



ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 4

CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 4

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

O levantamento de requisitos é uma etapa fundamental do projeto de sistemas. Dependendo da situação encontrada, uma ou mais técnicas podem ser utilizadas para a elicitación dos requisitos.

A respeito dessas técnicas, avalie as afirmativas.

- I. Workshop de requisitos consiste na realização de reuniões estruturadas e delimitadas entre os analistas de requisitos do projeto e representantes do cliente.
- II. Cenário consiste na observação das ações do funcionário na realização de uma determinada tarefa, para verificar os passos necessários para sua conclusão.
- III. As entrevistas são realizadas com os *stakeholders* e podem ser abertas ou fechadas.
- IV. A prototipagem é uma versão inicial do sistema, baseado em requisitos levantados em outros sistemas da organização.

É correto apenas o que se afirma em

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

1. Introdução teórica

Elicitación de requisitos

Um processo de desenvolvimento de software classifica em atividades as tarefas realizadas durante a construção de um sistema.

Uma dessas atividades, a de levantamento de requisitos, é fundamental para um bem-sucedido desenvolvimento de software. Seu principal objetivo é fazer com que os stakeholders (público estratégico, desde os usuários finais até os gerentes do sistema) e os desenvolvedores tenham a mesma perspectiva dos problemas a serem resolvidos.

Um requisito de sistema é a condição ou a capacidade que cumpre um contrato padrão específico ou outros documentos formalmente impostos. Os requisitos são

¹Questão 11 – Enade 2011.

normalmente identificados a partir de um domínio, ou seja, da área de conhecimento que corresponde à parte do mundo real que é relevante.

O produto do levantamento de requisitos é o documento de requisitos e suas principais seções são:

- requisitos funcionais, que definem as funcionalidades do sistema;
- requisitos não funcionais, que definem as características de qualidade do sistema.

As técnicas apresentam vantagens e desvantagens e podem ser utilizadas individualmente ou em conjunto. Descrevemos, a seguir, as principais técnicas de elicitação dos requisitos.

- **Etnografia.** É uma técnica de observação utilizada para entender os processos operacionais. O analista acompanha o ambiente de trabalho em que o sistema será usado. O trabalho diário é assistido e são feitas diversas anotações sobre as tarefas reais. Ela ajuda a refletir a realidade na qual as pessoas trabalham, em vez de refletir processos formais definidos pela organização.
- **Workshop.** É o encontro de uma equipe de analista de requisitos e uma equipe de stakeholders em uma reunião estruturada. O workshop tem o objetivo de promover o trabalho em equipe e de criar um documento de requisitos bem definido.
- **Prototipagem.** O protótipo é recomendado para avaliar a interface do usuário, os problemas de comunicação com outros produtos e a possibilidade de atendimento aos requisitos de desempenho.
- **Entrevistas.** As entrevistas podem ser formais ou informais. A equipe de engenharia de requisitos questiona os stakeholders sobre o sistema atual e as perspectivas do sistema que será desenvolvido. Os requisitos são coletados a partir das respostas a essas perguntas. As entrevistas são divididas em dois tipos: entrevistas fechadas (quando houver um conjunto pré-definido de perguntas) e entrevistas abertas (a equipe de engenharia desenvolve uma série de questões no momento da entrevista e, assim, compreende melhor as necessidades do sistema).
- **Brainstorming.** São selecionados tópicos específicos que estarão em pauta no brainstorming. Os participantes são escolhidos conforme a intimidade com o assunto e cada um elabora ideias sobre o tema proposto. Uma pessoa é selecionada para registrar todas as ideias e, ao final da reunião, é feita uma análise para selecionar as considerações mais relevantes.

As pessoas preferem relacionar-se com exemplos da vida real do que com descrições abstratas. Nesse sentido, o cenário é a descrição da utilização de uma funcionalidade do

sistema. Os cenários podem ser particularmente úteis para acrescentar detalhes na descrição dos requisitos.

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. No workshop de requisitos, é realizada uma reunião estruturada entre os analistas de requisitos e os representantes dos clientes (stakeholders).

II – Afirmativa incorreta.

JUSTIFICATIVA. Um cenário é a descrição de um fato ocorrido na utilização de uma funcionalidade do sistema.

III – Afirmativa correta.

JUSTIFICATIVA. As entrevistas podem ser abertas (não há perguntas definidas) ou fechadas (um questionário pré-definido deve ser respondido na entrevista).

IV – Afirmativa incorreta.

JUSTIFICATIVA. A prototipagem é baseada nas necessidades do cliente, apresentadas em documentos e análises.

Alternativa correta: B.

3. Indicações bibliográficas

- BEZZERRA, E. *Princípios de análise e projeto de sistemas com UML*. 3. ed. Rio de Janeiro: LTC, 2014.
- 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

O conjunto de casos de uso representa as possíveis interações que serão representadas nos requisitos do sistema. A figura a seguir desenvolve um exemplo de biblioteca e mostra outros casos de uso (use-cases) nesse ambiente.

