INSTITUTO FEDERAL
Rio Grande do Norte

Concurso Público para Provimento de Cargos de Professor da Carreira do Magistério de Ensino Básico, Técnico e Tecnológico do Quadro Permanente de Pessoal do IFRN.

EDITAL Nº 01/2025

EBTT CONCURSO PÚBLICO
PROFESSOR DE ENSINO BÁSICO TÉCNICO E TECNOLÓGICO

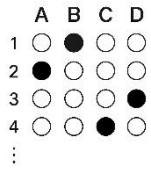
FUNCERN
FUNDAÇÃO DE APOIO AO IFRN

CADERNO DE PROVAS

EBTT – P24	SISTEMAS DE INFORMAÇÃO
Edital Nº. 01/2025 – REITORIA/IFRN	**Data: ___/___/___**

INSTRUÇÕES GERAIS PARA A REALIZAÇÃO DA PROVA

- ☑ Use apenas caneta esferográfica de tinta na cor preta e fabricada em material transparente.
- ☑ Escreva a data, a sua assinatura e o seu número de inscrição no espaço indicado nesta capa.
- ☑ A prova terá duração máxima de 4 (quatro) horas, incluindo o tempo para responder a todas as questões do **Caderno de Provas** e preencher as **Folhas de Respostas**.
- ☑ O tempo mínimo de permanência na sala de provas é de 1 (uma) hora.
- ☑ Antes de retirar-se **definitivamente** da sala, entregue as **Folhas de Respostas** e o **Caderno de Provas** ao fiscal.
- ☑ Este **Caderno de Provas** contém, respectivamente, 30 (trinta) questões Objetivas e 01 (uma) questão Discursiva.
- ☑ Se o **Caderno de Provas** contiver alguma imperfeição gráfica que impeça a leitura, comunique isso imediatamente ao fiscal, para que seja efetuada de imediato a troca do Caderno.
- ☑ Cada questão de múltipla escolha apresenta apenas **uma** resposta correta. Para a marcação da opção escolhida na **Folha de Respostas**, pinte completamente o campo correspondente conforme a figura a seguir:



- ☑ Os rascunhos e as marcações feitas neste **Caderno de Provas** não serão considerados para efeito de avaliação.
- ☑ Interpretar as questões faz parte da avaliação; portanto, não é permitido solicitar esclarecimentos aos fiscais.
- ☑ O preenchimento das **Folhas de Respostas** é de sua inteira responsabilidade.
- ☑ A quantidade de questões e respectivas pontuações deste Caderno de Provas estão apresentadas a seguir:

Provas	***Número de questões***	***Pontos***
Objetiva de Legislação do Serviço Público Federal	05 questões	70 pontos
Objetiva de Conhecimentos Específicos	25 questões	
Discursiva	01 questão	30 pontos
PONTUAÇÃO TOTAL		**100 pontos**

ASSINATURA DO CANDIDATO:	NÚMERO DE INSCRIÇÃO: _____

PROVA DE LEGISLAÇÃO – QUESTÕES DE MÚLTIPLA ESCOLHA

1. Constitui forma de provimento de cargo público, de acordo com a Lei nº 8112/90, a

A) recondução, que consiste no retorno do servidor ao cargo de origem, quando invalidada sua demissão por decisão administrativa ou judicial.

B) reversão, que consiste no retorno à atividade de servidor aposentado, no mesmo cargo ou cargo resultante de sua transformação.

C) readaptação, que consiste no retorno à atividade de servidor aposentado por invalidez, quando junta médica oficial declarar insubsistentes os motivos da aposentadoria.

D) reintegração, que consiste na investidura de servidor em cargo de menor complexidade, quando inabilitado em estágio probatório de cargo efetivo originalmente provido.

2. Dentre as hipóteses de afastamento do servidor, a Lei nº 8112/90 prevê a concessão para atividade política e a licença para tratar de assuntos particulares. Sobre tais atos administrativos, é correto afirmar que

A) o primeiro é ato discricionário e revogável, enquanto o segundo é ato vinculado e irrevogável.

B) ambos são atos discricionários e revogáveis.

C) o primeiro é ato vinculado e irrevogável, enquanto o segundo é ato discricionário e revogável.

D) ambos são atos vinculados e irrevogáveis.

3. Segundo a Lei nº 8027/90, a conduta do servidor de se ausentar da repartição, durante o expediente, sem prévia autorização da chefia imediata, enseja a aplicação de pena de

A) advertência.

B) multa.

C) demissão.

D) suspensão.

4. O Decreto nº 1171/94 aprova o Código de Ética Profissional do Servidor Público Civil do Poder Executivo Federal. Segundo esse decreto, é dever do servidor público

A) contribuir para a previdência social com a alíquota definida em decreto, tendo em vista garantir a futura aposentadoria.

B) declarar, anualmente, os bens e fontes de renda que estejam em nome do servidor público, excluindo os de seu cônjuge e dependentes.

C) participar de cursos de capacitação e ações de desenvolvimento oferecidos pela administração pública, visando aprimorar suas habilidades e conhecimentos

D) comunicar, imediatamente, aos seus superiores, todo e qualquer ato ou fato contrário ao interesse público ou que possa comprometer o serviço.

5. De acordo com a Lei nº 12772/2012, que trata da carreira do Magistério do Ensino Básico, Técnico e Tecnológico (EBTT), a progressão funcional será concedida em conformidade com

A) a obtenção de títulos acadêmicos em conjunto com a participação em cursos de aperfeiçoamento e em projetos de pesquisa e extensão.

B) a convergência de interstício de tempo na classe e a aprovação em avaliação de desempenho individual.

