



Struts e TagLibraries

Edgar Silva

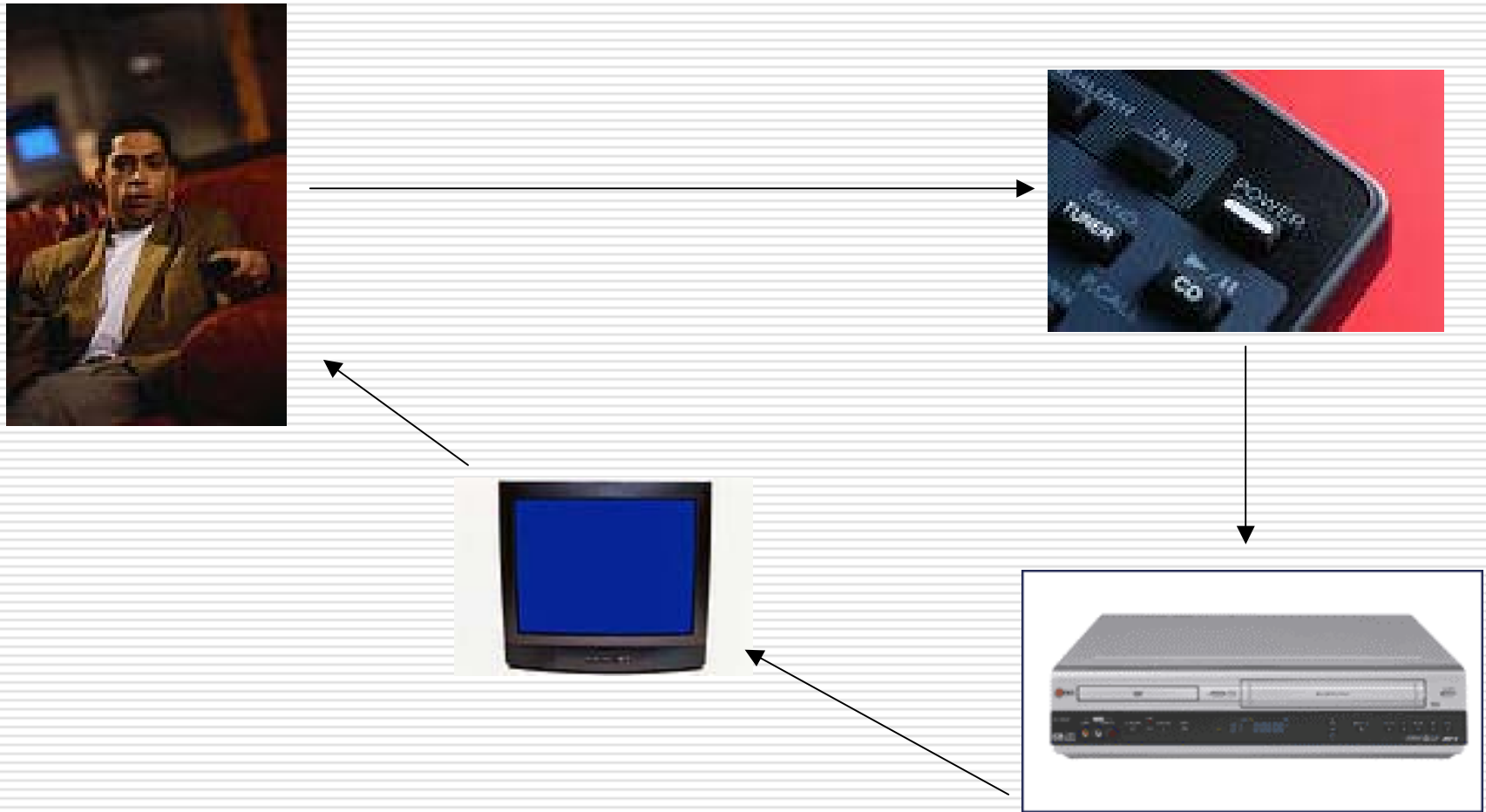
esilva@argostecnologia.com



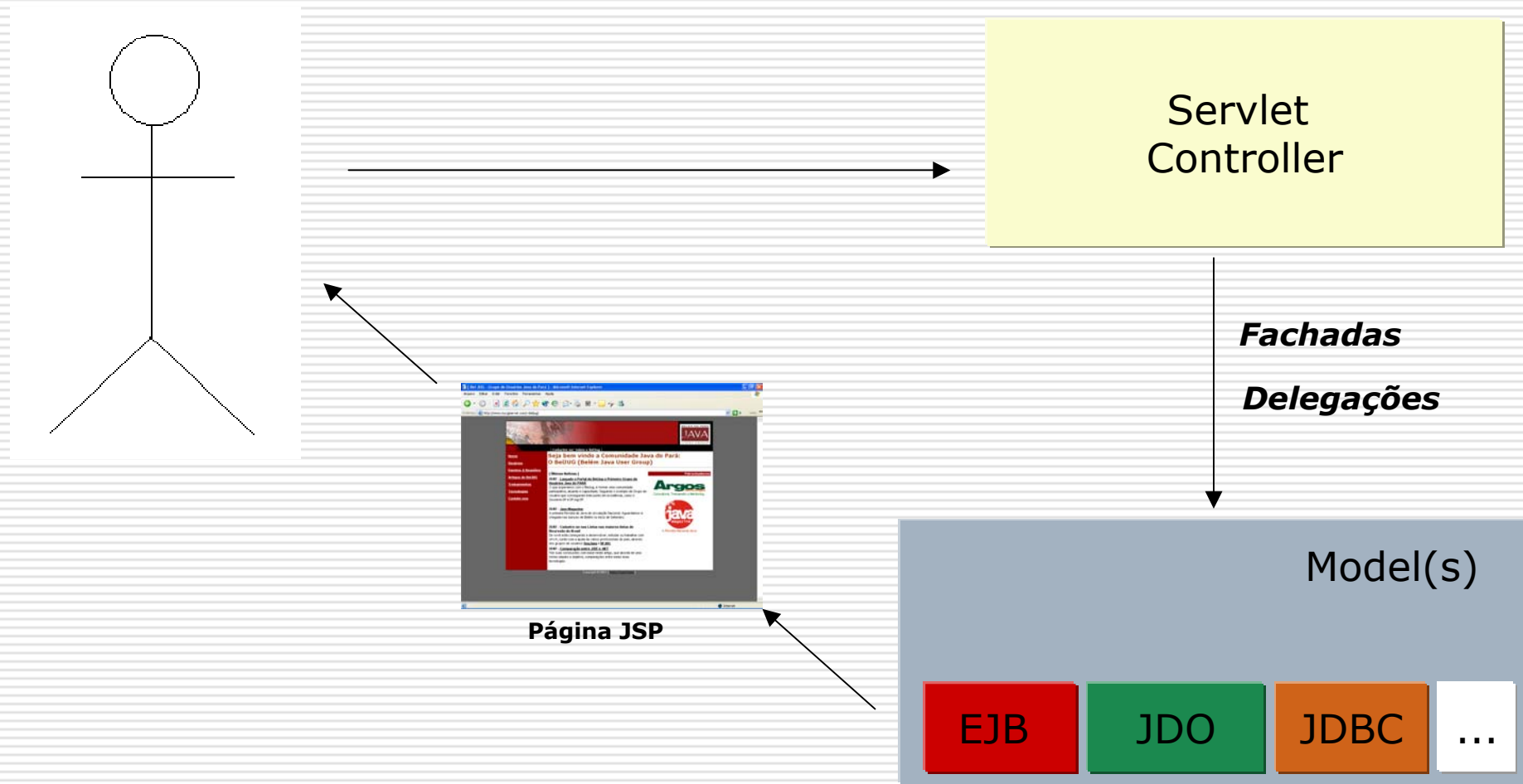
Agenda

- ❑ Arquitetura Model View Controller
- ❑ Patterns Participantes
- ❑ Entendendo o Struts
- ❑ Trabalhando com Validações
- ❑ Frameworks Adicionais
- ❑ Introdução a Bibliotecas de Tags
- ❑ Estudo de Caso: Aplicação Demo
- ❑ Realizando Deploy para JBoss

Básico de Model View Controller



Model View Controller Aplicado a Web



Papéis de Frameworks

- ❑ Frameworks = Padrões
- ❑ Padrões = Qualidade
- ❑ Qualidade = Menos riscos de falhas

Mas por quê o Struts pode ser tão importante?

- ❑ Uma das principais razões:
 - Prevenir erros de design de Aplicação, os famosos "Anti-Patterns".
- ❑ Tornou-se um padrão mundial
- ❑ Fonte de Inspiração do Java Server Faces (Sun)

