HTTP e HTTPS

Conceitos Gerais

1. Introdução

- A camada de aplicação possui uma série de protocolos que objetivam oferecer diversos tipos de serviços ao usuário.

- Entre eles, podemos citar o protocolo de DNS para solução de nomes, SMTP/POP para email, DHCP para endereçamento automático, entre muitos outros.

- Os protocolos HTTP/HTTPS também são muito importantes, formam a base da World Wide Web, utilizado por navegadores para carregar páginas web. O HTTPS é a versão segura, utilizando criptografia (SSL/TLS).

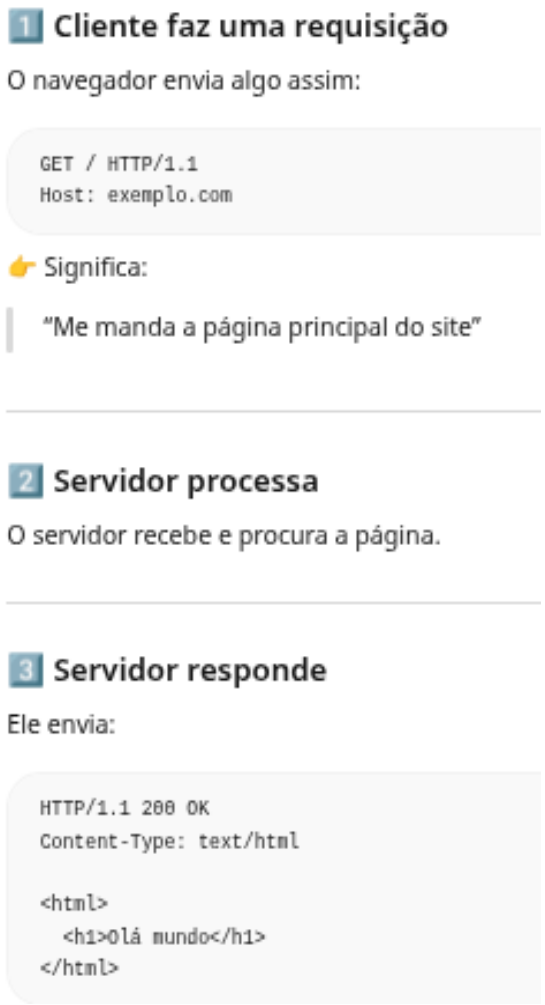
2. HTTP

- O **HTTP (Hypertext Transfer Protocol)** é o protocolo usado para **transferir páginas web** na internet. Ele define **como cliente e servidor conversam**.

- Na comunicação via HTTP, o Cliente geralmente é o navegador, e o Servidor é onde está o site. O HTTP funciona no modelo **Requisição (Request) → Resposta (Response).**

2. HTTP

- As imagens mostram uma sequência de passos de funcionamento do HTTP:



1️⃣ Cliente faz uma requisição

O navegador envia algo assim:

```
GET / HTTP/1.1
Host: exemplo.com
```

👉 Significa:

> "Me manda a página principal do site"

2️⃣ Servidor processa

O servidor recebe e procura a página.

3️⃣ Servidor responde

Ele envia:

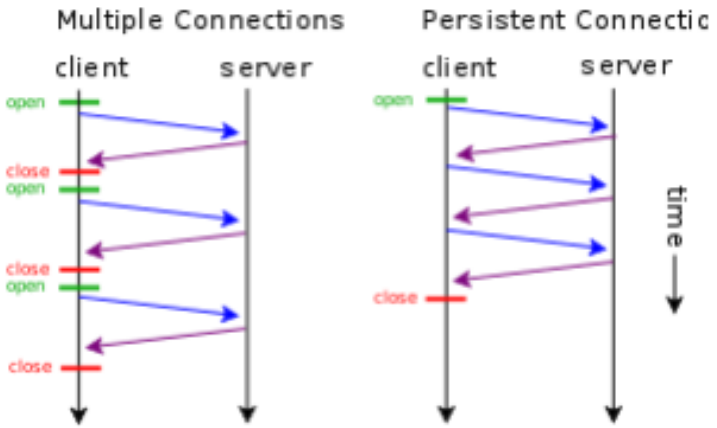
```
HTTP/1.1 200 OK
Content-Type: text/html

<html>
  <h1>Olá mundo</h1>
</html>
```

2. HTTP

- O HTTP possui duas versões:

- **HTTPv1.0** – Não realiza conexões persistentes. Isto é, para cada troca de informação entre cliente e servidor, necessita-se estabelecer e encerrar uma nova conexão TCP. Antiga e defasada.

