Aula 01

História do HTTP

- Com o surgimento da World Wide Web, ou simplesmente WWW, houve a necessidade de distribuir informações pela Internet. Para que essa distribuição fosse possível foi necessário criar uma forma padronizada de comunicação entre os clientes e os servidores da web.
- E que fosse também entendida por todos os computadores conectados à internet.
- A partir desta necessidade que o protocolo HTTP foi criado no ano de 1990.

O que é HTTP

- HTTP significa Hypertext Transfer Protocol.
- É um protocolo que permite a obtenção de recursos na WEB.
- Responsável por especificar como será a comunicação entre um cliente e um servidor.
- Esse sistema é a base da comunicação que existe em toda a Internet.

Funcionamento do HTTP

O protocolo utiliza uma estrutura de cliente-servidor, ou seja:

- Cliente faz uma solicitação.
- Uma mensagem HTTP é enviada para o servidor.
- O servidor recebe a mensagem e processa a informação.
- Depois que a informação é processada, o servidor retorna uma resposta ao cliente.

Funcionamento do HTTP

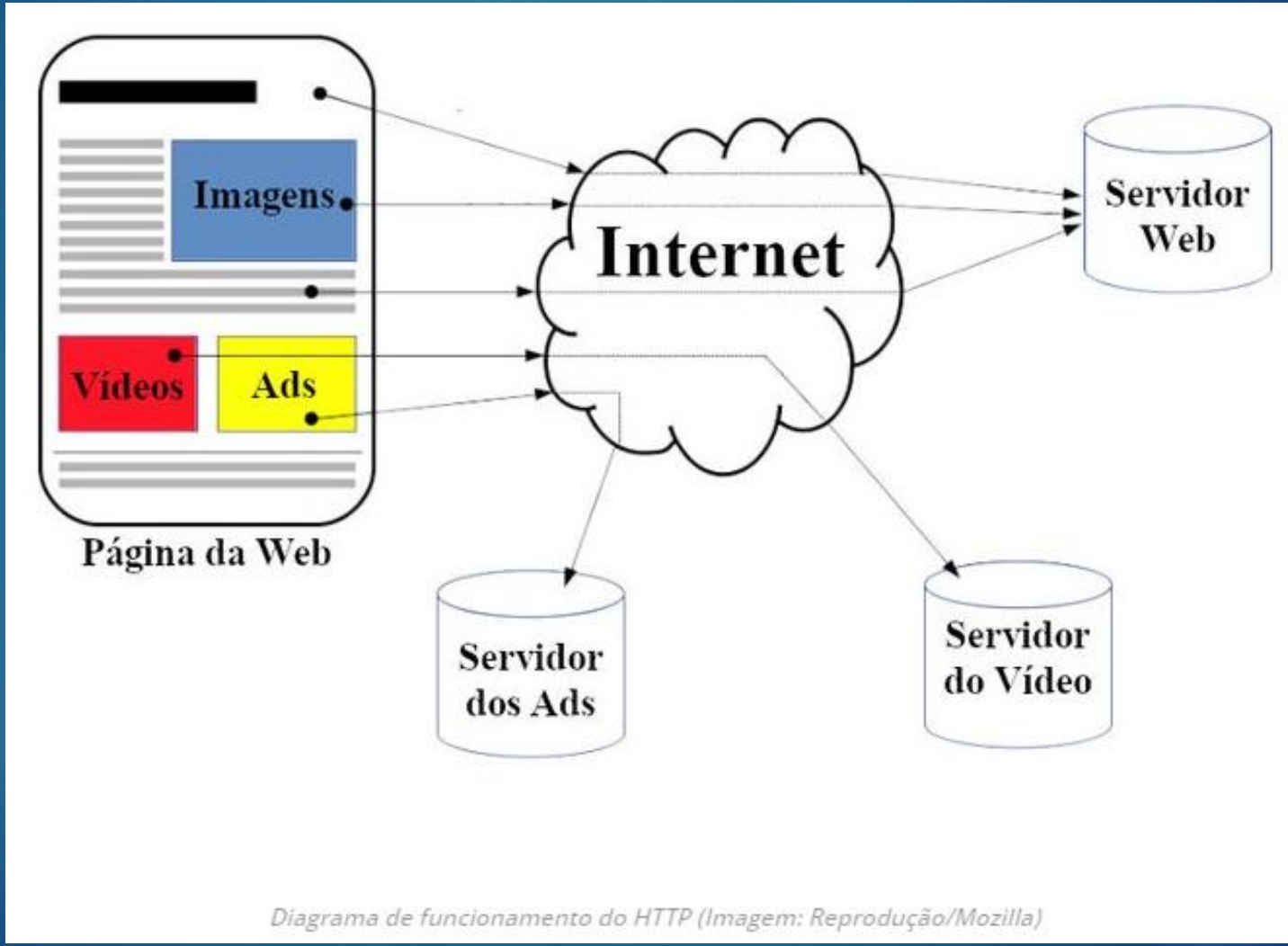


Diagrama de funcionamento do HTTP (Imagem: Reprodução/Mozilla)

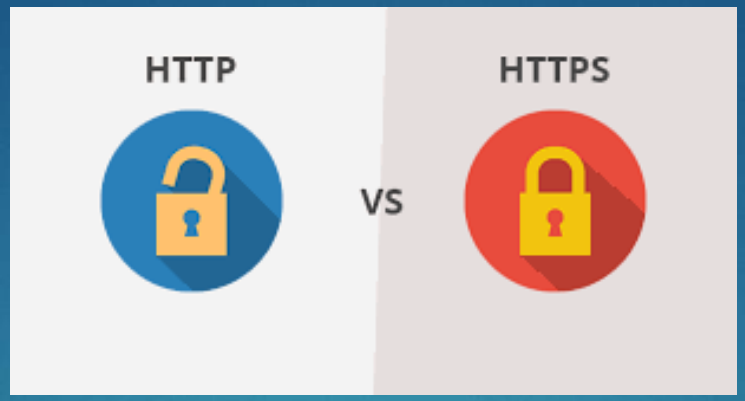
O QUE É HTTPS

O que é HTTPS

- O Hypertext Transfer Protocol Secure, ou somente HTTPS, é um protocolo HTTP utilizado em conjunto com o Certificado SSL.
- A sigla SSL significa Secure Sockets Layer.
- Essa tecnologia permite a criptografia dos dados que trafegam entre o cliente e o servidor certificado, além de ser o método mais utilizado para manter a segurança de sites e serviços na web.

Vantagens de Utilizar o HTTPS

- Privacidade de dados
  - Troca de informações de forma segura, utilizando criptografia.
- Menor taxa de rejeição
  - Se o site é apontado como seguro, é natural que os usuários sintam mais confiança e continuem navegando nas páginas.
- Aplicação melhor rankeada
  - Melhores estratégias de SEO.
  - Visibilidade, maior tráfego, mais conversões.