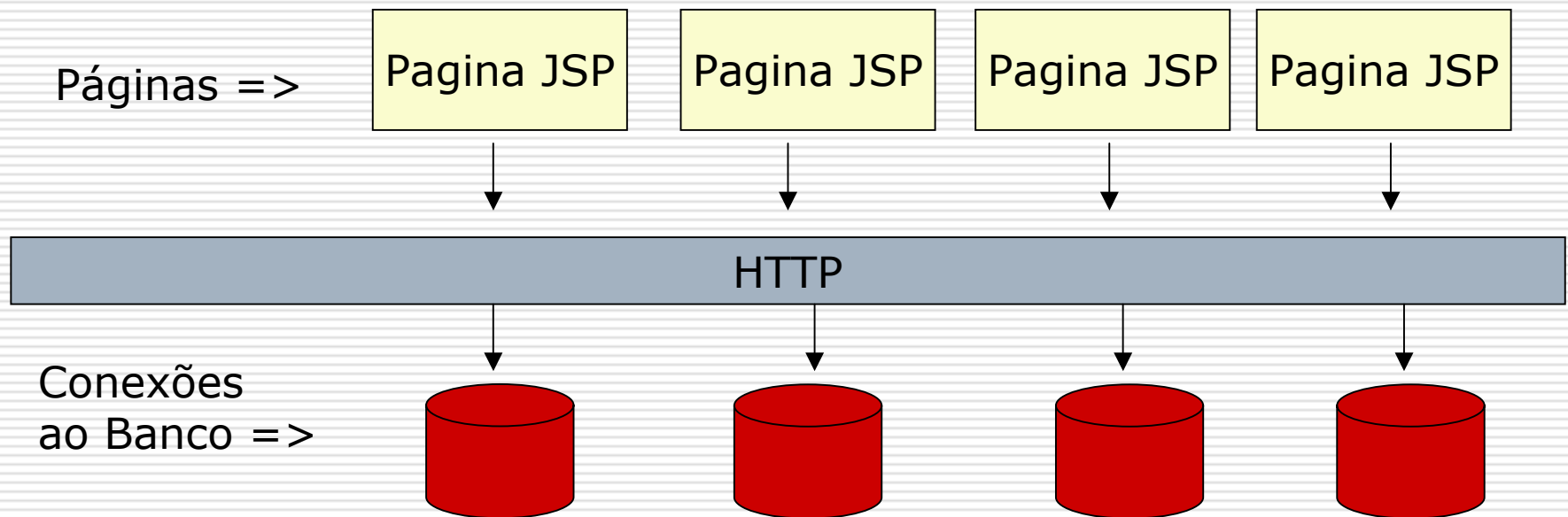
Anti-Patterns

- ❑ Tradução mais fiel para o nosso português cotidiano:

**NÃO FAÇA DESSE JEITO PARA NÃO
SE FERRAR DEPOIS ...**

[Anti-Pattern] Várias Conexões por Página.

- ❑ Uma aplicação onde cada página conecta, consulta, insere e exclui informações no banco de dados.



[Anti-Pattern] Dentro de uma Página JSP

- ❑ Lógica misturada com HTML
- ❑ Dificulta o design da programação
- ❑ Aproximação com Linguagens de Script

Solução

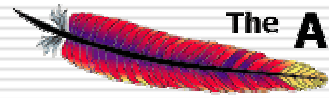
- ❑ Separar papéis de maneira simples, coesa e objetiva.
- ❑ Redefinir os papéis visando produtividade.
- ❑ Utilizar os casos de sucesso de mercado como exemplo [Design Patterns]

Patterns Participantes

em um bom projeto de camada web

- ❑ MVC (Model View Controller)
- ❑ Command
- ❑ Business Delegate
- ❑ Session Façades
- ❑ Data Transfer Objects ou Value Objects
- ❑ Interpreters
- ❑ Service to Worker
- ❑ FrontController

Jakarta Struts



The **Apache Jakarta Project**

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

Struts

Struts

□ O que é Struts

- Um framework que facilita o desenvolvimento Web com Java.
- Uma implementação fiel ao Model View Controller, bem como outros padrões.
- Você terá aplicações web padronizadas e dependendo do caso reutilizáveis.

Struts

- ❑ Várias classes utilitárias para validação, criação de máscaras e etc. Auxiliam no processo de concepção dos requisitos. **[Model]**
- ❑ Várias Bibliotecas de Tags para facilitar a renderização da Interface Gráfica. **[View]**
- ❑ Um Servlet Controlador, responsável por decidir e executar as ações (Actions), o qual é configurado por um arquivo XML externo. **[Controller]**

Elementos MVC do Struts

- ❑ Model
 - Geralmente Objetos Java (Java Beans)
- ❑ View
 - Páginas HTML e JSP
- ❑ Controller
 - `Org.apache.struts.action.ActionServlet` ou descendentes.
- ❑ Classes de Ajuda (Helpers)
 - FormBeans: dados e forms html (M)
 - Action: lógica de comandos
 - Custom Tags: apresentação (V)

E então compro agora?

Onde Assino?

- ❑ Não se compra.
- ❑ É gratuito.
- ❑ Integrante do Projeto Jakarta
- ❑ **jakarta.apache.org/struts**
- ❑ Faça o download da última versão.
 - (versao recomendada = (rc-1))

Principais Componentes do Struts

- ❑ Principais Componentes
 - ActionServlet: despacha as ações
 - ❑ FrontController Pattern
 - Action: Classe determinante da Ação
 - ❑ Command Pattern
 - Struts-Config.xml: Arquivo de configuração
 - Formulários (até mesmo dinâmicos)
 - Elementos de Validação

O que mais o Struts oferece?

- ❑ Pool de conexões de Banco de Dados
- ❑ Internacionalização de Aplicações
- ❑ Projetar uma vez...Reestruturar diversas vezes (Jakarta Tiles)
- ❑ Arquivo de mensagens
- ❑ Mecanismos de Upload
- ❑ Possibilidade de extender o Framework.

