

UNIVERSIDADE DO ALGARVE

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA

INSTITUTO SUPERIOR DE ENGENHARIA

LICENCIATURA EM ENGENHARIA DE SISTEMAS E

TECNOLOGIAS INFORMÁTICAS

Protocolo HTTP

TECNOLOGIAS WEB

1º Ano, 2º Semestre

2023/2024

Discente: Eline Snijder – 84785

Docente: Roberto Célio Lau Lam

Faro, 25 de março de 2024

Índice

1. Introdução	2
2. Protocolo HTTP.....	2
2.1. Histórico e Evolução	2
2.2. Conceitos Fundamentais	3
3. Funcionamento do HTTP	3
3.1. Request-Response Cycle	4
3.2. Métodos HTTP	5
4. Características do HTTP	5
4.1. Stateless e Stateful	5
4.2. Cabeçalhos HTTP	6
4.3. Cookies e Sessões	7
4.4. Segurança (HTTP vs. HTTPS)	7
5. Considerações e Conclusões	8
5.1. Importância Contínua do Protocolo HTTP	8
6. Perspetivas Futuras	8
7. Bibliografia.....	9
8. Dicionário	9

1. Introdução

A World Wide Web (WWW) tornou-se essencial na vida cotidiana, oferecendo acesso fácil a uma ampla gama de informações e serviços. Por trás dessa acessibilidade está o Protocolo de Transferência de Hipertexto (HTTP), que possibilita a comunicação entre servidores e clientes na web.

Este relatório busca explicar de forma sucinta o funcionamento e a importância do protocolo HTTP na internet, abordando seus conceitos essenciais, evolução ao longo do tempo e características fundamentais, como métodos de requisição, códigos de status, entre outros. O propósito deste relatório é oferecer uma análise do protocolo HTTP, permitindo aos leitores entender a sua função na internet.

2. Protocolo HTTP

O Hypertext Transfer Protocol (HTTP) é um protocolo de comunicação utilizado na World Wide Web para transferir dados, como documentos HTML, entre servidores e clientes, como navegadores da web.

2.1. Histórico e Evolução

O Protocolo de Transferência de Hipertexto (HTTP) é uma tecnologia essencial na World Wide Web (WWW), criada nos anos 1990 para facilitar a transferência de documentos hipermídia entre servidores e clientes, como navegadores da web.

Ao longo do tempo, o HTTP tem sido importante para a comunicação na internet, passando por várias atualizações e melhorias para proporcionar uma experiência de navegação mais eficiente e segura para os usuários.

A primeira versão do HTTP, conhecida como HTTP/0.9, era simples, com solicitações compostas por apenas uma linha iniciada pelo método GET. Em 1996, foi lançada a versão HTTP/1, mas sua limitação de permitir apenas uma requisição por conexão levou ao lançamento do HTTP/1.1 em 1997, introduzindo a persistência de conexão.

Em 2015, surgiu o HTTP/2, trazendo consigo a ideia de fluxos HTTP e o conceito de "push", melhorando significativamente a eficiência e o desempenho da comunicação. Essas evoluções do HTTP demonstram seu papel vital na infraestrutura da web e sua contínua adaptação para atender às demandas dos usuários e da tecnologia em constante evolução.

2.2. Conceitos Fundamentais

O HTTP é um protocolo vital na Internet, permitindo que clientes, enviem solicitações a servidores e recebam respostas. As solicitações contêm informações como método, URL e cabeçalhos, enquanto as respostas incluem códigos de status e dados. URLs identificam recursos, e os métodos HTTP definem a ação a ser executada. O HTTP é sem estado, mas o estado da aplicação pode ser mantido usando cookies, sessões, entre outros.

3. Funcionamento do HTTP

HTTP é um protocolo baseado em texto sem conexão. Isso significa que as pessoas que acessam a um site enviam solicitações a servidores que as exibem na forma do site em formato de texto, imagens, e outros tipos de mídia. Depois que a solicitação é atendida por um servidor, a conexão entre o usuário e o servidor é desconectada.