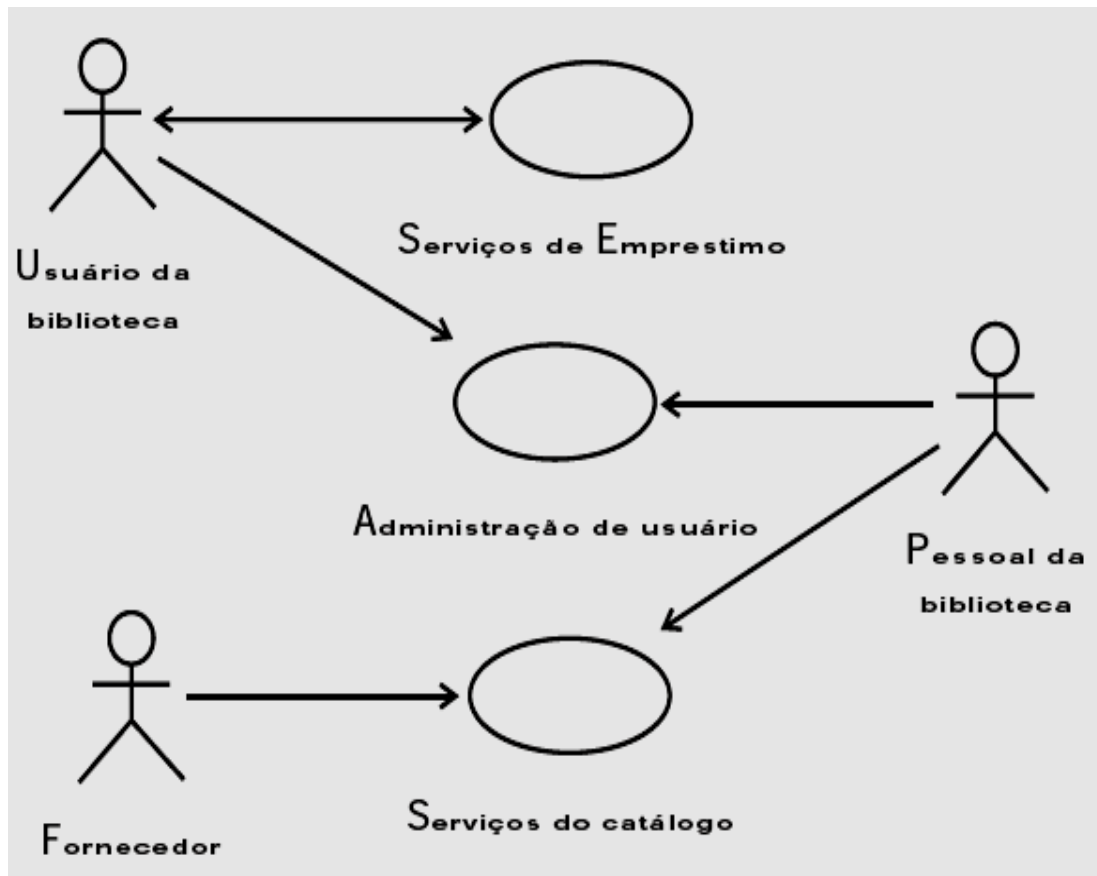


Figura 1. SOMMERVILLE, I. Engenharia de software. 6. ed. São Paulo. MAKRON BOOKS (2003).

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

I. A figura também ilustra os pontos essenciais da notação de casos de uso. Os agentes no processo são representados por bonecos e cada tipo de interação é representada por uma elipse com um nome.

PORQUE

II. A UML é um padrão para a modelagem orientada a objetos e, assim, os casos de uso e a obtenção de requisitos com base em casos de uso são comumente utilizados para obter requisitos.

²Questão 11 - Enade 2011 (com adaptações).

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.

1. Introdução teórica

Linguagem de Modelagem Unificada (Unified Modeling Language - UML)

A UML define uma notação padrão que pode ser utilizada por desenvolvedores de software orientado a objetos, isto é, um padrão para a modelagem orientada a objetos.

Devido à sua descrição em linguagem natural e à sua notação gráfica, que facilita a relação entre os especialistas do domínio e a equipe técnica, a modelagem de casos de uso é utilizada para realizar a documentação de requisitos funcionais de um sistema.

O diagrama de casos de uso é uma ferramenta disponível na UML para a modelagem de aspectos dinâmicos de sistemas. Um caso de uso representa uma descrição de uso de uma determinada funcionalidade do sistema. Esse diagrama é composto pelos itens a seguir.

- **Cenário:** é a descrição de um acontecimento na prática de alguma funcionalidade do sistema.
- **Casos de uso:** representam uma interação entre um elemento externo (por exemplo, usuários, organizações, outros sistemas de software e equipamentos com os quais o sistema deve se comunicar) e o sistema. A representação de um caso de uso é feita por uma elipse com o nome abaixo da figura.
- **Atores:** quaisquer elementos externos que trocam informação com o sistema. A notação utilizada para representar um ator é a figura de um boneco com o nome abaixo da figura.
- **Relacionamento de comunicação:** representa a maneira como os atores e os casos de uso de um diagrama relacionam-se.

2. Análise das asserções

I – Asserção verdadeira.

JUSTIFICATIVA. A figura apresenta um diagrama de caso de uso com as principais representações: os agentes (atores), correspondidos pelos bonecos, e as interações (casos de uso), constituídas pelas elipses.

II – Asserção verdadeira.

JUSTIFICATIVA. Por facilitar a comunicação entre os especialistas do domínio e a equipe técnica, a obtenção de requisitos com base em casos de uso é muito utilizada.

A primeira asserção não justifica a segunda, pois a primeira refere-se às notações do diagrama de caso de uso e a segunda, à utilização do diagrama para obtenção dos requisitos.

Alternativa correta: B.

3. Indicações bibliográficas

- BEZZERRA, E. *Princípios de análise e projeto de sistemas com UML*. 3. ed. Rio de Janeiro: LTC, 2014.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. 2. ed. Rio de Janeiro: Elsevier, 2012

Questão 3

Questão 3.³

Avalie as afirmativas sobre a UML (Linguagem de Modelagem Unificada).

- I. A UML é uma metodologia para o desenvolvimento de software orientado a objetos, uma vez que fornece um conjunto de representações gráficas e sua semântica para a modelagem de software.
- II. O diagrama de casos de uso procura, por meio de uma linguagem simples, demonstrar o comportamento externo do sistema. Esse diagrama apresenta o sistema sob a perspectiva do usuário, e é dentre todos da UML, o mais abstrato, flexível e informal.
- III. Um relacionamento de extensão de um caso de uso "A" para um caso de uso "B" significa que toda vez que "A" for executado ele incorporará o comportamento definido em "B".
- IV. Os diagramas de comportamento da UML demonstram como ocorrem as trocas de mensagens entre os objetos do sistema para se atingir um determinado objetivo.

É correto apenas o que se afirma em

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

1. Introdução teórica

Linguagem de Modelagem Unificada (Unified Modeling Language - UML)

A UML é uma linguagem de modelagem visual, projetada para ser simples e eficiente. Ela auxilia na visualização do sistema e na comunicação entre os objetos. A UML não afirma que existe uma receita para projetar um sistema, ela apenas apresenta um conjunto de representações gráficas e um estudo das relações entre seus signos e seus referentes para a modelagem do sistema.

Uma das representações gráficas da UML é o diagrama de casos de uso. Esse diagrama auxilia na compreensão das funcionalidades externas e nos elementos externos que interagem com o sistema.

³Questão 12 – Enade 2011.

Os diagramas de casos de uso são compostos pelos atores (usuários e sistemas que utilizam alguma funcionalidade do sistema), pelos casos de uso (funcionalidades necessárias do sistema) e pelos relacionamentos existentes entre esses componentes.

Um ator deve relacionar-se com um ou mais casos de uso do sistema. Também pode existir relacionamento entre atores ou entre casos de uso. Para o relacionamento, a UML define quatro modelos: comunicação, inclusão, extensão e generalização.

A comunicação acontece entre atores e casos de uso. Esse relacionamento indica que o ator troca informações com o sistema utilizando o caso de uso associado.