HTTP

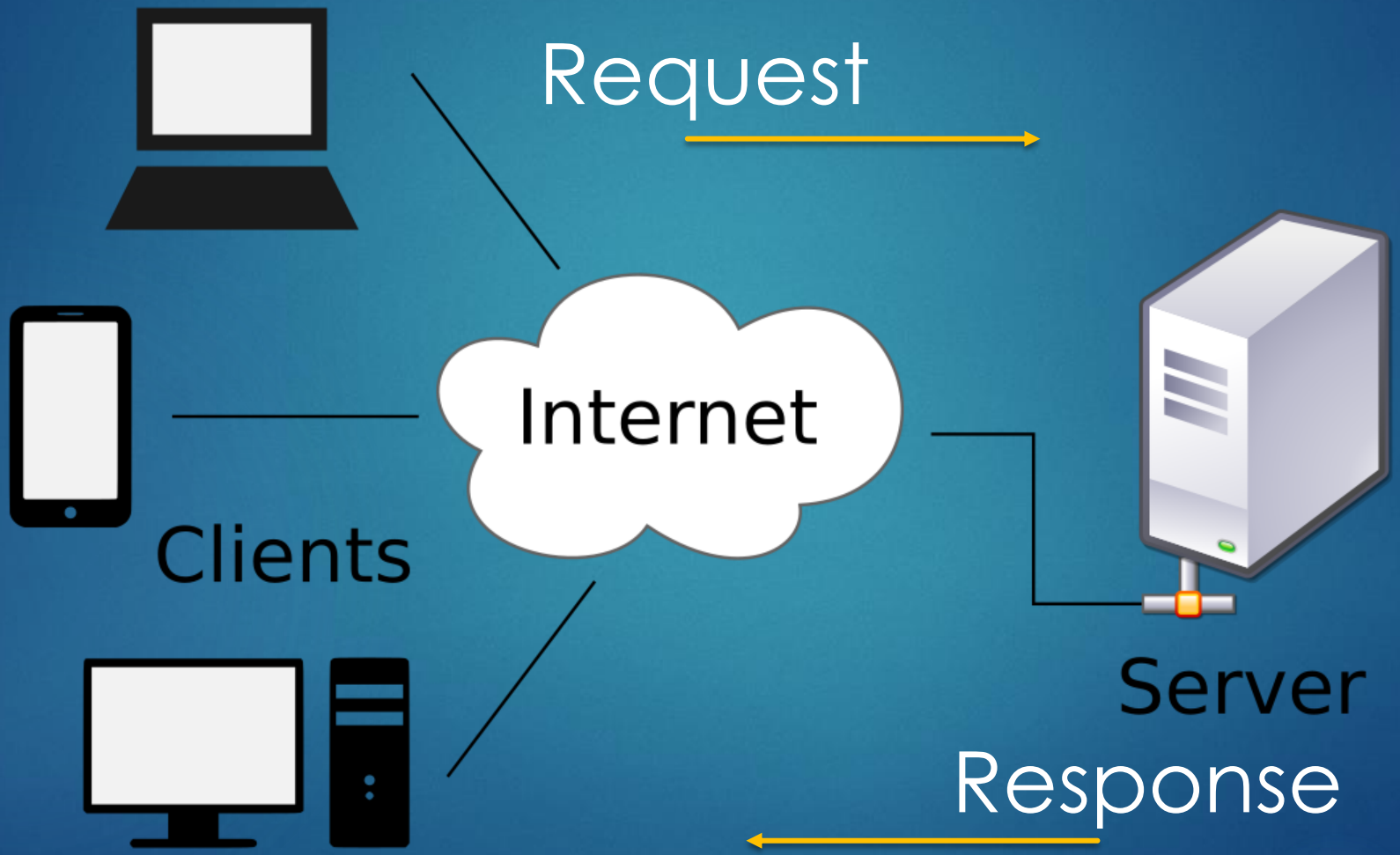
- Começa com http://
- Utiliza a porta 80
- Não é seguro
- Trafega dados em formato de hipertexto

HTTPS

- Começa com https://
- Utiliza a porta 443
- Seguro, utiliza a tecnologia SSL
- Trafega dados em formato criptografado

