



COMISSÃO DE VALORES MOBILIÁRIOS

TARDE

ANALISTA CVM - PERFIL 8 - TI - SISTEMAS E DESENVOLVIMENTO

PROVA OBJETIVA DE CONHECIMENTOS ESPECÍFICOS – NÍVEL SUPERIOR
TIPO 2 – VERDE



SUA PROVA

Além deste caderno de provas contendo setenta questões objetivas, você receberá do fiscal de sala:

- uma folha para a marcação das respostas das questões objetivas



TEMPO

- **4 horas** é o período disponível para a realização da prova, já incluído o tempo para a marcação da folha de respostas da prova objetiva
- **3 horas** após o início da prova é possível retirar-se da sala, sem levar o caderno de provas
- **30 minutos** antes do término do período de prova é possível retirar-se da sala **levando o caderno de provas**



NÃO SERÁ PERMITIDO

- Qualquer tipo de comunicação entre os candidatos durante a aplicação da prova
- Usar o sanitário ao término da prova, após deixar a sala



INFORMAÇÕES GERAIS

- As questões objetivas têm cinco alternativas de resposta (A, B, C, D, E) e somente uma delas está correta
- Verifique se este caderno de provas está completo, sem repetição de questões ou falhas. Caso contrário, notifique imediatamente o fiscal da sala, para que sejam tomadas as devidas providências
- Na folha de respostas da prova objetiva, confira seus dados pessoais, especialmente nome, número de inscrição e documento de identidade, e leia atentamente as instruções para preenchimento
- Use somente caneta esferográfica, fabricada em material transparente, com tinta preta ou azul
- Assine seu nome apenas no(s) espaço(s) reservado(s)
- Confira o cargo, a cor e o tipo do seu caderno de provas. Caso tenha recebido caderno de cargo, cor ou tipo diferente do impresso em sua folha de respostas, o fiscal deve ser **obrigatoriamente** informado para o devido registro na ata da sala
- O preenchimento das respostas da prova objetiva é de sua responsabilidade e não será permitida a troca da folha de respostas em caso de erro
- Para fins de avaliação, serão levadas em consideração apenas as marcações realizadas na folha de respostas da prova objetiva, não sendo permitido anotar informações relativas às respostas em qualquer outro meio que não seja o caderno de provas
- Os candidatos serão submetidos ao sistema de detecção de metais quando do ingresso e da saída de sanitários durante a realização das provas

Boa sorte!

PROVA OBJETIVA

CONHECIMENTOS ESPECÍFICOS

Engenharia de Software e Desenvolvimento de Sistemas

1

O cálculo da complexidade computacional é essencial para verificar a viabilidade do algoritmo. Observe o código a seguir, em Python, para o problema da torre de Hanoi.

```
def hanoi(n, o, d, a):
    if n==1:
        print("D1 de "+o+" p/ "+d)
    else:
        hanoi(n-1, o, a, d)
        print("D"+str(n)+" de "+o+" p/ "+d)
        hanoi(n-1, a, d, o)
```

A complexidade desse algoritmo no pior caso é:

- (A) $O(2^n)$;
- (B) $O(n)$;
- (C) $O(n \log n)$;
- (D) $O(n^2)$;
- (E) $O(\log n)$.

2

Pedro adotou o algoritmo apresentado a seguir para ordenar um vetor de inteiros V, com índices variando de 1 até n.

```
Para K de 2 até n faça:
    X ← V[K]
    W ← (K - 1)
    Enquanto W > 0 e V[W] > X faça:
        V[W+1] ← V[W]
        W ← (W-1)
    Fim Enquanto
    V[W+1] ← X
Fim Para
```

O algoritmo utilizado por Pedro foi o:

- (A) Selection Sort;
- (B) Insertion Sort;
- (C) Bubble Sort;
- (D) Merge Sort;
- (E) Quick Sort;

3

A analista Raquel foi designada para refatorar o código de um sistema à luz da Arquitetura Limpa. Raquel iniciou a tarefa assimilando os princípios de design utilizados pela arquitetura. A analista seguiu determinando a camada apropriada da Arquitetura Limpa para cada módulo do sistema. Logo após, Raquel revisou a implementação dos módulos A e B, que encapsulam regras de negócio específicas à aplicação. A analista removeu as referências diretas aos módulos A e B no código da camada interior adjacente, aplicando o princípio de design que garante a não violação da Regra da Dependência ao longo das camadas, durante o fluxo de controle.

Raquel removeu as referências diretas aos módulos A e B da camada de:

- (A) entidades, aplicando a segregação de interface;
- (B) entidades, aplicando a inversão de dependência;
- (C) casos de uso, aplicando a segregação de interface;
- (D) adaptadores, aplicando a segregação de interface;
- (E) adaptadores, aplicando a inversão de dependência.

4

A fase de testes de software em processos ágeis se caracteriza pela elaboração dos testes antes da implementação do código, permitindo a execução do teste enquanto o código está sendo escrito.

A característica do XP que tem como fundamento esse conceito de teste é o:

- (A) desenvolvimento de testes incrementais a partir de cenários;
- (B) envolvimento dos usuários no desenvolvimento de testes e validação;
- (C) desenvolvimento de test-first;
- (D) uso de frameworks de testes automatizados;
- (E) uso de workflows em testes.

5

O analista Ricardo adquiriu de um provedor as soluções de nuvem A, B e C, para a CVM. Os serviços da solução A replicam dados diretamente na solução B. A solução C não possui conectividade com as demais soluções. Todas as soluções atendem exclusivamente à CVM, porém as soluções A e C operam na Internet, enquanto a solução B opera na Intranet.

Em relação ao modelo de implantação da arquitetura de nuvem, as soluções:

- (A) A, B e C são nuvens privadas;
- (B) A, B e C formam uma multicloud;
- (C) A, B e C formam uma nuvem híbrida;
- (D) A e C são nuvens públicas, enquanto B é nuvem privada;
- (E) A e B formam uma nuvem híbrida, enquanto C é nuvem privada.

6

Considere o seguinte trecho de um documento em HyperText Markup Language:

```
<div id="secao1" class="align">
  <span class="item"></span>
  <span class="item"></span>
</div>
```

Considere o respectivo código de *Cascading Style Sheets*:

```
.align {
  width: 100px;
  height: 100px;
  display: flex;
  flex-direction: row-reverse;
}
.item {
  width: 10%;
  height: 10%;
  border-radius: 50%;
  flex-shrink: initial;
  background-color: red;
}
```

Na renderização por um navegador web padrão, a largura total ocupada pelo elemento "secao1" e o conteúdo exibido devem ser, respectivamente:

- (A) 100px e círculos dispostos em linha;
- (B) 20px e círculos dispostos em coluna;
- (C) 20px e semicírculos dispostos em linha;
- (D) 20px e semicírculos dispostos em coluna;
- (E) 100px e semicírculos dispostos em coluna.

7

Roberto está implementando uma solução SOA, com base em uma plataforma específica para orquestração de serviços, e escreveu o comando a seguir, na sintaxe BPEL.

```
<invoke name="InvokeSearchCEP"
  inputVariable="lSearchCEPInput"
  outputVariable="lSearchCEPOutput"
  partnerLink="SearchCEPService"
  portType="ns2:SearchCEPPort"
  operation="SearchCEP"/>
```

O comando escrito por Roberto será utilizado para:

- (A) receber a resposta do serviço de forma assíncrona;
- (B) copiar dados entre variáveis;
- (C) estabelecer a ordem de execução no processo;
- (D) definir uma variável;
- (E) efetuar uma chamada síncrona para o serviço.

8

A analista Letícia realizou a seguinte consulta em *Structured Query Language* (SQL):

```
WITH RECURSIVE anos (n) AS
(
  SELECT 2020 UNION ALL
  SELECT n + 1 FROM anos WHERE n < 2025
)
SELECT * FROM anos
```

Logo após, Letícia realizou outra consulta em SQL:

```
SELECT 2024 FROM anos
```

O sistema gerenciador de banco de dados usado por Letícia suporta o recurso Common Table Expression do SQL.

As consultas efetuadas por Letícia retornaram, respectivamente:

- (A) uma tupla e um erro;
- (B) seis tuplas e um erro;
- (C) um erro e duas tuplas;
- (D) uma tupla e uma tupla;
- (E) seis tuplas e duas tuplas.

