



Tecnologias Web

Aula 01 – O caminho por trás de uma requisição

GUILHERME FEULO

Sumário

- Entender o que é o modelo cliente-servidor.
- Qual o caminho de uma requisição na web.
- Entender os componentes da arquitetura cliente-servidor.
- Detalhar o funcionamento do HTTP.

Modelo Cliente - Servidor

- Principal padrão arquitetural adotado na Internet
- Processamento distribuído entre dois elementos
 - Cliente: requisita serviços.
 - Servidor: realiza os serviços pedidos pelos clientes.
- Exige comunicação entre os dois elementos
 - Necessidade de uma rede entre os computadores (internet).
 - Necessidade de um protocolo de comunicação (HTTP).
 - Necessidade de um mecanismo de localização (URL).

Internet - Intranet - Extranet

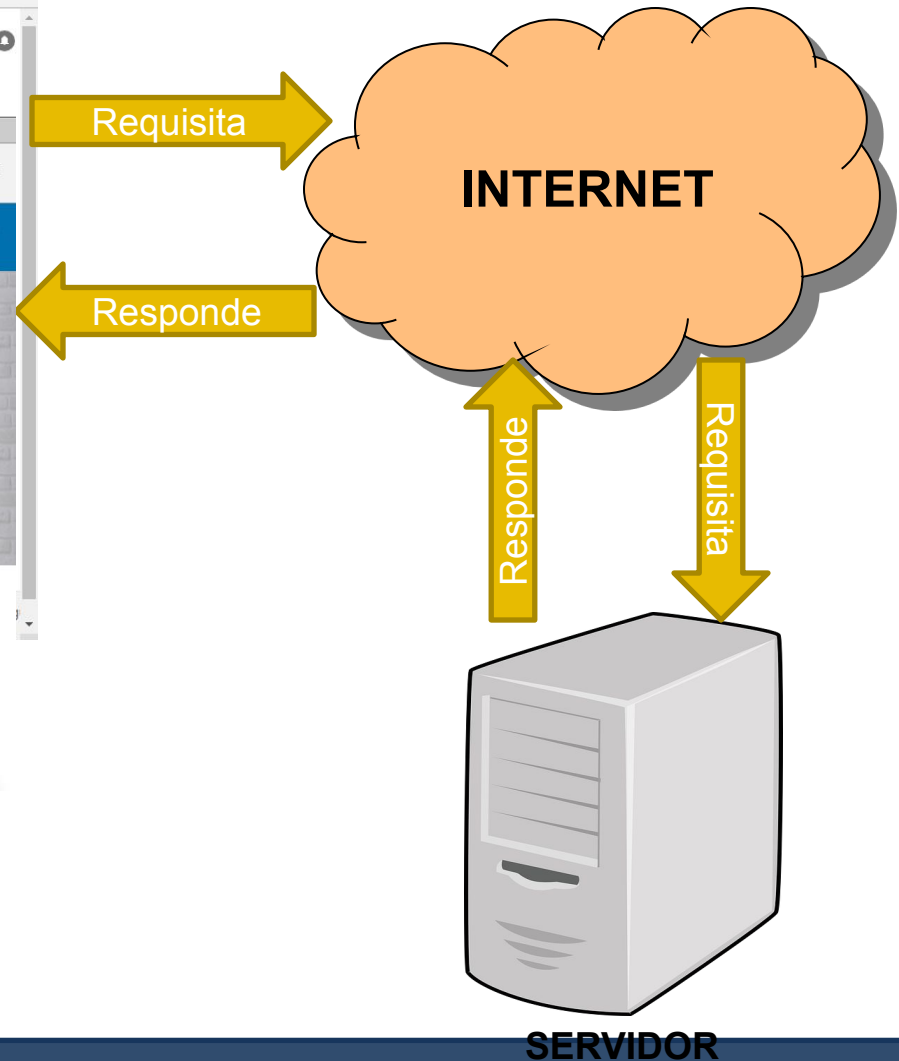
- **INTERNET:** Coleção de redes que utilizam protocolos abertos para se comunicar e formar a *world wide web*
- **INTRANET:** Rede de comunicação que está restrita a um determinado domínio, por exemplo, uma rede interna de empresa.
- **EXTRANET:** Uma extensão da *intranet*, que utiliza protocolos abertos para usar a *internet* como acesso para alguns usuários restritos a *intranet*.

