



# **Pesquisa sobre as camadas de aplicação - Web e HTTP.**

Nome: Lucas Silveira - 202402390562, Alexandre Filho - 202404434842, Bruno Moraes - 202404492567, Marisson Negrão Farias - 202404201236

Orientador: Paulo Tassio de Luz Melo



# Sumário

- Descrição Geral do HTTP
- Conexões persistentes e não persistentes
- Formato da mensagem HTTP
- Interação Usuário - Servidor: cookies
- Caches Web
- GET Condicional



## Descrição Geral do HTTP

- O HTTP, Protocolo de Transferência de Hipertexto, é um protocolo responsável pela transferência de dados na Internet, criando bases para a conexão entre o cliente eo servidor, por isso funciona no modelo de cliente-servidor. É um protocolo sem estado, onde o servidor não mantém nenhum dado entre duas requisições. Baseado em uma camada TCP/IP, pode ser usado em qualquer camada de transporte , que não perde mensagens como o UDP.



## Conexões persistentes e não persistentes

- As conexões persistentes permitem que vários objetos sejam solicitados dentro de uma mesma conexão. Nessas conexões, o cliente envia solicitações assim que encontra um objeto referenciado.
- RTT é o tempo que a informação leva para viajar do cliente ao servidor e voltar.
- Sem pipeline: estabelece uma conexão que leva dois RTTs e depois envia todos os arquivos de imagem/texto do objeto, que levam 1 RTT cada. Onde o TCP não é necessário para cada objeto.
- Com pipeline: Os dois RTTs são para o estabelecimento de conexão e o 1 RTT, que não há limites de janela, para todos os objetos.

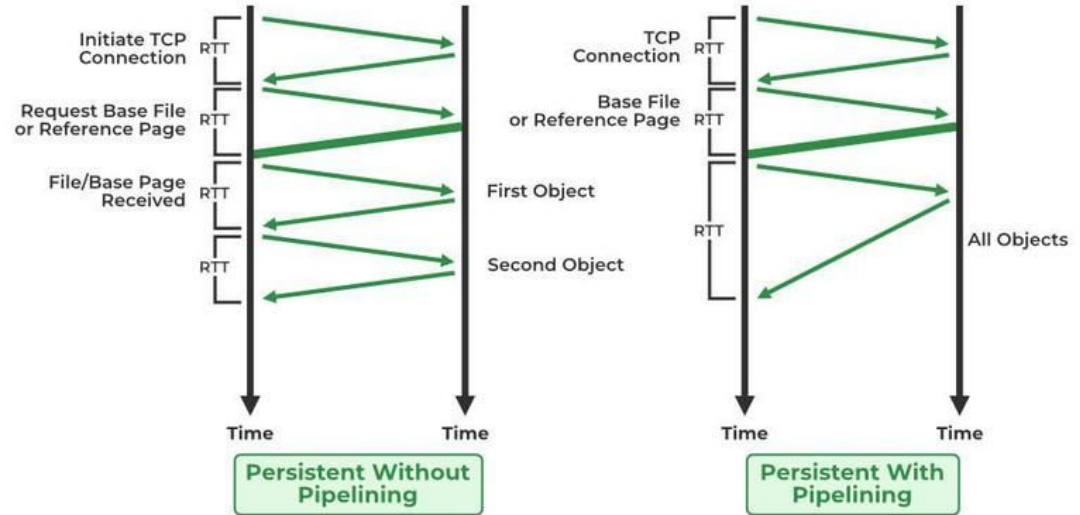


## Conexões persistentes e não persistentes

- As conexões não persistentes são as que para cada objeto tem que criar uma nova conexão para enviar um objeto da origem para seu destino. Geralmente, os navegadores abrem as conexões TCP paralelas para buscar objetos referenciados.
- Sem paralela: cada objeto leva dois RTTs, um para conexão TCP e outro para arquivo de imagem/texto HTTP.
- Com paralela: requer sobrecarga extra na transferência de dados.

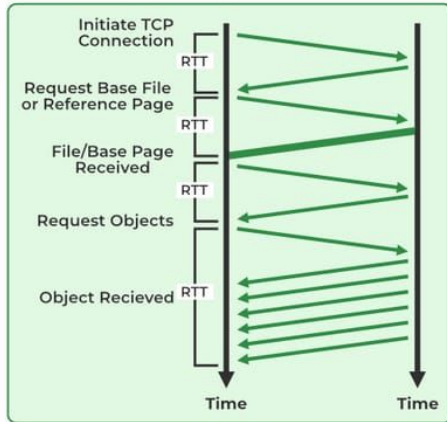
# Conexões persistentes e não persistentes

- Exemplo de conexão com pipeline e sem pipeline.



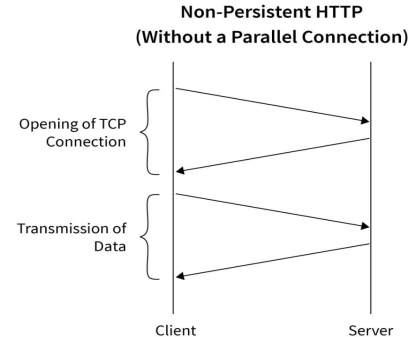
# Conexões persistentes e não persistentes

Exemplo de conexão  
paralela



**Non-Persistent  
&  
Parallel Connections**

Exemplo de conexão  
não paralela





## Formato da mensagem HTTP

O HTTP possui duas formas estruturais para mensagens, são elas: formato de mensagem de requisição e o formato de mensagem de resposta.

