

**-- CONHECIMENTOS ESPECÍFICOS --****PROCESSO****Questão 31**

No contexto de desenvolvimento de *software*, uma equipe que busca entregar incrementos contínuos e se adaptar rapidamente às mudanças do mercado deve priorizar a utilização de um modelo que possibilite

- A um ciclo iterativo e incremental, no qual pequenas entregas são realizadas regularmente e podem ser ajustadas de acordo com o *feedback* do cliente.
- B uma documentação detalhada e completa antes do início da codificação, evitando alterações ao longo do projeto.
- C um planejamento rígido e sequencial, no qual cada fase depende da conclusão total da anterior.
- D uma abordagem estruturada, com requisitos bem definidos desde o início do projeto e mínima interação entre desenvolvimento e cliente.
- E um fluxo de trabalho dividido em etapas fixas, com revisão de código apenas ao final do processo.

**Questão 32**

No *framework* Scrum, entre os papéis bem definidos para a execução do projeto, o profissional responsável por garantir que a equipe esteja aplicando corretamente os princípios e práticas do Scrum, ajudando a remover impedimentos e promovendo a colaboração, é denominado

- A gerente de projeto.
- B analista de negócios.
- C *product owner*.
- D desenvolvedor.
- E *scrum master*.

**Questão 33**

No desenvolvimento de *software*, a prática de TDD (*Test-Driven Development*) tem como característica principal um fluxo estruturado de criação de código baseado em testes. Se uma equipe adotar os princípios do TDD, então ela deverá iniciar o desenvolvimento de um novo recurso, por meio

- A da execução manual de testes exploratórios antes da implementação do código.
- B da criação de um conjunto de testes automatizados antes da implementação do código, para garantir que ele atenda aos requisitos desejados.
- C da escrita do código-fonte e, após sua implementação, da criação dos testes para verificar sua funcionalidade.
- D da definição de requisitos detalhados e de um planejamento extensivo antes da escrita dos testes.
- E da validação do código mediante revisões de pares antes da criação de qualquer teste automatizado.

**Questão 34**

Um dos principais objetivos de uma equipe que utiliza o Kanban como abordagem ágil de desenvolvimento é otimizar o fluxo de trabalho e reduzir gargalos no processo. Nesse caso, para atingir essa meta, a equipe poderá utilizar como estratégia

- A a revisão periódica de código como método principal de otimização do fluxo de desenvolvimento.
- B a organização das tarefas de acordo com uma hierarquia fixa, em que cada atividade precisa seguir um fluxo predefinido sem desvios.
- C a limitação do trabalho em progresso (WIP), para evitar sobrecarga e garantir que as tarefas fluam de forma eficiente.
- D a definição rígida de requisitos antes do início do projeto, de forma a minimizar alterações ao longo do desenvolvimento.
- E a adoção de *sprints* com duração fixa, que permite entregas incrementais e constantes ao final de cada ciclo.

**BANCO DE DADOS****Questão 35**

Sabendo que as funções de agregação em SQL são aplicadas em um grupo de valores e um único valor retorna para cada grupo, assinale a opção em que é apresentada a função de agrupação que leva em consideração os valores nulos existentes.

- A SUM ()
- B AVG ()
- C MIN ()
- D MAX ()
- E COUNT ()

**Questão 36**

**Tabela I**  
**Clientes**

IdCliente	Nome	Endereco	Cidade	CEP
1	Jose Silva	Av Rio Branco 200	Rio de Janeiro	21200
2	Maria Silveira	Av Contorno 100	São Paulo	11901
3	Rita Castro	Rua Sao Joao 132	Brasilia	71000

**Tabela II**  
**Pedidos**

IdOrdem	IdCliente	IdVendedor	Data
101	3	55	2025-03-01
102	23	23	2025-05-02
103	1	67	2025-01-23
104	24	37	2024-12-11
105	2	23	2024-11-10
106	3	21	2025-05-10

**Tabela III**  
**Resultado**

Jose Silva	Rio de Janeiro	2025-01-23
Maria Silveira	São Paulo	2024-11-10
Rita Castro	Brasilia	2025-03-01
Rita Castro	Brasilia	2025-05-10

Com base nas informações nas tabelas I e II, precedentes, assinale a opção em que são apresentados comandos SQL que permitem a obtenção do resultado demonstrado na tabela III.

