

Laboratorio. Introducción a Struts

Objetivo:

Instalar el entorno de desarrollo para la creación de aplicaciones que utilicen la tecnología Struts, modificar el contenido de los archivos de configuración necesarios de acuerdo a las clases a manejar y desarrollar una aplicación sencilla utilizando la tecnología Struts.

Pasos a seguir:

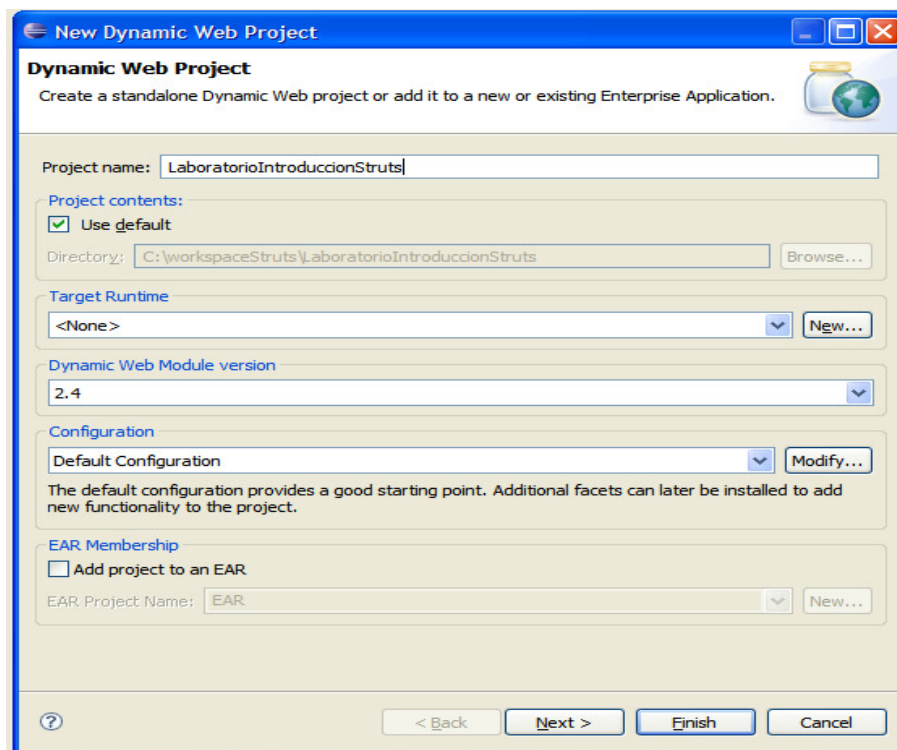
1. Configurar el entorno de desarrollo
2. Programación de las clases necesarias y modificación de los archivos de configuración

1. Configurar el entorno de desarrollo

Los archivos necesarios para el desarrollo de esta práctica son:

