-- CONHECIMENTOS ESPECÍFICOS –

Julgue os itens que se seguem, de acordo com os normativos relacionados à Plataforma Digital do Poder Judiciário Brasileiro (PDPJ-Br).

51 Se, no desenvolvimento do sistema informatizado de gestão de processos e documentos de um órgão do Poder Judiciário, for constatado conflito entre um requisito não funcional de segurança e um requisito não funcional de usabilidade, então, conforme a resolução do CNJ que instituiu o MoReq-Jus, o requisito não funcional relacionado à segurança deve ser aplicado em detrimento do requisito relacionado à usabilidade.

52 Se a PDPJ-Br contratar um serviço de computação em nuvem na modalidade de integrador de nuvem (*broker*), não é necessário que o serviço tenha a capacidade de mensurar o uso dos recursos da nuvem de forma individualizada por cliente de cada serviço provido na PDPJ-Br.

53 Se um juiz auxiliar da presidência do CNJ for designado como membro do Comitê Gestor Nacional da PDPJ-Br, então, nesse caso, ele deverá presidir o Comitê.

54 Se um novo serviço for projetado para a PDPJ-Br com o uso do modelo arquitetural de microsserviços, então, conforme normativo do CNJ, na interação entre os microsserviços, deve-se adotar o modelo de coreografia, por meio de um serviço centralizado de mensageria, o que garante o desacoplamento.

Julgue os itens subsequentes, conforme as Portarias CNJ n.º 131/2021 e n.º 162/2021, e as Resoluções CNJ n.º 396/2021 e n.º 468/2022.

55 Se forem implementadas mudanças no código-fonte de uma solução disponibilizada na PDPJ-Br, o código-fonte alterado deverá ser encaminhado para análise do Grupo Revisor de Código-Fonte das soluções da PDPJ-Br e do sistema Processo Judicial Eletrônico (PJe).

56 Na estrutura organizacional de um tribunal regional, conforme a resolução que institui a Estratégia Nacional de Segurança Cibernética do Poder Judiciário (ENSEC-PJ), a estrutura de segurança da informação deverá ser subordinada diretamente à área de TIC do respectivo órgão.

57 Para fins de investigação de ilícito, os registros históricos de eventos (*logs*) devem ser armazenados pelo período mínimo de 12 meses, a fim de atender aos normativos do CNJ.

58 Na contratação relativa a desenvolvimento de *software* em órgão submetido ao controle administrativo e financeiro do CNJ, diante da dificuldade de mensurar o esforço, a adoção da métrica homem-hora somente será permitida caso haja justificativa e vinculação à entrega de produtos de acordo com prazos e qualidade previamente definidos.

Julgue os itens a seguir, relativos a Java, Swagger, PostgreSQL e JSON.

59 O código em Java a seguir, ao ser executado, apresentará `22` como resultado.

```
public class Main {
  public static void main(String[] args) {
    Object valor = "22 TJ PA";
    if (valor instanceof String s) {
      valor = s.replaceAll("[^a-zA-Z]", "");
    }

    switch (valor) {
      case String s -> System.out.println(s);
      case Integer i -> System.out.println(i);
      default -> System.out.println(valor);
    }
  }
}
```

60 O Swagger, quando utiliza a OpenAPI 3.0, permite descrever as APIs protegidas por meio do esquema de segurança *bearer authentication*.

61 Considere o código em Java a seguir, especialmente em relação aos valores descritos na notação JSON.

```
import java.util.Arrays;
import java.util.Collections;
public class JsonSortDescendingExample {
  public static void main(String[] args) {
   String jsonString =
"{\"TJPA\":[\"Tribunal\",\"Justica\",\"Estado\"
,\"Para\"]}";
   String arrayContent =
jsonString.substring(
     jsonString.indexOf("[") + 1,
     jsonString.indexOf("]")
   );
   String[] values =
arrayContent.replaceAll("\"", "").split(",");
   Arrays.sort(values,
Collections.reverseOrder());

   // Exibindo os valores ordenados
   System.out.println("Resultado:");
   for (int i = values.length - 1; i >= 0; i--)
{
   System.out.println(values[i]);
   }
  }
}
```

Esse código, ao ser executado, apresentará o resultado a seguir.

```
Resultado:
Estado
Justica
Para
Tribunal
```

```
CREATE TABLE processos (
   idproc INT PRIMARY KEY,
   descricao VARCHAR(50),
   referencia INT REFERENCES processos(idproc)
);

INSERT INTO processos (idproc, descricao,
referencia) VALUES
(1, 'Gabinete', NULL),
(2, 'Processo A', 1),
(3, 'Processo B', 2),
(4, 'Recurso 1', 3);
```

Considerando que os comandos precedentes tenham sido executados no PostgreSQL 14, julgue o item subsequente.

