

Questões – Protocolo HTTP

Questões objetivas

1. O que significa a sigla HTTP?

- A) HyperText Transfer Protocol
 - B) HyperText Transmission Program
 - C) Hyper Transfer Text Process
 - D) Hyperlink Transfer Protocol
 - E) Host Transfer Text Protocol
-

2. Qual é a principal finalidade do protocolo HTTP?

- A) Gerenciar conexões Bluetooth.
 - B) Permitir a comunicação entre clientes e servidores na Web.
 - C) Armazenar informações em bancos de dados.
 - D) Criptografar arquivos no computador.
 - E) Gerenciar dispositivos de rede local.
-

3. Em uma comunicação HTTP, qual componente normalmente inicia a requisição?

- A) Servidor
 - B) Banco de dados
 - C) Cliente
 - D) Firewall
 - E) DNS
-

4. Qual dos exemplos abaixo representa uma requisição HTTP?

- A) GET /clientes HTTP/1.1
 - B) SELECT * FROM clientes
 - C) System.out.println("Olá")
 - D) INSERT INTO cliente VALUES (...)
 - E) ping 192.168.0.1
-

5. Qual método HTTP é normalmente utilizado para consultar informações?

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

- D) GET
 - E) PATCH
-

6. Qual método HTTP é normalmente utilizado para enviar dados ao servidor para criação de um novo recurso?

- A) GET
 - B) POST
 - C) DELETE
 - D) HEAD
 - E) OPTIONS
-

7. Qual método HTTP é utilizado para excluir um recurso?

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

8. Qual método HTTP é geralmente utilizado para atualizar completamente um recurso?

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

9. Qual método HTTP é utilizado para realizar uma atualização parcial de um recurso?

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

10. Considere a seguinte URL:

<https://www.exemplo.com/clientes/10>

Qual parte representa o protocolo utilizado?

- A) www
 - B) exemplo.com
 - C) /clientes/10
 - D) https
 - E) .com
-

11. Qual código de status HTTP indica que uma requisição foi processada com sucesso?

- A) 200
 - B) 301
 - C) 400
 - D) 404
 - E) 500
-

12. O que normalmente representa o código HTTP 201?

- A) Requisição inválida.
 - B) Recurso não encontrado.
 - C) Recurso criado com sucesso.
 - D) Erro interno do servidor.
 - E) Acesso não autorizado.
-

13. O que significa o código HTTP 400?

- A) Recurso criado.
 - B) Requisição inválida.
 - C) Recurso encontrado.
 - D) Erro interno do servidor.
 - E) Redirecionamento permanente.
-

14. O que significa o código HTTP 401?

- A) Recurso não encontrado.
 - B) Erro interno do servidor.
 - C) Não autorizado/autenticação necessária.
 - D) Requisição processada com sucesso.
 - E) Recurso criado.
-

15. O que significa o código HTTP 403?

- A) O servidor recebeu a requisição, mas recusou o acesso ao recurso.
 - B) O recurso foi criado.
 - C) O recurso não existe.
 - D) O servidor apresentou erro interno.
 - E) A requisição foi redirecionada.
-

16. O que significa o código HTTP 404?

- A) Acesso proibido.
 - B) Requisição inválida.
 - C) Recurso não encontrado.
 - D) Erro interno do servidor.
 - E) Recurso criado.
-

17. O que indica um código HTTP da série 5xx?

- A) Sucesso
 - B) Redirecionamento
 - C) Erro relacionado ao servidor
 - D) Erro relacionado exclusivamente ao cliente
 - E) Recurso criado
-

18. Qual é a função dos Headers em uma requisição HTTP?

- A) Armazenar permanentemente os dados no banco de dados.
 - B) Fornecer informações adicionais sobre a requisição ou resposta.
 - C) Substituir o corpo da requisição.
 - D) Executar código Java no servidor.
 - E) Criar automaticamente uma API REST.
-

19. Observe o Header HTTP:

Content-Type: application/json

O que ele informa?

- A) Que o servidor está utilizando Java.
 - B) Que a resposta possui código 200.
 - C) O tipo de conteúdo que está sendo enviado.
 - D) Que a requisição será excluída.
 - E) Que o conteúdo está criptografado.
-

20. Considere a seguinte requisição:

POST /clientes HTTP/1.1
Content-Type: application/json

```
{  
  "nome": "João",  
  "email": "joao@email.com"  
}
```

Onde estão os dados enviados ao servidor?

- A) No método HTTP.
 - B) Na URL.
 - C) No Header.
 - D) No Body da requisição.
 - E) No protocolo.
-

Questões discursivas

21.

Explique a diferença entre **cliente HTTP** e **servidor HTTP**.

22.

Explique o funcionamento básico de uma comunicação HTTP entre um navegador e um servidor web.

23.

Explique a diferença entre os métodos HTTP **GET** e **POST**.

24.

Explique a diferença entre os códigos HTTP **401**, **403** e **404**.

25.

O que são **Headers HTTP**? Cite dois exemplos e explique suas funções.