9

As metodologias ágeis surgiram com o intuito de oferecer com maior rapidez produtos consistentes e que agregam valor, por meio de entregas parciais em períodos curtos. Em termos de Scrum e XP, existem diversas regras e eventos que objetivam essa otimização de entregas, como:

- (A) as reuniões diárias do Scrum, com duração média de uma hora, onde é analisado o avanço das tarefas na Sprint;
- (B) o uso de programação em pares no XP, aliado ao rodízio de colaboradores durante o desenvolvimento;
- (C) as reuniões de planejamento da Sprint, com duração máxima de quinze minutos, para definir as funcionalidades que serão desenvolvidas pela equipe na Sprint;
- (D) a definição de um prazo médio de dois meses para completar cada Sprint e entregar as funcionalidades previstas;
- (E) a priorização do desenvolvimento frente aos testes no XP, de forma a entregar mais rapidamente os produtos.

10

O analista José precisa escolher entre dois algoritmos, Abusca e Cbusca. José sabe que, sendo N o tamanho da entrada do algoritmo, Abusca requer $2^N + \log_2(N)$ operações para ser executado. Já o Cbusca requer $N^4 + N$ operações para ser executado. José determinou, na notação O -grande, a complexidade de tempo no pior caso para cada algoritmo e, por fim, deve escolher o algoritmo que apresenta a menor ordem de complexidade no pior caso.

José deve escolher o algoritmo:

- (A) Cbusca, que possui complexidade $O(N)$;
- (B) Abusca, que possui complexidade $O(2^N)$;
- (C) Cbusca, que possui complexidade $O(N^4)$;
- (D) Cbusca, que possui complexidade $O(3N)$;
- (E) Abusca, que possui complexidade $O(\log(N))$.

11

O analista João desenvolveu o web service cvmWS, com base no protocolo *Simple Object Access Protocol* (SOAP). A fim de otimizar a serialização de um arquivo binário anexo à mensagem SOAP, João implementou em cvmWS a especificação *SOAP Message Transmission Optimization Mechanism* (MTOM).

Quando a serialização otimizada do MTOM é bem-sucedida, os clientes do cvmWS devem ler o conteúdo do arquivo anexo contido:

- (A) dentro do envelope SOAP, no elemento xop;
- (B) fora do envelope SOAP, no corpo da mensagem;
- (C) dentro do envelope SOAP, no elemento binaryData;
- (D) dentro do envelope SOAP, no elemento base64Binary;
- (E) fora do envelope SOAP, em uma mensagem subsequente.

12

Maurício é o líder técnico do Time de Tecnologia da Informação (TTI) de uma organização que está iniciando o uso do estilo de Desenvolvimento Orientado a Testes (TDD).

De forma a nivelar o conhecimento e obedecendo ao estilo TDD, Maurício orientou que os(as):

- (A) testes comecem por uma operação complexa;
- (B) casos de teste menores dispensem a escrita de asserções;
- (C) testes de recursos complexos, como acesso a dados, sejam especificados como testes de integração;
- (D) coleções de objetos sejam implementadas sem coleções e, após testadas, sejam reescritas para funcionar com coleções;
- (E) os desenvolvedores registrem nos comentários da ferramenta de versionamento os testes que foram criados, já que a execução de um teste afeta a execução de outro.

13

Natália está desenvolvendo um sistema financeiro, no qual os documentos podem receber diferentes tipos de tratamento, e pretende tirar proveito dos padrões de desenvolvimento. Para tratar os documentos, ela criou diversos processos, que serão colocados em uma fila. Assim, o documento passa por um dado processo da fila, é tratado e enviado para o processo seguinte.

Adotando essa estratégia, Natália usou o padrão de desenvolvimento denominado:

- (A) Chain of Responsibility;
- (B) Prototype;
- (C) Decorator;
- (D) Abstract Factory;
- (E) Factory Method.

14

Sônia precisa desenvolver um aplicativo para rastreamento de ambulâncias, em uma rede particular de saúde, tendo como base a arquitetura REST.

Sabendo que o GPS de cada veículo cadastrado irá fornecer o posicionamento a cada 5 segundos, com armazenamento no servidor junto à data e hora, e que a consulta irá alimentar um mapa do Google, os métodos HTTP utilizados por Sônia para a inclusão de coordenadas e para a consulta à base serão, respectivamente:

- (A) DELETE e GET;
- (B) POST e PUT;
- (C) POST e GET;
- (D) GET e DELETE;
- (E) GET e PUT.

15

Considere o seguinte código em Java:

```
import java.util.concurrent.atomic.AtomicInteger;

class MultiSomador extends Thread {
    int n = 1_000;
    int saida = 1;
    AtomicInteger contagem = new AtomicInteger(1);

    public void run() {
        for (short i = 0; i < n; i++)
            saida = contagem.getAndIncrement();
    }
}

public class Principal {
    public static void main(String[] args)
        throws InterruptedException {
        MultiSomador a = new MultiSomador();

        Thread b = new Thread(a, "b");
        Thread c = new Thread(a, "c");

        b.start(); b.join(0);
        c.start(); c.join(0);

        System.out.print(a.contagem+", "+a.saida);
    }
}
```

Ao ser executado em um ambiente Java com suporte à classe de biblioteca AtomicInteger, o código acima imprime na saída padrão:

- (A) 1, 1
- (B) Exception
- (C) 1000, 999
- (D) 2000, 1999
- (E) 2001, 2000

16

Maria está desenvolvendo um aplicativo desktop, com base em um ambiente de janelas, e precisa que alguns processos sejam disponibilizados de forma global no aplicativo, sem a necessidade de instanciar um objeto específico, e de forma que qualquer entidade possa acessar.

Por estar trabalhando dentro da metodologia orientada a objetos, Maria precisará adicionar aos métodos globais os modificadores:

- (A) privado e abstrato;
- (B) protegido e sobrecarregado;
- (C) público e abstrato;
- (D) protegido e sobrescrito;
- (E) público e estático.

17

A aplicação iAPLICA contém uma função de login para validar nome e senha de usuários. O usuário digita um nome de usuário e uma senha, e então clica no botão "login". O iAPLICA valida os dados digitados junto à base de dados e permite ao usuário acessar a aplicação ou fornece uma mensagem informando que os dados digitados estão incorretos. Há um requisito adicional de que, como parte do login, o usuário receba acesso apropriado às funcionalidades do iAPLICA com base na função que lhe foi atribuída. A função do usuário é verificada na base de dados. Considerando a Análise de Pontos de Função (APF), a função de login do iAPLICA é composta por:

- (A) uma Saída Externa;
- (B) uma Consulta Externa;
- (C) duas Consultas Externas;
- (D) uma Saída Externa e uma Entrada Externa;
- (E) uma Consulta Externa e uma Saída Externa.

18

Ao desenvolver um sistema em Java, Eliana decidiu usar os recursos para suporte ao paradigma funcional presentes na plataforma, como no código a seguir, para ordenar a lista.

```
List<String> dados =
    Arrays.asList("Ana", "Luiz", "Carlos");
dados.sort(expressão funcional);
```

Para que o código de Eliana funcione, a expressão funcional utilizada deverá ser:

- (A) lambda a,b : a>b
- (B) (a,b) => a.compareTo(b)
- (C) (a,b) -> a.equals(b)
- (D) lambda a,b: strcmp(a,b)
- (E) (a,b) -> a.compareTo(b)

19

O analista José desenvolveu a aplicação CVMaB observando a arquitetura hexagonal. Para a interação com o usuário, a CVMaB disponibiliza uma interface gráfica de usuário e uma interface de linha de comando, a cvmapb, com ambas as interfaces capazes de realizar as mesmas operações. A aplicação também disponibiliza uma *Application Programming Interface* (API) web RESTful, capaz de realizar parte das operações disponíveis na cvmapb. A CVMaB utiliza o sistema gerenciador de banco de dados da CVM para persistir os dados do usuário. A aplicação possui, ainda, implementações específicas para a publicação de métricas de desempenho. As métricas são remetidas pela CVMaB ao pipeline de processamento de dados em tempo real da CVM.