C) a avaliação unilateral da chefia imediata, baseada em critérios de reconhecimento do trabalho do servidor.

D) o tempo de efetivo exercício na classe atual, com mínimo de 36 meses, independentemente, de avaliação da chefia imediata.

PROVA DE CONHECIMENTOS ESPECÍFICOS – QUESTÕES DE MÚLTIPLA ESCOLHA

6. Uma Lista Duplamente Ligada (ou Lista Duplamente Encadeada) é uma estrutura de dados em que cada nó (node) da lista contém referências tanto ao nó anterior quanto ao nó seguinte. O nó anterior ao primeiro elemento da lista é definido como nulo (null), assim como o nó posterior ao último elemento. Essa característica permite uma navegação bidirecional ao longo da lista. Além disso, essa estrutura de dados oferece flexibilidade para inserir novos elementos em diferentes posições: no início da lista, em um ponto intermediário (após um nó específico) ou no final da lista.

Considere que os códigos apresentados a seguir estão implementados na linguagem de programação Java. Além disso, considere o construtor da classe Node, representando o nó, e o trecho inicial da classe Lista, conforme descritos nas figuras 1 e 2, respectivamente.

```java
public Node(int valor){
    this.valor = valor;
    this.proximo = null;
    this.anterior = null;
}
```

Figura 1

```java
public class Lista{
    private Node inicio;

    public Lista(){
        this.inicio = null;

    }
```

Figura 2

Diante do exposto, marque a opção que contém, na linguagem Java, o método para adicionar um novo nó (Node) ao final da Lista (método da classe Lista).

A)

```java
public void inserirNoFim(int valor){
    Node novo = new Node(valor);
    if(this.inicio != null){
        this.inicio = novo;
    }else{
        Node aux = inicio.getProximo();
        while(aux.getProximo() == null){
            aux = aux.getAnterior();
        }
        aux.setProximo(aux);
        novo.setAnterior(novo);
    }
}
```

B)

```java
public void inserirNoFim(int valor){
    Node novo = new Node(valor);
    if(this.inicio == null){
        this.inicio = novo;
    }else{
        Node aux = inicio.getProximo();
        while(aux.getProximo() != null){
            aux = aux.getAnterior();
        }
        aux.setProximo(aux);
        novo.setAnterior(novo);
    }
}
```

C)

```
public void inserirNoFim(int valor){
    Node novo = new Node(valor);
    if(this.inicio != null){
        this.inicio = novo;
    }else{
        Node aux = this.inicio;
        while(aux.getProximo() == null){
            aux = aux.getAnterior();
        }
        aux.setProximo(novo);
        novo.setAnterior(aux);
    }
}
```

D)

```
public void inserirNoFim(int valor){
    Node novo = new Node(valor);
    if(this.inicio == null){
        this.inicio = novo;
    }else{
        Node aux = this.inicio;
        while(aux.getProximo() != null){
            aux = aux.getProximo();
        }
        aux.setProximo(novo);
        novo.setAnterior(aux);
    }
}
```

7. Considerando os conceitos de Programação Orientada a Objetos na Linguagem Java, assinale a opção correta.

A) Uma classe concreta em Java deve conter métodos concretos, ou seja, nenhum método abstrato.

B) Uma herança múltipla em Java ocorre utilizando interfaces diferentes, onde cada interface herdada deve estar em um pacote diferente da classe concreta.

C) Uma herança múltipla em Java ocorre utilizando diversas classes, ou seja, é possível realizar herança ao herdar de mais de uma classe.

D) Um método abstrato em Java pode ter apenas sua assinatura em uma classe concreta.

8. Observe o código Python a seguir.

```
def metodo_ordenacao(array):
    n = len(array)
    for i in range(n):

        menor_indice = i
        for j in range(i + 1, n):
            if array[j] < array[menor_indice]:
                menor_indice = j

        array[i], array[menor_indice] = array[menor_indice], array[i]

array = [64, 25, 12, 22, 11]
print("array original:", array)

metodo_ordenacao(array)
print("arrayay ordenado:", array)
```

A partir do código apresentado, que implementa um algoritmo de ordenação, a função metodo_ordenacao que implementa o algoritmo é

A) Merge Sort.

B) Quick Sort.

C) Bubble Sort.

D) Selection Sort.

9. Mediana é o valor que separa a metade maior e a metade menor de uma amostra. Em termos mais simples, mediana pode ser o valor do meio de um conjunto de dados. No sistema gerenciador de banco de dados Postgres, é possível utilizar uma função de agregação para calcular a mediana de um conjunto de dados em SQL.

Com base nas informações apresentadas, a função de agregação que deve ser utilizada para calcular a mediana de um conjunto de dados é a

A) MEDIAN().

B) PERCENTILE_CONT(0.5).

C) AVERAGE().

D) MEDIAN_VALUE(0.5).

10. Uma loja online estabeleceu que os *K* clientes que mais compraram entre os meses de maio e dezembro de 2024 receberão 30% de desconto em suas próximas compras.
Como administrador do banco de dados da loja online, você foi encarregado de coordenar a equipe para desenvolver uma consulta SQL que atenda a essa demanda. O sistema gerenciador de banco de dados utilizado pela loja é o Postgres e as tabelas seguintes estão disponíveis para elaboração da consulta

- *cliente: cliente_id, nome, desconto*
- *compra: compra_id, cliente_id, data_compra*
- *itens: item_id, compra_id, valor*

Um cliente pode realizar várias compras, e cada compra pode conter um ou mais itens.
Diante da situação apresentada, marque a opção que mostra a consulta SQL que resolve o problema da loja online.