<https://docs.microsoft.com/en-us/style-guide/a-z-word-list-term-collections/i/internet-intranet-extranet>

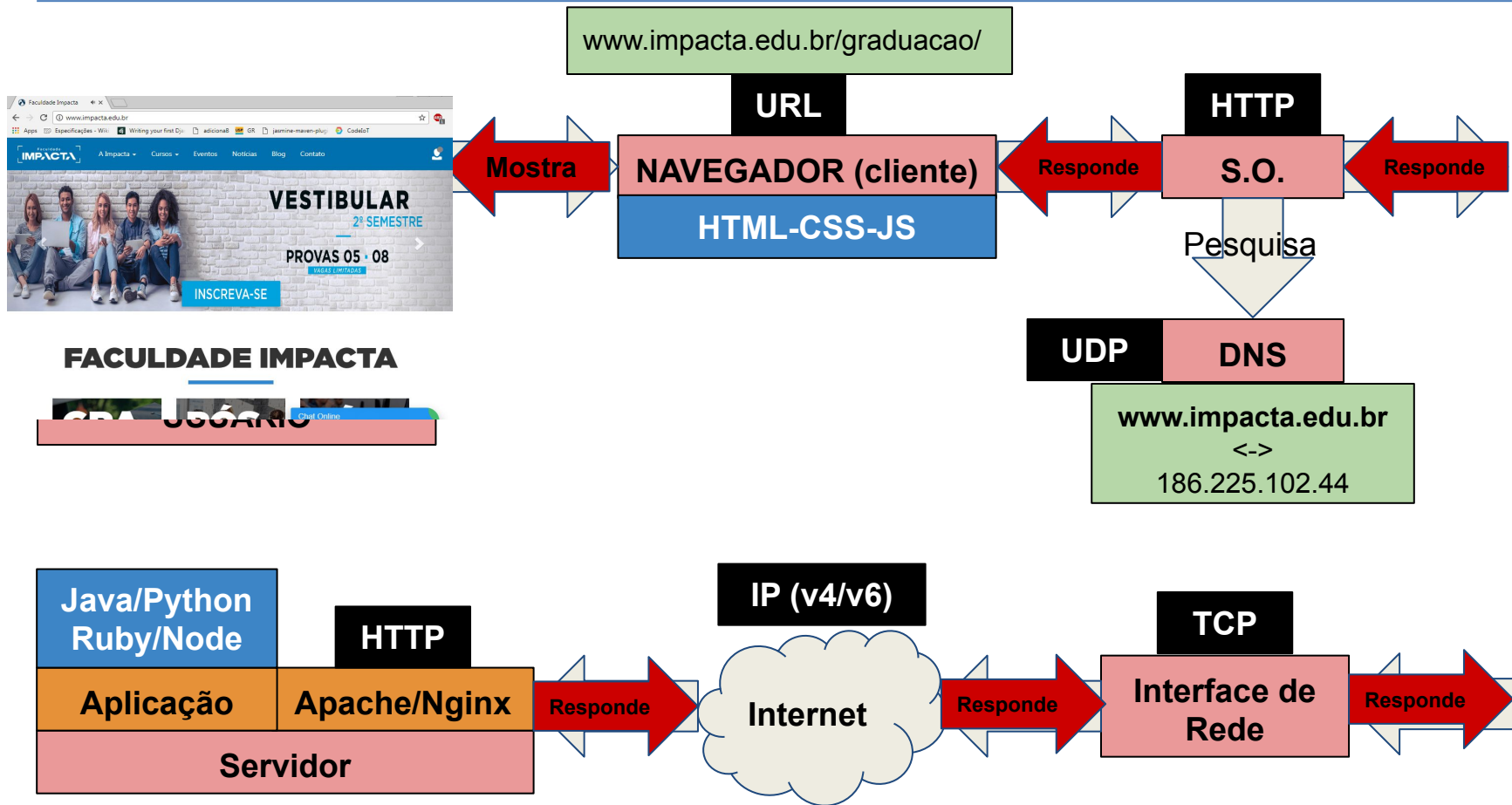
Comunicação cliente - servidor



FACULDADE IMPACTA



Caminho de uma Requisição



Componentes Cliente - Servidor

- Nessa estrutura temos os seguintes componentes
 - Navegador de Internet (Browser): Cliente, aquele que requisita algum recurso da internet.
 - URL (<http://www.impacta.edu.br/>): endereço padronizado de recursos na Web (veremos adiante).
 - Internet: uma sequência de redes de computadores toda interconectada.
 - Servidor: computador (ou nuvem de computadores) que responde uma requisição feita por um cliente.
 - Protocolos: Maneiras padronizadas de transmitir informações entre dois pontos em uma rede

