



# **TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**MATERIAL INSTRUCIONAL ESPECÍFICO**

**TOMO 7**

# **CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO**

## **TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

### **MATERIAL INSTRUCIONAL ESPECÍFICO**

#### **TOMO 7**

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

## Questão 1

### Questão 1.<sup>1</sup>

No cenário de uma economia globalizada, cresce o interesse pelo empreendedorismo e pela busca de meios que levem a uma maior produtividade, competitividade e inovação. Os avanços das tecnologias da informação e comunicação (TIC) auxiliam esse crescimento. Atualmente, uma parcela significativa dos negócios tem uma dependência forte das TIC. Desse modo, manter a disponibilidade da informação e comunicação e manter os negócios operando, sem qualquer paralização, é indispensável. Porém, é preciso analisar o que pode ser afetado, qual o impacto financeiro e quais os impactos na imagem e na reputação da empresa, se cada um dos processos de negócio sofresse uma paralização por conta da TIC. A fim de mitigar possíveis riscos, é recomendável documentar um plano para eliminar ou reduzir a possibilidade de ocorrer cenários de indisponibilidade da TIC. Nessa situação, é preciso elaborar um

- A. plano de negócio.
- B. documento de visão.
- C. plano de contingência.
- D. plano de gerência de riscos
- E. plano de gerenciamento de projetos.

## 1. Introdução teórica

### 1.1. Missão, visão e valores de uma empresa

Segundo Silva (2017), o conjunto formado pela missão, pela visão e pelos valores representa a identidade organizacional. Todos devem saber claramente a respeito do propósito e da razão da existência da organização, além dos valores em que ela está calcada.

Esse conjunto é o norteador de todas as ações da empresa e deve garantir a coerência entre o que a organização se propõe a fazer e o que ela realmente faz (MATOS, MATOS e ALMEIDA, 2007).

### 1.2. Plano de negócios

O plano de negócio é um documento que descreve os objetivos de um negócio e traça os procedimentos que serão adotados para que esses objetivos sejam alcançados, a

---

<sup>1</sup>Questão 12 – Enade 2014.

fim de que os riscos e as incertezas sejam diminuídos. Esse plano permite identificar possíveis erros de planejamento antes que eles sejam cometidos no mercado (SEBRAE, 2013).

Vale enfatizar que o plano de negócios é um documento em que estão descritas a caracterização do negócio, sua operação, suas estratégias e suas projeções de receitas, despesas e resultados financeiros (SALIM, 2005).

### **1.3. Plano de gerenciamento de projetos**

O plano de gerenciamento de projetos tem como objetivos propor a documentação relativa às premissas e às restrições de determinado projeto e delinear seu escopo, seus prazos, seus riscos, suas definições e suas responsabilidades (POSSI, 2006).

### **1.4. Plano de gerenciamento de riscos**

O plano de gerenciamento dos riscos descreve como os processos que envolvem riscos são estruturados e executados.

Esse plano identifica os riscos, realiza análises qualitativas e quantitativas, constrói a proposta de respostas e determina como os riscos são controlados e monitorados (ESCRITÓRIO DE PROJETOS, 2017).

### **1.5. Plano de contingência**

Para Romero (2016), contingência é uma eventualidade, um acaso, um acontecimento que tem como base a incerteza de ocorrência. Assim, qualquer que seja a atividade de uma empresa, é necessário que existam alternativas para mantê-la em perfeito funcionamento. Esse conjunto de alternativas é chamado de plano de contingência.

Segundo Rezende (2005), para cobrir uma possível falha nos sistemas originais, a área de tecnologia da informação e comunicação (TIC) deve estabelecer um plano de contingência que apresente as alternativas para o processamento de dados e a execução dos sistemas organizacionais.

## 2. Análise das alternativas.

A – Alternativa incorreta.

JUSTIFICATIVA. O plano de negócio estabelece os objetivos de um negócio e delinea os procedimentos adotados para que esses objetivos sejam alcançados, o que diminui os riscos e as incertezas.

B - Alternativa incorreta.

JUSTIFICATIVA. A missão, a visão e os valores de uma organização representam a identidade organizacional. Esse conjunto constitui os princípios sobre os quais a empresa se estabelece.

C - Alternativa correta.

JUSTIFICATIVA. O plano de contingência procura traçar as alternativas para evitar que, em função de uma falha ou de uma eventualidade, seja interrompida a prestação de determinado serviço ou a produção de um item.

D - Alternativa incorreta.

JUSTIFICATIVA. O plano de gerência de risco identifica os riscos, faz as análises qualitativas e quantitativas desses riscos, constrói o plano de respostas e determina como os riscos são controlados e monitorados.

E - Alternativa incorreta.

JUSTIFICATIVA. O plano de gerenciamento de projetos documenta as bases e as restrições existentes para o desenvolvimento de determinado projeto. Esse plano estabelece o escopo, os prazos, os riscos, as definições e as responsabilidades.

## 3. Indicações bibliográficas

- ESCRITÓRIO DE PROJETOS. *Plano de gerenciamento de riscos*. Disponível em <<https://escritoriodeprojetos.com.br/projeto.../129-plano-de-gerenciamento-dos-riscos>>. Acesso em 23 ago. 2017.

- MATOS, J. G. R.; MATOS, R. M. B.; ALMEIDA, J. R. *Análise do ambiente corporativo: do caos organizado ao planejamento estratégico das organizações*. Rio de Janeiro: E-papers, 2007.
- POSSI, M. *Gerenciamento de projetos: guia do profissional*. v. 3. Rio de Janeiro: Brasport, 2006.
- REZENDE, D. A. *Engenharia de software e sistemas de informação*. Rio de Janeiro: Brasport, 2005.
- ROMERO, G. B. *Plano de contingência*. São Paulo: Clube dos Autores, 2016.
- SALIM, C. S. et al. *Construindo planos de negócios*. Rio de Janeiro: Elsevier, 2005.
- SERVIÇO BRASILEIRO DE APOIO ÀS MICROS E PEQUENAS EMPRESAS - SEBRAE. *Como elaborar um plano de negócios*. Brasília: Sebrae, 2013.
- SILVA, R. D. *Planejamento estratégico – missão, visão, valores. Sua importância para a estrutura do seu negócio*. Disponível em <<http://www.administradores.com.br/artigos/negocios/planejamento-estrategico-missao-visao-valores-sua-importancia-para-a-estrutura-do-seu-negocio/91374/>>. Acesso em 23 ago. 2017.

## Questão 2

### Questão 2.<sup>2</sup>

Leia o texto a seguir.

