

# **Introdução ao Desenvolvimento *Web* em Java**

# Licença para uso e distribuição

Este material está disponível para uso não-comercial e pode ser derivado e/ou distribuído, desde que utilizando uma licença equivalente.



Atribuição-Uso Não-Comercial-  
Compatilhamento pela mesma  
licença, versão 2.5

<http://creativecommons.org/licenses/by-nc-sa/2.5/deed.pt>

Você pode copiar, distribuir, exibir e executar a obra, além de criar obras derivadas, sob as seguintes condições: (a) você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante; (b) você não pode utilizar esta obra com finalidades comerciais; (c) Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.



# Sobre o autor – Vítor Souza

- Formação:

- Graduação em Ciência da Computação, mestrado em Engenharia de Software, pela UFES.

- Java:

- Desenvolvedor Java desde 1999;
  - Especialista em desenvolvimento *Web*;
  - Co-fundador do ESJUG.

- Profissional:

- Pesquisador pela Università degli Studi di Trento;
  - Engenho de Software Consultoria e Desenvolvimento.

- Contato: [vitorsouza@gmail.com](mailto:vitorsouza@gmail.com)

# Agenda

- Introdução ao desenvolvimento *Web*;
- Evolução do desenvolvimento *Web* em Java;
- Servlets, JSP e frameworks MVC na prática;
- O padrão JSF e o futuro do desenvolvimento *Web*;
- Características do desenvolvimento *Web* em Java;
- Para saber mais...



## **Histórico do Desenvolvimento *Web***

Grupo de Usuários de Java do Estado do Espírito Santo

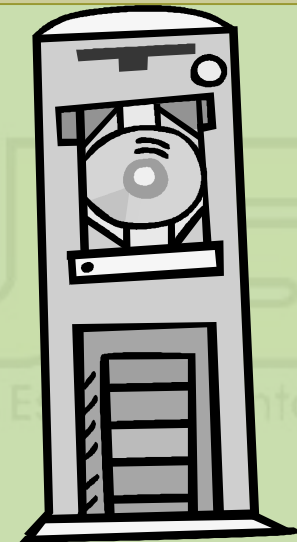
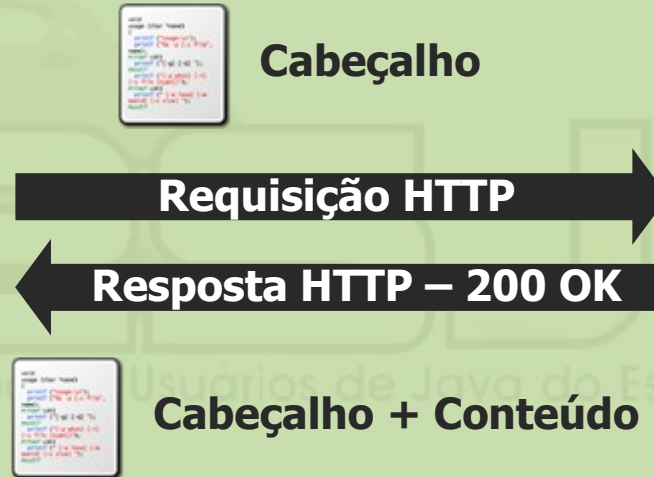
# Histórico

- 1993: CGI – *Common Gateway Interface* – C, C++, Fortran, Perl, etc.;
- 1994: Macromedia Coldfusion, PHP;
- 1995: Microsoft ASP;
- 1996: Java Servlets;
- 1999: JavaServer Pages (JSP);
- ...
- Hoje: AJAX, RoR, SOA, Flash ActionScript, etc.

# Servidor de páginas estáticas

```
GET /index.html HTTP/1.0  
Host: www.site.com  
[...]
```

1. Procura arquivo /index.html;
2. Arquivo existe – OK (200);
3. Monta resposta HTTP com conteúdo do arquivo.



**www.site.com**

```
HTTP/1.1 200 OK  
Date: Fri, 15 Apr 2005 22:12:30 GMT  
Server: Apache/1.3.26 [...]  
Last-Modified: Wed, 23 Mar 2005 00:43:22 GMT  
Content-Length: 11379  
Content-Type: text/html  
[...]
```

# Servidor de páginas dinâmicas

```
GET /index.asp HTTP/1.0  
Host: www.site.com  
[...]
```



**Requisição HTTP**

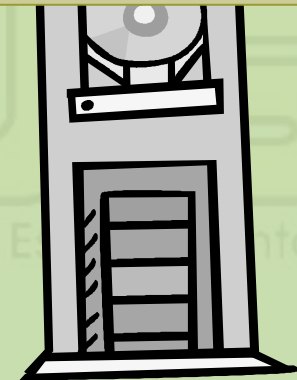
**Resposta HTTP – 200 OK**

**Cabeçalho + Conteúdo**



1. Procura arquivo /index.asp;
2. Interpreta script do arquivo;
3. Arquivo existe e não houve erros – OK (200);
4. Monta resposta HTTP mesclando conteúdo estático e dinâmico (gerado pelo script).

```
HTTP/1.1 200 OK  
Date: Fri, 15 Apr 2005 22:12:30 GMT  
Server: Apache/1.3.26 [...]  
Last-Modified: Wed, 23 Mar 2005 00:43:22 GMT  
Content-Length: 11379  
Content-Type: text/html  
[...]
```



**www.site.com**



**Criação de interfaces (páginas) que dão acesso a funcionalidade dinâmica (software) em um servidor acessível na Internet ou numa Intranet por meio de interação requisição – resposta.**



## **Evolução do Desenvolvimento *Web* em Java**

Grupo de Usuários de Java do Estado do Espírito Santo

# Java é uma linguagem...

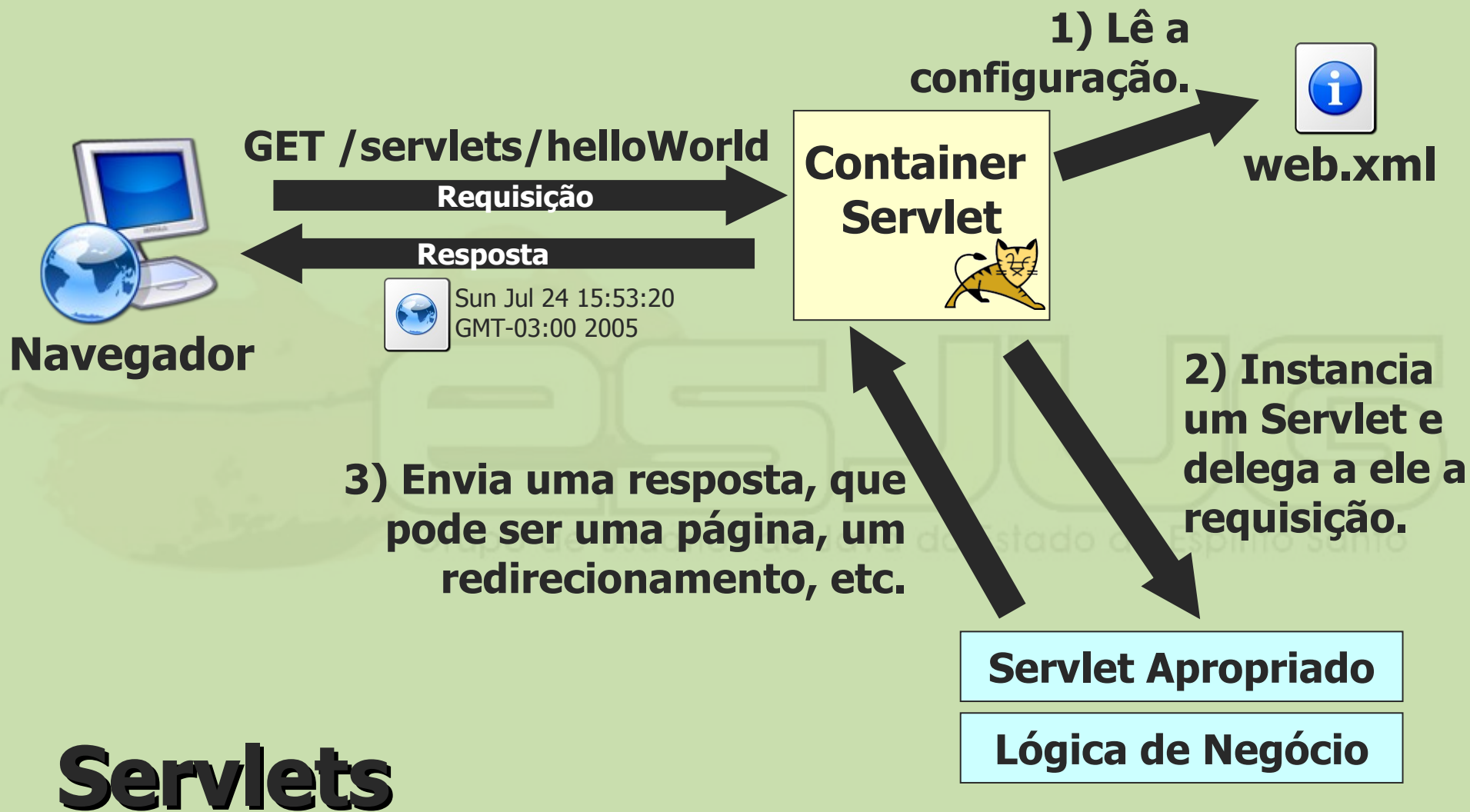
- Simples, orientada a objetos;
- Híbrida (compilada e interpretada);
- Semelhante a C / C++;
- Independente de plataforma (WORA);
- Portável, baseada no poder das redes;
- Versátil, robusta, segura e open source;
- Fácil de usar e desenvolver;
- Popular e rodeada por uma comunidade vibrante!



