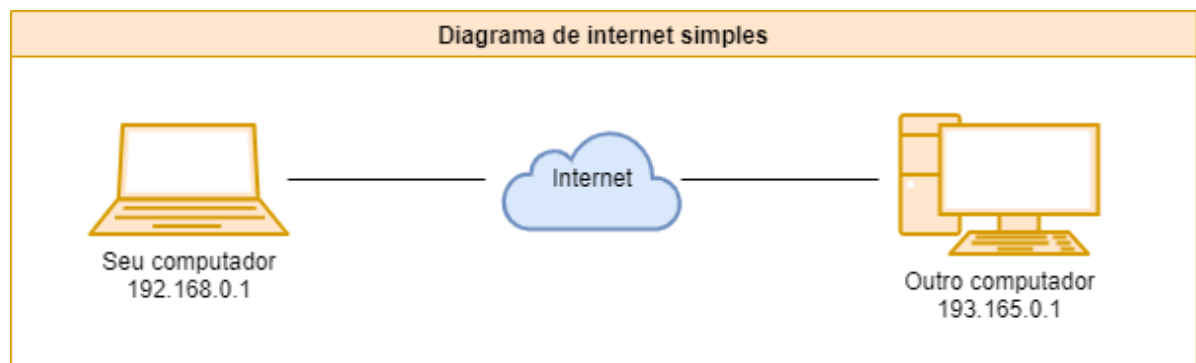


INTRO-CS-1 - Introdução à Ciência da Computação

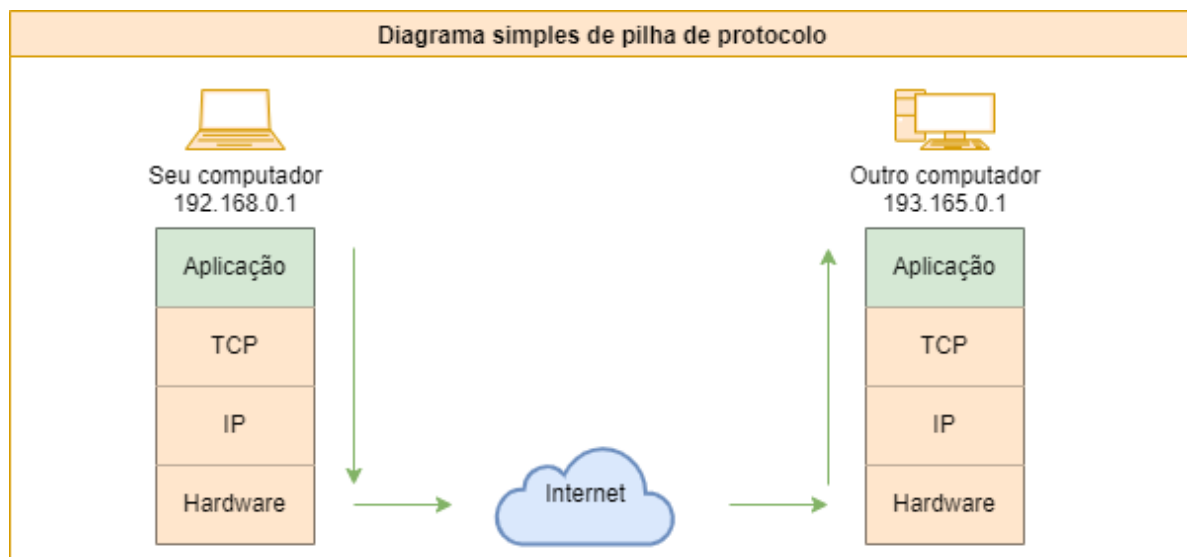
1. Internet
2. Cliente e Servidor
3. TCP/IP
4. HTTP
5. Verbos
6. Status code

1. Internet

É uma rede global de computadores, cada computador conectado à Internet **deve** ter um endereço exclusivo. Os endereços da Internet estão no formato **nnn.nnn.nnn.nnn**, onde nnn deve ser um número de 0 a 255. Esse endereço é conhecido como endereço IP (IP significa Internet Protocol). A seguir segue uma ilustração simples de como é a internet:

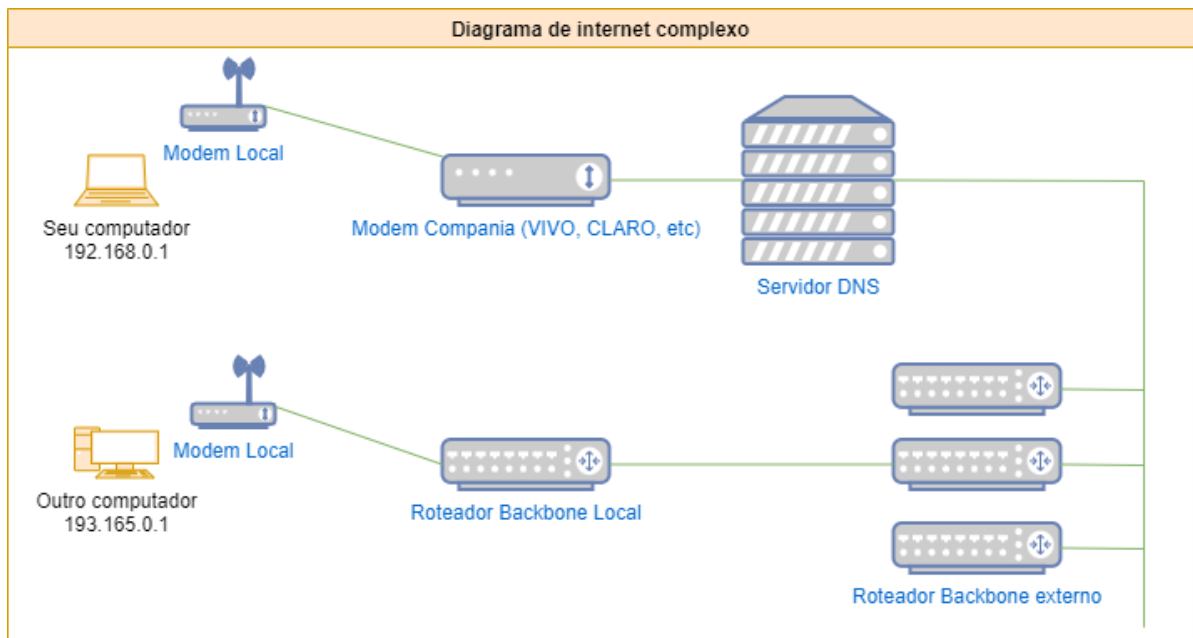


Para transitar informações deve se utilizar uma pilha de protocolos. Todo computador já possui pilhas de protocolos para se comunicar, estes recursos estão embutidos no sistema operacional (Windows, Unix, etc.). De forma geral o exemplo a seguir demonstra o caminho que uma informação percorre dentro do computador para outro. A informação é passada por camadas de protocolos aonde é empacotada, datada, codificada e enviada:



| Camada de protocolo | Comentários |
|--|---|
| Camada de protocolos de aplicativos | Protocolos específicos para aplicações como WWW, e-mail, FTP, etc. |
| Camada do Protocolo de Controle de Transmissão | O TCP direciona os pacotes para um aplicativo específico em um computador usando um número de porta. |
| Camada de protocolo da Internet | O IP direciona os pacotes para um computador específico usando um endereço IP. |
| Camada de Hardware | Converte dados de pacotes binários em sinais de rede e vice-versa. (Por exemplo, placa de rede ethernet, modem para linhas telefônicas, etc.) |