Trabalhando com Struts a primeira vez

- ❑ Faça o download
- ❑ Observe o exemplo struts-blank
- ❑ Copie o arquivo struts-blank.war para a pasta webapps dentro do TomCat.
- ❑ A aplicação estará funcionando no TomCat
 - <http://seuhost:8080/struts-blank>

Utilizando Ferramentas

- ❑ Você pode utilizar várias ferramentas para desenvolvimento de soluções Struts:
- ❑ Gratuitas
 - Eclipse + plugins
 - NetBeans
 - JBuilder Open Edition + plugins
- ❑ Comerciais
 - Sun One Studio
 - Pramati Studio
 - JDeveloper
 - Borland JBuilder

Ferramentas Gratuitas e Genéricas

Struts Console

- Uma das principais ferramentas para desenvolvimento de aplicações Struts

□ <http://www.jamesholmes.com/struts/console/>

Ferramentas que oferecem suporte ao Struts Console:

- [Borland JBuilder](#) version 4.0 ou >
- [Eclipse](#) version 1.0 ou >
- [IBM WebSphere Appl. Dev.](#) version 4.0.3 ou >
- [IntelliJ IDEA](#) version 3.0 (build 668) ou >
- [NetBeans](#) version 3.2 ou >
- [Oracle JDeveloper](#) version 9i ou >
- [Sun Forte for Java](#) version 3.0 ou >

Instrumentos de Automação

- ❑ Jakarta Ant
- ❑ XDoclet
- ❑ Velocity
- ❑ Middlegen (usa Struts)
- ❑ UML2EJB (UmlTOEJB) (usa Struts)

Mostrando na Prática

❑ Requisitos:

- Usuário vai logar na aplicação
- É necessário validar a entrada do usuário
- É necessário autenticar esse usuário
- Logo após o Logon é imperativo armazenar informações do usuário logado na sessão da aplicação.

Praticando Struts

- ❑ Criando as Páginas
 - Você pode utilizar Tiles para definir templates.
 - Velocity também pode ser usado para definir páginas
 - Não perca a palestra:
 - ❑ **Template Engines: Uma alternativa real ao JSP** (Paulo Silveira)

Praticando Struts

❑ Página de Login



The mockup shows a web page for 'Suaempresa.com [Pagina de Topo]'. It features a dark blue sidebar with the word 'Menu' in white. The main content area has a light gray header 'Paginas de Login'. Below this is a light blue box containing the login form. The form has two input fields: 'Usuário:' with the value 'usuario' and 'Senha:' with the value '*****'. At the bottom of the form are two buttons: 'Ok' and 'Limpar'.

Suaempresa.com [Pagina de Topo]

Menu

Paginas de Login

Usuário:

Senha:

O que será necessário?

- ❑ A Classe do ServletController, a qual será automática.
- ❑ Classe ActionForm (opcional)
- ❑ Classe Action (opcional)
- ❑ Um registro de um ActionMapping no Struts-Config.xml

Por que é necessário um **ActionForm**?

- ❑ Para fazer as validações necessárias.
- ❑ Para que possamos criar um ciclo completo com Struts.
- ❑ Validar uma entrada de dados é um recurso muito útil do Struts.
- ❑ Observe que um **ActionForm** não é um **Model(VO)**, ele tem finalidade muito mais simples.
- ❑ Você pode reduzir a utilização de Classes por **Forms Dinâmicos (DynaActionForm)**.

