimo de documentação e uso máximo de interação entre as pessoas envolvidas no projeto), a simplicidade (o objetivo é buscar as soluções mais simples para resolver os problemas, contribuindo também para minimizar os custos com as manutenções futuras), o feedback (contribui para que todas as pessoas envolvidas no projeto aprendam cada vez mais sobre o sistema) e a coragem (é necessária para aplicar o processo corretamente).

Todo código criado em XP é escrito por um par de programadores: um coloca seu foco na codificação, pensando no algoritmo e na lógica de programação; e o outro observa o

¹Questão 10 - Enade 2011.

código e tenta encontrar uma forma estratégica de simplificá-lo, apontando erros decorrentes das possíveis falhas.

2. Análise da questão

O trabalho de elaboração de código e o trabalho da determinação dos padrões de codificação são feitos em conjunto, o que contribui para reduzir problemas individuais.

Alternativa correta: A.

3. Indicações bibliográficas

- PRESSMAN, R. S; MAXIM, B. R. *Engenharia de software*. 9. ed. Porto Alegre: AMGH, 2021.
- SOMMERVILLE, I. *Engenharia de software*. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

Questão 2

Questão 2.²

Leia o texto a seguir.

Os mapas de navegação enfocam como as pessoas se movimentam por um site ou aplicação. Cada página do site ou local da aplicação é representado com uma caixa ou cabeçalho e todas as páginas que puderem ser acessadas a partir dela devem usá-la como referência. Uma técnica bastante utilizada é colocar todos os fluxos possíveis no mapa de navegação, já que isso destacará seções onde há necessidade de uma revisão no projeto da interface.

BENYON, D. *Interação humano-computador*. 2. ed. São Paulo: Pearson Prentice Hall, 2011.

Com relação ao tema, avalie as asserções e a relação proposta entre elas.

I. Os mapas de navegação são redesenhados muitas vezes no decorrer do ciclo de vida do projeto

PORQUE

II. A interface deve ser projetada para atender pessoas (capacidades e limitações motoras, neurológicas, cognitivas etc.), atividades que as pessoas querem realizar (questões temporais, com ou sem cooperação, se são críticas em termos de segurança etc.), contextos nos quais a interação acontece (ambiente físico, contexto social ou organizacional etc.), e ater-se às tecnologias empregadas (*hardware*, plataformas, normas, linguagens de programação, políticas de acesso em ambientes de trabalho e residencial etc.). As combinações desses elementos são muito diferentes, por exemplo, em um quiosque público, em um sistema de agenda compartilhada, em uma cabine de avião ou em um telefone celular.

Acerca dessas asserções, assinale a opção correta.

- A. As duas asserções são proposições verdadeiras, e a segunda justifica a primeira.
- B. As duas asserções são proposições verdadeiras, e a segunda não justifica a primeira.
- C. A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- D. A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- E. As duas asserções são proposições falsas.

²Questão 15 - Enade 2011.

1. Introdução teórica

Mapa de navegação

O mapa de navegação é uma ferramenta gráfica que representa as interfaces gráficas dos sites ou das aplicações, os produtos e serviços, o fluxo da navegação da interface e as transações remotas e locais acessados. Ele deve ser orientado pelo fluxo do processo de negócio.

Existem vários modelos de ciclo de vida do projeto, que incluem a coleta de dados, a análise de informações, o projeto inicial, a implementação, os testes, a validação, a implantação e a manutenção.

A interface para o usuário é muito importante na área de aplicação computacional, pois ela envolve toda a parte visível ao usuário. Utilizando a interface, os usuários têm acesso às funções da aplicação. Fatores como satisfação, eficiência, segurança, custo de treinamento e de retorno de investimento dependem de um bom design de interface.

2. Análise das asserções

As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira, pois podemos redesenhar os mapas de navegação muitas vezes ao longo do ciclo de vida do projeto. Mas, para corrigirmos as interfaces gráficas dos sites ou das aplicações, os produtos e serviços, o fluxo da navegação da interface, as transações remotas e os locais acessados, o mapa de navegação deve ser orientado pelo fluxo do processo de negócio.