A)

```sql
UPDATE cliente
SET desconto = 30
WHERE cp.data_compra BETWEEN '2024-05-01' AND '2024-12-31'
    AND cliente_id IN (
        SELECT c.cliente_id
        FROM cliente c
        JOIN compra cp ON c.cliente_id = cp.cliente_id
        JOIN itens i ON cp.compra_id = i.compra_id
        GROUP BY c.cliente_id
        ORDER BY SUM(i.valor) DESC
        LIMIT K
);
```

B)

```sql
WITH total_gastos_por_cliente AS (
    SELECT
        c.cliente_id,
        SUM(i.valor) AS total_gasto
    FROM cliente c
    JOIN compra cp ON c.cliente_id = cp.cliente_id
    JOIN itens i ON cp.compra_id = i.compra_id
    WHERE cp.data_compra >= '2024-05-01'
    GROUP BY c.cliente_id
)
UPDATE cliente
SET desconto = 30
WHERE cliente_id IN (
    SELECT cliente_id
    FROM total_gastos_por_cliente
    ORDER BY total_gasto DESC
    LIMIT K
);
```

C)

```
WITH total_gastos_por_cliente AS (
    SELECT
        c.cliente_id,
        SUM(i.valor) AS total_gasto
    FROM cliente c
    JOIN compra cp ON c.cliente_id = cp.cliente_id
    JOIN itens i ON cp.compra_id = i.compra_id
    WHERE cp.data_compra BETWEEN '2024-01-01' AND '2024-12-30'
    GROUP BY c.cliente_id
)
UPDATE cliente
SET desconto = 30
WHERE cliente_id IN (
    SELECT cliente_id
    FROM total_gastos_por_cliente
    ORDER BY total_gasto DESC
    LIMIT K
);
```

D)

```
WITH total_gastos_por_cliente AS (
    SELECT
        c.cliente_id,
        SUM(i.valor) AS total_gasto
    FROM cliente c
    JOIN compra cp ON c.cliente_id = cp.cliente_id
    JOIN itens i ON cp.compra_id = i.compra_id
    WHERE cp.data_compra BETWEEN '2024-05-01' AND '2024-12-31'
    GROUP BY c.cliente_id
)
UPDATE cliente
SET desconto = 30
WHERE cliente_id IN (
    SELECT cliente_id
    FROM total_gastos_por_cliente
    ORDER BY total_gasto DESC
    LIMIT K
);
```

11. Quando um usuário é criado no banco de dados Postgres por meio do comando "*CREATE USER nome;*", ele automaticamente recebe permissão para se conectar ao sistema gerenciador de banco de dados. A consulta que tem resultado equivalente a *"CREATE USER nome"* é

A) CREATE ROLE nome USER;

B) CREATE ROLE nome PRIVILEGES;

C) CREATE ROLE nome LOGIN;

D) CREATE ROLE nome NEW_USER;

12. Analise as sentenças a seguir sobre Mapeamento Objeto-Relacional utilizando o ORM SQLAlchemy e Python e marque a opção correta.

A) A classe *BaseModel* do ORM *SQLAlchemy* serve como a classe base na qual todos os modelos ORM do SQLAlchemy serão definidos. Isso permite que as classes Python sejam associadas diretamente às tabelas do banco de dados, simplificando a criação de um mapeamento.

B) A *Engine* no *SQLAlchemy* é uma classe que pode instanciar uma conexão de dados, mantendo essa conexão dentro de uma *Session* para reutilização rápida. Dessa forma, o acesso aos atributos também ocorre com maior rapidez.

C) As instruções SELECT são produzidas pela função *select()* que retorna um objeto Select. As entidades e expressões SQL a serem retornadas são passadas posicionalmente para a função. A partir daí, métodos adicionais são usados para gerar a instrução completa, como *Select.where().*

D) As cláusulas *Select.order_by(), Select.group_by(),* em *SQLAlchemy,* são métodos para ordenar e agrupar por atributos específicos de uma tabela. Esses atributos são definidos a priori pela cláusula *parameter_by* para que possa ser associada às funções de ordenação e agrupamento.

13. Considerando as técnicas de validação de software é correto afirmar que a técnica de

A) análise de valor limite explora os limites dos valores, cria casos de teste mais eficientes e ajuda, ainda, a perceber se há uma maior chance de erros quando as entradas são referentes ao limite do domínio.

B) classe de equivalência é utilizada para ampliar o número de casos de teste a um nível de garantir sua cobertura. Todas as combinações de possíveis dados de entrada são divididas em blocos de cobertura.

C) tabela de decisão define quais ações serão executadas pelo sistema quando não houver um conjunto de entradas predeterminado. As regras de negócio do sistema podem ser descritas em uma tabela cujas colunas representam diferentes tipos de entradas.

D) transição de estados é similar à técnica de análise do valor limite; no entanto, o foco é na transição dos comportamentos e não nas entradas do sistema. Assim, casos de testes são baseados nas possíveis saídas que causam uma mudança de comportamento no sistema.