Componentes Cliente - Servidor

- O que vamos estudar na disciplina:
 - Tecnologias que rodam no Navegador: HTML, CSS e JavaScript
 - Estrutura básica de URL's.
 - Programação para servidor (Python).
 - Estrutura básica do HTTP

Componentes - URI VS URL

- Falamos de URI e de URL, mas qual a diferença?
 - URI: Uniform Resource Identifier
 - URL: Uniform Resource Locator
- Além do nome, qual a diferença??
 - Primeiro: URL é um tipo de URI (existe o URN também).
 - Segundo: URL é uma URI que possui uma “maneira” de se chegar em um recurso, um protocolo de como chegar.

Componentes - URL

`http://www.impacta.edu.br/professor/default.php`

- A URL (Uniform Resource Locator) é um endereço para encontrar um determinado recurso na rede. Esse recurso pode ser qualquer coisa, como uma página php, imagem, vídeo, etc.
- A rede em questão pode ser a internet, mas pode ser um recurso local da sua própria rede.
- A URL é dividida em algumas partes, cada uma com a sua função.

Componentes - URL

<http://www.impacta.edu.br/professor/default.php>

- **http** – Chamado também de esquema, informa qual protocolo de comunicação será usado na requisição.
- **impacta.edu.br** – Indica o domínio da rede que hospeda o recurso desejada (ver relação entre domínio e IP: https://pt.wikipedia.org/wiki/Endere%C3%A7o_IP).
- **professor** – caminho interno até o recurso (chamado de path).
- **default.php** – recurso requisitado.

Componentes - URL - Protocolo

<http://www.impacta.edu.br/professor/default.php>

- Esquema ou Protocolo:
 - Protocolos na computação são convenções de como a informação deve trafegar da sua origem até o destino.
 - Na web, o protocolo mais utilizado é o HTTP – HyperText Transfer Protocol.
 - Para entender o protocolo, o computador que vai receber as requisições é chamado de servidor HTTP.

Componentes - URL - Domínio

`http://www.impacta.edu.br/professor/default.php`

Domínio:

- É um nome ou identificador representando algum componente dentro de uma rede.
- É usado para relacionar os endereços de IP de uma rede com um nome mais fácil de entender para humanos.
- Obedece às regras do DNS – Domain Name System

Componentes - URL - Diretório

`http://www.impacta.edu.br/professor/default.php`

Diretório (Path):

- Caminhos internos para encontrar o recurso desejado.
- Em sites clássicos isso significa pastas dentro de um computador, por exemplo, se o site da Impacta está hospedada na pasta **/var/www/impacta/** dentro no Linux, então o script acima está no caminho **/var/www/impacta/professor/default.php**
- O diretório pode significar também caminhos internos na aplicação (mapeamentos específicos), que não correspondem a pastas no servidor. Por exemplo, é comum em micro serviços termos requisições como as que seguem:
 - <http://api.impacta.edu.br/aluno/listarAlunos>
 - <http://api.impacta.edu.br/professor/gerarListaChamadas>

Componentes - URL - Recurso

<http://www.impacta.edu.br/professor/default.php>

Recurso requisitado:

- Identifica algum recurso no servidor.
- Pode ser qualquer tipo de arquivo: imagem, vídeo, música, documentos, scripts, etc.
- Em geral identificado pelo nome e extensão do arquivo a se obter (**default.php**). Para arquivos **index.html** o nome pode ser omitido (www.impacta.edu.br).

Componentes - Domínios

- Cada país tem uma entidade responsável por atribuir endereços de IP e fazer sua associação com um nome.
- No Brasil, o sistema de domínios é gerenciado pela NIC.br (Núcleo de Informação e Coordenação do Ponto BR).
- Mundialmente, foram estabelecidas convenções para classificar o conteúdo das páginas:
 - .com: comercial
 - .edu: organização educacional
 - .gov: entidade governamental
 - .int: organização internacional
 - .mil: instituição militar
 - .net: operadora de rede
 - .org: outros tipos de organizações
 - Lista de domínios .br: <http://registro.br/dominio/categoria.html>

Componentes - Domínios

- Foi ainda estabelecida uma terminologia para indicar endereços de cada país:
 - .br : Brasil
 - .de : Alemanha (Deutschland)
 - .pt : Portugal
 - .fr : França
 - .tv: Tuvalu
 - .ht: Hait