*A Lei Nº 12.965, de 23 de abril de 2014, originalmente denominada Marco Civil da Internet, estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil, contribuindo principalmente nas relações de consumo e segurança da informação pessoal para com os prestadores de serviço de acesso à Internet.*

BRASIL. Lei Nº 12.965, de 23 de abril de 2014.

Suponha que o cliente de determinado provedor de acesso à Internet contrate um serviço com velocidade de conexão de 5 Megabits/s (Mbps), mas tenha continuamente enfrentado problemas de perda de conexão e, adicionalmente, não obtém a velocidade contratada, apesar de o cliente realizar todos os pagamentos em dia.

Nessa situação, avalie as seguintes asserções e a relação proposta entre elas.

I. A manutenção da qualidade contratada da conexão à Internet é uma obrigação prevista no Marco Civil da Internet.

PORQUE

II. Não pode haver suspensão da conexão à Internet, salvo por débito diretamente decorrente de sua utilização.

A respeito dessas asserções, assinale a opção correta.

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

### 1. Introdução teórica

#### Marco Civil da Internet

O artigo primeiro do Capítulo I da Lei Nº 12.965/2014 estabelece os princípios do Marco Civil da Internet, conforme indicado a seguir.

*Esta lei estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil e determina as diretrizes para atuação da União, dos Estados, do Distrito Federal e dos Municípios em relação à matéria.*

---

<sup>2</sup>Questão 13 – Enade 2014.

No Capítulo II da Lei Nº 12965/2014, é abordada a questão dos direitos e garantias dos usuários. O artigo 7º desse capítulo está reproduzido a seguir.

*Art. 7º O acesso à Internet é essencial ao exercício da cidadania, e ao usuário são assegurados os seguintes direitos:*

*I - Inviolabilidade da intimidade e da vida privada, sua proteção e indenização pelo dano material ou moral decorrente de sua violação;*

*II - inviolabilidade e sigilo do fluxo de suas comunicações pela Internet, salvo por ordem judicial, na forma da lei;*

*III - inviolabilidade e sigilo de suas comunicações privadas armazenadas, salvo por ordem judicial;*

*IV - Não suspensão da conexão à Internet, salvo por débito diretamente decorrente de sua utilização;*

*V - Manutenção da qualidade contratada da conexão à Internet;*

*VI - informações claras e completas constantes dos contratos de prestação de serviços, com detalhamento sobre o regime de proteção aos registros de conexão e aos registros de acesso a aplicações de Internet, bem como sobre práticas de gerenciamento da rede que possam afetar sua qualidade.*

## **2. Análise das asserções**

I – Asserção verdadeira.

JUSTIFICATIVA. De acordo com o item V do artigo 7º do Capítulo II da Lei Nº 12.965/2014, os provedores devem garantir a qualidade do serviço contratado.

II – Asserção verdadeira.

JUSTIFICATIVA. De acordo com o item IV do artigo 7º do Capítulo II da Lei Nº 12.965/2014, não deve haver suspensão arbitrária da conexão. A desconexão é aplicável apenas no caso de o cliente não estar em dia com os pagamentos do serviço contratado, o que não é a situação proposta na questão.

Ainda que ambas as asserções sejam verdadeiras, não existe relação de causalidade entre elas. Essas asserções dizem respeito a aspectos diferentes da legislação: a asserção I refere-se à obrigação dos provedores de serviço com relação à qualidade e a asserção II refere-se às condições que podem levar à suspensão do serviço.

Alternativa correta: B.



### 3. Indicações bibliográficas

- BRASIL. Lei Nº 12.965, de 23 de abril de 2014. Disponível em <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2014/lei/l12965.htm](http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm)>. Acesso em 17 mai. 2017.
- DE JESUS, D.; MILAGRE, J. A. *Marco Civil da Internet – comentários à Lei Nº 12.965/14*. São Paulo: Saraiva, 2014.

### Questão 3 e 4

#### Questão 3.<sup>3</sup>

Requisitos funcionais e não funcionais podem definir características e funcionalidades presentes na interface a ser desenvolvida para um sistema.

Em relação aos requisitos não funcionais, também chamados de requisitos de qualidade, avalie as afirmativas.

- I. São levantados e elicitados após os requisitos funcionais, uma vez que os requisitos funcionais determinarão as funcionalidades da interface.
- II. Sempre serão definidos de forma mais concreta, através de requisitos funcionais, uma vez que o usuário manipula na interface somente as funcionalidades levantadas.
- III. Podem complementar os requisitos funcionais.

É correto o que se afirma em

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

#### Questão 4.<sup>4</sup>

A engenharia de software considera diversos aspectos para a garantia da qualidade. Os requisitos funcionais definem como um sistema deverá se comportar em relação as suas funcionalidades básicas, já os requisitos não funcionais avaliam outros aspectos do software.

São exemplos de requisitos não funcionais a serem considerados em um software

- A. segurança, desempenho, estresse e sistema.
- B. usabilidade, segurança, aceitação e confiabilidade.
- C. usabilidade, segurança, desempenho e confiabilidade
- D. segurança, aceitação, testabilidade e confidencialidade.
- E. usabilidade, confidencialidade, aceitação e confiabilidade.

---

<sup>3</sup>Questão 19 – Enade 2014.

<sup>4</sup>Questão 25 – Enade 2014.

## 1. Introdução teórica

### Requisitos não funcionais

Projetos de software envolvem a identificação de diversos tipos de requisitos. Alguns deles estão diretamente atrelados ao comportamento do software sob determinada condição (WIEGERS e BEATTY, 2013), enquanto outros referem-se a propriedades do sistema que não estão diretamente ligados à funcionalidade principal e, por isso, costumam ser chamados de requisitos não funcionais (WAGNER, 2013). Frequentemente, os requisitos não funcionais estão atrelados a aspectos da qualidade de software (WAGNER, 2013).

Devido aos diversos usos que a palavra “requisito” pode ter na área de software, alguns autores, como Wiegers e Beatty (2013), costumam fornecer um vocabulário dos vários possíveis significados associados a esse termo e a outros correlatos. A tabela 1 apresenta um resumo desses significados.

