

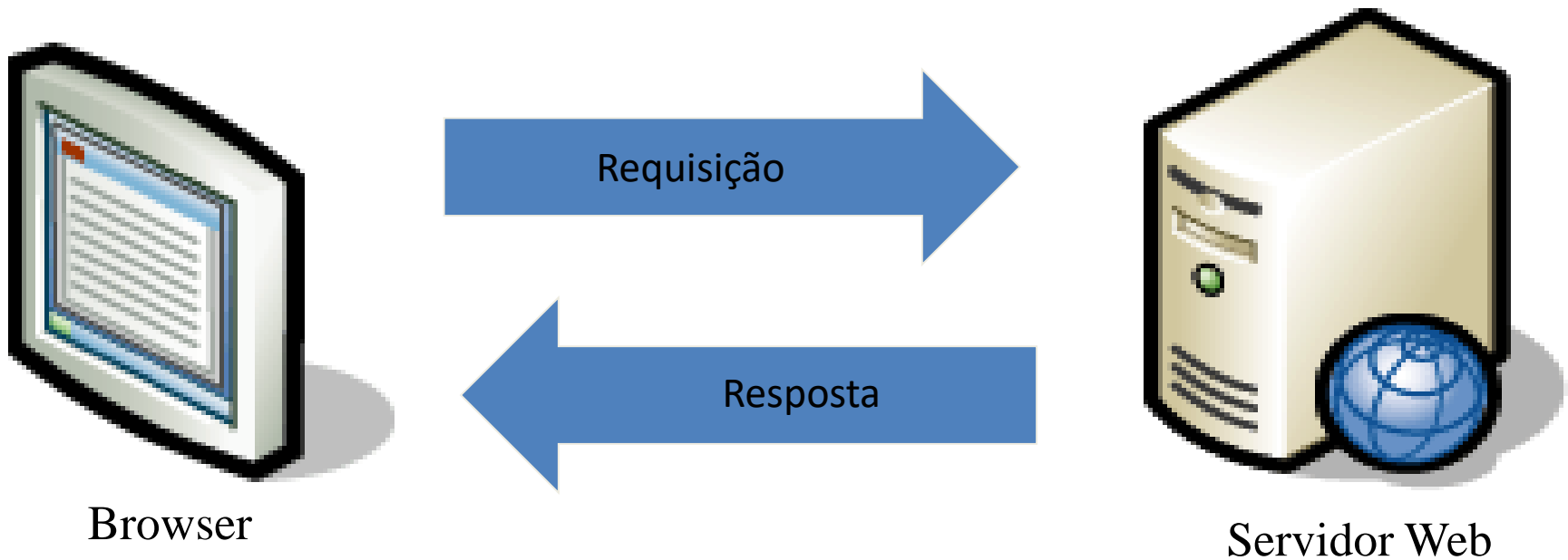
FIAP GRADUAÇÃO

DESENVOLVIMENTO DE SISTEMAS WEB

PROF. RAFAEL MATSUYAMA
profrafael.matsuyama@fiap.com.br

INTRODUÇÃO JAVA NA WEB

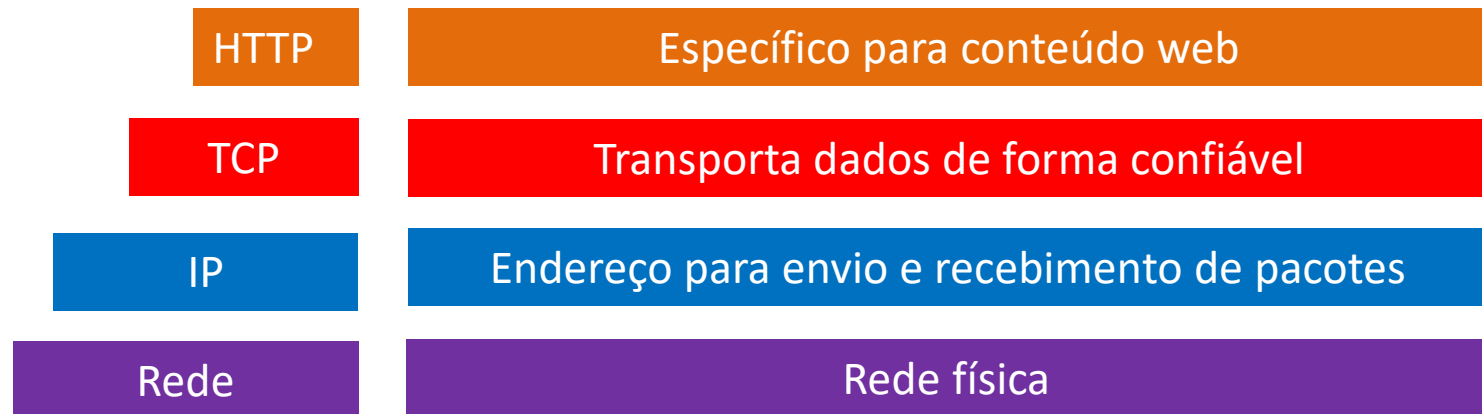
I Modelo Cliente/Servidor



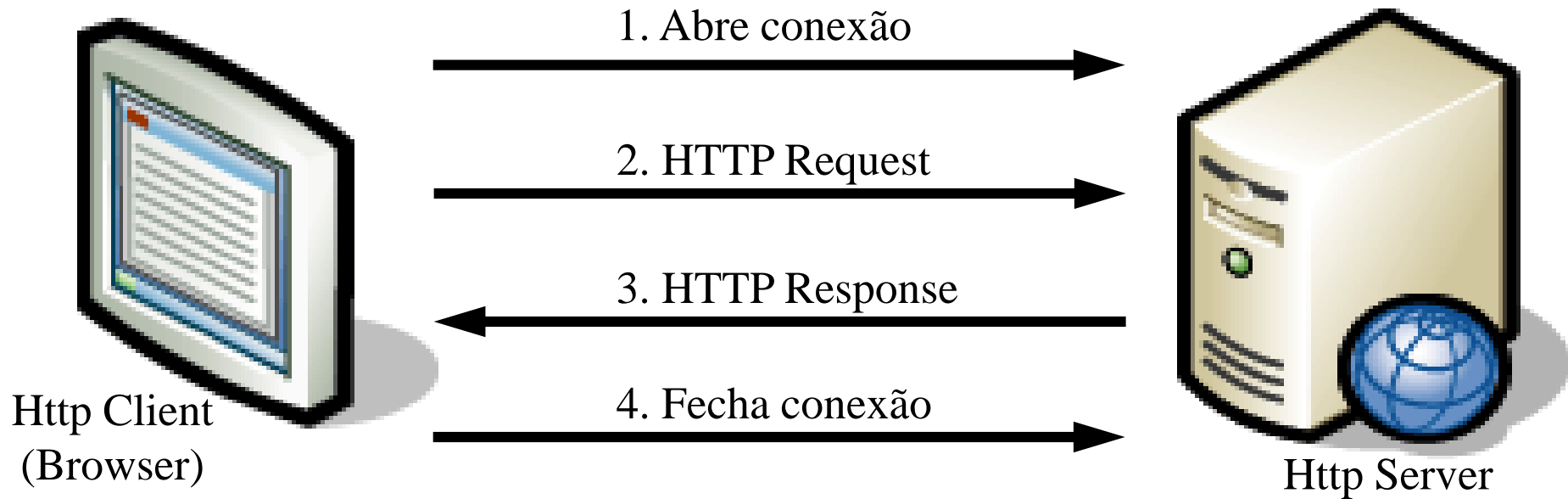
HTTP Básico

I HTTP Básico

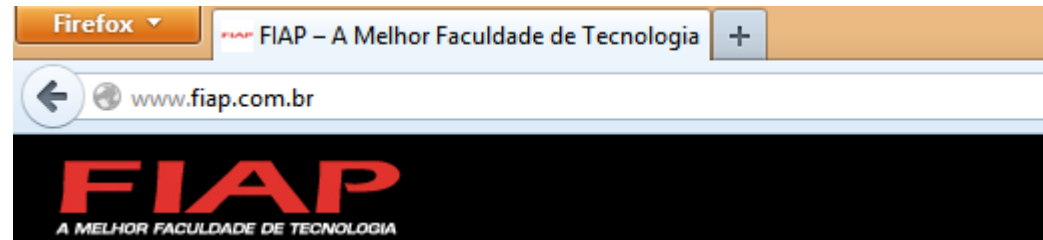
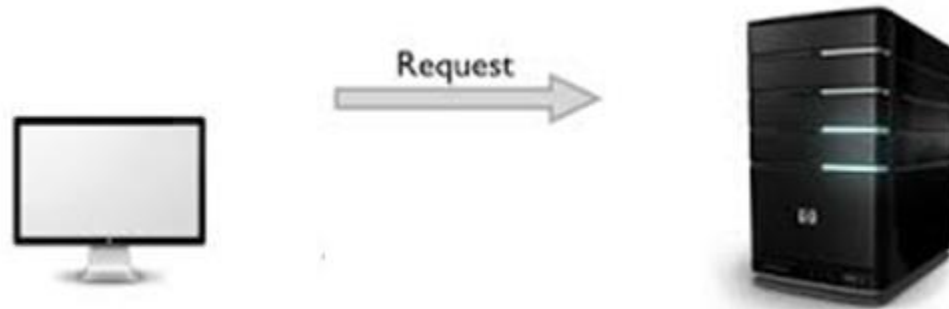
- Hypertext Transfer Protocol
- Um protocolo de rede, responsável pela transferência de conteúdo pela internet



