



Tutorial Struts² – Roteiro para Demonstrações

Vítor E. Silva Souza (vitorsouza@gmail.com)

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

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.

01. Preparação:

Instalações:

- ◆ Tomcat 5.5;
- ◆ Eclipse Europa com ferramentas Java EE;
 - ✓ Configuração do Tomcat no Eclipse (Window – Preferences, Server – Installed Runtimes).

Aumentar a fonte do Eclipse:

- ◆ Windows – Preferences, General – Appearance – Colors and Fonts
 - ✓ Basic – Text Font: 16 negrito

Novo projeto:

- ◆ File – New – Project, Web – Dynamic Web Project
 - ✓ Name: TutorialStruts2 (o runtime já deve estar configurado!)
 - ✓ Fazer um index.jsp e rodar o Tomcat no Eclipse para demonstrar.

Bibliotecas necessárias:

- ◆ Copiar build.xml e ivy.xml para pasta do projeto;
- ◆ Copiar ivy-1.4.1.jar para Workspace;
- ◆ Window – Preferences, Ant – Runtime, clique em Ant Home Entries, Add JARs..., selecione ivy-1.4.1.jar e clique OK;
- ◆ Execute o build.xml e mostre as dependências no diretório lib (é necessário Refresh);

- ◆ Project Properties, J2EE Module Dependencies, Add JARs..., selecione todos e OK.

02. Configuração básica e primeira ação:

Copiar arquivos da aplicação em branco

HelloStruts2Action

```
public class HelloStruts2Action extends ActionSupport {
    private String mensagem = "Hello, Struts2! (" + new Date() + ")";

    public String getMensagem() { return mensagem; }

    /* Método execute() não precisa ser sobrescrito. */
    /* Implementação de ActionSupport retorna SUCCESS. */
}
```

struts.xml

```
<package [...]>
    <action name="helloStruts2"
class="net.java.dev.esjug.tutorialstruts2truts2.
                                controller.HelloStruts2Action">
        <result>/WEB-INF/pages/hello.jsp</result>
    </action>
</package>
```

hello.jsp

```
<%@ taglib prefix="s" uri="/struts-tags" %>

<html>
<head>
    <title>Tutorial Struts2</title>
    <style type="text/css">
        body { font-size: 16pt; }
        h1 { font-size: 20pt; }
    </style>
</head>
<body>

<h1>Tutorial Struts2</h1>

<p><s:property value="mensagem" /></p>

</body>
</html>
```

home.jsp

```
<p>2 - <a href="helloStruts2.action">Primeira ação</a></p>
```

03. Entendendo melhor as ações

Apresentar DateUtils.java

home.jsp

```
<p>3 - <a href="passo03/calcularIdade!input.action">Calcular idade</a></p>
```

struts.xml

```
<package name="passo03" extends="struts-default" namespace="/passo03">
    <action name="calcularIdade"
        class="net.java.dev.esjug.tutorialstruts2.controller.CalcularIdadeAction">
        <result name="input">/WEB-INF/pages/passo03/form.jsp</result>
        <result>/WEB-INF/pages/passo03/resultado.jsp</result>
    </action>
</package>
```

passo03/form.jsp

```
<s:form namespace="/passo03" 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>
```

CalcularIdadeAction.java

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

    public String getNome() { return nome; }

    public void setNome(String nome) { this.nome = nome; }

    public int getIdade() { return idade; }

    public void setDataNascimento(Date dataNascimento) {
        this.dataNascimento = dataNascimento;
    }

    public String execute() throws Exception {
        idade = DateUtils.calculateYearDifference(dataNascimento,
            new Date(System.currentTimeMillis()));

        return SUCCESS;
    }
}
```

resultado.jsp (renomear hello.jsp)

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

ATENÇÃO AO EXECUTAR:

Se o Locale da máquina não usar datas no formato brasileiro (dd/MM/yyyy), colocar datas neste formato causará o não funcionamento da ação! O struts espera a data no formato correto! Veja a configuração struts.locale no arquivo struts.properties!

04. Entendendo melhor os resultados

Mostrar ControleLogin.java e Funcionario.java

home.jsp

<p>4 - Formulário de login</p>

struts.xml

```
<package name="passo04" extends="struts-default" namespace="/passo04">
  <action name="login"
    class="net.java.dev.esjug.tutorialstruts2.controller.LoginAction">