Alternativa correta: B.

3. Indicações bibliográficas

- PREECE, J.; ROGERS, Y.; SHARP, H. *Design de interação: além da interação homem-computador*. 3. ed. Porto Alegre: Bookman, 2013.
- PRESSMAN, R. S; MAXIM, B. R. *Engenharia de software*. 9. ed. Porto Alegre: AMGH, 2021.

Questão 3

Questão 3.³

Eclipse, *Netbeans*, *Jdeveloper* são exemplos de ambientes integrados de desenvolvimento, chamados de IDE, que têm por objetivo facilitar o desenvolvimento de *softwares*, provendo maior produtividade e gerenciamento de projetos. A especificação *JavaBeans* foi criada para ser um padrão de desenvolvimento de componentes que possam ser facilmente usados por outros desenvolvedores em diferentes IDE.

Com relação ao tema, avalie as asserções e a relação proposta entre elas.

I. Seja para o *Netbeans* ou para o *Eclipse*, é possível adquirir componentes de terceiros que facilitem a implementação do seu projeto.

PORQUE

II. Como o código desses componentes está em linguagem intermediária, ou seja, independente da arquitetura de um computador real, só é necessário que a máquina virtual esteja instalada no computador onde o aplicativo será executado e a máquina virtual será a responsável pela interpretação do código para a linguagem de máquina do computador em execução.

Acerca dessas asserções, assinale a alternativa correta.

- A. As duas asserções são proposições verdadeiras, e a segunda justifica a primeira.
- B. As duas asserções são proposições verdadeiras, e a segunda não justifica a primeira.
- C. A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- D. A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- E. As duas asserções são proposições falsas.

1. Introdução teórica

JavaBeans

O modelo de componentes JavaBeans foi desenvolvido para reutilizar componentes em outros componentes JavaBeans e applets, podendo rodar em um web browser usando uma Java Virtual Machine (JVM).

A JVM é um programa que executa e carrega os aplicativos Java, converte o bytecodes (formato intermediário de código projetado para permitir que vários hardwares e

³Questão 16 - Enade 2011.

softwares executem o mesmo código) em códigos de máquina executável e gerencia os aplicativos conforme são executados.

Dessa forma, os programas em Java podem funcionar em qualquer plataforma de software ou hardware que tenha uma JVM instalada.

2. Análise das asserções

As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira porque é possível adquirir componentes de terceiros que facilitem a implementação de um projeto, com ganho de tempo e custo mais baixo do que ser desenvolvido internamente.

Alternativa correta: B

3. Indicações bibliográficas

- ARNOLD, K.; GOSLING, J.; HOMES, D. *A linguagem de programação Java*. 4. ed. Porto Alegre: Bookman, 2007.
- HORSTMANN, C. *Padrões e projeto orientados a objetos*. 2. ed. Porto Alegre: Bookman, 2007.
- MENDES, D. R. *Programação Java com ênfase em orientação a objeto*. Rio de Janeiro: Novatec, 2009.

Questão 4

Questão 4.⁴

Leia o texto a seguir.

Os processos de negócio aumentam o entendimento de "o que precisa ser feito" e de "como precisa ser feito" dentro de uma organização.

BALDAM, R. 2009, 2. ed.

Sabendo disso, a fase da gestão de processos de negócio que evita a estagnação dos processos organizacionais é

- A. a fase de gerenciamento.
- B. a fase de planejamento.
- C. a fase de monitoração.
- D. a fase de modelagem.
- E. a fase de otimização.

1. Introdução teórica

Processo de negócio

O processo de negócio é um conjunto de ações e de atividades que são executadas para atingir determinado objetivo. Os processos são caracterizados pelas suas entradas, pelas técnicas que podem ser aplicadas e pelos resultados obtidos.