14. Considere o código abaixo, em Python, que utiliza a técnica de classe de equivalência e análise de valor limite para elaboração de teste de unidade.

```python
import unittest

def verificar_idade(idade):
    if not isinstance(idade, int):
        raise TypeError("Idade deve ser um número inteiro.")
    if 18 <= idade <= 60:
        return "Idade válida"
    else:
        return "Idade inválida"

class TesteClasseEquivalencia(unittest.TestCase):
    def test_verificando_idades(self):
        self.assertEqual(verificar_idade(18), "Idade válida")
        self.assertEqual(verificar_idade(60), "Idade válida")
        self.assertEqual(verificar_idade(35), "Idade válida")

    def test_verificando_idades_novamente(self):
        self.assertEqual(verificar_idade(17), "Idade inválida")
        self.assertEqual(verificar_idade(61), "Idade inválida")
        self.assertEqual(verificar_idade(-1), "Idade inválida")

    def test_tipos_invalidos(self):
        with self.assertRaises(TypeError):
            verificar_idade("vinte")
        with self.assertRaises(TypeError):
            verificar_idade(None)
        with self.assertRaises(TypeError):
            verificar_idade(19.5)

if __name__ == '__main__':
    unittest.main()
```

Nesse contexto, é correto afirmar que os testes implementam, respectivamente,

A) 1 classe de equivalência com 2 valores limites.
B) 2 classes de equivalência: uma com 2 valores limites, e outra com 3 valores limites.
C) 2 classes de equivalência: uma com 1 valor limite, e outra com 3 valores limites.
D) 1 classe de equivalência com 1 valor limite.

15. O Django REST Framework simplifica a criação de APIs robustas e eficientes por meio das classes de *viewsets*, que integra a lógica de múltiplas visualizações relacionadas em uma única classe. Imagine que você está desenvolvendo uma aplicação para gerenciar produtos em um sistema de ponto de venda. Com o seu uso, é possível configurar endpoints que implementam todas as operações padrão de uma API REST (GET, POST, PUT, PATCH e DELETE) de forma prática, organizada e consistente. Além disso, as operações de *CRUD (Create, Read, Update, Delete)* do modelo Produto podem ser centralizadas em uma única estrutura, garantindo melhor reutilização de código e facilitando a manutenção ao longo do ciclo de vida do projeto.

Considerando a situação apresentada, o código que implementa corretamente os *endpoints* de Produto na classe ***ProdutoViewSet*** é:

A)

```python
from rest_framework import viewsets # type: ignore
from .models import Produto # type: ignore
from .serializers import ProdutoSerializer # type: ignore

class ProdutoViewSet(viewsets.ModelViewSet):
    queryset = Produto.objects.all().generate() # type: ignore
    serializer_class = ProdutoSerializer
```

B)

```python
from rest_framework import viewsets # type: ignore
from .models import Produto # type: ignore
from .serializers import ProdutoSerializer # type: ignore

class ProdutoViewSet(viewsets.ViewSet):
    queryset = Produto.objects.all(Produto) # type: ignore
    serializer_class = ProdutoSerializer
```

C)

```python
from rest_framework import viewsets # type: ignore
from .models import Produto # type: ignore
from .serializers import ProdutoSerializer # type: ignore

class ProdutoViewSet(viewsets.ViewSet):
    queryset = Produto.objects.all().generate() # type: ignore
    serializer_class = ProdutoSerializer
```

D)

```python
from rest_framework import viewsets # type: ignore
from .models import Produto # type: ignore
from .serializers import ProdutoSerializer # type: ignore

class ProdutoViewSet(viewsets.ModelViewSet):
    queryset = Produto.objects.all()
    serializer_class = ProdutoSerializer
```

16. O desenvolvimento através de ReactJS considera a utilização de componentes para melhorar a modularização e manutenção das aplicações *Frontend*. Os componentes ReactJS podem ser reutilizados, facilitando a construção da aplicação.

Considerando o desenvolvimento de uma aplicação para gerenciar *Produtos* em um sistema de ponto de venda, o código de um componente react para renderizar os produtos em uma tabela é:

A)

```
import React from 'react';

const TabelaProdutos = ({ produtos }) => {
    return (
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nome</th>
                    <th>Preço</th>
                </tr>
            </thead>
            <tbody>
                {produtos.map(produto => (
                    <tr key={produto.id}>
                        <td>{produto.id}</td>
                        <td>{produto.nome}</td>
                        <td>{produto.preco}</td>
                    </tr>
                ))}
            </tbody>
        </table>
    );
};

export default TabelaProdutos;
```

B)

```
import React from 'react';

const TabelaProdutos = ({ produtos }) => {
    return (
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nome</th>
                    <th>Preço</th>
                </tr>
            </thead>
            <tbody>
                {produtos.map(produto :: (
                    <tr>
                        <td>{produto.id}</td>
                        <td>{produto.nome}</td>
                        <td>{produto.preco}</td>
                    </tr>
                ))}
            </tbody>
        </table>
    );
};

export default TabelaProdutos;
```

C)

```
import React from 'react';

const TabelaProdutos = (props) => {
    return (
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nome</th>
                    <th>Preço</th>
                </tr>
            </thead>
            <tbody>
                {produtos.map(produto => (
                    <tr key={produto.id}>
                        <td>{produto.id}</td>
                        <td>{produto.nome}</td>
                        <td>{produto.preco}</td>
                    </tr>
                ))}
            </tbody>
        </table>
    );
};
```

D)

```
import React from 'react';


const TabelaProdutos = (props.produtos) => {
    return (
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nome</th>
                    <th>Preço</th>
                </tr>
            </thead>
            <tbody>
                {produtos.map(produto :: (
                    <tr>
                        <td>{produto.id}</td>
                        <td>{produto.nome}</td>
                        <td>{produto.preco}</td>
                    </tr>
                ))}
            </tbody>
        </table>
    );
};

export default TabelaProdutos;
```

17. Considerando os conceitos de HTML, CSS e JavaScript, assinale a opção correta.

A) JavaScript é uma linguagem de marcação de script usada para adicionar interatividade e lógica às páginas web e também ao CSS.

B) O conceito de "cascata" no CSS define o ciclo de vida de como serão aplicados os estilos nas tags HTML, considerando os tipos e as sequências dos objetos DOM (*Document Object Model*).

C) Funções em CSS são blocos reutilizáveis de código, permitindo executar ações específicas para tags HTML com o objetivo de aplicar diferentes estilos.