    <result name="input">/WEB-INF/pages/passo04/form.jsp</result>
    <result name="desenvolvedor">
      /WEB-INF/pages/passo04/homeDesenvolvedor.jsp</result>
    <result name="gerente">
      /WEB-INF/pages/passo04/homeGerente.jsp</result>
    <result name="diretor">
      /WEB-INF/pages/passo04/homeDiretor.jsp</result>
  </action>
</package>
```

passo04/form.jsp

```
<p style="color:red"><s:property value="mensagemErro" /></p>

<s:form namespace="/passo04" action="login" method="post">
  <s:textfield label="Login" name="login" />
  <s:password label="Senha" name="senha" />
  <s:submit value="Login" />
</s:form>
```

LoginAction.java

```
public class LoginAction extends ActionSupport {
  private String login;
  private String senha;
  private Funcionario funcionario;
  private String mensagemErro;

  /* Getters & setters... */

  public String execute() throws Exception {
    funcionario = ControleLogin.recuperarPorLogin(login);

    if (funcionario == null) {
      mensagemErro = "Login inexistente!";
      return INPUT;
    }

    if (! funcionario.getSenha().equals(senha)) {
      mensagemErro = "Senha incorreta!";
      return INPUT;
    }

    switch (funcionario.getTipo()) {
      case Funcionario.TIPO_DESENVOLVEDOR: return "desenvolvedor";
      case Funcionario.TIPO_GERENTE: return "gerente";
      case Funcionario.TIPO_DIRETOR: return "diretor";
    }
  }
}
```

```

        return INPUT;
    }
}

```

passo04/: ./homeGerente.jsp, ./homeDiretor.jsp, ./homeDesenvolvedor.jsp

```
<p>Olá, Sr. <s:property value="funcionario.nome" />!</p>
```

```
<p>Olá, Dr. <s:property value="funcionario.nome" />!</p>
```

```
<p>Qualé <s:property value="funcionario.nome" />?</p>
```

05. A biblioteca de tags do struts

Mostrar (src): TagsAction.java, TagsSubAction.java

Mostrar (WebContent): diretório template

Mostrar (WebContent/WEB-INF/pages/pass05): tags.jsp, paginaInclude.htm

home.jsp

```
<s:url id="urlPasso05" namespace="/passo05" action="tags" />
<p>5 - <s:a href="%{urlPasso05}">Tags do struts</s:a></p>
```

struts.xml

```
<package name="passo05" extends="struts-default" namespace="/passo05">
    <action name="tags"
        class="net.java.dev.esjug.tutorialstruts2.controller.TagsAction">
        <result>/WEB-INF/pages/pass05/tags.jsp</result>
    </action>
    <action name="sub"
        class="net.java.dev.esjug.tutorialstruts2.controller.TagsSubAction"
    />
</package>
```

06. Trocando JSP por FreeMarker

Alterar CalculaIdadeAction.java

```
private String instrucoes = "Por favor informe seu nome e sua data de
nascimento.";
public String getInstrucoes() { return instrucoes; }
```

Mostrar no web.xml

```
<servlet>
    <servlet-name>JspSupportServlet</servlet-name>
    <servlet-class>
        com.opensymphony.struts.views.JspSupportServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
```

home.jsp

<p>6 - Cálculo da idade (com FreeMarker)</p>

struts.xml

```
<package name="struts-freemarker" extends="struts-default">
  <result-types>
    <result-type name="freemarker"
      class="org.apache.struts2.views.freemarker.FreemarkerResult"
      default="true" />
  </result-types>
</package>

<package name="passo06" extends="struts-freemarker" namespace="/passo06">
  <action name="calcularIdade"

    class="net.java.dev.esjug.tutorialstruts2.controller.CalcularIdadeAction">

    <result name="input">/WEB-INF/pages/passo06/form.ftl</result>
    <result>/WEB-INF/pages/passo06/resultado.ftl</result>
  </action>
</package>
```

passo06/form.ftl (copie do passo03)

```
<#assign ww=JspTaglibs["/WEB-INF/lib/struts-2.2.2.jar"]>

<p>${instrucoes?capitalize}</p>

<@s.form namespace="/passo06" 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>
```

passo06/resultado.ftl (copie do passo03)