**Tabela 1.** Tipos de requisitos.

<b>Termo</b>	<b>Definição</b>
Requisito de negócio	Um objetivo de negócio de alto nível de que uma organização dispõe na construção de um produto.
Regra de negócio	Uma prática, uma diretriz, um padrão ou uma regulamentação que define ou restringe algum aspecto do negócio. Está na origem de diversos tipos de requisitos de software.
Restrição	A restrição no projeto e a construção de um produto que são impostas nas escolhas disponíveis para o desenvolvedor.
Requisito de interface externa	A descrição de uma conexão entre um sistema de software e um usuário, um outro sistema ou um equipamento de que a empresa dispõe.
Característica ( <i>feature</i> )	Uma ou mais competências do sistema, que são descritas por um conjunto de requisitos funcionais.
Requisitos funcionais	A descrição de um comportamento, que é exibido pelo sistema em condições específicas.
Requisitos não funcionais	Uma descrição ou uma característica de uma propriedade que um sistema exibe ou uma restrição a ser respeitada.
Atributo de qualidade	Um tipo de requisito não funcional que descreve um serviço ou uma característica da performance de um produto.
Requisito de sistema	Um requisito de alto nível para um produto que contém múltiplos subsistemas e pode referir-se a todo o software ou ao software e ao hardware.
Requisito de usuário	Um objetivo ou uma tarefa específica para uma classe de usuários, a que o sistema precisa ser capaz de atender.

**Fonte.** WIEGERS e BEATTY, 2013 (com adaptações).

Ainda em Wiegers e Beatty (2013), encontra-se um capítulo totalmente dedicado às qualidades externas e internas na área de software.

As qualidades externas estão ligadas aos atributos do software, perceptíveis quando o software é executado, enquanto as qualidades internas contribuem de forma indireta para a satisfação do usuário. Entre os aspectos externos de qualidade, podemos destacar os que seguem.

- **Usabilidade:** facilidade que o usuário tem de aprender, memorizar e utilizar o sistema.
- **Segurança:** sistema que protege o usuário contra danos (isso é especialmente importante em máquinas e dispositivos mecânicos) ou sistema que está protegido em relação ao seu acesso e ao acesso das informações nele contidas.
- **Desempenho:** agilidade com que o sistema responde às entradas dos usuários e a outros eventos.
- **Confiabilidade:** ligada ao tempo que o sistema executa sem experimentar uma falha.

## 2. Análise das afirmativas e das alternativas

### Questão 3.

I - Afirmativa incorreta.

JUSTIFICATIVA. Os requisitos não funcionais não devem ser levantados apenas depois dos requisitos funcionais. Os requisitos não funcionais não são “menos importantes” do que os requisitos funcionais: eles são apenas diferentes.

II - Afirmativa incorreta.

JUSTIFICATIVA. Os requisitos não funcionais, em geral, envolvem aspectos ligados à qualidade e à usabilidade e, em função disso, podem não estar totalmente ligados aos requisitos funcionais, como a afirmativa sugere.

III - Afirmativa correta.

JUSTIFICATIVA. Os requisitos não funcionais complementam os requisitos funcionais, incorporando elementos importantes para o cliente, como, por exemplo, a facilidade de utilização do sistema.

Alternativa correta: B.

#### Questão 4.

A - Alternativa incorreta.

JUSTIFICATIVA. Estresse e sistema não são exemplos de requisitos não funcionais.

B, D e E - Alternativas incorretas.

JUSTIFICATIVA. A aceitação não é um requisito não funcional.

C - Alternativa correta.

JUSTIFICATIVA. Usabilidade, segurança, desempenho e confiabilidade são bons exemplos de requisitos não funcionais.

Alternativa correta: C.

#### 3. Indicação bibliográfica

- WAGNER, S. *Software product quality control*. Berlin: Springer, 2013.
- WAZLAWICK, R. *Análise e design orientados a objetos para Sistemas de Informação: Modelagem com UML, OCL e IFML*. 3. ed. Rio de Janeiro: Campus, 2016.
- WIEGERS, K.; BEATTY, J. *Software requirements*. 3. ed. Redmond: Microsoft Press, 2013.

## Questão 5

### Questão 5.<sup>5</sup>

Leia o texto a seguir.

*O barramento é o elemento de conexão entre todos os componentes do computador, como memória, CPU e dispositivos de entrada e saída. O barramento de dados é o meio por onde serão trafegados os dados. O barramento de endereços transporta a informação do endereço do dispositivo que poderá acessar o barramento de dados; e o barramento de controle serve para determinar o sentido do fluxo de dados (se os dados são de entrada ou saída da CPU), e se os dados devem ser destinados à memória ou aos dispositivos de I/O e também para controlar o clock no barramento.*

STALLINGS, W. *Arquitetura e Organização de Computadores*. 8 ed. São Paulo: Pearson Prentice Hall, 2010 (com adaptações).

Considerando um computador com um barramento de dados de 4 bits e barramento de endereços de 3 bits, ele poderá endereçar, respectivamente, quantas posições de memória e quantos dispositivos de I/O?

- A. 4 e 4.
- B. 4 e 8.
- C. 8 e 3.
- D. 8 e 4.
- E. 8 e 8.

## 1. Introdução teórica

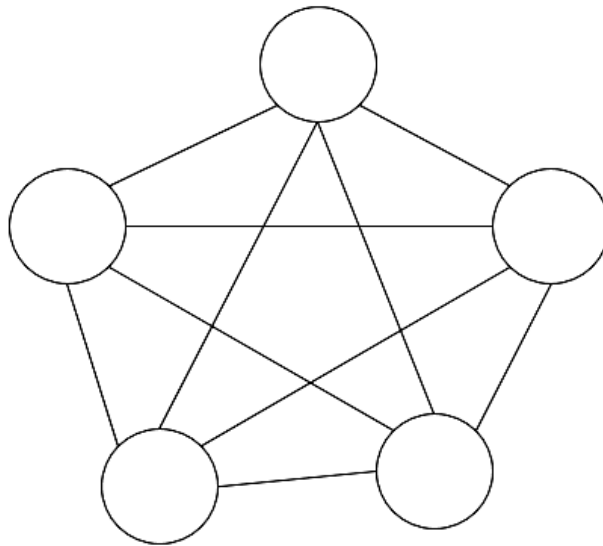
### Barramentos

Um computador é um dispositivo eletrônico formado por diversos componentes que se comunicam de forma rápida e eficiente. Podemos dizer que seus elementos principais são a CPU, as memórias e os dispositivos de entrada e saída. É necessário que esses dispositivos se comuniquem uns com os outros e a forma como isso é realizado tem efeitos no custo e no desempenho de um equipamento. Isso leva à questão da interconexão entre os diversos elementos que constituem um computador.

Para compreendermos o problema da interconexão em uma situação na qual dispomos de vários dispositivos (por exemplo, 5 elementos), vamos pensar na elaboração de um sistema em que todos os dispositivos possam se comunicar uns com os outros. Uma forma de fazer isso é conectar diretamente todos os elementos entre si, como indicado na figura 1.