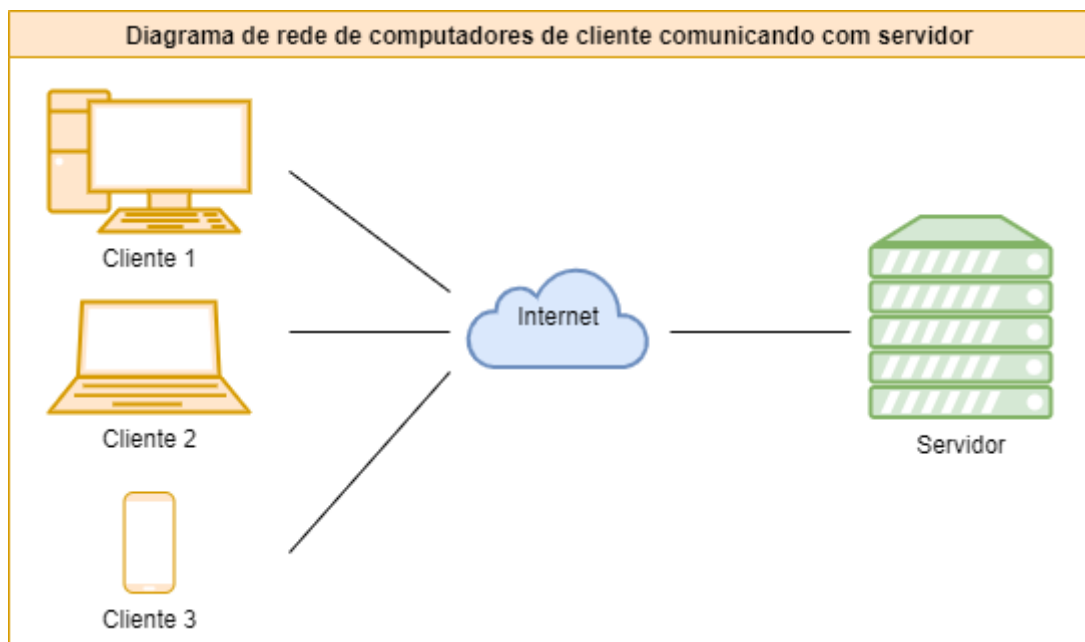
O caminho percorrido pela informação enviada por um computador é bem mais complexo do que apresentado acima. Os dados passam pelo modem quem se encontra dentro de nossas residências. Através de uma rede publica é redirecionado para um servidor de domínio, responsável por validar o endereçamento existente. Caso o endereço for encontrado o tráfego é feito entre roteadores Backbone até chegar no roteador de destino e entregue para o endereço correto. O seguinte exemplo pode ser visto na imagem abaixo.



Se você estiver usando o Microsoft Windows ou uma versão do Unix e tiver uma conexão com a Internet, aqui está outro programa de Internet útil. Este é chamado **tracert** e mostra o caminho que seus pacotes estão tomando para um determinado destino da Internet. Assim como o ping, você deve usar o tracert em um prompt de comando. No Windows, use **tracert** www.yahoo.com. Em um prompt do Unix, digite **tracert** www.yahoo.com. Assim como o ping, você também pode inserir endereços IP em vez de nomes de domínio. O Traceroute imprimirá uma lista de todos os roteadores, computadores e quaisquer outras entidades da Internet pelas quais seus pacotes devem passar para chegar ao destino.

2. Cliente e Servidor

O modelo cliente-servidor é uma estrutura de aplicação distribuída que particiona tarefas ou cargas de trabalho entre os **provedores** de um recurso ou serviço, chamados servidores, e **solicitantes** de serviço, chamados clientes. Para melhor compreensão abaixo segue um exemplo simples de diagrama aonde demonstra este relacionamento cliente servidor:



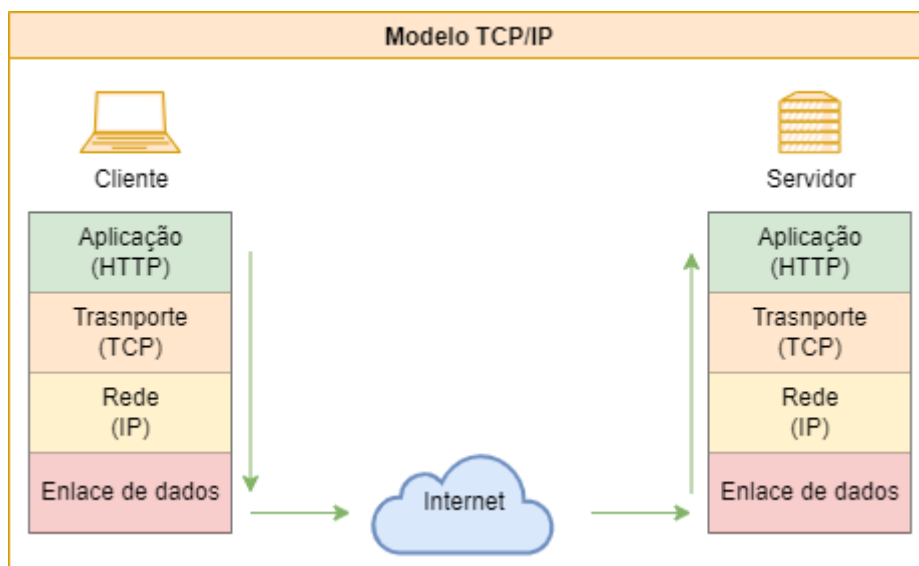
A característica "cliente-servidor" descreve a relação de programas cooperativos em um aplicativo. O componente servidor **fornece** uma função ou serviço para um ou vários clientes, que iniciam **solicitações** para tais serviços. Os servidores são classificados pelos serviços que prestam. Por exemplo, um servidor web serve páginas da web e um servidor de arquivos serve arquivos de computador. Um recurso compartilhado pode ser qualquer software e componentes eletrônicos do computador servidor, desde programas e dados até processadores e dispositivos de armazenamento.