MENSAGEMS HTTP

Mensagens HTTP



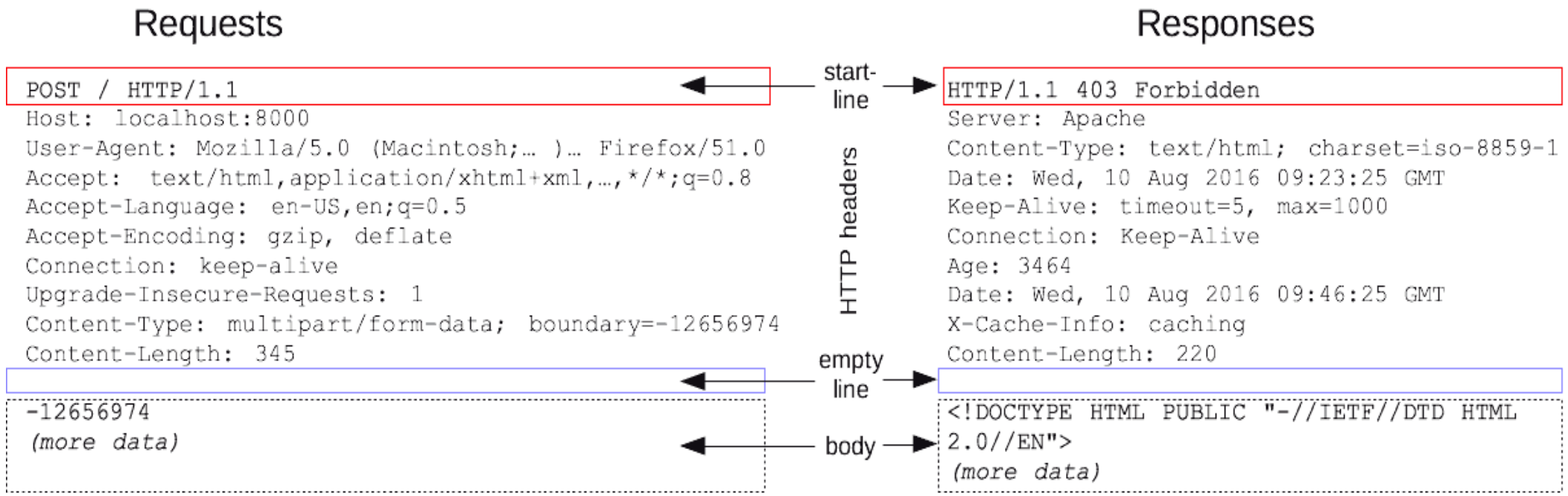
Vantagens de Utilizar o HTTPS

- As informações são trocadas entre servidor e cliente através mensagens HTTP.
- Essa comunicação ocorre através de requisições e respostas.
- As requisições e respostas HTTP possuem uma estrutura semelhante e são compostas de:

1. Linha inicial (start line)
2. Cabeçalhos HTTP (headers)
3. Linha em branco (empty line)
4. Corpo contendo as informações trafegadas (body)

Mensagens HTTP

Request → Response



REQUISIÇÕES HTTP

Requisições HTTP

- A requisição é um pedido que um cliente realiza para um servidor. Esse pedido contém uma série de dados que são utilizados para descrever o que o cliente precisa.
- Basicamente são mensagens HTTP enviadas pelo cliente para iniciar uma determinada ação no servidor.
- As requisições possuem a seguinte estrutura:
  - Linha inicial
  - Cabeçalhos
  - Corpo da requisição

Requisições HTTP

Suas linhas iniciais contêm três elementos:

- Método HTTP
  - Descreve qual ação a ser executada.
- Path
  - O alvo da requisição, normalmente um URL.
- Versão do protocolo
  - A versão HTTP, que define a estrutura do restante da mensagem, atuando como um indicador da versão esperada para uso na resposta.

```
POST / HTTP/1.1
GET /background.png HTTP/1.0
```

Requisições HTTP

Os cabeçalhos contém informações adicionais para os servidores.

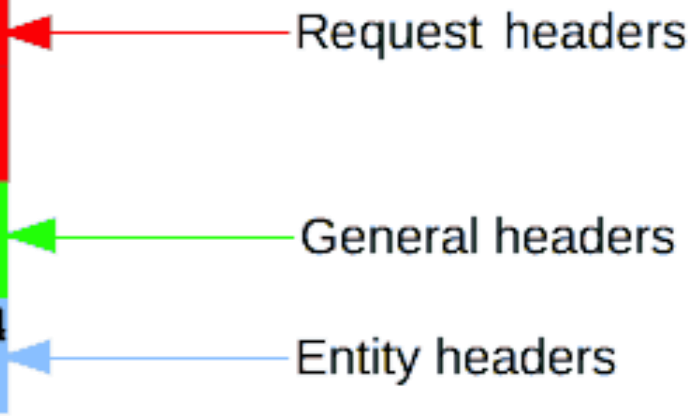
Existem diversos cabeçalhos de requisição disponíveis e eles podem ser divididos nos seguintes grupos:

