

-- CONHECIMENTOS ESPECÍFICOS --**BLOCO I**

```
import java.util.Scanner;
public class ExemploTGS {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try {
            System.out.print("Digite          um          número
inteiro...");
            int numero1 = s.nextInt();
            System.out.print("Digite          outro          número
inteiro...");
            int numero2 = s.nextInt();
            System.out.println("A soma dos dois números = "
+ (numero1+numero2));
        } throw (Exception ex) {
            System.out.println("ERRO - Valor digitado não é
um número inteiro!");
        }
    }
}
```

Com base no código precedente, escrito em Java, julgue o próximo item.

- 31** Se a entrada realizada pelo usuário nas variáveis `numero1` ou `numero2` não for um valor inteiro, então, de acordo com a sintaxe e a lógica do programa para tratar a exceção de erro, será mostrada a mensagem ERRO - Valor digitado não é um número inteiro!.

```
class ClassTGS
{
    public void displayMethod(String msg){
        System.out.println(msg);
    }
}
class NewClass @Override ClassTGS
{
    public void displayMethod(String msg){
        System.out.println("Message is: "+ msg);
    }
    public static void main(String args[]){
        NewClass obj = new NewClass();
        obj.displayMethod("TGS!!");
    }
}
```

Tendo como referência o programa precedente, escrito em Java, julgue o item a seguir, relativo à programação orientada a objetos.

- 32** A sintaxe e a lógica do programa apresentado mostram corretamente uma forma de polimorfismo do tipo *override*.

```
import { Component } from '@angular/core';

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
})
export class AppComponent {
    document.write ('TBG!');
}
```

Em relação ao código precedente, julgue o item seguinte.

- 33** Considerando-se que o código apresentado é voltado para o Angular com Typescript, é correto afirmar que `@Component` é um método da classe de `AppComponent`.

```
package com.example.springboot;

import
org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @GetMapping("/")
    public String index() {
        return "Greetings from Spring Boot!";
    }
}
```

A partir do código precedente, julgue o item subsecutivo, relativo a Spring.

- 34** No Spring, a `RestController` é usada para marcar a classe como um controlador em que cada método retorna um objeto de domínio em vez de uma exibição.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 XPTO
6 </style>
7 </head>
8 <body>
9 <h1>TBG</h1>
10 <div class="tbg">
11 Nossa missão é prover as melhores soluções
logísticas conectando as ofertas de gás natural
aos mercados, agregando valor em benefício do
progresso sustentável da indústria, da geração
termelétrica e do aquecimento dos lares.
12 </div>
13 </body>
14 </html>
```

Considerando o código precedente, julgue o item subsequente.

- 35** Para se apresentar corretamente o texto presente na linha 11 em 3 colunas, o comando `.tbg { column-count: 3;}` deve ser inserido no lugar dos caracteres XPTO, na linha 5.

Espaço livre

```
<!DOCTYPE html>
<html>
<body>
<p id="tbg" style="display:none">Transportadora
Brasileira Gasoduto Bolívia-Brasil S.A.
(TBG)</p>
<button type="button"
onclick="document.getElementById('tbg').style.di
splay='block'">Mostre</button>
</body>
</html>
```

A partir do código JavaScript precedente, julgue o item a seguir.

- 36** Ao ser executado, o código inicialmente irá mostrar na tela apenas um botão similar ao mostrado a seguir.

Se esse botão for clicado, será apresentado um resultado similar ao que se segue.

Transportadora Brasileira Gasoduto Bolívia-Brasil S.A. (TBG)

Data Access Layer	
Java Persistence API	Hibernate Native API
JDBC	
Relational Database Hibernate	

Com base na hierarquia precedente, julgue o item seguinte, a respeito de Hibernate.

- 37** Considerado o conceito da solução ORM Hibernate, conclui-se que a hierarquia apresentada anteriormente está errada. O correto seria a hierarquia que se segue, uma vez que o Hibernate não se incorpora ao banco de dados relacional, mas se encontra entre a camada de acesso a dados do aplicativo Java e o banco de dados relacional.

Data Access Layer	
Java Persistence API	Hibernate Native API
Hibernate	
JDBC	
Relational Database	

Julgue os seguintes itens, relativos a DevOps, Jenkins e GIT.

- 38** No DevOps, por meio das práticas de CI/CD (*pipeline* de integração e implantação contínuas), é possível que equipes de operação e desenvolvimento disponibilizem aplicações para os clientes por meio de microsserviços, usando práticas de métodos ágeis.
- 39** A principal função do Jenkins em uma automação CI/CD (*continuous integration/continuous delivery*) é revisar o código com base no *clean code* à procura de *bugs*, falhas de vulnerabilidade ou problemas de manutenção que tornem seu código confuso e difícil de manter.
- 40** No ambiente GIT, uma *branch* é definida como uma coleção de referências junto com um banco de dados de objetos que contém todos os objetos que são acessíveis a partir das referências dos “ramos” do desenvolvimento.

idFuncionario	Nome	Salario
1	André	1000.00
2	João	2000.00
3	Maria	9000.00
4	Pedro	3000.00
5	Thiago	7000.00
6	Matheus	5000.00

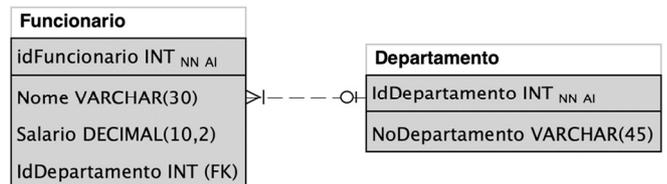
Considerando que a tabela precedente, de nome *funcionario*, possui apenas os registros mostrados, julgue o item subsequente.

- 41** A seguir, é apresentado o comando correto em SQL para se alterar o valor do salário (*Salario*) para o maior salário da tabela apenas para o funcionário que possui *IdFuncionario* igual a 1.