A inclusão acontece entre casos de uso. Quando dois ou mais casos de uso apresentam uma rotina comum, podemos representar essa rotina em outro caso de uso. O relacionamento de inclusão obrigatoriamente faz com que a execução do primeiro caso de uso obrigue a execução do segundo.

A extensão acontece entre casos de uso. Ao contrário do relacionamento de inclusão, esse relacionamento não apresenta obrigatoriedade, isto é, ao executarmos um caso de uso, sua extensão será executada somente se determinada condição for satisfeita.

A generalização pode acontecer entre dois casos de uso ou entre dois atores. Esse relacionamento apresenta uma herança de características entre os elementos.

Os diagramas da UML são divididos em dois grupos: diagrama estrutural e diagrama comportamental. Para compor o grupo do diagrama estrutural, temos diagramas de classe, de componentes, de objetos, de estrutura composta, de implantação e de pacotes. Para compor o grupo do diagrama comportamental, temos diagramas de atividade, de casos de uso, de máquina de estado, de interação, de sequência, de comunicação, de visão, de interação e de temporização.

2. Análise das afirmativas

I – Afirmativa incorreta.

JUSTIFICATIVA. A UML não é uma metodologia de desenvolvimento de software, pois ela não apresenta um método de desenvolvimento, mas um conjunto de representações gráficas que auxilia no entendimento do sistema.

II – Afirmativa correta.

JUSTIFICATIVA. O diagrama de caso de uso, assim como todos os diagramas da UML, é projetado para ser simples e eficiente. Esse diagrama auxilia na compreensão das

funcionalidades externas e apresenta o sistema baseando-se na visão do usuário. Dessa forma, ele deve ser abstrato, flexível e informal.

III – Afirmativa incorreta.

JUSTIFICATIVA. O relacionamento de extensão não apresenta obrigatoriedade, por isso, ao executarmos um caso de uso A, sua extensão B será executada somente se determinada condição for satisfeita.

IV – Afirmativa correta.

JUSTIFICATIVA. Todos os diagramas que compõem o grupo de comportamento apresentam as trocas de mensagens entre os objetos do sistema para se atingir determinado objetivo.

Alternativa correta: B.

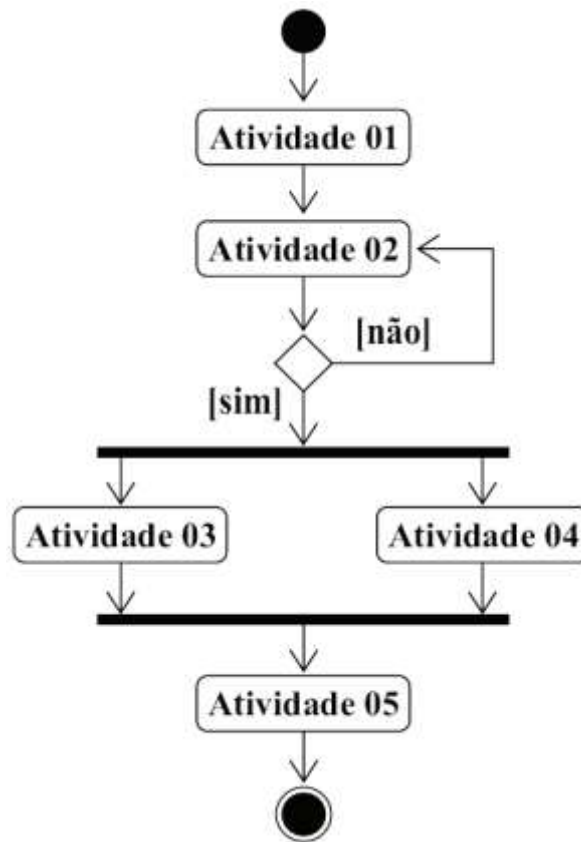
3. Indicações bibliográficas

- BEZZERRA, E. *Princípios de análise e projeto de sistemas com UML*. 3. ed. Rio de Janeiro: LTC, 2014.
- GUEDES, G. *UML 2: uma abordagem prática*. 2. ed. São Paulo: Novatec, 2011.

Questão 4

Questão 4.⁴

O diagrama de atividades é um dos diagramas disponíveis na UML (Linguagem de Modelagem Unificada) para a modelagem de aspectos dinâmicos de sistemas.



Com relação ao diagrama de atividades apresentado, avalie as afirmativas.

- I. A atividade 05 será executada se a atividade 03 ou a atividade 04 for concluída.
- II. A ramificação sequencial existente após a atividade 02 significa que, caso o fluxo seja [não], é necessário que sejam executadas novamente as atividades 01 e 02.
- III. As atividades 03 e 04 vão ter início ao mesmo tempo, entretanto, não significa que terminarão ao mesmo tempo.
- IV. Caso o fluxo da ramificação sequencial existente após a atividade 02 tenha o fluxo [sim], a atividade 02 não será mais executada.

É correto apenas o que afirma em

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

⁴Questão 14 – Enade 2011.

1. Introdução teórica

Diagrama de atividades

A partir da versão 2.0 da Unified Modeling Language (UML), o diagrama de atividade passou a ser visto como um diagrama independente. Ele é utilizado para modelar atividades.

Uma atividade é composta por uma coleção de ações, ou seja, pelos passos necessários para que a atividade seja realizada, conforme mostra a figura 1.