D) O DOM (*Document Object Model*) permite que JavaScript manipule elementos HTML dinamicamente, alterando conteúdo, estilos e estrutura da página.

18. Considerando os conceitos de Django Rest Framework, assinale a opção correta.

A) Os *Serializers* em Django REST Framework convertem dados complexos, como objetos do modelo, em uma forma de representação que podem ser facilmente renderizados como JSON ou XML.

B) A utilização de relacionamentos no modelo em Django Rest Framework, como *ForeignKey, OneToOneKey e ManyToManyKey*, tem por objetivo mapear relações do banco de dados, refletindo a lógica do negócio dos modelos na API.

C) As *views* do Django Rest Framework podem ser criadas usando ViewController para controle total da lógica de manipulação das requisições HTTP, como GET, POST, PUT e DELETE.

D) O *Serializer* evita validar dados de entrada enviados pelos clientes. Essa validação deve ser feita no *frontend*, assegurando que eles respeitam os tipos e formatos esperados antes de serem enviados para um modelo.

19. O acrônimo SOLID refere-se a um conjunto de cinco princípios de projeto de sistemas que visam tornar o código mais compreensível, flexível e fácil de manter. Esses princípios são fundamentais para a programação orientada a objetos e ajudam os desenvolvedores a evitar problemas comuns, como a rigidez e a complexidade excessiva. Um dos princípios do acrônimo SOLID é

A) Inversão de Controle.

B) Programação Estruturada.

C) Modularidade.

D) Responsabilidade Única.

20. A avaliação de interação e os testes de usabilidade são fundamentais para garantir que as interfaces sejam intuitivas e eficazes, permitindo que os usuários realizem suas tarefas de forma eficiente e satisfatória. Um dos principais objetivos da avaliação de interação e testes de usabilidade em interfaces humano-computador é

A) aumentar a complexidade das interfaces para usuários avançados.

B) reduzir o tempo de desenvolvimento de software sem considerar a usabilidade.

C) identificar problemas de usabilidade e melhorar a experiência do usuário.

D) focar apenas na estética visual da interface, ignorando a funcionalidade.

21. Os princípios da orientação a objetos são importantes para a análise e projeto de sistemas, permitindo uma melhor organização e reutilização do código. Entre os princípios, destacam-se a abstração, encapsulamento, herança e polimorfismo. Marque a opção que define, corretamente, um desses princípios.

A) O encapsulamento permite que diferentes classes compartilhem a mesma interface, mas implementem comportamentos distintos.

B) A herança permite que uma classe derive características e comportamentos de outra classe, promovendo a reutilização de código.

C) A abstração é o processo de ocultar a complexidade do sistema, permitindo que o usuário interaja apenas com a interface.

D) O polimorfismo consiste na habilidade de um objeto em apresentar diferentes formas, sendo um conceito distinto da herança.

22. No contexto de Arquitetura de Software, os estilos arquiteturais definem padrões de organização e interação entre componentes, influenciando diretamente atributos de qualidade como modularidade, escalabilidade e desempenho. Na descrição correta de um aspecto fundamental do estilo arquitetural Baseado em Eventos (Event-Driven), define-se que

A) a arquitetura é rigidamente estruturada em camadas, onde cada nível só pode se comunicar com a camada imediatamente inferior ou superior.
B) o sistema é projetado para reagir a eventos assíncronos, com componentes comunicando-se por meio de mensagens ou publicações/assinaturas, promovendo baixo acoplamento.
C) os seus componentes compartilham um estado global único, e as alterações são propagadas de forma síncrona e centralizada.
D) a comunicação entre módulos é estritamente bilateral e baseada em chamadas diretas (requisição-resposta), exigindo alta coordenação temporal.

23. No Scrum, a Definição de Pronto (*Definition of Done*, ou DoD) é um critério essencial para garantir que um incremento do produto seja considerado completo. Ele estabelece um conjunto de requisitos que devem ser atendidos antes que uma funcionalidade seja entregue, assegurando qualidade e consistência. No entanto, equipes maduras em Scrum frequentemente enfrentam desafios ao aplicar o DoD em contextos de integração contínua e entrega contínua (CI/CD), em que a automação e a rápida entrega de valor são prioridades.

A descrição correta de uma característica crítica da Definição de Pronto em um ambiente de CI/CD, considerando as melhores práticas do Scrum, implica que o DoD

A) precisa ser flexível e adaptável a cada *Sprint*, permitindo que a equipe ajuste os critérios conforme a complexidade das tarefas.

B) deve incluir verificações automatizadas, como testes de unidade, integração e *deploy* em ambiente de *staging*, para garantir que o incremento seja considerado "pronto".

C) necessita ser definido exclusivamente pelo *Product Owner*, pois ele é o responsável por priorizar o Backlog do Produto.

D) dispensa a inclusão testes automatizados, desde que a equipe realize revisões manuais rigorosas antes da entrega.

24. A interação multimodal combina múltiplos modos de entrada (como toque, voz e gestos) para criar experiências mais intuitivas e acessíveis em interfaces humano-computador. Em ambientes de realidade aumentada (RA), por exemplo, a combinação precisa desses modos é essencial para evitar conflitos de interpretação e garantir respostas coerentes do sistema. Um desafio crítico nesse contexto é a sincronização temporal entre modais, onde atrasos na captura ou processamento da entrada podem degradar a experiência do usuário.

Considerando as melhores práticas em IHC, a estratégia mais eficiente para mitigar problemas de sincronização temporal em sistemas multimodais de RA é

A) priorizar exclusivamente o modal de menor latência e ignorar os demais quando houver discrepância temporal.