```
<#assign ww=JspTaglibs["/WEB-INF/lib/struts-2.2.2.jar"]>

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

07. Nos bastidores, interceptadores

home.jsp

<p>7 - Cumprimento</p>

CumprimentoInterceptor.java

```
public class CumprimentoInterceptor implements Interceptor {
  public String intercept(ActionInvocation invocation) throws Exception {
    // Decide qual cumprimento utilizar.
    Calendar calendar = Calendar.getInstance();
    int hora = calendar.get(Calendar.HOUR_OF_DAY);
    String cumprimento = (hora < 6) ? "Boa noite" : ((hora < 12) ?
      "Bom dia" : ((hora < 18) ? "Boa tarde" : "Boa noite"));

    // Coloca o cumprimento na sessão.
    invocation.getInvocationContext().getSession()
      .put("cumprimento", cumprimento);
  }
}
```

```

        // Executa a ação.
        String result = invocation.invoke();

        // É possível executar código
        // depois da ação aqui...

        // Retorna o resultado da ação.
        return result;
    }
}

```

struts.xml

```

<package name="passo07" extends="struts-freemarker" namespace="/passo07">
    <interceptors>
        <interceptor name="cumprimento"
class="net.java.dev.esjug.tutorialstruts2.controller.CumprimentoInterceptor" />

        <interceptor-stack name="cumprimentoStack">
            <interceptor-ref name="defaultStack" />
            <interceptor-ref name="cumprimento" />
        </interceptor-stack>
    </interceptors>

    <action name="cumprimento">
        <interceptor-ref name="cumprimentoStack" />
        <result>/WEB-INF/pages/passo07/cumprimento.ftl</result>
    </action>
</package>

```

passo07/cumprimento.ftl

```

<p>${Session['cumprimento']}!</p>

```

08. Validando dados automaticamente

Mostrar dwr.xml e web.xml

home.jsp

```

<p>8 - <a href="passo08/validar!input.action">Validação</a></p>

```

struts.xml

```

<package name="passo08" extends="struts-freemarker" namespace="/passo08">
    <action name="validar"
class="net.java.dev.esjug.tutorialstruts2.controller.ValidarAction">

        <result name="input">/WEB-INF/pages/passo08/form.ftl</result>
        <result>/WEB-INF/pages/passo08/resultado.ftl</result>
    </action>
</package>

```

form.ftl

```

span.errorMessage { color: red; }
label.errorLabel { font-weight: bold; }

```

```

<!-- Para a validação server-only, não use <@s.head /> -->
<!-- Para a validação JS, não use theme="ajax" -->
<@s.head theme="ajax" />

```

```

<!-- Para a validação server-only, não use os atributos validate e theme -->
<!-- Para a validação JS, não use theme="ajax" -->
<@s.form namespace="/passo08" action="validar" method="post" validate="true"
theme="ajax">
    <@s.textfield label="Nome" name="nome" size="40" required="true" />
    <@s.radio label="Tipo" name="tipo" list="%{'A', 'B', 'C'}"
required="true" value="'${tipo?default('')}'" />
    <@s.textfield label="Percentual de comissão" name="percentual" size="10"
required="true" />
    <@s.textfield label="Data de nascimento" name="dataNascimento"
required="true" />
    <@s.textfield label="E-mail" name="email" size="40" required="true" />
    <@s.textfield label="Página pessoal" name="paginaPessoal" size="40"
required="true" />
    <@s.submit value="Enviar" />
</@s.form>

```

ValidarAction.java

```

public class ValidacaoAction extends ActionSupport {
    private String nome;
    private String tipo;
    private Integer percentual;
    private Date dataNascimento;
    private String email;
    private String paginaPessoal;
    private Date agora = new Date(System.currentTimeMillis());

    /* Getters e setters para todos. */
}

```

Mostrar ValidarAction-validar-validation.xml

resultado.ftl

```

<ul>
    <li>Nome: ${nome}</li>
    <li>Tipo: ${tipo}</li>
    <li>Percentual de Comissão: ${percentual}</li>
    <li>Data de Nascimento: ${dataNascimento?date?string.medium}</li>
    <li>E-mail: ${email}</li>
    <li>Página pessoal: ${paginaPessoal}</li>
</ul>

```

09. Invertendo o controle – IoC/DI

Mostrar CarrinhoCompras.java, APlEfetuarCompra.java e APlEfetuarComprasImpl.java

home.jsp

```

<p>9 - <a href="passo09/comprar!input.action">Carrinho de compras</a></p>

