

Prova de Desenvolvimento de APIs (Nível Médio)

Instruções:

- Leia atentamente cada questão.
 - Selecione a alternativa correta para cada pergunta.
-

Parte 1: Autogestão

1. **Qual das alternativas a seguir é uma técnica eficaz para melhorar a concentração durante o desenvolvimento de software?**
 - A) Trabalhar continuamente sem pausas para manter o foco.
 - B) Dividir o trabalho em tarefas menores e gerenciáveis.
 - C) Alternar entre várias tarefas para evitar monotonia.
 - D) Evitar o uso de qualquer ferramenta de gerenciamento de tempo.
-

Parte 2: Linguagem de Programação para APIs

2. **Qual é a principal vantagem do uso de JSON em APIs em comparação com XML?**
 - A) JSON é mais legível para humanos.
 - B) JSON suporta melhor a formatação de dados binários.
 - C) JSON é mais robusto para definir esquemas de dados complexos.
 - D) JSON tem melhor suporte para namespaces.
3. **Em qual situação o uso de XML pode ser mais vantajoso do que JSON?**
 - A) Quando a interoperabilidade com sistemas legados é necessária.
 - B) Quando a simplicidade e a legibilidade são prioridades.
 - C) Quando o desempenho é uma preocupação crítica.
 - D) Quando se trabalha exclusivamente com dados numéricos.
4. **Como o tratamento de exceções pode ser implementado em uma API RESTful em Python utilizando Flask?**
 - A) Definindo um manipulador global de erros com `app.errorhandler`.
 - B) Usando blocos `try-except` em todas as funções.
 - C) Implementando um middleware para captura de exceções.
 - D) Ignorando exceções para melhorar a performance.
5. **Qual código de status HTTP deve ser retornado quando um recurso é criado com sucesso?**
 - A) 200 OK
 - B) 201 Created
 - C) 202 Accepted
 - D) 204 No Content
6. **Quais são os benefícios de usar um framework como Django para desenvolvimento de APIs em vez de um microframework como Flask?**
 - A) Django é mais fácil de configurar e usar.

- B) Django oferece um ecossistema completo, incluindo ORM, administração, e autenticação.
 - C) Flask é mais robusto e escalável.
 - D) Flask tem melhor suporte para grandes equipes de desenvolvimento.
7. **O que é a técnica de injeção de dependências e como ela pode beneficiar a construção de APIs?**
- A) É o processo de embutir dependências diretamente no código fonte, o que facilita a manutenção.
 - B) É a prática de fornecer dependências a uma classe em vez de serem criadas internamente, facilitando testes e manutenção.
 - C) É a técnica de reduzir a quantidade de dependências para otimizar o desempenho.
 - D) É a prática de eliminar a necessidade de dependências externas para aumentar a segurança.
8. **Qual é a função do Swagger/OpenAPI na documentação de APIs?**
- A) Automatizar a criação de código para clientes de API.
 - B) Proporcionar uma interface gráfica para interação com a API.
 - C) Gerar automaticamente a documentação da API com base no código.
 - D) Melhorar a performance da API através da otimização de código.
9. **Quais técnicas de depuração são mais eficazes para resolver problemas em APIs?**
- A) Uso de print statements no código.
 - B) Ferramentas de logging e rastreamento de erros.
 - C) Desativação de todas as otimizações do compilador.
 - D) Testes manuais contínuos.
10. **Qual das seguintes práticas é essencial para garantir a segurança de uma API?**
- A) Manter o código fonte público para auditoria.
 - B) Implementar autenticação e autorização robustas.
 - C) Reduzir o número de endpoints para minimizar a superfície de ataque.
 - D) Evitar o uso de bibliotecas de terceiros.
-

Parte 3: Padrão Model View Control (MVC)

11. **No padrão MVC, qual é a responsabilidade principal do Model?**

- A) Renderizar a interface do usuário.
- B) Gerenciar os dados e a lógica de negócios.
- C) Receber e processar as entradas do usuário.
- D) Controlar o fluxo de aplicação.

12. **Como o Controller interage com o Model e a View no padrão MVC?**

- A) O Controller apenas gerencia as requisições HTTP.
- B) O Controller recebe as entradas do usuário, chama o Model para processar os dados e atualiza a View.
- C) O Controller renderiza diretamente a interface do usuário.
- D) O Controller atua como intermediário apenas entre a View e o banco de dados.

13. Quais são as vantagens de usar o padrão MVC em projetos de desenvolvimento de software?

- A) Simplificação da arquitetura e redução do código.
- B) Separação de responsabilidades, facilitando a manutenção e o escalonamento.
- C) Melhor integração com bibliotecas de terceiros.
- D) Aumento da velocidade de execução do aplicativo.

14. Qual é a principal função da View no padrão MVC?

- A) Armazenar dados persistentes.
- B) Processar a lógica de negócios.
- C) Apresentar dados ao usuário.
- D) Receber e validar as entradas do usuário.

15. Em um aplicativo web baseado em MVC, onde as validações de dados devem ocorrer idealmente?

- A) Apenas no Controller.
- B) Apenas no Model.
- C) No Model e no Controller.
- D) Apenas na View.