Com base no enunciado e à luz da arquitetura hexagonal, são identificáveis na CVMaB:

- (A) três portas primárias, duas portas secundárias, um adaptador primário e um adaptador secundário;
- (B) duas portas primárias, duas portas secundárias, um adaptador primário e dois adaptadores secundários;
- (C) duas portas primárias, uma porta secundária, dois adaptadores primários e três adaptadores secundários;
- (D) duas portas primárias, uma porta secundária, três adaptadores primários e dois adaptadores secundários;
- (E) uma porta primária, duas portas secundárias, três adaptadores primários e dois adaptadores secundários.

20

A analista Camila executou o seguinte script em JavaScript:

```
const map1 = new WeakMap();
const e1 = {};

map1.set(e1, e1);
console.log(map1.get(e1))
```

Logo após, Camila executou outro script em JavaScript:

```
const map2 = new WeakMap();
const e2 = "dois";

map2.set(e2, e2);
console.log(map2.get(e2))
```

O interpretador JavaScript usado por Camila suporta a estrutura de dados WeakMap do JavaScript.

Os scripts executados por Camila imprimiram no console, respectivamente:

- (A) TypeError e TypeError;
- (B) TypeError e a string "dois";
- (C) SyntaxError e a string "dois";
- (D) um objeto vazio e TypeError;
- (E) um objeto vazio e a string "dois".

21

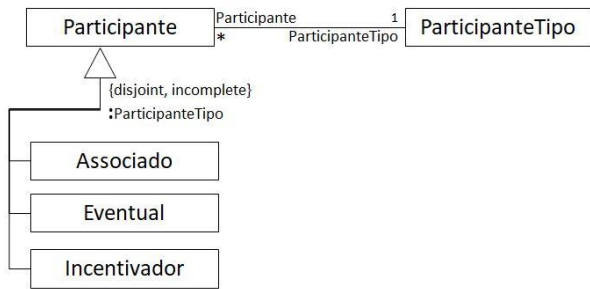
A Equipe de Desenvolvimento de Soluções de Software (EDSS) recebeu a demanda de desenvolvimento de um software complexo e, por isso, pretende utilizar a abordagem *Domain Driven Design* (DDD).

Com foco no modelo de domínio principal, a EDSS assumirá que:

- (A) a lógica da aplicação deve considerar o modo de persistência de objetos nos repositórios;
- (B) as entidades serão definidas pelos atributos que as descrevem;
- (C) os analistas de negócio e de requisitos serão os responsáveis pela definição da Linguagem Ubíqua;
- (D) os objetos do domínio serão modelados com responsabilidades do próprio armazenamento, mas não da própria exibição;
- (E) uma operação deve ser adicionada ao modelo como uma interface autônoma, declarada como um serviço, quando não for uma reponsabilidade natural de uma Entidade ou de um Objeto de Valor.

22

O diagrama abaixo, modelado na linguagem UML 2.5.1, representa um conceito do negócio para o Sistema PARCEIROS utilizado por uma associação civil sem fins lucrativos.



O diagrama acima explicita que:

- (A) a designação *powerType* :ParticipanteTipo no *GeneralizationSet* em Participante especifica uma outra tipificação para as subclasses de Participante;
- (B) cada instância de Associado tem seus próprios atributos e compartilha instâncias com Eventual e Incentivador;
- (C) as instâncias das subclasses de Participante podem corresponder a várias instâncias de ParticipanteTipo simultaneamente;
- (D) as instâncias de ParticipanteTipo têm um relacionamento semântico de realização com as instâncias de Participante;
- (E) o *GeneralizationSet* em Participante tem tipificação forte e está cobrindo semanticamente seus classificadores, que não têm instâncias em comum.

23

O Time de Desenvolvimento de Sistemas (TDS) de uma organização recebeu demandas de alterações para o sistema SuperVALORES. Uma parte interessada solicitou duas mudanças na forma de apresentação de dados nos relatórios emitidos pelo SuperVALORES:

- substituição do formato para datas de "MMDDAAAA" (mês-dia-ano) para "AAAAMMDD" (ano-mês-dia); e
- conversão de moeda de real (R\$) para dólar (US\$).

Com base no SNAP (*Software non-Functional Assessment Process*), as demandas de alteração solicitadas podem ser mensuradas utilizando a categoria:

- (A) arquitetura, subcategoria componentes;
- (B) operações de dados, subcategoria formatação de dados;
- (C) design de interface, subcategoria múltiplos métodos de saída;
- (D) ambiente técnico, subcategoria múltiplas interfaces de entrada/saída;
- (E) operações lógicas e matemáticas, subcategoria operações para formatação.

24

Considere o seguinte código em Python:

```

from math import prod
strPow = lambda r: prod(len(r) for x in r)
print(strPow('25'*2), "", sep=";")
  
```

Ao ser executado em um interpretador Python com suporte à função de biblioteca prod, o código acima imprime na saída padrão:

- (A) 0;
- (B) 40;
- (C) 256;
- (D) 625;
- (E) erro.

25

João foi direcionado, pela consultoria na qual trabalha, para um novo cliente, a fim de iniciar a elicitação de requisitos. Após analisar alguns documentos e entrevistar alguns gestores, ele resolveu utilizar a técnica de *card sorting*, na qual contará com a participação de um grupo de 15 usuários. Após a utilização da técnica, João observou que o *card sorting*:

- (A) envolve muitos custos e não oferece respostas adequadas em tempo viável;
- (B) é voltado exclusivamente para dados quantitativos;
- (C) facilita a definição de arquiteturas complexas e de meios de interoperabilidade necessários;
- (D) ajuda a fornecer informações bem estruturadas;
- (E) permite apenas a execução de forma presencial.

26

Ao implementar algumas funcionalidades em sua página pessoal, Lucas fez um teste, com base no código JavaScript apresentado a seguir, em um trecho do HTML.

```

<script>
  const f = function(...v){
    a = v.length>0?v.length*v[0]:0;
    v.splice(0,1);
    return a==0?a:a+f(...v);
  }
  console.log(f(...[1,2,3,...[0,1],3,2]));
</script>
  
```

Quando Lucas abriu a página, o valor impresso no console foi:

- (A) 6;
- (B) 12;
- (C) 34;
- (D) 36;
- (E) 45.

27

Ana está desenvolvendo o novo aplicativo da sua empresa e quer garantir um melhor nível de usabilidade para o produto.

Como se trata de uma empresa voltada para a terceira idade, Ana irá precisar de alguns cuidados a mais, entre eles:

- (A) expirar as sessões em menor tempo, visando a evitar fraudes;
- (B) trabalhar com relação de contraste mínima de 2:1;
- (C) evitar o uso de botões e barras de ferramentas, priorizando sempre os menus deslizantes;
- (D) apresentar apenas as informações necessárias, evitando sempre que possível a rolagem de tela;
- (E) trabalhar com fontes de tamanho pequeno e fixo, sem a possibilidade de utilizar zoom.

28

Ao configurar o design de uma página, Roberto usou o código CSS a seguir, em uma tag style, fazendo com que um menu de navegação lateral fique ao lado da área de conteúdo.

```
aside {float: left; width: 30%}
main {float: left; width: 70%}
```

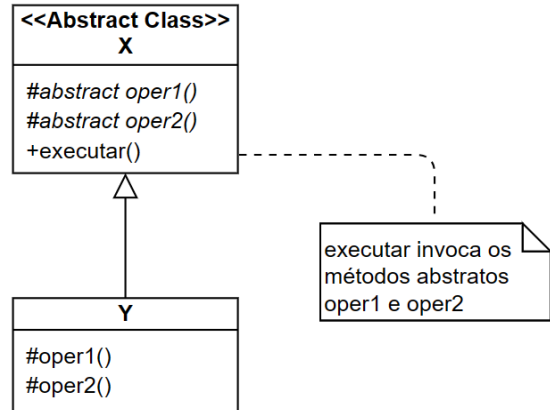
O resultado em tablets e desktops foi bom, mas o acesso pelo celular apresentou alguns problemas de usabilidade, sendo mais adequado que o menu ficasse acima do conteúdo.