- A `SELECT Clientes.Nome, Clientes.Cidade, Pedidos.Data FROM Pedidos LEFT JOIN Clientes ON Pedidos.IdCliente=Clientes.IdCliente;`
- B `SELECT Clientes.Nome, Clientes.Cidade, Pedidos.Data FROM Pedidos, Clientes INNER JOIN Clientes ON Pedidos.IdCliente=Clientes.IdCliente;`
- C `SELECT Clientes.Nome, Clientes.Cidade, Pedidos.Data FROM Pedidos AND Clientes INNER JOIN Clientes ON Pedidos.IdCliente=Clientes.IdCliente;`
- D `SELECT Clientes.Nome, Clientes.Cidade, Pedidos.Data FROM Pedidos INNER JOIN Clientes ON Pedidos.IdCliente=Clientes.IdCliente;`
- E `SELECT Clientes.Nome, Clientes.Cidade, Pedidos.Data FROM Pedidos, Clientes LEFT JOIN Clientes ON Pedidos.IdCliente=Clientes.IdCliente;`

**Questão 37**

Suponha a existência de uma tabela de um banco de dados relacional no qual esteja armazenado um volume muito grande de dados, em muitas linhas de registros. Nessa situação, considerada a otimização do acesso aos dados dessa tabela,

- A a criação de um novo índice pode piorar o desempenho das operações feitas na tabela.
- B a criação de índices não seria a solução adequada, dado o tamanho da tabela.
- C recomenda-se a criação de vários índices, já que o volume de dados é muito grande.
- D um novo índice sempre vai implicar ganho de desempenho para as operações feitas na tabela.
- E a criação de um índice para cada coluna da tabela seria uma boa estratégia.

**Questão 38**

Em bancos de dados com suporte à indexação funcional,

- A pode ser criado um índice a partir do resultado de uma função aplicada à coluna.
- B é impossível implementar um índice funcional em uma coluna que contenha valores NULL.
- C somente é possível indexar colunas numéricas.
- D qualquer índice funcional tem de ser criado com algum procedimento armazenado.
- E o índice funcional pode duplicar os dados da tabela original.

**Questão 39**

Na arquitetura de banco de dados relacional, a camada de abstração de dados tem como principal objetivo

- A disponibilizar uma interface amigável ao administrador de dados.
- B realizar o balanceamento de carga entre os servidores que são usados pelo banco de dados.
- C facilitar a automatização de *backups* dos dados.
- D tornar transparente a camada física, de modo que os usuários possam trabalhar com uma visão lógica dos dados.
- E disponibilizar dados de consultas frequentes, armazenando-as de forma temporária.

**Questão 40**

Na arquitetura de banco de dados relacional, composta por vários elementos dispostos em camadas, a camada lógica tem como função

- A definir tabelas, relacionamentos e restrições de integridade, que são estruturas do banco de dados.
- B manter o bom nível de desempenho do banco de dados.
- C disponibilizar visões específicas de dados para os usuários.
- D gerenciar o armazenamento físico dos dados nas unidades de disco.
- E efetuar o controle para que transações concorrentes sejam corretamente executadas.

**Questão 41**

Em uma grande empresa, a prática mais adequada para o balanceamento entre normalização e desempenho em consultas de um banco de dados, considerando-se que o negócio exige leitura intensa e baixa taxa de atualização, seria

- A criar índices para todas as colunas que sejam usadas em qualquer tipo de consulta.
- B desconsiderar as formas normais e manter os dados em um único JSON, o que aumentaria a velocidade das operações.
- C normalizar todas as tabelas até 5FN.
- D permitir desnormalizações controladas, como colunas duplicadas, por exemplo, para obtenção de ganho de desempenho.
- E usar somente bancos de dados que sejam mais rápidos para leitura, como o NoSQL.

**Questão 42**

Um administrador de banco de dados pretende melhorar o desempenho de relatórios executivos que são lidos com muita frequência e executam *joins* complexos. Para tanto, ele considera desnormalizar algumas tabelas.