Componentes - Servidor de Hospedagem

- Consiste em um computador visível e acessível pela internet, que contenha todos os recursos do seu site ou aplicação.
- Em geral contratados por serviços especializados:
 - Alguns gratuitos:
 - <http://www.host.sk/>
 - <http://www.tripod.lycos.es/>
 - <http://github.com> (veremos como usar!)
 - Outros pagos:
 - <http://www.uolhost.com.br/>
 - <http://www.locaweb.com.br>
- Toda a organização de pasta e arquivos podem ser definidas no servidor, bem como os recursos disponíveis

Componentes - Tecnologias da Web

- Para a web, temos basicamente dois atores: **Cliente e Servidor**. Cada um destes possui suas próprias tecnologias para operar:

CLIENTE

- HTML(5)
- CSS(3)
- JavaScript
- ~~Componentes~~

SERVIDOR

- Servidores de Aplicação/HTTP (ou similar)
- Linguagens de Programação - Aplicação
- Bancos de Dados

Componentes - Tecnologias da Web - HTML

HyperText Markup Language

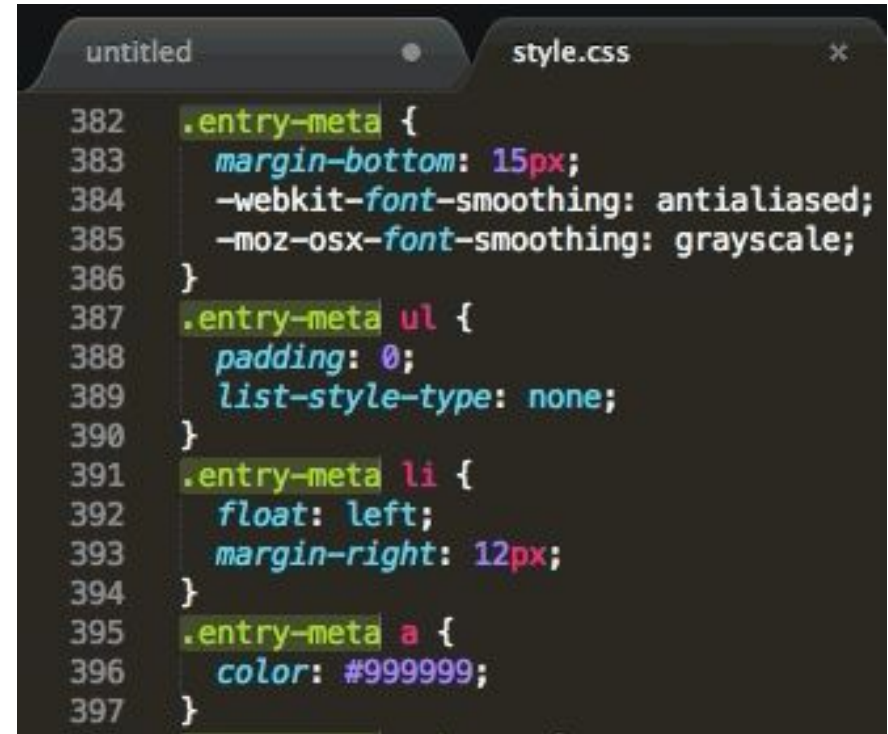
- Linguagem utilizada para representar o conteúdo e o formato visual de uma página no navegador.
- HTML é uma tagged language: suas tags definem a apresentação do conteúdo no navegador.

```
test.html
1  <div class="container">
2    <div class="header">
3      <header>
4        <h1>My Website</h1>
5      </header>
6    </div>
7    <div class="main"></div>
8    <div class="footer">
9      <footer class="container">
10        <div class="header">
11          <header>
12            <h1>My WEbsite</h1>
13          </header>
14        </div>
15        <div class="main"></div>
16        <div class="footer">
17          <footer></footer>
18        </div>
19      </footer>
20    </div>
21  </div>
```

Componentes - Tecnologias da Web - CSS

Folhas de Estilo em Cascata (Cascading Style Sheets)

- Padrão de formatação para páginas web que permite ao desenvolvedor ir além das limitações do HTML
- Oferecem maior flexibilidade visual na programação de páginas HTML, gerando efeitos que somente poderiam ser conseguidos com imagens
- Redução do tamanho das páginas HTML, com a redução do número de imagens necessárias no visual



```

382 .entry-meta {
383     margin-bottom: 15px;
384     -webkit-font-smoothing: antialiased;
385     -moz-osx-font-smoothing: grayscale;
386 }
387 .entry-meta ul {
388     padding: 0;
389     list-style-type: none;
390 }
391 .entry-meta li {
392     float: left;
393     margin-right: 12px;
394 }
395 .entry-meta a {
396     color: #999999;
397 }
    
```