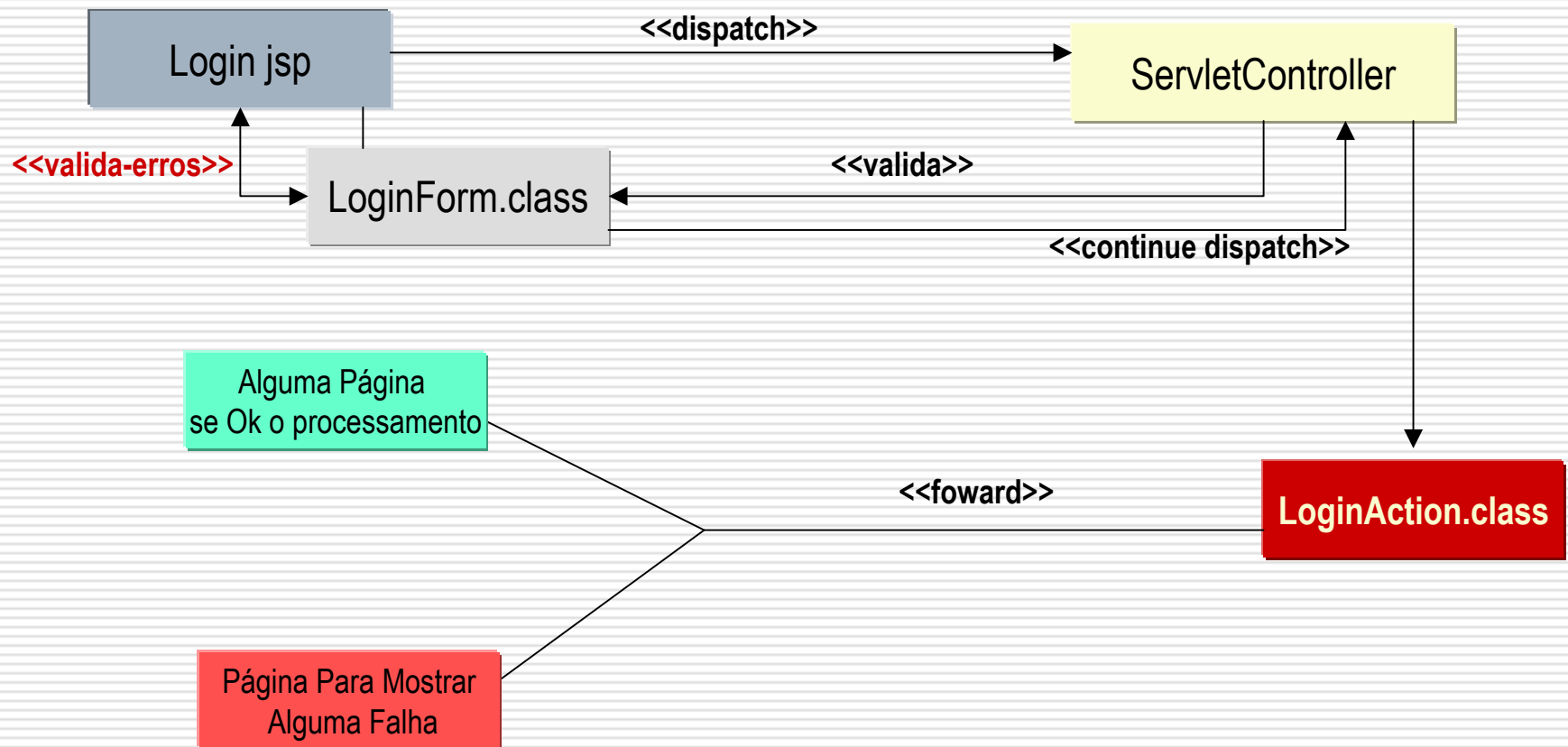
Por que usar uma classe Action

- ❑ Definir o comportamento da Ação
- ❑ Definir o que será feito
- ❑ É delegada pelo Servlet de Controle
- ❑ Conversa com os seus possíveis Modelos
- ❑ Para diminuir a quantidade de classes Action você pode usar:
 - `org.apache.struts.actions.DispatchAction`

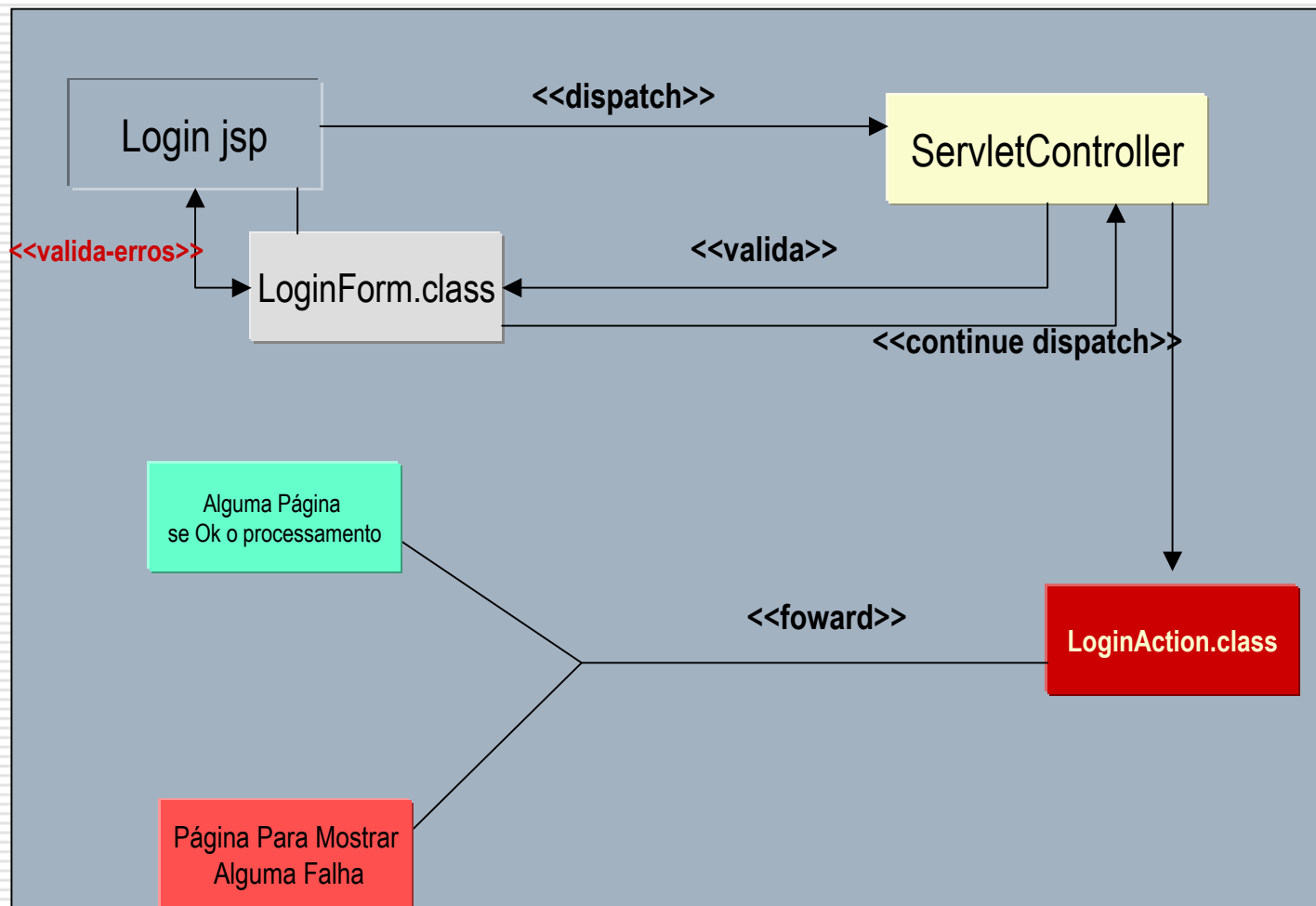
Por que uma Entrada no struts-config.xml?