A gestão de processos de negócio (Business Process Management ou BPM) é um conceito que une gestão de negócios e tecnologia da informação, com a aplicação de ferramentas, técnicas e conhecimentos adequados às atividades do projeto, com foco na otimização dos resultados das organizações por meio da melhoria dos processos de negócio.

As fases da gestão de processos de negócio são as descritas a seguir.

- **Fase de planejamento.** Nessa fase, definem-se os processos mais importantes para a organização e verificam-se as possíveis falhas podem causar prejuízos à empresa. Estabelece-se um plano de ação para a implantação e a priorização dos processos que necessitam de ação imediata.
- **Fase de modelagem.** Nessa fase, executam-se atividades, como a compreensão e a documentação de processos e a coleta de dados relativos ao custo, aos recursos e ao

⁴Questão 17 - Enade 2011.

tempo, com o objetivo de obterem-se informações sobre o processo atual e sobre o modelo de processo futuro.

- **Fase de otimização.** Nessa fase, aplicam-se otimizações nos processos da organização. Nesse momento, podem ser removidos gargalos e eliminados custos não necessários ao seu negócio. Trata-se da fase em que se evita a estagnação dos processos organizacionais.
- **Fase de execução de processos.** Essa fase engloba as atividades de implementação e de execução dos processos, como a implantação dos planos de transferência de tecnologia e de treinamentos.
- **Fase de controle e de análise de dados.** O ideal é que o controle seja feito durante todas as fases. É importante o uso de indicadores, gerando informações que posteriormente realimentarão as atividades de otimização e planejamento.

2. Análise da questão

A gestão de processos estimula a otimização e, conseqüentemente, melhorias contínuas dos processos organizacionais. Utilizando técnicas gerenciais atuais, a gestão de processos faz com que essa abordagem se torne rotineira para muitas organizações, evitando a estagnação dos processos.

Alternativa correta: E.

3. Indicações bibliográficas

- BALDAM, R.; VALLE, R.; PEREIRA, H.; HILST, S.; ABREU, M.; SOBRAL, V.; *Gerenciamento de processo de negócio: BPM – Business Process Management*. 2. ed. São Paulo, Érica, 2009.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. 9. ed. Porto Alegre: AMGH, 2021.

Questão 5

Questão 5.⁵

Leia o texto a seguir.

Padrões de criação (creational patterns) abstraem a forma como objetos são criados, tornando o sistema independente de como os objetos são criados, compostos e representados. Um padrão de criação de classe usa a herança para variar a classe que é instanciada, enquanto que um padrão de criação de objeto delegará a instanciação para outro objeto. Há dois temas recorrentes nesses padrões. Primeiro, todos encapsulam conhecimento sobre quais classes concretas são usadas pelo sistema. Segundo, ocultam o modo como essas classes são criadas e montadas. Tudo que o sistema sabe no geral sobre os objetos é que suas classes são definidas por classes abstratas. Os padrões de criação são classificados em Abstract Factory, Builder, Factory Method, Prototype e Singleton.

GAMMA, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1994 (com adaptações).

O padrão *Abstract Factory* é usado quando

- A. o sistema deve ser independente da maneira como seus produtos são criados, relacionados e representados.
- B. o algoritmo de criação de um objeto deve ser independente das suas partes e da maneira como ele é montado.
- C. houver uma única instância de uma classe e esta for acessada a partir de um ponto de acesso conhecido.
- D. classes delegam responsabilidade a alguma das subclasses e deseja-se localizar qual é a subclasse acessada.
- E. as classes utilizadas para instanciação são especificadas em tempo de execução ou carregadas dinamicamente.

1. Introdução teórica

Design Pattern e Abstract Factory

Os Design Patterns (Padrões de Projetos) são uma coleção de padrões de projetos de software que reúnem soluções para problemas recorrentes e conhecidos, descrevendo uma solução legítima para os problemas comuns.