62 Suponha que, no mesmo banco de dados e imediatamente após a execução dos comandos apresentados, tenha sido executado o código SQL a seguir.

```
WITH RECURSIVE processop AS (
    SELECT idproc, descricao, referencia
    FROM processos
    WHERE referencia IS NOT NULL

    UNION ALL

    SELECT p.idproc, p.descricao, p.referencia
    FROM processos p
    INNER JOIN processop pp ON p.referencia =
pp.idproc
)
SELECT distinct
    idproc,
    descricao,
    (SELECT descricao FROM processos WHERE
idproc = processop.referencia) AS
descricao_pai
FROM processop
ORDER BY idproc;
```

Nessa situação hipotética, após as referidas ações, será apresentado o resultado a seguir.

```
 idproc | descricao  | descricao_pai
--------+------------+---------------
      2 | Processo A | Gabinete
      3 | Processo B | Processo A
      4 | Recurso 1  | Processo B
```

Julgue os próximos itens, relativos a Keycloak, RabbitMQ, GitHub e Kubernetes.

63 As políticas de autorização do Keycloak, em sua versão mais recente, embora não permitam controle de acesso com referência temporal por questões de segurança, permitem o controle de acesso com base no usuário e em atributos.

64 No RabbitMQ, em sua versão mais recente, as filas de quórum devem ser utilizadas especialmente quando envolvem filas temporárias ou transitórias.

65 No GitHub, pode-se utilizar o comando `git stash` caso seja necessário mudar de *branch* por um tempo sem fazer o *commit* de um trabalho incompleto.

66 Kubernetes é uma plataforma unificada nativa que, ainda que não ofereça autenticação centralizada, permite a implantação de *bare metal* e de nuvens públicas com a utilização de contêineres.

Acerca dos modelos de versionamento semântico e dos padrões de *branches*, julgue os itens a seguir.

67 O incremento do número `MAJOR` é obrigatório apenas quando há mudanças que quebram a compatibilidade com versões anteriores, mesmo que a alteração inclua novos recursos sem impacto na API existente.

68 No GitFlow, *branches* do tipo `hotfix` devem ser criadas a partir da *branch main* e, após corrigir o problema, mescladas tanto em *main* quanto em *develop* para garantir sincronização.

Julgue os itens subsequentes a respeito de CI/CD (*continuous integration*/*continuous delivery*).

69 Na etapa de CI, desenvolvedores devem mesclar (*merge*) suas alterações na *branch* principal apenas uma vez por semana para reduzir conflitos e economizar tempo de *build*.

70 O CD exige que toda alteração no código seja automaticamente implantada em produção sem intervenção humana.

Em relação a Kubernetes e Openshift, julgue os próximos itens.

71 Ao usar, em um `StatefulSet` com volume persistente, um `PodDisruptionBudget` com `maxUnavailable: 0`, o Kubernetes garantirá que nenhum *pod* seja interrompido durante operações de manutenção, mesmo que isso impeça a atualização de nós do *cluster* ou a aplicação de *patches* de segurança críticos.

72 No Openshift, o uso de `ImageStream` para gerenciar imagens de contêineres é essencial para garantir atualizações automáticas de *deployments* quando novas versões de uma imagem são construídas, mesmo em ambientes com múltiplos *namespaces* ou *clusters*.

A respeito de arquiteturas de microsserviços, julgue os itens que se seguem.

73 No trecho de código a seguir, se `config.name` estiver definido como `Spring` em algum arquivo de configuração da aplicação, o Spring Boot injetará o valor na variável `name`, exibindo `Hello Spring` ao acessar a rota "`/`".

```
@Configuration
@EnableAutoConfiguration
@RestController
public class Application {

  @Value("$ {config.name}")
  String name = "World";

  @RequestMapping("/")
  public String home() {
    return "Hello " + name;
  }

  public static void main(String[] args) {
    SpringApplication.run(Application.class,
args);
  }

}
```

74 O MapStruct exige que todas as propriedades dos objetos de origem e destino tenham exatamente o mesmo nome; caso contrário, o mapeamento falhará durante a compilação.