- Baseado em Request-Response
- Cliente abre conexão e envia um request p/ servidor
- Cliente recebe response e fecha conexão
- Comunicação sem estado (stateless)
- Para o servidor cada request é o primeiro request enviado pelo cliente
- Cada recurso é identificado por único e exclusivo Uniform Resource Identifier ([URI](#))



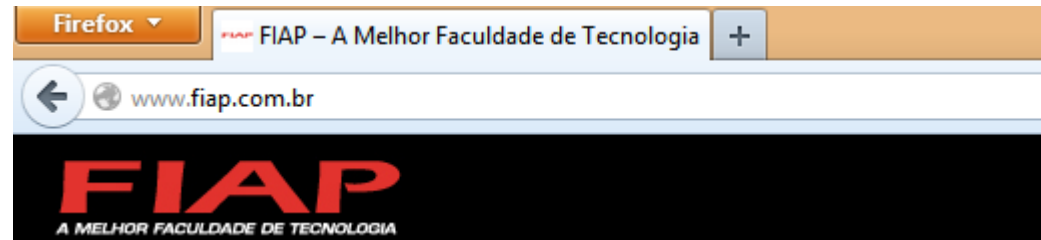
I HTTP Request



HTTP Request Header

```
GET http://www.fiap.com.br
Accept text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding gzip, deflate
Accept-Language en,pt-br;q=0.8,en-us;q=0.5,pt;q=0.3
Connection keep-alive
Cookie qtrans_cookie_test=qTranslate+Cookie+Test;
__utma=194850098.1067742818.1375301595.1375301595.1375301595.1;
__utmb=194850098.1.10.1375301595; __utmc=194850098;
__utlz=194850098.1375301595.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)
Host www.fiap.com.br
User-Agent Mozilla/5.0 (Windows NT 6.2; WOW64; rv:22.0) Gecko/20100101 Firefox/22.0
```