---

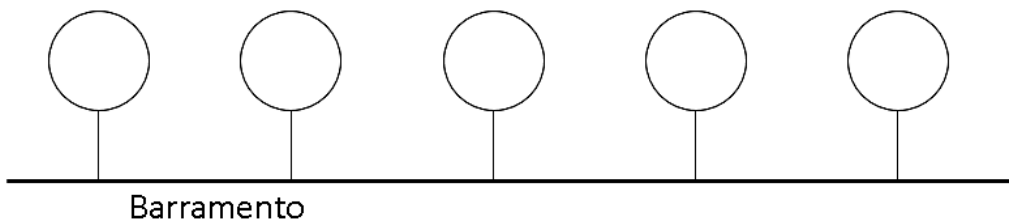
<sup>5</sup>Questão 27 – Enade 2014.



**Figura 1.** Interconexão de vários elementos.

O problema dessa abordagem é que o número de interconexões aumenta de forma muito rápida com o crescimento do número de elementos a serem conectados, o que torna essa abordagem inviável para dispositivos complexos, como um computador.

Uma outra abordagem consiste em se construir um barramento, uma espécie de “rodovia” que interconecta todos os elementos, como indicado na figura 2. Dessa forma, ao conectarmos novos elementos no sistema, não precisamos conectar o novo elemento a todos os demais; podemos conectá-lo apenas ao barramento.



**Figura 2.** Interconectando vários elementos através de um barramento.

## 2. Análise da questão

Considerando que o barramento de endereços atua tanto para o endereçamento da memória quanto para o atendimento aos dispositivos de entrada e saída, e sabendo que esse barramento tem largura de 3 bits (valor dado na questão), ele deve ser capaz de endereçar  $2^3=8$  posições diferentes. Na arquitetura do enunciado, o barramento de controle vai definir quando esses bits endereçam posições de memória ou dispositivos de entrada e

saída, sendo que os barramentos de dados serão utilizados para ambos os casos, endereçando 8 posições de memória diferentes ou 8 dispositivos diferentes.

Alternativa correta: E.

### **3. Indicações bibliográficas**

- PATTERSON, D.; HENESSY, J. L. *Arquitetura de computadores: uma abordagem quantitativa*. Rio de Janeiro: Elsevier, 2014.
- RIBEIRO, C; DELGADO, J. *Arquitetura de computadores*. São Paulo: LTC, 2014.



## Questão 6 e 7

### Questão 6.<sup>6</sup>

Leia o texto a seguir.

*UML é uma linguagem padrão para desenvolver e documentar projetos de software e permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Ela surgiu como uma proposta de ser uma linguagem para modelagem de dados que usava diversos artefatos para representar o modelo de negócio e um desses artefatos é o diagrama de classes.*

PRESSMAN, R.S. *Engenharia de software*. 6. ed. Porto Alegre: Bookman, 2006 (com adaptações).

Se um projeto não tem a documentação apropriada ou se está com a documentação desatualizada, uma opção é a engenharia reversa que possibilita mapear códigos para diagramas UML. A seguir, é apresentado um código na linguagem de programação JAVA.

```

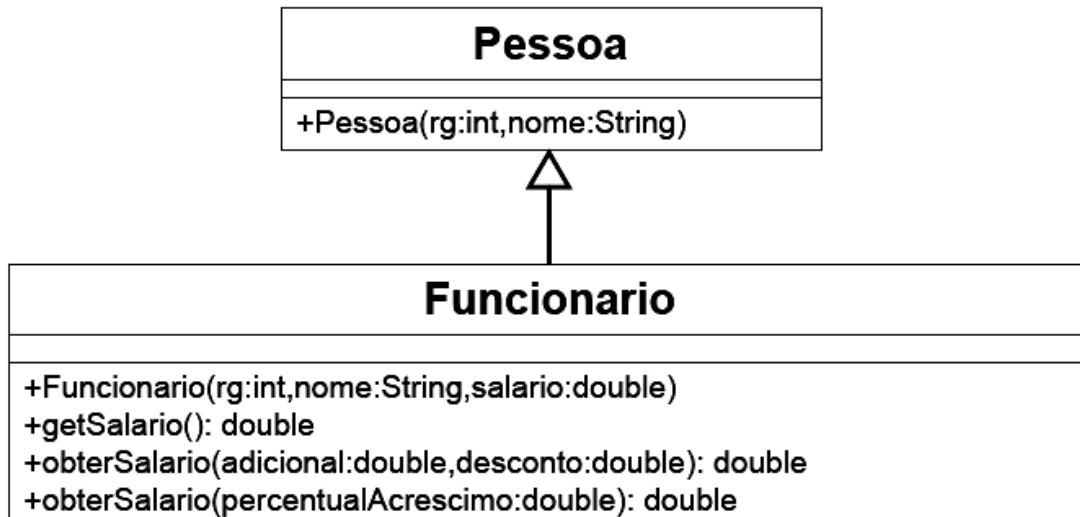
1 package default;
2
3 public class Funcionario extends Pessoa {
4     private double salario;
5
6     public Funcionario(int rg, String nome,
7         double salario) {
8         super(rg, nome);
9         this.salario = salario;
10    }
11    public double getSalario() {
12        return salario;
13    }
14    }
15    public double obterSalario(double
16        percentualAcrescimo) {
17        double salarioReajustado = salario +
18            salario * percentualAcrescimo / 100;
19        return salarioReajustado;
20    }
21    }
22    public double obterSalario(double
23        adicional, double desconto){
24        return this.getSalario() + adicional - desconto;
25    }
26    }
27 }
```

Utilizando a engenharia reversa nesse trecho de código, o diagrama UML de classes correspondente é

