

SI401

Programação para a Web

2º Semestre – 2024

Tópico 1

Introdução

Unidade 01

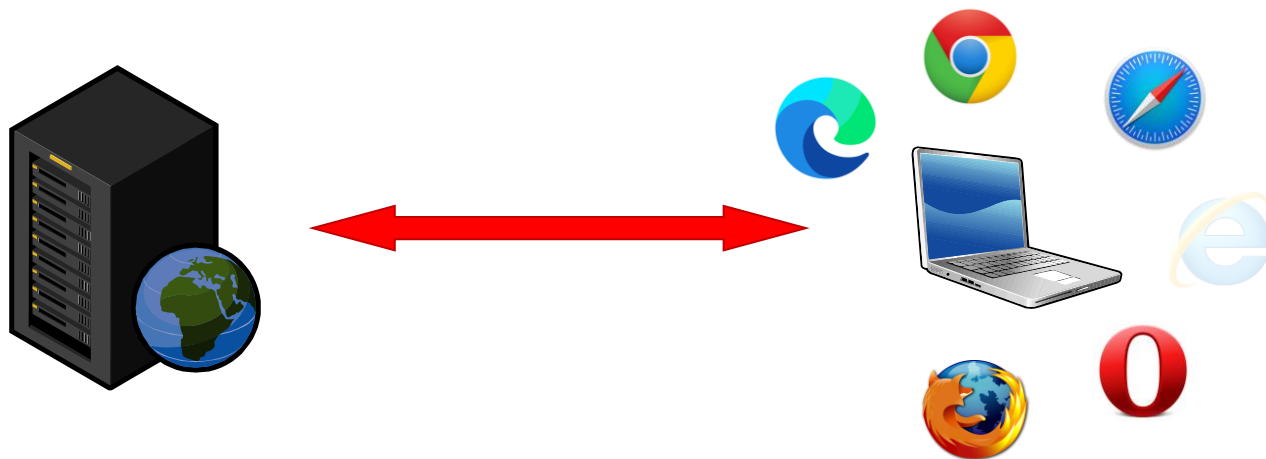
Prof. Guilherme Palermo Coelho
guilherme@ft.unicamp.br

Roteiro

- Como funciona a Web;
- Interação *browser* – servidor:
 - Decodificação (*parsing*) de uma URL;
 - Envio de requisição (HTTP);
 - A resposta do servidor;
 - O funcionamento de um servidor web;
- O que é “Programação Web”?
- Referências.

COMO FUNCIONA A INTERNET

Como funciona a Web



- Quando digitamos uma URL (*Uniform Resource Locator* ou *Universal Resource Locator*) ou clicamos em um *link* exibido em um navegador, quais são os passos envolvidos até a exibição do documento?

Como funciona a Web

- Ao ser solicitado um documento para o navegador (ou *browser*), são realizados os seguintes passos:
 - O navegador decodifica a URL → identifica o servidor web responsável pelo armazenamento do documento desejado;
 - O navegador entra em contato com o servidor e faz a requisição do documento;
 - O servidor envia uma mensagem de volta ao *browser*, contendo o documento desejado (ou uma mensagem de erro);
 - Em sua resposta, o servidor também notifica o navegador sobre o tipo do documento retornado (HTML, PDF, ZIP, etc.);

Como funciona a Web

- Ao ser solicitado um documento para o navegador (ou ***browser***), são realizados os seguintes passos:
 - De posse do documento e tendo ciência do seu tipo, o navegador exibe o documento para usuário:
 - Caso seja um formato suportado, o próprio navegador se responsabiliza pela exibição;
 - Caso contrário, ele dá ao usuário a opção de salvar o documento ou de abrir com outro programa.