Nessa situação hipotética,

- A devem ser usadas, no processo de geração dos relatórios, somente tabelas que sejam temporárias e não persistentes.
- B devem ser mantidas somente visões materializadas com dados agregados, eliminando-se as tabelas originais.
- C a melhor solução é manter as tabelas normalizadas e fazer que os relatórios executem os *joins* complexos, embora com resultado mais lento.
- D alguns dados devem ser duplicados em tabelas de relatório, embora o procedimento crie certo nível de redundância controlada.
- E devem ser criados novos índices para todas as tabelas que sejam usadas nos relatórios.

**TÉCNICAS E LINGUAGENS DE PROGRAMAÇÃO****Questão 43**

Julgue os itens a seguir, de acordo com o princípio da responsabilidade única (PRS) do SOLID.

- I Uma classe deve ser aberta para extensão, mas fechada para modificação.
- II Objetos em um programa devem ser substituíveis por instâncias de seus subtipos sem alteração da correção do programa.
- III Uma classe deve depender de abstrações, e não de implementações concretas.

Assinale a opção correta.

- A Nenhum item está certo.
- B Apenas o item I está certo.
- C Apenas o item II está certo.
- D Apenas os itens I e III estão certos.
- E Apenas os itens II e III estão certos.

**Espaço livre**

**Questão 44**

```
<html>
<script type="text/javascript" >
function passol(valor) {
    return valor + 2;}
function passo2(valor) {
    return valor + 3;}
function passo1a(valor, callback) {
    const resultadola = valor + 2;
    callback(resultadola);}
function passo2a(valor, callback) {
    const resultadola = valor + 3;
    callback(resultadola);}
let resultado1 = 0 ;
resultado1 = passol(resultado1);
resultado1 = passo2(resultado1);
console.log('resultado1: ${resultado1}');
passola(0, (resultadola) => {
    passo2a(resultadola, (resultado2) => {
        console.log('resultado2: ${resultado2}');
    });
});
</script>
</html>
```

Assinale a opção em que é apresentado o resultado que aparecerá no console de um navegador caso o código precedente seja salvo em um arquivo .html e esse arquivo seja aberto nesse navegador.

- A resultado1: 5  
resultado2: 2
- B resultado1: 5  
resultado2: 5
- C resultado1: 2  
resultado2: 2
- D resultado1: 2  
resultado2: 3
- E resultado1: 3  
resultado2: 5

**Espaço livre**

**Questão 45**

```
public class Pessoa{
    protected string cpf = "444.555.666-77";
    protected string nome = "Pedro Maria Silva";
    public virtual void GetInfo(){
        Console.WriteLine($"Nome: {nome}");
        Console.WriteLine($"CPF: {cpf}");
    }
}
class Colaborador : Pessoa {
    public readonly string id = "ABC567EFG";
    public override void GetInfo(){
        base.GetInfo();
        Console.WriteLine($"ID colaborador: {id}");
    }
}
class TestClass
{
    static void Main()
    {
        Colaborador E = new Colaborador();
        E.GetInfo();
    }
}
```

Assinale a opção em que é apresentado o resultado da execução do código em C# precedente.

- A Nome: Pedro Maria Silva  
CPF: 444.555.666-77
- B Nome: Pedro Maria Silva  
CPF: 444.555.666-77  
ID colaborador: ABC567EFG
- C ID colaborador: ABC567EFG
- D Nome: Pedro Maria Silva
- E CPF: 444.555.666-77

**Questão 46**

```
public class aluno {
    private String nome;
    private int idade;
    public String getNome() { return nome; }
    public void setNome(String nome) {
        this.nome = nome; }
    public int getIdade() { return idade; }
    public void setIdade(int idade) {
        if (idade > 0) { this.idade = idade; }
    }
}
```

Assinale a opção em que é citada a propriedade da programação orientada a objetos evidenciada no código Java precedente.

- A composição (*composition*)
- B interface
- C herança
- D polimorfismo
- E encapsulamento

**Questão 47**

Assinale a opção correspondente ao caractere utilizado em CSS3 como seletor universal.