16. Como a modularidade do padrão MVC contribui para o desenvolvimento ágil de software?

- A) Permite que desenvolvedores trabalhem simultaneamente em diferentes componentes.
 - B) Facilita a criação de código monolítico.
 - C) Reduz a necessidade de testes automatizados.
 - D) Diminui a complexidade de integração contínua.
-

Parte 4: Métodos de Requisição HTTP

17. Qual método HTTP é utilizado para excluir um recurso no servidor?

- A) GET
- B) POST
- C) DELETE
- D) PATCH

18. Em qual cenário o uso do método PUT é mais apropriado que o PATCH?

- A) Quando é necessário criar um novo recurso.
- B) Quando é necessário atualizar parcialmente um recurso existente.
- C) Quando é necessário substituir completamente um recurso existente.
- D) Quando é necessário enviar grandes quantidades de dados binários.

19. Qual código de status HTTP indica que a solicitação foi bem-sucedida e um novo recurso foi criado?

- A) 200 OK
- B) 201 Created
- C) 204 No Content

- D) 404 Not Found

20. Quando deve-se usar o método PATCH em vez de PUT?

- A) Para substituir completamente um recurso.
- B) Para obter dados de um recurso.
- C) Para atualizar parcialmente um recurso existente.
- D) Para excluir um recurso.

21. Qual método HTTP deve ser usado para enviar dados ao servidor para processamento, sem necessariamente criar um novo recurso?

- A) GET
- B) DELETE
- C) POST
- D) PUT

22. Qual a principal diferença entre os métodos GET e POST?

- A) GET é usado para enviar dados e POST para receber dados.
- B) GET solicita dados de um recurso e POST envia dados para serem processados por um recurso.
- C) GET é seguro e idempotente, enquanto POST não é seguro e não é idempotente.
- D) GET cria novos recursos e POST atualiza recursos existentes.

23. Qual método HTTP e código de status são apropriados para atualizar parcialmente um recurso e indicar sucesso sem retornar um corpo de resposta?

- A) PUT e 200 OK
- B) PATCH e 204 No Content
- C) POST e 201 Created
- D) GET e 304 Not Modified

24. Qual método HTTP e código de status são utilizados para obter a representação de um recurso existente?

- A) POST e 201 Created
- B) DELETE e 204 No Content
- C) GET e 200 OK
- D) PUT e 202 Accepted

25. Qual código de status HTTP indica que o servidor entende o tipo de conteúdo da solicitação, mas a solicitação foi malformada?

- A) 200 OK
- B) 400 Bad Request
- C) 401 Unauthorized
- D) 403 Forbidden

26. Qual método HTTP é utilizado para aplicar modificações parciais a um recurso?

- A) GET
- B) POST
- C) PUT
- D) PATCH

27. Qual código de status HTTP indica que a solicitação foi bem-sucedida, mas que não há conteúdo a ser enviado na resposta?

- A) 200 OK
- B) 204 No Content
- C) 304 Not Modified
- D) 400 Bad Request

28. Em um cenário onde é necessário garantir que uma solicitação não crie múltiplas cópias de um recurso, qual método HTTP deve ser utilizado?

- A) GET
 - B) POST
 - C) PUT
 - D) DELETE
-

Parte 5: Interface de Programação de Aplicativos (API)

29. Qual das seguintes práticas não é recomendada para garantir a segurança de uma API?

- A) Implementar autenticação e autorização adequadas.
- B) Expor diretamente o banco de dados através de endpoints da API.
- C) Utilizar tokens de acesso com expiração curta.
- D) Encriptar dados sensíveis em trânsito e em repouso.

30. Por que a criptografia é importante na segurança de APIs?

- A) Para melhorar a velocidade de processamento.
- B) Para garantir que os dados em trânsito não sejam interceptados e lidos por terceiros não autorizados.
- C) Para reduzir o tamanho das requisições.
- D) Para evitar que a API seja descoberta por scanners de segurança.

31. Qual é a principal função do backup na segurança da informação de APIs?

- A) Garantir a confidencialidade dos dados.
- B) Assegurar a disponibilidade dos dados em caso de falha.
- C) Proteger os dados contra acessos não autorizados.
- D) Reduzir a necessidade de controle de acesso.

32. Como a auditoria contribui para a segurança de uma API?

- A) Aumentando a complexidade do código.
- B) Monitorando e registrando atividades e acessos para detectar e prevenir possíveis ataques.
- C) Reduzindo a quantidade de documentação necessária.
- D) Facilitando o desenvolvimento ágil.

33. O que garante a integridade dos dados em uma API?

- A) Assegurar que apenas indivíduos autorizados possam acessar informações sensíveis.
- B) Implementar mecanismos para verificar que os dados não foram alterados ou corrompidos.
- C) Garantir a disponibilidade dos dados em todos os momentos.
- D) Melhorar a velocidade de resposta da API.

34. Qual prática é essencial para garantir a confidencialidade em APIs?

- A) Realizar backups regulares dos dados.
- B) Utilizar criptografia de ponta a ponta para dados sensíveis.
- C) Implementar controle de acesso baseado em funções (RBAC).
- D) Monitorar a performance da API continuamente.