Para utilizar o design indicado em telas menores, garantindo a responsividade da página, Roberto acrescentou ao código CSS:

- (A) @media screen and (max-width: 600px) {
 aside, main {width: 100%}
 }
- (B) @media handheld {
 aside, main {width: 50%}
 }
- (C) @media screen and (max-width: 600px) {
 aside, main {position: absolute}
 }
- (D) @media screen and (min-width: 768px) {
 aside, main {width: 100%}
 }
- (E) @media (max-width: 480px) {
 aside, main {width: 50%}
 }

29

Os padrões de projeto são extremamente úteis para organizar a arquitetura do sistema e o modelo de programação. Eles são projetados em diagramas da UML, como no modelo a seguir.



O diagrama expressa o padrão de projeto:

- (A) Observer;
- (B) Singleton;
- (C) Data Access Object;
- (D) Intercept Filter;
- (E) Template Method.

30

O analista Joaquim precisou ordenar um array com N elementos. Para economizar tempo, Joaquim optou por usar um algoritmo já disponível na biblioteca de ordenação. A biblioteca contém as implementações originais dos algoritmos Quicksort, Selection Sort, Insertion Sort, Merge Sort e Heap Sort. O analista escolheu o algoritmo que, no pior caso, apresenta uma relação quadrática entre a quantidade de operações necessárias para a ordenação e o número de elementos do array. No caso médio, a quantidade de operações necessárias se aproxima de N multiplicado por um logaritmo de N.

Joaquim escolheu o algoritmo de ordenação:

- (A) Quicksort;
- (B) Heap Sort;
- (C) Merge Sort;
- (D) Insertion Sort;
- (E) Selection Sort.

31

William está trabalhando com a base de dados de uma clínica, onde estão presentes as entidades descritas a seguir.

```
Paciente (Id: Integer PK,
  Nome: Varchar(50))
Medico (Id: Integer PK, Nome: Varchar(50))
Atendimento (Id: Integer PK,
  IdPaciente: Integer FK(Paciente),
  IdMedico: Integer FK(Medico),
  Data: Date)
```

Para que William obtenha os nomes dos pacientes que foram tratados por todos os médicos, a instrução SQL a ser utilizada é:

- (A) `select p.nome from paciente p, atendimento a where a.idpaciente = p.id and a.idmedico = all (select id from medico);`
- (B) `select p.nome from paciente p where not exists (select id from medico m where id not in (select idmedico from atendimento where idpaciente = p.id));`
- (C) `select p.nome from paciente p where exists (select idmedico from atendimento where idpaciente = p.id);`
- (D) `select p.nome from paciente p, atendimento a where a.idpaciente=p.id group by p.nome having count(*) = (select count(*) from medico);`
- (E) `select p.nome from paciente p, atendimento a where a.idpaciente = p.id and a.idmedico = any (select id from medico);`

32

A plataforma Docker disponibiliza um ambiente muito propício para a implementação de microsserviços. A equipe de Gabriel trabalha com os contêineres da Docker para disponibilização de seus serviços. Recentemente ocorreu um problema com o contêiner para o serviço de envio de e-mails, mas as falhas não impactaram os demais serviços, já que o projeto foi pensado desde o início com foco em independência. Isso permitiu que Gabriel se concentrasse na correção do contêiner, enquanto o restante da equipe continuou a trabalhar normalmente.

Entre os benefícios dos microsserviços, esse é um exemplo de:

- (A) heterogeneidade de ambiente;
- (B) resiliência;
- (C) escalabilidade;
- (D) portabilidade simplificada;
- (E) agilidade na entrega.

33

Diversas operações matemáticas podem ser implementadas de forma recursiva, como no algoritmo seguinte.

```
Função X (J: inteiro, K: inteiro)
Início
  Se J < K Então
    Retorne J
  Senão
    Retorne X (J-K, K)
Fim
```

Considerando o domínio dos inteiros positivos, a função terá como resultado o(a):

- (A) adição entre J e K;
- (B) multiplicação de J por K;
- (C) resto da divisão de J por K;
- (D) subtração entre J e K;
- (E) elevação de J à potência K.

34

A fábrica de software SeuSW estabeleceu formalmente um Processo Geral (PG) para condução de projetos de desenvolvimento de diferentes tipos de produtos de software. Assim, a descrição do processo contém a definição do ciclo de vida do projeto e a lista de tarefas a serem executadas. Considerando o Modelo de Referência MPS.BR (Melhoria de Processo do Software Brasileiro), o PG da SeuSW estará, minimamente, aderente a um resultado esperado para o nível:

- (A) "G" quando protótipos, demonstrações ou revisões forem utilizados para validar os requisitos;
- (B) "G" quando a rastreabilidade bidirecional entre requisitos, atividades e produtos de trabalho do projeto for estabelecida e mantida;
- (C) "F" quando os treinamentos identificados como necessários para capacitação dos colaboradores forem realizados e registrados;
- (D) "F" quando as estimativas de dimensão de tarefas e produtos de trabalho do projeto forem estabelecidas com a utilização da Análise de Pontos por Função;
- (E) "E" quando o processo definido para o projeto, derivado da estratégia para adaptação do processo da organização, for estabelecido, mantido atualizado e utilizado.

35

Para acelerar a busca sobre uma lista de mensagens, Beatriz adotou uma tabela de dispersão, na qual o e-mail do emissor é quem define o hash.

```
N: INTEIRO
V: VETOR [0..N-1] de LISTA<MENSAGEM>
Algoritmo Adicionar (M: MENSAGEM)
  H <- 0
  Para i de 0 até Tamanho (M.email) - 1
    H <- H + Ord (M.email[i])
  Fim Para
  H <- H Mod N
  V[H].Incluir(M)
Fim Algoritmo
```

O hash é dado pelo resto da divisão entre a soma dos códigos ASCII do email e o tamanho do vetor de listas.

Para que Beatriz obtenha a melhor distribuição das mensagens nas listas:

- (A) o valor dos códigos ASCII, obtidos pela função Ord, deve ser multiplicado por N;
- (B) a soma dos códigos ASCII deve ser feita do final para o início do campo email de M;
- (C) o número N deve ser primo;
- (D) a mensagem M deve ser incluída na lista da posição $N - H$ do vetor V;
- (E) o número N precisa ser par.

36

O formato JSON é utilizado, por padrão, para a transmissão de dados nos Web Services REST. Nesse sentido ele é diferente do SOAP, que utiliza o formato XML. Observe o trecho XML a seguir.

```
<usuario>
  <codigo>101</codigo>
  <nome>Ana Maria</nome>
  <setores>
    <setor>Financeiro</setor>
    <setor>Comercial</setor>
  </setores>
</usuario>
```

A representação da mesma entidade no formato JSON seria:

- (A) {"usuario": 101, "Ana Maria"; "setores": "Financeiro", "Comercial"}
- (B) [{"codigo": 101,"nome": "Ana Maria", "setor": "Financeiro", "setor": "Comercial"}]
- (C) {"codigo": 101,"nome": "Ana Maria", "setores": ["Financeiro", "Comercial"]}
- (D) [{"codigo": "101","nome": "Ana Maria"}, {"setores": ["Financeiro", "Comercial"]}]
- (E) {"codigo": "101","nome": "Ana Maria", "setores": {"Financeiro", "Comercial"}}

37

Observe as seguintes classes, definidas na sintaxe do PHP.

```
class SaidaSimples {
  protected $texto;
  public function __construct($texto) {
    $this->texto = $texto;}
  public function render() {
    echo($this->texto);}
}
class SaidaNegrito extends SaidaSimples{
  public function render() {
    echo("<b>");
    #COMANDO
    echo("</b>");
  }
}
```

Se o objetivo de SaidaNegrito é imprimir o texto herdado de SaidaSimples na página, entre as etiquetas de formatação, o comentário #COMANDO deve ser substituído por:

- (A) parent::render();
- (B) \$super->render();
- (C) SaidaSimples->render();
- (D) base::render();
- (E) \$inherited->render();

38

O modelo CMMI utiliza níveis de maturidade que descrevem o caminho evolutivo recomendado para as empresas que objetivam melhorar os seus processos no desenvolvimento de produtos e serviços.

Quando a empresa possui políticas e procedimentos básicos para gerenciar os requisitos, bem como os recursos, os prazos e os custos dos projetos, o nível de maturidade dessa empresa é:

- (A) nível 1- inicial;
- (B) nível 2 - gerenciado;
- (C) nível 3 - definido;
- (D) nível 4 - gerenciado quantitativo;
- (E) nível 5 - otimizado.