# Histórico

- 1995: lançamento oficial, no início eram as Applets;
- 1996: Java Servlets;
- 1997: Swing/JFC;
- 1999: JavaServer Pages (JSP);
- 1999: plataforma J2EE;
- 2004: JavaServer Faces;
- 2005: Java completa 10 anos – de ponta a ponta;
- 2006: Java EE 5.

# Evolução da plataforma Web Java





# Container Servlet

- Servidor *Web* Java;
- *Container* = gerenciador de objetos com ciclo de vida específico;
- Tem parte das funcionalidades de um Servidor de Aplicações J2EE;
- Ex.: Tomcat, Jetty, Resin, WebLogic, Oracle AS, WebSphere, JBoss, etc.

# Especificação

- Java é um conjunto de especificações definidas pelo JCP;
- JSR 53 = Servlet 2.3 e JSP 1.2;
- JSR 152 = JSP 2.0;
- JSR 154 = Servlet 2.4;
- JSR 245 = JSP 2.1;
- JSR 315 = Servlet 3.0;
- Os containers implementam as especificações.

# Servlets

- Não necessariamente para a *Web*;
- Na *Web*: classes que implementam `javax.servlet.http.HttpServlet`;
  - `service()`, `doGet()`, `doPost()`, etc.;
  - `init()`, `destroy()`;
  - `HttpServletRequest/Response`;
  - `response`: `setContentType()`, `getWriter()`.

# Aplicação Web

- Conjunto de classes e páginas que formam um sistema (software) cuja interface é acessível via navegador;

- Estrutura:

- aplicacaoWeb/

- WEB-INF/

- classes/

- lib/

- web.xml

- index.html

- [...]

**Recursos (classes e demais arquivos).**

**Recursos empacotados em JARs.**

**Configurações.**

# Mapeamento de Servlets

- Navegadores acessam endereços *Web* (URLs);
- Servlets são classes em pacotes;
- Arquivo `web.xml` faz o mapeamento URL -> Servlet.

ESJUG  
Grupo de Usuários de Java do Estado do Espírito Santo



# Servlet Container

```
GET /servlets/cadCli HTTP/1.0
Host: www.site.com
[...]
```

1. Verifica nas configurações se há um servlet para /servlets/cadCli;
2. Inicializa e executa o servlet;
3. Se existe um servlet mapeado e não houve erros – OK (200);
4. Retorna a saída gerada pelo Servlet.



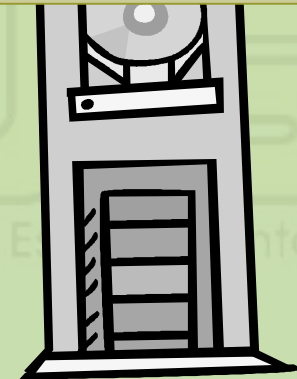
**Requisição HTTP**

**Resposta HTTP – 200 OK**

**Cabeçalho + Conteúdo**



```
HTTP/1.1 200 OK
Date: Fri, 15 Apr 2005 22:12:30 GMT
Server: Apache/1.3.26 [...]
Last-Modified: Wed, 23 Mar 2005 00:43:22 GMT
Content-Length: 11379
Content-Type: text/html
[...]
```



**www.site.com**

# Exemplo – Servlet

```
package exemplo;  
  
public class HelloServlet extends HttpServlet {  
  
    protected void doGet(HttpServletRequest request,  
        HttpServletResponse response) throws  
        ServletException, IOException {  
        doService(request, response);  
    }  
  
    protected void doPost(HttpServletRequest request,  
        HttpServletResponse response) throws  
        ServletException, IOException {  
        doService(request, response);  
    }  
}
```

## Exemplo – Servlet (cont.)

```
private void doService(...) throws ... {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    String param = request.getParameter("nome");  
    String nome = ((param == null) ||  
(param.length() == 0)) ? "estranho" : param;  
  
    out.write("<html><head><title>HelloServlet  
</title></head>");  
    out.write("<body><h1>Olá, " + nome +  
"</h1></body></html>");  
    out.close();  
}  
}
```

# Exemplo – Página inicial

```
<html>
<head>
  <title>Exemplo Servlets</title>
</head>

<body>
<h1>Qual é o seu nome?</h1>

<form action="servlets/hello" method="post">
<input type="text" name="nome" size="20" />
<input type="submit" value="OK" />
</form>
</body>
</html>
```