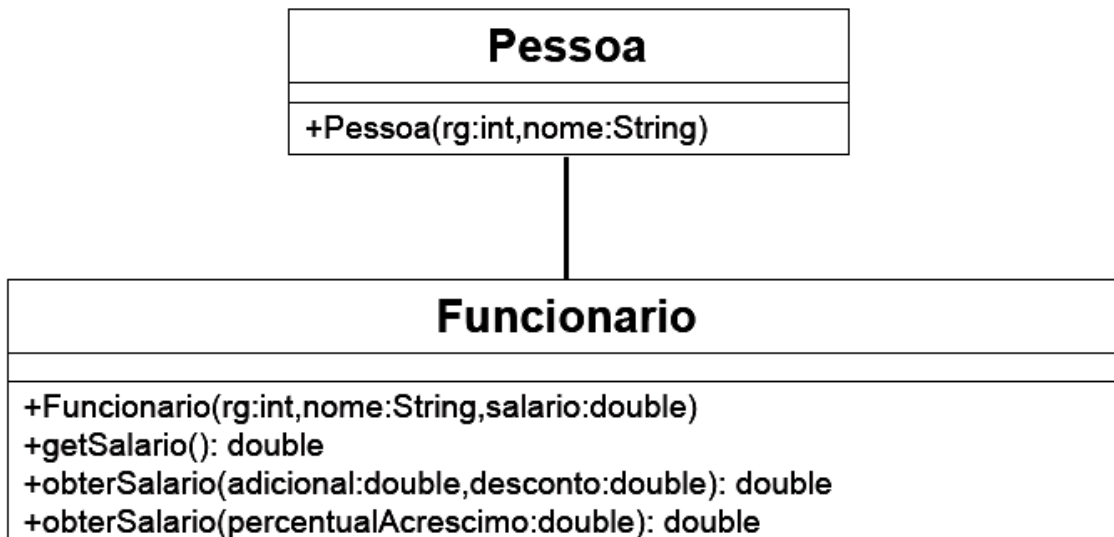
---

<sup>6</sup>Questão 29 – Enade 2014.

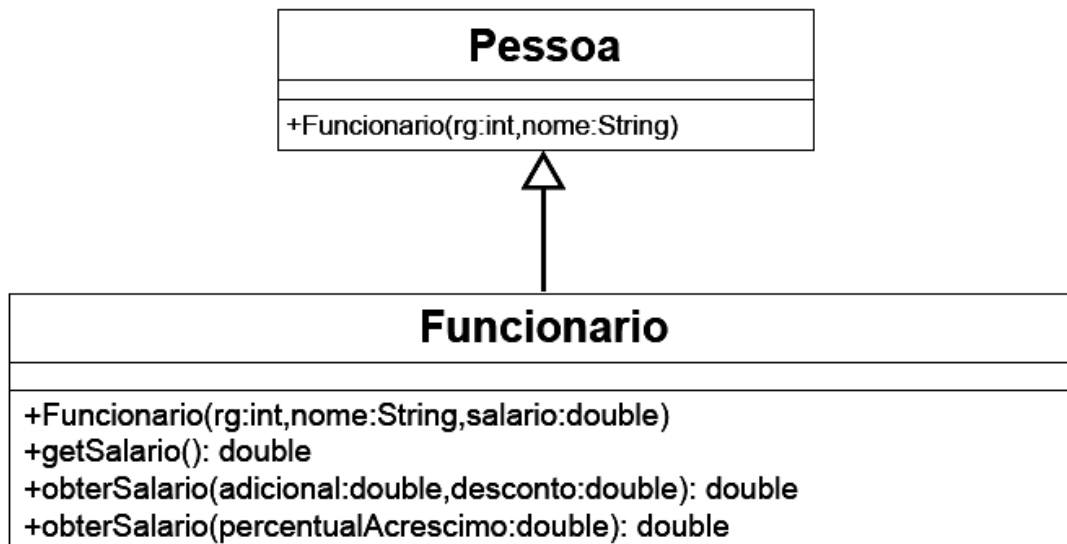
A.



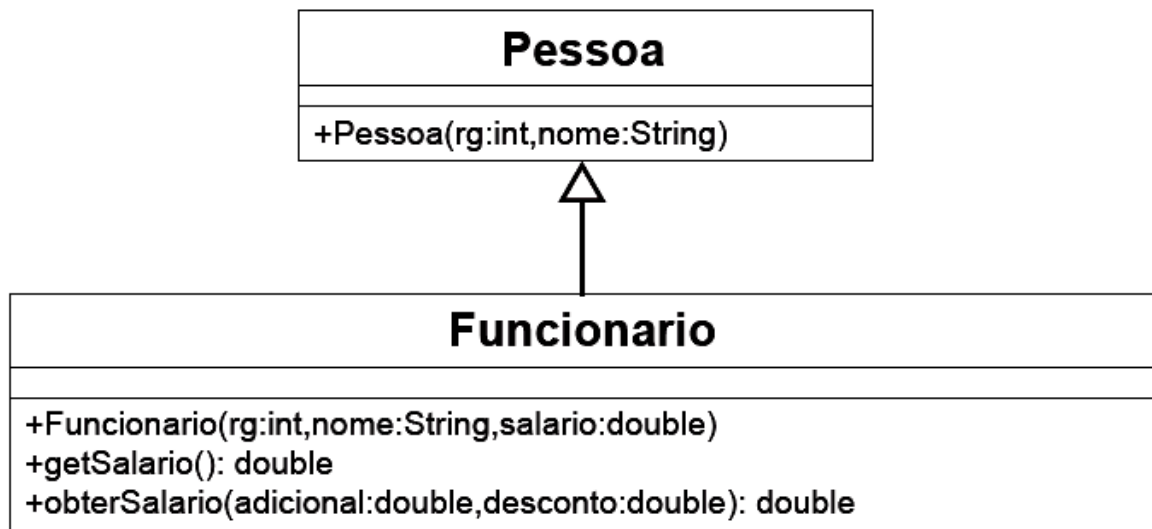
B.



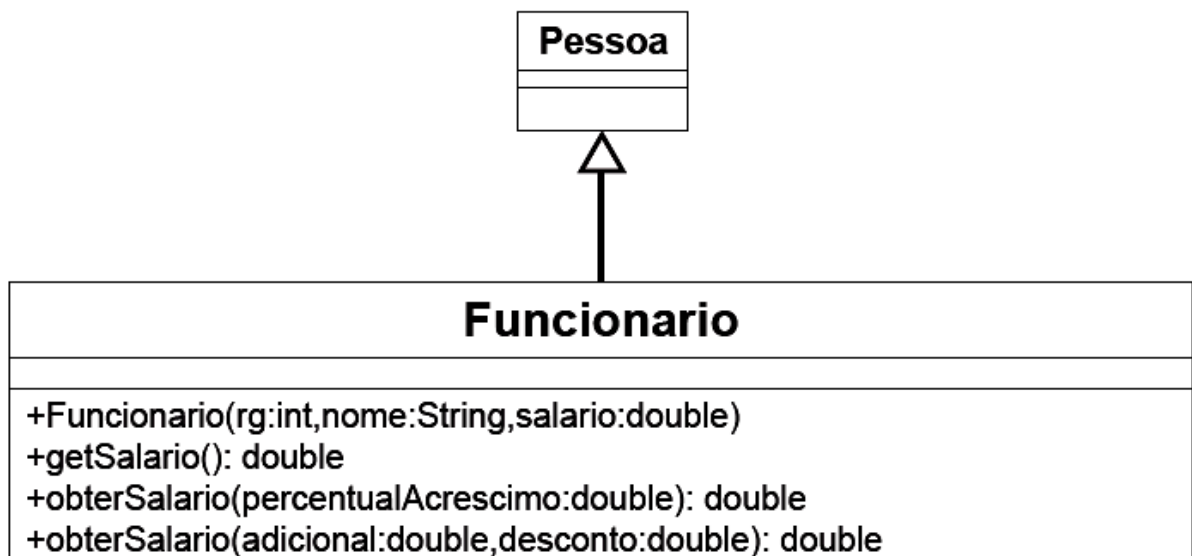
C.