75 Considere que o arquivo `application.yml` contenha as instruções a seguir.

```
zuul :
   ignoredServices : '*'
  rotas :
    usuarios : /myusers*
```

Nesse caso, a configuração `ignoredServices : '*'` faz que o Zuul ignore todos os serviços automaticamente, enquanto a rota personalizada `usuarios : /myusers*` define um caminho específico para redirecionamento, o que garante que apenas as rotas explicitamente mapeadas sejam atendidas pelo *gateway*.

76 Durante a inicialização, microsserviços configurados como clientes Eureka registram-se automaticamente no servidor Eureka, fornecendo metadados, como nome do serviço, endereço IP ou *hostname*, e porta.

77 No Swagger/OpenAPI, o atributo `format` em JSON Schema é obrigatório — de modo que sua ausência invalida o esquema — e serve para refinar a semântica de validação de um tipo primitivo, como no caso de *strings* formatadas como `date` ou `email`.

Acerca de *domain-driven design*, de especificação OpenAPI e de *design patterns*, julgue os itens a seguir.

78 Em *domain-driven design*, a *infrastructure layer* deve conter a lógica de negócios central e coordenar as regras de domínio, enquanto a *domain layer* é responsável apenas por implementar detalhes técnicos, como conexões com bancos de dados e chamadas a APIs externas.

79 Em arquivos de configuração baseados em esquema, dados YAML devem adotar estrutura e sintaxe compatíveis com o formato JSON, para garantir validação adequada.

80 Em JSON Schema, o formato `format` é opcional e não valida dados por padrão, sendo alguns tipos, como `int32` e `password`, definidos pela OpenAPI, mas seu suporte varia entre ferramentas.

81 Em *design patterns*, o padrão *strategy* introduz forte hierarquia de herança entre algoritmos, violando o OCP (*open*/*closed principle*) ao exigir modificações no contexto sempre que uma nova estratégia é adicionada.

A respeito de criptografia, do *clean code* e das técnicas de *refactoring*, julgue os itens seguintes.

82 Considere que um tribunal necessite transmitir dados de processos sensíveis para outras varas remotas, mas esteja enfrentando restrições de largura de banda e requisitos de baixa latência. Considere, ainda, que, a fim de proteger os dados, a equipe pretenda usar criptografia simétrica de fluxo, sem impactar o desempenho da transmissão. Nessa situação, para máxima segurança, será necessário gerar uma nova chave simétrica para cada pacote de dados, que deverá ser enviada junto com o pacote criptografado.

83 Um código limpo ou *clean code* deve priorizar legibilidade semântica e baixo ruído de manutenção, devendo comentários ser usados apenas quando a intenção da lógica não puder ser claramente expressa pelo código em si.

84 O *refactoring extract superclass* é utilizado para esconder dependências indiretas entre objetos, promovendo encapsulamento de delegações e reduzindo o acoplamento estrutural do sistema.

Acerca das ferramentas Karma e SonarQube, julgue os itens que se seguem.

85 O parâmetro `browserDisconnectTolerance` do Karma Runner especifica quantas reconexões são permitidas entre servidor e navegadores durante falhas temporárias de rede, com valor padrão zero (sem tentativas de reconexão).

86 O SonarQube exige que todo código analisado seja compilado antes da execução da análise estática, mesmo para linguagens interpretadas como Python ou JavaScript.

Julgue os itens a seguir, em relação aos serviços de mensageria, *webhooks* e JSON.

87 O RabbitMQ permite a comunicação assíncrona entre aplicações por meio de filas, em que *producers* enviam mensagens e *consumers* as recebem, o que garante entrega confiável em sistemas distribuídos.

88 No Kafka, embora *producer* e *consumer* utilizem a mesma dependência Maven (`kafka-clients`), cada um exige configurações distintas: o *producer* requer, por exemplo, serializadores (`key.serializer`), enquanto o *consumer* utiliza desserializadores (`key.deserializer`).

89 Provedores de API reversa devem usar WebSockets para enviar notificações em tempo real, dispensando filas ou mecanismos de retentativa com sistemas externos.

90 O código JSON a seguir está de acordo com boas práticas de representação de dados de usuários, e pode ser utilizado diretamente, sem adaptações na integração entre *front-end* e banco de dados.

```
{
  "usuario": {
    "id": 12345,
    "nome": "Maria Oliveira",
    "ativo": "true",
    "permissoes": ["admin", "leitura"],
    "$ schema":
"http://meuesquema.com/schema"
  },
  "status": true
}
```

Julgue os itens a seguir, a respeito de servidores de aplicação e ferramentas de versionamento.

