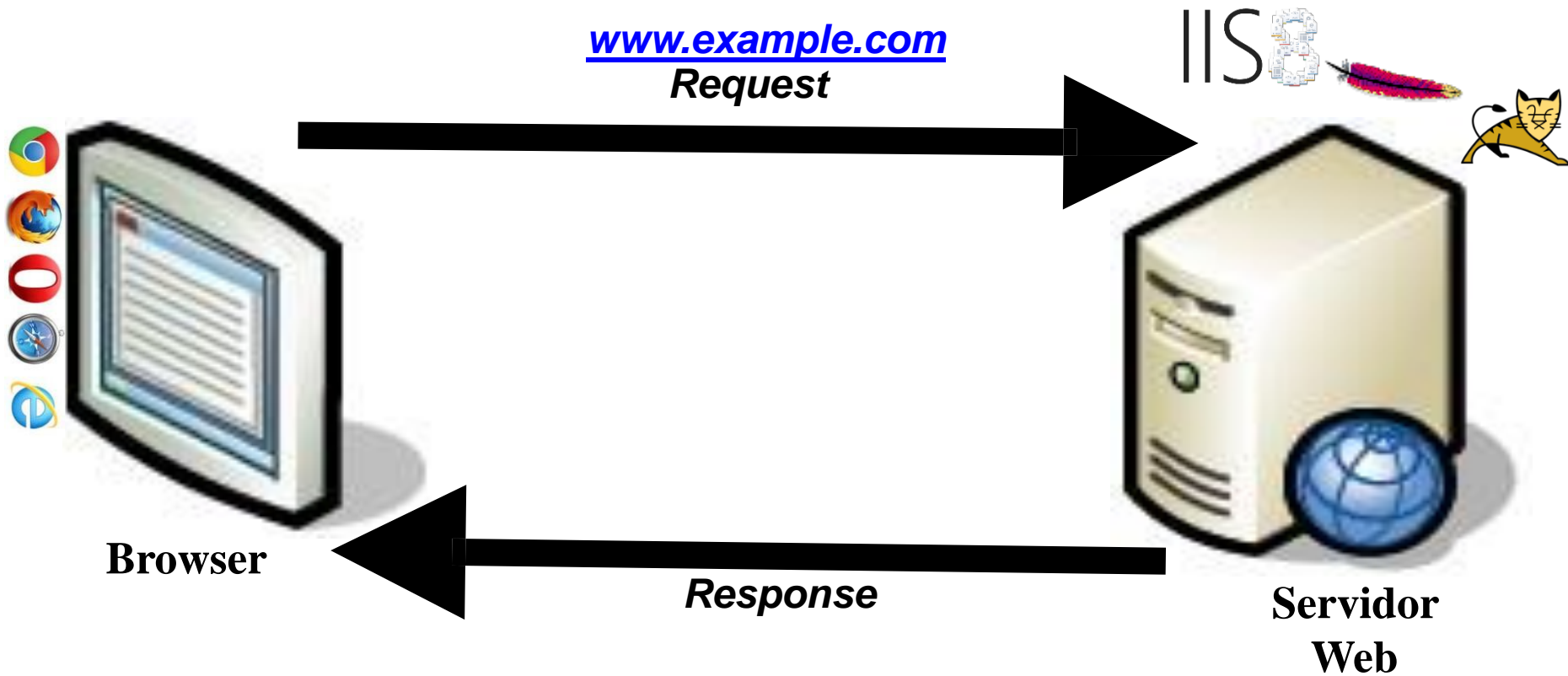




PROF. Gustavo Molina

HTTP Básico

Modelo Client/Server



HTTP Básico

- **H**ypertext **T**ransfer **P**rotocol
- Um protocolo de rede, responsável pela transferência de conteúdo pela internet

<i>HTTP</i>	Específico para conteúdo web
<i>TCP</i>	Transporta dados de forma confiável
<i>IP</i>	Endereço para envio e recebimento de pacotes
<i>Rede</i>	Rede física