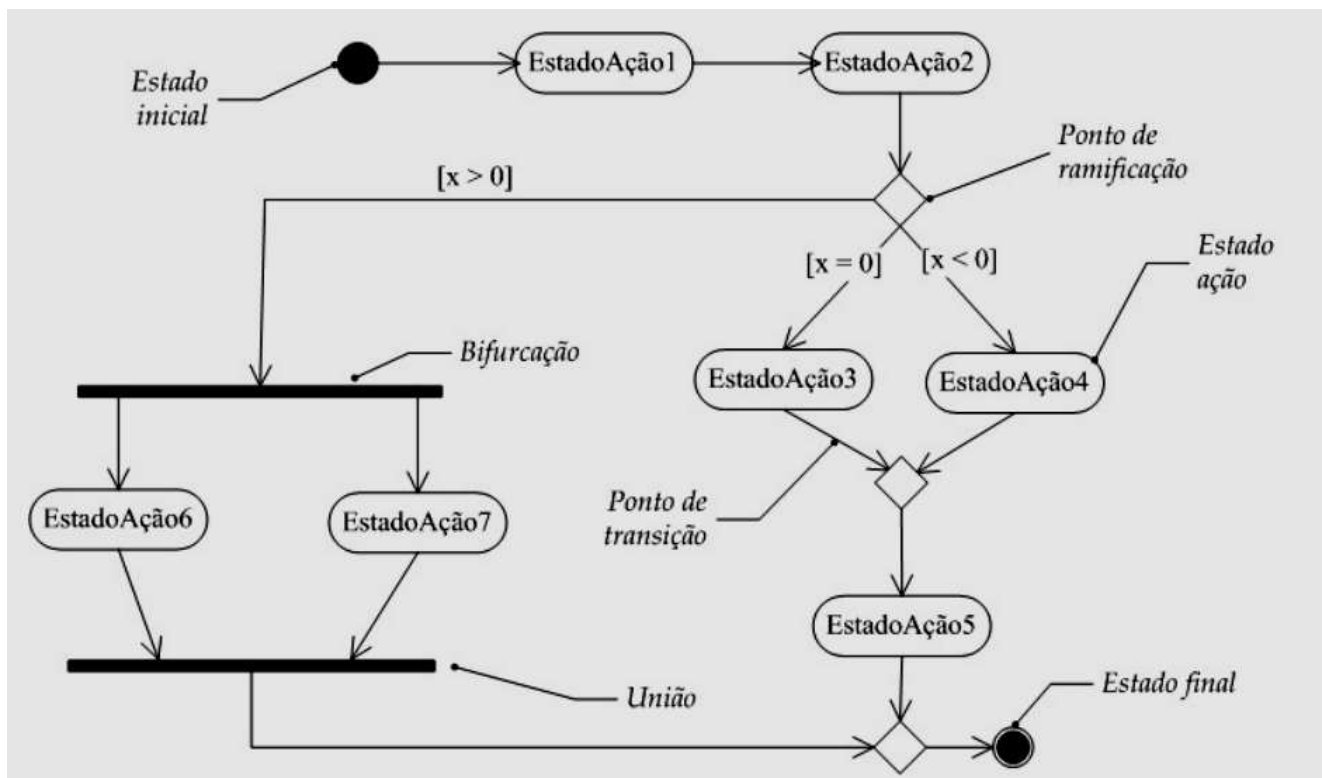


Figura 1. Elementos de um diagrama de atividades.
BEZERRA; EDUARDO (2007).

As coleções de ações podem ser divididas conforme mostrado a seguir.

- **Estado inicial:** é representado por um círculo preenchido. Todo diagrama de atividade deve ter um estado inicial.
- **Estado de ação:** é realizado instantaneamente e indica que o sistema está realizando algo.
- **Estado de atividade:** necessita de um tempo para ser executado e indica que o sistema está realizando algo.
- **Ponto de ramificação:** é representado por um losango e apresenta uma transição de entrada e várias transições de saída.

- **Ponto de união:** é representado por um losango e reúne várias transições de entrada e uma transição de saída.
- **Ponto de transição:** ocorre no término de um estado de ação e, consequentemente, no início de outro.
- **Bifurcação:** pode ser representada por uma barra horizontal. Recebe uma transição de entrada e cria dois ou mais fluxos de controle que serão executados independentemente e em paralelo.
- **União:** é representada por uma barra horizontal. Recebe duas ou mais transições de entrada e une os fluxos em um único fluxo de controle.
- **Estado final:** um diagrama de atividade pode não apresentar estado final (quando o processo ou procedimento que está sendo modelado é cíclico) e também pode ter um ou vários estados finais.

2. Análise das afirmativas

I – Afirmativa incorreta.

JUSTIFICATIVA. Para a atividade 05 ser executada, tanto a atividade 03 quanto a atividade 04 devem ser concluídas. A barra horizontal que representa a união das duas atividades conecta os fluxos em um único fluxo de controle.

II – Afirmativa incorreta.

JUSTIFICATIVA. Caso o fluxo seja [não], a análise do ponto de ramificação existente após a atividade 02 mostra a necessidade de que apenas ela seja executada novamente.

III – Afirmativa correta.

JUSTIFICATIVA. As atividades 03 e 04 estão entre uma barra de bifurcação e uma barra de união. A barra de bifurcação cria dois fluxos de controle que serão executados independentemente e em paralelo, o que não significa que terminarão ao mesmo tempo. A barra de união recebe as duas entradas e une os fluxos em um único fluxo de controle.

IV – Afirmativa correta.

JUSTIFICATIVA. Caso o fluxo da ramificação sequencial após a atividade 02 siga [sim], a atividade 02 não será mais executada, pois o fluxo segue para a barra de bifurcação.

Alternativa correta: C.

3. Indicações bibliográficas

- BEZZERRA, E. *Princípios de análise e projeto de sistemas com UML*. 3. ed. Rio de Janeiro: LTC, 2014.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do usuário*. 2. ed. Rio de Janeiro: Elsevier, 2012.

Questão 5

Questão 5.⁵

Com relação ao valor lógico, avalie as afirmativas.

- I. $\neg (p \wedge \neg q)$
- II. $p \rightarrow (q \rightarrow p)$
- III. $(p \vee \neg q) \rightarrow \neg p$
- IV. $(p \wedge q) \vee (\neg p \wedge \neg q)$

É tautologia apenas o que se afirma em

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

1. Introdução teórica

Proposição

Quando nos expressamos em português por meio da escrita, criamos sentenças utilizando sinais de pontuação, como interrogações e exclamações, e elementos coesivos, como conjunções.

Denominamos de proposição um conjunto de símbolos que expressa um pensamento de sentido completo, uma frase que pode ser apenas verdadeira ou falsa.

A seguir, são dados dois exemplos de sentenças, uma verdadeira e uma falsa.

- O número 4 é par (sentença verdadeira).
- O número 6 é ímpar (sentença falsa).

Os conectivos lógicos (não, e, ou etc.) são expressões que permitem a criação de novas proposições, conforme exemplificado a seguir.

- **Não** existe divisão por zero.
- O número 4 é par **e** o número 5 é ímpar.
- O número 7 é par **ou** ímpar.

Para analisarmos as proposições compostas, precisamos utilizar a tabela verdade, que verifica o valor lógico da proposição.

⁵Questão 19 – Enade 2011.

Existem quatro operações lógicas sobre proposições: **Negação** ($\neg$), **Conjunção** ($\wedge$), **Disjunção** ($\vee$) e **Condicional** ($\rightarrow$), conforme descrito a seguir.

Negação ($\neg$). A negação da proposição p é dada por $\neg p$. Sua tabela verdade está mostrada a seguir.

p	$\neg p$
V	F
F	V

V - Verdadeiro e F - Falso

Conjunção ($\wedge$). Considerando duas proposições p e q , a conjunção é dada por $p \wedge q$. Sua tabela verdade está mostrada a seguir.

