

-- CONHECIMENTOS ESPECÍFICOS --**CIÊNCIA DE DADOS**

A respeito de aprendizagem de máquina, *deep learning* e processamento de linguagem natural, julgue os itens seguintes.

- 51** BoW (*bag of words*) é uma técnica de processamento de linguagem natural que transforma um texto em um vetor, considerando a frequência e a ordem ou o contexto das palavras presentes.
- 52** A validação cruzada é uma técnica utilizada para prover a generalização de um modelo treinado, aprimorando seu desempenho e identificando problemas potenciais, como *overfitting* e *underfitting*.
- 53** O *deep learning* é um subconjunto do *machine learning* que elimina parte do pré-processamento de dados, pois, por meio de técnicas como descende de gradiente e retropropagação, o algoritmo se ajusta para melhorar a precisão do modelo.
- 54** Na validação cruzada K-fold em que $k=5$, os dados são divididos em 5 partes, sendo um subconjunto de tamanho $k-1$ para teste e o restante para validação, o que resulta em uma métrica de avaliação média para o modelo.

No que se refere ao Hadoop, ferramenta-chave no processamento de *Big Data*, julgue o próximo item.

- 55** No Hadoop, o `Secondary NameNode` executa *checkpoints* periódicos do *namespace* e auxilia na gestão do tamanho do arquivo que registra as alterações do HDFS, garantindo que permaneça dentro de limites específicos no `NameNode`.

Relativamente à qualidade dos dados no âmbito da ciência de dados, julgue o item que se segue.

- 56** A consistência dos dados é crucial para sua qualidade, pois garante que não haja discrepâncias entre conjuntos de dados diferentes ou dentro do mesmo conjunto.

A respeito dos tipos de aprendizado de máquina, julgue os itens a seguir.

- 57** O agrupamento é um método de aprendizado não supervisionado que busca a similaridade entre os dados e, uma vez identificada tal similaridade, organiza os itens em *clusters* específicos.
- 58** O aprendizado por transferência transdutiva ocorre quando os domínios de origem e alvo são os mesmos, mas as tarefas que o modelo deve concluir são diferentes.

Em relação aos grandes modelos de linguagem (LLM) e à inteligência artificial generativa, julgue os itens subsequentes.

- 59** O parâmetro *temperature* em LLM regula a amplitude da diversidade e criatividade nas previsões produzidas pelo modelo.
- 60** Uma *thread* representa uma entidade que pode ser configurada para responder às mensagens de um usuário a partir de vários parâmetros, como modelo, instruções e ferramentas.

Com relação aos tipos de redes neurais, julgue o item seguinte.

- 61** As redes neurais LSTM (*long short-term memory*) são um tipo de rede neural convulacional projetada para lidar com problemas de memória de longo prazo em sequências de dados.

Julgue os itens subsecutivos, relativos a versionamento e automação do ciclo de produção de modelos de aprendizado de máquina em operações de aprendizado de máquina (MLOps).

- 62** A automação do ciclo de produção de modelos de aprendizado de máquina percorre diversas etapas, desde a ingestão de dados até a implantação, e inclui atividades como pré-processamento, treinamento de modelos e validação.
- 63** Em MLOps, o controle de versão abrange o monitoramento das mudanças nos ativos de aprendizado de máquina, visando assegurar a reprodutibilidade e a auditabilidade dos treinamentos desses modelos.

Em relação à privacidade em governança e ética de inteligência artificial, julgue o item seguinte.

- 64** De acordo com os princípios de governança e ética relativos à inteligência artificial, a opacidade sobre os algoritmos é encorajada como prática padrão para garantir a eficácia dos sistemas.

SEGURANÇA DA INFORMAÇÃO

A respeito do SAML 2.0, julgue os itens a seguir.

- 65** O SAML é considerado o principal padrão de autenticação, sendo o OAuth2 a sua correspondente implementação.
- 66** O sujeito, o provedor de identidade e o provedor de serviços são as partes envolvidas em um processo do tipo autenticação, considerando-se o SAML e o início de sessão única (SSO).

Em relação aos principais tipos de ataques a aplicações *web* e a sua prevenção, julgue os seguintes itens.

- 67** *Fuzzing* é um tipo de teste de prevenção de ataques que consiste na inserção de poucos dados não aleatórios em uma aplicação *web* com a expectativa de se obter um código de erro 500.
- 68** XSS (*Cross-site scripting*) é um tipo de ataque por meio do qual o atacante explora erros nos códigos de entrada de dados, e a prevenção de ataques XSS requer sanitização das entradas de dados, de maneira a impedir a injeção de códigos maliciosos.