- ❑ Como alguns livros de receitas:
 - “Put All Together”
- ❑ Finalizar o Ciclo da atividade, ou seja criando um ActionMapping (mapeamento da Ação)
- ❑ Todos os acoes.do serão encaminhados para os seus respectivos actionMapping: acoes.
- ❑ Ex:
 - Login.do ➔ ActionMapping = Login
 - listarClientes.do ➔ ActionMapping = listarClientes

Revisando o Esquema do Struts e seus Elementos



Revisando o Esquema do Struts e seus Elementos



A
C
T
I
O
N

M
A
P
P
I
N
G

A Classe ActionForm

- ❑ Possui basicamente dois métodos
 - validate()
 - reset()

Validando Struts com LoginForm

□ A Classe :

```
package com.argos.web.util;

import org.apache.struts.action.*;
import javax.servlet.http.*;
import com.argos.web.util.LoginValidator;
import com.argos.util.Usuario;

public class LoginForm extends ActionForm {
    private String senha;
    private String usuario;
    public void setSenha(String senha) {
        this.senha = senha;
    }
    public String getSenha() {
        return senha;
    }
}
```

Validando Struts com LoginForm

❑ A Classe :[continuacao]

```
public void setUsuario(String usuario) {  
    this.usuario = usuario;  
}  
  
public String getUsuario() {  
    return usuario;  
}
```

Validando Struts com LoginForm

□ A Classe :[continuacao – validate()]

```
public ActionErrors validate(ActionMapping  
actionMapping, HttpServletRequest httpRequest) {  
    ActionErrors errors = new ActionErrors();  
    if (usuario.equalsIgnoreCase("")) {  
        ActionError erroUsuario = new  
        ActionError("erro.global.campo.requerido");  
        errors.add("usuario", erroUsuario);  
    } ...  
}
```

Validando Struts com LoginForm

❑ A Classe :[continuacao – validate()]

```
if (senha.equalsIgnoreCase("")) {  
    ActionError erroEmail = new  
ActionError("erro.global.campo.requerido");  
    errors.add("senha", erroEmail);  
}  
return errors;  
}
```

Validando Struts com LoginForm

❑ A Classe :[continuacao – validate()]

```
public ActionErrors validate(ActionMapping  
actionMapping, HttpServletRequest httpRequest) {  
  
    ActionErrors errors = new ActionErrors();  
  
    if (usuario.equalsIgnoreCase("")) {  
  
        ActionError erroUsuario = new  
        ActionError("erro.global.campo.requerido");  
  
        errors.add("usuario", erroUsuario);  
  
    } ...  
}
```

Validando Struts com LoginForm

❑ A Classe :[continuacao – reset()]

```
public void reset(ActionMapping actionMapping,  
HttpServletRequest httpRequest) {  
  
    usuario="";  
  
    senha="";  
  
}
```

Validando Struts com LoginForm

❑ O registro no struts-config.xml :

```
<form-bean name="loginForm"
            type="com.argos.web.util.LoginForm">
    <form-property name="usuario"
                  type="java.lang.String" />
    <form-property name="senha"
                  type="java.lang.String" />
</form-bean>
```

Validando Struts com LoginForm

- ❑ Adicionando Elementos no struts-config :
 - Message Resources
 - Plugings

Validando Struts com LoginForm

❑ Exemplo do ApplicationResources_pt.properties:

```
#Labels para validacao

forms.login.usuario=Usuario

forms.login.senha=Senha

loginForm.title=Andromeda

# Labels do Andromeda

loginForm.usuario=Nome :

loginForm.senha=Senha :

#globais

erro.global.campo.requerido= {0} é requerido para executar a tarefa.

#Mensagens de Erro do Andromeda

pagina.login.usuario=Usuário Inválido

pagina.login.senha=Senha inválida tente novamente por favor.
```

Validando Struts com LoginForm

❑ Configurando o Validator.xml

```
<form-validation>
  <formset>
    <form name="loginForm">
      <field
        property="usuario"
        depends="required">
        <arg0 key="forms.login.usuario"/>
      </field>
      <field
        property="senha"
        depends="required">
        <arg0 key="forms.login.senha"/>
      </field>
    </form>
  </formset>
</form-validation>
```

Criando a Classe de Action

- ❑ Após passar pela LoginForm, o Controller delega para a classe Action:

```
public class LoginAction extends Action {  
    public ActionForward execute(ActionMapping actionMapping,  
                                ActionForm actionForm,  
                                HttpServletRequest  
httpServletRequest,  
                                HttpServletResponse  
httpServletResponse) {  
    //Aqui vai seu código.....a Sua Lógica  
}
```