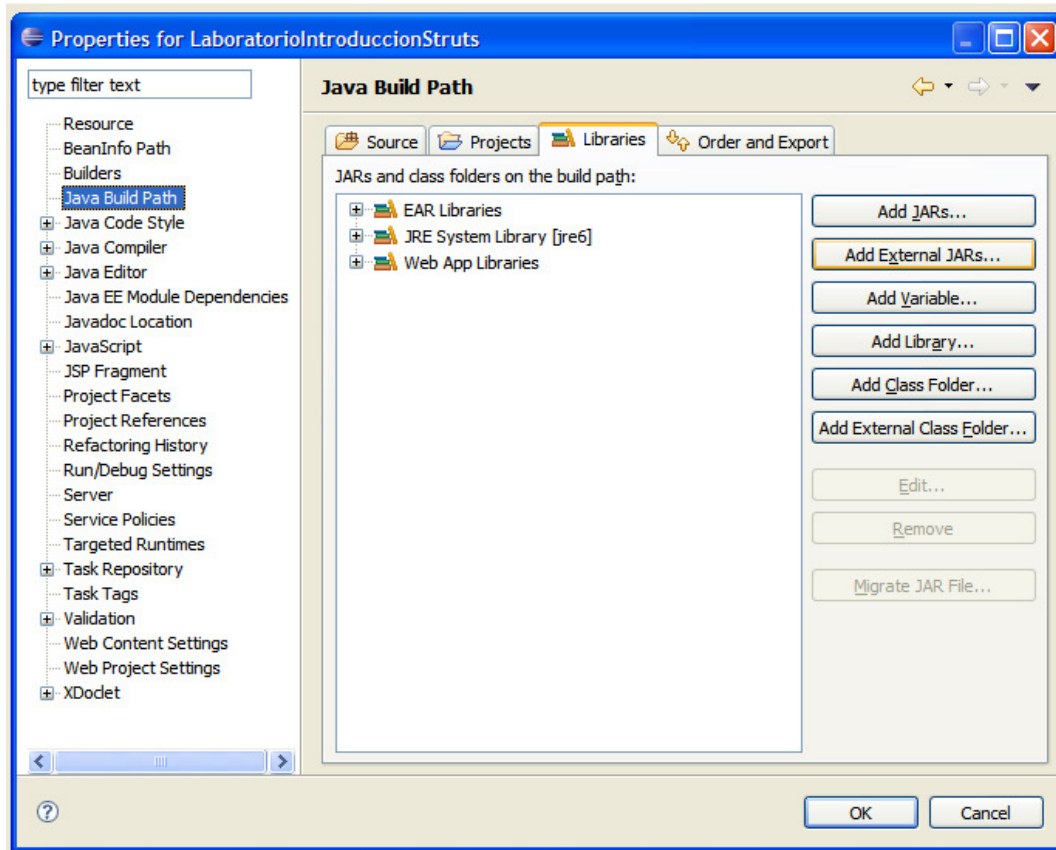
- **j2ee_14.jar** para el manejo de Servlets
- Las jars necesarias para el manejo de Struts se encuentran en el archivo **struts.rar**

Lo primero que se debe hacer es crear un nuevo proyecto del tipo Dynamic Web Project

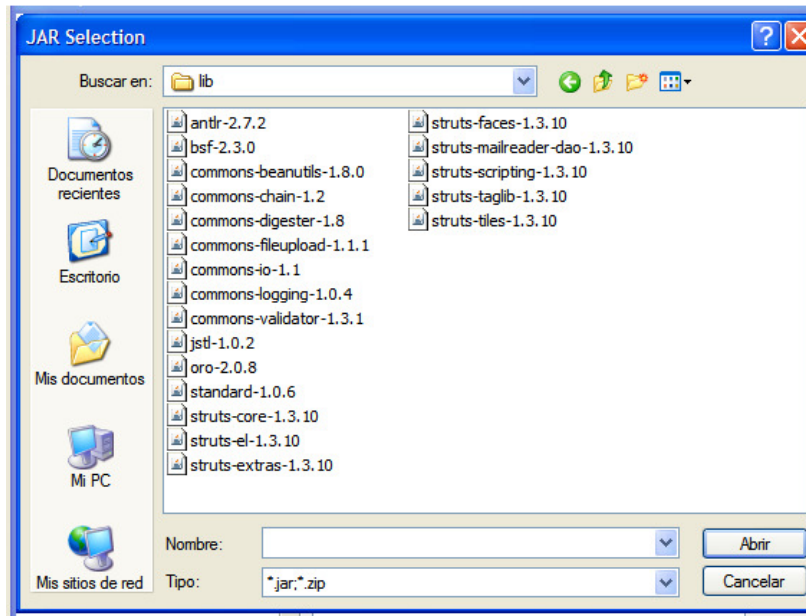


Agregar las bibliotecas necesarias para el desarrollo con Struts:

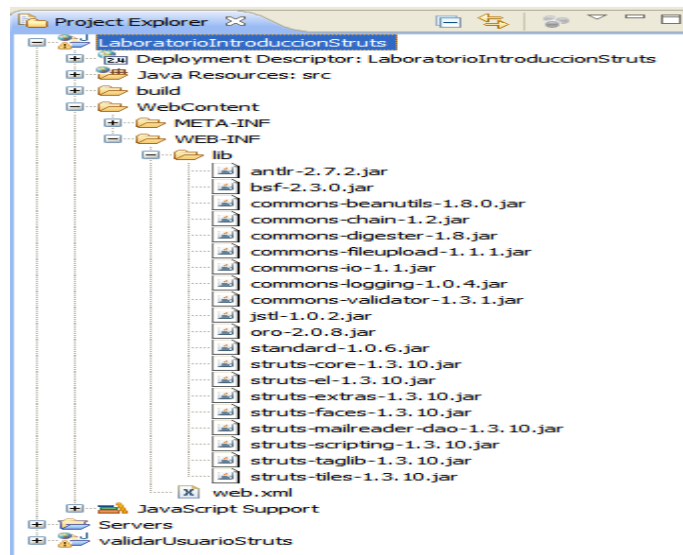
- Clic derecho sobre el nombre del proyecto
- Clic en Properties
- Seleccionar Java Build Path
- Seleccionar la pestaña Libraries
- Seleccionar Add External JARs...