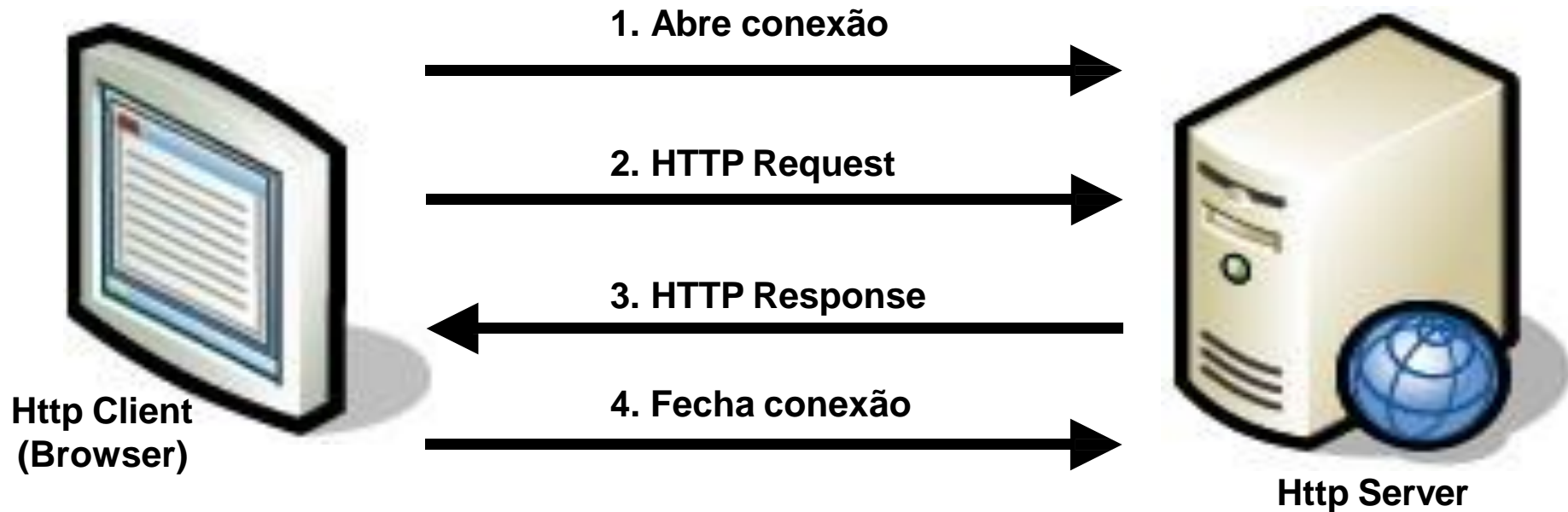
HTTP Básico



- 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**)