HTTP Response

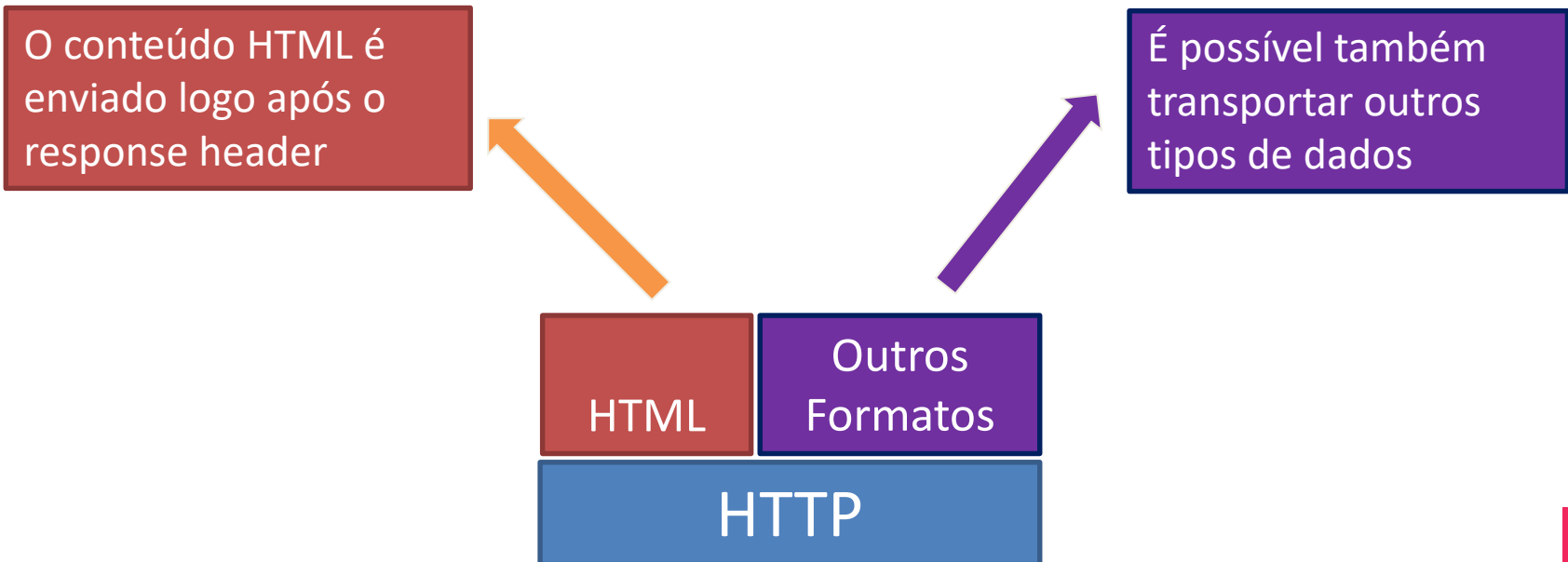


HTTP Response Header

HTTP/1.1 Status OK

Content-Length	35198
Content-Type	text/html; charset=UTF-8
Date	Wed, 31 Jul 2013 20:13:39 GMT
Server	Microsoft-IIS/7.0
Set-Cookie	qtrans_cookie_test=qTranslate+Cookie+Test; path=/; domain=www.fiap.com.br
Vary	Accept-Encoding
X-Pingback	http://www.fiap.com.br/xmlrpc.php
X-Powered-By	ASP.NET

- Os arquivos HTML são enviados ao cliente através do protocolo HTTP:



I Anatomia de uma URL

- Uniform Resource Locator (URL)
- Especificam um recurso na web

http://www.fiap.com.br :80/paginas/login.html

Protocolo

Server/Host

Porta

Path

Resource

Requisições: GET x POST

- **GET**

- O objetivo principal é requisitar dados do servidor
- Pode ser usado para enviar alguns dados ao servidor
- Dados são enviados diretamente na URL

- **POST**

- O objetivo principal é enviar dados ao servidor
- Os dados são enviados no corpo do protocolo HTTP

I Requisição GET

- Passagem de parâmetros:

http://localhost:8080/myapp/LoginServlet?login=jose&senha=1234

Login	jose
Senha	1234
<input type="button" value="Enviar"/>	

<FORM action="LoginServlet">

Login<INPUT type="text" name="login"

