INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

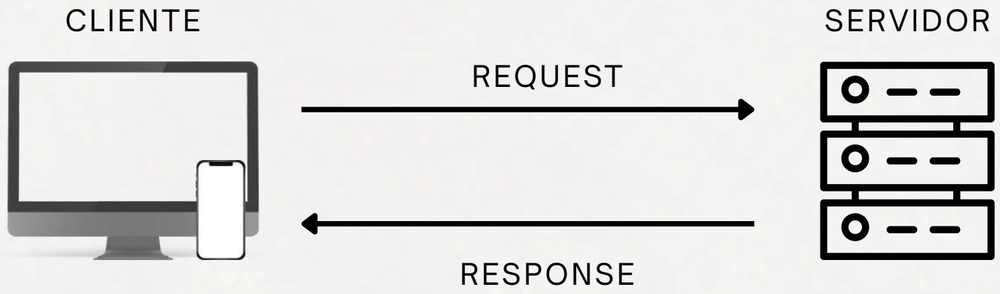
HTTP

HYPERTEXT TRANSFER PROTOCOL

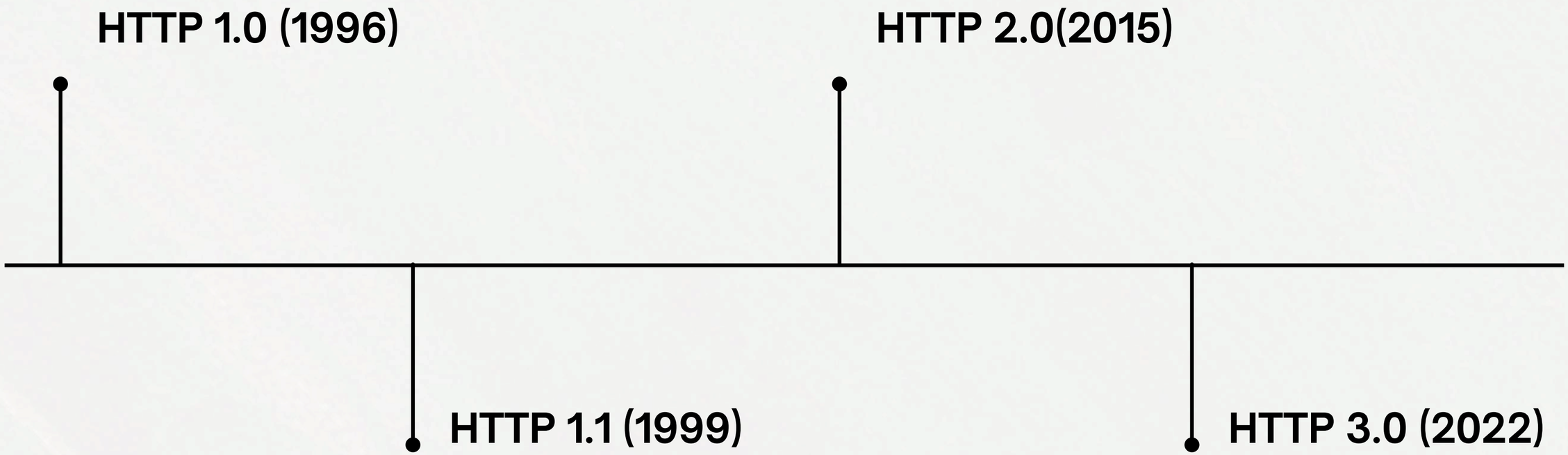
Natan Lopes

Protocolo HTTP

O HTTP (Hypertext Transfer Protocol) é um protocolo de aplicação que permite a comunicação entre clientes (navegadores, apps) e servidores web. Ele opera sobre TCP/IP e segue um modelo request-response (solicitação-resposta).