91 O comando `git-cherry-pick` aplica as alterações introduzidas por alguns *commits* existentes, sendo necessário, nesse caso, que a árvore de trabalho esteja limpa.

92 Um `realm` no servidor Tomcat verifica a integridade de um arquivo ao comparar o seu resumo de mensagem com o do arquivo original.

Acerca de Ajax e TypeScript e seus *frameworks*, julgue os itens seguintes.

93 O código a seguir, ao ser executado, apresentará `number` como resultado.

```
let x: alk = "ABC";
x = 10;
console.log(typeof x);
```

94 O objeto `XMLHttpRequest` manipula somente dados em formato JSON.

Julgue os itens subsequentes, referentes a Python e Java.

95
```
x = [10, 20, 30]
y = x
y += [60, 50, 40]
print(x)
print(y)
```

O código Python precedente, ao ser executado, apresentará o resultado a seguir.

```
[10, 20, 30]
[60, 60, 60]
```

96
```
import java.util.*;
public class Prova {
  public static void main(String[] args) {
    Set<String> conjunto = new HashSet<>();
    conjunto.add("Java");
    conjunto.add("Python");
    conjunto.add("Javascript");
    conjunto.add("Java");
    System.out.println(conjunto);
  }
}
```

O código Java precedente, ao ser executado, apresentará o resultado a seguir.

```
[Java, Javascript, Python]
```

```
<html>
  <table border=1>
    <tr>
      <td rowspan=2> abcd </td>
      <td colspan=2> efgh </td>
    </tr>
    <tr>
      <td> ijkl </td>
      <td rowspan=2> mnop </td>
    </tr>
      <tr> <td colspan=2> rstu </td>
    </tr>
  </table>
</html>
```

Com base no código precedente, em HTML 5, julgue o item seguinte, acerca dos conceitos de XML e de HTML.

97 O código apresentará como resultado uma tabela com 3 linhas e 3 colunas: as linhas 1, 2 e 3 contendo 2, 3 e 2 elementos, respectivamente; e as colunas 1, 2 e 3 contendo 2, 3 e 2 elementos, respectivamente.

No que concerne aos conceitos de eMAG e de SPA (*single-page application*), julgue os itens a seguir.

98 De acordo com o eMAG, todas as funções de uma página *web* desenvolvidas utilizando-se linguagens de *script* (JavaScript) devem ser programadas, primeiramente, para o uso com teclado.

99 Um SPA é uma implementação de aplicativo *web* em que *frameworks* PHP com Zend são uma parte fundamental do desenvolvimento *web front-end*.

Julgue os próximos itens, a respeito de *Big Data*, *data mining* e Oracle Data Integrator (ODI).

100 Define-se *Big Data* apenas pelo volume de dados, sendo o armazenamento eficiente desses dados o principal objetivo dos sistemas modernos de processamento de informações.

101 Em soluções de *Big Data* orientadas à análise em tempo real, os dados são invariavelmente persistidos antes de serem processados, o que assegura a integridade analítica e a eliminação completa de latência na geração de *insights*.

102 Uma das funções de *data mining* é realizar previsões com base em dados históricos, utilizando algoritmos como regressão, redes neurais e árvores de decisão.

103 As técnicas de *data mining* operam exclusivamente com dados rotulados, utilizando apenas algoritmos supervisionados, uma vez que a descoberta de padrões requer o conhecimento prévio das classes dos dados.

104 No ODI, os *knowledge modules* são componentes reutilizáveis que encapsulam as lógicas de integração e transformação e que podem ser personalizados em linguagens como SQL, Groovy e Shell Script.

105 O ODI suporta integração nativa com tecnologias *Big Data* por meio do ODI Big Data Adapter e é capaz de gerar código HiveQL, Spark e Pig sem que o desenvolvedor escreva manualmente tais *scripts*.

De acordo com a Resolução CNJ n.º 615/2025, que estabelece diretrizes para o desenvolvimento, a utilização e a governança de soluções desenvolvidas com recursos de inteligência artificial no Poder Judiciário, julgue os próximos itens.

106 É vedado o uso de modelos de linguagem de grande escala e sistemas de inteligência artificial generativa, privados ou externos ao Poder Judiciário, para processar ou gerar conteúdo com base em documentos sigilosos, mesmo que os dados tenham sido anonimizados ou que existam mecanismos técnicos para garantir sua proteção.