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Disjunção ($\vee$). Considerando duas proposições p e q , a disjunção é dada por $p \vee q$. Sua tabela verdade está mostrada a seguir.

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Condicional ($\rightarrow$): Considerando duas proposições p e q , a condicional é dada por $p \rightarrow q$. Sua tabela verdade está mostrada a seguir.

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Sempre que a tabela verdade resultar em verdadeiro, independentemente dos valores das proposições simples que a compõem, é denominada **tautologia**. De maneira oposta, sempre que a tabela verdade resulta em falso, independentemente dos valores das proposições simples que a compõem, é denominada **contradição**. Se a proposição composta não é tautológica e não é uma contradição, é denominada **contingência**.

2. Análise das afirmativas

I – Afirmativa incorreta.

JUSTIFICATIVA. Vejamos a tabela verdade a seguir.

p	q	¬q	p ∧ ¬q	¬(p ∧ ¬q)
V	V	F	F	V
V	F	V	V	F
F	V	F	F	V
F	F	V	F	V

II – Afirmativa correta.

JUSTIFICATIVA. Vejamos a tabela verdade a seguir.

p	q	q → p	p → (q → p)
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

III – Afirmativa incorreta.

JUSTIFICATIVA. Vejamos a tabela verdade a seguir.

p	q	¬q	p ∨ ¬q	¬p	(p ∨ ¬q) → ¬p
V	V	F	V	F	F
V	F	V	V	F	F
F	V	F	F	V	V
F	F	V	V	V	V

Essa expressão lógica sempre resulta em verdadeiro, independentemente dos valores das proposições simples que a compõem. Concluimos, então, que essa expressão lógica é tautológica.

IV – Afirmativa incorreta.

JUSTIFICATIVA. Vejamos a tabela verdade a seguir.

p	q	$\neg p$	$\neg q$	$p \wedge q$	$\neg p \wedge \neg q$	$(p \wedge q) \vee (\neg p \wedge \neg q)$
V	V	F	F	V	F	V
V	F	F	V	F	F	F
F	V	V	F	F	F	F
F	F	V	V	F	V	V

A expressão lógica é uma contingência.

Alternativa correta: B.

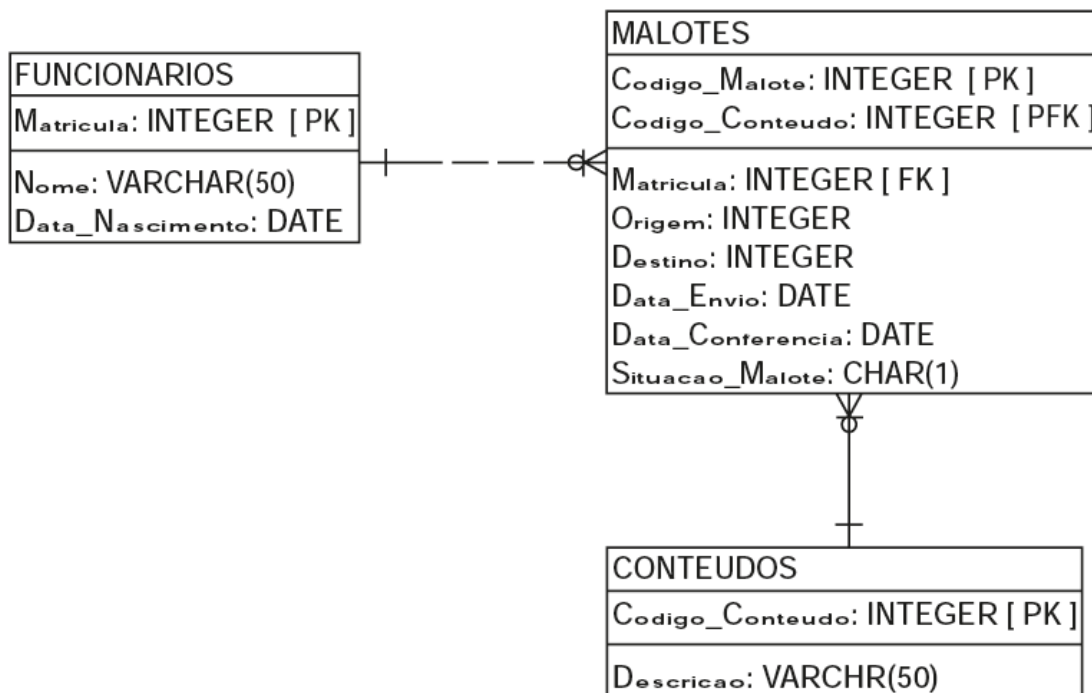
3. Indicações bibliográficas

- CARVALHO, F.; MINORO, J. *Tomadas de decisão com ferramentas de lógica paraconsistente anotada: método paraconsistente de decisão: MPD*. 5. ed. São Paulo: Blücher, 2011.
- ALENCAR FILHO, E. *Iniciação à lógica matemática*. São Paulo: Nobel, 2002.

Questão 6

Questão 6.⁶

Pedro foi contratado como desenvolvedor de *software* de uma empresa. Em seu primeiro dia de trabalho ele se deparou com o DER (Diagrama Entidade-Relacionamento), que representa os dados de um sistema de controle de malotes. Foi solicitado a Pedro relatório para o sistema contendo os seguintes dados: o nome de todos os funcionários que enviaram os malotes, o código dos malotes enviados, a descrição de seus conteúdos e a situação dos malotes. Para a geração do relatório, Pedro tem que fazer uma consulta utilizando o comando SELECT da linguagem SQL.



Conhecidos o modelo conceitual de dados e os dados necessários para a tarefa de Pedro, o comando SELECT que ele deve executar para realizar a consulta e produzir o relatório corretamente é

- A. `SELECT NOME,CODIGO_MALOTE,DESCRICAO,SITUACAO_MALOTE FROM MALOTES
INNER JOIN CONTEUDOS ON (CODIGO_CONTEUDO = CODIGO_CONTEUDO) INNER
JOIN FUNCIONARIOS ON (MATRICULA = MATRICULA);`
- B. `SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES,
CONTEUDOS, FUNCIONARIOS WHERE (CODIGO_CONTEUDO = CODIGO_CONTEUDO)
AND (MATRICULA = MATRICULA);`
- C. `SELECT NOME,CODIGO_MALOTE,DESCRICAO,SITUACAO_MALOTE FROM MALOTES
INNER JOIN CONTEUDOS INNER JOIN FUNCIONARIOS`

⁶Questão 22 - Enade 2011.

```
ON(MALOTES.CODIGO_CONTEUDO = CONTEUDOS.CODIGO_CONTEUDO)
ON(MALOTES.MATRICULA = FUNCIONARIOS.MATRICULA);
```