Componentes - Tecnologias da Web - JS

Scripts de Cliente – ECMAScript e JavaScript

- Um script é uma sequência de comandos descritos em uma linguagem de programação para executar alguma ação.
- Os scripts de cliente são utilizados para dar “vida” ao HTML, ou seja, colocar comportamentos dentro das páginas (ex: ao clicar, abrir janela ou mostrar uma área que não existia).
- Atualmente o ECMAScript é uma especificação e JavaScript é uma das implementações possíveis desses Scripts.

Componentes - Tecnologias da Web - JS

```
var React = require("react");
var _ = require("underscore");

var PizzaButton = React.createClass({
  shoutName: function (name) {
    return name.toUpperCase();
  },
  getFirstPizza: function (pizzas) {
    return _.first(pizzas);
  },
  render: function () {
    return(
      <div>
        <button>
          Say hi to {this.shoutName(this.props.name)}
        </button>

        The best pizza is {this.getFirstPizza(this.props.pizzas)}
      </div>
    );
  }
});

module.exports = PizzaButton;
```

Componentes - Tecnologias da Web - Outros

Componentes de Cliente

- Componentes de cliente podem aprimorar a apresentação de uma página HTML (novos controles de edição).
- Com componentes de cliente, uma página não fica limitada aos controles definidos pela linguagem HTML.
- Ex: Flash e Java Applets
- A utilização destes componentes está cada vez mais próxima de acabar (Adobe anunciou o fim do Flash para 2020).

Componentes - Tecnologias da Web - Servidores HTTP

Servidores Web

- Para que o computador que receberá a requisição seja capaz de interpretá-la, ele precisa entender o protocolo HTTP.
- Para entender esse protocolo, é necessário um software conhecido como servidor HTTP (ou web server).



Componentes - Tecnologias da Web - Aplicação (Programação)

Programação no Servidor - Linguagens

- Para executar a lógica associada da sua aplicação, é necessário uma linguagem de programação que o seu servidor (computador) possa executar.
- Diversos tipos (compilado x interpretado), diversos objetivos (orientação a objeto, estrutural, funcional, aspectos), diversas preferências.



Componentes - Tecnologias da Web - Banco de Dados

Sistema Gerenciador de Banco de Dados

- Software utilizado para gerenciar bases de dados.
- Cuida do acesso, manutenção e persistência de dados nas suas bases. Garante o acesso a elas através de uma série de drivers e de uma linguagem específica de consultas e alterações (SQL)



Componentes - Protocolo HTTP

- É um protocolo de troca de informações na internet.
- Versão atual é a 1.1 (RFC 2616 em 1999), mas a versão HTTP2 já é uma realidade em muitos provedores e servidores.
- Utilizado sobretudo para navegação na internet, mas também por integrações de sistemas via web (REST).

Componentes - Protocolo HTTP

- É composto de requisições e respostas entre um cliente (um navegador na maioria dos casos) e um servidor.
- Uma requisição é um pedido de um determinado recurso. Esse pedido vai com a sua especificação em termos de cabeçalhos de requisição (request headers) e com um identificador (URL) do recurso.
- Para que o servidor saiba o que deve fazer com essa requisição, o HTTP manda também um verbo, indicando qual ação executada pelo servidor.

Componentes - Protocolo HTTP

- É ***stateless*** (sem estado): uma requisição tem o ciclo de vida baseada na resposta. Após o retorno do servidor todas as informações sobre a requisição são perdidas.
- Para persistir informações entre requisições, precisamos usar recursos específicos:
 - **Cookies:** Conjunto de pares chave-valor em texto que ficam armazenados no cliente e são enviados em cada requisição de um mesmo domínio.
 - **Sessão:** Objeto em memória guardando dados específicos sobre a origem da requisição (dentro do servidor)

Componentes - Protocolo HTTP

- Verbos ou Métodos HTTP indicam o que o servidor deve fazer ao receber a requisição, são eles:
 - **GET**: Usado para obter um recurso qualquer do servidor.
 - **POST**: Envia dados para serem processados pelo servidor, em geral para criar ou alterar um recurso.
 - **DELETE**: Remove um determinado recurso do servidor.
 - **PUT**: Atualiza todas as informações de um recurso no servidor.
 - **PATCH**: Atualiza parte das informações de um recurso no servidor.
 - Outros:
<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

Componentes - Protocolo HTTP

- Após a requisição, o servidor processa toda a informação necessária e formata a resposta para enviar ao cliente.
Essa resposta é formada por:
 - cabeçalhos de resposta (response headers): metadados da resposta.
 - status: código informando sobre a situação da resposta (se deu ou não certo).
 - Corpo da resposta: o corpo contém informações disponibilizadas ao cliente, podendo ser o recurso pedido ou apenas uma confirmação de execução.