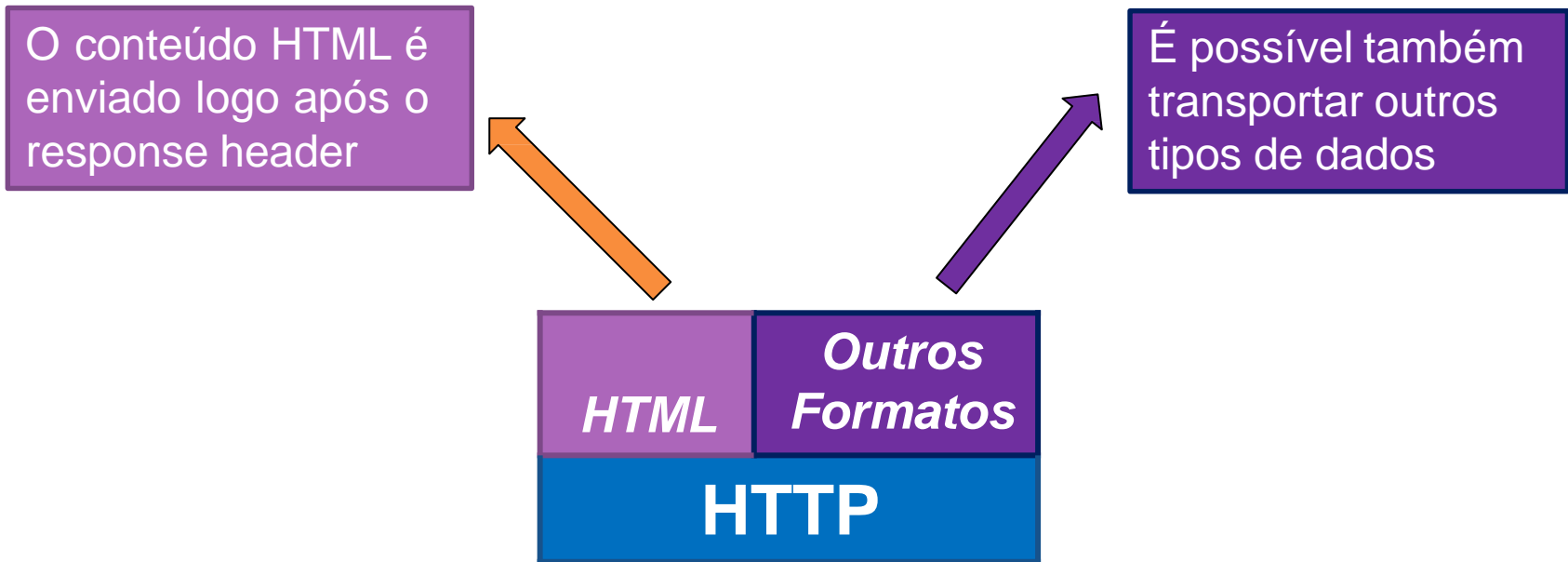
* recurso = arquivos(html, xml, jpg, png, flv, etc)

HTTP Básico



HTML

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



Anatomia de uma URL

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

<http://www.unip.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

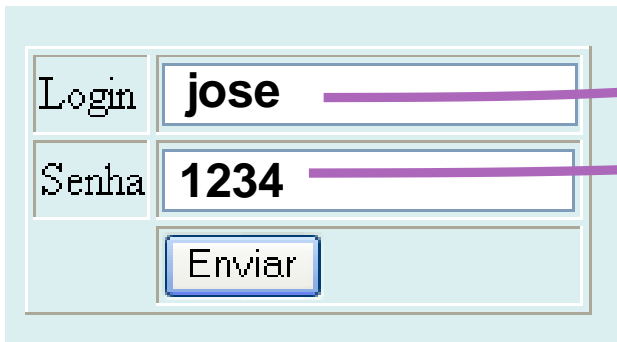
- **POST**

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

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>`

Requisição GET

GET /myapp/LoginServlet?login=jose&senha=1234 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

Header

*Linha em
branco*

Requisição POST

POST /myapp/LoginServlet HTTP/1.1

Header {
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

Linha em branco {

Dados { **login=jose&senha=1234**

Qual requisição utilizar?

- **GET**

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

- **POST**

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

Responda?

- Qual o modelo mais utilizado na internet?

R: Modelo Client-Server

- No que é baseado o HTTP?

