

Conceitos básicos do funcionamento do HTTP e modelo requisição-resposta

1. Introdução

O HTTP (HyperText Transfer Protocol) é o protocolo responsável por viabilizar a comunicação entre clientes e servidores na Web. Seu funcionamento se baseia em um modelo simples, porém extremamente eficaz, conhecido como modelo requisição-resposta (request-response). Nesse modelo, um cliente envia uma solicitação a um servidor, que a processa e devolve uma resposta correspondente. Compreender essa dinâmica é essencial para entender como praticamente toda a Web funciona, desde a navegação em sites até o consumo de APIs por aplicativos móveis.

2. O modelo cliente-servidor

No modelo cliente-servidor, o cliente é geralmente um navegador, um aplicativo móvel ou qualquer programa capaz de iniciar uma comunicação. O servidor, por sua vez, é o sistema que armazena os recursos (páginas, imagens, dados) e responde às solicitações recebidas. O HTTP é considerado um protocolo sem estado (stateless), o que significa que cada requisição é tratada de forma independente, sem que o servidor guarde memória de interações anteriores por conta própria. Para contornar essa limitação, mecanismos como cookies e sessões são utilizados quando é necessário manter informações entre diferentes requisições, como o estado de um login.

3. Estrutura de uma requisição HTTP

Uma requisição HTTP é composta por quatro elementos principais: o método, a URL do recurso, os cabeçalhos (headers) e, opcionalmente, um corpo (body) com dados adicionais. Um exemplo simplificado de requisição é apresentado a seguir:

```
GET /produtos/123 HTTP/1.1
Host: www.exemplo.com
User-Agent: Mozilla/5.0
Accept: text/html
```

Nesse exemplo, o cliente solicita, por meio do método GET, o recurso localizado em "/produtos/123" ao servidor identificado por "www.exemplo.com", informando ainda dados sobre o navegador utilizado e o tipo de conteúdo aceito como resposta.

4. Estrutura de uma resposta HTTP

A resposta enviada pelo servidor também segue uma estrutura padronizada, contendo uma linha de status, cabeçalhos e, geralmente, um corpo com o conteúdo solicitado. Um exemplo de resposta é mostrado abaixo:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 1024

<html>...</html>
```

A linha de status indica que a requisição foi processada com sucesso (código 200), enquanto os cabeçalhos informam o tipo e o tamanho do conteúdo retornado. O corpo da resposta traz o conteúdo propriamente dito, neste caso, um documento HTML.

5. O ciclo requisição-resposta

De forma resumida, o ciclo de comunicação HTTP ocorre nas seguintes etapas:

- 1 O cliente estabelece uma conexão com o servidor (geralmente por meio de TCP/IP).
- 2 O cliente envia uma requisição HTTP, especificando o método, o recurso desejado e os cabeçalhos necessários.
- 3 O servidor recebe e processa a requisição, podendo consultar bancos de dados ou executar lógica de negócio.
- 4 O servidor devolve uma resposta HTTP, contendo o código de status e o conteúdo solicitado (ou uma mensagem de erro).
- 5 O cliente processa a resposta recebida, exibindo o conteúdo ou tratando o erro apresentado.

6. Métodos e códigos de status mais comuns

Método	Descrição
GET	Solicita a obtenção de um recurso
POST	Envia dados para criação de um recurso
PUT	Atualiza um recurso existente por completo
DELETE	Remove um recurso

Código	Significado
200	Requisição bem-sucedida
301	Recurso movido permanentemente
400	Requisição inválida (erro do cliente)

404	Recurso não encontrado
500	Erro interno do servidor

7. Exemplos de aplicação

O modelo requisição-resposta pode ser observado em diversas situações do cotidiano digital:

- **Acesso a um site de notícias** — o navegador envia uma requisição GET para o servidor, que responde com o HTML da página, exibindo as manchetes ao usuário.
- **Envio de um formulário de contato** — ao clicar em "enviar", o navegador realiza uma requisição POST com os dados preenchidos (nome, e-mail, mensagem); o servidor processa e responde confirmando o recebimento.
- **Atualização de perfil em uma rede social** — ao alterar a foto de perfil, o aplicativo envia uma requisição PUT ou POST contendo a nova imagem, e o servidor responde indicando sucesso na atualização.
- **Exclusão de um item em uma lista de tarefas** — um aplicativo de produtividade envia uma requisição DELETE referente à tarefa selecionada, e o servidor responde confirmando a remoção do item.
- **Consulta a uma API de previsão do tempo** — um aplicativo de celular faz uma requisição GET a uma API, que responde com dados em formato JSON contendo temperatura, umidade e previsão para os próximos dias.
- **Erro ao acessar uma página inexistente** — quando o usuário digita um endereço incorreto, o servidor responde com o código de status 404, indicando que o recurso não foi encontrado.

8. Considerações finais

O modelo requisição-resposta é a base sobre a qual toda a comunicação na Web é construída. Ao compreender como as requisições são montadas, como o servidor as processa e como as respostas são estruturadas, torna-se possível entender com mais clareza o funcionamento de sites, aplicativos e APIs utilizados diariamente, além de facilitar o diagnóstico de problemas e o desenvolvimento de novas aplicações web.