O padrão de criação Abstract Factory encapsula a criação de objetos. Esse padrão é muito útil para a construção de objetos individuais com propósito específico, sem a necessidade de serem conhecidas as classes específicas instanciadas.

⁵Questão 20 – Enade 2011.

Um exemplo de utilização do Abstract Factory é o caso de um sistema que precisa de suporte para múltiplos tipos de interfaces gráficas (Windows, Mac etc). Informa-se à factory que o sistema deve se parecer com o Windows. Após esse suporte, o sistema retorna à fábrica GUI (Graphical User Interface) dos objetos relativos ao Windows. Quando utilizamos um objeto (botão, check-box e janelas), a fábrica de GUI retorna às instâncias desses componentes para o Windows.

2. Análise da questão

O padrão Abstract Factory é utilizado quando o sistema opera independentemente da maneira como seus produtos são criados, relacionados e representados.

Alternativa correta: A.

3. Indicações bibliográficas

- ARNOLD, K.; GOSLING, J.; HOMES, D. *A linguagem de programação Java*. 4. ed. Porto Alegre: Bookman, 2007.
- HORSTMANN, C. *Padrões e projeto orientados a objetos*. 2. ed. Porto Alegre: Bookman, 2007.
- MENDES, D. R. *Programação Java com ênfase em orientação a objeto*. Rio de Janeiro: Novatec, 2009.

Questão 6

Questão 6.⁶

O paradigma de programação orientado a objetos tem sido largamente utilizado no desenvolvimento de sistemas.

Considerando o conceito de herança, avalie as afirmativas.

- I. Herança é uma propriedade que facilita a implementação de reuso.
- II. Quando uma subclasse é criada, ela herda todas as características da superclasse, não podendo possuir propriedades e métodos próprios.
- III. Herança múltipla é uma propriedade na qual uma superclasse possui diversas subclasses.
- IV. Extensão é uma das formas de se implementar herança.

É correto apenas o que se afirma em

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

1. Introdução teórica

Paradigma de orientação a objetos

O paradigma de orientação a objetos é uma forma de analisar, projetar e programar um sistema de software com base na interação e na composição de várias unidades de software, chamadas de objetos. Um objeto apresenta atributos (características) e pode executar tarefas reagindo (por meio dos métodos) a mensagens enviadas a ele e enviar mensagens a outros objetos.

O conceito de herança afirma que uma classe pode herdar operações de uma superclasse e as suas operações podem ser herdadas de uma ou de mais subclasses. A herança permite definir uma nova classe com base em uma classe já existente, com modificações e/ou extensões de comportamento, pois isso facilita a implementação de reuso.

⁶Questão 21 – Enade 2011.

Na figura 1, podemos observar que a Classe A é superclasse da Classe B e a Classe B é subclasse da Classe A.

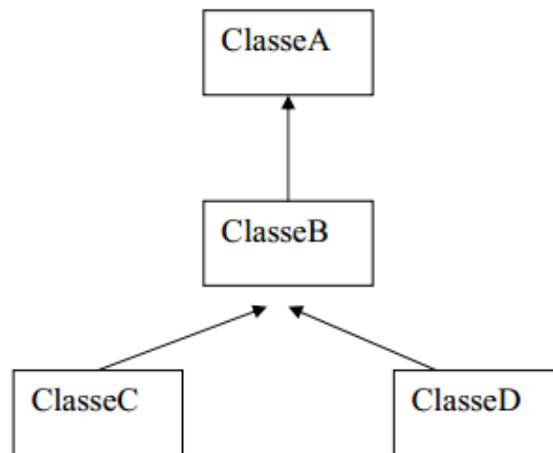


Figura 1. Herança de superclasse e subclasse.

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. Herança é uma propriedade que facilita a implementação de reuso, pois podemos utilizar uma classe inicial existente e criar várias outras classes a partir dela.

II – Afirmativa incorreta.