R: Request e Response

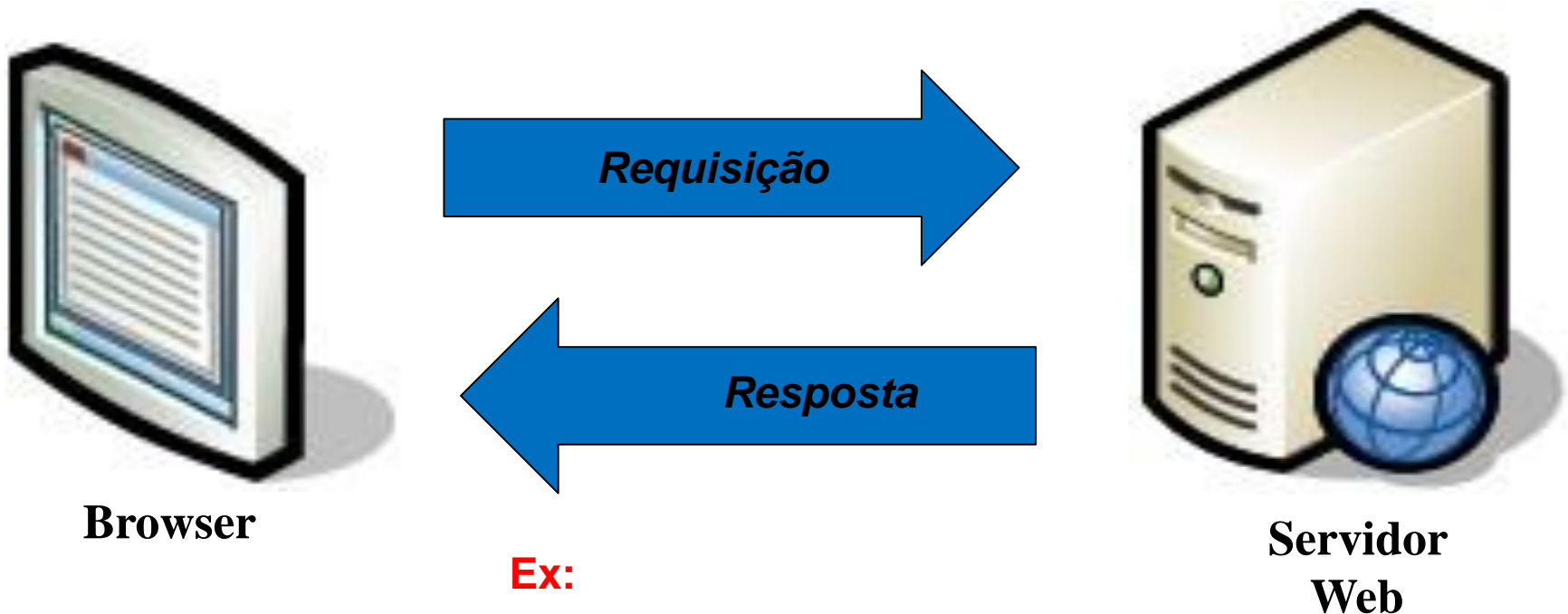
- Cada recurso é identificado por único e exclusivo ?

R: Uniform Resource Identifier (URI)