- D. SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES
 INNER JOIN CONTEUDOS ON (MALOTES.CODIGO_CONTEUDO =
 CONTEUDOS.CODIGO_CONTEUDO) INNER JOIN FUNCIONARIOS
 ON(MALOTES.MATRICULA = FUNCIONARIOS.MATRICULA);
- E. SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES,
 CONTEUDOS, FUNCIONARIOS INNER JOIN WHERE (MALOTES.CODIGO_CONTEUDO =
 CONTEUDOS.CODIGO_CONTEUDO) AND (MALOTES.MATRICULA =
 FUNCIONARIOS.MATRICULA);

1. Introdução teórica

Diagrama Entidade-Relacionamento (DER)

DER é uma modelagem semântica baseada no modelo de entidade/relacionamento. A primeira etapa para se desenvolver um DER é identificar um conjunto de conceitos semânticos que parecem ser úteis para descrever o mundo real. A seguir, temos exemplos desses conceitos.

- **Entidade:** é um objeto perceptível e que pode ser identificado distintamente. Exemplo: funcionários, malotes etc.
- **Propriedade:** todas as entidades apresentam certos tipos de propriedades/informações que as descrevem. Exemplo: matrícula do funcionário, código do malote etc.
- **Relacionamento:** é a associação entre as entidades, isto é, o relacionamento serve para interconectar duas ou mais entidades. Exemplo: malotes (funcionários-conteúdo).
- **Subtipo:** o tipo de entidade A é um subtipo do tipo de entidade B caso todo A seja necessariamente um B. Exemplo: funcionário é um subtipo de pessoa.

Podemos classificar o relacionamento entre entidade/relacionamento de três formas diferentes, conforme segue.

- **Um para um:** se uma entidade A está relacionada a uma única entidade B e vice-versa. Exemplo: um funcionário gerencia um único departamento e um departamento é gerenciado apenas por um funcionário.
- **Um para muitos:** se uma entidade A está relacionada com várias entidades B. Exemplo: o envio de vários malotes por um funcionário.

- **Muitos para muitos:** se várias entidades A estão relacionadas com várias entidades em B e vice-versa. Exemplo: um funcionário pode participar de vários projetos e um projeto pode ter vários funcionários.

A chamada SQL (Structured Query Language) é uma linguagem para definição e manipulação de dados relacionais que foi desenvolvida pela IBM e padronizada pelos comitês ISO/Ansi.

Um poderoso comando da linguagem SQL é o Select, pois é utilizado para fazer consultas (queries) e suas variantes servem para operações de inserção, remoção e atualização de tabelas.

O comando Select apresenta diversas cláusulas que devem respeitar uma ordem prefixada. Exemplo: **SELECT** MATRICULA, NOME **FROM** FUNCIONARIOS **WHERE** MATRICULA = 10.

A cláusula select é obrigatória e compreende uma lista de atributos (MATRICULA e NOME) que definem o esquema de resultado.

A cláusula from é obrigatória e indica a tabela (FUNCIONARIOS) da qual queremos obter os dados. Podemos precisar de duas ou mais tabelas. Isso corresponde ao produto cartesiano dessas tabelas.

A cláusula where é opcional e seleciona as linhas da tabela que satisfazem à condição dada. No exemplo, o funcionário que tem matrícula igual a 10.

2. Análise da questão

A questão pede uma análise da operação de junção de três tabelas. É solicitado também um relatório dessa operação, contendo o nome de todos os funcionários que enviaram os malotes, o código dos malotes enviados, a descrição de seus conteúdos e a situação dos malotes.

Podemos observar que a tabela MALOTES tem da tabela FUNCIONARIOS a matrícula, e da tabela CONTEUDOS, o CODIGO_CONTEUDO. Para obter o nome de todos os funcionários (essa informação está na tabela FUNCIONARIOS), devemos fazer a junção das tabelas MALOTES e FUNCIONARIOS utilizando a coluna MATRICULA.

O código dos malotes e a situação estão na tabela MALOTES nas colunas CODIGO_MALOTES e SITUAÇÃO_MALOTES, respectivamente. A descrição de seus conteúdos está na tabela CONTEUDOS. Também devemos fazer a junção entre as tabelas MALOTES e CONTEUDOS utilizando a coluna CODIGO_CONTEUDO.

Para atender corretamente à solicitação do relatório, cada linha da tabela MALOTES deve ter correspondência, em seus valores de atributos, com as tabelas FUNCIONARIOS e CONTEUDOS, isto é, uma junção (join) com correspondência interna (inner).

Dessa forma, nosso Select teria o aspecto mostrado a seguir.

```
SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES  
INNER JOIN CONTEUDOS ON (MALOTES.CODIGO_CONTEUDO =  
CONTEUDOS.CODIGO_CONTEUDO) INNER JOIN FUNCIONARIOS ON(MALOTES.MATRICULA  
= FUNCIONARIOS.MATRICULA);
```

Observe que, para evitarmos ambiguidade, usamos a notação MALOTES.CODIGO_CONTEUDO com o significado “a coluna CODIGO_CONTEUDO da tabela MALOTES”.

Alternativa correta: D.

3. Indicações bibliográficas

- GUIMARÃES, C. *Fundamentos de bancos de dados: modelagem, projeto e linguagem SQL*. 2. ed. São Paulo: Unicamp, 2003.
- DATE, C. *Introdução a sistemas de banco de dados*. 8. ed. Rio de Janeiro: Elsevier, 2003.
- SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. *Sistemas de banco de dados*. 5. ed. Rio de Janeiro: LTC, 2020.
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2005.