JUSTIFICATIVA. Quando uma subclasse é criada, ela herda todas as características da superclasse. A subclasse pode apresentar propriedades e métodos próprios.

III – Afirmativa incorreta.

JUSTIFICATIVA. Herança múltipla é a propriedade em que uma subclasse tem diversas superclasses, isto é, sempre que uma subclasse apresentar duas ou mais superclasses imediatas, ela é "filha" de mais de uma classe. Essa propriedade possibilita a combinação das características de diversas superclasses.

IV – Afirmativa correta.

JUSTIFICATIVA. A herança torna a extensão ou especialização de classe existente. Extensão é uma das formas de se implementar herança.

Alternativa correta: C.

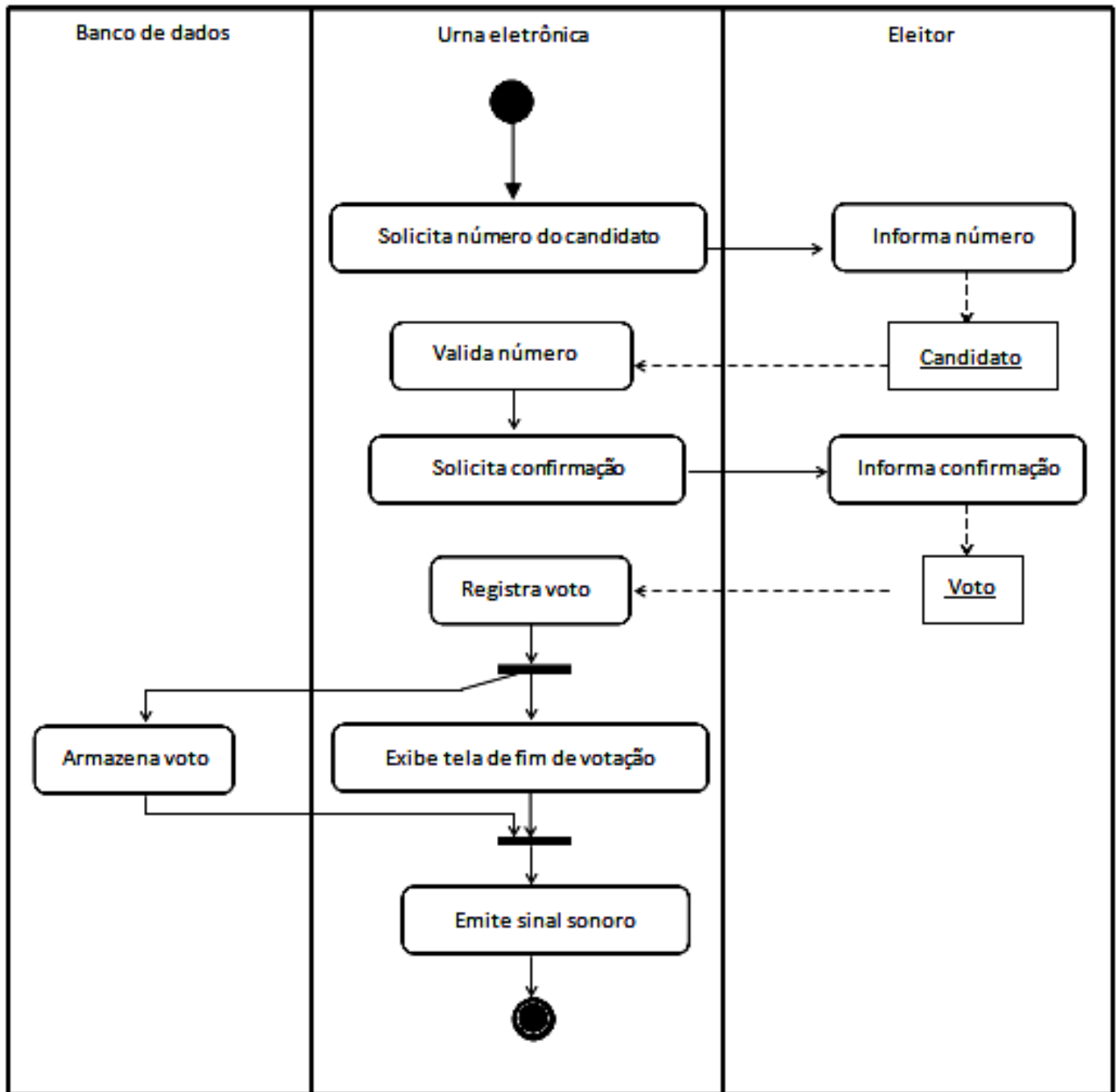
3. Indicações bibliográficas

- MARQUES, P.; PEDROSO, H. *C# 2.0*. Rio de Janeiro: LTC, 2007.
- SHARP, J. *Microsoft visual C# 2008 passo a passo*. Porto Alegre: Bookman, 2008.

Questão 7

Questão 7.⁷

Leia a figura a seguir.



A figura acima mostra um diagrama de

- A. comunicação, pois modela o comportamento do sistema e ilustra as interações entre atores e objetos.
- B. estados, pois apresenta os possíveis estados do objeto Urna eletrônica, além dos eventos que dão início à transição de um estado para outro.

⁷Questão 24 – Enade 2008.

- C. estados, pois, a partir de um estado inicial, descreve a sequência de possíveis estados que todos os objetos podem assumir.
- D. atividades, pois as ações ilustram a forma como o ator Eleitor interage com os objetos em um caso de uso.
- E. atividades, pois modela o fluxo de controle de um processo composto por ações sequenciais e paralelas partindo de um estado inicial.

1. Introdução teórica

Diagramas UML

A questão refere-se à correta identificação dos diagramas UML (Unified Modeling Language). Como a UML é uma linguagem desenvolvida para ser extensível e de fácil uso, seus elementos constituintes são partilhados entre diferentes diagramas. Desse modo, a adequada identificação de alguns diagramas só se torna possível pela interpretação da informação contida internamente neles.