Senha<INPUT type="text" name="senha"

</FORM>

I Requisição GET

Header

Linha em
branco

```
GET /hello/HelloServlet?user=Joao HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, application/x-shockwave-flash, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; windows NT 5.1; SV1)
Host: 127.0.0.1
Connection: Keep-Alive
```

Requisição POST

POST /hello/HelloServlet HTTP/1.1

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, application/x-shockwave-flash, */*

Referer: http://localhost:1234/hello/hello.html

Accept-Language: en-us

Content-Type: application/x-www-form-urlencoded

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; windows NT 5.1;
SV1)

Host: 127.0.0.1

Content-Length: 12

Connection: Keep-Alive

Cache-Control: no-cache

Header

Linha em
branco

Dados

user=Joao

I Qual requisição utilizar?

- **GET**

- Formulário de pesquisa
- Requisição de página através de hiperlink
- Navegação

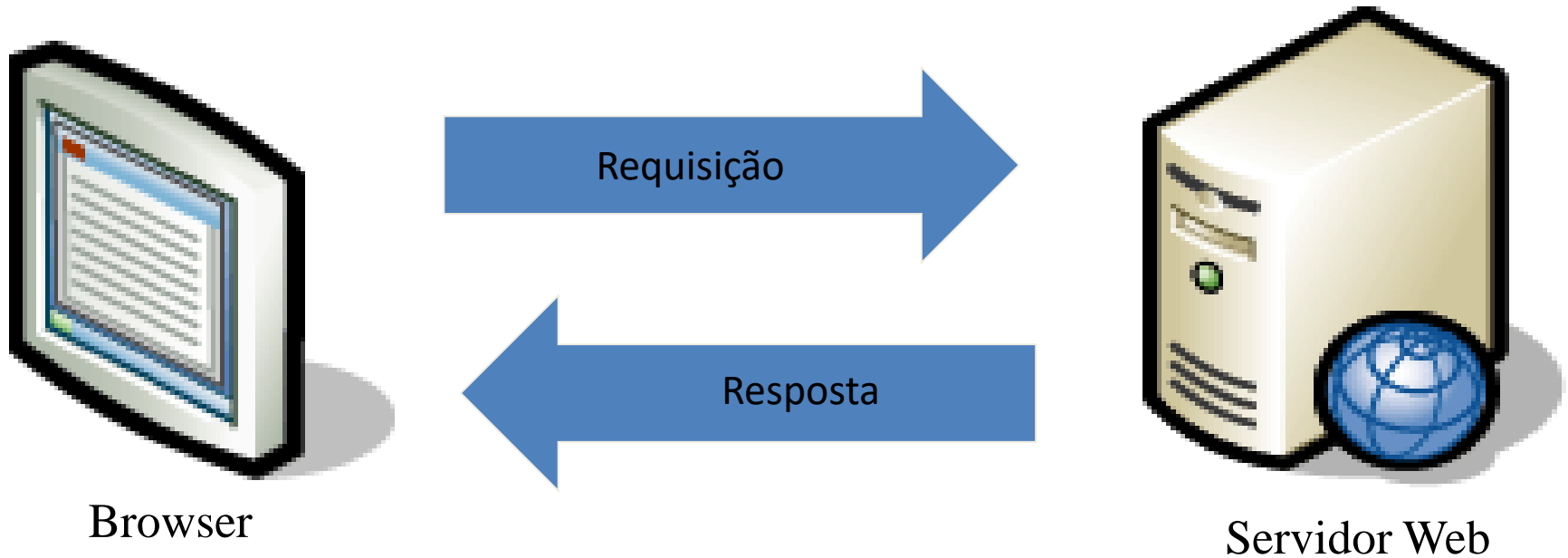
- **POST**

- Formulário de cadastro
- Formulário de login
- Upload de arquivo

APLICAÇÕES WEB

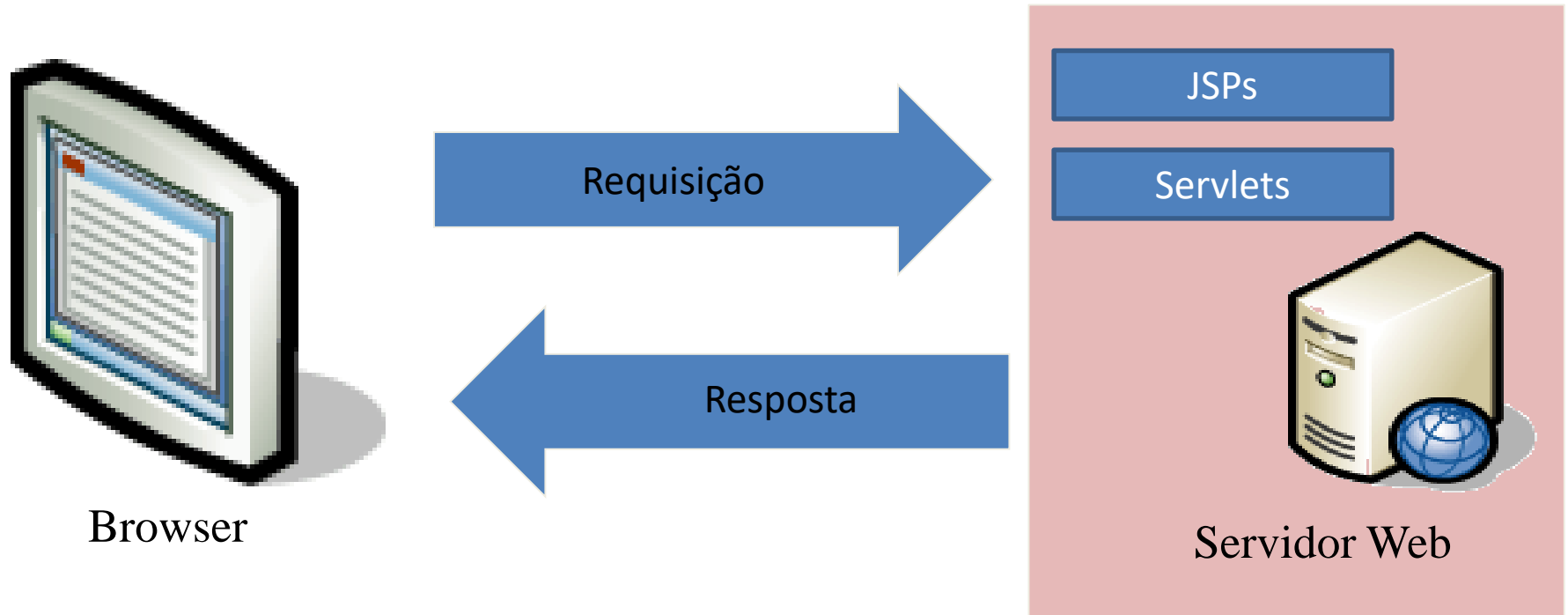
I Conteúdo Estático

- As páginas html existentes são enviados aos clientes



Conteúdo Dinâmico

- A resposta é montada dinamicamente, mas o cliente continua recebendo conteúdo html