Clientes e servidores trocam mensagens em um padrão de mensagens de solicitação-resposta. O **cliente** envia uma **solicitação** e o **servidor** retorna uma **resposta**. A linguagem e as regras de comunicação são definidas em um protocolo de comunicação.

Todos os protocolos operam na camada de aplicação. O protocolo da camada de aplicação define os padrões básicos do diálogo para o desenvolvimento dentro do curso iremos utilizar o protocolo HTTP. Para formalizar ainda mais a troca de dados, o servidor pode implementar uma interface de programação de aplicativos (API). A API é uma camada de abstração para acessar um serviço. Ao restringir a comunicação a um formato de conteúdo específico, facilita a análise. Ao abstrair o acesso, facilita a troca de dados entre plataformas.

3. TCP/IP

A comunicação entre computadores em uma rede são feitas por meio de termos e protocolos. Existem duas maneiras de ligar um cliente a um servidor é utilizando os protocolos TCP ou UDP. No caso o TCP, é orientado a conexão e será o que abordaremos durante o curso. O protocolo TCP/IP é considerado um sistema de 4 camadas e segue o esquema da imagem abaixo:



Camada de Aplicação

Essa camada inclui aplicativos ou processos que usam protocolos da camada de transporte para entregar os dados aos computadores de destino. a camada de aplicação também possui vários protocolos que os aplicativos usam para se comunicar com a segunda camada, a camada de transporte. Alguns dos protocolos de camada de aplicação populares são:

- HTTP (protocolo de transferência de hipertexto);
- FTP (protocolo de transferência de arquivos);
- SMTP (Protocolo de transferência de correio simples);
- SNMP (protocolo de gerenciamento de rede simples).

Camada de Transporte

Esta camada recebe dados da camada de aplicação acima dela. Existem muitos protocolos que funcionam nesta camada, mas os dois protocolos mais usados na camada de transporte são TCP e UDP. No caso do TCP ele é o responsável por dividir os dados (vindos da camada de aplicação) em pedaços de tamanho adequado e, em seguida, passa esses pedaços para a rede evitando a perda de pacotes. Já o UDP não possui medida extra para garantir que os dados enviados sejam de fato recebidos pelo cliente.

Camada de Rede

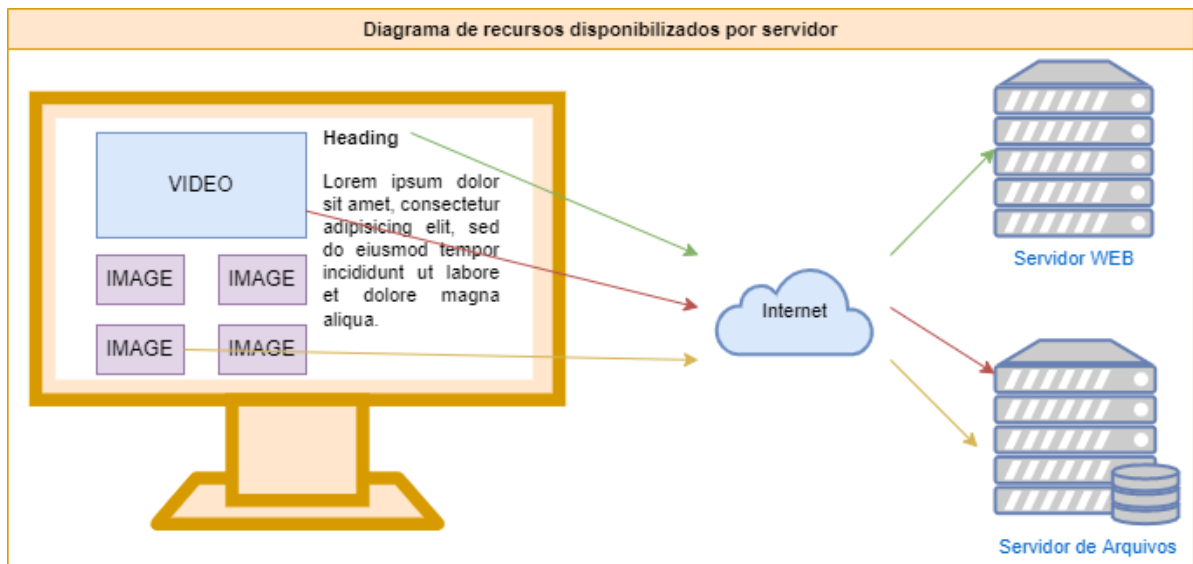
Essa camada também é conhecida como camada de Internet. O principal objetivo desta camada é organizar ou lidar com a movimentação de dados na rede. Por movimento de dados, geralmente queremos dizer roteamento de dados pela rede. O principal protocolo usado nesta camada é o IP. Enquanto ICMP (usado pelo popular comando 'ping') e IGMP também são usados nesta camada.