```
UPDATE funcionario AS f, (SELECT max(salario)
as maxsalario FROM funcionario) AS s
SET f.salario = s.maxsalario
WHERE f.idfuncionario = 1;
```

Julgue os próximos itens, relativos a SQL, banco de dados e Azure.

- 42** No Azure DevOps, é possível implementar a entrega contínua do *software* para registros de contêiner como o Docker Hub e o registro de contêiner do Azure, não sendo possível, nesses casos, a utilização do Kubernetes.
- 43** No modelo entidade-relacionamento a seguir, a coluna *IdDepartamento* da tabela *Funcionario* não pode ser nula, uma vez que ela é a chave estrangeira advinda do relacionamento com a tabela *Departamento*.



```
programa{
funcao inicio(){
real num, resultado=0.0, total = 0.0
inteiro cont = 0
faca {
cont = cont +1
escreva("\nDigite o número : ")
leia(num)
total = total + num
} enquanto(cont < 2 )
resultado = total / cont -1
escreva("Resultado: ", resultado)
}
}
```

Considerando o algoritmo precedente, julgue o item subsequente.

- 44** A seguir, é corretamente mostrado o resultado da execução do algoritmo caso o usuário entre com os números 9 e 5.

```
Digite o número : 9
Digite o número : 5
Resultado: 7.0
```

```

programa {
  funcao inicio() {
    inteiro numero, cont=1
    logico primo = verdadeiro
    escreva("Informe um número inteiro acima de 3: ")
  }
  leia(numero)
  faca
  {
    cont = cont +1
    se (numero % cont == 0)
    { primo = falso
    cont=numero*3
    }
  } enquanto( cont < numero/2)

  se (primo == verdadeiro) {
    escreva("O número ", numero, " é primo")
  }
  senao{
    escreva("O número ", numero, " não é primo")
  }
}

```

Com base no algoritmo precedente, julgue o item a seguir.

- 45** O algoritmo testa se um número é primo (ou seja, que só possa ser dividido por 1 e por ele mesmo), de modo que, se o usuário entrar com o número 7, por exemplo, o resultado será o mostrado a seguir.

