



Linguagem de Programação Comercial 2020/1

DEPARTAMENTO DE COMPUTAÇÃO



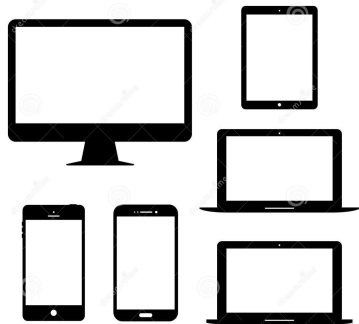
Arquitetura de aplicações WEB

Roteiro

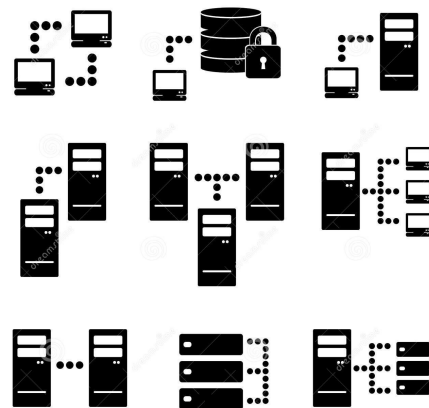
- ▶ Cliente / Servidor;
- ▶ Plataforma WEB;
- ▶ Navegadores e Servidores;
- ▶ Navegação;
- ▶ Tipos de Resposta;
- ▶ Comunicação;
- ▶ Tecnologias de Apresentação;
- ▶ Tecnologias de Interatividade;

Cliente / Servidor

- ▶ Método de distribuição multiplataforma de aplicações;
- ▶ Dois ou mais computadores interagem;
- ▶ Comunicação por requisições;



**CLIENT
E**



SERVIDOR

Cliente / Servidor

Cliente (*front-end*)

- É o processo ativo na relação;
- Inicia e termina as conversações com os Servidores;
- Não se comunica com outros Clientes;

Servidor (*back-end*)

- É o processo reativo na relação;
- Possui uma execução contínua;
- Recebe e responde às solicitações dos Clientes;
- Presta serviços distribuídos;
- Atende a diversos Clientes simultaneamente.

Plataforma WEB

- ▶ Navegadores e Servidores
- ▶ Disponibilização de recursos através de páginas WEB;
 - ▶ Hipertexto;
- ▶ Comunicação por meio do protocolo **HTTP** (*HyperText Transfer Protocol*);
 - ▶ Navegação por URL (*Uniform Resource Locator*);
 - ▶ Tipos MIME (*Multipart Internet Mail Extensions*);
- ▶ Tecnologias de Apresentação;
- ▶ Tecnologias interativas;

HTTP???

- ▶ Trata-se de um protocolo que estabelece como deve ocorrer a comunicação entre uma máquina cliente que faz pedidos para uma máquina servidora. Ele é normatizado por uma especificação, a RFC 2616 (<http://tools.ietf.org/html/rfc2616>).
- ▶ Baseado na comunicação entre uma máquina cliente que faz requisições para uma máquina servidora;
- ▶ “Request” :
 - ▶ Cabeçalho
 - ▶ Corpo
- ▶ “Response”;

X	Headers	Preview	Response	Timing
▼ General				
Request URL: http://ulbra-to.br/imagem/2019/02/07/BNR-Akademo-Inscricoes-breve/crop/978/200/80				
Request Method: GET				
Status Code: 🟢 200 OK (from memory cache)				
Remote Address: 200.199.229.77:80				
Referrer Policy: no-referrer-when-downgrade				
▼ Response Headers				
Cache-Control: max-age=172800, must-revalidate, proxy-revalidate, public				
Content-Transfer-Encoding: binary				
Content-Type: image/jpeg				
Date: Fri, 15 Feb 2019 16:17:43 GMT				
Expires: Sun, 17 Feb 2019 16:17:44 GMT				
Memcache: true				
Pragma: public				

Cabeçalho da Requisição

- ▶ A URL que gerou a requisição (Request URL);
- ▶ O tipo de resposta esperada do servidor (Accept);
- ▶ O idioma nativo do browser que disparou a requisição (Accept-Language).