```

struts.xml

```

<package name="passo09" extends="struts-freemarker" namespace="/passo09">
    <action name="comprar"
        class="net.java.dev.esjug.tutorialstruts2.controller.ComprarAction">
        <result name="input">/WEB-INF/pages/pass09/home.ftl</result>
    </action>

```



```
</package>
```

ComprarAction.java

```
public class ComprarAction extends ActionSupport implements Preparable {
    private AplEfetuarCompra aplEfetuarCompra;
    private CarrinhoCompras carrinhoCompras;
    private Map<String, String> itens = new TreeMap<String, String>();
    private String item;

    @Override
    public void prepare() throws Exception {
        itens.put("Teclado", "Teclado");
        itens.put("Mouse", "Mouse");
        itens.put("Monitor", "Monitor");
        itens.put("Caixa de Som", "Caixa de Som");
    }

    // Adicionar via Eclipse:
    // =====
    // setCarrinhoCompras
    // setAplEfetuarCompra
    // getItens
    // setItem
    // getProdutos

    public String execute() throws Exception {
        aplEfetuarCompra.adicionarAoCarrinho(carrinhoCompras, item);
        return INPUT;
    }
}
```

applicationContext-core.xml

```
<beans [...]>
    <bean id="aplEfetuarCompra"
        class="net.java.dev.esjug.tutorialstruts2.util.AplEfetuarCompraImpl" />

    <bean id="carrinhoCompras"
        class="net.java.dev.esjug.tutorialstruts2.util.CarrinhoCompras"
        scope="session" />
</beans>
```

passo09/home.ftl

```
<p>Produtos no Carrinho de Compras:</p>
<ul>
<#list produtos as produto>
    <li>${produto}</li>
</#list>
</ul>

<@s.form namespace="/passo09" action="comprar" method="post">
    <@s.select label="Produto" name="item" list="itens" />
    <@s.submit value="Comprar" />
</@s.form>
```

10. Ações encadeadas

Mostrar ControleCadastro.java

home.jsp

```
<p>10 - <a href="passo10/cadastrar!input.action">Cadastro de usuários</a></p>
```

struts.xml

```
<package name="passo10" extends="struts-freemarker" namespace="/passo10">
    <action name="cadastrar"
class="net.java.dev.esjug.tutorialstruts2.controller.CadastrarAction">
        <result name="input">/WEB-INF/pages/passo10/form.ftl</result>
        <result type="chain">
            <param name="actionName">ver</param>
            <param name="namespace">/passo10b</param>
        </result>
    </action>
</package>
<package name="passo10b" extends="struts-freemarker" namespace="/passo10b">
    <action name="ver"
class="net.java.dev.esjug.tutorialstruts2.controller.VerCadastroAction">
        <result>/WEB-INF/pages/passo10b/resultado.ftl</result>
    </action>
</package>
```

passo10/form.ftl

```
<@s.form namespace="/passo10" action="cadastrar" method="post">
    <@s.textfield label="Nome" name="funcionario.nome" />
    <@s.textfield label="Login" name="funcionario.login" />
    <@s.password label="Senha" name="funcionario.senha" />
    <@s.radio name="funcionario.tipo" list="mapaTipos" listKey="key"
        listValue="value" />
    <@s.submit value="Cadastrar" />
</@s.form>
```

CadastrarAction.java

```
public class CadastrarAction extends ActionSupport {
    private Long id;
    private Funcionario funcionario = new Funcionario();
    private static Map<Integer, String> mapaTipos
        = new TreeMap<Integer, String>();
    static {
        mapaTipos.put(Funcionario.TIPO_DESENVOLVEDOR, "Desenvolvedor");
        mapaTipos.put(Funcionario.TIPO_DIRETOR, "Diretor");
        mapaTipos.put(Funcionario.TIPO_GERENTE, "Gerente");
    }

    /* Getters & setters. */

    public String execute() throws Exception {
        id = ControleCadastro.gravarFuncionario(funcionario);
        return SUCCESS;
    }
}
```

VerCadastroAction.java

```
public class VerCadastroAction extends ActionSupport {
    private Long id;
    private Funcionario funcionario;

    /* Getters & setters. */

    /* Método execute() não precisa ser sobrescrito. */
    /* Implementação de ActionSupport retorna SUCCESS. */
}
```

passo10b/resultado.ftl

```
<p>Funcionário cadastrado:</p>

<#assign tipo = '' />
<#switch funcionario.tipo>
    <#case 1><#assign tipo = 'Desenvolvedor' /><#break>
    <#case 2><#assign tipo = 'Gerente' /><#break>
    <#case 3><#assign tipo = 'Diretor' /><#break>
</#switch>

<p>Nome: ${funcionario.nome}<br />
Login: ${funcionario.login}<br />
Senha: ${funcionario.senha}<br />
Tipo: ${tipo}</p>
```