Método execute() do Action

- ❑ O método execute está para uma classe Action, assim como um main() está para uma execução de uma classe normal Java.
- ❑ É o ponto de execução de uma chamada de execução da Classe xxxxAction.
- ❑ Na versão 1.0 do Struts o execute() “chamava-se” perform().
- ❑ O método perform() foi depreciado, mas ainda existe. O execute é um método novo usado pelo controller, mas o controller ainda reconhece o perform()

Método execute() do Action

```
try {String usuario = ((LoginForm)actionForm).getUsuario();
    String senha = ((LoginForm)actionForm).getSenha();
    httpRequest.getSession(true).getServletContext().log("Iniciando Login de
" + usuario);
    Usuario usuarioemLogin = LoginValidator.executeLogin(usuario,senha);
    if (usuarioemLogin!=null) {
        httpRequest.getSession(true).setAttribute("usuario",usuarioemLogin);
        httpRequest.getSession(true).getServletContext().log("Login Efetuado
com sucesso para " + usuario);
        return actionMapping.findForward("sucesso_no_login");
    } else {
        httpRequest.getSession(true).getServletContext().log("Falha Login
Generico");
        httpRequest.getSession(true).setAttribute("erro","Falha no Login , por
favor tente novamente.");
        return actionMapping.findForward("falha_no_login");
    }
}
.....
.....
```

Entrada do Action Mapping

□ Porção do struts-config.xml:

```
<action name="loginForm"
        type="com.argos.web.util.LoginAction"
        validate="true"
        input="/login.jsp" scope="session" path="/login">
    <forward name="sucesso_no_login"
path="/usuario_logado.jsp" />
    <forward name="falha_no_login" path="/erros.jsp" />
</action>
```

Visualizando a Aplicação [Módulo de Login]

- ❑ Aplicação Rodando no JBoss
 - <http://localhost:8080/argos>

Aspectos de View (Visualização)

- ❑ O Struts possui várias TagLibraries que facilitam demais a qualidade da visualização.

Struts Tag Libraries

- ❑ Basicamente temos
 - Logic: Operações Lógicas/Condicionais e de caminhamento de Collections;
 - HTML: Recursos melhorados do HTML, para formulários, botões, javascript e etc.
 - Bean: Manipulação de Propriedades de Beans, em qualquer escopo (aplicação, request, sessão);
 - Template: Para usar recursos de Templates
 - Tiles: Nova maneira de criar templates no Struts 1.1.

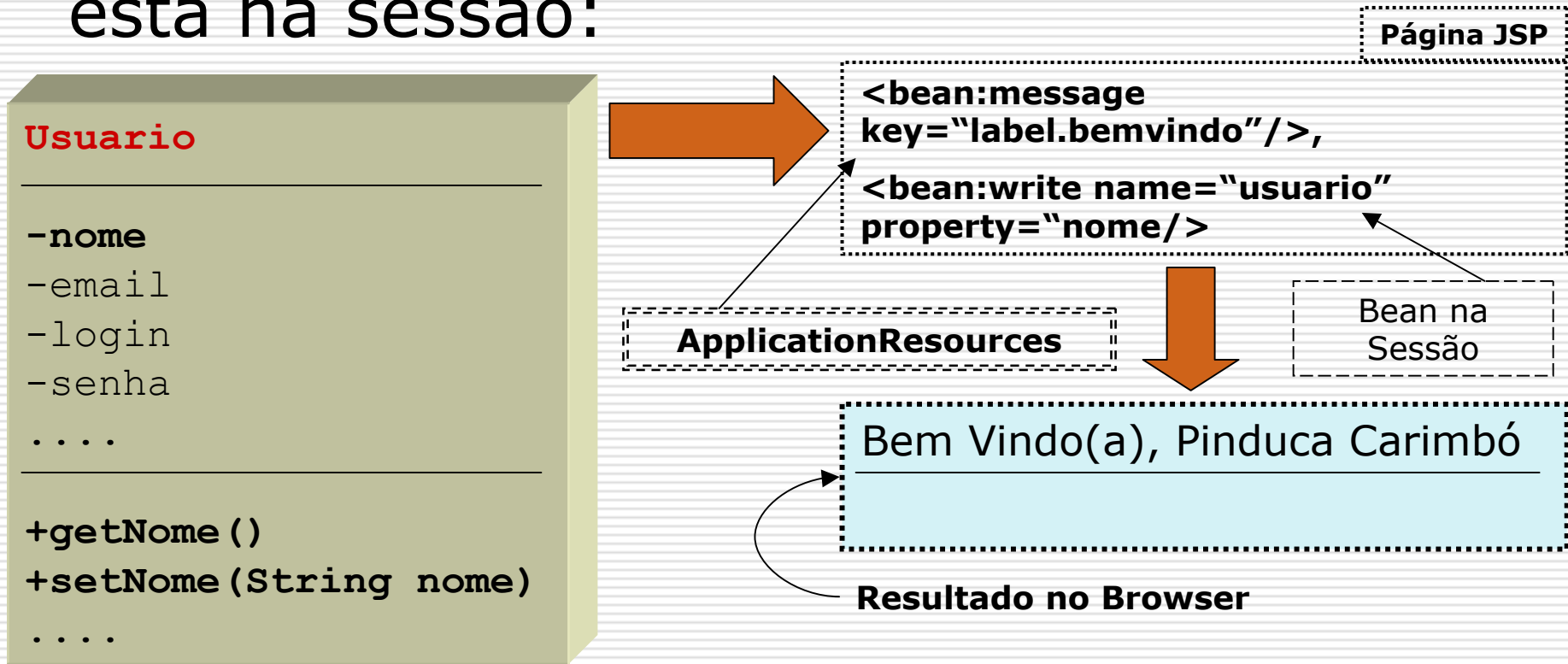
Usando de dentro do JSP

```
<% taglib uri="/WEB-INF/struts-logic.tld" prefix="logic"%>  
<% taglib uri="/WEB-INF/struts-template.tld" prefix="template"%>  
<% taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>  
<% taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
```