D.



E.



### Questão 7.<sup>7</sup>

Leia o texto a seguir.

*Casos de uso podem ser organizados agrupando-os em pacotes do mesmo modo como são organizadas as classes. Também podem ser organizados pela especificação de relacionamentos de generalização, inclusão e extensão, existentes entre eles.*

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *UML - Guia do Usuário*. Campus, 2006 (com adaptações).

Considerando os relacionamentos existentes entre os casos de uso, avalie as afirmativas a seguir.

- I. Para casos de uso, a generalização significa que o caso de uso filho herda o comportamento e o significado do caso de uso pai e no caso de uso filho deverá acrescentar ou sobrescrever o comportamento de seu pai.
- II. Um relacionamento de inclusão entre casos de uso significa que o caso de uso base incorpora explicitamente o comportamento de outro caso de uso em uma localização especificada. O caso de uso base poderá permanecer isolado, mas, sob certas condições, seu comportamento poderá ser incluído pelo comportamento de outro caso de uso.
- III. Um relacionamento estendido entre casos de uso significa que o caso de uso base incorpora implicitamente o comportamento de outro caso de uso em um local especificado indiretamente pelo caso de uso estendido. O caso de uso estendido nunca permanece isolado, mas é apenas instanciado como parte de alguma base maior que o estende.
- IV. Um relacionamento estendido é utilizado para a modelagem de parte de um caso de uso que o usuário poderá considerar como um comportamento opcional do sistema e para a modelagem de um subfluxo separado, que é executado somente sob determinadas condições.

É correto apenas o que se afirma em

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

---

<sup>7</sup>Questão 17 – Enade 2014.

## **1. Introdução teórica**

### **1.1. UML (Unified Modeling Language)**

De acordo com Booch, Rumbaugh e Jacobson (2006), a UML (Unified Modeling Language) é definida como “uma linguagem-padrão para a elaboração da estrutura de projetos de software”.

Os autores destacam os quatro objetivos básicos da UML, quanto aos artefatos de um sistema complexo de software: visualizar, especificar, construir e documentar.

O principal objetivo da UML é estabelecer um modelo ou representar conceitual e fisicamente um sistema (BOOCH, RUMBAUGH e JACOBSON; 2006).

#### **1.1.1. UML para a visualização**

Um projeto de software envolve mais do que o código-fonte. Além disso, um mesmo problema pode ser resolvido de diferentes formas, com códigos muito distintos. Ainda que programas com códigos diferentes possam atender aos mesmos requisitos, nem todas as soluções são igualmente úteis ou interessantes.

Se dispusermos apenas do código-fonte de um programa, sem nenhum tipo de documentação, o entendimento da solução fica muito mais difícil. Programas comerciais podem ser muito longos e conter milhões de linhas de código-fonte. Dessa forma, é importante ter uma maneira de visualizar o todo, a estrutura geral de um software, sem que, necessariamente, haja a obrigação de se ler o código-fonte de modo completo.

Desenhos e diagramas informais podem ajudar nessa visualização, mas a UML fornece uma linguagem-padrão que permite que diferentes pessoas e empresas façam e interpretem diagramas e modelos de uma mesma forma.

#### **1.1.2. UML para especificação**

Antes de escrevermos o código para um programa, é preciso conhecer quais são as necessidades do cliente. Além disso, grandes projetos envolvem problemas complexos de software.

Muitas empresas contratam profissionais especializados para encontrar a melhor solução, como os arquitetos de software. Esses profissionais precisam de uma linguagem

para especificar, construir e documentar a solução e a UML oferece uma linguagem especificamente desenvolvida para esse propósito.

### **1.1.3. UML para construção**

Ainda que a UML não seja uma linguagem de programação propriamente dita, é possível estabelecer um mapeamento entre os modelos construídos pela UML e o código-fonte em diversas linguagens de programação, como Java ou C# (BOOCH, RUMBAUGH e JACOBSON, 2006).

Essa facilidade do mapeamento vem do alto grau de expressividade aliado à baixa ambiguidade, o que não é o caso quando consideramos outras formas de especificação, como um texto puro. Essa precisão faz com que existam ferramentas capazes de converter alguns diagramas UML em código-fonte com pouca ou nenhuma intervenção do usuário.

### **1.1.4. UML para documentação**

De acordo com Booch, Rumbaugh e Jacobson (2006), um projeto de software envolve mais do que apenas o código-fonte. Empresas de software costumam produzir diversos outros elementos, chamados de artefatos, como:

- requisitos do software;
- documentações da arquitetura;
- projetos e planos de projeto;
- código-fonte;
- testes e planos de teste;
- protótipos.

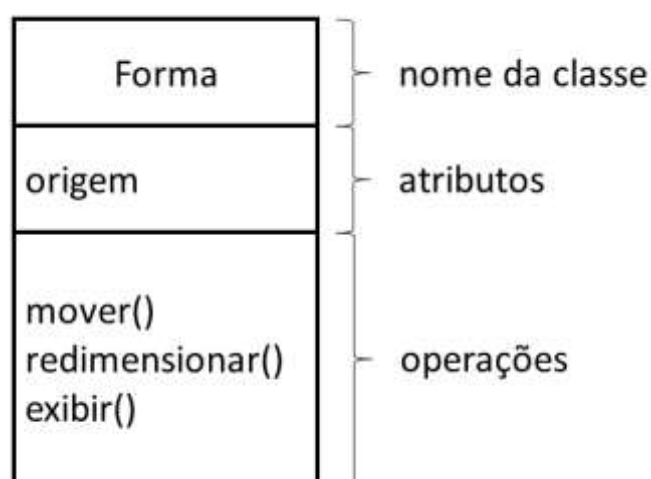
A UML é utilizada para documentar diversos desses artefatos, de modo formal e com baixa ambiguidade.

## **1.2. Diagrama de classes**

Formalmente, define-se uma classe como “uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica” (BOOCH, RUMBAUGH e JACOBSON; 2006).