11. Enviando arquivos

Adicionar dependência: Apache Commons Upload 1.2

```
<dependency org="apache" name="commons-fileupload" rev="1.2" />
```

home.jsp

```
<p>11 - <a href="passo11/enviarArquivo!input.action">Envio de arquivo</a></p>
```

struts.xml

```
<package name="passo11" extends="struts-freemarker" namespace="/passo11">
    <action name="enviarArquivo"

class="net.java.dev.esjug.tutorialstruts2.controller.EnviaArquivoAction">

        <result name="input">/WEB-INF/pages/passo11/form.ftl</result>
        <result>/WEB-INF/pages/passo11/resultado.ftl</result>
    </action>
</package>
```

passo11/form.ftl

```
<@s.form namespace="/passo11" action="enviarArquivo" method="post"
enctype="multipart/form-data">
    <@s.file label="Arquivo" name="arquivo" />
    <@s.submit value="Enviar" />
</@s.form>
```

EnviarArquivoAction.java

```
public class EnviarArquivoAction extends ActionSupport {
    private File arquivo;
    private String arquivoFileName;
```

```

private String arquivoContentType;
private long tamanho;

/* Getters & setters. */

public String execute() throws Exception {
    tamanho = arquivo.length();
    return SUCCESS;
}
}

```

passo11/resultado.ftl

```

<p>0 arquivo ${arquivoFileName} (tipo ${arquivoContentType}) possui:</p>
<ul>
    <li>${tamanho} bytes</li>
    <#if (tamanho >= 1024)><li>${tamanho / 1024} KB</li></#if>
    <#if (tamanho >= (1024 * 1024))><li>${(tamanho / 1024) / 1024}
MB</li></#if>
</ul>

```

12. Interfaces internacionais – i18n

home.jsp

```

<p>12 - <a href="passo12/calcularIdade!input.action">Cálculo da idade (com
i18n)</a></p>

```

struts.xml

```

<package name="passo12" extends="struts-freemarker" namespace="/passo12">
    <action name="calcularIdade"

        class="net.java.dev.esjug.tutorialstruts2.controller.CalcularIdadeAction">

        <result name="input">/WEB-INF/pages/passo12/form.ftl</result>
        <result>/WEB-INF/pages/passo12/resultado.ftl</result>
    </action>
</package>

```

passo12/form.ftl

```

<p><@s.text name="texto.boasvindas" /></p>

<@s.form namespace="/passo12" action="calcularIdade" method="post">
    <@s.textfield label="%{getText('form.nome')}" name="nome" />
    <@s.textfield label="%{getText('form.dataNascimento')}"
        name="dataNascimento" />

    <@s.submit value="%{getText('form.submit')}" />
</@s.form>

<p><@s.text name="texto.mudaridioma" />:
<a href="?request_locale=en_US"><@s.text name="texto.ingles" /></a> |
<a href="?request_locale=pt_BR"><@s.text name="texto.portugues" /></a></p>

```

CalcularIdadeAction-validation.xml

```

<validators>
    <field name="nome">
        <field-validator type="requiredstring">
            <message key="erro.nome.vazio" />
        </field-validator>
    </field>
</validators>

```

```

        </field-validator>
    </field>
    <field name="dataNascimento">
        <field-validator type="date">
            <message key="erro.dataNascimento.dataInvalida" />
        </field-validator>
    </field>
</validators>

```

passo12/resultado.ftl

```

<p><@s.text name="texto.resposta">
    <@s.param value="nome" />
    <@s.param value="idade" />
</@s.text></p>

```

Mostrar CalcularIdadeAction_pt_BR.properties e CalcularIdadeAction_en.properties

13. Usando o resultado stream

home.jsp

```

<p>13 - <a href="passo13/gerarGrafico!input.action">Gráfico</a></p>

```

struts.xml

```

<package name="passo13" extends="struts-freemarker" namespace="/passo13">
    <action name="gerarGrafico"

        class="net.java.dev.esjug.tutorialstruts2.controller.GerarGraficoAction">

        <result name="input">/WEB-INF/pages/passo13/form.ftl</result>
        <result type="stream">
            <param name="contentType">{%contentType}</param>
        </result>
    </action>
</package>