Verssões HTTP



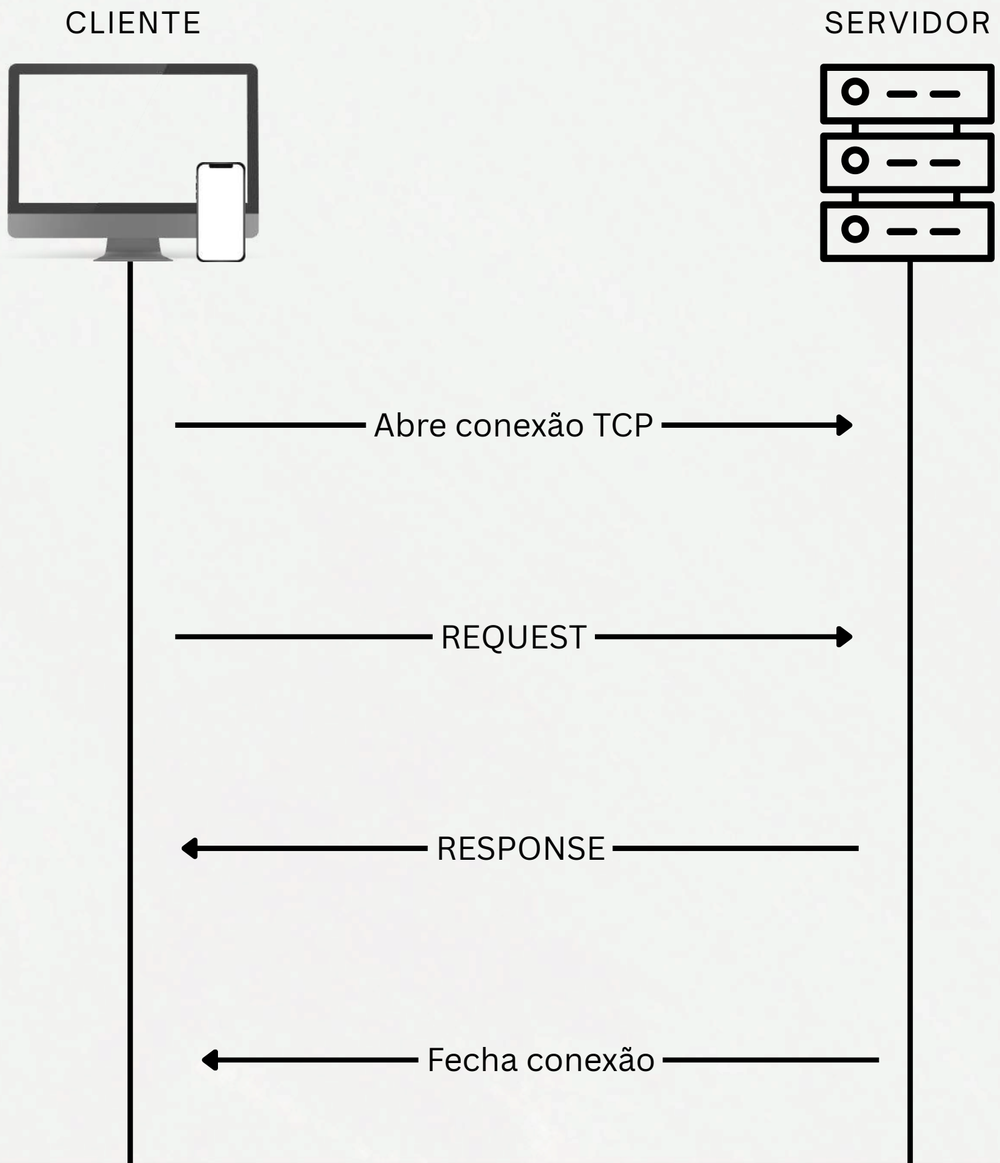
HTTP 1.0

- Introdução de headers (metadados como Content-Type, User-Agent).
- Suporte a métodos adicionais (POST, HEAD).
- Códigos de status (200 OK, 404 Not Found).
- Suporte a outros formatos (imagens, JS, CSS via Content-Type).

Problemas

- Conexão TCP fechada após cada resposta.