Componentes - Protocolo HTTP

- O código de status é um número que varia de 100 a 600, cada um associado com um tipo de situação que pode ser encontrado durante a navegação na web.
- Alguns dos mais comuns:
 - 200 (Ok): Tudo deu certo.
 - 403 (Forbidden): Você não possui permissão para este recurso (login).
 - 404 (Not Found): Recurso não foi encontrado.
 - 405 (Method not Allowed): Método HTTP é bloqueado pelo servidor.
 - 500 (Internal Server Error): Algum problema interno do servidor (bug).
 - Lista completa:
<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Componentes - Protocolo HTTP

- As faixas de Códigos de Status são reservadas para significados específicos:
 - **1xx**: dados informativos sobre a requisição no momento (uso por servidores de aplicação).
 - **2xx**: requisição terminou com sucesso.
 - **3xx**: indica alguma ação que será tomada para término da requisição.
 - **4xx**: erro no cliente ao construir e executar a requisição.
 - **5xx**: erro no servidor ao processar a requisição ou construir a resposta.
- Links sobre StatusCodes:
 - Cachorros: <https://httpstatusdogs.com/>
 - Gatos: <https://http.cat>

Componentes - Protocolo HTTP - Exemplo

Headers Preview Response Cookies Timing

General

Request URL: <http://www.impacta.edu.br/>
 Request Method: GET
 Status Code: 200 OK
 Remote Address: 189.39.8.7:80
 Referrer Policy: no-referrer-when-downgrade

Response Headers (12)

Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
 Accept-Encoding: gzip, deflate
 Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4
 Cache-Control: no-cache
 Connection: keep-alive
 Host: www.impacta.edu.br
 Pragma: no-cache
 Upgrade-Insecure-Requests: 1
 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36

Response Headers [view source](#)

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
 Content-Length: 32011
 Content-Type: text/html; charset=UTF-8
 Date: Mon, 14 Aug 2017 20:05:15 GMT
 Expires: Thu, 19 Nov 1981 08:52:00 GMT
 Link: <<http://www.impacta.edu.br/wp-json/>>; rel="https://api.w.org/"
 Link: <<http://wp.me/89GCP>>; rel=shortlink
 Pragma: no-cache
 Server: Microsoft-IIS/7.5
 Set-Cookie: PHPSESSID=15eigm41mb46cgk5r7j9ggmn91; path=/
 X-Powered-By: ASP.NET
 X-Powered-By: PHP/5.3.24



FACULDADE IMPACTA



Componentes - Protocolo HTTPS

- O protocolo HTTP transfere seus dados de maneira pura, ou como chamamos, em *plain text*.
- Isso quer dizer que se alguém estiver espionando (*sniffando*) a rede, ele lerá toda a informação que estamos trafegando na rede (senhas, cartão de crédito, etc).
- Para isso, existe um tipo específico de HTTP chamado de **HTTP Secure** (HTTPS).
- Durante a transmissão das informações há a criptografia dos dados, onde apenas o servidor e cliente conseguem ler.

Componentes - Protocolo HTTP/2

O protocolo HTTP teve diversas evoluções ao longo do tempo:



A versão mais largamente usada é a HTTP/1.1.

A versão HTTP/2 já é completamente estável e recomendada para todos os projetos recentes. (Baseada no protocolo SPDY do Google)

Componentes - Protocolo HTTP/2

- Principais diferenças entre as versões:
 - **Segurança:** Mesmo não fazendo parte da especificação, os navegadores estão obrigando a todas as conexões HTTP2 serem seguras (HTTPS).
 - **Compressão:** Por padrão comprime os dados enviados em cada requisição (cookies, cabeçalhos e corpo).
 - **Multiplexação:** Permite o retorno de diversos arquivos em uma única conexão com o servidor.
 - **Server Push:** O servidor envia arquivos dependentes em conjunto com o recurso pedido.

Obrigado!

Perguntas?