39

A consultora Fernanda encontrou o método Java a seguir em um código de terceiros.

```
//hx é do tipo HashMap<Float,List<Float>>
float w =
    hx.entrySet().stream().map(
        x-> x.getValue().stream().map(
            y -> y * x.getKey())
        .map(x->x.reduce(0f, (a,b)->a+b))
        .reduce(0f, (a,b)->a+b) /
    hx.entrySet().stream().map(
        x->x.getKey()*x.getValue().size())
        .reduce(0f, (a,b)->a+b);
```

Como Fernanda conhece muito bem as operações de map e reduce do Java, ela logo descobriu que w representa:

- (A) o desvio padrão do conjunto de valores do HashMap;
- (B) a média ponderada dos valores do HashMap, considerando as chaves como peso;
- (C) o produtório do conjunto de valores do HashMap;
- (D) a obtenção da chave relacionada à lista de maior valor médio;
- (E) o somatório dos produtos entre as chaves do HashMap e as listas de valores relacionadas.

40

A analista Ana precisou implementar rapidamente uma função hash denominada AHash. A AHash deve determinar um valor numérico entre 0 e 5 para uma chave de entrada. Ana optou por implementar em AHash o método de hashing denominado Método da Divisão. Para compatibilizar possíveis chaves alfanuméricas com o Método da Divisão, Ana implementou um dicionário que atribui a cada caracter um valor numérico. Internamente, a AHash utiliza como chave o produto dos números correspondentes aos caracteres da chave original. Sabendo que os caracteres C, V e M correspondem, respectivamente, aos números 67, 86 e 77, a AHash retornará para a chave "CVM":

- (A) 0;
- (B) 1;
- (C) 2;
- (D) 3;
- (E) 4.

Arquitetura de Computadores, Sistemas Operacionais e Redes

41

Durante uma análise detalhada da comunicação de dados em uma grande corporação, o analista Claudio identificou uma série de problemas relativos ao processamento de dados na rede interna. Para solucionar esses problemas, ele decidiu aplicar seus conhecimentos sobre a Arquitetura OSI da ISO, uma vez que essa estrutura fornece um modelo para compreender e solucionar questões de comunicação de rede.

A função da camada de Sessão na Arquitetura OSI, escolhida por Claudio, é a de:

- (A) gerenciar a transmissão de bits brutos sobre um meio físico;
- (B) fornecer conectividade e seleção de caminho entre dois sistemas de host;
- (C) entregar dados entre dois processos em diferentes sistemas;
- (D) controlar o estabelecimento, gerenciamento e término de conexões entre aplicações;
- (E) realizar a conversão de dados para garantir a transferência segura entre sistemas.

42

Enquanto trabalha na otimização de um sistema operacional de alta performance, a analista Jane se depara com o desafio de aprimorar o gerenciamento de memória dos processos em execução. Sua tarefa é analisar as áreas de memória de um processo e identificar estratégias para melhorar a eficiência e a performance do sistema.

Considerando as funções específicas e características de cada área de memória de um processo, a melhor estratégia para otimizar o gerenciamento de memória é:

- (A) limitar o tamanho da Stack a um valor fixo e pequeno para todos os processos, a fim de forçar uma programação mais eficiente e evitar o uso excessivo de chamadas de função;
- (B) alocar um grande segmento de memória para o código em todos os processos, presumindo que mais espaço para código resultará em melhor performance;
- (C) implementar uma política de alocação de memória que ajuste dinamicamente o tamanho da Heap e da Stack com base no comportamento em tempo real dos processos;
- (D) manter a área de dados não inicializados (BSS) em um tamanho mínimo constante, independentemente das necessidades específicas do programa, para simplificar o gerenciamento de memória;
- (E) aumentar o espaço dedicado ao segmento de dados inicializados, supondo que isso automaticamente otimizará o acesso a variáveis globais e constantes.

43

Em um cenário em que o analista Ken enfrenta o desafio de implementar protocolos de segurança para a transmissão de dados confidenciais em uma organização internacional, ele precisa escolher a opção que descreve a função e aplicação do protocolo SSL (*Secure Sockets Layer*) no contexto de HTTPS, FTPS e SFTP.

Considerando as características e funcionalidades desses protocolos, Ken deve optar por:

- (A) SSL, um protocolo de camada de aplicação que facilita a criptografia de páginas da web estáticas, tornando o HTTPS ineficaz para conteúdo dinâmico;
- (B) FTPS, que utiliza SSL para criptografar apenas os comandos de login, deixando a transferência de arquivos em si desprotegida;
- (C) SFTP, uma extensão do protocolo *Secure Shell* (SSH) que não utiliza SSL ou TLS para a criptografia de dados, dependendo exclusivamente de algoritmos de compressão para segurança;
- (D) HTTPS, uma combinação de HTTP e SSL/TLS que assegura a criptografia de dados entre o navegador do usuário e o servidor web, sendo adequado tanto para conteúdo estático quanto dinâmico;
- (E) SSL, que foi projetado exclusivamente para o envio seguro de e-mails, tornando-se incompatível com protocolos como HTTPS, FTPS e SFTP.

44

Em um sistema operacional, diversos algoritmos de alocação de memória são empregados para gerenciar eficientemente o espaço de memória disponível. Nesse contexto, o algoritmo de alocação de memória que soluciona o problema da fragmentação externa, ao escolher o menor espaço adequado para a alocação, é o:

- (A) First Fit;
- (B) Best Fit;
- (C) Worst Fit;
- (D) Next Fit;
- (E) Quick Fit.

45

A arquitetura de um computador define seu conjunto de instruções, modos de endereçamento e tipos de dados, enquanto a organização refere-se aos aspectos operacionais concretos do sistema. Com base nessa distinção, analise as afirmações a seguir sobre um hipotético computador X.

- I. O computador X utiliza uma arquitetura de conjunto de instruções complexas (CISC) que permite uma ampla variedade de modos de endereçamento.
- II. A organização do computador X inclui um pipeline de instruções de cinco estágios para aumentar a taxa de execução de instruções.
- III. A organização do computador X é otimizada para computação em nuvem, utilizando uma matriz de processadores para processamento paralelo.

Com base nessas afirmações, a opção que descreve uma característica que é estritamente parte da organização do computador X é:

- (A) suporte a uma ampla variedade de modos de endereçamento;
- (B) utilização de uma arquitetura CISC;
- (C) suporte a operações de ponto flutuante no conjunto de instruções;
- (D) implementação de um pipeline de instruções de cinco estágios;
- (E) otimização para computação em nuvem através de processamento paralelo.

46

No contexto de processadores modernos que implementam técnicas como execução fora de ordem e especulativa, o ciclo de execução de instruções se torna mais complexo.

Considerando essas técnicas avançadas, a opção que descreve um aspecto do processo de execução de instruções que é fundamental para otimizar o desempenho do processador, mantendo a precisão dos resultados, está corretamente descrito em:

- (A) o processador executa instruções estritamente na ordem em que aparecem no programa, sem considerar a possibilidade de dependências de dados entre instruções subsequentes;
- (B) antes de buscar instruções na memória, o processador sempre espera a conclusão da execução de todas as instruções anteriores, garantindo a execução sequencial para minimizar erros de previsão de salto;
- (C) o processador utiliza uma unidade de previsão de salto para executar instruções especulativamente, antes da confirmação de que essas instruções precisam ser executadas, revertendo alterações se a especulação estiver incorreta;
- (D) as instruções são decodificadas em uma ordem fixa e armazenadas em um buffer até que todos os operandos estejam disponíveis, momento no qual o processador interrompe todas as outras operações para executar essa instrução;
- (E) para cada instrução, o processador primeiro atualiza o contador de programa e, em seguida, executa uma análise completa do código do programa para identificar possíveis otimizações de execução antes de proceder à busca de instruções.

47

O analista José está diante de um sistema operacional hipotético que implementa um modelo avançado de gerenciamento de processos. Esse modelo define cinco estados distintos pelos quais um processo pode passar durante seu ciclo de vida, além de incorporar mecanismos sofisticados para o escalonamento e a comunicação entre processos. Considerando esse cenário, José precisa resolver um problema complexo de deadlock que ocorreu devido a uma sequência específica de transições de estado e solicitações de recursos entre quatro processos distintos. Para entender e resolver esse problema, é fundamental que o analista compreenda os conceitos e estados de processo no contexto dos sistemas operacionais.