Aplicações Web

Conteúdo Estático

- As páginas html existentes são enviados aos clientes



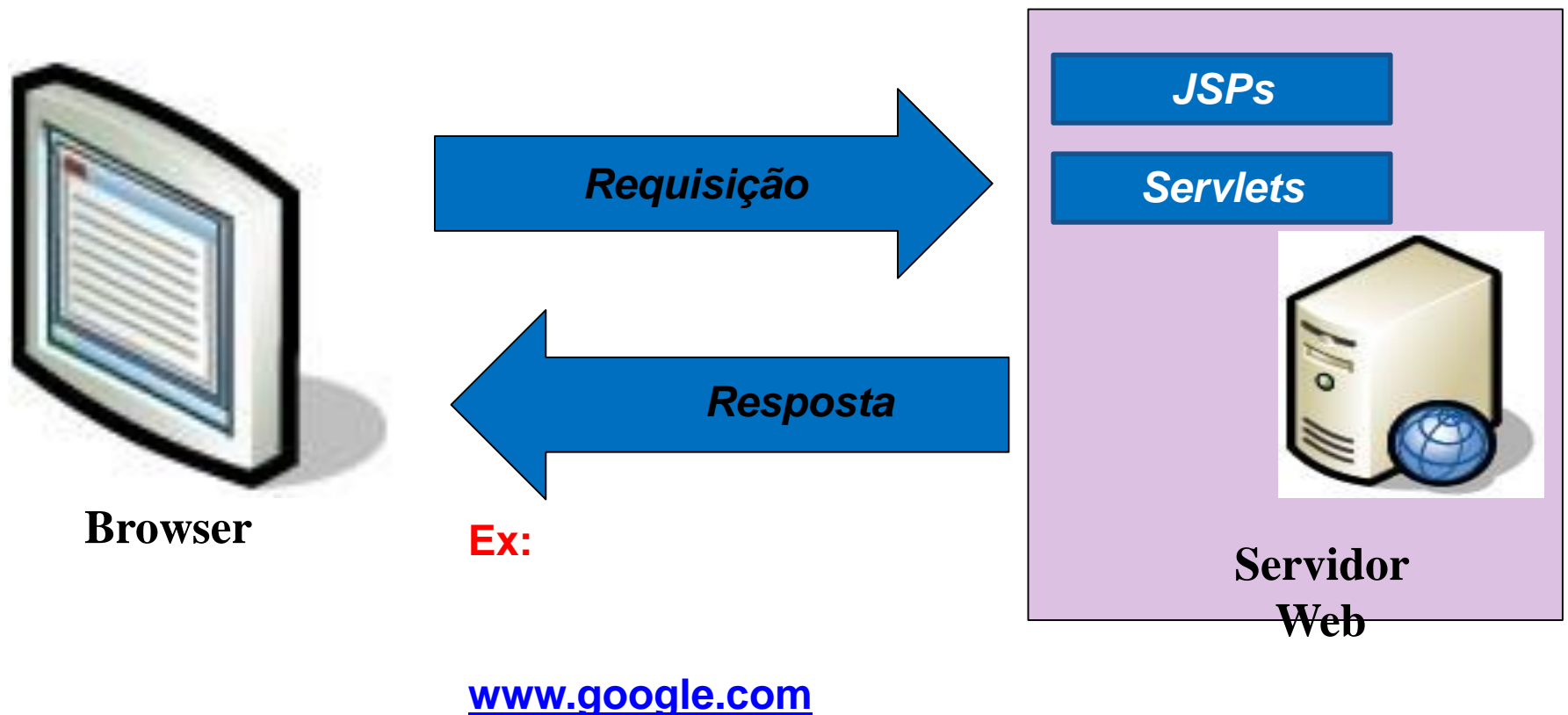
Ex:

<http://getbootstrap.com.br/>

<http://jquery.com/download/>

Conteúdo Dinâmico

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



Java - Plataforma

Existem quatro plataformas para a programação em Java, cada uma com sua JVM e API:

– Java Platform, Standard Edition (Java SE) • Provê a API base para o desenvolvimento em Java, essa plataforma define desde os tipos básicos e objetos até classes de alto nível usadas para segurança, rede, GUI, acesso a bases de dados...

– Java Platform, Enterprise Edition (Java EE) • É construída a partir da Java SE. Provê uma API e um ambiente de desenvolvimento para larga escala, escalável, fidedigno, com múltiplas camadas.

– Java Platform, Micro Edition (Java ME) • Provê uma API e um ambiente de desenvolvimento para dispositivos móveis.

– Java FX • Provê um ambiente para desenvolvimento de aplicações escritas em Java FX Script.

Aplicação Web

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

“Serviços Web são aplicações modulares autodescritas e autocontidas, que podem ser conectadas e acopladas a outros web services.” [IBM]

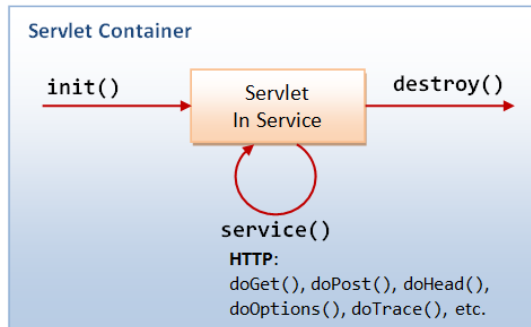
“Serviços Web são componentes de software com baixo fator de acoplamento, utilizados por meio de padrões de tecnologia Internet. Um Serviço Web representa uma função de negócio ou um serviço que pode ser acessado por uma outra aplicação.” [Gartner]

Aplicação Web

Na plataforma Java EE, componentes web fornecem funcionalidades dinâmicas para um servidor web.