A respeito de múltiplos fatores de autenticação (MFA), julgue os próximos itens.

- 69** MFA adaptativa é uma variação de MFA em que se pode considerar o número de tentativas de *login* ou até a localização geográfica do usuário.
- 70** MFA é um processo que requer uma etapa de variação criptográfica da senha do usuário por diferenciação de chave pública.

Acerca do NIST CSF 2.0, julgue o item a seguir.

- 71** O núcleo da estrutura do NIST CSF 2.0 está organizado em torno de três funções principais: identificar, proteger e detectar.

ENGENHARIA DE SOFTWARE

No que diz respeito à arquitetura de sistemas *web*, julgue os itens a seguir.

- 72** No WebSocket, utiliza-se um fluxo de *bytes* no lugar de um fluxo de mensagens para fornecer comunicações completas em uma conexão TCP.
- 73** Com o balanceamento de carga no modo espera ativa, uma instância do controlador de entrega de aplicações é colocada *online* quando houver falha no nó principal.
- 74** No HTTP/2, são utilizadas conexões persistentes para o atendimento a diversas solicitações em sequência, ao passo que, no HTTP/1, as conexões atendem a várias solicitações simultâneas.
- 75** No gRPC, o protocolo de transporte utilizado é o HTTP/2.

Acerca dos princípios e das práticas de DevOps e DevSecOps, integração e entrega contínuas, julgue os seguintes itens.

- 76** A realização de testes no DevSecOps apoia-se no uso de ferramentas de teste dinâmico de segurança de aplicativos (DAST) com base nas recomendações do OWASP.
- 77** Integração, entrega e implementação contínuas e com segurança em todo o ciclo de vida do desenvolvimento de *software* são os princípios básicos do DevOps.

No que se refere a técnicas de desenvolvimento seguro, julgue os itens que se seguem.

- 78** A redução de velocidade de resposta do servidor é uma forma de prevenção de ataques como o que causa o erro HTTP 429.
- 79** XSS (*Cross-site scripting*) e entidades externas de XML são exemplos de controles e boas práticas para o desenvolvimento seguro.

Julgue os itens subsequentes, acerca de testes de *software*.

- 80** No BDD, os nomes dos métodos de testes não são códigos, mas frases com significado real, a exemplo das histórias de usuários.
- 81** O TDD (*test driven development*) é um tipo de teste unitário em que a especificação de teste é escrita logo após o código para validar o comportamento desejado.

A área de tecnologia da informação e comunicação (TIC) do Banco Central do Brasil (BCB) necessita disponibilizar determinado aplicativo para transações digitais que o BCB vem desenvolvendo, utilizando contêineres no Kubernetes junto ao seu provedor de nuvem. Existe o requisito de que essa aplicação seja gerida no modo *serverless*.

Tendo como referência a situação hipotética apresentada, julgue os próximos itens, relativos a microsserviços e *serverless*.

- 82** Na orquestração de serviços em *cluster kubernetes*, um *pod* sempre será executado em um nó, ainda que seja em uma máquina física.
- 83** No caso hipotético em tela, a área de TIC deve dimensionar a quantidade de *Pods* de acordo com a demanda esperada, uma vez que os *Pods* nesse modo são o ponto focal para avaliação de *performance*. Isso se deve ao fato de que, ao se utilizar o Kubernetes no modo *serverless*, cada *pod* gerencia os *kubelets* e os contêineres que estiverem contidos nos nós do *cluster*.

- 84** Na situação em apreço, se a equipe de TIC estiver usando o modelo *serverless*, não haverá necessidade de o BCB gerenciar servidores, uma vez que, nesse caso, cabe ao provedor de nuvem gerenciar a infraestrutura da nuvem e escalar as aplicações, dimensionando e gerenciando automaticamente a infraestrutura necessária para executar o código.

- 85** No cenário em questão, sendo utilizado o modelo *serverless*, os desenvolvedores têm acesso, no BaaS (*backend as a service*), a serviços e aplicações de terceiros, tais como serviços de autenticação, banco de dados e criptografia, que podem ser acionados por meio de APIs.

- 86** No modelo FaaS (*function as a service*), os recursos de computação necessários para executar a lógica do aplicativo em resposta às solicitações são executados de forma orientada a eventos em contêineres *stateless*, totalmente gerenciados pelo provedor de serviços de nuvem.

No que se refere a programação assíncrona, UX *design* e UI *design*, julgue os itens a seguir.