Como funciona a Web

- **Servidor web** (ou *web server*): programa de computador que ouve/recebe requisições de navegadores e as responde/executa;
- **Navegador**: *programa cliente* que solicita e obtém informações de servidores
 - É capaz de exibir documentos codificados em HTML (*HyperText Markup Language*);
 - Caso este documento contenha referências a imagens, clipes de áudio, vídeo, etc., o navegador pode se encarregar de gerar requisições para os servidores em que cada um destes recursos está armazenado (caso esteja configurado para isso);

Como funciona a Web

- As requisições e respostas trocadas entre navegadores e servidores web são feitas em mensagens geralmente codificadas no protocolo HTTP (*HyperText Transfer Protocol*):
 - Uma “linguagem comum” utilizada entre navegadores e servidores web;
 - Outros protocolos HTTPS, FTP...
- A comunicação propriamente dita (fluxo de dados – bits, bytes) é feita seguindo os protocolos TCP e IP.

INTERAÇÃO BROWSER-SERVIDOR

Decodificação (*parsing*) de URL

- Estrutura geral de uma URL:

protocolo://servidor/URI-requisicao

- **protocolo**: especifica o protocolo a ser utilizado na comunicação entre o navegador e o servidor;
- **servidor**: indica qual servidor deve ser contatado;
- **URI-requisicao**: corresponde ao nome (URI = *Unified Resource Identifier*) usado pelo servidor para identificar o documento solicitado;

Decodificação (*parsing*) de URL

- Exemplo:

`http://www.ft.unicamp.br/sites/default/files/logo02_1_0.png`

- protocolo: HTTP;
- servidor: www.ft.unicamp.br;
- URI-requisicao: sites/default/files/logo02_1_0.png
- Documento:



Faculdade de
Tecnologia

Envio da requisição (HTTP)

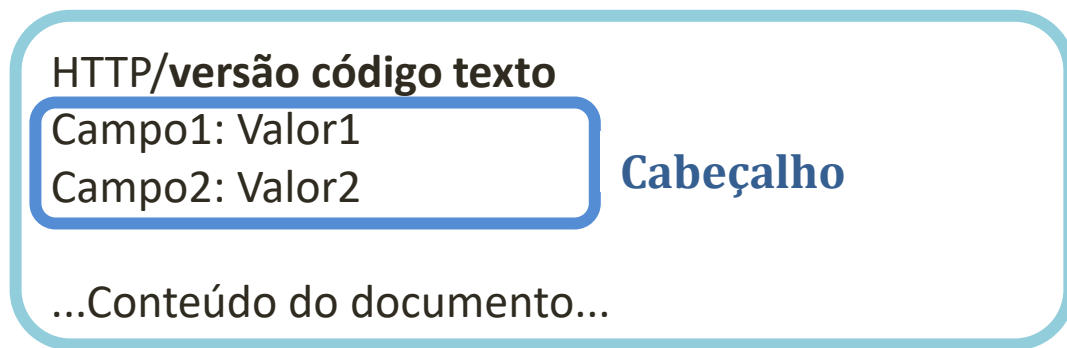
- Para requisitar um documento via HTTP para um dado servidor, o browser envia uma mensagem parecida com a seguinte:

GET /**URI-requisicao** HTTP/**version**

- **GET**: indica o comando de solicitação de um recurso;
 - **URI-requisicao**: corresponde ao identificador do documento desejado;
 - **HTTP/version**: indica a versão do protocolo HTTP a ser usada na comunicação.
- A mensagem de solicitação geralmente contém mais informações que não serão tratadas neste momento (vide [1] e [2]).