- **HTTPv1.1** – Realiza conexões persistentes. Estabelece-se apenas uma requisição TCP para a troca de diversas mensagens entre o cliente e servidor. Além disso, pode-se enviar mais de uma requisição sem necessariamente aguardar a confirmação da requisição anterior.



2. HTTP

- Outra característica do protocolo HTTP é o fato dele ser considerado um protocolo **sem estado (stateless)**, pois não armazena informações do usuário.

- Um ponto importante a mencionar é que o servidor pode enviar informações ao usuário com vistas a manter a sessão entre eles aberta, além de poder recuperar certas informações futuramente.

- Esse recurso pode ser provido com o uso de COOKIES, que podem ser armazenados no browser do cliente.

- Assim tem-se um ambiente **statefull**, porém, vale lembrar que isso é um recurso complementar. O HTTP nativamente é **stateless**.

3. Estrutura da Mensagem HTTP

- Como visto, existem dois tipos de mensagem HTTP: requisição e resposta. A imagem abaixo mostra a requisição:

3. Estrutura da Mensagem HTTP

- A imagem abaixo mostra a resposta:

4. Métodos do HTTP

- Cada método é responsável por determinar o tipo de requisição feita e a forma como o dado será tratado. A tabela abaixo mostra alguns dos principais:

Método	Ação principal	Altera servidor?	Idempotente?
GET	Buscar dados	❌	✔
POST	Criar recurso	✔	❌
PUT	Atualizar tudo	✔	✔
PATCH	Atualizar parcial	✔	⚠
DELETE	Remover	✔	✔
HEAD	Ver metadados	❌	✔
OPTIONS	Ver opções	❌	✔

5. Códigos de Estado do HTTP

- Os códigos de estado são definidos em classes, conforme a seguir, com a descrição dos principais códigos:

1xx - Classe informacional - Esta classe indica uma resposta provisória, que consiste de informações do estado da requisição e cabeçalhos opcionais.

2xx - Classe de Sucesso - Indica que a requisição foi recebida, entendida, aceita e processada.

3xx - Classe de Redirecionamento - Indica a necessidade de atuação por parte do cliente HTTP para completar a requisição. Pode ou não ser o caso de atuação direta do usuário.

4xx - Classe de Erro de Cliente - Indica a possibilidade de que houve um erro na requisição por parte do cliente. Caso não seja uma requisição com método HEAD, o servidor enviará uma explicação da situação do erro e se esta é permanente ou temporária.

5. Códigos de Estado do HTTP

- A imagem mostra alguns dos principais erros:

400 (BAD REQUEST) - A requisição não pode ser entendida pelo servidor devido erro de sintaxe.

401 (UNAUTHORIZED) - A requisição depende de autenticação por parte do usuário.

403 (FORBIDDEN) - O servidor entendeu a requisição, mas se recusa a atendê-la. Pode ser enviado a descrição do motivo da recusa.

404 (NOT FOUND) - O servidor não encontrou nenhum documento que coincida com a URI informada.

5. Códigos de Estado do HTTP

- A imagem mostra alguns dos principais erros da categoria 5 (erro no servidor):

5xx - Classe de Erro de Servidor - Indica que o servidor reconheceu um erro interno ou a incapacidade de atender a requisição.

500 (INTERNAL SERVER ERROR) - Erro inesperado que impediu o atendimento a requisição.

503 (SERVICE UNAVAILABLE) - Servidor está incapacitado de atender as requisições devido à sobrecarga ou manutenção. Indica uma condição temporária.

505 (VERSION NOT SUPPORTED) - O servidor não suporta ou não está habilitado a responder para a versão requisitada. O servidor indica o motivo do erro, além de informar as versões que são suportadas e permitidas.

6. Cache WEB

- O funcionamento do CACHE WEB reside na possibilidade de otimização do procedimento de Requisição e Resposta entre o cliente e o servidor.