3.1. Request-Response Cycle

Uma nova conexão deve ser feita para cada solicitação. Explicando melhor, sempre que alguém digita um link de um site no browser, é isto que acontece:

- se a URL pertencer a um domínio próprio, o navegador conecta-se primeiro a um servidor e recuperará o endereço IP correspondente ao servidor;
- o navegador conecta-se ao servidor e envia uma solicitação HTTP para a página de web desejada;
- o servidor recebe a solicitação e verifica a página desejada.
 - Se a página existir, o servidor a mostrará.
 - Se o servidor não conseguir encontrar a página solicitada, ele enviará uma mensagem de erro HTTP 404, ou seja, página não encontrada;
- o navegador, então, recebe a página de volta e a conexão é fechada;
- caso a página exista, o navegador analisa e procura outros elementos necessários para concluir a sua exibição, o que inclui seus textos, imagens e afins;
- para cada um desses elementos, o navegador faz conexões adicionais e solicitações HTTP para o servidor para cada elemento;
- quando o navegador terminar de carregar todos os elementos, a página será carregada na janela do navegador.

Esse ciclo de solicitação-resposta é a base do funcionamento do HTTP e é repetido para cada interação entre cliente e servidor na web. Ele permite a comunicação eficaz e o compartilhamento de recursos entre diferentes dispositivos conectados à internet.

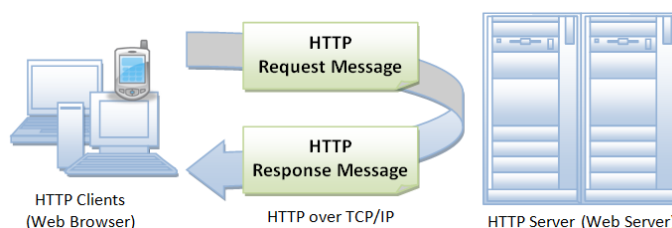


Figura 1 | Request-Response Cycle

3.2. Métodos HTTP

Os métodos HTTP são verbos que indicam a ação que o cliente deseja realizar num recurso específico do servidor. Aqui estão alguns dos principais métodos HTTP e suas funções:

GET – solicita dados de um recurso específico do servidor, permitindo que os clientes recuperem informações sem realizar modificações; **HEAD** - solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta. É usado para obter informações sobre um recurso, como tamanho do arquivo, tipo de conteúdo, data de modificação, entre outros; **POST** - envia dados ao servidor para serem processados ou armazenados, resultando numa alteração no mesmo; **PUT** - substitui todas as representações atuais de um recurso específico no servidor pela carga de dados da requisição; **DELETE** - remove um recurso específico do servidor; **CONNECT** - estabelece um túnel para o servidor identificado no recurso destinado; **OPTIONS** - descreve as opções de comunicação com o recurso de destino; **TRACE** - executa um teste de chamada “loop-back” junto com o caminho para o recurso de destino; **PATCH** - é utilizado para aplicar modificações parciais em um recurso. Envia apenas os dados que precisam ser modificados, ao contrário do método PUT.

4. Características do HTTP

4.1. Stateless e Stateful

A distinção entre Stateless (sem estado) e Stateful (com estado) diz respeito à maneira como um sistema lida com as interações entre diferentes solicitações do cliente.

Stateless - refere-se a um sistema em que cada solicitação é tratada de forma independente, sem armazenar o estado das interações anteriores. No contexto de aplicativos da web, o HTTP é um exemplo de protocolo stateless, onde cada solicitação é processada sem conhecimento do estado anterior do cliente. Isso traz benefícios como a redução do uso de memória no servidor.

Stateful - envolve sistemas que mantêm o estado das interações anteriores do cliente com o servidor. Por exemplo, em aplicativos da web que requerem login, o servidor mantém informações sobre a sessão do usuário. Isso permite controlar o usuário e pode melhorar o desempenho ao recuperar dados apenas uma vez.

A escolha entre uma abordagem Stateless ou Stateful depende dos requisitos do aplicativo e das necessidades de complexidade e segurança.

4.2. Cabeçalhos HTTP

Os cabeçalhos HTTP são componentes fundamentais das mensagens HTTP, fornecendo informações adicionais sobre a solicitação ou resposta. Existem quatro tipos de cabeçalhos HTTP.

Genérico – usados tanto em solicitações como em respostas, porém sem relação com os dados transmitidos no corpo da mensagem; **Solicitação** – contém mais informação sobre o recurso a ser obtido ou sobre o próprio cliente; **Resposta** – contém informação adicional sobre a solicitação, como a sua localização ou sobre o servidor; **Entidade** – contém mais informação sobre o conteúdo da entidade tal como o tamanho do conteúdo.