Método HTTP

- ▶ RequestMethod.
- ▶ Indica que tipo de ação a URL que foi disparada para o servidor deverá realizar, dando sentido semântico à requisição.
- ▶ O protocolo **HTTP** tem uma série de métodos, como **GET**, **POST**, **PUT**, **DELETE**, HEAD, OPTIONS, TRACE e CONNECT.
- ▶ Nós, na maioria do tempo, utilizamos mais os métodos GET, POST, PUT e DELETE.

Método	Significado
GET	Significa que queremos “pegar” algo no servidor: uma página, por exemplo.
POST	Significa que estamos querendo incluir alguma coisa no servidor.
PUT	Significa que estamos querendo atualizar alguma coisa no servidor.
DELETE	Significa que estamos querendo apagar alguma coisa do servidor.

X

Headers

Preview

Response

Timing

▼ General

Request URL:

http://ulbra-to.br/imagem/2019/02/07/BNR-Akademo-Inscricoes-breve/crop/978/200/80

Request Method:

GET

Status Code:

🟢 200 OK (from memory cache)

Remote Address:

200.199.229.77:80

Referrer Policy:

no-referrer-when-downgrade

▼ Response Headers

Cache-Control:

max-age=172800, must-revalidate, proxy-revalidate, public

Content-Transfer-Encoding:

binary

Content-Type:

image/jpeg

Date:

Fri, 15 Feb 2019 16:17:43 GMT

Expires:

Sun, 17 Feb 2019 16:17:44 GMT

Memcache:

true

Pragma:

public

Status de Resposta



200 Resultado OK

400 Erro no Client

500 Erro no Server

CATEGORIAS DE CÓDIGOS

Códigos de Status
(como está a
comunicação?)

Familia 2xx

- ▶ **200** OK
- ▶ **201** Created
- ▶ **202** Accepted

Familia 4xx

- ▶ **400** Bad Request
- ▶ **401** Unauthorized
- ▶ **403** Forbidden
- ▶ **404** Not Found
- ▶ **405** Method Not Allowed



Familia 5xx

- ▶ **500** Internal Server Error
- ▶ **502** Bad Gateway
- ▶ **503** Service Unavailable
- ▶ **504** Gateway Timeout
- ▶ **508** Loop Detected



Navegadores e Servidores

- ▶ Plataforma baseada no protocolo HTTP;
- ▶ Hipertexto é uma forma não linear de publicação de informações:
- ▶ O hipertexto baseia-se em ligações que são chamadas de vínculos (links) ou hipervínculos (hyperlinks);
- ▶ O servidor Web administra um sistema de arquivos e diretórios virtuais para atender à requisições dos clientes HTTP.

URL

- ▶ Todas as requisições utilizam a sintaxe de URL – *Uniform Resource Locator*,
- ▶ Todos os recursos disponibilizados por um serviço estão associados a uma URL;
- ▶ Contém duas informações essenciais:
 - ▶ COMO transferir o objeto;
 - ▶ ONDE encontrá-lo.

`<protocolo>://<maquina>:<porta>/<caminho_para_recurso>`

URL no Servidor

URL	Recurso Mapeado
/perfil	C:/aplicativo/perfil.html
/noticia	C:/sistema_noticias/index.html

Tipos de Resposta

- ▶ Informação utilizada pelo navegador para representar a informação na tela;
 - ▶ Imagem, um arquivo HTML, PDF ou até um formato não suportado.
- ▶ Para padronização o servidor envia o tipo dos dados baseando-se nos tipos MIME – *Multipart Internet Mail Extensions*:
 - ▶ **tipo/subtipo**