26.

Explique o que é o **Body** de uma requisição HTTP e dê um exemplo de situação em que ele pode ser utilizado.

27.

Analise a seguinte requisição:

```
PUT /api/clientes/15 HTTP/1.1
Content-Type: application/json
```

```
{
  "nome": "Maria",
  "email": "maria@email.com"
}
```

Responda:

- a) Qual método HTTP está sendo utilizado?
 - b) Qual é o recurso acessado?
 - c) Qual é o identificador do recurso?
 - d) Qual é o formato dos dados enviados?
 - e) Onde estão os dados enviados?
-

28.

Uma aplicação Java possui o seguinte endpoint:

```
@GetMapping("/clientes/{id}")
public Cliente buscar(@PathVariable Long id) {
    // ...
}
```

Qual método HTTP deve ser utilizado para acessar esse endpoint? Dê um exemplo de requisição.

29.

Uma API REST retorna a seguinte resposta:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "id": 10,
  "nome": "Carlos"
}
```

Explique o significado de:

- a) 200 OK
 - b) Content-Type: application/json
 - c) Corpo da resposta
-

30.

Explique a diferença entre **HTTP** e **HTTPS**.

Gabarito

Questões objetivas

Questão	Resposta	Explicação
1	A	HTTP significa <i>HyperText Transfer Protocol</i> .
2	B	HTTP permite a comunicação entre clientes e servidores na Web.
3	C	Normalmente o cliente inicia a comunicação enviando uma requisição.
4	A	GET /clientes HTTP/1.1 é uma requisição HTTP válida.
5	D	GET é utilizado normalmente para consultar recursos.
6	B	POST é normalmente utilizado para criação de recursos.
7	D	DELETE é utilizado para remover recursos.
8	C	PUT é utilizado convencionalmente para substituição/atualização completa.
9	D	PATCH é utilizado para atualização parcial.
10	D	https identifica o esquema/protocolo da URL.
11	A	200 significa que a requisição foi processada com sucesso.
12	C	201 significa <i>Created</i> , indicando criação de recurso.
13	B	400 indica uma requisição inválida.
14	C	401 indica que a requisição requer autenticação válida.
15	A	403 indica que o acesso foi recusado.
16	C	404 indica que o recurso solicitado não foi encontrado.
17	C	Códigos 5xx representam erros relacionados ao servidor.
18	B	Headers transportam metadados da requisição ou resposta.
19	C	Content-Type informa o tipo/formato do conteúdo.
20	D	Os dados JSON estão no Body da requisição.

Respostas esperadas – discursivas

21.

O **cliente HTTP** é o componente que realiza a requisição, como um navegador, aplicativo mobile ou aplicação Java. O **servidor HTTP** recebe a requisição, processa-a e devolve uma resposta.

22.

O cliente envia uma **requisição HTTP** ao servidor. Essa requisição pode conter método, URL, headers e, dependendo do caso, um body. O servidor processa a requisição e devolve uma **resposta HTTP**, contendo um código de status, headers e, opcionalmente, um body.

23.

GET é normalmente utilizado para consultar ou obter informações. **POST** é normalmente utilizado para enviar dados ao servidor, frequentemente para criação de um novo recurso.

Exemplo:

```
GET /clientes/10
```

Consulta o cliente 10.

```
POST /clientes
```

Pode criar um novo cliente.

24.

- **401 Unauthorized:** a requisição não possui autenticação válida ou requer autenticação.
- **403 Forbidden:** o servidor entendeu a requisição, mas o acesso ao recurso foi recusado.
- **404 Not Found:** o recurso solicitado não foi encontrado.

25.

Headers são informações adicionais que acompanham uma requisição ou resposta HTTP.

Exemplos:

```
Content-Type: application/json
```

Indica o formato do conteúdo.

```
Authorization: Bearer token
```

Transporta informações de autenticação/autorização, conforme o mecanismo utilizado pela aplicação.

26.

O **Body** contém os dados transportados na requisição ou resposta. É muito utilizado em APIs para enviar dados estruturados.

Exemplo:

```
{  
  "nome": "João",  
  "idade": 30  
}
```

27.

- a) O método é **PUT**.
- b) O recurso é `/api/clientes/15`.
- c) O identificador é **15**.
- d) O formato é **JSON**.
- e) Os dados estão no **Body** da requisição.

28.

O método é **GET**.

Exemplo:

`GET /clientes/10`

Nesse caso, o valor **10** será recebido pelo parâmetro `id`.

29.

- a) **200 OK** indica que a requisição foi processada com sucesso.
- b) **Content-Type: application/json** informa que o conteúdo da resposta está no formato JSON.
- c) O Body contém os dados retornados pela API:

```
{
  "id": 10,
  "nome": "Carlos"
}
```

30.

HTTP é o protocolo utilizado para comunicação entre clientes e servidores. **HTTPS** utiliza HTTP sobre uma conexão protegida por **TLS**, proporcionando criptografia e proteção contra alterações e interceptação do tráfego durante o transporte.