- A \*
- B :
- C @
- D .
- E #

**Questão 48**

Assinale a opção em que é apresentada a *tag* que, em HTML5, representa um conteúdo independente de um documento.

- A <nav>
- B <dialog>
- C <article>
- D <aside>
- E <section>

**Questão 49**

Em Java, uma *composition*

- A é um conjunto bem definido de interfaces e classes para representar e tratar grupos de dados como uma única unidade.
- B consiste no processo de reconstrução de um objeto a partir de um fluxo de *bytes*.
- C permite a representação de sistemas complexos, concentrando-se em características essenciais e ocultando detalhes de implementação desnecessários.
- D garante que uma classe seja projetada com um propósito único e bem definido.
- E mantém a referência de uma classe dentro de alguma outra classe.

**Questão 50**

Assinale a opção em que é indicado o recurso da linguagem C# que permite a um código inspecionar e manipular sua própria estrutura e comportamento em tempo de execução.

- A IntPtr
- B *list patterns*
- C *reflexion*
- D *lambda expressions*
- E *auto-default structs*

**Questão 51**

Acerca da programação orientada a objetos, assinale a opção correta.

- A A classe é a estrutura que define os atributos e comportamentos dos objetos.
- B Toda linguagem de programação define automaticamente todas as classes necessárias em tempo de compilação.
- C Um objeto não pode ser instanciado a partir de uma classe.
- D A classe representa uma instância específica de um objeto.
- E Atributos e métodos são definidos apenas no objeto, não na classe.

**Questão 52**

A respeito das interfaces na programação orientada a objetos, julgue os itens a seguir.

- I A implementação de uma interface impede que o desenvolvedor declare métodos a ela.
- II Várias interfaces podem ser implementadas em uma classe.
- III A implementação de uma interface dispensa a necessidade de métodos públicos.

Assinale a opção correta.

- A Nenhum item está certo.
- B Apenas o item I está certo.
- C Apenas o item II está certo.
- D Apenas os itens I e III estão certos.
- E Apenas os itens II e III estão certos.

**Questão 53**

Assinale a opção correta em relação ao polimorfismo.

- A O uso de tipos genéricos elimina a necessidade de polimorfismo.
- B No polimorfismo, uma variável do tipo superclasse pode referenciar um objeto da subclasse.
- C O polimorfismo por subtipagem requer que os métodos tenham a mesma assinatura na superclasse e subclasse.
- D A substituição de métodos não afeta o comportamento polimórfico.
- E O polimorfismo impede a aplicação do princípio de substituição, proposto por Liskov.

**Questão 54**

No que se refere à herança na programação orientada a objetos, assinale a opção correta.

- A A herança promove acoplamento fraco entre as classes.
- B A herança elimina completamente a necessidade de composição.
- C A herança permite que uma classe derivada reutilize métodos e atributos da superclasse.
- D A herança permite que uma classe filha acesse diretamente os atributos privados da classe pai.
- E Na herança, uma classe pode herdar múltiplas superclasses diretamente em todas as linguagens.

**Questão 55**

Assinale a opção correta, a respeito de composição na programação orientada a objetos.

- A Na composição forte, objetos compostos não podem existir independentemente de seus componentes.
- B Composição e herança têm exatamente os mesmos efeitos no tempo de execução.
- C A composição estabelece uma relação “é um”.
- D A composição representa uma relação “possui um” ou “é parte de”.
- E A composição é uma forma de herança indireta.

**Questão 56**

Acerca do polimorfismo, julgue os itens a seguir.

- I O uso de polimorfismo reduz a duplicação de código visto que permite abstrações genéricas.
- II Interfaces são frequentemente utilizadas na aplicação de polimorfismo por contrato.
- III Métodos polimórficos devem ser declarados exclusivamente como privados.

Assinale a opção correta.

- A Apenas o item II está certo.
- B Apenas o item III está certo.
- C Apenas os itens I e II estão certos.
- D Apenas os itens I e III estão certos.
- E Todos os itens estão certos.

**Questão 57**

Assinale a opção correta no que se refere a interfaces e classes abstratas.