Tipos de Resposta

image/jpg

text/html

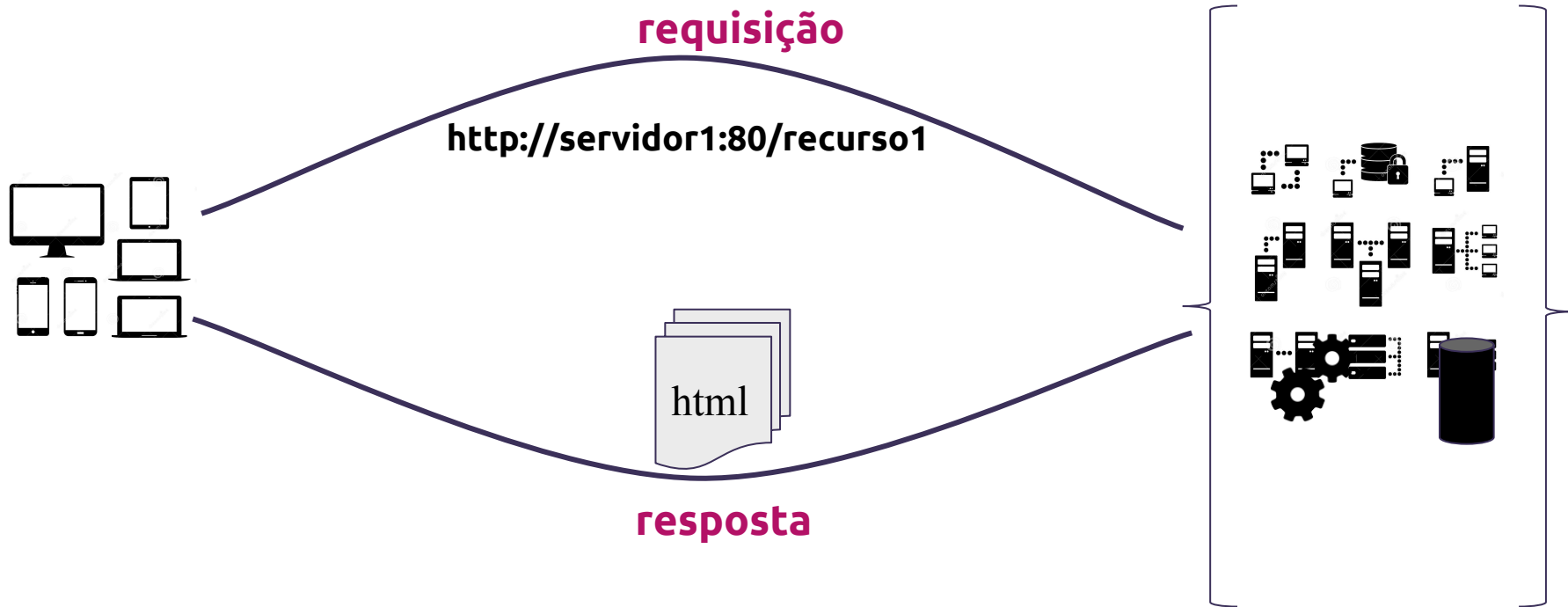
text/plain

.jpe, .jpg, .jpeg

.html, .htm, .jsp, .asp, .shtml

.txt

Comunicação





```
python -m http.server 8000
```

Estado do protocolo HTTP

- ▶ *Stateless:*
 - ▶ Requisições independentes;
 - ▶ Navegador fecha comunicação;
 - ▶ As partes não guardam informação do **estado** das requisições.

Tecnologias de apresentação

- ▶ Tecnologias destinadas à formatação e estruturação das páginas Web;
- ▶ Utilizadas para construir a interface de aplicações Web;
- ▶ Os principais padrões em uso atualmente são:
 - ▶ HTML;
 - ▶ CSS (*Cascading Style Sheets*);
 - ▶ XML (*eXtensible Markup Language*);

Tecnologias de Interatividade

- ▶ Necessidade estender as capacidades de aplicações cliente ou servidor;
- ▶ Interatividade cliente;
 - ▶ *JavaScript*;
 - ▶ *Applets*;
 - ▶ *ActiveX*;
- ▶ Interatividade servidor:
 - ▶ *PHP*;
 - ▶ *Java*;
 - ▶ *Python*;

Referências Bibliográficas

MENDES, Antonio. Arquitetura de Software: desenvolvimento orientado para arquitetura. Editora Campus. Rio de Janeiro - RJ, 2002

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: Uma abordagem profissional 8ª edição. AMGH Editora Ltda. São Paulo SP, 2016.

TANENBAUM, A. S.; WETHERALL, D. Rede de computadores 5ª edição, São Paulo, p 407 - 437, 2013.

Atividade



Arquitetura de aplicações WEB

FÁBIO CASTRO ARAÚJO