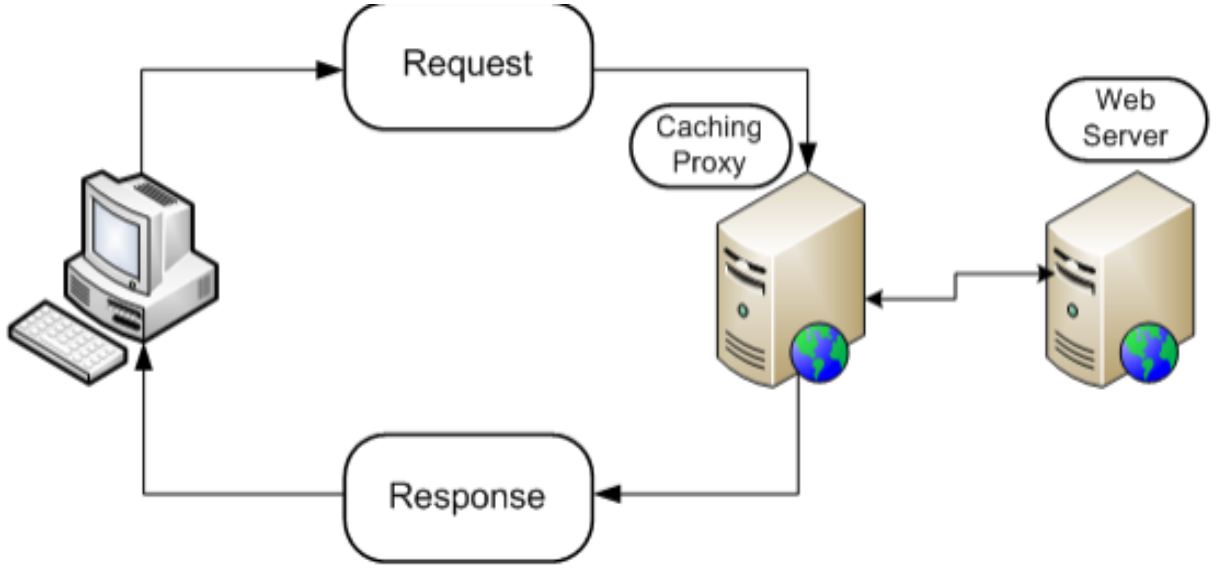
- Esse CACHE WEB busca evitar que novas consultas que sejam idênticas a consultas anteriores consumam recursos do servidor de destino, além de diminuir o tempo de resposta.

- Temos basicamente 3 formas principais de implementar o conceito de cache web, são elas: **Servidor Proxy, Proxy Reverso e Cache Local.**

6. Cache WEB

- **Servidor Proxy:** Pode-se adicionar um elemento intermediário entre o cliente e o servidor, de tal forma que as consultas necessariamente passem pelo nó intermediário antes de chegar ao destino.

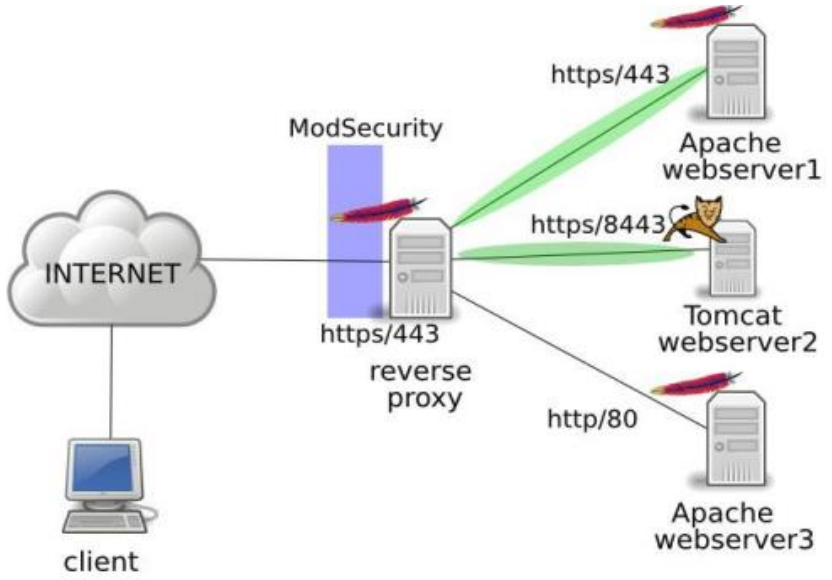
- Esse nó, é chamado de Proxy e armazena as últimas informações requisitas pelos clientes aos servidores. Caso haja uma nova requisição em que o proxy possua as informações para resposta, este não repassará a consulta ao servidor, atendendo a requisição imediatamente.