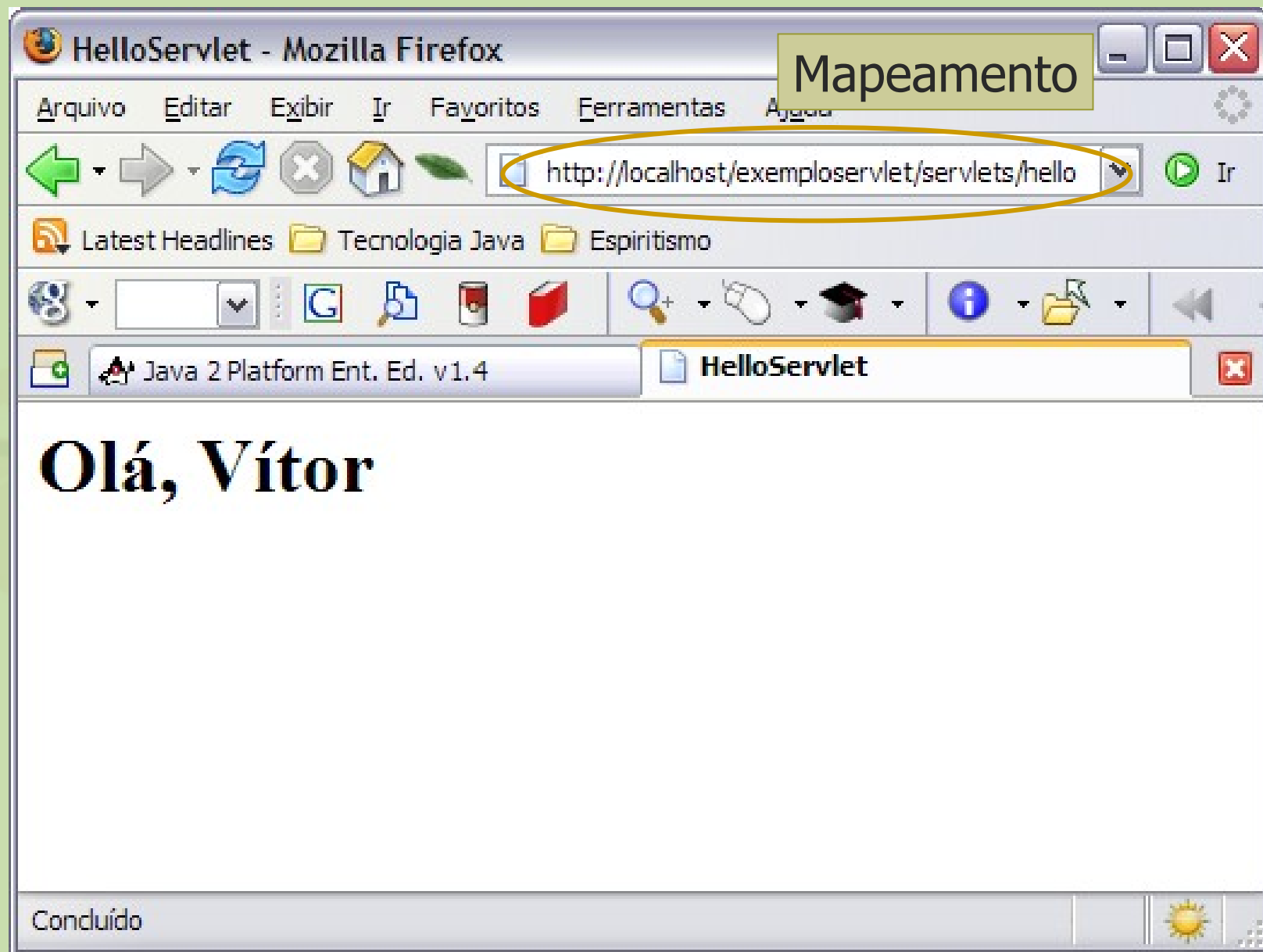
# Exemplo – web.xml

```
<web-app [...]>
  <servlet>
    <servlet-name>helloServlet</servlet-name>
    <servlet-class>exemplo.HelloServlet</servlet-
class>
  </servlet>
  <servlet-mapping>
    <servlet-name>helloServlet</servlet-name>
    <url-pattern>/servlets/hello</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```



# Exemplo – Screenshot



# Desvantagens dos Servlets

- Escrever HTML como Strings para passar como parâmetro para `out.println()`;
- Adicionar mapeamento para cada Servlet.

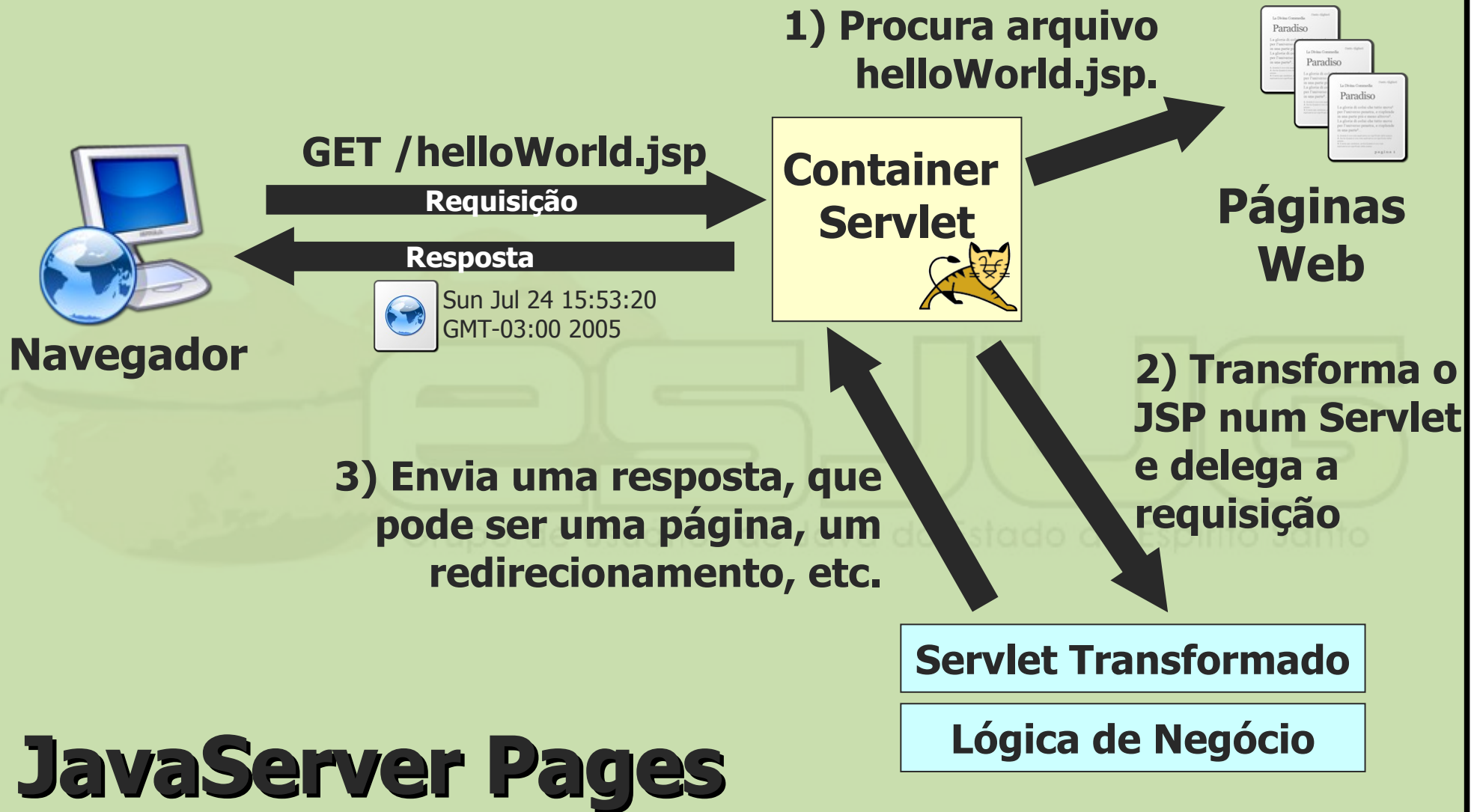


# JavaServer Pages

- São páginas HTML com código Java embutido, como PHP e ASP;
- O container “compila” as página JSPs, transformando-as em Servlets e mapeando-as automaticamente.

ESJUG  
Grupo de Usuários de Java do Estado do Espírito Santo

# Evolução de Java para a Web



# Exemplo – Página JSP