- Buscar y seleccionar el archivo **j2ee_14.jar**
- Buscar y seleccionar los archivos que se muestran a continuación que son necesarias para el manejo de Struts, seleccionar todos los archivos y dar clic en Abrir



- Dar clic en OK para agregar las bibliotecas al proyecto
- Copiar estos últimos archivos al directorio `\WebContent\WEB-INF\lib` del directorio creado al generar el workspace
- Dar clic sobre el nombre del proyecto y **F5** para actualizar, los archivos aparecerán dentro del directorio **lib** en **WEB-INF**

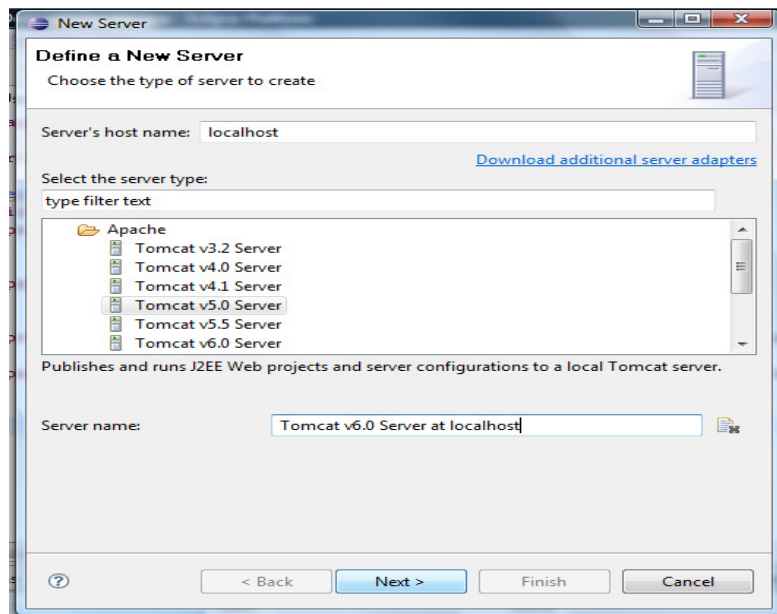


Instalación del servidor Apache Tomcat

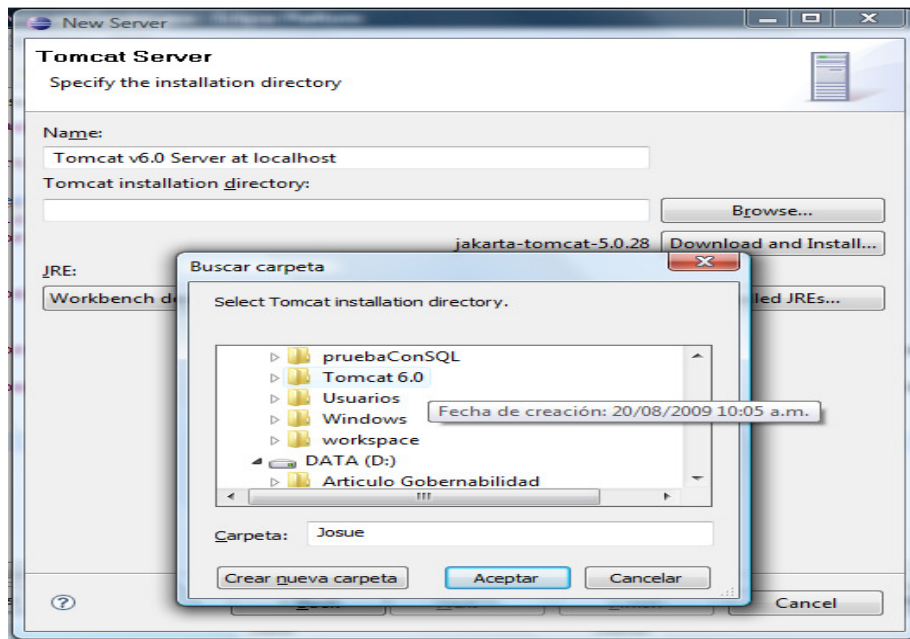
Al tratarse de una aplicación web, es necesario que se ejecute en algún servidor web, en este caso se trabajará con el servidor Apache Tomcat 6.0