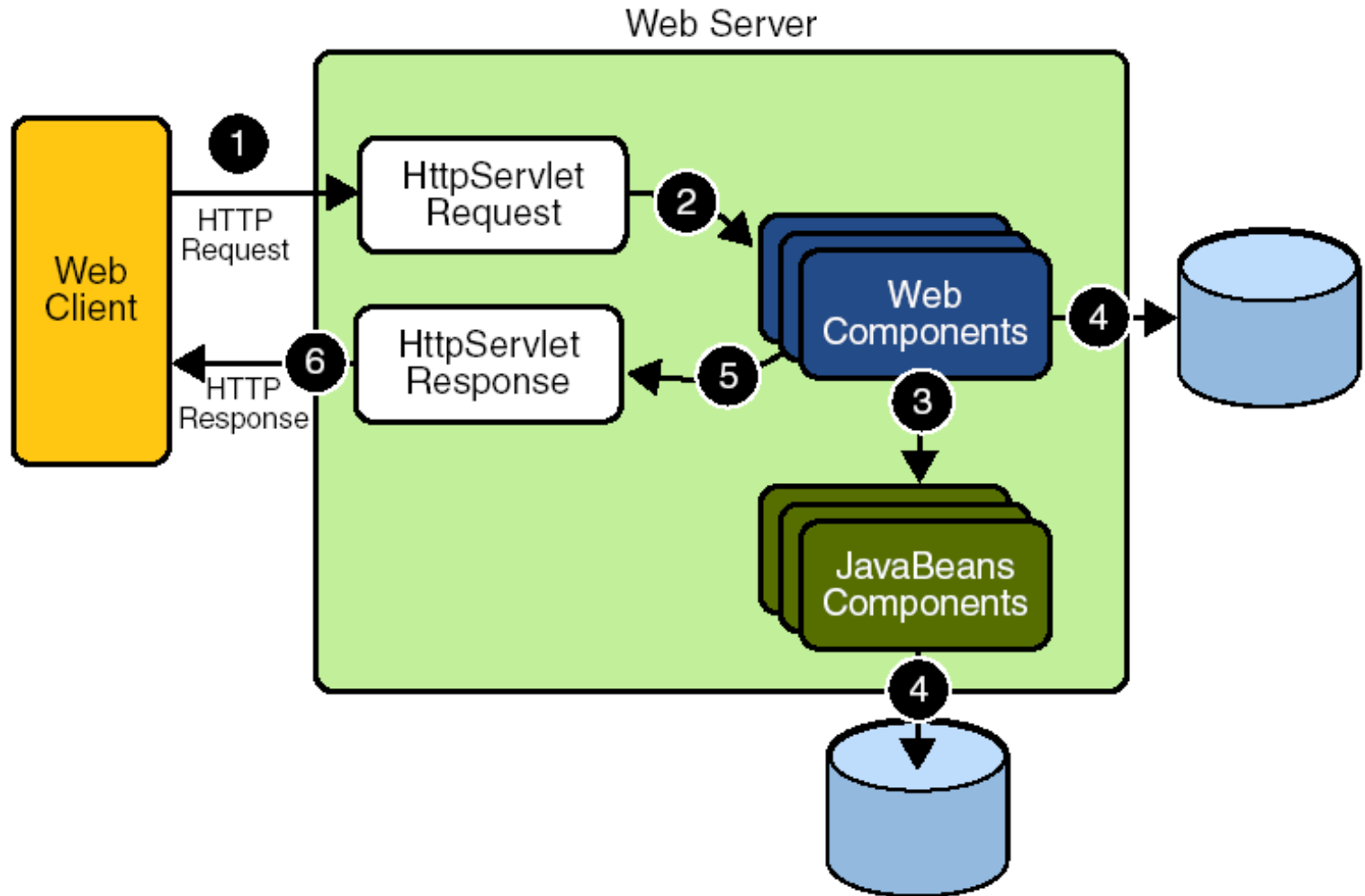


- **Java SE (Standard Edition)**
 - Base do Java
 - Ambiente de execução e bibliotecas comuns
- **Java ME (Micro Edition)**
 - Dispositivos móveis
- **Java EE (Enterprise Edition)**
 - Aplicações corporativas e internet

- **Há dois tipos de aplicações web:**
 - **Orientadas à apresentação**
 - **Páginas interativas contendo linguagens tais como HTML, XML, etc**
 - **Orientadas à serviços**
 - **Web Services**

[SUN MICROSYSTEMS, Tutorial JEE, 2007]

- Na plataforma Java EE, componentes web fornecem funcionalidades dinâmicas para um servidor web.
- Componentes web são:
 - Java Servlets
 - Páginas JSP
 - Web Services
 - JSF