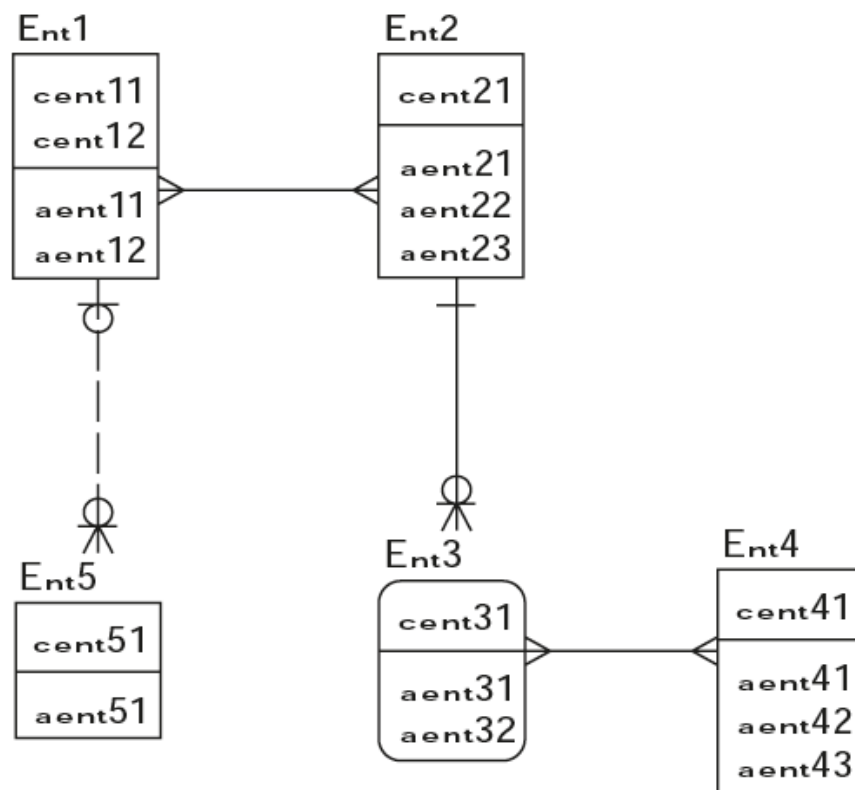
Questão 7

Questão 7.⁷

Considere o diagrama de entidades e relacionamentos a seguir, onde as chaves primárias de cada entidade se encontram na parte superior dos retângulos. As entidades fortes são representadas por retângulos e as entidades fracas são representadas por retângulos com cantos arredondados.

O diagrama atende às seguintes restrições

- I. entre Ent1 e Ent2, tem-se um relacionamento de muitos para muitos;
- II. entre as Entidades Ent2 e Ent3, tem-se um relacionamento de um para nenhum, um ou muitos;
- III. entre Ent1 e Ent5, tem-se um relacionamento de zero ou um para zero, um ou muitos; e
- IV. entre Ent3 e Ent4, tem-se um relacionamento de muitos para muitos.



Aplicando a terceira forma normal ao modelo, qual será o total de colunas que deve ser criado para representar as chaves estrangeiras?

- A. 3.
- B. 5.
- C. 7.
- D. 8.
- E. 9.

⁷Questão 23 - Enade 2011.

1. Introdução teórica

Diagrama Entidade-Relacionamento (DER)

DER é uma modelagem semântica baseada no modelo de entidade/relacionamento. A primeira etapa para se desenvolver um DER é identificar um conjunto de conceitos semânticos que parecem ser úteis para descrever o mundo real.

Entidade é um objeto perceptível e que pode ser identificado distintamente, como, por exemplo, funcionários, malotes etc.

Podemos classificar as entidades em fracas e fortes.

Uma entidade fraca é uma entidade cuja existência depende de outra. Como exemplo, temos os dependentes de um funcionário, pois o dependente não pode existir se o funcionário também não existir.

Uma entidade forte é uma entidade cuja existência não depende de outra. Como exemplo, temos o funcionário.

Para garantir a integridade dos dados em um banco de dados, precisamos implementar restrições que garantam essa integridade. Para isso, utilizamos as chaves, que são conceitos importantes da modelagem de dados.

Há dois tipos de chaves: a primária e a estrangeira. A chave primária é um atributo ou uma combinação de atributos que possibilita a identificação exclusiva de cada registro armazenado na tabela. Na chave estrangeira, há um relacionamento entre tabelas distintas de um banco de dados. Utilizamos a chave primária de uma tabela como atributo de outra tabela, esse novo atributo é chamado de chave estrangeira.

O processo formal que analisa as relações entre as tabelas, com o objetivo de evitar anomalias nos processos de inclusão, exclusão e alteração de registros, chama-se normalização de dados. O processo de normalização utiliza uma série de regras sobre as tabelas de um banco de dados para projetá-las corretamente.

Das cinco regras existentes utilizamos na prática três Formas Normais (FN), indicadas por 1FN, 2FN e 3FN, conforme descrito a seguir.

- **Primeira Forma Normal (1FN).** Os valores atômicos (ou indivisíveis) são os únicos permitidos, conforme exemplificado a seguir.

Código_Cliente	Nome	Endereço
C001	José da Silva	Rua Vinte, 153 – Moema - 12345-23
C002	João dos Santos	Rua Trinta, 259 – Ipiranga – 56478-45
C003	Maria Pereira	Rua Quarenta, 698 – Liberdade – 25978-89

A tabela não está na primeira forma normal, pois as informações de Rua, CEP e Bairro estão na mesma célula da tabela. Para normalizar, devemos colocar cada informação em uma coluna diferente, conforme segue.

Código_Cliente	Nome	Rua	Bairro	CEP
C001	José da Silva	Rua Vinte, 153	Moema	12345-23
C002	João dos Santos	Rua Trinta, 259	Ipiranga	56478-45
C003	Maria Pereira	Rua Quarenta, 698	Liberdade	25978-89

- **Segunda Forma Normal (2FN).** A tabela está na 2FN se ela estiver na 1FN e se todos os atributos não chave estiverem dependentes da chave primária. A 2FN é baseada no conceito de dependência funcional total e evita que os valores fiquem redundantes no banco de dados, conforme exemplificado na tabela de pedidos a seguir.

Num_Pedido	Codigo_Produto	Produto	Quant	Valor_unit	Subtotal
101	P025	Desodorante	200	1,15	230,00
102	P032	Sabonete	500	0,50	250,00
103	P013	Escova	300	2,00	600,00

A tabela não está na segunda forma normal, pois o nome do produto depende de seu código e não depende do Número do Pedido. Para normalizar, construímos uma tabela para produto, conforme segue.

Codigo_Produto	Produto
P025	Desodorante
P032	Sabonete
P013	Escova

A tabela de pedido estará em 2FN, conforme segue.

Num_Pedido	Codigo_Produto	Quant	Valor_unit	Subtotal
101	P025	200	1,15	230,00
102	P032	500	0,50	250,00
103	P013	300	2,00	600,00

- **Terceira Forma Normal (3FN).** A tabela está na 3FN se ela estiver na 2FN e se nenhuma coluna não chave depender de outra coluna não chave. A 3FN é baseada no conceito de dependência transitiva. Como exemplo, vamos reutilizar a tabela de pedidos que está em 2FN.

Num_Pedido	Codigo_Produto	Quant	Valor_unit	Subtotal
101	P025	200	1,15	230,00
102	P032	500	0,50	250,00
103	P013	300	2,00	600,00

Podemos eliminar o campo subtotal (não chave), pois podemos resolver uma equação ($\text{Quant} * \text{Valor_unit}$) utilizando outros campos não chave da tabela para obter o valor. A tabela de pedido estará em 3FN, conforme segue.