HTTP 1.0


CLIENTE
SERVIDOR
Abre conexão TCP
REQUEST
RESPONSE
Fecha conexão

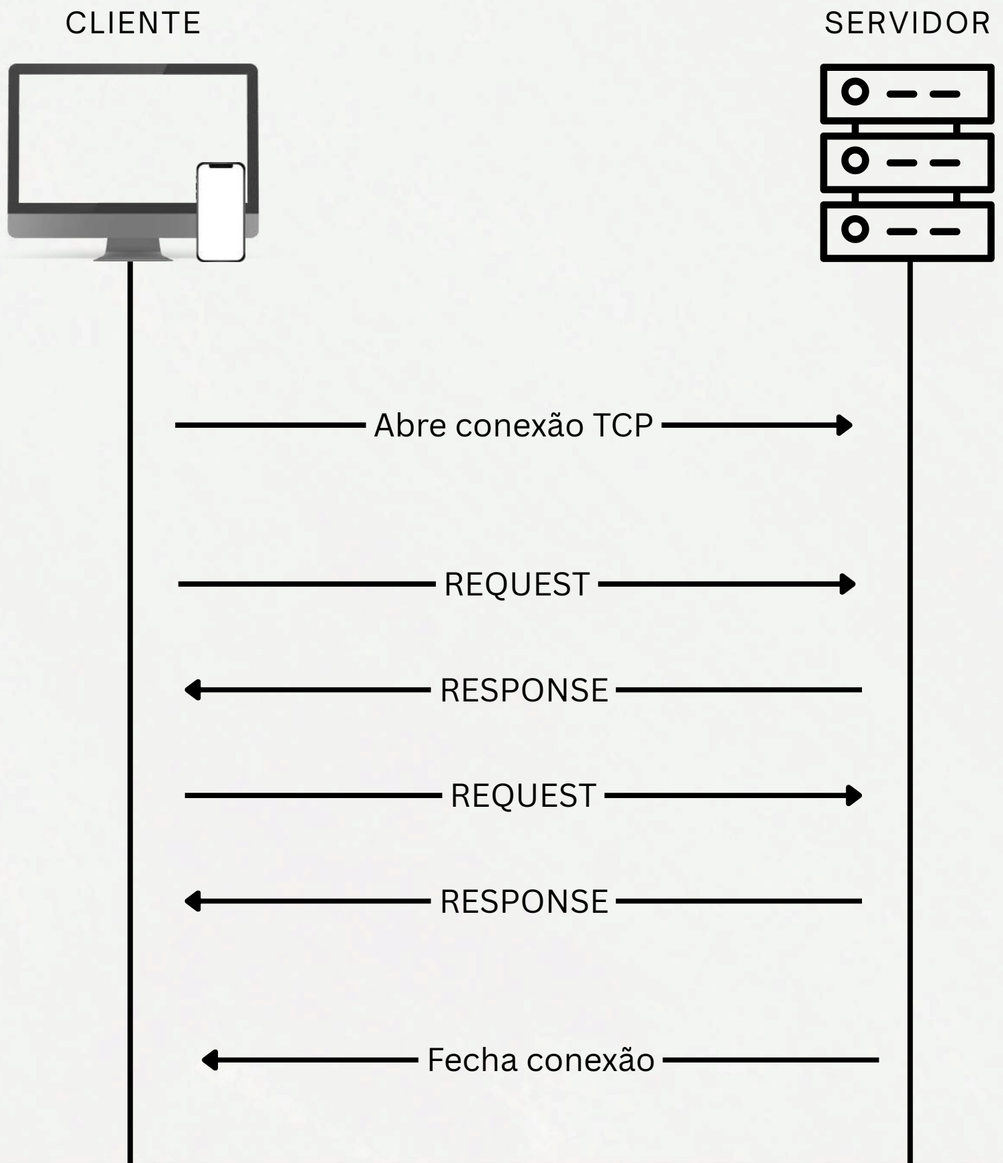
HTTP 1.1

- Conexões persistentes (keep-alive): Reutiliza a mesma conexão TCP para múltiplas requisições.
- Host Virtual: Permite múltiplos sites no mesmo IP (via header Host).
- Novos métodos: PUT, DELETE, OPTIONS, PATCH.

Limitações

- Requisições são processadas em sequência (uma atrasa as outras).