Analisemos, por exemplo, o conteúdo de um diagrama de **máquina de estados** e de **atividades**. Seus desenhos podem ser muito parecidos. Uma das principais coisas que os diferencia são os verbos utilizados para determinar suas atividades ou seus estados. Verbos no infinitivo (ou, num critério menos rigoroso, em outros tempos do indicativo) são aceitos como padrão para definir atividades. Verbos no gerúndio e particípio passado são usados para definir estados. Portanto, Registrar Voto ou Registra Voto designa uma atividade, enquanto Registrando Voto ou Voto Registrado designam estados.

Não podemos, contudo, confundir diagramas com simbologias radicalmente distintas. O diagrama de **comunicação**, por exemplo, contém elementos que tornam o seu desenho completamente diferente dos citados anteriormente. Neste, falamos da interação entre atores e objetos, não mais de interação entre atividades e estados. Embora possa ser trabalhoso identificar se um elemento é uma atividade ou um estado mediante um tempo verbal mal colocado, a semântica do diagrama de comunicação é completamente diferente. E, visualmente, são diagramas muito diversos, não se admitindo que sejam confundidos.

As mesmas considerações são feitas em relação ao diagrama de **casos de uso**, que mostra as interações entre atores e funcionalidades do sistema. Sua simbologia é característica, o que faz com que não seja facilmente confundido com outros diagramas UML. Portanto, os quatro diagramas UML considerados foram os indicados a seguir.

1. **Atividades e Estados.** Diagramas que podem ser confundidos um com o outro, caso não seja seguido um padrão rigoroso de nomenclatura dos elementos constituintes.
2. **Comunicação.** Mostra a interação entre atores e objetos. Visualmente, tem características particulares e inconfundíveis.
3. **Caso de Uso.** Mostra a interação entre atores e funcionalidades, com características visuais inconfundíveis.

3. Análise das alternativas

A – Alternativa incorreta.