- Cabeçalhos gerais
  - Podem ser usados tanto em solicitações quanto em respostas, porém sem relação com os dados eventualmente transmitidos no corpo da mensagem.
- Cabeçalhos de requisição
  - Contém mais informação sobre o recurso a ser obtido ou sobre o próprio cliente.
- Cabeçalhos de entidade
  - Contém mais informação sobre o conteúdo da entidade, como o tamanho do conteúdo (Text/Html , JSON, etc..).

Requisições HTTP

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```



Requisições HTTP

- A última parte de uma requisição HTTP é o corpo, mas nem todas as requisições possuem um.
- As requisições que pegam recursos, normalmente não possuem um corpo e já aquelas que precisam atualizar ou cadastrar determinado recurso, necessitam enviar informações para o servidor, nesse caso é necessário o uso do body.

RESPOSTAS HTTP

Respostas HTTP

Aprendemos que o cliente envia uma requisição ao servidor, que por sua vez recebe essas informações e retorna uma resposta ao cliente.

Essa resposta pode conter os dados que o cliente espera receber ou uma resposta informando que alguma coisa deu errado.

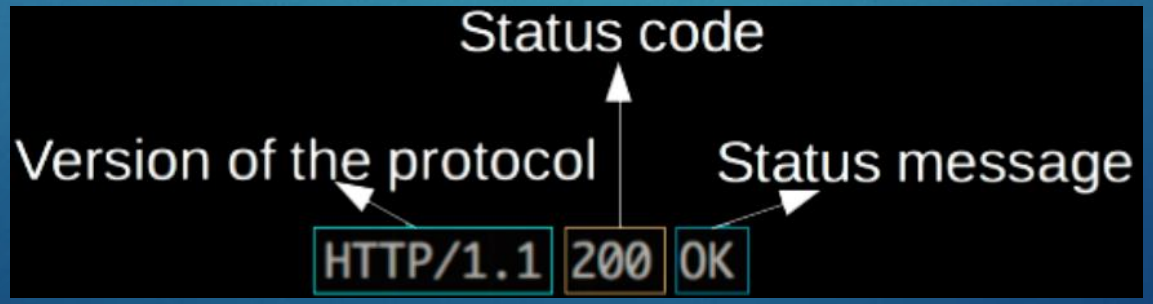
As respostas HTTP possuem a seguinte estrutura:

- Linha de status
- Cabeçalhos
- Corpo da resposta

Respostas HTTP

A linha inicial de uma resposta HTTP, chamada de linha de status, contém as seguintes informações:

- A versão do protocolo, normalmente HTTP/1.1
- Um código de status que indica se a requisição foi processada com sucesso ou se houve alguma falha.
  - Os códigos mais comuns são 200, 400, 404 e 500.
- Um texto de status que descreve brevemente o código de status.



VERBOS HTTP

Respostas HTTP

Para cada resposta enviada pelo servidor, existe um código responsável por indicar o que aconteceu. Esses códigos estão entre 100 e 500, sendo que cada centena indica uma categoria:

- 1xx – Informativos - indica que o servidor ainda está processando a solicitação.
- 2xx – Indicativos de sucesso - indica que a ação solicitada pelo cliente foi recebida, compreendida, aceita e processada com êxito.
- 3xx – Redirecionamentos - sinaliza que o agente de usuário precisa realizar ações adicionais para que a solicitação seja atendida
- 4xx – Erros do cliente na hora de fazer a solicitação - é destinada para os casos onde um erro foi cometido pelo cliente. O problema pode estar no navegador ou na própria solicitação.
- 5xx – Erros no lado do servidor - indicam uma solicitação com falha devido a um erro do servidor

O que é HTTP Error ?