```

passo13/form.ftl

```

<p>Número de projetos que usam:</p>

<@s.form namespace="/passo13" action="gerarGrafico" method="post">
    <@s.textfield label="Spring MVC" name="usoSpring" />
    <@s.textfield label="Struts 1" name="usoStruts1" />
    <@s.textfield label="WebWork" name="usoWebWork" />
    <@s.textfield label="Struts 2" name="usoStruts2" />
    <@s.submit value="Enviar" />
</@s.form>

```

GerarGraficoAction.java

```

public class GerarGraficoAction extends ActionSupport {
    private int usoSpring;
    private int usoStruts1;
    private int usoWebWork;
    private int usoStruts2;
    private String contentType = "image/jpeg";
    private InputStream inputStream;

    /* Getters & setters. */

```

```

public String execute() throws Exception {
    // Monta o dataset que populará o gráfico.
    DefaultPieDataset dataset = new DefaultPieDataset();
    dataset.setValue("Spring MVC", usoSpring);
    dataset.setValue("Struts 1", usoStruts1);
    dataset.setValue("WebWork", usoWebWork);
    dataset.setValue("Struts 2", usoStruts2);

    // Gera o gráfico.
    JFreeChart grafico = ChartFactory.createPieChart3D(
        "Uso de Frameworks Web", dataset, true, false, false);

    // Coloca o gráfico no input stream que será lido pelo struts.
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    ChartUtilities.writeChartAsJPEG(outputStream, grafico, 600, 300);
    inputStream = new ByteArrayInputStream(outputStream.toByteArray());

    return SUCCESS;
}
}

```

14. Relatórios com JasperReports

home.jsp

```
<p>14 - <a href="passo14/gerarRelatorio!input.action">Relatório</a></p>
```

struts.xml

```

<package name="passo14" extends="struts-freemarker" namespace="/passo14">
    <action name="gerarRelatorio"
class="net.java.dev.esjug.tutorialstruts2.controller.GerarRelatorioAction">

        <result name="input">/WEB-INF/pages/passo14/form.ftl</result>
        <result name="success" type="jasper">
            <param name="location">
                WEB-INF/reports/usoFramework.jasper
            </param>
            <param name="dataSource">usosFramework</param>
            <param name="documentName">usoFramework</param>
        </result>
    </action>
</package>

```

passo14/form.ftl

```
<!-- Igual ao passo13, apenas muda a ação do formulário -->
```

```
<@s.form namespace="/passo14" action="gerarRelatorio" method="post">
```

GerarRelatorioAction.java: copiar de GerarGraficoAction e adicionar:

```

/* Na classe: */
private List<UsoFramework> usosFramework = new ArrayList<UsoFramework>();

/* Adicionar getter para usosFramework */

/* No método execute: */
usosFramework.add(new UsoFramework("Spring MVC", usoSpring));
usosFramework.add(new UsoFramework("Struts 1", usoStruts1));
usosFramework.add(new UsoFramework("WebWork", usoWebWork));

```

```

        usosFramework.add(new UsoFramework("Struts 2", usoStruts2));
    }

    /* Fora da classe: */
    class UsoFramework {
        private String nome;
        private int quantidade;

        public UsoFramework(String nome, int quantidade) {
            this.nome = nome;
            this.quantidade = quantidade;
        }

        public String getNome() {
            return nome;
        }

        public int getQuantidade() {
            return quantidade;
        }
    }
}

```

15. Integração com SiteMesh

WEB-INF/web.xml

```

<!-- Entre o filtro do Spring e do struts. -->
<filter>
    <filter-name>strutsCleanup</filter-name>
    <filter-class>
        org.apache.struts2.dispatcher.ActionContextCleanUp
    </filter-class>
</filter>
<filter>
    <filter-name>sitemesh</filter-name>
    <filter-class>
        org.apache.struts2.sitemesh.FreeMarkerPageFilter
    </filter-class>
</filter>

<!-- Entre o filtro do Spring e do struts. -->
<filter-mapping>
    <filter-name>strutsCleanup</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>sitemesh</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

WEB-INF/decorators.xml

```

<decorators defaultdir="/WEB-INF/decorators">
    <decorator name="main" page="main.ftl">
        <pattern>/*</pattern>
    </decorator>
</decorators>

```

Mostrar WEB-INF/decorators/main.ftl