Para poder utilizarlo es necesario agregarlo en la lista de servidores disponibles en el proyecto que se encuentra abajo del área de desarrollo, en caso de que no aparezca es necesario visualizarla seleccionando en la barra de menú **Window, Show View y Server**

- Dar clic en la pestaña **Servers**
- Clic derecho y seleccionar **New** y posteriormente **Server**
- Seleccionar en **Apache, Tomcat v6.0 Server**



- Clic en **Next**
- Clic en **Browse**
- Buscar la carpeta en donde esté instalado Tomcat6.0



- Seleccionarla y dar clic en **Aceptar**

El archivo struts-config.xml

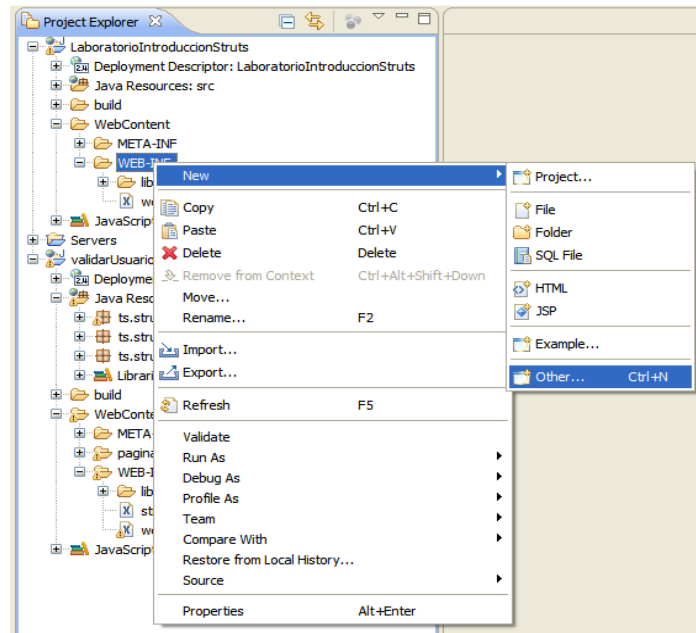
Es necesario crear o agregar el archivo **struts-config.xml** al directorio **WEB-INF**

Para agregarlo (lo más recomendable), se puede trabajar con un archivo **struts-config.xml** de un proyecto en blanco

- Copiar el archivo **struts-config.xml** del directorio en donde se encuentra la carpeta **WebContent\WEB-INF** en el directorio del proyecto
- Actualizar el proyecto (presionando **F5**)

Para crear un archivo en blanco es necesario

- Dar clic derecho sobre el directorio **WEB-INF**
- Seleccionar **New**
- Seleccionar **Other**



- Seleccionar **XML**
- En el campo del nombre de archivo, escribir **struts-config.xml**
- Dar clic en **Finish**

El archivo **struts-config.xml** debe tener el siguiente contenido de manera inicial

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD
Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
  <data-sources />
  <form-beans />
  <global-exceptions />
  <global-forwards />
  <action-mappings />
  <controller bufferSize="4096" debug="0" />
  <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

2. Programación de las clases

La aplicación consistirá en una pantalla de validación en donde un usuario introducirá su *login* y su *password* para que sea validado, de manera inicial el resultado de la validación se mostrará en la Consola.

Es necesario crear los siguientes paquetes en el proyecto

ts.struts.javabeans

ts.struts.modelo

ts.struts.servlets

Clase ValidacionUsuarioForm (JavaBean)

Crear un JavaBean llamado **ValidacionUsuarioForm** que contenga los datos que el usuario introducirá en la pantalla de validación de usuario, en este caso se deben tener los datos miembros para *login* y *password* junto con sus métodos *get/set*.