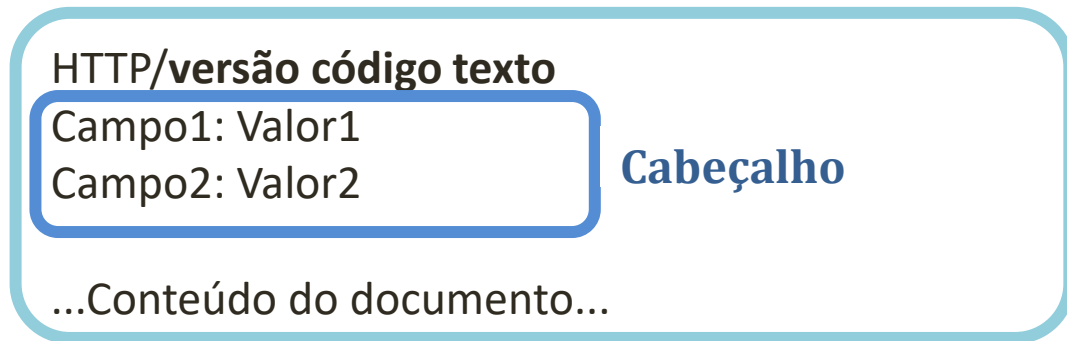
Resposta do servidor (HTTP)

- Quando um servidor recebe uma requisição HTTP do tipo GET, ele localiza o documento solicitado e o retorna para o cliente.
- A mensagem de resposta de um servidor tem o seguinte formato:



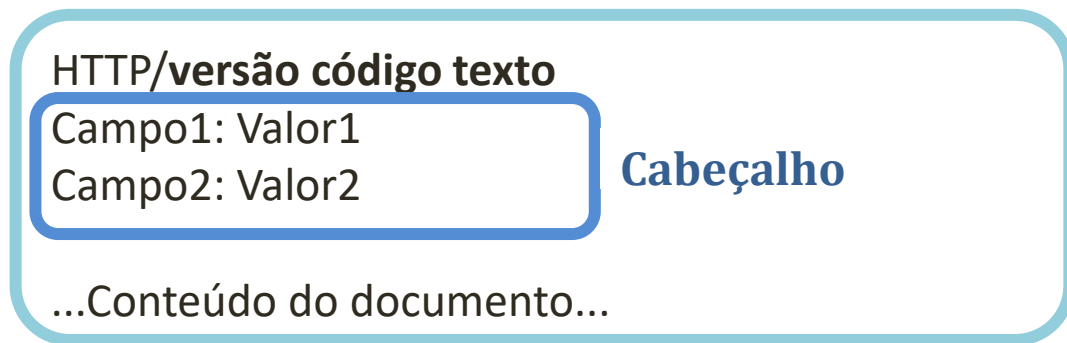
- HTTP/versão: indica a versão do protocolo HTTP usada na comunicação.

Resposta do servidor (HTTP)



- **código**: número de três dígitos que indica o status da resposta.
 - **200**: OK (recurso encontrado);
 - **404**: não encontrado;
 - **500**: erro interno do servidor;
 - ... (mais informações em [1]).

Resposta do servidor (HTTP)



- **texto**: explicação textual da resposta, para facilitar a compreensão do usuário.
- **CampoX:ValorX**: contém outras informações, inclusive sobre o documento (são os *headers*).

Resposta do servidor (HTTP)

- Exemplo de resposta:

```
HTTP/1.0 200 OK  
Date: Fri, 31 Dec 1999 23:59:59 GMT  
Content-Type: text/html  
Content-Length: 1354
```

```
<html>  
<body>  
<h1>Happy New Millennium!</h1>  
(more file contents)  
.  
.  
.  
</body>  
</html>
```


Resposta do servidor (HTTP)

- É importante notar que, para o navegador, o servidor é uma **caixa-preta**:
 - Não importa se o servidor já tinha o documento armazenado, se o documento foi gerado dinamicamente após a solicitação etc.
 - Para o cliente, só importa o **recebimento da resposta**, seja com o documento solicitado ou com uma mensagem de erro.
- Até a versão 1.1 do protocolo HTTP, as mensagens eram trocadas em arquivos-texto.
 - A partir de HTTP/2 (ou HTTP 2.0), a transmissão passou a ser em arquivos binários.