É importante diferenciar as classes dos objetos: uma classe define um conjunto de objetos, mas não é um objeto. Por exemplo, podemos definir a classe das “xícaras de café” como sendo objetos cerâmicos, ocios, que servem para armazenar uma quantidade pequena de líquido quente. Essa classe abrange todas as possíveis xícaras, de diferentes formas e tamanhos. Se tomarmos um exemplo específico, como uma xícara de restaurante, dizemos que essa xícara é um objeto, ou uma instância específica de uma classe. Utilizamos uma classe para definir uma categoria de objetos similares.

A UML apresenta um diagrama específico para a representação das classes, chamado de diagrama de classes. Nesse diagrama, as classes são representadas por retângulos contendo nomes, atributos e operações, como ilustrado na figura 1. Os atributos representam propriedades associadas a cada um dos objetos que serão instanciados pela classe. Por exemplo, a classe “Cliente” pode ter um atributo chamado de “nome”. Outros atributos que essas classes podem ter são: “endereço”, “telefone” e “email”.



**Figura 1.** Representação de uma classe na UML.

**Fonte.** BOOCH, RUMBAUGH e JACOBSON, 2006 (com adaptações).

Além de atributos, uma classe também apresenta comportamentos associados. Por exemplo, suponha um editor de imagens que tenha uma funcionalidade de desenho de formas geométricas, como losangos. Essa funcionalidade pode ser implementada por meio de uma classe, chamada de “Losango”, que apresenta uma série de comportamentos associados, como a movimentação do losango e a alteração do seu tamanho, entre outras. Essas operações são representadas no retângulo da classe, logo abaixo dos atributos.

As operações têm um nome e costumam terminar com parênteses “()”: elas são similares a “funções” e podem receber parâmetros. Os parâmetros são representados nos parênteses e separados por vírgulas (no caso da existência de mais de um parâmetro).

As diversas classes que compõem um programa podem estar relacionadas entre si e existem, essencialmente, três tipos de relacionamentos entre classes:

- dependência;
- generalização;
- associação.

Utilizamos uma relação de dependência quando queremos dizer que uma classe necessita de outra classe para o seu funcionamento. Por exemplo, suponhamos uma classe chamada “Cachorro” e uma classe chamada “Animal”. Podemos dizer que “Animal” representa uma generalização de “Cachorro”, uma vez que um cachorro é um animal, mas nem todo animal é um cachorro. Finalmente, uma associação é um “relacionamento semântico entre dois elementos do modelo” (PENDER, 2004). Uma associação estabelece o motivo pelo qual duas classes relacionam-se e estabelece as regras que controlam esse relacionamento.

### 1.3. Casos de uso

Segundo Booch, Rumbaugh e Jacobson (2006), um caso de uso

*especifica o comportamento de um sistema ou de parte de um sistema e é uma descrição de um conjunto de sequências de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.*

Desenvolvedores de software precisam compreender as necessidades dos usuários e comunicá-las, sem precisar entrar em detalhes da implementação. A elaboração de casos de uso facilita esse processo de comunicação entre os diversos envolvidos em um projeto de software.

## 2. Análise das alternativas e das afirmativas

### Questão 6.

A - Alternativa correta.

JUSTIFICATIVA. O trecho de diagrama de classes apresentado está correto. Nele, a classe Pessoa é uma generalização da classe Funcionário.



B - Alternativa incorreta.

JUSTIFICATIVA. O diagrama está incorreto, pois as classes Pessoa e Funcionário estão representadas com relacionamento de associação. Na realidade, essas duas classes deveriam apresentar relacionamento de generalização.

C - Alternativa incorreta.

JUSTIFICATIVA. O construtor da classe Pessoa está representado de forma errada. Deveria ser: +Pessoa(rg:int, nome:String), e não +Funcionário(re:int, nome:String).

D - Alternativa incorreta.

JUSTIFICATIVA. Falta um método na classe Funcionário.

E - Alternativa incorreta.

JUSTIFICATIVA. A classe Pessoa do diagrama não apresenta o seu construtor.

Alternativa correta: A.

### **Questão 7.**

I - Afirmativa correta.

JUSTIFICATIVA. A generalização nos casos de uso tem papel similar à generalização nos diagramas de classes (BOOCH, RUMBAUGH e JACOBSON; 2006).

II - Afirmativa incorreta.

JUSTIFICATIVA. Como é o caso de uso-base que incorpora o incluído, o relacionamento incluído não pode ser isolado. Além disso, esse relacionamento tem seu comportamento incluído (por isso o nome) e, também, não é o caso de uso-base.

III - Afirmativa incorreta.

JUSTIFICATIVA. A descrição não corresponde a um relacionamento de extensão, pois o caso de uso-base pode permanecer isolado.

IV - Afirmativa correta.

JUSTIFICATIVA. O relacionamento de extensão é utilizado justamente para a modelagem de comportamentos opcionais, que podem ou não ocorrer em determinadas condições.

Alternativa correta: B.

### **3. Indicações bibliográficas**

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.; *UML: Guia do usuário*. Rio de Janeiro: Campus, 2006.
- PENDER, T. *UML: a Bíblia*. Rio de Janeiro: Campus, 2004.

## Questão 8

### Questão 8.<sup>8</sup>

José discutindo futebol com João declarou que se o time X venceu o último jogo, então o time Y é campeão. João depois de algumas horas descobriu que a declaração era falsa. Em consequência, teve certeza de que é verdade que

- A. o time X não venceu o último jogo e o time Y é campeão.
- B. o time X venceu o último jogo e o time Y não é campeão.
- C. ou o time X não venceu o último jogo ou o time Y é campeão.
- D. o time X não venceu o último jogo e o time Y não é campeão.
- E. se o time Y não é campeão, então o time X não venceu o último jogo.

## 1. Introdução teórica

### 1.1. Proposições

Segundo de Souza (2008), uma proposição “é uma sentença declarativa que pode ser interpretada como verdadeira ou falsa”. O autor ainda adiciona que uma proposição não deve ser ambígua e não deve permitir mais de uma única interpretação.

De acordo com Alencar Filho (2002), proposição é “todo o conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo”.

Essencialmente, dizemos que uma proposição é uma frase para a qual podemos atribuir um valor único de verdadeiro ou falso. Temos, a seguir, alguns exemplos de proposições.

- Júpiter é um planeta do sistema solar (proposição verdadeira).
- A Terra é um planeta do sistema solar (proposição verdadeira).
- A Lua é um planeta (proposição falsa).

### 1.2. Proposição condicional

Segundo Alencar Filho (2002), a proposição condicional é representada por **se p então q**, cujo valor lógico é falso (F) no caso em que p é verdadeira e q é falsa e verdadeiro (V) nos demais casos.