```
<html><head><title>Hello JSP</title></head>

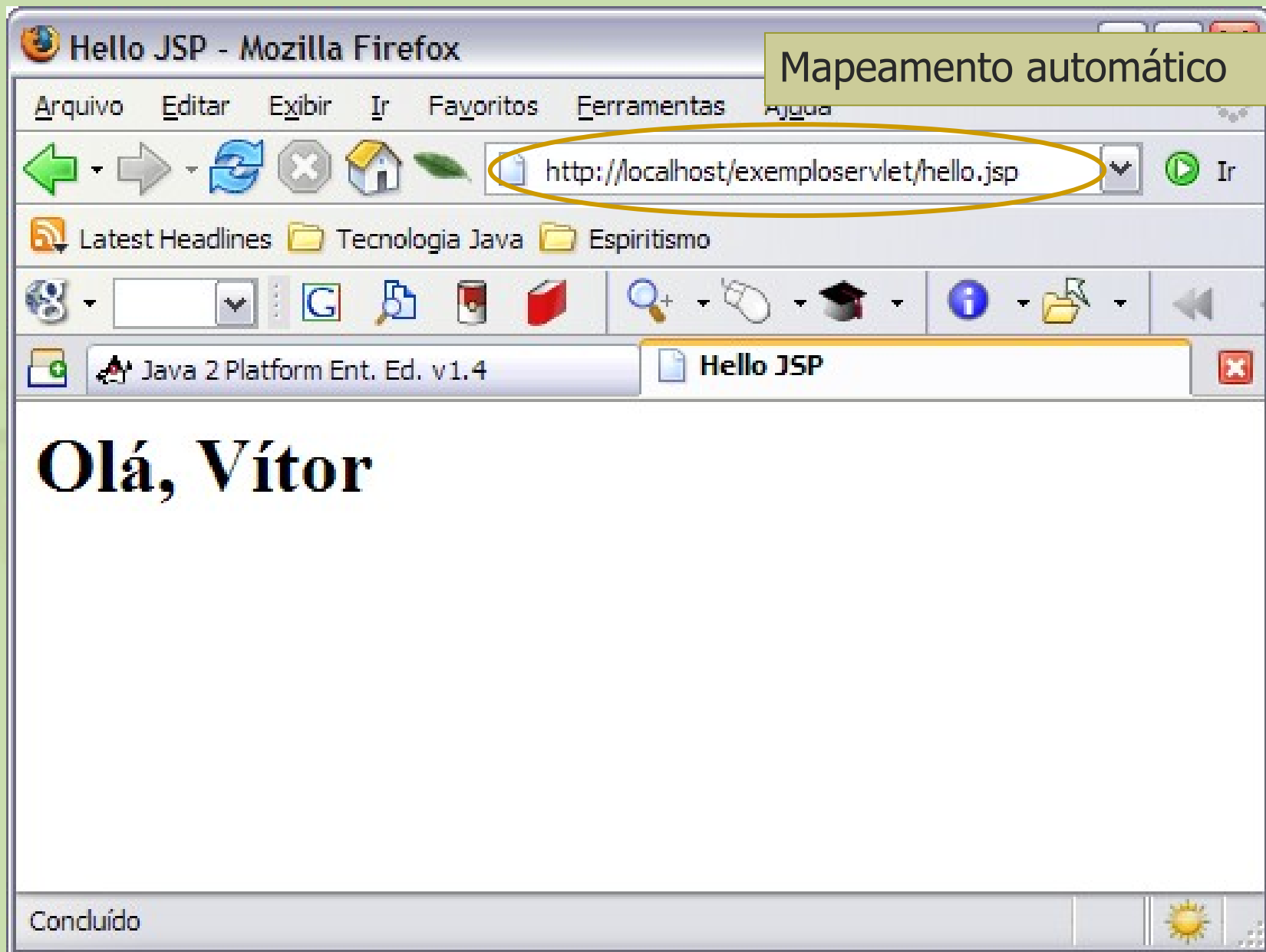
<body>
<%
String paramNome = request.getParameter("nome");
String nome = ((paramNome == null) ||
    (paramNome.length() == 0)) ? "estranho" :
    paramNome;
%>

<h1>Olá, <%= nome %></h1>

</body>
</html>
```



# Exemplo – Screenshot



# JSP não é lugar de lógica de negócio

```
<html>[...]
```

```
<%
```

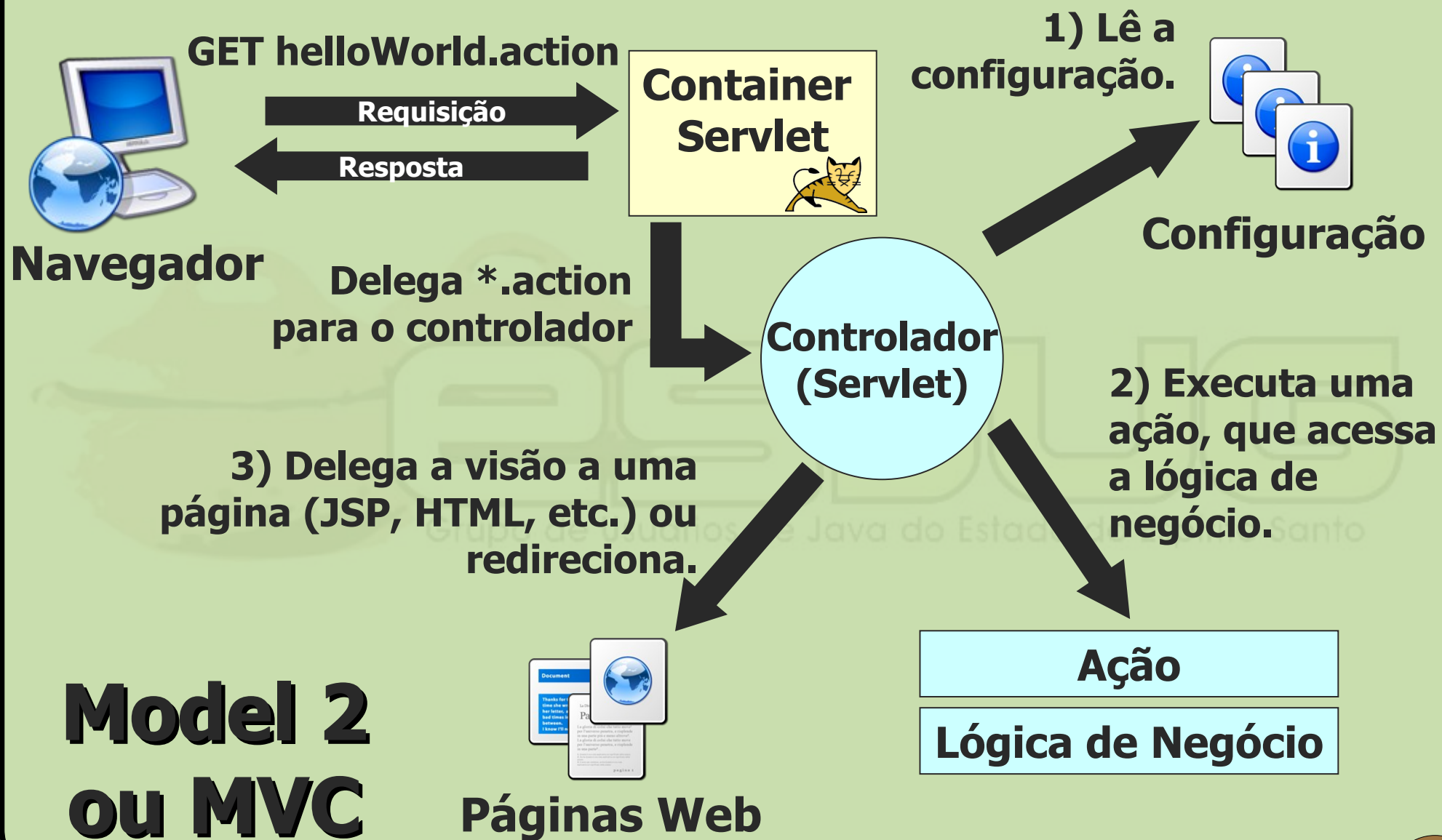
```
Connection conn; PreparedStatement stmt;  
conn = BancoDados.conectar();  
stmt = conn.prepareStatement("SQL");  
ResultSet rs = stmt.executeQuery();
```

```
// [...] Lógica de negócio.
```

```
stmt = conn.prepareStatement("SQL");  
stmt.executeUpdate();  
%>
```

```
[...]</html>
```

# Evolução de Java para a Web



# Separação de interesses

- Páginas *Web* (JSP, HTML, etc.) cuidam da parte visual;
- Servlet central faz o controle mediante configuração;
- Ações manipulam classes de lógica de negócio (modelo).

Grupo de Usuários de Java do Estado do Espírito Santo

# Nascem os *frameworks* MVC

- O nome correto seria "*Front Controller*";
- A lógica do MVC é altamente generalizável;
- Podemos listar mais de 50 *frameworks* diferentes:

Action Framework, Barracuda, Bento, Bishop, Cameleon, Canyamo, Cassandra, Chiba, Cocoon, Dinamica, Dovetail, Echo, Espresso, Folium, Genie, Helma, Jacquard, Jaffa, Japple, JATO, JBanana, Jeenius, JFormular, JPublish, jStatemachine, Jucas, JWAA, JWarp, jZonic, Macaw, Maverick, Melati, Mentawai, Millstone, MyFaces, Nacho, Niggle, OpenEmcee, OXF, RIFE, Scope, Shocks, Smile, SOFIA, Spring MVC, Struts, Tapestry, TeaServlet, Turbine, Verge, VRaptor, Warfare, WebOnSwing, WebWork, wingS, Xoplon

Fonte: <http://www.manageability.org/blog/stuff/how-many-java-web-frameworks>

# Destques

- Struts Classic:
  - Padrão “de facto”, mais antigo e usado no mercado, também mais odiado pelos desenvolvedores.
- WebWork / Struts 2:
  - Junção dos projetos Struts com WebWork, bastante aceito pela comunidade.
- Spring MVC:
  - Parte do Spring Framework.
- VRaptor2:
  - Projeto brasileiro, uso de anotações para diminuir a quantidade de configuração XML.

## Exemplo – Formulário