Para resolver o problema de deadlock, a sequência dos estados de processo e os conceitos que o analista José deve considerar são:

- (A) criação (o processo é criado), pronto (aguarda pela alocação do processador), executando (o processo está sendo executado pelo processador), espera/bloqueado (o processo espera por algum evento ou recurso), terminado (o processo conclui sua execução). Deadlocks ocorrem quando processos em estado de espera/bloqueado mantêm recursos que outros processos estão tentando acessar, enquanto simultaneamente tentam acessar recursos mantidos por outros processos, criando um ciclo de dependência sem resolução;
- (B) executando (o processo está sendo executado pelo processador), pronto (aguarda pela alocação do processador), espera/bloqueado (o processo espera por algum evento ou recurso), criação (o processo é criado), terminado (o processo conclui sua execução). Deadlocks são resolvidos automaticamente pelo sistema operacional através de preempção regular dos processos;
- (C) pronto (aguarda pela alocação do processador), executando (o processo está sendo executado pelo processador), criação (o processo é criado), terminado (o processo conclui sua execução), espera/bloqueado (o processo espera por algum evento ou recurso). Deadlocks não são problemáticos em sistemas que implementam algoritmos de escalonamento baseados em prioridades;
- (D) espera/bloqueado (o processo espera por algum evento ou recurso), pronto (aguarda pela alocação do processador), executando (o processo está sendo executado pelo processador), criação (o processo é criado), terminado (o processo conclui sua execução). A detecção de deadlocks requer que o sistema operacional monitore o grafo de alocação de recursos para identificar ciclos de dependência;
- (E) terminado (o processo conclui sua execução), espera/bloqueado (o processo espera por algum evento ou recurso), pronto (aguarda pela alocação do processador), executando (o processo está sendo executado pelo processador), criação (o processo é criado). Em sistemas operacionais modernos, mecanismos de comunicação entre processos, como semáforos e monitores, são utilizados para prevenir e resolver deadlocks.

48

Em um ambiente de alta disponibilidade e escalabilidade, o analista Joabe precisa escolher o sistema de armazenamento para uma aplicação que demanda acesso rápido aos dados, tolerância a falhas e capacidade de escalar horizontalmente.

Nesse contexto, Joabe deve optar pelo(a):

- (A) utilização exclusiva de discos rígidos externos USB conectados em série para garantir a escalabilidade;
- (B) implementação de fitas magnéticas como solução principal de armazenamento, devido à sua durabilidade e baixo custo;
- (C) adoção de um sistema RAID 0 com dois discos para maximizar a tolerância a falhas;
- (D) emprego de armazenamento em nuvem com serviços que ofereçam elasticidade, durabilidade e disponibilidade de dados;
- (E) configuração de um array de disquetes em paralelo, visando a aumentar a velocidade de acesso e a capacidade de armazenamento.

49

O analista Paulo está investigando um problema complexo de performance em um sistema operacional distribuído que utiliza um modelo de sistema de arquivos específico. Ele tem o objetivo de melhorar a eficiência no acesso e manipulação de dados nesse ambiente de alta disponibilidade.

Nesse contexto, considerando os diferentes modelos de sistemas de arquivos e suas características, o sistema de arquivos analisado por Paulo é o:

- (A) FAT32 (*File Allocation Table 32*);
- (B) NTFS (*New Technology File System*);
- (C) ext4 (*Fourth Extended Filesystem*);
- (D) HDFS (*Hadoop Distributed File System*);
- (E) APFS (*Apple File System*).

50

Durante a análise de um sistema computacional avançado, o analista Clementino identifica uma falha crítica de desempenho que pode ser atribuída à arquitetura do computador. Ele observa que a falha está relacionada à forma como os diferentes componentes do sistema interagem entre si, afetando diretamente a eficiência do processamento de dados.

Com base nos seus conhecimentos sobre a organização e funcionamento dos componentes principais de um computador, sistemas de memória, sistemas de armazenamento e periféricos de entrada e saída, é correto afirmar que a causa raiz do problema identificado por Clementino é o(a):

- (A) aumento da frequência do processador além do suportado pela placa-mãe, causando superaquecimento e falhas esporádicas no sistema;
- (B) incompatibilidade entre o tipo de memória RAM utilizada e a velocidade máxima suportada pelo controlador de memória do processador, resultando em gargalos de desempenho;
- (C) uso, no SSD, de um sistema de arquivos ineficiente, que não está otimizado para as operações de leitura e escrita aleatórias, reduzindo a velocidade geral do sistema;
- (D) falha na configuração do DMA (*Direct Memory Access*) para os periféricos de entrada e saída, causando um uso excessivo da CPU para operações de E/S, o que diminui a eficiência do processamento de dados;
- (E) implementação de um algoritmo de criptografia muito complexo que exige capacidade computacional além do que o hardware atual pode fornecer, resultando em atrasos significativos no processamento de dados.

51

Considerando os conceitos de paralelismo e multiprocessamento em sistemas de computação de alto desempenho, é correto afirmar que:

- (A) a eficiência do sistema é inversamente proporcional ao número de processadores, ou seja, quanto mais processadores, menor a eficiência do sistema, devido ao aumento da complexidade de gerenciamento;
- (B) a eficiência do sistema pode ser melhorada até um certo ponto pelo aumento do número de processadores, mas está sujeita à Lei de Amdahl, que estabelece que há um limite para o quanto a performance pode ser aumentada através do paralelismo devido à fração de código que deve ser executada sequencialmente;
- (C) o multiprocessamento garante que todas as tarefas serão executadas em paralelo, eliminando a necessidade de qualquer execução sequencial e maximizando a eficiência do sistema;
- (D) em sistemas que implementam paralelismo e multiprocessamento, a eficiência é maximizada apenas quando cada processador executa uma única tarefa de cada vez, sem qualquer forma de troca de contexto ou multitarefa;
- (E) o uso de técnicas de paralelismo e multiprocessamento implica que não há necessidade de otimizações no código, pois a adição de mais processadores resolve automaticamente quaisquer problemas de desempenho.

52

O analista sênior de redes Ribeirão recebeu a incumbência de esclarecer as nuances e atributos fundamentais das redes de computadores para um grupo de estagiários em uma corporação de grande porte.

No que se refere às propriedades e funcionalidades vitais das redes P2P, WAN, LAN e WLAN, é correto afirmar que:

- (A) as redes LAN (*Local Area Networks*) são projetadas para cobrir áreas geográficas extensas, como cidades ou estados, facilitando a comunicação de longa distância com alta latência;
- (B) redes WAN (*Wide Area Networks*) operam exclusivamente dentro do espectro de frequência sem fio, o que as torna indistinguíveis das WLANs (*Wireless Local Area Networks*) em termos de tecnologia de transmissão;
- (C) a tecnologia WLAN (*Wireless Local Area Network*) é inerentemente incapaz de suportar protocolos de criptografia, o que limita severamente sua aplicabilidade em ambientes que requerem comunicações seguras;
- (D) em redes P2P (*Peer-to-Peer*), a presença de um servidor central é crucial para a gestão de tráfego e autorização de acesso, garantindo a estabilidade e segurança da rede;
- (E) as redes WAN (*Wide Area Networks*) são utilizadas para conectar redes menores em distâncias geograficamente extensas, incluindo LANs (*Local Area Networks*) entre diferentes locais, cidades ou até países.

53

No contexto de sistemas computacionais, uma interrupção (IRQ - *Interrupt Request*) é um sinal enviado a um processador, indicando um evento que precisa de atenção imediata.

Com base nessa definição, é correto afirmar, em relação ao funcionamento das interrupções, que:

- (A) uma IRQ pode ser gerada exclusivamente por dispositivos externos ao processador, como teclados e mouses, e nunca por software;
- (B) todas as IRQs têm a mesma prioridade, sendo tratadas na ordem exata em que são recebidas pelo processador, independentemente de sua fonte ou importância;
- (C) uma vez que uma IRQ é recebida, o processador imediatamente interrompe qualquer tarefa que esteja executando, sem salvar o estado atual da tarefa interrompida;
- (D) as IRQs são desnecessárias em sistemas operacionais modernos, pois eles possuem mecanismos alternativos que substituem a necessidade de interrupções;
- (E) as IRQs permitem que o processador seja alertado sobre eventos importantes, pausando temporariamente a execução do programa atual para tratar o evento, e, em seguida, restaurando o estado anterior do programa e continuando a execução de onde havia parado.