---

<sup>8</sup>Questão 35 – Enade 2014.

Costuma-se expressar a proposição condicional pelo símbolo  $\rightarrow$ , com a tabela verdade ilustrada na tabela 1.

**Tabela 1.** Tabela verdade para a proposição condicional.

<b>p</b>	<b>q</b>	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

## 2. Resolução da questão

A questão pode ser enunciada no formato  $p \rightarrow q$ , sendo p a afirmação sobre o time X e q a afirmação sobre o time Y.

Se observarmos a segunda linha da tabela 1, que é a tabela verdade para a proposição condicional, ela só é falsa quando p for verdadeira e q for falsa. Assim, podemos afirmar que o time X venceu o último jogo e o time Y não é campeão.

Alternativa correta: B.

## 3. Indicações bibliográficas

- ALENCAR FILHO, E. *Iniciação à lógica matemática*. São Paulo: Nobel, 2002.
- DE SOUZA, J. N. *Lógica para a Ciência da Computação*. Rio de Janeiro: Elsevier Brasil, 2008.

## Questão 9

### Questão 9.<sup>9</sup>

Leia o texto a seguir.

*Vivemos em um mundo de produtos de alta tecnologia e praticamente todos requerem interação humana. Para que um produto de software seja bem-sucedido, deve apresentar boa usabilidade. Se os mecanismos de interface tiverem sido bem projetados, o usuário flui suavemente através da interação usando um ritmo cadenciado que permite que o trabalho seja realizado sem grandes esforços. Entretanto, se a interface for mal concebida, o usuário se move aos trancos e barrancos, e o resultado será frustração e baixa eficiência no trabalho.*

*Três regras de ouro são a base para um conjunto de princípios para o projeto de interfaces do usuário:*

- 1 - Deixar o usuário no comando;*
- 2 - Reduzir a carga de memória do usuário;*
- 3 - Tornar a interface consistente.*

PRESSMAN, R.S. *Engenharia de Software: uma abordagem profissional*. 7. ed. McGraw Hill, 2011, p.287-288 (com adaptações).

Com base nessas três regras, avalie as afirmativas a seguir.

- I. Um sistema que permita ao usuário desfazer qualquer ação, respeita a regra de ouro 1.
- II. Um sistema de pagamento de contas que usa uma imagem de um cartão de crédito para orientar o usuário pelo processo de pagamento de uma conta, respeita a regra de ouro 2.
- III. Um conjunto de aplicações ou produtos que implementam as mesmas regras de projeto de modo padronizado, respeita a regra de ouro 3.

É correto o que se afirma em

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

## 1. Introdução teórica

### Usabilidade

Segundo Koscianski e Soares (2007), usabilidade corresponde à “facilidade” ou à “dificuldade” na utilização de determinado sistema por um usuário. Um sistema empregado por cientistas tem provavelmente um critério de usabilidade diferente de um sistema utilizado por artistas ou profissionais de outras áreas.

Algumas normas foram desenvolvidas para auxiliar no processo de avaliação da usabilidade de um programa. A Norma ISO 9241 provê “requisitos, recomendações e

---

<sup>9</sup>Questão 32 – Enade 2014.

princípios para as atividades de projeto centradas em pessoas ao longo do ciclo de vida de sistemas interativos baseados em computador” (ISO 9241-210, 2010). Essa norma refere-se aos requisitos ergonômicos do software.

Outra norma importante para a área é a Norma ISO 9126, reorganizada pela Norma ISO/IEC 25000. A Norma ISO 9126 é constituída por várias partes, conforme tabela 1.

**Tabela 1.** Norma ISO-9126 e suas partes.

<b>Norma</b>	<b>Conteúdo</b>
9126-1	Modelo de qualidade de software
9126-2	Métricas externas
9126-3	Métricas internas
9126-4	Métricas para qualidade em uso

**Fonte.** KOSCIANSKI e SOARES, 2007 (com adaptações).

Adicionalmente, a Norma ISO-14598, que também foi incorporada à norma ISO/IEC 25010:2011, define processos de avaliação e documentação. Ela é conhecida como SQuaRE (*Systems and software Quality Requirements and Evaluation*).

Essas normas definem uma série de subcaracterísticas da usabilidade, como:

- operabilidade;
- compreensibilidade;
- apreensibilidade;
- atratividade.

A operabilidade está ligada à forma como o software pode ser controlado pelo usuário. A compreensibilidade refere-se à capacidade do usuário em compreender se o programa é adequado ou não às suas tarefas. A apreensibilidade aplica-se à facilidade do usuário em aprender a utilizar o programa. Finalmente, a atratividade está vinculada à capacidade de o usuário manter a atenção na utilização do programa (KOSCIANSKI e SOARES, 2007).

## 2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. Sem oportunidade de desfazer ações, um usuário pode sentir apreensão ao utilizar o sistema e ter a sensação de que tem pouco controle sobre o funcionamento do software.

II – Afirmativa correta.

JUSTIFICATIVA. A imagem de um cartão remete facilmente à função desejada. Dessa forma, o usuário não precisa memorizar nenhum comando ou nome especial, de acordo com a regra 2.

III – Afirmativa correta.

JUSTIFICATIVA. A consistência entre as diversas telas e funcionalidades é fundamental. Durante o uso de um programa, o usuário tende a memorizar a forma como o sistema funciona, de modo que a falta de padrão entre diferentes telas e funcionalidades pode desorientar o usuário.

Alternativa correta: E.

### **3. Indicações bibliográficas**

- KOSCIANSKI, A.; SOARES, M. *Qualidade de software*. 2. ed. São Paulo: Novatec, 2007.
- NORMA ISO 9241-210:2010. *Ergonomics of human-system interaction*. Disponível em <<https://www.iso.org/standard/52075.html>>. Acesso em 10 ago. 2017.

## ÍNDICE REMISSIVO

<b>Questão 1</b>	Riscos. Gerenciamento de riscos. Gerenciamento de projetos.
<b>Questão 2</b>	Legislação: Marco Civil da Internet.
<b>Questão 3</b>	Engenharia de requisitos. Requisitos não funcionais. Engenharia de software.
<b>Questão 4</b>	Engenharia de requisitos. Requisitos não funcionais. Engenharia de software.
<b>Questão 5</b>	Arquitetura de computadores. Barramentos. Endereçamento.
<b>Questão 6</b>	UML. Engenharia de software. Diagrama de classes.
<b>Questão 7</b>	UML. Engenharia de software. Casos de uso.
<b>Questão 8</b>	Lógica matemática. Proposições. Tabela verdade.
<b>Questão 9</b>	Usabilidade. Qualidade de software. Engenharia de software.