```
<s:form namespace="/exemplos"
  action="calcularIdade" method="post">

  <s:textfield label="Nome" name="nome" />

  <s:textfield label="Data de Nascimento"
    name="dataNascimento" />

  <s:submit value="Calcular Idade" />

</s:form>
```



## Exemplo – Classe de Ação

```
public class CalcularIdadeAction extends
                                   ActionSupport {
    private String nome;
    private Date dataNascimento;
    private int idade;

    // + getters and setters

    public String execute() throws Exception {
        idade = calcular(dataNascimento,
                           new Date());
        return SUCCESS;
    }
}
```

## Exemplo – Resultado

`<!-- Versão JSP: -->`

```
<p>Caro(a) <s:property value="nome" />, você  
possui  
<s:property value="idade" /> ano(s).</p>
```

`<!-- Versão FreeMarker: -->`

```
<p>Caro(a) ${nome}, você possui ${idade}  
ano(s).</p>
```



## Instalação do Container e Implantação das Aplicações

# Tomcat

- <http://tomcat.apache.org/>;
- Versão atual: 6.0;
- Apache Software Foundation;
- Apache Software License.



Grupo de Usuários de Java do Estado do Espírito Santo

# Tomcat - instalação

- É necessária uma JVM instalada;
- No Windows: instalador executável;
- No Linux: basta descompactar o arquivo `tar.gz` ou instalar o pacote (`rpm`, `deb`, `tgz`) da sua distribuição;
- `mod_jk`: integra o Tomcat com o Apache já instalado (opcional).

# Tomcat – estrutura

- `$CATALINA_HOME/`
  - `bin/` – scripts de startup/shutdown;
  - `conf/` – arquivos de configuração;
  - `logs/` – arquivos de log;
  - `shared/` – classes e JARs compartilhados;
  - `webapps/` – aplicações web.

Grupo de Usuários de Java do Estado do Espírito Santo

# Tomcat – implantação (*deploy*)

- Duas opções:
  - Copiar a aplicação *Web* para a pasta `$CATALINA_HOME/webapps`;
    - Pode ser compactada num arquivo WAR;
  - Adicionar a declaração da aplicação *Web* em `conf/server.xml`.

Grupo de Usuários de Java do Estado do Espírito Santo



# server.xml