54

Em um sistema de gerenciamento de recursos de uma grande corporação de tecnologia, o analista Hop precisa resolver um problema de Starvation que afeta a distribuição equitativa de recursos computacionais entre vários processos críticos. Após uma análise profunda, Hop identifica que o algoritmo de escalonamento atual prioriza processos com requisitos de recursos mais altos, deixando processos com requisitos menores em espera indefinida. Para resolver esse problema, Hop propõe uma solução inovadora que reestrutura a lógica de alocação de recursos, garantindo que todos os processos recebam uma fatia justa de tempo de CPU, sem comprometer a eficiência global do sistema.

Para resolver o problema de Starvation, Hop adotou a seguinte abordagem:

- (A) implementação de um algoritmo de escalonamento Round-Robin, que distribui o tempo de CPU igualmente entre todos os processos, independentemente de seus requisitos de recursos;
- (B) criação de um sistema de prioridades dinâmicas, onde os processos que esperam há mais tempo têm sua prioridade aumentada gradualmente até que sejam atendidos;
- (C) adoção do algoritmo de escalonamento *First-Come, First-Served* (FCFS), garantindo que todos os processos sejam atendidos pela ordem de chegada, sem priorização baseada em requisitos de recursos;
- (D) desenvolvimento de um mecanismo de feedback que ajusta a alocação de recursos em tempo real, baseado na análise de uso de CPU e na previsão de demanda futura;
- (E) introdução de um sistema de cotas de recursos, onde cada processo recebe uma cota fixa, que pode ser ajustada manualmente conforme a necessidade.

55

O estagiário Bebeto, durante um treinamento, constatou que a primeira implementação de memória virtual foi realizada no início da década de 1960, no sistema Atlas, desenvolvido na Universidade de Manchester (Kilburn, 1962). Atualmente, a maioria dos sistemas implementa memória virtual, com exceção de alguns sistemas operacionais de supercomputadores.

Nesse contexto, Bebeto pode concluir, em relação à funcionalidade da memória virtual dentro de um sistema operacional contemporâneo, que ela:

- (A) é empregada primordialmente para acelerar a execução direta de instruções na unidade de processamento central (CPU), eliminando a necessidade de armazenamento secundário durante a execução de programas;
- (B) é uma técnica projetada para reduzir proposadamente a capacidade de memória RAM física acessível pelo sistema, com o intuito de otimizar a alocação de recursos e melhorar a eficiência do sistema por meio de uma gestão de memória mais restritiva;
- (C) se baseia na utilização de espaço em dispositivos de armazenamento secundário, como discos rígidos, para simular uma capacidade de memória RAM mais extensa, possibilitando assim a execução de programas que excedem o tamanho da memória física disponível;
- (D) é uma tecnologia obsoleta, visto que os sistemas operacionais modernos dependem de armazenamento secundário para todas as operações de memória;
- (E) é aplicada ao armazenamento e acesso de dados via soluções de cloud computing, eliminando a necessidade de armazenamento local ao prover um acesso ininterrupto e remoto a recursos de memória virtualizados.

56

Durante a implementação de uma infraestrutura de rede altamente segura para uma organização governamental, o analista Zoroastro se depara com um grande desafio. A complexidade do projeto exige uma compreensão profunda da Arquitetura TCP/IP, especialmente no que tange à otimização do tráfego e à segurança dos dados.

Nesse contexto, é correto afirmar, acerca da função do protocolo ICMP (*Internet Control Message Protocol*) escolhida por Zoroastro, que ele:

- (A) é primariamente utilizado para a transmissão de dados em tempo real, como chamadas de voz e vídeo;
- (B) é responsável pela retransmissão de pacotes perdidos durante a transmissão de dados;
- (C) é usado para a alocação dinâmica de endereços IP dentro de uma rede;
- (D) permite a fragmentação e remontagem de pacotes de dados para eficiência no roteamento;
- (E) é utilizado para diagnóstico e geração de mensagens de erro na rede.

57

Em um projeto de desenvolvimento de um sistema operacional voltado para aplicações críticas de tempo real, o analista Jony está avaliando diferentes métodos de Comunicação entre Processos (IPC) para garantir a máxima eficiência, confiabilidade e atendimento aos requisitos de tempo real.

Considerando os desafios associados a esses sistemas, o método de IPC para sistemas de tempo real escolhido por Jony é:

- (A) pipes anônimos;
- (B) semáforos;
- (C) filas de mensagens prioritárias;
- (D) memória compartilhada;
- (E) sockets de rede.

58

Considerando as técnicas avançadas de gerenciamento de memória e operações de entrada/saída (I/O), o analista Pedro é desafiado a otimizar o desempenho do sistema operacional de uma organização. A estratégia correta, escolhida por Pedro, que se alinha com os princípios apresentados para melhorar simultaneamente o gerenciamento de memória e a eficiência de I/O, sem comprometer o desempenho geral do sistema, é:

- (A) aumentar a frequência de operações de swapping para maximizar a disponibilidade de memória física;
- (B) implementar um esquema de alocação de memória fixa, dedicando partes específicas da memória para I/O exclusivo;
- (C) utilizar intensivamente o spooling de dispositivos de I/O para simular operações de E/S assíncronas em dispositivos que originalmente requerem operações síncronas;
- (D) integrar técnicas de memória virtual com paginação sob demanda, otimizando o uso da memória e o acesso a dispositivos de I/O;
- (E) priorizar o uso de interrupções de hardware para todas as operações de I/O, minimizando o uso de polling.

59

O analista Zeferino está trabalhando em um sistema operacional que necessita de melhorias em seu algoritmo de escalonamento e nas primitivas de sincronização para otimizar o processamento de tarefas em um ambiente de múltiplos processadores.

Visando à otimização do sistema em questão, a abordagem que deve ser adotada é:

- (A) implementar o algoritmo de escalonamento Round-Robin com quantum fixos para todos os processos, independentemente de suas prioridades;
- (B) utilizar semáforos binários para todas as operações de sincronização, aplicando o mesmo valor de inicialização para todos os semáforos;
- (C) utilizar o algoritmo de escalonamento de Múltiplas Filas com Feedback, ajustando dinamicamente as prioridades dos processos baseado em suas características de execução;
- (D) implementar um único lock global para gerenciar o acesso a todas as seções críticas, independentemente de sua relevância ou frequência de acesso;
- (E) priorizar a implementação de algoritmos de escalonamento baseados exclusivamente em prioridades fixas, sem considerar a possibilidade de inversão de prioridades.

60

No contexto de um sistema operacional que suporta execução paralela, o analista Léo é encarregado de otimizar um aplicativo que faz uso intensivo de threads. O aplicativo em questão tem experimentado tempos de resposta inconsistentes, suspeitando-se de uma ineficiência na forma como as threads são gerenciadas e escalonadas.

Considerando um ambiente com múltiplos núcleos de CPU, para melhorar o desempenho do aplicativo, o analista Léo deve:

- (A) incrementar o número de threads para igualar o número de núcleos de CPU disponíveis, partindo do princípio de que mais threads resultarão em melhor paralelismo e desempenho;
- (B) implementar um algoritmo de escalonamento de threads baseado em prioridades, onde threads que realizam operações críticas para o desempenho recebem maior prioridade;
- (C) forçar todas as threads a executarem em um único núcleo para evitar a sobrecarga associada à troca de contexto entre diferentes núcleos;
- (D) utilizar um modelo de programação baseado em eventos em vez de threads, eliminando a necessidade de gerenciamento de threads;
- (E) reduzir a prioridade do processo do aplicativo no gerenciador de tarefas do sistema operacional para que outros processos possam ser executados com maior eficiência, partindo do princípio de que isso indiretamente melhorará o desempenho do aplicativo multithreaded ao otimizar o uso geral de recursos do sistema.