Camada de Enlace de Dados

Essa camada também é conhecida como camada de interface de rede. Essa camada normalmente consiste em drivers de dispositivo no sistema operacional e na placa de interface de rede conectada ao sistema. Tanto os drivers de dispositivo quanto a placa de interface de rede cuidam dos detalhes de comunicação com a mídia que está sendo usada para transferir os dados pela rede. Na maioria dos casos, essa mídia está na forma de cabos. Alguns dos famosos protocolos usados nesta camada incluem ARP (Protocolo de resolução de endereço), PPP (Protocolo ponto a ponto) etc.

4. HTTP

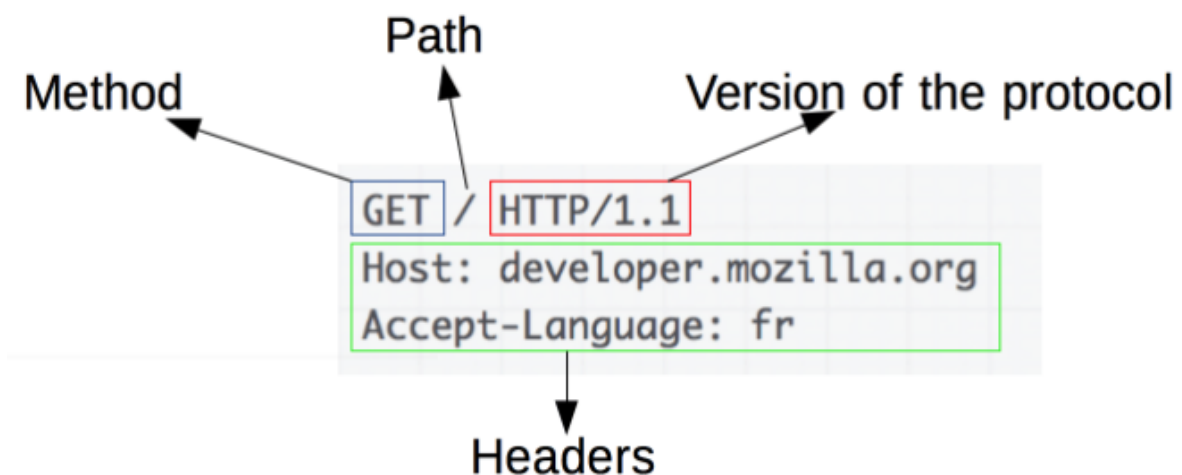
HTTP é um protocolo para buscar recursos como documentos HTML. É a base de qualquer troca de dados na Web e é um protocolo cliente-servidor, o que significa que as solicitações são iniciadas pelo destinatário, geralmente o navegador da Web. Um documento completo é reconstruído a partir dos diferentes sub-documentos obtidos, por exemplo, texto, descrição do layout, imagens, vídeos, scripts e muito mais.