```
<Server [...]>  
  <Service name="Catalina">  
    <Connector port="80" [...] />  
    <Engine name="Catalina" defaultHost="localhost">  
      <Host name="localhost" appBase="webapps" [...]>  
        <Context path="/exemploservlet"  
docBase="D:\Workspace\ExemploServlets\webapp"  
debug="999" reloadable="true" />  
      </Host>  
    </Engine>  
  </Service>  
</Server>
```

Porta de conexão com o servidor.

URL no servidor e diretório dos arquivos no disco.

# Tomcat – recarga automática

- Configure-o para recarregar alterações automaticamente:
  - Arquivo `conf/context.xml`;
  - `<Context reloadable="true">`.
- Perde um pouco no desempenho, portanto não use no servidor de produção.

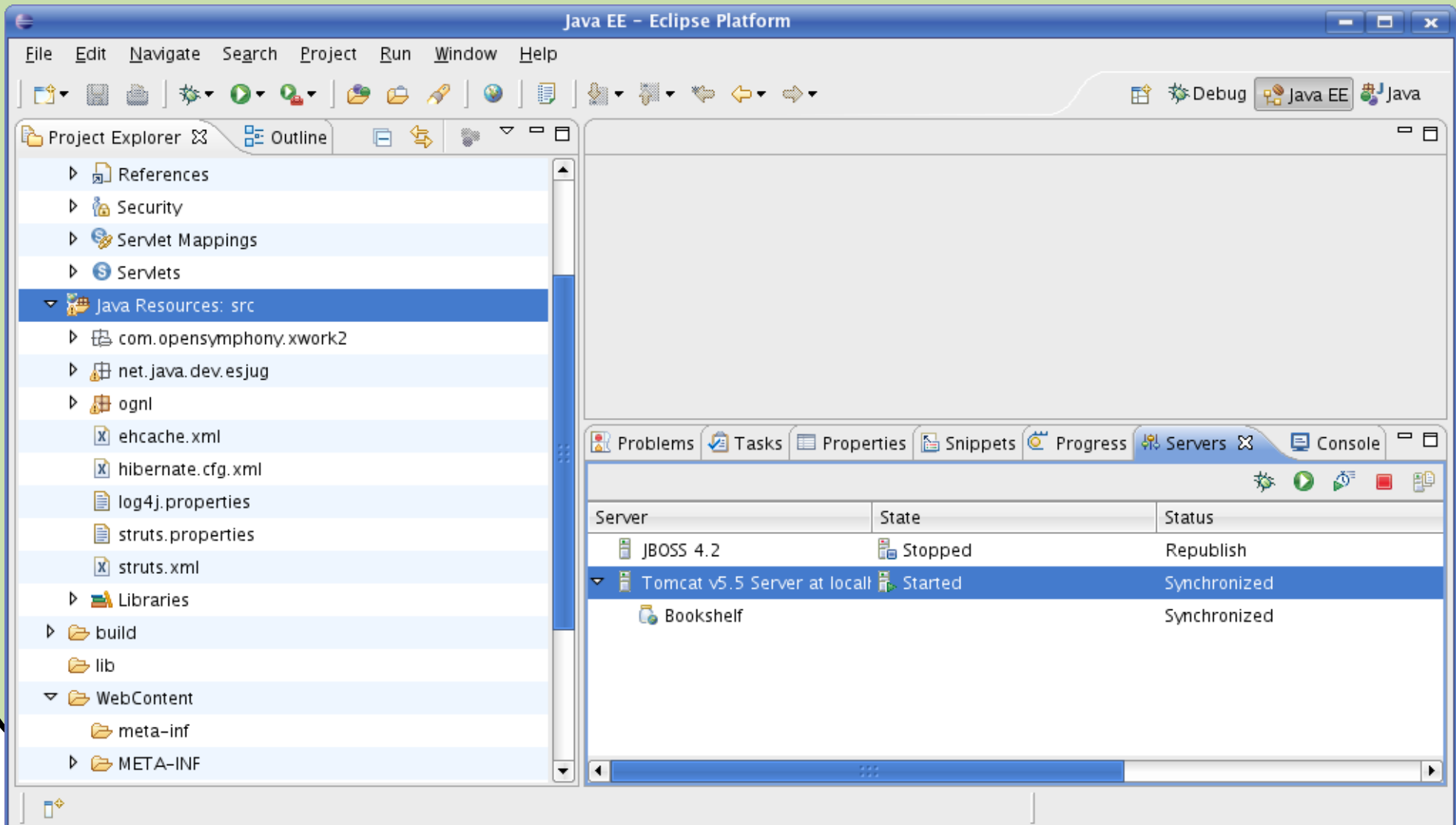
Grupo de Usuários de Java do Estado do Espírito Santo

# Tomcat – Execução

- No Windows 2000/XP:
  - É instalado um serviço e um monitor;
- Outros Windows:
  - `CATALINA_HOME\bin\startup.bat`
- No Linux:
  - `CATALINA_HOME/bin/startup.sh`
  - `tail -f CATALINA_HOME/logs/stdout.log`
  - Ou pode ser iniciado como daemon.

# Uso de IDEs

- Ex.: Eclipse Web Tools
  - Criação, implantação, execução e testes integrados.





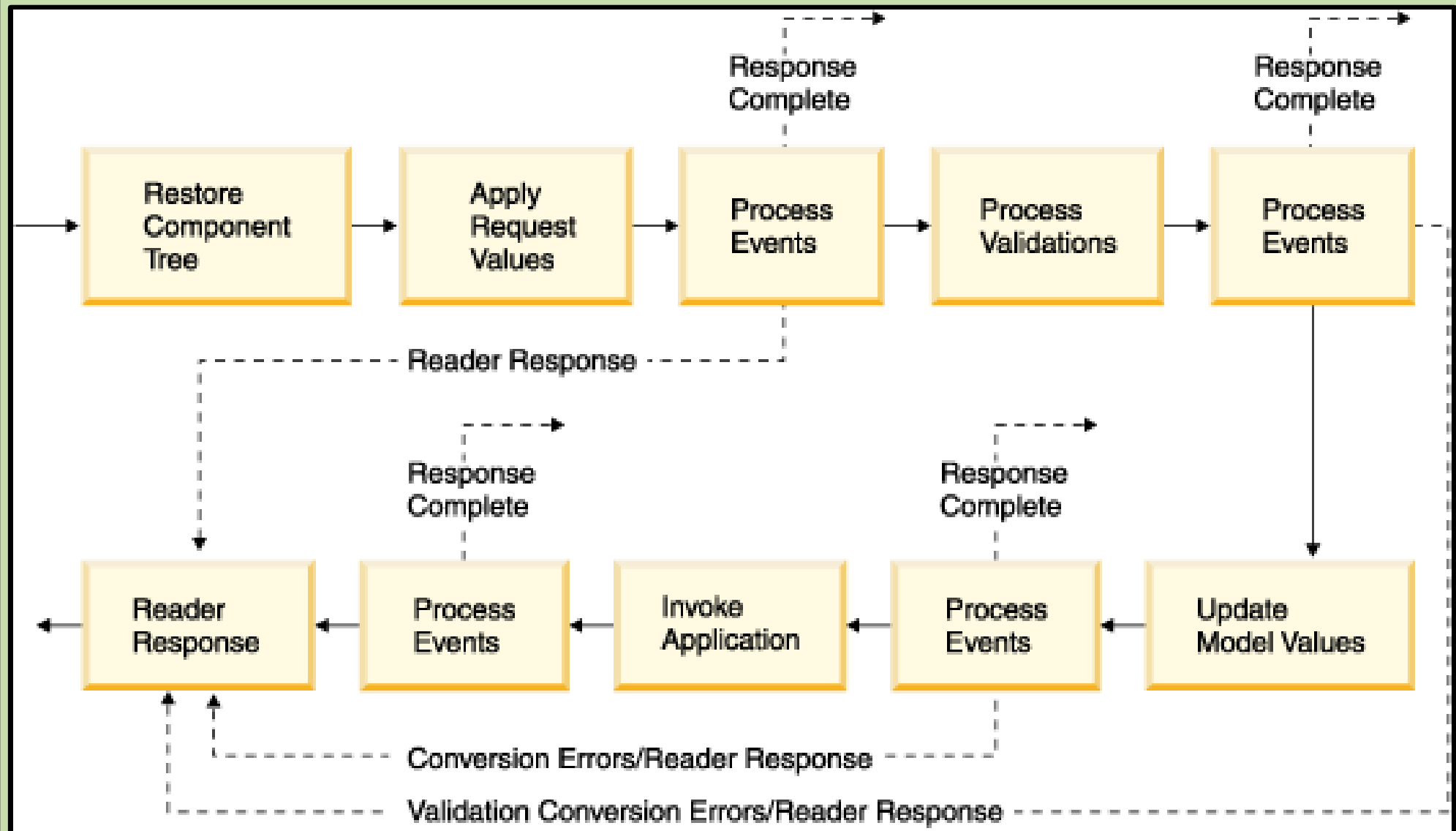
## O padrão JavaServer Faces e o que vem por aí...

Grupo de Usuários de Java do Estado do Espírito Santo

# JavaServer Faces

- JSR 127 – padrão oficial (27/05/2004);
  - Várias implementações;
  - Garantia de continuidade.
- Similar aos *frameworks* MVC;
- Foco no desenvolvedor:
  - Projetado para ser utilizado por IDEs;
  - Componentes UI extensíveis;
  - Tratamento de eventos (como no Swing!);
  - Suporte à navegação simples.

# Ciclo de vida JSF





# Suporte de IDEs

The screenshot displays the NetBeans IDE 6.0 interface. The main window shows a web application titled "CORPORATE TRAVEL CENTER" with a header and a main content area. The "Palette" on the right lists various UI components like Label, Text Field, Button, Image Hyperlink, Listbox, Checkbox, Radio Button, etc. The "PageFlow" tab is active, showing a diagram with two pages: "Page1.jsp" and "Results.jsp". "Page1.jsp" contains "viewButton" and "voteButton". "Results.jsp" contains "homeButton" and "refreshButton". Arrows indicate navigation: "home" from Results.jsp to Page1.jsp, "view results" from Page1.jsp to Results.jsp, and "vote" from Page1.jsp to Results.jsp. A "faces-config.xml" file is also visible in the project view.

NetBeans IDE 6.0

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Home x Help x Header x Details x

Design JSP Java Any Size

CORPORATE TRAVEL CENTER

Empowered by NetBeans 6

Employee: [dropdown]

Trips

Trip ID	Type	Date	Departure	Destination
123	abc	Oct 22, 2007	abc	abc

Message sum

Sys

List

Image Hyperlink

Drop Down List

Listbox

Checkbox

Checkbox Group

Radio Button

Radio Button Group

Image

Table

Table Column

Table Row Group

Password Field

Hidden Field

Calendar

File Upload

Tree

Tree Node

Anchor

Page1 \* Results \* faces-config.xml \*

PageFlow XML Project

Page1.jsp

viewButton

voteButton

view results

vote

Results.jsp

homeButton

refreshButton

home



Applications

Connections

storeFront.jsp

web-content

- images
- javadocs
- META-INF
- WEB-INF
- bottomMatter.jsp
- carDetail.jsp
- chooseLocale.jsp
- confirmChoices.jsp
- customerInfo.jsp
- finish.jsp
- index.jsp
- optionsPanel.jsp
- storeFront.jsp
- stylesheet.css

jsf-components

storeFront.jsp - Structure


- h:form
  - h:graphicImage - /images/carde
  - h:panelGrid
    - h:panelGrid
      - Jalopy
      - h:graphicImage
      - h:outputText
      - h:outputText
      - h:commandButton
      - Roadster
      - h:graphicImage
      - h:outputText
      - h:outputText
      - h:commandButton

storeFront.jsp

None Default None

A B I U


# CAR DEMO



Duke's Stripped-Down Jalopy

If you're the type who doesn't care what anyone thinks, this is the car for you. Strictly for point-a-to-point-b types.


More



Duke's Vulgar Luxury Car

Those Joneses next door have everything, don't they? Well, show them that you're the one pulling in the big bucks by driving up in this loaded chassis.


More



Duke's MLC Roadster

Getting on in years? Need to make one last-ditch grasp at youth? You'll feel 20 years younger when

More



Duke's Bloated SUV

Sometimes you can never be high enough or big enough compared to the car next to you. And those newborns come out bigger every time.

More

h:outputText

Component Palette

JSP

- Pointer
- Declaration
- Expression
- Fallback
- Forward
- GetProperty
- Include
- Include
- Page
- Param
- Params
- Plugin

Graphic Image - Pro...

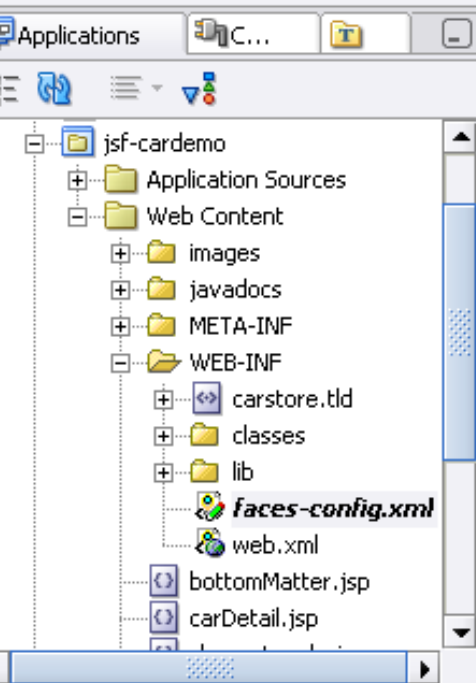
General

Url	
Alt	
Binding	#{carstore.mo...
Height	
IsMap	false
LongDesc	
Rendered	true
UseMap	
Value	
Width	

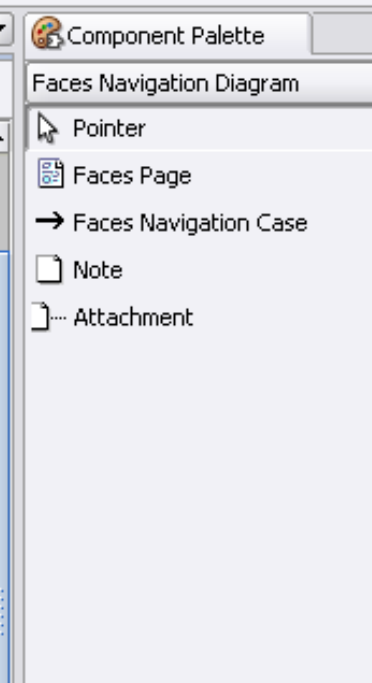
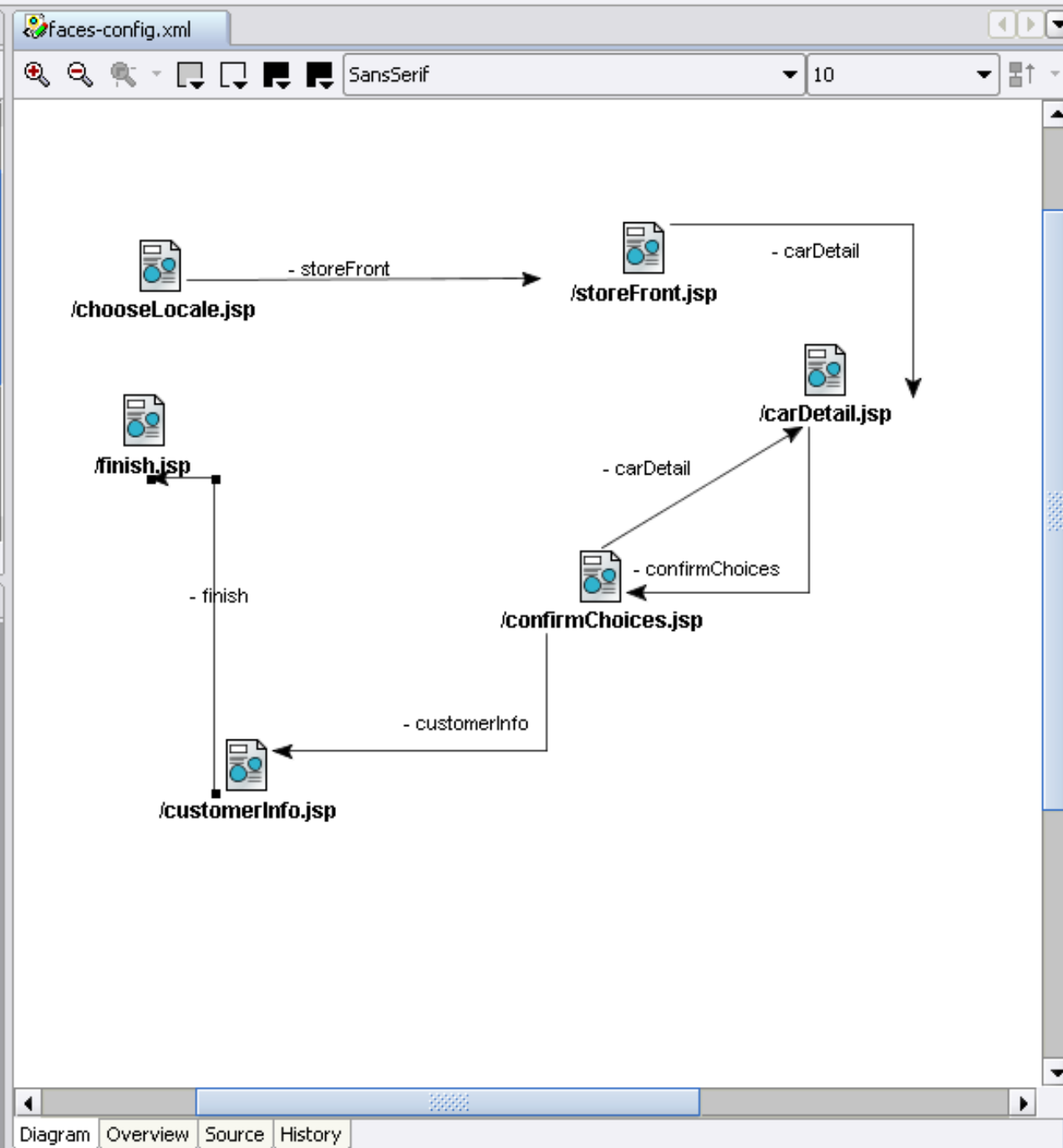
Core

Id

Reset Size



Structure Faces-config...



navigation-case - /fi...

General	
to-view-id *	/finish.jsp
from-action	
from-outc...	finish
redirect	false
description	Any action that
display-n...	
icon	

# Futuro... ?

- JSR 299 – Web Beans;
- Unificação dos modelos EJB 3 e JSF 2;
  - EJB 3 traz idéias bem-sucedidas: ORM, DI, etc., porém a integração com JSF ainda é trabalhosa e tediosa.
- Web Beans unifica os modelos de componentes;
- Antecipe-se: conheça o JBoss Seam:
  - O criador do Seam é Spec Lead do Web Beans.

# JBoss Seam

- Informações gerais:
  - Criado por Gavin King (criador do Hibernate);
  - Versão 1.0 lançada em 11/06/2006;
  - Versão atual: 2.0;
  - Website: <http://www.seamframework.org/>

ESJUG  
Grupo de Usuários de Java do Estado do Espírito Santo



# Visão geral

- Integração JSF – EJB3 (modelo de componentes unificado);
- AJAX e jBPM integrados;
- Gerenciamento de estado declarativo;
- Bijection, Conversation e Workspaces;
- Utilização de POJOs com anotações;
- Testabilidade;
- I18n, autenticação, depuração, URLs RESTful, seam-gen, eventos, interceptadores, etc.

# Criando uma entidade persistente