Projetos e Governança TI

61

Yanni foi questionada pelo seu chefe sobre qual a metodologia que ela usa para mensuração do desempenho de TI. Ela respondeu que utiliza um sistema de planejamento e gerenciamento estratégico que serve para alinhar atividades de negócio à visão e à estratégia da organização.

É correto afirmar que Yanni utiliza:

- (A) RACI;
- (B) SWOT;
- (C) value it;
- (D) gap analysis;
- (E) balanced scorecard.

62

O analista João é responsável por realizar o planejamento do Projeto XYZ. O supervisor de João determinou que fosse dada máxima prioridade ao projeto. João deve cumprir o prazo estabelecido, mesmo que o custo seja maior do que o estimado inicialmente, tendo em vista que dele dependem outros projetos. A abordagem definida para o Projeto XYZ é do tipo preditiva, e, ao estimar o cronograma inicial, João verificou que a data de término extrapola o seu prazo. No Projeto XYZ todas as atividades são sequenciais e possuem relacionamento do tipo Término-Início.

Diante desse cenário, para cumprir o prazo estabelecido para o Projeto XYZ, João deverá:

- (A) empregar o paralelismo a fim de realizar atividades simultaneamente;
- (B) retirar atividades não desejadas do planejamento;
- (C) diminuir a data de estimativa para cada uma das atividades;
- (D) empregar a compressão para encurtar a duração de algumas atividades;
- (E) manter o planejamento realizado, pois o cronograma inicial é uma estimativa.

63

A analista Helena é a gerente de um projeto de alta complexidade. Uma grande empresa foi contratada para executar diferentes atividades relacionadas a ele. Essa empresa não está acompanhando o projeto e apresenta atrasos constantes. Helena acredita que os atrasos se devem ao fato de o preposto da empresa não estar devidamente comprometido com o projeto, o que o leva a não gerenciar corretamente a sua equipe.

Para resolver esse problema, e tendo em vista que o atraso do projeto deve ser o menor possível, Helena deve:

- (A) tomar ações para engajar mais o preposto, pois ele é uma parte interessada de grande importância para o projeto, além de compreender e analisar as partes interessadas de forma contínua;
- (B) informar à empresa contratada que o preposto não está desempenhando a sua função corretamente e solicitar que ele seja trocado o quanto antes, pois é função da empresa colocar profissionais capacitados em seus quadros;
- (C) continuar normalmente com o projeto, pois o maior interessado em realizar as entregas é a empresa e, ao não realizá-las, a empresa não receberá o pagamento devido;
- (D) parar de comunicar-se com o preposto e tratar somente com o pessoal técnico da empresa, pois quem, de fato, faz as entregas são eles, de modo que não há necessidade de interação com o preposto;
- (E) alterar o cronograma do projeto estimando que haverá atraso nas entregas dessa empresa, pois isso impactará menos do que a troca do preposto.

64

As dimensões descritas na ITIL 4 proporcionam uma abordagem holística no gerenciamento de serviços, oferecendo uma visão abrangente para a organização.

A dimensão em que são definidas as atividades, a sequência de trabalho e os controles e procedimentos necessários para a entrega de produtos e serviços é a de:

- (A) clientes e práticas;
- (B) organizações e pessoas;
- (C) informação e tecnologia;
- (D) parceiros e fornecedores;
- (E) fluxos de valor e processos.

65

Ao examinar o relatório de atendimentos do último semestre, o analista Luiz verificou que um mesmo incidente ocorre semanalmente no seu departamento. Apesar de o serviço ter sido normalizado em todos os incidentes, o analista resolveu investigar sua causa raiz e propor uma solução definitiva.

A prática de gerenciamento de serviços, prevista no ITIL 4, que Luiz está realizando é o gerenciamento de:

- (A) incidentes;
- (B) problemas;
- (C) continuidade de serviço;
- (D) requisição de serviço;
- (E) ativos de TI.

66

O analista Juca quer implementar o Sistema de Valor de Serviço (SVS) da ITIL 4. Para tanto, ele deve observar o componente que representa as recomendações que guiam as organizações em todas as circunstâncias, independente de mudanças em suas metas, estratégias, tipo de trabalho ou estrutura gerencial.

Portanto, Juca deve observar o seguinte componente:

- (A) governança;
- (B) cadeia de valor de serviço;
- (C) práticas;
- (D) princípios orientadores;
- (E) fluxos de valor.

67

A Lei nº 13.709/2018 - Lei Geral de Proteção de Dados Pessoais (LGPD) impõe às pessoas jurídicas de direito público obrigações de transparência ativa.

Um conjunto de informações previstos na LGPD, para o qual deve ser dada publicidade, inclui:

- (A) e-mail funcional, além de primeiro e último nome do controlador, do operador e do encarregado;
- (B) nome e cargo do encarregado, juntamente com telefone e e-mail específico para orientação e esclarecimento de dúvidas;
- (C) nome completo e matrícula do controlador, juntamente com e-mail funcional para orientação e esclarecimento de dúvidas;
- (D) e-mail e telefone, para ligações gratuitas, da pessoa jurídica de direito público ou privado responsável pelos agentes de tratamento;
- (E) CNPJ do operador, quando pessoa jurídica de direito público ou privado, além de e-mail específico para pleito de direitos por parte da pessoa natural.

68

Maiara, gerente de TI, proferiu uma palestra sobre o plano diretor de TIC para os diretores e gerentes da empresa em que trabalha. Ao ser questionada sobre a elaboração do plano, a gerente de TI respondeu que acontece em fases e com a participação de papéis distintos.

Acerca desses papéis, é correto afirmar que o(a):

- (A) comitê gestor de TI atuará como aprovador, monitorará o desempenho de TI e direcionará os investimentos para alcançar os objetivos estratégicos;
- (B) autoridade máxima atuará como apoiadora, tomadora de decisões, publicadora, priorizadora de ações e definidora de diretrizes, isto é, será a responsável pelo PDTI;
- (C) equipe de elaboração será responsável pela obtenção, organização e compilação de informações que compõem o PDTI, composto exclusivamente por pessoas de TI;
- (D) comitê gestor de TI atuará como priorizador de ações, monitorará o desempenho de TI e direcionará os investimentos para alcançar os objetivos estratégicos;
- (E) autoridade máxima atuará como apoiadora, aprovadora, tomadora de decisões, publicadora, definidora de diretrizes e direcionará os investimentos para alcançar os objetivos estratégicos.

69

Em um projeto de alta complexidade e de grande relevância para a CVM, houve alteração do patrocinador, tendo em vista uma reorganização da estrutura do seu órgão. O novo patrocinador solicitou uma reunião com o gerente do projeto, Gustavo, para tratar das mudanças que serão necessárias. Algumas entregas terão que ser ajustadas, tendo em vista ter ocorrido uma mudança no ambiente do projeto.

Nesse cenário, o gerente Gustavo deve:

- (A) informar ao patrocinador que os documentos de planejamento já foram confeccionados e que não são mais aceitas mudanças nessa fase do projeto. Ainda, destacar que, caso o patrocinador deseje alguma informação sobre o projeto, o gerente está à disposição para repassar;
- (B) acatar a solicitação do patrocinador de realizar a reunião, mas destacar que não será fácil realizar mudanças no projeto, tendo em vista que todos os recursos externos já estão alocados em outros projetos;
- (C) entender a posição do novo patrocinador, mas informar que, em projetos de alta complexidade, mudanças são difíceis de serem implementadas;
- (D) informar ao patrocinador que não será necessária a reunião, mas somente que ele envie as mudanças desejadas, para que essas sejam implementadas o mais rápido possível;
- (E) informar ao patrocinador que está disponível para a reunião e comentar que mudanças são normais durante a execução de um projeto, porém destacar que essas propostas de mudança serão acompanhadas por uma avaliação de novos riscos e impactos nos recursos do projeto.

70

Para implantar um programa de governança de TIC, é importante elencar pontos críticos como:

- (A) definir o escopo, gerenciar riscos e realizar auditorias;
- (B) definir processos, monitorar o andamento e alinhar politicamente;
- (C) premiar a equipe, definir contingências e informar regularmente;
- (D) comprometer a alta direção, monitorar a transição cultural e capacitar a equipe;
- (E) ter um patrocinador poderoso, comprometer a instituição e definir receitas futuras.

RASCUNHO

Realização