4.3. Cookies e Sessões

O protocolo HTTP é stateless, o que significa que não mantém um estado ou conexão contínua entre o cliente e o servidor. Para contornar essa limitação, são utilizados cookies e sessões.

Uma sessão em uma aplicação web é um estado temporário mantido pelo servidor para facilitar a interação contínua entre o usuário e a aplicação, identificado por um ID único enviado ao navegador por meio de um cookie. Durante a sessão, informações como dados de login e preferências do usuário são armazenados e acessados pelo servidor.

Já um cookie é um pequeno arquivo de dados armazenado pelo navegador, enviado pelo servidor durante a primeira interação e devolvido em solicitações seguintes. Os cookies contêm informações como o ID da sessão e preferências do usuário, permitindo uma interação personalizada entre o usuário e a aplicação web.

4.4. Segurança (HTTP vs. HTTPS)

A diferença entre HTTP (Hypertext Transfer Protocol) e HTTPS (Hypertext Transfer



Protocol Secure) está na segurança proporcionada pelo HTTPS. Enquanto o HTTP é um protocolo de transferência de hipertexto não seguro, o HTTPS representa uma versão segura do mesmo, incorporando mecanismos de criptografia para proteger os dados transmitidos entre o cliente e o servidor.

O SSL no HTTPS protege as informações trocadas entre as partes contra ameaças como



espionagem, garantindo a segurança de dados sensíveis, como credenciais de login e detalhes de pagamento. Além da criptografia, o HTTPS oferece autenticação, assegurando que o cliente se comunique com o servidor correto, aumentando a segurança.

5. Considerações e Conclusões

5.1. Importância Contínua do Protocolo HTTP

O HTTP continua a ser um componente vital da World Wide Web, mesmo com o surgimento de novas tecnologias. Facilitando a comunicação entre clientes e servidores de forma eficiente e padronizada, a sua simplicidade e flexibilidade o mantêm como um dos principais protocolos da web. No entanto, com a crescente preocupação com a segurança dos dados, o HTTPS, que oferece criptografia, tornou-se essencial. O HTTP está evidente na evolução contínua para atender às necessidades dos usuários e desenvolvedores.

6. Perspetivas Futuras

O HTTP está evoluindo para tornar a web mais rápida, segura e acessível. O HTTP/3, baseado no protocolo QUIC, promete melhorias significativas, especialmente em redes móveis. Além disso, novos formatos de dados, como streaming de vídeo, continuarão a depender do HTTP. A integração com tecnologias emergentes, como IoT e IA, será essencial para suportar casos de uso avançados. Concluindo, o futuro do HTTP aponta para uma web mais eficiente e adaptada às necessidades dos usuários e desenvolvedores.

7. Bibliografia

- Alexandre Marotel, (2022), Definição HTTP, <https://www.twaino.com/pt/definicao/h/definicao-http/> (Acedido a 21 de março de 2024)
- Ivan de Souza, (2019), <https://rockcontent.com/br/blog/http/> (Acedido a 22 de março de 2024)
- Phil Meadows, (2022), Capítulo 1 – Introdução ao HTTP e ao HTTPS, [Internet], <https://learn.microsoft.com/pt-pt/azure/rto/netx-duo/netx-duo-web-http/chapter1> (Acedido a 22 de março de 2024)
- Mdn web docs __, (2018), <https://developer.mozilla.org/pt-BR/docs/Web/HTTP> (Acedido a 21 de março de 2024)
- Bruna, (2024), <https://www.hostinger.com.br/tutoriais/https-vs-http> (Acedido a 23 de março de 2024)
- https://gitbook.ganeshicmc.com/web/semana-1/11_cookies_e_sessoes (Acedido a 23 de março de 2024)

8. Dicionário

“loop-back” – Refere-se ao encaminhamento de fluxo de dados digitais que retornam para a sua origem, sem processamento ou modificação.

“QUIC” – Quick UDP Internet Connections

“SSL” - Camada de Soquetes Seguros (Secure Sockets Layer)

“URL” - Localizador de Recursos Uniforme (Uniform Resource Locator)