- 87** Considerando-se que o trecho de código a seguir tenha sido desenvolvido em Python e que seja executado em um ambiente *multithreading*, é correto afirmar que este é um exemplo de programação assíncrona, uma vez que as três funções não dependem umas das outras para serem executadas, ainda que não possam ser executadas simultaneamente.

```
def a():
    print("Banco")

def b():
    print("Central")

def c():
    print("do Brasil")

a()
b()
c()
```

- 88** Considere que a área de TIC do BCB precise desenvolver uma solução voltada a um aplicativo que o BCB vem desenvolvendo e que, nesse projeto, haja as seguintes demandas: a) projetar a interatividade, a aparência e a sensação da tela do aplicativo; b) desenvolver a interação do usuário com o serviço, assegurando que ela ocorra da forma mais estável possível. Nessa situação hipotética, as demandas “a” e “b” são afetas, respectivamente, às práticas de UI *design* e UX *design*.

A equipe de TIC do BCB necessita desenvolver uma API a ser utilizada em uma solução que o BCB vem desenvolvendo e que será disponibilizada pelo órgão. Como requisitos, tal API deve apresentar: solicitações gerenciadas por HTTP; possibilidade de os clientes solicitarem recursos em qualquer ordem; e isolamento de cada solicitação em relação às outras solicitações.

Considerando a situação hipotética anterior, julgue os itens que se seguem, relativos a RESTful e GraphQL.

- 89** Na situação em apreço, caso seja necessário que a API consulte dados em uma base de dados relacional compatível com a linguagem SQL, poder-se-á utilizar a GraphQL para obter dados nos servidores em questão, uma vez que a GraphQL executa comandos SQL em arquiteturas RESTful diretamente no(s) sistema(s) de gerenciamento de banco de dados (SGBD) atrelado(s) à solução.
- 90** Caso seja utilizada a arquitetura RESTful, a citada API, além de atender aos requisitos descritos, será *stateless*, ou seja, o servidor completará cada solicitação do cliente independentemente de todas as solicitações anteriores, e todas as solicitações serão separadas e desconectadas entre si.

Julgue os itens subsequentes, relativos a padrões GoF.

- 91** Os padrões *abstract factory* e *facade* são, respectivamente, de criação e estrutural, uma vez que o *abstract factory* fornece uma interface para criação de famílias de objetos relacionados, ao passo que o *facade* separa a construção de um objeto complexo da sua representação.
- 92** O padrão *composite* atribui responsabilidades adicionais a um objeto dinamicamente e o *memento*, sem violar o encapsulamento, captura e externaliza um estado interno de um objeto, de modo que este possa, posteriormente, ser restaurado para aquele estado.

Julgue o item seguinte, acerca de DLT (*distributed ledger technology*).

- 93** Uma DLT é uma tecnologia de criptografia de chave assimétrica em que as chaves pública e privada ficam separadas, sendo utilizada para proteger livros-razão em *blockchain*, em que os dados são armazenados em blocos interligados em uma cadeia.

Julgue os itens a seguir, relativos a Python e Java.

- 94** Considere o seguinte código desenvolvido em Java.

```
public class Main {
    public static void main(String[] args) {
        int resultado = fun(3);
        System.out.println(resultado);
    }
    public static int fun(int y) {
        if (y > 1) {
            return y + fun(y - 1);
        } else {
            return 0;
        }
    }
}
```

Ao ser executado, esse código apresentará o seguinte resultado.

- 95** Considere o seguinte código desenvolvido em Python.

```
print(len("Banco"))
print(len(["Banco", "Central", "do Brasil"]))
print(len({"Presidente": "Roberto Campos Neto",
"Email": "presidencia@bcb.gov.br"}))
```

Ao ser executado, esse código apresentará o seguinte resultado.

5
3
2

INFRAESTRUTURA EM TI

A respeito de infraestrutura como código, Docker e Kubernetes, julgue os itens a seguir.

- 96** Ao se executar o arquivo YAML descrito a seguir, será criado um *deployment* com uma imagem do NGINX.

```
apiVersion: v1
kind: Service
metadata:
  name: my-nginx-svc
  labels:
    app: nginx
spec:
  type: LoadBalancer
  ports:
    - port: 80
  selector:
    app: nginx
```

- 97** No modelo de infraestrutura como código, é possível a implantação de uma infraestrutura com definição de sub-redes, balanceadores de carga e máquinas virtuais com a utilização de código, sem a necessidade de configurações manuais.
- 98** O *kube-scheduler* é o escalonador do Kubernetes e sua função consiste em controlar o aumento e a diminuição da quantidade de *Pods* de acordo com as configurações de consumo.
- 99** O Kubernetes pode operar sem usar o Docker e suporta a execução com CRI-O e Containerd.

Acerca de monitoramento e observabilidade de sistemas de TI, julgue os itens subsequentes.