- Quando você visita um site, o seu navegador faz uma solicitação ao servidor, e o servidor responde o navegador com um código, os códigos HTTP.
- Error HTTP são os códigos HTTP que referem-se a erros de cliente e servidor, respectivamente, e impedem o carregamento de um site.
- Fica fácil deduzir que os códigos HTTP Error são sempre começados pelos números 4 ou 5

Quais os principais códigos de erro HTTP?

Os principais códigos de erros HTTP que você deve ter em mente são os erros 403, 404, 500 e 503

- Erro 403
  - **Forbidden é o erro "Proibido"**. Isso significa que o servidor entendeu a solicitação do navegador mas se recusa a fazê-lo, pois o navegador não possui autorização para isso.
- Erro 404
  - Quando você digita uma URL e recebe a mensagem ERROR 404 – PAGE NOT FOUND quer dizer que esta URL não te levou a lugar nenhum. Os motivos podem ser: a página não existe mais, a URL deste site mudou ou você digitou a URL errada.

Quais os principais códigos de erro HTTP?

Os principais códigos de erros HTTP que você deve ter em mente são os erros 403, 404, 500 e 503

- Erro 500
- Este erro não é tão comum de acontecer mas uma hora ou outra ele aparece!
- Erro 500 significa que algum script ou solicitação não foi compreendida – o que nem sempre indica um problema com o servidor.
- Os motivos podem ser arquivos .htaccess corrompidos, permissões de arquivo incorreto, tempo limite de *script*, versão do PHP incompatível ou atualizações de um **site WordPress**.

Quais os principais códigos de erro HTTP?

- Erro 503
- Este é um erro que pode confundir os webmasters. Isso porque o status ERRO 503 significa serviço temporariamente indisponível. Não é muito claro,
  - Plugins e temas bugados;
  - Um mau comportamento de um script PHP customizado;
  - Servidores com recursos insuficientes;
  - Problemas de servidor;
  - Ataques maliciosos, como o infame Ataque Distribuído por Negação de Serviço (DDoS).

https://www.hostinger.com.br/tutoriais/o-que-e-http-error-e-principais-codigos-http

para maiores informações

Verbos HTTP

- GET
  - O método GET é utilizado quando existe a necessidade de se obter um recurso, então quando você quer "pegar" dados você utiliza o GET.
  - GET não deve modificar dados, somente trazer os dados.
  - Por exemplo, considerando a URL www.dominio.com/rest/notas/, se enviarmos para ela uma requisição HTTP utilizando o verbo GET, provavelmente obteremos como resultado uma listagem de registros (notas, nesse caso)
  - Da mesma forma, a URL www.dominio.com/rest/notas/1, essa URL provavelmente deveria nos retornar o registro de ID 1 (nesse caso, a nota de ID = 1).

Verbos HTTP

- POST
  - O método POST, estaremos tentando adicionar um novo registro, cujos dados serão enviados no corpo da requisição
  - Por exemplo, considerando a URL www.dominio.com/rest/notas/

Verbos HTTP

- PUT
  - Normalmente usado com parâmetro;
  - Use para Editar/Alterar
- DELETE
  - Usado para remover o recurso

Verbos HTTP

- Diferença entre PUT e POST
  - Encontramos na **literatura** indicações de que apenas três verbos são suficientes para um CRUD completo: GET, DELETE e PUT – sendo o PUT utilizado para criar ou editar um recurso.
  - PUT é usado para criar ou editar um recurso, enquanto POST pode ser utilizado para qualquer coisa, cabendo ao back-end a definição dessa semântica.
  - O mundo não-acadêmico, no entanto, adotou por convenção o uso do POST para incluir e do PUT para alterar – e em situações de programação mais elegantes, também o uso de PATCH para editar parcialmente.

Dúvidas??

PROF. JUNIO FIGUEIREDO