Com essas Bibliotecas de Tags Você pode controlar, gerenciar e apresentar conteúdo dinâmico com extrema facilidade.

Utilizando estas TagLibraries

- ❑ Levando-se em consideração que um Bean está na sessão:



Exibindo o Erro a Nível de Páginas

- ❑ `<html:errors/>`
 - Exibe todos os Erros de uma só vez.
- ❑ `<html:errors property="NOMECAMPO"/>`
 - Você preenche o nome do campo no espaço property.

Exibindo os Erros em JavaScript

□ Fácil:

```
<html:javascript formName="loginForm"/>
```



```
<html:form action="/login.do" focus="usuario"
onsubmit='return validateLoginForm(this);' method="POST">
```

Como age o Validator?

- ❑ ApplicationResources.properties
 - valor.requerido={0} É requerido para a ação.

```
<form-validation>
  <formset>
    <form name="loginForm">
      <field
        property="usuario"
        depends="required">
        <arg0 key="forms.login.usuario"/>
      </field>
```

Melhorando ainda Mais a Visualização (view)

□ Struts-Layout

■ <http://struts.application-servers.com/>

Ótimos recursos adicionais para o Struts

Do que é composto? [Struts-Layout]

□ Tags Para:

- **Paginação**
- **Abas (TabControl)**
- **Exibição de erros**
- **Menu**
- **TreeView**
- **Abrir um Calendário e Retornar para um Campo**
- **Clicar num botão, abrir uma janela, pesquisa e retornar um resultado para o Campo de origem do clique.**
- **Skins**
- **Transferir items de uma Collection para outra.**
- **Formatação de dados**

Recursos na Internet

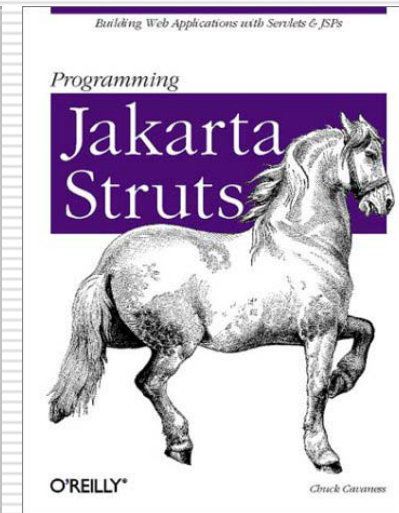
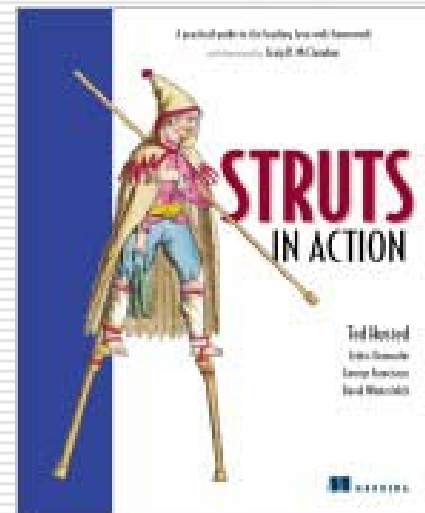
- ❑ Jakarta.apache.org/struts
- ❑ www.jspbrasil.com.br
- ❑ Artigos interessantes:
 - <http://jakarta.apache.org/struts/resources/articles.html>

Publicações

- ❑ Revista JavaMagazine do mês de Abril/02
 - www.javamagazine.com.br

Livros

- ❑ **Struts in Action: Building Web Applications with the Leading Java Framework**
by Ted Husted, Cedric Dumoulin, George Franciscus, David Winterfeldt, Craig R. McClanahan
- ❑ **Programming Jakarta Struts**
by Chuck Cavaness



Perguntas?

□ ?????

Agradecimentos

□ esilva@argostecnologia.com

Materiais Disponíveis em:

→ <http://www.argostecnologia.com/edgar/justjava/>

Argos Tecnologia

www.argostecnologia.com

Belém-Pará