- 100** O Grafana é capaz de fazer *queries* em diferentes tipos de *data source*, transformar os dados obtidos e gerar a visualização em forma de *dashboard*.
- 101** A métrica do tipo *gauge*, suportada pelo Prometheus, representa um contador único, que pode ser usado, por exemplo, para representar a quantidade de chamadas para um *endpoint* HTTP.
- 102** Uma ferramenta de APM é capaz de capturar *traces* de uma aplicação e registrar a taxa de erros de requisições com retorno HTTP 500.

Em relação a protocolos de aplicação, julgue os itens seguintes.

- 103** O protocolo HTTP funciona na camada de aplicação do protocolo TCP/IP, sendo capaz de fornecer acesso a páginas *web* e de também enviar e receber *emails*, estabelecendo comunicação com servidores de *email*.
- 104** O LDAP é comumente utilizado em serviços de diretório como o Microsoft Active Directory, sendo capaz de autenticar usuários em aplicativos.

Julgue os próximos itens, relativos a tolerância a falhas e continuidade de operação, implantação e administração de serviços de nuvem e administração e gerenciamento de ambiente de virtualização.

- 105** Com a virtualização de máquinas em nuvens IaaS, pode-se fazer a migração de máquinas virtuais enquanto elas ainda são operacionais, utilizando-se técnicas como a migração de memória com pré-cópia, conhecida como migração viva (*live migration*).
- 106** A categoria de serviço em nuvem IaaS oferta uma plataforma robusta, flexível, que permite a execução e o desenvolvimento de serviços de aplicação (*middleware*), gestão de processos de negócio e serviço de banco de dados.
- 107** Em TI, a alta disponibilidade de um serviço caracteriza-se por ele estar continuamente disponível para o cliente, havendo pouco *downtime* e recuperação rápida.
- 108** A capacidade de alternar para um servidor, sistema ou rede redundante ou em *standby*, mediante a falha ou o desligamento de um ativo existente, é chamada de *failback*.

Julgue os itens subsequentes, relacionados à administração e à operação do Microsoft Windows Server e do Linux e a conceitos de LAN, WAN e SDN.

- 109** As principais desvantagens na implementação de uma rede SDN são alto custo de manutenção, baixa flexibilidade e escalabilidade, e alto grau de dificuldade de gerenciamento.
- 110** O Windows Admin Center é um conjunto de ferramentas com interface baseada em um navegador *web*, usadas, entre outras finalidades, para a configuração de recursos de administração quando da implementação de *clusters* hiperconvergentes.
- 111** Uma rede LAN é formada por computadores interligados por cabos ou ondas de rádio, em que todos pertencem a um mesmo local físico, e uma rede WAN é formada pela interligação de pequenas ou grandes redes LAN.
- 112** No Linux, ao se executar o comando `dpkg`, todas as dependências do pacote solicitado são instaladas, sem a necessidade de confirmação do administrador.

BANCOS DE DADOS

Julgue os próximos itens, relativos a bancos de dados.

- 113** Um dos objetivos do *sharding* de arquivos em bancos de dados NoSQL é melhorar o balanceamento de carga entre os nós da aplicação.
- 114** Quando utilizado o comando `SELECT`, podem-se incluir várias *subqueries* internas na mesma instrução.
- 115** Considera-se que uma tabela está na terceira forma normal (3FN) quando não existem dependências multivaloradas, ou seja, campos que se repetem em relação à chave primária.
- 116** Em um *data warehouse*, a ocorrência de *deadlocks* causados por grande fluxo de dados é um fator crítico para o desempenho.

GESTÃO EM TI

Julgue os próximos itens, relativos a Kanban e Scrum.

- 117** Na *sprint planning*, o *product owner* é o responsável, para cada item do *product backlog*, por planejar e definir a forma como será executado o trabalho necessário para criar um incremento que atenda à definição de pronto no Scrum.
- 118** O Kanban é uma ferramenta que prevê o progresso de uma *sprint* no Scrum.

No que concerne a governança de dados e ITIL v4, julgue os itens a seguir.

- 119** Entender as necessidades estratégicas de dados da empresa e desenvolver e manter uma estratégia de dados são exemplos de objetivos na atividade planejamento da gestão de dados de governança de dados.
- 120** No ITIL v4, a prática gerenciamento de risco visa garantir que a organização entenda e lide com os riscos de forma eficaz, como, por exemplo, no gerenciamento do nível de serviço, em que se deve identificar e gerenciar os riscos que possam afetar os níveis de serviço, bem como relatar esses riscos aos clientes e a outras partes interessadas.

Espaço livre