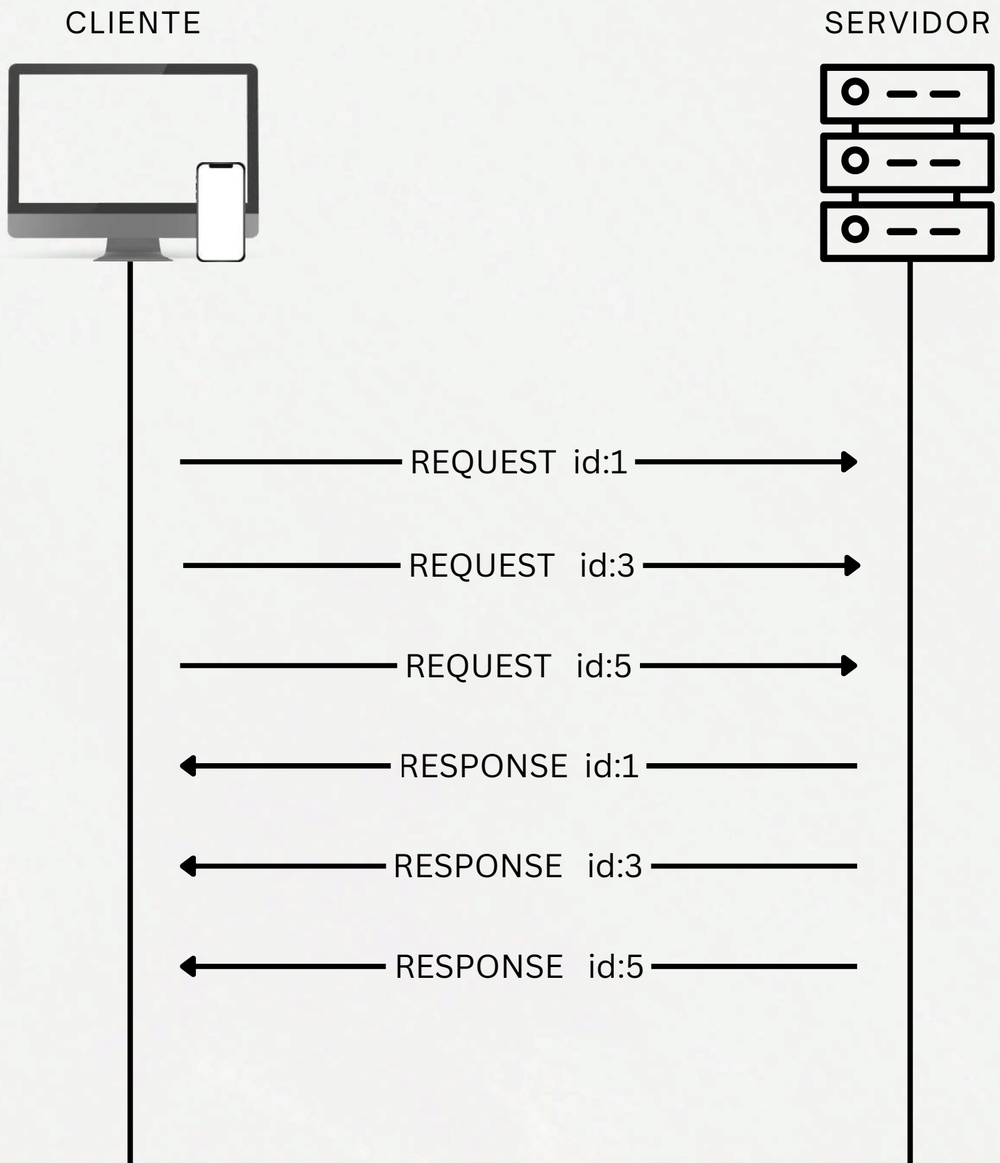
HTTP 1.1


CLIENTE
SERVIDOR
Abre conexão TCP
REQUEST
RESPONSE
REQUEST
RESPONSE
Fecha conexão

HTTP 2.0

- Multiplexação: Várias requisições/respostas em uma única conexão TCP.
- Compressão de headers (HPACK): Reduz overhead.
- Server Push: Envio proativo de recursos (ex: CSS/JS antes do HTML ser processado).
- Priorização de streams: Dá preferência a recursos críticos (ex: HTML antes de imagens).
- Dependência do TCP (retransmissão de pacotes perdidos atrasa toda a conexão)

HTTP 2.0


CLIENTE
SERVIDOR
REQUEST id:1
REQUEST id:3
REQUEST id:5
RESPONSE id:1
RESPONSE id:3
RESPONSE id:5

HTTP 3.0 (QUIC)

- Uso do QUIC ao inves do protocolo TCP.
- Conexões mais rápidas.
- Pacotes perdidos não bloqueiam outros streams.

Principais Verbos HTTP

- **GET:** Solicita dados (não modifica o servidor).
- **POST:** Envia dados para processamento (ex: formulários).
- **PUT:** Atualiza/replace um recurso existente.
- **DELETE:** Remove um recurso
- **PATCH:** Atualização parcial de um recurso.
- **HEAD:** Similar ao GET, mas retorna apenas headers (sem corpo).
- **OPTIONS:** Lista métodos suportados pelo servidor.

Principais servidores HTTP

- **Apache:** Open-source e altamente modular.
- **Nginx:** Focado em alta performance e baixo consumo de recursos.
- **LiteSpeed:** Alternativa otimizada ao Apache, com suporte nativo a HTTP/3 (QUIC)
- **Microsoft IIS:** Servidor da Microsoft para ambientes Windows Server.

Virtual host - Apache

- **Criar o Diretório do Site**

```
/var/www/meusite.com/public_html
```

Virtual host - Apache

- **Criar o Arquivo de Virtual Host**

```
sudo nano /etc/apache2/sites-available/meusite.com.conf
```

```
<VirtualHost *:80>                    # Define um host virtual na porta 80 (HTTP)
    ServerName meusite.com            # Domínio principal do site
    ServerAlias www.meusite.com       # Aliases (subdomínios alternativos)
    DocumentRoot /var/www/meusite.com/public_html  # Diretório raiz do site
</VirtualHost>
```

Virtual host - Apache

- **Ativar o Site e Recarregar o Apache**

```
sudo a2ensite meusite.com.conf  # Habilita o site

sudo systemctl restart apache2
```

Virtual host - Nginx

```
server {
    listen 80;                      # Escuta na porta 80 (HTTP)
    server_name meusite.com www.meusite.com;  # Domínios atendidos
    root /var/www/meusite.com/html;     # Diretório raiz do site
    index index.html;                   # Arquivo padrão ao acessar a raiz
}
```

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Obrigado!