- A O uso de interfaces impede o uso de herança.
- B Uma interface define o comportamento interno de uma classe.
- C Em qualquer versão do Java, uma interface pode conter implementações concretas de métodos.
- D Classes abstratas não podem conter métodos concretos.
- E Interfaces são adequadas quando se pretende simular múltiplas heranças de comportamento.

**Questão 58**

Considerando que a escolha entre herança e composição impacta diretamente a flexibilidade e o acoplamento de um sistema orientado a objetos, assinale a opção correta.

- Ⓐ Essa escolha não afeta a manutenibilidade do sistema.
- Ⓑ Por modelar comportamentos dinâmicos, a composição permite maior flexibilidade.
- Ⓒ Herança favorece maior desacoplamento que composição.
- Ⓓ A composição dificulta a reutilização de código.
- Ⓔ A herança múltipla é recomendada sempre que possível.

**Questão 59**

Constitui característica fundamental da comunicação assíncrona entre serviços

- Ⓐ a capacidade de enviar e processar mensagens de forma independente, sem que o emissor precise aguardar uma resposta imediata.
- Ⓑ a imposição de tempos de espera fixos para cada mensagem enviada, que garante que todas as mensagens sejam processadas no mesmo intervalo de tempo.
- Ⓒ o bloqueio do fluxo principal da aplicação, que impede que outras tarefas sejam executadas enquanto a resposta é processada.
- Ⓓ o acoplamento rígido entre os serviços, que garante que todas as mensagens sejam entregues de forma síncrona.
- Ⓔ o processamento imediato da resposta dentro da mesma execução do cliente que enviou a requisição.

**Questão 60**

Assinale a opção em que é apresentada característica de um sistema que utiliza filas de mensagens para comunicação entre serviços.

- Ⓐ limitação na quantidade de mensagens que podem ser enviadas, com restrição do fluxo de comunicação entre sistemas
- Ⓑ dependência de um protocolo exclusivo para garantir a comunicação síncrona entre componentes
- Ⓒ capacidade de desacoplar processos, que permite maior flexibilidade e escalabilidade na comunicação entre componentes
- Ⓓ necessidade de que todos os serviços envolvidos na comunicação tenham disponibilidade total e simultânea
- Ⓔ obrigação de processar todas as mensagens no exato instante em que são enviadas, sem possibilidade de armazenamento temporário

**Questão 61**

Em um sistema distribuído que usa eventos para a comunicação entre serviços, uma ação que melhora a escalabilidade da arquitetura é

- Ⓐ a exigência de que cada serviço processador de eventos esteja vinculado a um único emissor, sem possibilidade de assinaturas múltiplas.
- Ⓑ a redução da quantidade de eventos gerados para evitar sobrecarga nos serviços consumidores.
- Ⓒ o envio de eventos de maneira síncrona, para garantir que cada componente processe os dados no mesmo instante em que o evento ocorre.
- Ⓓ a utilização de um barramento de eventos que permita a comunicação assíncrona entre múltiplos serviços interessados na mesma mensagem.
- Ⓔ a imposição de um fluxo fixo e sequencial para cada evento, garantindo-se processamento ordenado sem variações.

**Questão 62**

Em arquiteturas que utilizam mensageria para a comunicação entre sistemas, uma técnica eficiente para a garantia da entrega de mensagens sem perda é

- Ⓐ o processamento imediato de todas as mensagens sem necessidade de recuperação em caso de falha.
- Ⓑ a criação de múltiplos canais simultâneos para envio indiscriminado das mensagens, independentemente da ordem de processamento.
- Ⓒ o armazenamento temporário das mensagens em uma fila persistente até que elas sejam processadas com sucesso.
- Ⓓ o envio repetido de mensagens idênticas sem controle de duplicação para que cheguem ao destino.
- Ⓔ a comunicação direta entre serviços em tempo real, eliminando-se a necessidade de armazenamento intermediário.