B) utilizar um único modal por vez, alternando dinamicamente com base no contexto da interação.

C) implementar buffers de sincronização com janelas de tempo adaptativas, que compensam atrasos variáveis entre modais sem descartar entradas válidas.

D) descartar as entradas multimodais que não sejam recebidas simultaneamente em todos os canais.

25. Em projetos com alta complexidade e requisitos em constante evolução, a escolha da metodologia de gerenciamento impacta diretamente o sucesso da entrega. O Guia *PMBOK* e o Guia do *Scrum* representam abordagens distintas, porém complementares, sendo comum sua combinação em projetos híbridos. Um dos principais desafios nesse contexto é garantir que os *marcos tradicionais* (típicos de metodologias preditivas) sejam harmonizados com as *entregas iterativas* das abordagens ágeis, sem sobrecarregar a equipe ou perder o foco no valor entregue. Dessa forma, a melhor estratégia para equilibrar a necessidade de marcos tradicionais com a flexibilidade das entregas ágeis em um projeto híbrido é

A) manter marcos rígidos baseados em cronogramas tradicionais, exigindo que cada Sprint entregue funcionalidades completas para atender às datas pré-definidas.

B) dispensar os marcos tradicionais e trabalhar apenas com metas de Sprint, sem preocupação com prazos externos e metas pré-definidas.

C) definir marcos baseados em objetivos de negócio (ex.: liberação de MVP) e alinhá-los com as Sprints, usando revisões de Sprint para validar o progresso em direção aos marcos.

D) criar marcos independentes do time ágil, gerenciados separadamente pela alta gestão, sem integração com o backlog do produto.

26. A gestão de riscos é um processo contínuo no gerenciamento de projetos, visando identificar, analisar e responder a potenciais ameaças que podem impactar os objetivos do projeto. Uma ação típica da análise qualitativa de riscos consiste em

A) realizar simulações estatísticas complexas para calcular o impacto financeiro esperado de cada risco.

B) ignorar riscos com baixa probabilidade, embora eles possam ter um alto impacto para o projeto.

C) classificar os riscos em uma matriz de probabilidade e impacto para definir prioridades de tratamento.

D) aplicar soluções de contorno que eliminem todos os riscos identificados antes do início do projeto.

27. Em projetos complexos, a avaliação de desempenho e o registro de lições aprendidas são fundamentais para a melhoria contínua. No entanto, organizações frequentemente enfrentam o desafio de transformar essas percepções em melhorias concretas. Uma abordagem eficaz requer não apenas a identificação e documentação, mas também a integração sistemática desses aprendizados nos processos organizacionais. Estudos indicam que projetos que implementam mecanismos estruturados para aproveitar lições aprendidas têm 30% mais chances de sucesso em iniciativas futuras. Diante disso, uma estratégia que demonstra maior eficácia na institucionalização de lições aprendidas em uma organização que realize o gerenciamento de projetos consiste em

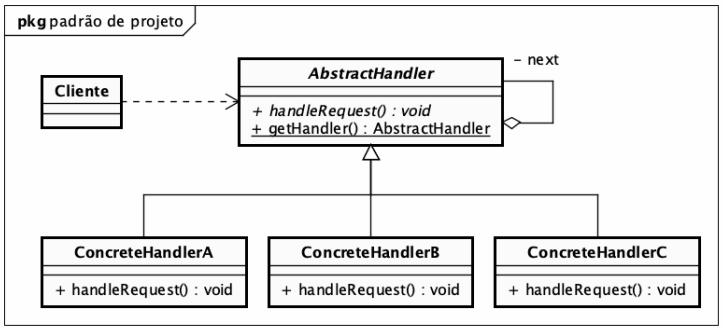
A) implementar um sistema de gestão do conhecimento com revisão obrigatória das lições relevantes durante a inicialização de novos projetos e atualização periódica dos processos organizacionais.

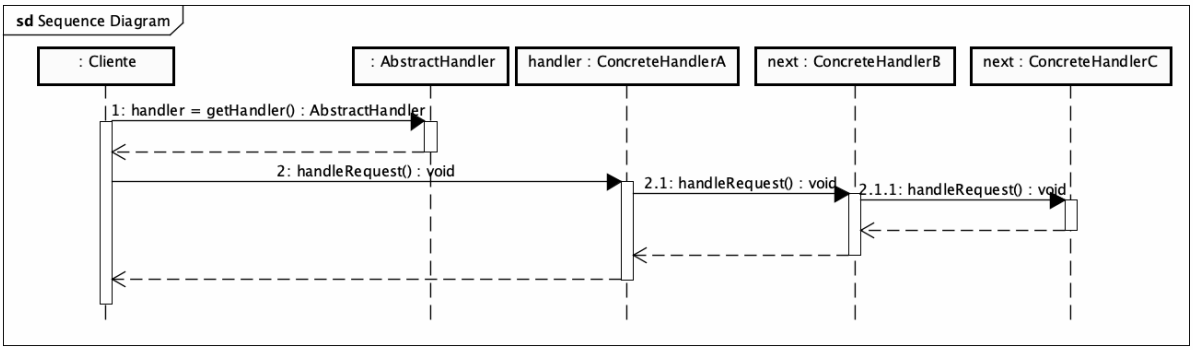
B) armazenar as lições aprendidas em um banco de dados centralizado com acesso livre para consulta, a qualquer tempo, pelos membros da equipe.

C) reunir, anualmente, os gerentes seniores da organização e realizar um evento voltado à revisão e discussão das lições aprendidas nos projetos específicos.

D) distribuir relatórios mensais através de e-mail destinado aos membros das equipes de projeto, contendo as lições aprendidas mais relevantes.