Clientes e servidores se comunicam trocando mensagens individuais (em oposição a um fluxo de dados). As mensagens enviadas pelo cliente, geralmente um navegador da Web, são chamadas de **requisições** e as mensagens enviadas pelo servidor como **resposta** são chamadas de respostas .

Requisição

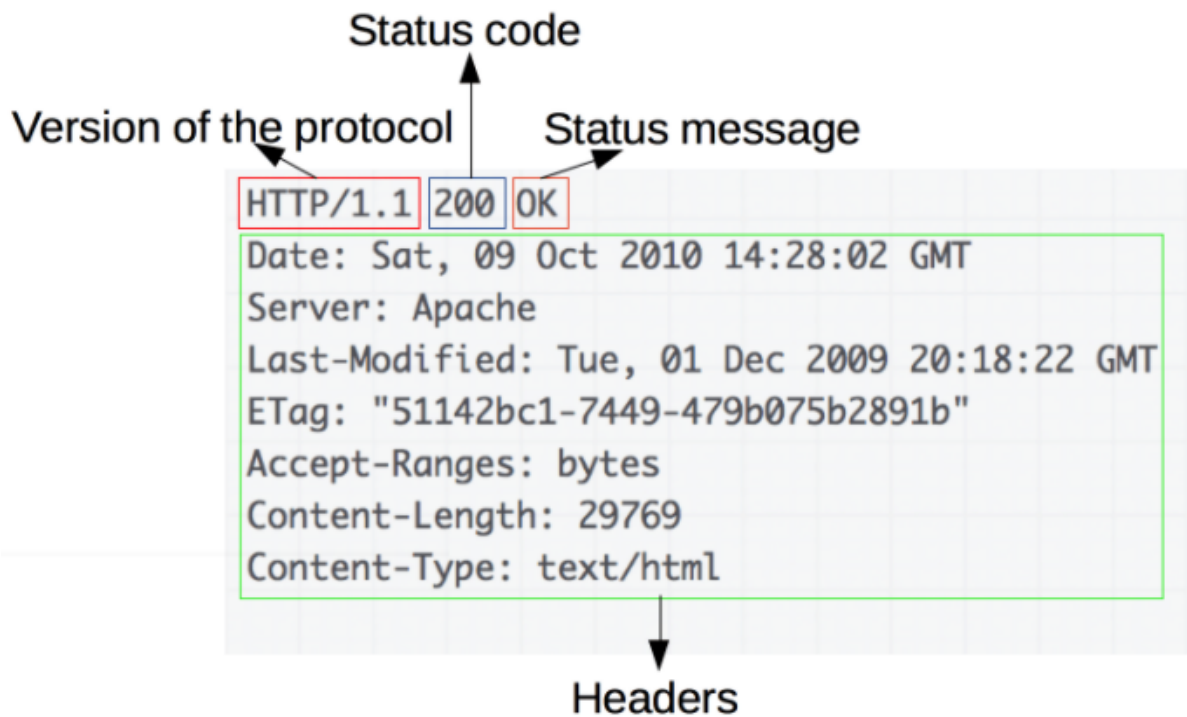
```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```



Resposta

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (here come the 29769 bytes of the requested web page)
```



5. Verbos (metodos)

O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como HTTP Verbs (Verbos HTTP). Cada um deles implementa uma semântica diferente, sendo os principais listados abaixo:

| VERBO | DESCRIÇÃO |
|---------|---|
| GET | O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados. |
| POST | O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor. |
| PUT | O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição. |
| DELETE | O método DELETE remove um recurso específico. |
| HEAD | O método HEAD solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta. |
| CONNECT | O método CONNECT estabelece um túnel para o servidor identificado pelo recurso de destino. |
| OPTIONS | O método OPTIONS é usado para descrever as opções de comunicação com o recurso de destino. |
| TRACE | O método TRACE executa um teste de chamada loop-back junto com o caminho para o recurso de destino. |
| PATCH | O método PATCH é utilizado para aplicar modificações parciais em um recurso. |

6. Status code

Os códigos de status das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída. As respostas são agrupadas em cinco classes:

- Respostas de informação (100-199)
- Respostas de sucesso (200-299)
- Redirecionamentos (300-399)
- Erros do cliente (400-499)
- Erros do servidor (500-599)

Para ser proativo leia a documentação do link abaixo:

[Link códigos de status de respostas HTTP](#)

[Link status cat](#)

[Link status dog](#)