107 Após o treinamento de modelos de inteligência artificial, recomenda-se eliminar os dados pessoais não anonimizados dos repositórios de dados, mantendo apenas os dados tokenizados estritamente necessários ao modelo.

Acerca do processamento de linguagem natural e dos conceitos de modelos preditivos (supervisionados) e descritivos (não supervisionados), julgue os itens a seguir.

108 Em um *pipeline* de pré-processamento de linguagem natural aplicado ao domínio jurídico, a tokenização deve, necessariamente, ser precedida pela lematização, uma vez que a lematização opera sobre formas canônicas já segmentadas.

109 Em análise preditiva, algoritmos de classificação podem ser treinados a partir de conjuntos de dados sem rótulos; sendo o processo de ajuste dos parâmetros pelo algoritmo com base nesses dados denominado aprendizado supervisionado.

Julgue os itens subsequentes, no que concerne ao uso da biblioteca Hugging Face Transformers com PyTorch e aos sistemas baseados na arquitetura RAG (*retrieval-augmented generation*).

110 Embora a biblioteca Hugging Face ofereça suporte a múltiplos *frameworks*, alguns modelos específicos — especialmente os mais antigos, originalmente treinados em TensorFlow — podem apresentar dificuldades de integração em *pipelines* de inferência desenvolvidos em PyTorch, como os utilizados em aplicações jurídicas; nesses casos, pode ser recomendável adaptar ou reconstruir o modelo em PyTorch para assegurar compatibilidade e controle total sobre o fluxo de dados.

111 Considerando-se que os sistemas baseados na arquitetura RAG são projetados para mitigar limitações dos grandes modelos de linguagem, como a alucinação factual e a dificuldade de atualização constante, é correto afirmar que o uso do RAG permite incorporar documentos normativos, jurisprudência ou doutrina diretamente no processo de geração de respostas, viabilizando consultas mais confiáveis, com rastreabilidade das fontes e sem a necessidade de retreinamento do modelo-base.

A respeito da proteção de dados pessoais no Brasil, dos grandes modelos de linguagem (LLMs) e do uso de redes neurais no setor jurídico, julgue os itens que se seguem.

112 De acordo com a Lei Geral de Proteção de Dados Pessoais, no contexto de um sistema informatizado de um órgão público, é atribuição do encarregado de dados tomar decisões referentes ao tratamento de dados pessoais, a exemplo da definição das finalidades do respectivo tratamento.

113 Os LLMs se destacaram recentemente em decorrência do aprimoramento das técnicas de treinamento com *feedback* humano; da maior acessibilidade via interfaces de uso simples como ChatGPT e Gemini; do avanço da potência computacional com GPUs; e da melhoria dos dados de treinamento.

114 Modelos *transformer* superam, devido ao mecanismo de autoatenção, as redes neurais recorrentes tradicionais na captura de dependências de longo alcance em documentos jurídicos extensos, sendo esta a principal razão para a adoção dos *transformers* em tarefas como sumarização e busca semântica no campo jurídico.

A respeito do desenvolvimento de *software* seguro, julgue os itens a seguir.

115 Uma prevenção recomendada pelo OWASP Top 10 contra quebra de controle de acesso é a implementação de verificações de autorização nos modelos de domínio que apliquem as restrições de negócios da aplicação.

116 Os princípios da arquitetura e da governança Zero Trust no SDL incluem a presunção de que o sistema já está comprometido, a verificação explícita da confiança e a concessão do menor privilégio necessário para cada conta de usuário, cada identidade de máquina/serviço e cada componente da aplicação.

117 Na prática "Produzir *Software* Bem Protegido" do SSDF (*secure software development framework*), o NIST incentiva o reúso de credenciais de autenticação entre diferentes ambientes de desenvolvimento para facilitar o acesso dos desenvolvedores e agilizar o processo de integração contínua.

118 Na estrutura de desenvolvimento de *software* seguro do NIST, o grupo de prática "Preparar a Organização" inclui a recomendação de que todos os componentes dos ambientes de desenvolvimento de *software* sejam fortemente protegidos contra ameaças internas e externas, a fim de prevenir comprometimentos.

119 O *privacy by design* é mais eficiente quando implementado como uma camada de segurança adicional sobre sistemas já existentes.

120 O *privacy by design* fundamenta-se em princípios indivisíveis que devem ser implementados de forma integral e holística, não permitindo seleção parcial ou adaptação baseada em preferências setoriais específicas.

Espaço livre