Para que esta clase pueda ser manejada por Struts, debe heredar a la clase **ActionForm**

Será necesario importar el paquete **org.apache.struts.action.***

```
package ts.struts.javabeans;
import org.apache.struts.action.*;

public class ValidacionUsuarioForm extends ActionForm{

    private String usuario;
    private String password;

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

Registro del JavaBean en el archivo struts-config.xml

Para que se pueda utilizar el JavaBean, es necesario registrarlo agregando las siguientes líneas en el archivo **struts-config.xml**

```
<form-beans >
<form-bean name="ValidacionUsuarioForm"
type="ts.struts.javabeans.ValidacionUsuarioForm"></form-bean>
</form-beans>
```

Clase ValidarAction (Action)

Es la clase encargada de recibir los datos capturados en el formulario de validación, se crea dentro del paquete **ts.struts.servlets**, para crearla se debe hacer como una clase normal, NO como se crea una clase Servlet.

```
package ts.struts.servlets;
import javax.servlet.http.*;
import org.apache.struts.action.*;

import ts.struts.javabeans.*;
import ts.struts.modelo.*;

public class ValidarAction extends Action{

    public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response) throws Exception{

        ValidarUsuario v = new ValidarUsuario();
        ValidacionUsuarioForm vf = (ValidacionUsuarioForm)form;

        if(v.validar(vf)==0)
            System.out.println("Eres un profesor");

        else if(v.validar(vf)==1)
            System.out.println("Eres un alumno");

        else
            System.out.println("Eres un don nadie");
        return null;

    }
}
```


Registro de la clase Action en el archivo struts-config.xml

Se debe registrar la clase **Action** en el archivo **struts-config.xml** agregando la siguiente información al archivo dentro del campo entre las etiquetas **<action-mappings>** y **</action-mappings>**

```
<action-mappings>
<action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction"></action>
</action-mappings>
```

El archivo **struts-config.xml** queda de la siguiente manera

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD
Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
  <data-sources />
  <form-beans >
    <form-bean name="ValidacionUsuarioForm"
type="ts.struts.javabeans.ValidacionUsuarioForm"></form-bean>
  </form-beans>
  <global-exceptions />
  <global-forwards />
  <action-mappings>
    <action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction"></action>
  </action-mappings>
  <controller bufferSize="4096" debug="0" />
  <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

Clase ValidarUsuario (auxiliar)

Dentro del paquete **ts.struts.modelo** se crea una clase llamada **ValidarUsuario** encargada de la validación de los datos, por cuestiones de ejemplo, solo se hará una validación muy sencilla sin acceso a base de datos.

```
package ts.struts.modelo;
import ts.struts.javabeans.ValidacionUsuarioForm;
public class ValidarUsuario {
    public int validar(ValidacionUsuarioForm vf) {