**Questão 63**

Segundo o princípio da responsabilidade única (*single responsibility principle*), em boas práticas de desenvolvimento orientado a objetos, uma classe deve

- Ⓐ acumular várias responsabilidades para minimizar a quantidade de código escrito.
- Ⓑ possuir métodos genéricos que possam atender a diversas necessidades sem segregação funcional.
- Ⓒ ser capaz de realizar múltiplas funções relacionadas a diferentes aspectos do sistema.
- Ⓓ implementar diversas funcionalidades para reduzir a necessidade de criar novas classes.
- Ⓔ ter apenas uma razão para ser modificada, garantindo que sua responsabilidade seja única e bem definida.

**Questão 64**

O princípio de inversão de dependência (*dependency inversion principle*) visa tornar o código mais flexível e desacoplado e favorecer sua manutenção e testabilidade. Uma abordagem alinhada a esse princípio consiste em

- Ⓐ evitar o uso de abstrações na comunicação entre módulos para garantir que cada implementação se conecte diretamente com outras classes concretas.
- Ⓑ estabelecer um modelo em que módulos de alto nível interajam de maneira indireta com módulos de baixo nível de forma a garantir que a comunicação entre eles ocorra por meio de abstrações.
- Ⓒ permitir que módulos de alto nível dependam diretamente de módulos de baixo nível para garantir execução eficiente.
- Ⓓ estabelecer dependência direta entre classes concretas para facilitar a implementação e evitar abstrações desnecessárias.
- Ⓔ criar dependências explícitas entre os componentes do sistema para eliminar a necessidade de interfaces.

**Questão 65**

Durante o desenvolvimento de sistemas que exigem controle sobre a criação de instâncias, o padrão de projeto cuja adoção pode evitar múltiplos objetos redundantes e facilitar o acesso centralizado caracteriza-se por

- Ⓐ controlar a criação de instâncias e fornecer um único ponto de acesso global à classe.
- Ⓑ implementar um mecanismo de destruição automática da instância após cada uso.
- Ⓒ criar múltiplas instâncias da mesma classe para otimizar o desempenho do sistema.
- Ⓓ permitir que diferentes partes do código acessem instâncias distintas da mesma classe.
- Ⓔ utilizar herança para garantir que todas as subclasses tenham múltiplas instâncias.

**Questão 66**

O padrão de projeto *factory method*, utilizado para delegar a criação de objetos a subclasses e permitir maior flexibilidade e desacoplamento,

- A garante que todas as instâncias criadas sejam idênticas e compartilhadas globalmente.
- B impede a criação de novos objetos, limitando a reutilização de instâncias existentes.
- C cria objetos diretamente na classe principal, sem necessidade de subclasses.
- D estabelece um mecanismo baseado em abstrações para a criação de objetos, o que possibilita que subclasses escolham qual implementação concreta utilizar.
- E utiliza um único método estático para criar todas as instâncias do sistema.

**ENGENHARIA DE SOFTWARE****Questão 67**

Considerando que, na engenharia de *software*, há várias formas de se escrever uma especificação de requisitos de sistema, assinale a opção em que é apresentada a abordagem baseada no uso de uma linguagem de programação com características mais abstratas, para especificar os requisitos, definindo um modelo operacional do sistema.

- A notações gráficas
- B especificações matemáticas
- C linguagem de descrição de projeto
- D sentenças em linguagem natural
- E linguagem natural estruturada

**Questão 68**

Na utilização do padrão MVC (*model-view-controller*) em uma arquitetura de aplicação *web*, cabe ao *model*

- A encapsular o estado de aplicação e o banco de dados.
- B gerir a lógica específica da aplicação e enviar eventos de usuário ao *controller*.
- C validar os dados de entrada e gerenciar os formulários *web*.
- D notificar a visão de mudanças de estado e o processamento de solicitações HTTP.
- E gerar as páginas dinamicamente e responder às solicitações de atualização dos dados.