```

Informe um número inteiro acima de 3: 7
O número 7 é primo

```

P1: "Se o custo operacional aumenta, aumentamos o preço cobrado pelo nosso produto."

P2: "Se o preço do produto substituto aumenta, aumentamos o preço cobrado pelo nosso produto."

P3: "Aumentamos o preço cobrado pelo nosso produto, mas o custo operacional não aumentou."

C: "Logo, o preço do produto substituto aumentou."

Considerando o argumento constituído pelas premissas P1, P2 e P3 e pela conclusão C, anteriormente apresentadas, julgue os itens seguintes.

- 46** Na análise do argumento, as premissas P1 e P2 podem ser resumidas à seguinte proposição: Se o custo operacional ou o preço do produto substituto aumenta, aumentamos o preço cobrado pelo nosso produto.
- 47** A tabela-verdade da proposição $P1 \wedge P2$ contém mais de dez linhas.
- 48** O argumento constituído pelas premissas P1, P2 e P3 e pela conclusão C é válido.
- 49** A proposição P3 pode ser vista como uma negação da proposição P1.
- 50** A proposição P2 pode ser corretamente reescrita como: O preço do produto substituto não aumenta ou aumentamos o preço cobrado pelo nosso produto.
- 51** No caso em que é falsa a proposição "o preço do produto substituto aumenta", a proposição P2 será verdadeira independentemente do valor lógico de seu consequente.

"Avisei em 1984, mas vocês não me ouviram", disse James Cameron, diretor do filme **O Exterminador do Futuro**, acerca dos riscos do avanço descontrolado da inteligência artificial.

Acerca da afirmação precedente, do diretor James Cameron, julgue os itens subsequentes.

- 52** Sob o ponto de vista lógico, a afirmação de Cameron é equivalente à afirmação "Avisei em 1984 e vocês não me ouviram".
- 53** A afirmação de James Cameron é a negação de "Se avisei em 1984, então vocês me ouviram".
- 54** Há apenas uma hipótese de falsidade para a afirmação de James Cameron.
- 55** A negação da afirmação de Cameron pode ser escrita como "Não avisei em 1984 e vocês não me ouviram".

Julgue os próximos itens, relativos a técnicas de desenvolvimento seguro.

- 56** Ao contrário do teste de segurança estático (SAST) e do teste de segurança dinâmico (DAST), o teste interativo de segurança (IAST) opera dentro do aplicativo, além de permitir uma saída mais precisa.
- 57** O teste de segurança estático (SAST) trabalha diretamente com o código e é empregado de forma complementar ao teste de segurança dinâmico (DAST).

Com base nos princípios e modelos de DEVOPS, julgue os itens subsequentes.

- 58** A evolução natural da entrega contínua (*continuous delivery*) é a implantação contínua (*continuous deployment*), que consiste no *deploy* automático em produção após a execução com sucesso dos testes automatizados e das validações previstas.
- 59** O uso da integração contínua (*continuous integration*) é um requisito fundamental para o DEVOPS e ocorre quando existe o interesse de expandir os benefícios da automação dos testes e do *feedback* imediato para os próximos estágios que não são cobertos pela entrega contínua (*continuous delivery*).

Julgue os itens seguintes, a respeito de contêineres e virtualização.

- 60** Contêineres são similares a máquinas virtuais, mas estas são mais leves e mais integradas ao sistema operacional da máquina *host*.
- 61** Ao contrário das máquinas virtuais, os contêineres isolam, tanto do sistema operacional quanto de outros contêineres, um único aplicativo e suas dependências; assim, nenhum contêiner atrapalha o funcionamento de outro.
- 62** Os contêineres compartilham o mesmo *kernel* do sistema operacional e isolam os processos da aplicação do restante do sistema, dificultando, assim, a migração e a abertura de seus componentes para utilização em desenvolvimento, testes e produção.

Julgue os itens subsequentes, referentes a contêineres.

- 63** Em sua utilização, cada contêiner passa por um rodízio de responsabilidades, que integra todos os processos da aplicação.
- 64** O uso de contêineres torna as aplicações portáteis de modo que o mesmo código possa ser executado em qualquer dispositivo; a máquina virtual, por sua vez, é uma cópia digital de uma máquina física.
- 65** Um contêiner particiona o *hardware* físico em vários ambientes, virtualizando o sistema operacional.
- 66** Contêineres incluem apenas *software* de alto nível, são rápidos para modificar e fazer iterações e fornecem controle granular da escalabilidade da aplicação por permitirem o uso de microsserviços.

Com base nos conceitos de microsserviços, julgue os itens que se seguem.

- 67** Na arquitetura baseada em microsserviços, a aplicação é estruturada em um único componente para implantação, e a troca de informações entre os módulos internos do sistema ocorre dentro de um mesmo processo, o que torna a comunicação rápida e eficiente.
- 68** Em infraestrutura como código, a abordagem declarativa define os comandos específicos necessários para o alcance da configuração desejada.
- 69** Microsserviços é uma maneira de se desenvolver uma aplicação facilmente escalável, desacoplada e independente; e cada serviço pode ser escrito em uma linguagem de programação diferente.
- 70** Em infraestrutura como código, o modelo controlado por interface de programação de aplicativos (API) da nuvem permite que os desenvolvedores interajam com a infraestrutura de modo programático e em escala, em vez de precisarem instalar e configurar manualmente os recursos.

BLOCO II