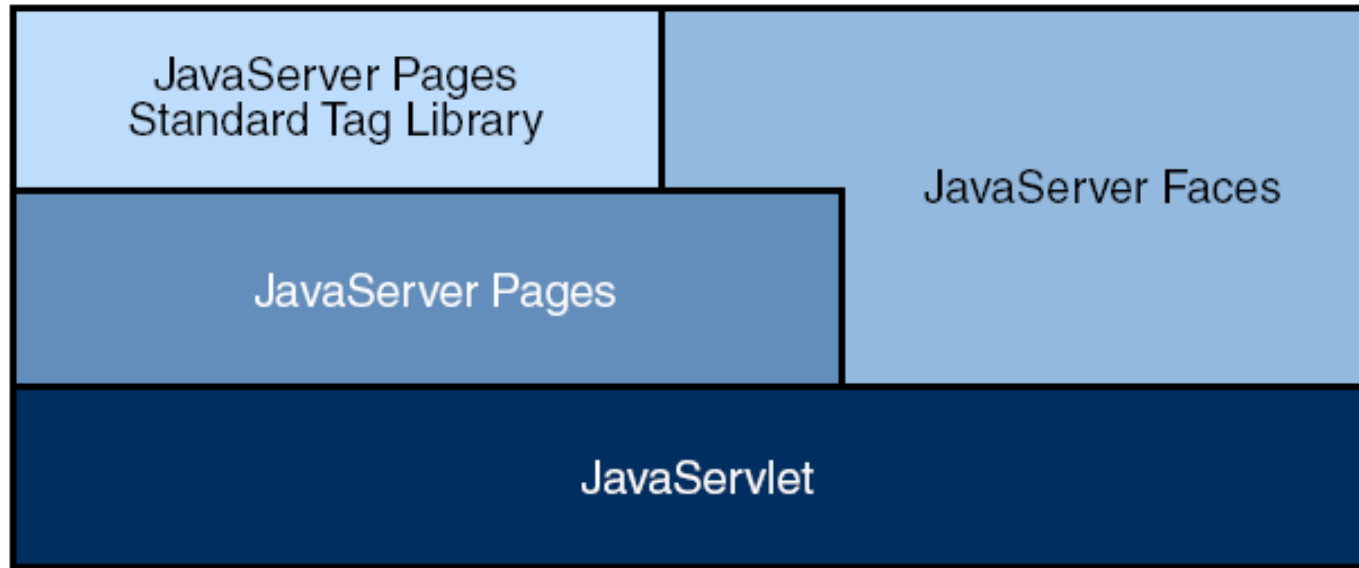


- **Servlets**

- Classes Java que processam requests dinamicamente e controem response
- Recomendados para controlar a camada de apresentação (controller) de aplicações orientadas a apresentação e para implementar serviços em aplicações orientadas a serviços (web services são normalmente implementados como servlets).

- **Páginas JSP**

- Documentos texto que são executados como servlets
- Recomendadas para páginas baseadas em HTML, WML, DHTML, Web Designers



- Tecnologia Java Servlet é a base de todas as tecnologias Web
- Mesmo que você não tenha intenção de escrever Servlets é bom familiarizar-se com essa tecnologia

- Todos os componentes Web devem estar dentro de um servidor de aplicações (container web)
- Um container Web fornece serviços tais como
 - Encaminhamento de requests (request dispatching)
 - Segurança
 - Acesso concorrente
 - Gerenciamento do ciclo de vida dos componentes web
 - Controle de transações
 - email
 - etc

I Servidor de Aplicações

- Cuida da parte de infraestrutura da nossa aplicação;
- Neste semestre iremo utilizar o tomcat versão 9 como servidor de aplicações.
- Mas existem outros:
 - Glassfish
 - Jetty
 - JBoss



<http://tomcat.apache.org/>

Copyright © 2018 Prof. Rafael Matsuyama / Prof. Luís Carlos de S Silva

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).