**Questão 69**

```
class ContaBanrisul {
    consultarSaldo() {}
}

class ContaCorrente extends ContaBanrisul {
    consultarSaldo() { return "Conta Corrente:
R$ 22.000,01"; }
}

class PrevidenciaPrivada extends ContaBanrisul {
    consultarSaldo() { return "Previdência:
R$ 120.000,53"; }
}

const contaCorrente = new ContaCorrente();
const previdencia = new PrevidenciaPrivada();

console.log(contaCorrente.consultarSaldo());
console.log(previdencia.consultarSaldo());
```

Com base no código precedente, em JavaScript, e nos conceitos da orientação a objetos, é correto afirmar que, nas duas últimas linhas, `consultarSaldo()` constitui

- A uma agregação na sobrecarga do método.
- B uma composição na classe estática `ContaBanrisul`.
- C uma herança, com encapsulamento privado.
- D um polimorfismo.
- E uma herança, com encapsulamento protegido.

**Questão 70**

Assinale a opção em que é apresentada a estratégia para teste de integração de sistemas orientados a objetos com base em sequência de execução que integra o conjunto de classes necessárias para responder a uma entrada onde cada sequência de execução é integrada e testada individualmente.

- A *class partition based testing*
- B *software process based testing*
- C *unitary cluster testing*
- D *usage-based testing*
- E *thread-based testing*

**Questão 71**

Assinale a opção em que é apresentado o princípio de segurança do DevSecOps segundo o qual é recomendável a implementação de controles de segurança no início do ciclo de desenvolvimento de *software* para melhorar a postura de segurança, incluídas práticas como revisões de código, verificação de vulnerabilidades e testes de segurança para detectar e corrigir *bugs* rapidamente.

- A *infrastructure as code automation*
- B *shift left*
- C *system of continuous monitoring*
- D *continuous integration*
- E *continuous delivery*

**Questão 72**

Assinale a opção em que são apresentados os princípios da segurança da informação que foram violados, respectivamente, nas seguintes situações hipotéticas: *i*) um funcionário de uma agência bancária acessou informações pessoais de clientes, como saldos e dados de contato, sem justificativa profissional, utilizando essas informações para fins pessoais, compartilhando-as com terceiros; *ii*) durante o processamento de uma transferência bancária, um funcionário mal-intencionado alterou deliberadamente o valor da transação, o que resultou em registros inconsistentes e potencial perda financeira para o banco.

- A autenticidade e irretratabilidade
- B irretratabilidade e autenticidade
- C confidencialidade e integridade
- D integridade e não repúdio
- E não repúdio e confidencialidade

**ARQUITETURA****Questão 73**

Acerca da API RESTful, julgue os itens subsequentes.

- I API RESTful é uma interface que fornece dados em um formato padronizado baseado em requisições HTTP.
- II A interface uniforme indica que o servidor transfere informações em formato-padrão, sendo o recurso formatado denominado representação em REST.
- III Um dos benefícios das API RESTful é a independência, o que possibilita a escrita de aplicações de cliente e servidor em várias linguagens de programação sem afetar o *design* da API.

Assinale a opção correta.

- A Apenas o item II está certo.
- B Apenas o item III está certo.
- C Apenas os itens I e II estão certos.
- D Apenas os itens I e III estão certos.
- E Todos os itens estão certos.

**Questão 74**

Assinale a opção em que é apresentada a denominação dada ao método HTTP que permite atualizar recursos existentes no servidor e cujo envio da mesma solicitação várias vezes em um serviço RESTful da Web tem o mesmo resultado.

- A PATCH
- B OAuth
- C GET
- D POST
- E PUT

**Questão 75**

A respeito da arquitetura em camadas, julgue os itens que se seguem.

- I O conceito de separação das preocupações facilita a criação de funções eficientes e de modelos de responsabilidade dentro da arquitetura.
- II O padrão *sinkhole* da arquitetura é um antipadrão que ocorre quando as solicitações passam de camada em camada, sem nenhuma lógica de negócio realizada dentro de cada uma das camadas.
- III Camada aberta significa que as alterações feitas em uma camada da arquitetura normalmente não impactam os componentes nas outras camadas.
- IV A implementabilidade e a testabilidade são muito bem suportadas no estilo da arquitetura em camadas.

Assinale a opção correta.

- A Apenas o item I está certo.
- B Apenas o item IV está certo.
- C Apenas os itens I e II estão certos.
- D Apenas os itens II e III estão certos.
- E Apenas os itens III e IV estão certos.