A respeito de UX, julgue os itens a seguir.

- 71** *Moodboard* corresponde a uma coleção de imagens e itens que podem ser usados para construir o estilo visual do produto em desenvolvimento.
- 72** O *wireframe* de um projeto deve apresentar o leiaute final e todos os casos de uso do produto.
- 73** Um dos objetivos do uso de *personas* é criar empatia com os usuários do produto ou serviço que está sendo desenvolvido.

Acerca de testes de *software*, julgue os próximos itens.

- 74** O teste de regressão deve ser efetuado para garantir que novos componentes não tenham causado problema nas funções que antes funcionavam corretamente.
- 75** Os testes dos tipos alfa e beta são executados em um ambiente controlado e com a presença de, pelo menos, um desenvolvedor.

No que se refere à arquitetura de soluções, julgue os itens subsequentes.

- 76** Para definir se os módulos de um sistema foram bem divididos, é utilizado o conceito de acoplamento, que mede a interdependência entre esses módulos.
- 77** Na coesão funcional, as tarefas a serem executadas são agrupadas de acordo com a sequência de operações.
- 78** O *design* detalhado da arquitetura descreve de forma ampla como um *software* é decomposto e especifica os módulos e as suas relações.

No que diz respeito à gestão de requisitos de negócio, julgue os itens seguintes.

- 79** Para fins de modelagem de requisitos, o cliente e o usuário final são a mesma pessoa, com os mesmos objetivos e características.
- 80** Requisito funcional especifica o que o *software* deve fazer, enquanto requisito não funcional especifica como o sistema deve se comportar.

A respeito de qualidade de *software*, julgue os seguintes itens.

- 81** Em um projeto de sistema, o analista de QA (*quality assurance*) é responsável por preparar e executar o plano de testes, enquanto a equipe de desenvolvimento é responsável por documentar os desvios nos padrões vigentes.
- 82** O princípio de conformidade define se o *software* está de acordo, entre outros itens, com as convenções predefinidas do projeto.

A respeito de *design thinking*, julgue os próximos itens.

- 83** A etapa de imersão é dividida em duas fases: a preliminar e a profunda; na imersão profunda, é feito o detalhamento das especificidades do problema.
- 84** A técnica narrativa de *storyboard* permite visualizar uma ideia durante o seu desenvolvimento temporal.
- 85** *Design thinking* consiste em uma técnica linear, em que todas as etapas devem ser seguidas, sendo cada uma a continuidade da etapa anterior.

BLOCO III

Acerca da nomeação na área de exploração e transporte de gás natural, julgue os itens seguintes.

- 86** A nomeação de uma área deve ser feita por pessoa jurídica exclusivamente constituída sob leis brasileiras.
- 87** A nomeação de área para exploração e transporte de gás natural possui caráter confidencial, devendo ser indicada a bacia sedimentar em que se encontra a área.
- 88** A nomeação de área incluída em processo de oferta permanente para exploração e transporte não gera revisão na geometria do bloco exploratório ou da área com acumulações marginais.
- 89** A nomeação de área, caso a área nominada venha a ser licitada, não gera qualquer direito ou dever para a pessoa jurídica responsável.
- 90** A nomeação de área obriga a Agência Nacional do Petróleo, Gás Natural e Biocombustíveis a ofertá-la em rodada de licitação.

Com relação ao transporte do gás natural, julgue os itens a seguir.

- 91** O gás natural liquefeito é aquele submetido a processo de liquefação para acondicionamento e transporte.
- 92** Ponto de entrega ou ponto de saída, nos gasodutos de transporte, é o ponto no qual o gás natural é entregue ao transportador pelo carregador ou por quem este venha a indicar.
- 93** A certificação de independência do transportador é o procedimento para a verificação do enquadramento do transportador nos requisitos de independência e autonomia, consoante regulação dos órgãos estaduais.
- 94** O gasoduto de transferência é aquele destinado à movimentação de gás natural, considerado de interesse próprio e restrito de seu proprietário, com início e término em suas próprias instalações.
- 95** O código comum de rede é a inscrição dada aos equipamentos destinados à exploração e ao transporte do gás natural, os quais são acessados por vários transportadores.

Julgue os itens subsecutivos, a respeito dos processos operacionais na área de transporte de gás natural.

- 96** O plano de contingência de transporte deve estabelecer as regras de atuação dos agentes envolvidos na operação, o protocolo de comunicação e a prioridade de atendimento das demandas.
- 97** O carregador é o agente que utiliza o serviço de transporte de gás natural em gasoduto de transporte, por meio da autorização dos órgãos reguladores estaduais.
- 98** A programação logística representa a programação operacional realizada pelo transportador, em atendimento às demandas dos carregadores, de acordo com o estabelecido nos contratos de serviço de transporte.
- 99** A capacidade de transporte é o volume máximo diário de gás natural que o transportador pode movimentar nos pontos de entrada ou de saída de um gasoduto.
- 100** O gasoduto de escoamento da produção é o conjunto de instalações destinadas à movimentação de gás natural produzido, que se encontra antes do sistema de medição, com a finalidade de alcançar as instalações em que será processado.

Espaço livre
