

1. Qual a principal função do HTTP (Hypertext Transfer Protocol)?
  - a) Armazenar dados em bancos de dados locais.
  - b) Gerenciar o acesso a arquivos de sistema.
  - c) Definir como a web se comunica para troca de dados entre servidor e cliente.
  - d) Controlar o hardware do dispositivo móvel.
2. Qual das seguintes afirmações descreve corretamente uma característica fundamental do HTTP?
  - a) HTTP mantém uma conexão contínua e stateful entre cliente e servidor.
  - b) HTTP é um protocolo que permite comunicação bidirecional sem polling.
  - c) HTTP é um protocolo sem estado (stateless), onde a conexão é interrompida após cada requisição-resposta.
  - d) HTTP possui conhecimento prévio de todas as requisições anteriores e futuras.
3. Em um ciclo de requisição-resposta HTTP, o que o "HTTP Verb" dentro da requisição indica?
  - a) O tipo de dados que o cliente espera receber.
  - b) A versão do protocolo HTTP utilizada.
  - c) A ação que o servidor deve realizar.
  - d) O tempo limite para a resposta do servidor.
4. Qual verbo HTTP é utilizado para solicitar informações de um servidor, como ao carregar uma página web ou uma imagem?
  - a) POST
  - b) PUT
  - c) DELETE
  - d) GET
5. Qual verbo HTTP é apropriado para enviar dados a um servidor com o objetivo de criar um novo recurso, como o envio de um formulário?
  - a) GET
  - b) POST
  - c) PUT
  - d) UPDATE
6. O que o código de status HTTP 200 OK na resposta do servidor geralmente significa?
  - a) O servidor encontrou um erro interno.
  - b) A requisição foi recusada por motivos de segurança.
  - c) O recurso solicitado não foi encontrado.
  - d) A requisição foi bem-sucedida.
7. Quando um recurso é criado com sucesso no servidor, qual código de status HTTP é comumente retornado?
  - a) 200 OK
  - b) 400 Bad Request
  - c) 201 CREATED
  - d) 500 Internal Server Error

**8.** Qual código de status HTTP é retornado quando o recurso solicitado pelo cliente não pode ser encontrado no servidor?

- a) 204 No Content
- b) 301 Moved Permanently
- c) 401 Unauthorized
- d) 404 Not Found

**9.** Qual pacote Flutter é amplamente recomendado e multiplataforma para realizar requisições HTTP?

- a) dart:io
- b) dart:html
- c) web\_socket\_channel
- d) http

**10.** Para que um aplicativo Flutter no Android possa acessar a internet e fazer requisições de rede, qual permissão deve ser declarada no arquivo AndroidManifest.xml?

- a) android.permission.READ\_PHONE\_STATE
- b) android.permission.ACCESS\_NETWORK\_STATE
- c) android.permission.WRITE\_EXTERNAL\_STORAGE
- d) android.permission.INTERNET

**11.** Em aplicativos Flutter para macOS, qual entrada é necessária nos arquivos .entitlements para permitir o acesso à rede?

- a) <key>com.apple.security.device.microphone</key><true/>
- b) <key>com.apple.security.personal-information.contacts</key><true/>
- c) <key>com.apple.security.files.user-selected.read-write</key><true/>
- d) <key>com.apple.security.network.client</key><true/>

**12.** No Dart, qual é a classe fundamental para trabalhar com operações assíncronas, representando um valor ou erro que estará disponível em algum momento no futuro, comumente retornado por métodos HTTP?

- a) Stream
- b) Future
- c) Response
- d) HttpClient

**13.** Por que é uma boa prática chamar funções que realizam requisições de API no método initState() (ou didChangeDependencies()) de um widget Stateful no Flutter, e não diretamente no método build()?

- a) O método initState() garante que a requisição seja feita a cada segundo.
- b) O método build() não pode lidar com operações assíncronas.
- c) O método build() é chamado com frequência, o que causaria requisições HTTP repetitivas e desnecessárias se a chamada da API estivesse lá.
- d) O método initState() é específico para a interface do usuário e não para dados.

**14.** Qual widget do Flutter é projetado para facilitar a exibição de dados assíncronos (como os retornados por um Future), permitindo definir o que renderizar durante os estados de carregamento, sucesso ou erro?

- a) StatelessWidget
- b) ListView.builder
- c) FutureBuilder
- d) GestureDetector

**15.** Ao converter uma http.Response em um objeto Dart personalizado, o que deve ser feito se o servidor retornar um código de status de erro (como 404 Not Found ou 500 Internal Server Error) em vez de um 200 OK ou 201 CREATED?

- a) Retornar null para indicar falha.
- b) Retornar um objeto vazio com valores padrão.
- c) Lançar uma Exception.
- d) Continuar processando a resposta como se fosse um sucesso.

**16.** Como é possível adicionar cabeçalhos de autorização (autenticação) a uma requisição HTTP usando o pacote http no Flutter?

- a) Passando os dados de autorização diretamente na URL.
- b) Incluindo os dados de autorização no corpo da requisição (body).
- c) Através do parâmetro headers do método HTTP (ex: http.get(..., headers: {HttpHeaders.authorizationHeader: '...'})).
- d) Definindo uma variável global para autenticação antes de cada requisição.

**17.** Qual é a principal diferença de conexão entre HTTP e WebSockets?

- a) HTTP e WebSockets são ambos stateless.
- b) Ambos mantêm uma conexão contínua para comunicação bidirecional.
- c) HTTP é stateless, enquanto WebSockets permitem uma comunicação bidirecional e contínua sem polling.
- d) HTTP é apenas para texto, e WebSockets são apenas para binários.

**18.** No Flutter, qual pacote é usado para facilitar a comunicação com servidores WebSocket, permitindo tanto ouvir quanto enviar mensagens?

- a) http
- b) dart:async
- c) socket\_io
- d) web\_socket\_channel

**19.** Como se envia dados para um servidor WebSocket utilizando um WebSocketChannel no Flutter?

- a) Através do método channel.send() diretamente.
- b) Utilizando a propriedade channel.write() com a mensagem.
- c) Adicionando a mensagem ao sink fornecido pelo WebSocketChannel (ex: channel.sink.add('mensagem');).
- d) Chamando o método channel.postMessage().

**20.** Qual classe do Dart, utilizada pelo WebSocketChannel, é fundamental para escutar eventos assíncronos de uma fonte de dados e pode entregar múltiplos eventos ao longo do tempo, ao contrário de um Future que entrega apenas um?

- a) Future
- b) StreamSink
- c) Stream
- d) HttpClientResponse

=====

**Gabarito:**

- 1. C
- 2. C
- 3. C
- 4. D
- 5. B
- 6. D
- 7. C
- 8. D
- 9. D
- 10. D
- 11. D
- 12. B
- 13. C
- 14. B
- 15. C
- 16. C
- 17. C
- 18. D
- 19. C
- 20. C