28. Considere os diagramas UML, apresentados a seguir, os quais representam um dado padrão de projeto comportamental. Considere também as seguintes características: em sistemas complexos onde múltiplos objetos podem processar uma requisição, esse padrão é particularmente útil para: (I) desacoplar o remetente de uma requisição de seus receptores, (II) permitir que vários objetos tenham a chance de processar a requisição e (III) compor comportamentos em tempo de execução.





Fonte: Adaptado de Gamma, E., Helm, R., Johnson, R., Vlissides, J., et all. (2000). Padrões de Projetos: Soluções Reutilizáveis de Software Orientados a Objetos. Edição Português. Capa comum.

Com base nos diagramas e nas características apresentadas, a opção que corresponde ao respectivo padrão de projeto é

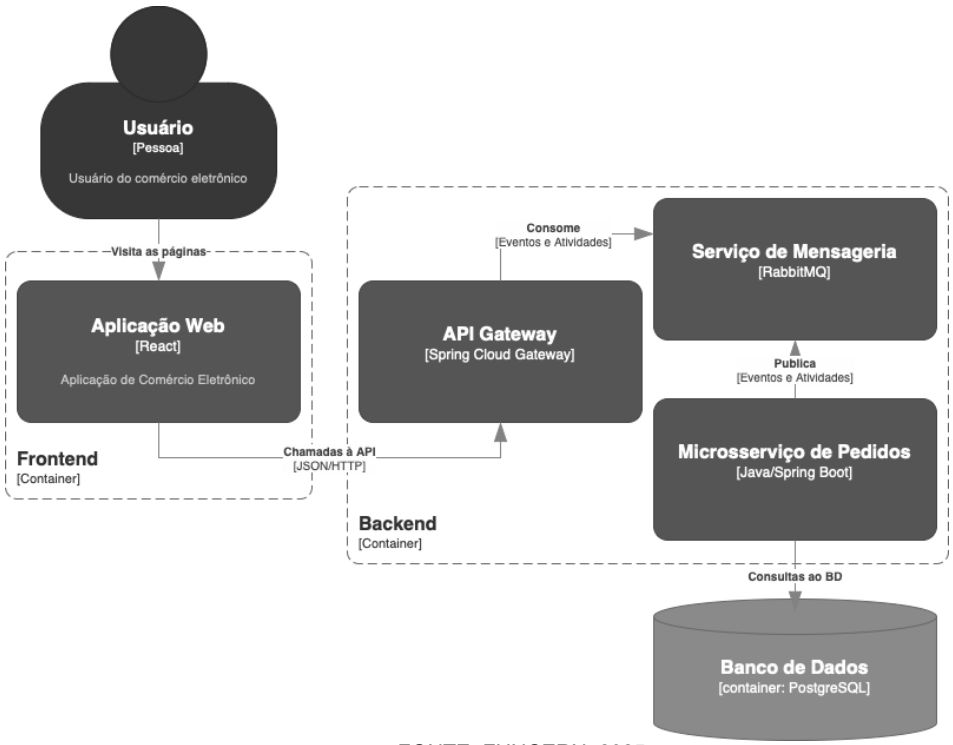
A) Observer.

B) Chain of Responsibility.

C) Strategy.

D) Decorator.

29. O Modelo C4 é uma das abordagens utilizadas para documentação de arquitetura de software. Nessa abordagem são definidos quatro níveis de abstração: Contexto (context), Contêineres (containers), Componentes (components) e Código (code) para representar sistemas complexos de forma escalável. O diagrama a seguir apresenta um modelo C4, o qual foca em tecnologias, protocolos de comunicação e responsabilidades de cada parte do sistema, sendo essencial para arquitetos e desenvolvedores entenderem como as peças se conectam.



FONTE: FUNCERN, 2025.

Considerando as informações fornecidas e o diagrama, o nível do Modelo C4 que está representado no diagrama é o de

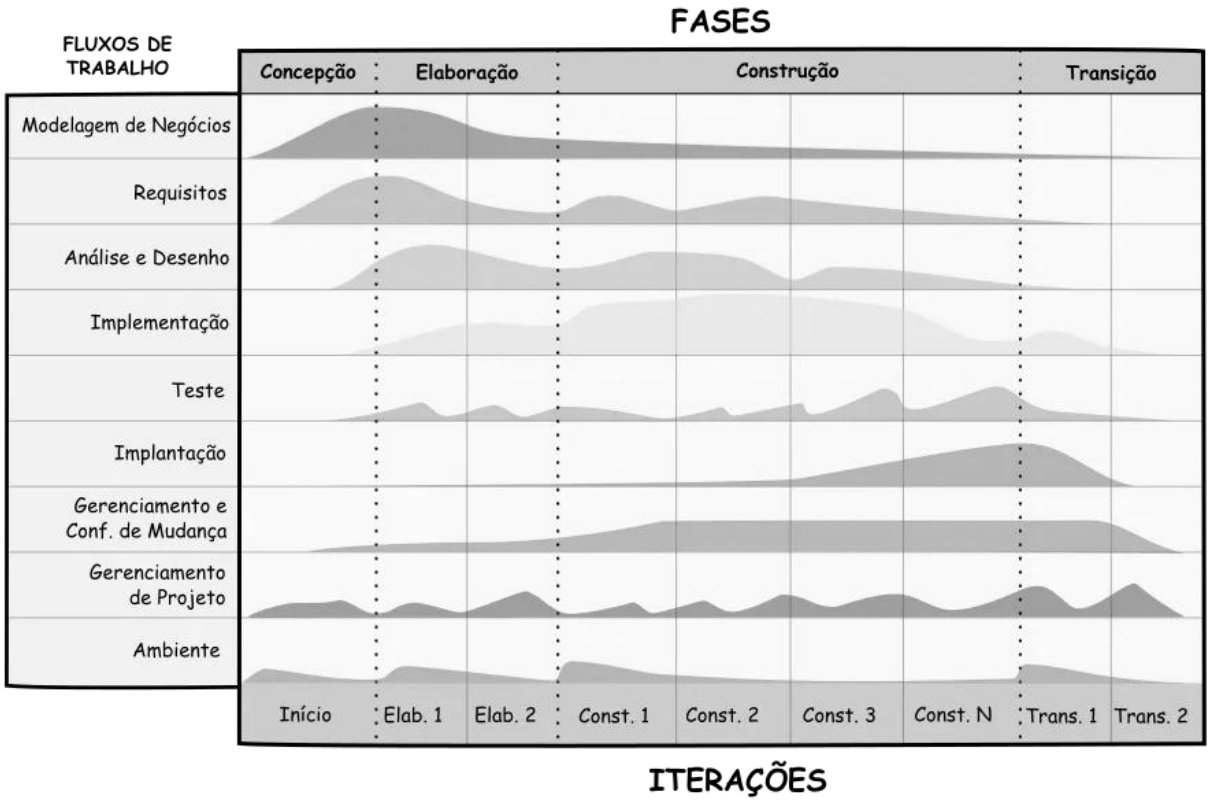
A) Contexto.