```
@Entity
public class BlogEntry {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Version @Column(nullable = false)
    private Long version;

    @Basic @Column(length = 50)
    private String title;

    @Temporal(TemporalType.TIMESTAMP)
    private Date time;

    @Lob
    private String text;
    ...
}
```

# Criando um DAO

```
@AutoCreate
@Name("blogEntryDAO")
@Scope(ScopeType.APPLICATION)
@Stateless
public class SeamBlogEntryDAO implements LogEntryDAO
{
    @PersistenceContext
    private EntityManager entityManager;

    @Transactional
    public Collection<T> retrieveAll() {
        Query query = entityManager.createQuery(
                                                    "from BlogEntry");
        return query.getResultList();
    }
    ...
}
```



# Criando um serviço

```
@Local
public interface ManageBlogService { ... }

@AutoCreate
@Name("manageBlogService")
@Scope(ScopeType.APPLICATION)
@Stateless
public class ManageBlogServiceImpl implements
ManageBlogService {

    @In
    private BlogEntryDAO blogEntryDAO; // + setter

    ...
}
```

# Criando uma ação

```
@Name("manageBlogAction")
@Scope(ScopeType.SESSION)
public class ManageBlogAction {

    @In
    private ManageBlogService manageBlogService;

    private BlogEntry entry = new BlogEntry();

    ...

    public void save() {
        manageBlogService.addBlogEntry(entry);
    }
}
```

## Se referindo à ação nas páginas JSF

```
<h:form>
<h:panelGrid columns="2">
<h:outputText value="#{messages['form.title']}" />
<h:inputText
value="#{manageBlogAction.entry.title}" />

<h:outputText value="#{messages['form.text']}" />
<h:inputText
value="#{manageBlogAction.entry.text}" />
</h:panelGrid>

<h:commandButton type="submit"
value="#{messages['form.button.add']}"
actionListener="#{manageBlogAction.save}" />