- Componentes web são:

- Java Servlets



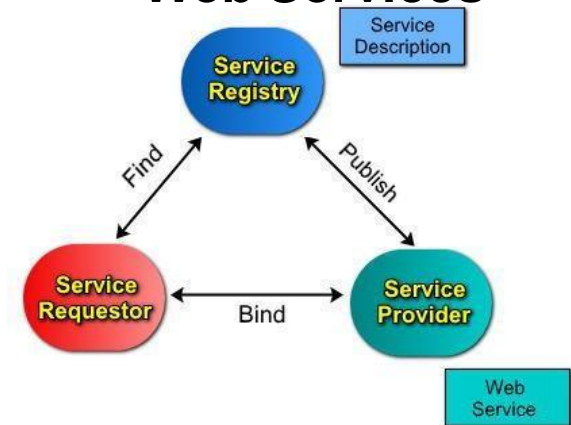
- JSF



- Páginas JSP

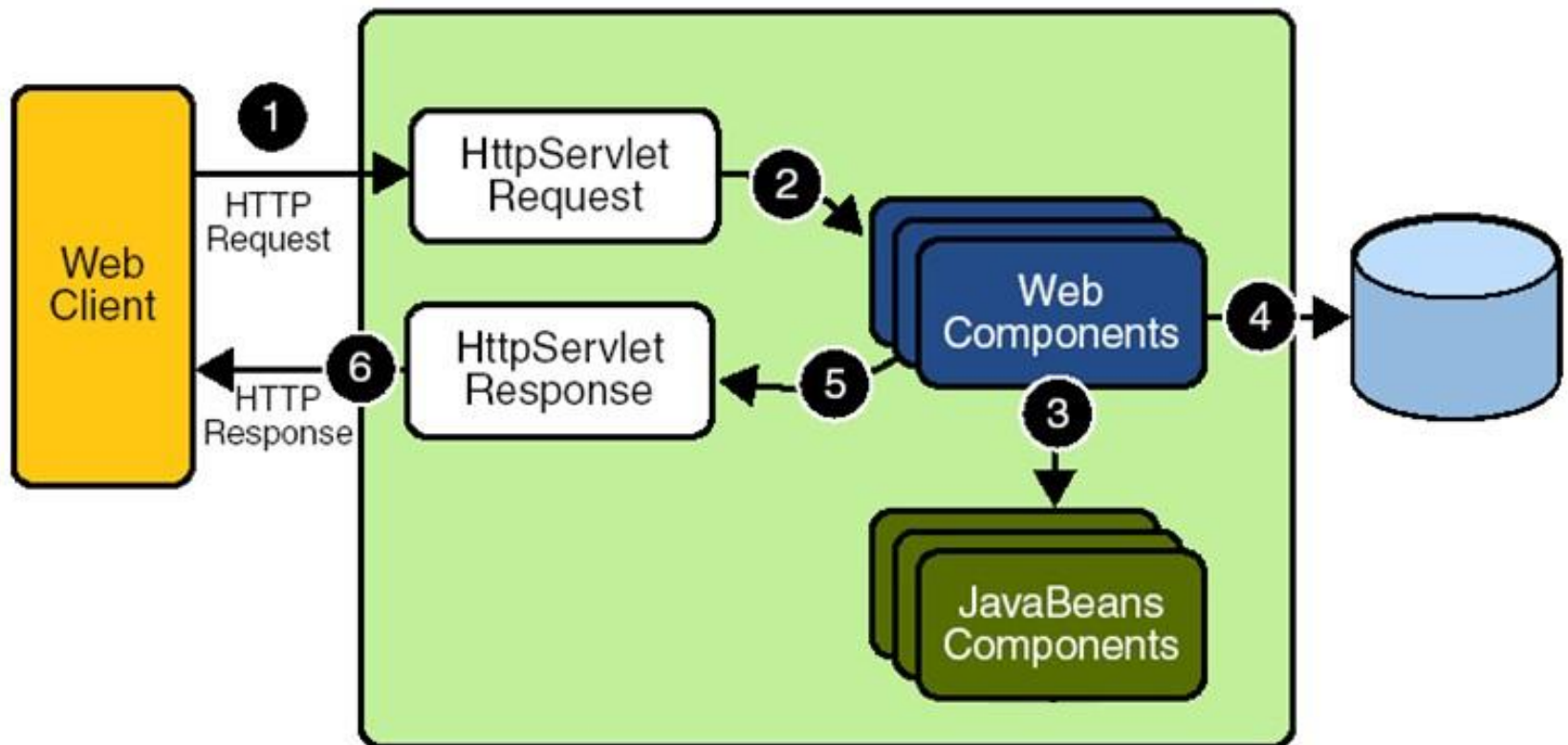


- Web Services



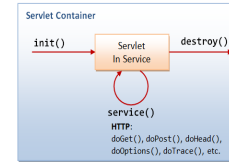
Aplicação Web

Servidor de Aplicação



Aplicação Web

- Java Servlets



- **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).**

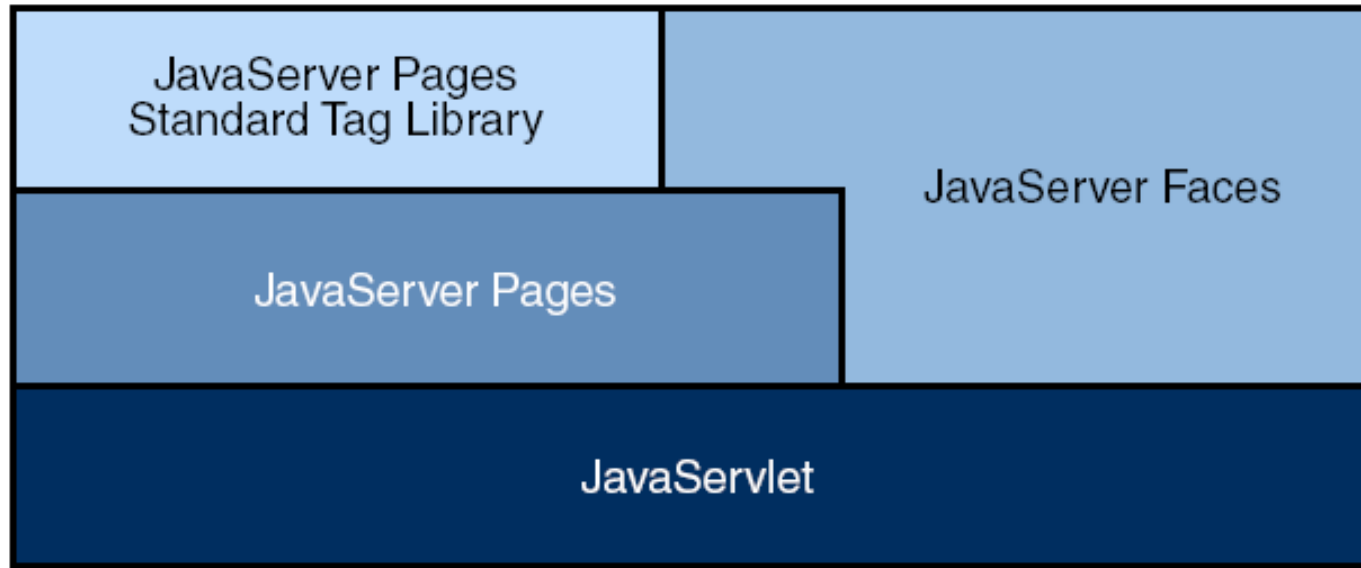
- Páginas JSP



- **Páginas JSP**

- **Documentos texto que são executados como servlets**
- **Recomendadas para páginas baseadas em HTML**

Aplicação Web



- **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**

Aplicação Web

- **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**
 - **etc**

Bibliografia

- http://www.les.inf.puc-rio.br/wiki/images/b/bf/Javall.1_-_Containers.pdf
- <http://tomcat.apache.org/index.html>
- https://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- Décio Jr. PROGRAMAÇÃO JAVA PARA A WEB. 2.ed. São Paulo:NOVATEC, 2016. 680p.



Copyright © 2017 Prof. Gustavo Molina

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