6. Cache WEB

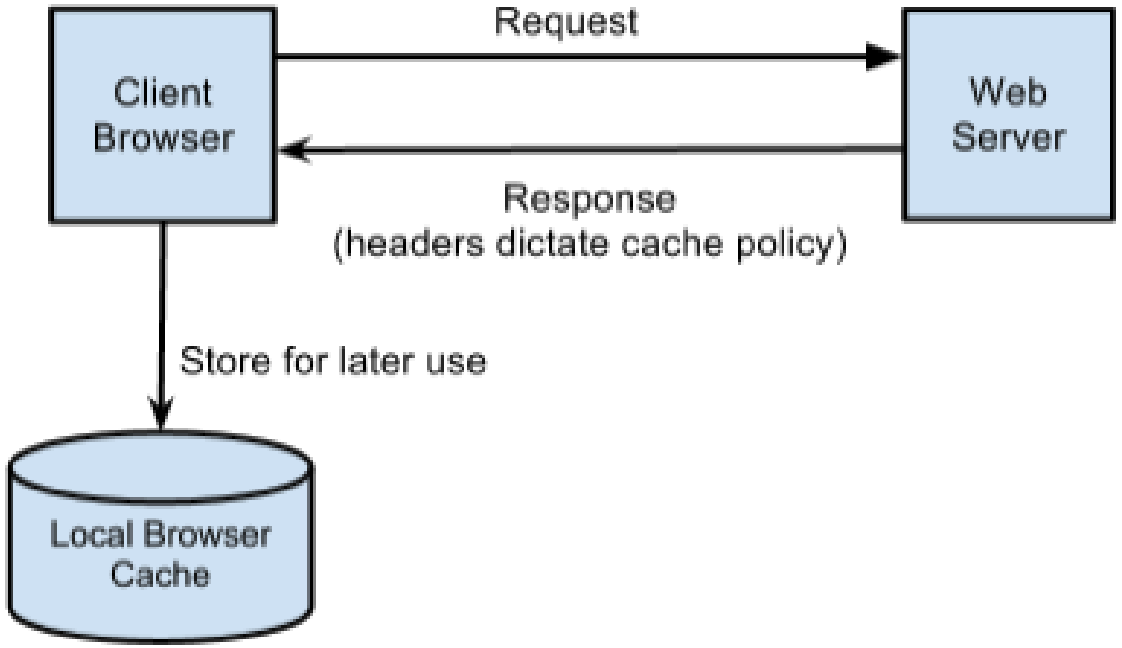
- **Proxy Reverso:** Oferece benefícios em termos de proteção, balanceamento e distribuição de requisições e armazenamento em cache das informações estáticas.

- Quando há uma requisição a um objeto estático, o proxy reverso é capaz de responder diretamente à requisição. Já quando há uma requisição a objetos dinâmicos, este repassa a requisição aos servidores internos conforme a porta utilizada do serviço específico.



6. Cache WEB

- **Cache Local:** Os browsers possuem a capacidade de armazenar as informações recebidas do servidor de tal forma que uma nova requisição idêntica à anterior não enseje uma nova consulta ao servidor. Desse modo, a requisição será atendida diretamente pelo Browser.



7. HTTPS

- O **HTTPS** é a versão **segura** do HTTP. A principal diferença entre eles é que no **HTTP** os dados trafegam em texto puro (qualquer um pode ver), enquanto no **HTTPS** os dados trafegam **criptografados.**

- O que o HTTPS garante:

1. **Confidencialidade:** Ninguém consegue ler os dados (ex: senhas)

2. **Integridade:** Os dados não podem ser alterados no caminho

3. **Autenticação:** Você sabe com quem está se comunicando

8. HTTP 2

- O **HTTP/2** é uma versão mais moderna do HTTP, criada para **Aumentar a velocidade, Melhorar o uso da conexão e Reduzir latência.** Ele funciona normalmente sobre HTTPS. A tabela abaixo mostra as principais diferenças entre as versões:

Característica	HTTP/1.1	HTTP/2
Formato	Texto	Binário
Requisições	Sequenciais	Paralelas
Conexões	Várias	Uma
Compressão headers	✗	✓
Server Push	✗	✓