Funcionamento de um Servidor Web

- Um servidor web geralmente utiliza um diretório em disco (na máquina em que está sendo executado) como diretório-raiz;
- É possível configurar um arquivo *default* a ser retornado caso “/” seja solicitado:
 - Solicitação: `http://www.meusite.com`
 - Retorno: `http://www.meusite.com/index.html`
- Na configuração mais simples: solicitação para ***/teste/arquivo.html*** retornará para o cliente o arquivo ***arquivo.html*** armazenado no subdiretório ***teste*** do diretório-raiz do servidor.
- É possível adotar outros tipos de configurações.

Funcionamento de um Servidor Web

- Exemplos de servidores web:
 - Apache:
 - 19,80% do mercado (*all sites*, Junho 2024 – [4]);
 - <http://httpd.apache.org/>
 - nginx (pronuncia-se “Engine-X”):
 - 21,53% do mercado (*all sites*, Junho 2024 – [4]);
 - <http://www.nginx.org/en/>
 - Cloudfare:
 - 10,80% do mercado (*all sites*, Junho 2024 – [4]);
 - <https://www.cloudflare.com/pt-br/>

O QUE É “PROGRAMAÇÃO WEB”?

O que é “Programação Web”?

- **Programação web:** pode ser entendida como o desenvolvimento de software baseados na estrutura da web existente atualmente.
 - Software que utilizam recursos como *browsers*, *servidores web*, *protocolos Web* (HTTP...)...
- Tecnologias básicas associadas à programação web [3]:
 - **HTML/HTML5:** linguagem de marcação utilizada para criar páginas web;
 - **CSS/CSS3 (*Cascading Style Sheets*):** utilizado para definir como uma página HTML deverá ser exibida pelo navegador;
 - **XML (*eXtensible Markup Language*):** linguagem de marcação proposta para o armazenamento e transporte de dados. Mantém os dados de forma estruturada;

O que é “Programação Web”?

- **Software para Web** podem ser divididos em dois grupos:

Software baseado em *scripts* executados tanto do lado do cliente (***browser scripting***) quanto do servidor (***server scripting***)



Aplicativos que disponibilizam suas funcionalidades para uso via web (***web services***)

Software baseado em *scripts*

- Algumas das principais tecnologias (***browser scripting*** - [3]):



- **JavaScript***: linguagem de *script* mais utilizada atualmente;

- **jQuery**: é uma biblioteca de funções JavaScript.



- **HTML DOM**: define um padrão para acessar e manipular documentos HTML;



- **DHTML**: é o termo dado à combinação de HTML, *JavaScript*, DOM e CSS;
- **React**: uma biblioteca *JavaScript* para construção de interfaces de usuário.



- **AJAX** (*Asynchronous JavaScript and XML*): é uma forma de utilização de JavaScript e XML que permite a criação de páginas rápidas e dinâmicas.

Software baseado em *scripts*

- Algumas das principais tecnologias (*server scripting* – [3]):



- **PHP** (*PHP: Hypertext Preprocessor*): uma das principais linguagens atuais para *server scripting*;
- **ASP** (*Active Server Pages*): linguagem de *server scripting* criada pela Microsoft;
- **ASP.Net**: é uma geração mais recente de ASP. Não é compatível com ASP;
- **Node.js**: ambiente para execução de *JavaScript* voltado para o *back-end* de aplicações.

REFERÊNCIAS

Referências

- [1] HTTP Semantics:

<https://www.rfc-editor.org/rfc/rfc9110.html>

- [2] Mozilla Developer Network – Uma visão geral do HTTP:

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

- [3] GRAY N. *Web Server Programming*. 1ª ed. Wiley, 2003.

- [4] Netcraft – June 2024 Web Server Survey:

<https://www.netcraft.com/blog/june-2023-web-server-survey/>

- [5] W3Schools Online Web Tutorials:

<http://w3schools.com/default.asp>