Num_Pedido	Codigo_Produto	Quant	Valor_unit
101	P025	200	1,15
102	P032	500	0,50
103	P013	300	2,00

Podemos classificar o relacionamento entre Entidade/Relacionamento de três formas diferentes, conforme descrito a seguir.

- **Um para um:** uma entidade A se relaciona a uma única entidade B e vice-versa. Exemplo: um funcionário gerencia um único departamento e um departamento é gerenciado apenas por um funcionário.
- **Um para muitos:** uma entidade A se relaciona a várias entidades B. Exemplo: um funcionário envia vários malotes.
- **Muitos para muitos:** várias entidades A se relacionam a várias entidades B e vice-versa. Exemplo: um funcionário pode participar de vários projetos e um projeto pode ter vários funcionários.

2. Análise da questão

A questão pede a quantidade de chaves estrangeiras criadas no modelo apresentado e solicita que esse modelo esteja na 3FN.

Para resolver a questão, devemos analisar todos os relacionamentos e identificar onde é necessária a criação das chaves estrangeiras.

Entre Ent1 e Ent2 e entre Ent3 e Ent4, encontramos os relacionamentos de muitos para muitos, que devem ser transformados em dois relacionamentos “um” para “muitos”, resultando em uma nova tabela.

Entre Ent1 e Ent5 e entre Ent2 e Ent3, encontramos os relacionamentos de um para muitos, que são mapeados de maneira que a chave primária do lado “um” seja representada do lado “muitos” como chave estrangeira.

Entre Ent1 e Ent5, temos um relacionamento que necessita de uma nova tabela. Podemos exemplificar o relacionamento entre essas entidades como Empregado (Ent1) e Máquina (Ent5). Um ou zero empregado pode utilizar zero, uma ou muitas máquinas. Dessa forma, temos o que segue.

Ent1 e Ent2	Nova tabela Ent6 com as chaves estrangeiras cent11 , cent12 e cent21
Ent3 e Ent4	Nova tabela Ent7 com as chaves estrangeiras cent31 , cent41
Ent1 e Ent5	Nova tabela Ent8 com as chaves estrangeiras cent11 , cent12 e cent51
Ent3	Receberá a chave estrangeira cent21 .

Alternativa correta: E.

3. Indicações bibliográficas

- DATE, C. *Introdução a sistemas de banco de dados*. 8. ed. Rio de Janeiro: Elsevier, 2003
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2005.
- GUIMARÃES, C. *Fundamentos de bancos de dados: modelagem, projeto e linguagem SQL*. 2. ed. São Paulo: Unicamp, 2003.
- SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. *Sistemas de banco de dados*. 5. ed. Rio de Janeiro: LTC, 2020.

Questão 8

Questão 8.⁸

A pilha é uma estrutura de dados que permite a inserção/remoção de itens dinamicamente seguindo a norma de último a entrar, primeiro a sair. Suponha que para uma estrutura de dados, tipo pilha, são definidos os comandos:

- PUSH (p, n): Empilha um número “n” em uma estrutura de dados do tipo pilha “p”;
- POP (p): Desempilha o elemento no topo da pilha.

Considere que, em uma estrutura de dados tipo pilha “p”, inicialmente vazia, sejam executados os seguintes comandos

PUSH (p, 10)

PUSH (p, 5)

PUSH (p, 3)

PUSH (p, 40)

POP (p)

PUSH (p, 11)

PUSH (p, 4)

PUSH (p, 7)

POP (p)

POP (p)

Após a execução dos comandos, o elemento no topo da pilha “p” e a soma dos elementos armazenados na pilha “p” são, respectivamente,

- A. 11 e 29. B. 11 e 80. C. 4 e 80. D. 7 e 29. E. 7 e 40.

1. Introdução teórica

Pilha

Pilha é uma estrutura de dados, isto é, uma forma particular de armazenar um conjunto ordenado de itens. Essa estrutura permite a inserção de novos itens ou a exclusão de itens existentes.

A estrutura de dados do tipo pilha funciona analogamente a uma pilha de pratos. Caso precisemos adicionar um novo prato na pilha, devemos colocá-lo no topo, e para retirarmos um prato, também devemos extrair o prato que está no topo.

⁸Questão 29 - Enade 2011.

A estrutura de dados do tipo pilha tem como característica a ordenação LIFO - Last In First Out, ou seja, a última informação a entrar é a primeira a sair.

Quando utilizamos a pilha, precisamos implementar duas operações básicas: a operação para empilhar/inserir um novo elemento no topo da pilha e a operação de desempilhar/remover um elemento do topo. É comum referirmo-nos a essas duas operações pelos termos em inglês PUSH (empilhar) e POP (desempilhar). A operação PUSH(p,n) irá acrescentar um novo elemento "n" ao topo da pilha "p" e a operação POP(p) irá remover da pilha "p" o elemento que está no topo.

As aplicações computacionais que precisam de uma estrutura de dados do tipo pilha sabem antecipadamente o número máximo de elementos que devem ser armazenados. Nesses casos, podemos utilizar vetores, pois a quantidade de elementos é conhecida. A utilização de pilhas é mais comum em compiladores e sistemas operacionais do que em controle de dados, alocação de variáveis na memória etc.

2. Análise da questão

Executando os comandos considerados na questão, os elementos serão armazenados na pilha "p", conforme mostra a figura 1.

							7		
						4	4	4	
			40		11	11	11	11	11
		3	3	3	3	3	3	3	3
	5	5	5	5	5	5	5	5	5
10	10	10	10	10	10	10	10	10	10

Figura 1. Comandos em pilha.

Após a execução de todos os comandos, o elemento 11 está no topo da pilha "p" e a soma de todos os elementos (11+3+5+10) resulta em 29.

Alternativa correta: A.

3. Indicações bibliográficas

- LAUREANO, M. *Estruturas de dados com algoritmos e C*. Rio de Janeiro: Brasport, 2008.
- SCHILDT, H. *C completo e total*. 3. ed. São Paulo: Makron, 1997.

ÍNDICE REMISSIVO

Questão 1	Elicitação de requisitos.
Questão 2	Linguagem de Modelagem Unificada (Unified Modeling Language – UML)
Questão 3	Linguagem de Modelagem Unificada (Unified Modeling Language – UML).
Questão 4	Diagrama de atividades.
Questão 5	Proposição.
Questão 6	Diagrama Entidade-Relacionamento (DER).
Questão 7	Diagrama Entidade-Relacionamento (DER).
Questão 8	Pilha.