- Requisição: Mensagem enviada pelo cliente para solicitar um recurso ao servidor.
- Resposta: Mensagem enviada pelo servidor em resposta a solicitação do cliente.





# Formato da mensagem HTTP

## Requisição:

As requisições HTTP são usadas pelos clientes para solicitar recursos aos servidores. Uma requisição geralmente contém as seguintes informações:

- Linha de Requisição: Linha de texto composta pelo método, o recurso solicitado, e a versão do protocolo HTTP.
- Método: Define o que o cliente solicita ao servidor. Pode ser a visualização do conteúdo, substituição, alteração ou exclusão de recursos.
- Recurso: Página web, arquivo, serviço ou outro item solicitado ao servidor.
- Cabeçalho: Informações adicionais sobre a requisição. Ex: Navegador, tipo de conteúdo, etc.



# Formato da mensagem HTTP

## Resposta:

As respostas são usadas pelo servidor para responder às requisições dos clientes. Uma resposta, geralmente, contém as seguintes informações:

- Código da resposta: Status da requisição. Indica o resultado de uma requisição, pertencendo a uma classe.
- Cabeçalho: Informações adicionais da resposta. Ex: Tipo de conteúdo, tempo de resposta, data, etc.
- Corpo da resposta: Conteúdo da requisição feita. Ex: HTML, TXT, etc.

# Formato da mensagem HTTP

## - Mensagem de Requisição

Método      Recurso / Página      Versão

```
GET /index.html HTTP/1.1  
Host: www.example.com  
User-Agent: curl/8.2.1  
Accept: */*
```

} Linha de Requisição (Request Line)

} Cabeçalho (Header)

## - Mensagem de Resposta

Código de Resposta

```
HTTP/1.1 200 OK  
Accept-Ranges: bytes  
Cache-Control: max-age=604800  
Content-Type: text/html; charset=UTF-8  
Date: Tue, 23 Jan 2024 09:25:38 GMT  
ETag: "3147526947"  
Expires: Tue, 30 Jan 2024 09:25:38 GMT  
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT  
Connection: keep-alive
```

} Cabeçalho (Header)

```
<!doctype html>  
<html>  
<head>  
  <title>Example Domain</title>  
</head>  
<body>  
  <h1>Example Domain</h1>  
</body>  
</html>
```

} Corpo (Body)



## Interação Usuário - Servidor: cookies

- Cookies são partes de dados que são enviados do servidor web para o navegador do usuário, eles tem a função de determinar o status de login e acesso no servidor, além de salvar informações que o usuário deixou na página. Através dos cookies, o servidor web é capaz de trocar informações com o navegador do usuário



# Caches Web

- Os caches web são mecanismos que armazenam temporariamente cópias de dados, páginas e recursos da internet para melhorar a eficiência da navegação. Quando acessamos uma página web, o navegador armazena partes dessa página, como imagens, arquivos CSS e JavaScript, permitindo que futuras visitas ao mesmo site sejam mais rápidas e fluidas.
- Existem vários tipos de cache web, entre os principais estão: cache do navegador, cache de proxy e cache de CDN.



# Caches Web

- Cache do Navegador: Normalmente armazenado no seu dispositivo para acelerar o carregamento de sites que você já visitou.
- Cache de Proxy: Usado por servidores intermediários (proxies) para armazenar conteúdo acessado por vários usuários, permitindo que os dados sejam entregues mais rapidamente a partir do proxy.
- Cache de CDN (Content Delivery Network): Armazena conteúdo em vários servidores distribuídos ao redor do mundo, permitindo que os usuários acessem os dados a partir de um servidor fisicamente mais próximo, diminuindo a latência.



## GET Condicional

- O GET condicional é uma técnica usada em requisições HTTP para otimizar o fluxo de dados entre o cliente eo servidor. Ele permite que o navegador solicite informações apenas se elas tiverem sido modificadas desde a última vez que foram baixadas.
- Quando o navegador faz uma solicitação GET condicional, o servidor pode responder de duas formas: Caso o conteúdo tenha sido modificado e caso o conteúdo não tenha sido modificado.



## GET Condicional

- Caso o conteúdo não tenha sido modificado: O servidor responde com o código de status HTTP 304 Not Modified e não envia o conteúdo novamente, permitindo que o navegador utilize a versão já armazenada em cache.
- Caso o conteúdo tenha sido modificado: O servidor responde normalmente com o novo conteúdo.





# GET Condicional

## Vantagens:

- Melhora a eficiência: Reduz o tráfego de rede, evitando o download de conteúdo desnecessário.
- Acelera o tempo de resposta: O navegador pode usar os recursos em cache, o que torna o processo mais rápido.
- Diminui a sobrecarga no servidor: O servidor processa menos dados e consome menos recursos ao responder com o código 304.



# Referências

<https://www.hostinger.com.br/tutoriais/http#:~:text=HTTP%20%C3%A9%20a%20sigla%20para,no%20modelo%20computacional%20cliente%2Dservidor>

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>

<https://www.ibm.com/docs/pt-br/cics-ts/6.x?topic=concepts-persistent-connections>

<https://www.geeksforgeeks.org/http-non-persistent-persistent-connection/>

<https://dev.to/ortizdavid/protocolo-http-2cbk>

<https://www.cloudflare.com/pt-br/learning/cdn/what-is-caching/>

<https://www.hostnet.com.br/blog/o-que-e-cache-e-qual-a-sua-importancia-na-hospedagem-de-sites/>

<https://www.azion.com/pt-br/blog/o-que-e-requisicao-condicional/>