JUSTIFICATIVA. O diagrama não é de comunicação porque está modelando atividades (processos) e não a interação entre atores e objetos. No diagrama, embora sejam apresentados alguns atores, por exemplo, o Eleitor, não apresenta a comunicação entre objetos.

B – Alternativa incorreta.

JUSTIFICATIVA. Não é um diagrama de estados porque está mostrando o fluxo do processo executado pelo objeto Urna Eletrônica, representado em forma de atividades (e não estados). Os estados para esse objeto seriam, por exemplo, Número do Candidato Informado, Validando Candidato, Solicitando Confirmação e Registrando Voto.

C – Alternativa incorreta.

JUSTIFICATIVA. É uma variação do erro da resposta anterior ampliada para os demais objetos. O diagrama não mostra os estados da Urna Eletrônica nem de qualquer outro objeto.

D – Alternativa incorreta.

JUSTIFICATIVA. É um diagrama de atividades, mas as interações do ator não são com casos de uso. O diagrama de atividades é procedural e, o de caso de uso, funcional. São visões distintas, representadas por diagramas diferentes e separados.

E – Alternativa correta.

JUSTIFICATIVA. O diagrama de atividades modela processos e seu fluxo de controle a partir de um estado inicial. As ações podem ser sequenciais e paralelas. O diagrama de estados

modela os estados que também podem ser sequenciais e paralelos. Portanto, é necessário muito cuidado para não confundir estes diagramas.

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 8

Questão 8.⁸

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

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

A partir das informações acima, é correto concluir que

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

⁸Questão 34 – Enade 2008.

1. Análise da questão

Bancos de dados relacionais

Na modelagem de bancos de dados relacionais, chamamos de **chave** um atributo ou um conjunto de atributos que permitem identificar de modo unívoco (ou seja, de forma única) cada registro de uma tabela.

A modelagem do enunciado apresenta um relacionamento “muitos-para-muitos”, entre *Produto* e *Fazenda*, porque vários produtos podem ser comprados de várias fazendas. Para resolver esse tipo de relacionamento, criamos a tabela associativa *Estoque*, que atua como uma tabela intermediária.

Ao fazermos isso, o “muitos-para-muitos” ($M * N$) é transformado em dois relacionamentos “1-para-muitos” ($1 * N$). Assim, teremos um relacionamento “1-para-muitos” entre *Produto* e *Estoque* e outro relacionamento “1-para-muitos” entre *Fazenda* e *Estoque*.

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



Figura 1. Resolução do relacionamento NxM.

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



Figura 2. Ligação entre campos.

A chave *DataCompra* é necessária na entidade *Estoque* porque, caso contrário, os produtos somente poderiam ser comprados em uma data específica (fosse ela qual fosse), não permitindo registrar as datas de dois dias de compra diferentes, por exemplo.

2. Análise das alternativas

A – Alternativa correta.

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

B – Alternativa incorreta.

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

C – Alternativa incorreta.

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

D – Alternativa incorreta.

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

E – Alternativa incorreta.

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

3. Indicações bibliográficas

- DATE, C. J. *Introdução a sistemas de bancos de dados*. 8. ed. São Paulo: Campus, 2004.
- ELMASRI, R.; NAVATHE, S. *Sistemas de bancos de dados*. São Paulo: Pearson, 2019.

- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de bancos de dados*. São Paulo: LTC, 2012.
- TEORY, T. *Projeto e modelagem de bancos de dados*. São Paulo: Campus, 2013.

ÍNDICE REMISSIVO

Questão 1	Programação extrema (XP).
Questão 2	Mapa de navegação.
Questão 3	JavaBeans.
Questão 4	Processo de negócio.
Questão 5	Design Pattern e Abstract Factory.
Questão 6	Paradigma de orientação a objetos.
Questão 7	Diagramas UML. Diagramas de comunicação, de estados, de atividades e de casos de uso.
Questão 8	Bancos de dados relacionais.