35. O que é OAuth e como ele é utilizado em APIs?

- A) Um protocolo de criptografia de dados.
- B) Um método para autenticação e autorização, permitindo que aplicativos acessem recursos em nome do usuário.
- C) Um tipo de firewall para APIs.
- D) Uma técnica de compressão de dados.

36. Qual a diferença entre autenticação e autorização em APIs?

- A) Autenticação verifica a identidade do usuário, enquanto autorização define o que o usuário pode fazer.
- B) Autenticação define permissões de acesso, enquanto autorização verifica credenciais do usuário.
- C) Autenticação é opcional, enquanto autorização é obrigatória.
- D) Autenticação e autorização são termos intercambiáveis.

37. Como as metodologias ágeis como Scrum e Kanban podem ser aplicadas no desenvolvimento de APIs?

- A) Reduzindo o tempo de desenvolvimento ao eliminar testes.
- B) Facilitando a adaptação a mudanças de requisitos e entregas incrementais.
- C) Melhorando a segurança ao limitar o número de desenvolvedores envolvidos.
- D) Aumentando a quantidade de documentação detalhada.

38. Qual é a principal diferença entre Scrum e Kanban?

- A) Scrum utiliza sprints com duração fixa, enquanto Kanban é baseado em um fluxo contínuo.
- B) Kanban é uma metodologia ágil, enquanto Scrum não é.
- C) Scrum foca em visualização do fluxo de trabalho, enquanto Kanban foca em reuniões diárias.
- D) Kanban requer uma equipe dedicada, enquanto Scrum não.

39. O que é um token JWT e como ele é utilizado em APIs?

- A) Um método de criptografia de dados.
- B) Um token usado para autenticação e autorização, garantindo acesso seguro a recursos protegidos.
- C) Um tipo de firewall para APIs.
- D) Uma técnica de compressão de dados para melhorar a performance.

40. Qual prática ajuda a manter a integridade dos dados em uma API?

- A) Usar somente bibliotecas de terceiros.
- B) Implementar controle de versão e assinaturas digitais.
- C) Permitir acesso anônimo a todos os recursos.
- D) Evitar a criptografia dos dados.

41. Qual das opções abaixo não é um pilar da segurança da informação?

- A) Confidencialidade
- B) Disponibilidade
- C) Integridade
- D) Escalabilidade

42.O que é um controle de acesso baseado em função (RBAC)?

- A) Um método para criptografar dados sensíveis.
- B) Uma abordagem para atribuir permissões aos usuários com base em suas funções na organização.
- C) Um protocolo para autenticação de usuários.
- D) Um tipo de firewall para proteger APIs.

43.Como a criptografia ajuda a garantir a segurança de uma API?

- A) Reduzindo o tamanho das requisições.
- B) Garantindo que dados sensíveis não possam ser lidos por interceptadores não autorizados.
- C) Aumentando a velocidade de processamento dos dados.
- D) Eliminando a necessidade de autenticação.

44.Qual técnica pode ser utilizada para proteger uma API contra ataques de força bruta?

- A) Implementar limites de taxa (rate limiting).
- B) Utilizar somente métodos HTTP seguros.
- C) Aumentar o tamanho dos tokens de autenticação.
- D) Reduzir a quantidade de dados transferidos.

45.O que é um certificado SSL/TLS e por que ele é importante para APIs?

- A) Um método de compactação de dados para melhorar a performance.
- B) Um certificado que permite a encriptação do tráfego entre cliente e servidor, garantindo a segurança dos dados em trânsito.
- C) Uma técnica para aumentar a velocidade de resposta da API.
- D) Um protocolo de controle de versão para APIs.

46.Por que o monitoramento contínuo é importante para a segurança de uma API?

- A) Para garantir a integridade dos dados armazenados.
- B) Para identificar e responder rapidamente a possíveis incidentes de segurança.
- C) Para reduzir o tempo de resposta da API.
- D) Para aumentar a disponibilidade dos serviços.

47.O que é uma política de CORS (Cross-Origin Resource Sharing) e por que é importante para APIs?

- A) Uma política para definir limites de taxa em APIs.
- B) Uma configuração de segurança que define quais domínios podem acessar recursos da API.
- C) Um método de criptografia de dados.
- D) Um protocolo de autenticação de usuários.

48.Como a técnica de injeção de dependências pode beneficiar o desenvolvimento de APIs?

- A) Facilita a manutenção e teste de código ao desacoplar dependências.
- B) Aumenta o tempo de execução da API.

- C) Reduz a necessidade de validação de dados.
- D) Aumenta a complexidade do código.

49. O que é um endpoint em uma API?

- A) Um método para criptografar dados sensíveis.
- B) Um ponto de entrada onde uma API aceita solicitações e retorna respostas.
- C) Um tipo de firewall para proteger APIs.
- D) Uma técnica de compressão de dados.

50. Qual é a função principal do método GET em uma API?

- A) Excluir um recurso.
- B) Atualizar parcialmente um recurso.
- C) Obter a representação de um recurso.
- D) Criar um novo recurso.