</h:form>
```



## Vantagens e Desvantagens

Grupo de Usuários de Java do Estado do Espírito Santo

# Padrão aberto

- A especificação é definida por um conjunto de empresas e pessoas físicas (*Java Community Process*);
  - Qualquer um pode implementar a estrutura por trás;
  - Você não fica preso a uma implementação: várias empresas fornecem *containers* .

Grupo de Usuários de Java do Estado do Espírito Santo

# Java é multiplataforma

- A Sun dá suporte às seguintes plataformas:
  - Windows i586 / AMD64;
  - Linux i586 / AMD64;
  - Solaris SPARC / x86 / x64.
- A Apple desenvolve versão para MacOS.
- Outras:
  - HP UX, FreeBSD, OS/2, AIX, etc.;
  - Projeto Open JDK: <http://openjdk.java.net/>;
  - Nem todas são atuais ou completas.
- *"Write once, run anywhere".*

# Arquitetura *enterprise*

- Promove a separação das páginas *Web* e da lógica de negócio;
  - Principalmente com o uso de *frameworks* ou dos padrões JSF e, futuramente, Web Beans.
- Em algumas linguagens, colocar o código em páginas *Web* é a única opção;
- Curva de aprendizado se torna maior.

# Plataforma Java EE

- Conjunto de especificações que complementam Servlets/JSP:
  - Persistência automática;
  - Conectividade remota;
  - Serviço de nomes;
  - Acesso a sistemas legados;
  - Segurança e transações;
  - Troca de mensagens (processamento assíncrono);
  - Etc.



# Desempenho

- Ao contrário do que dizem, Java possui ótimo desempenho;
- Java é híbrida (compilada para bytecode) – mais rápida do que linguagens puramente interpretadas;
- Linguagens interpretadas tornam os testes mais rápidos (*deploy* rápido).

# Produtividade

- Se comparado a .NET, perde em produtividade devido às ferramentas da Microsoft:
  - Com a especificação JSF, muitos fornecedores de ferramentas (Oracle, Borland, etc.) começaram a correr atrás;
  - Se pagar, também terá produtividade!
- Se comparado com linguagens não-OO, ganha em produtividade devido ao reuso ser facilitado.

# Custo Baixo

- É possível desenvolver com custo zero:

Componente	Opção	Gratuito	Open Source
Máquina Virtual Java	Sun	Sim	Sim
Ambiente de Desenvolvimento	Eclipse / NetBeans	Sim	Sim
Container	Tomcat / JBoss	Sim	Sim
Sistema Operacional	Linux	Sim	Sim
Banco de Dados	MySQL / PostgreSQL	Sim	Sim
Frameworks	Vários	Sim	Sim

# Ferramentas

- Centenas de ferramentas disponíveis (muitas gratuitamente/*open-source*):
  - Persistência de objetos;
  - Geração de relatórios;
  - Funcionalidade de busca;
  - *Frameworks* de infra-estrutura;
  - IDEs e ferramentas CASE;
  - Dentre muitas outras...

# Comunidade Java

- Java possui uma comunidade vibrante e ativa para troca de experiências:
  - Milhares de projetos *open-source*;
  - Grupos de usuário Java (JUGs);
  - Portais no Brasil e no exterior;
  - Eventos.



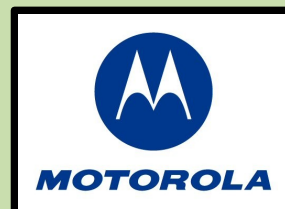
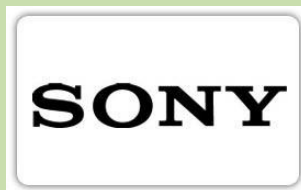
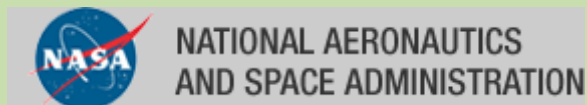
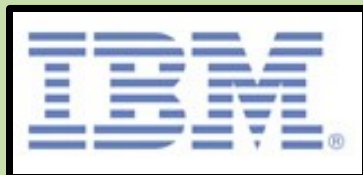
**III Encontro de Java do  
Espírito Santo  
Agosto 2008**

# Suporte comercial

- Muitas empresas investem em Java:
  - Sun Microsystems;
  - IBM;
  - Oracle;
  - Borland;
  - BEA;
  - PowerLogic;
  - Dentre muitas outras...



# Algumas empresas envolvidas









# Números do JavaOne 2005

- No mundo, Java está em 635 modelos de celular, 32 fabricantes e 140 operadoras;
- A plataforma J2EE já foi baixada mais de 3 milhões de vezes, e existem 28 servidores diferentes;
- A e-Bay confia em Java para todas suas vendas que ultrapassam US\$ 1.400,00 por segundo!
- Estimativa que Java movimentava US\$ 100 bilhões / ano;
- 100% das "Fortune 500" usam Java.

# Números do JavaOne 2007

- 6 milhões de desenvolvedores Java no mundo (15.000 no JavaOne!);
- 5,5 bilhões de aparelhos Java-powered (2 bilhões de celulares);
- Real Time Java: Nasdaq processa 5 bilhões de transações por dia usando Java;
- Blu-ray: interatividade desenvolvida em Java;
- A nova onda: JavaFX;
- JavaOne 2008: 6 a 9 de Maio!

**Para saber mais...**



# **Dicas para quem quer aprender Java, Java Servlets, JSP e JSF**

Grupo de Usuários de Java do Estado do Espírito Santo

# Para aprender Java

- Orientação a Objetos é fundamental!
- Livros:
  - Core Java (Cay Horstmann, et. al) – I e II  
<http://www.temporeal.com.br/produtos.php?id=169228>  
<http://www.temporeal.com.br/produtos.php?id=164250>
  - Thinking in Java (Bruce Eckel)  
<http://www.mindview.net/Books/TIJ/>  
<http://pensandoemjava.dev.java.net/>
  - Java 2 – Use a Cabeça  
<http://www.temporeal.com.br/produtos.php?id=169941>

# Para aprender Java

- Livros (em inglês):

- Head First Java (Kathy Sierra, et. al)

<http://www.temporeal.com.br/produtos.php?id=169004>

- Java in a Nutshell (David Flanagan)

<http://www.temporeal.com.br/produtos.php?id=169483>

- Sites:

- PortalJava – [www.portaljava.com](http://www.portaljava.com)

- GUJ – [www.guj.com.br](http://www.guj.com.br)

- JavaFree – [www.javafree.org](http://www.javafree.org)

- ESJUG – <http://esjug.dev.java.net>

# Para aprender Java

- Sites (em inglês):
  - JavaLobby – [www.javalobby.org](http://www.javalobby.org)
  - TheServerSide – [www.theserverside.com](http://www.theserverside.com)
  - JavaRanch – [www.javaranch.com](http://www.javaranch.com)
  - ON Java – [www.onjava.com](http://www.onjava.com)
  - Java.net (Sun) – [www.java.net](http://www.java.net)
  - Sun Microsystems – [www.java.sun.com](http://www.java.sun.com)

# Para aprender Servlets/JSP/JSF

## ● Livros:

- Core Servlets and JSP (Marty Hall)

<http://www.temporeal.com.br/produtos.php?id=169384>

- Head First Servlets & JSP (Bryan Basham)

<http://www.temporeal.com.br/produtos.php?id=168603>

- Core Java Server Faces (David Geary)

<http://www.temporeal.com.br/produtos.php?id=169119>

- EJB 3 em Ação (Panda, Rahman, Lane)

<http://www.temporeal.com.br/produtos.php?id=171670>

# Ferramentas

- Java: [www.java.sun.com/j2se](http://www.java.sun.com/j2se)
- IDEs (Ambientes de desenvolvimento):
  - NetBeans: [www.netbeans.org](http://www.netbeans.org)
  - Eclipse: [www.eclipse.org](http://www.eclipse.org)
- Servidores Web (Containers):
  - Tomcat: [jakarta.apache.org/tomcat](http://jakarta.apache.org/tomcat)
  - JBoss: [www.jboss.org](http://www.jboss.org)



# Junte-se à Comunidade!

- Conheça nosso site:  
<http://esjug.dev.java.net>
- Cadastre-se na lista de e-mails:  
[esjug-subscribe@esjug.dev.java.net](mailto:esjug-subscribe@esjug.dev.java.net)
- Participe de nossas atividades! Informe-se no site.



# Introdução ao Desenvolvimento Web em Java