        if(vf.getUsuario().compareTo("profesor")==0)
            return 0;
        else if(vf.getUsuario().compareTo("alumno")==0)
            return 1;
        else
            return 2;
    }
}
```

Registro del servlet Action en el archivo web.xml

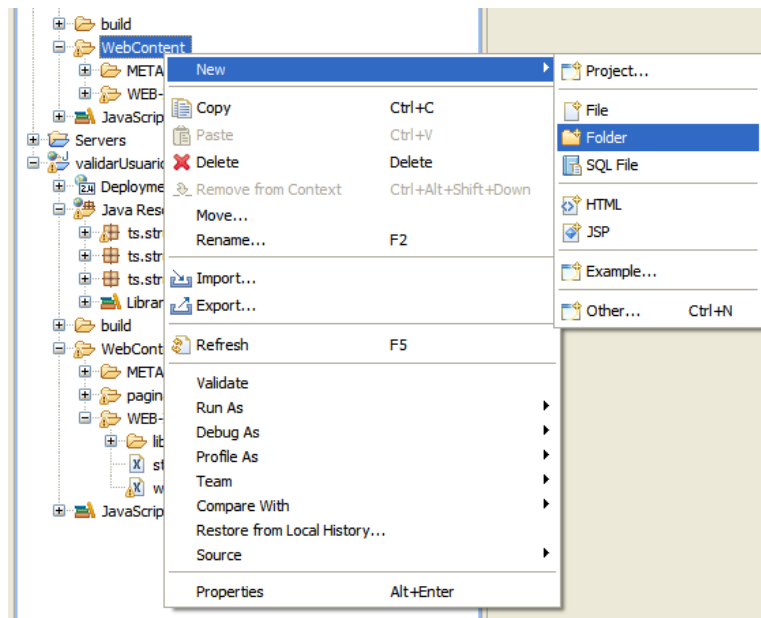
Es necesario registrar el servlet **Action** en el archivo **web.xml** agregando la información necesaria para el manejo de este servlet

Lo más recomendable es utilizar el archivo **web.xml** que se proporciona con los archivos de un proyecto en blanco.

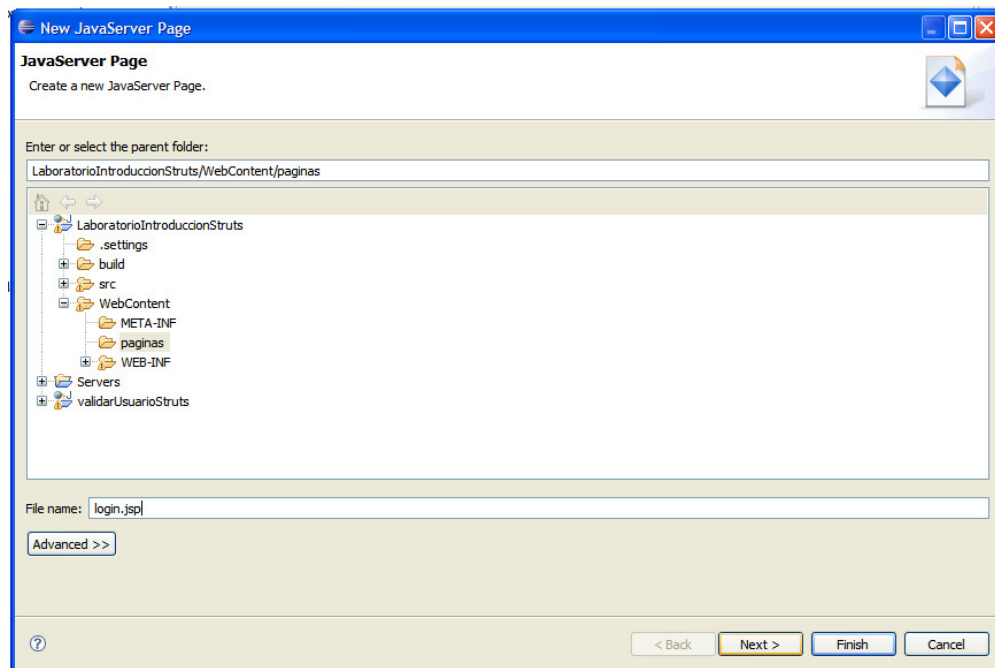
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>LaboratorioIntroduccionStruts</display-name>
    <servlet>
        <servlet-name>action</servlet-name>
        <servlet-class>org.apache.struts.action.ActionServlet</servlet-
class>
        <init-param>
            <param-name>config</param-name>
            <param-value>/WEB-INF/struts-config.xml</param-value>
        </init-param>
        <init-param>
            <param-name>debug</param-name>
            <param-value>2</param-value>
        </init-param>
        <init-param>
            <param-name>detail</param-name>
            <param-value>2</param-value>
        </init-param>
        <load-on-startup>2</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>action</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>30</session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file></welcome-file>
    </welcome-file-list>
</web-app>
```

Programación de la Vista login.jsp

- Dentro de la carpeta WebContent, crear un directorio llamado **paginas**



- Dentro de la carpeta **paginas**, crear un nuevo archivo JSP llamado **login.jsp**



- Dar clic en **Next**
- Seleccionar **New JSP File (html)**
- Dar clic en **Finish**

Para poder utilizar las tags que proporciona Struts en una página JSP, es necesario agregar la siguiente línea

```
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
```

El código de la página **login.jsp** es el siguiente

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html:html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pantalla de validación de usuario</title>
</head>
<body>
<center>
<h1>Validación de usuario</h1>
<html:form action="/validarUsuario" method="POST">
<table>
<tr>
<td>Usuario:</td>
<td><html:text property="usuario"/></td>
</tr>
<tr>
<td>Password:</td>
<td><html:password property="password"/></td>
</tr>
<td colspan="2">
<br/>
<html:submit property="submit" value="Validar"/>
<html:reset value="Limpiar"/>
</td>
</table>
</html:form>
</center>
</body>
</html:html>
```

Probar la aplicación ejecutándola en el servidor