**Questão 76**

A respeito da arquitetura orientada a eventos, julgue os próximos itens.

- I Na topologia do mediador da arquitetura orientada, o fluxo de mensagens é distribuído nos componentes do processador do evento em uma transmissão em cadeia.
- II Uma boa prática na topologia *broker* consiste em enviar cada evento iniciador para uma fila de eventos iniciadores, que é aceita pelo mediador do evento.
- III O aumento da capacidade de resposta do sistema e a dificuldade de lidar com as condições de erro constituem, respectivamente, uma vantagem e uma desvantagem da comunicação assíncrona.
- IV Características como desempenho, escalabilidade e tolerância a falhas orientam o uso da arquitetura.

Estão certos apenas os itens

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

**Questão 77**

Em relação à arquitetura de microsserviços, julgue os itens a seguir.

- I Os microsserviços formam uma arquitetura distribuída, na qual cada serviço roda em seu próprio processo.
- II A finalidade, as transações e a coreografia são diretrizes que os arquitetos podem usar para encontrar os limites certos de serviços nos microsserviços.
- III No estilo de comunicação de orquestração, não há nenhum coordenador central para controlar a sequência de chamadas entre microsserviços, sendo a comunicação geralmente assíncrona.

Assinale a opção correta.

- A Apenas o item I está certo.
- B Apenas o item III está certo.
- C Apenas os itens I e II estão certos.
- D Apenas os itens II e III estão certos.
- E Todos os itens estão certos.

**TECNOLOGIAS EMERGENTES****Questão 78**

De acordo com os conceitos que dão base à inteligência artificial (IA), existe uma entidade autônoma que se caracteriza por perceber seu ambiente através de sensores, atuar sobre esse ambiente por meio de atuadores, processar internamente dados percebidos e conhecimentos armazenados, e selecionar ações específicas que maximizam sua medida de desempenho esperada, conforme definido pela sequência de percepções e seu conhecimento interno. Essa entidade é definida na IA como

- A processo de decisão sequencial.
- B rede neural.
- C agente racional.
- D algoritmo evolutivo.
- E modelagem cognitiva.

**Questão 79**

Um sistema de inteligência artificial foi desenvolvido para dirigir um veículo em um jogo de corrida virtual. O sistema começa sem conhecimento prévio e não recebe exemplos rotulados de como dirigir corretamente. Durante o aprendizado, ele não identifica ou utiliza padrões da pista ou do comportamento de outros veículos. Sua única fonte de informação são as pontuações: pontos positivos ao manter o veículo na pista e fazê-lo completar voltas, pontos negativos quando o veículo sai da pista ou colide, e bônus quando ultrapassa outro veículo ou completa mais rápido a corrida. O sistema ajusta seu comportamento baseando-se unicamente nessas pontuações recebidas após suas ações.

Na situação hipotética precedente, é caracterizado o aprendizado de máquina

- A passivo.
- B por transferência.
- C supervisionado.
- D não supervisionado.
- E por reforço.

**Questão 80**

Nas IA generativas baseadas em arquiteturas *transformer*, com mecanismo de atenção, o modelo consegue relacionar diferentes palavras para produzir texto coerente; ao gerar, por exemplo, a frase “Maria foi ao mercado porque ela precisava comprar leite”, o modelo deve lembrar que “ela” se refere a “Maria”. Com base nesse contexto, assinale a opção correta.

- Ⓐ Modelos de IA que geram texto processam simultaneamente todas as palavras de saída, sem considerar a ordem em que elas aparecem.
- Ⓑ Na geração de texto palavra por palavra, a cada nova palavra gerada, o modelo de IA deve dedicar a atenção às palavras que já foram geradas anteriormente.
- Ⓒ A principal função do mecanismo de atenção é verificar a gramática do texto, não suas relações semânticas.
- Ⓓ O mecanismo de atenção distribui o mesmo peso para todas as palavras anteriores, sem priorizar palavras quanto à relevância no contexto.
- Ⓔ O mecanismo de atenção cria conexões para relações entre palavras adjacentes, não conectando palavras que estão distantes entre si no texto.

**Espaço livre**