B) Contêineres.

C) Componentes.

D) Nível de Código.

30. O Processo Unificado (*Unified Process* ou simplesmente UP) é um modelo de processo de desenvolvimento de software prescritivo, iterativo e incremental, o qual organiza o desenvolvimento em fases e disciplinas. A figura a seguir ilustra o ciclo de vida do UP. Um dos princípios fundamentais do UP é o balanceamento dinâmico entre atividades de diferentes disciplinas ao longo das fases.



FONTE: Adaptado de GUEDES, 2018.

Considerando a figura apresentada, marque a opção que descreve, corretamente, a característica que distingue o UP de outros modelos prescritivos.

A) Os casos de uso direcionam as iterações enquanto a arquitetura é estabilizada progressivamente, com disciplinas sendo enfatizadas de forma não uniforme ao longo das fases. Essa abordagem permite que a maioria dos riscos técnicos sejam mitigados antes da fase de Construção.

B) O UP concentra-se unicamente na criação rápida de protótipos descartáveis durante todas as fases. A arquitetura é intencionalmente deixada em segundo plano, com foco principal na obtenção de feedback imediato dos envolvidos (*stakeholders*) através de demonstrações frequentes.

C) As fases pré-definidas não devem ser rígidas a ponto de não permitir que as disciplinas sejam reorganizadas livremente conforme as necessidades do projeto, principalmente aquelas necessidades apresentadas pelo “dono do produto” (*product owner*).

D) O UP exige que a maior parte dos artefatos sejam elaborados antes do início da codificação. Antes da fase de Construção devem ser finalizados os artefatos, tais como: Documento de Visão, Modelo de Casos de Uso e a Especificação Arquitetural.

PROVA DISCURSIVA – QUESTÃO ÚNICA

Leia o excerto a seguir.

Em seu Projeto Político-Pedagógico (PPP), o IFRN

> assume a necessidade de implementar um processo educativo que desvele práticas mediadoras e emancipatórias, capazes de contemplar, em consonância com o rigor científico e com a omnilateralidade humana, as dimensões culturais, linguísticas, artísticas, sociais, técnicas e tecnológicas. A educação, assim entendida, só é possível se "[...] esforçar-se no sentido da desocultação da realidade. Desocultação na qual o homem existencialize sua real vocação: a de transformar a realidade". (FREIRE, *apud*, IFRN, 2012, p. 48).

Esse posicionamento fundamenta-se tanto nos princípios e nas diretrizes orientadores da prática pedagógica quanto nas concepções de ser humano, de sociedade, de cultura, de ciência, de tecnologia, de trabalho e de educação assumidas pelo IFRN e explicitadas em seu PPP (IFRN, 2012, p. 33-48).

PROPOSTA DE PRODUÇÃO TEXTUAL

Considerando a relevância da discussão sobre currículo integrado no âmbito da Educação Profissional e Tecnológica brasileira, escreva um **texto argumentativo** em que seja apresentado um ponto de vista sobre **a relação entre a formação humana integral fundamentada na omnilateralidade/politecnia e o posicionamento institucional exposto no PPP do IFRN.**

ORIENTAÇÕES ÀS PESSOAS CANDIDATAS

Sua produção textual deverá atender aos seguintes critérios:

- ser redigida no espaço destinado à versão definitiva na Folha de Resposta;
- ser redigida na norma-padrão (linguagem culta) da língua portuguesa escrita;
- ser redigida em prosa (e não em verso);
- conter, no mínimo, 20 (vinte) e, no máximo, 30 (trinta) linhas; e
- não estar assinada (nem mesmo com pseudônimo).

ATENÇÃO!

Será atribuída **NOTA ZERO** à produção textual em **qualquer UM** dos seguintes casos:

- se o espaço destinado ao texto definitivo na Folha de Resposta estiver em branco;
- se for redigida fora do espaço destinado ao texto definitivo na Folha de Resposta;
- se for redigida de forma ilegível;
- se não for redigida com caneta esferográfica de tinta na cor preta;
- se contiver quantidade mínima inferior a 20 (vinte) linhas;
- se fugir ao tema central ou à proposta da questão; e
- se contiver identificação da pessoa candidata fora do espaço reservado para esse fim.

RASCUNHO

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	