

# Laboratorio. Aplicación de registro y validación

Objetivo.

Realizar una aplicación que permita la validación y el registro de un usuario utilizando el *framework* Struts.

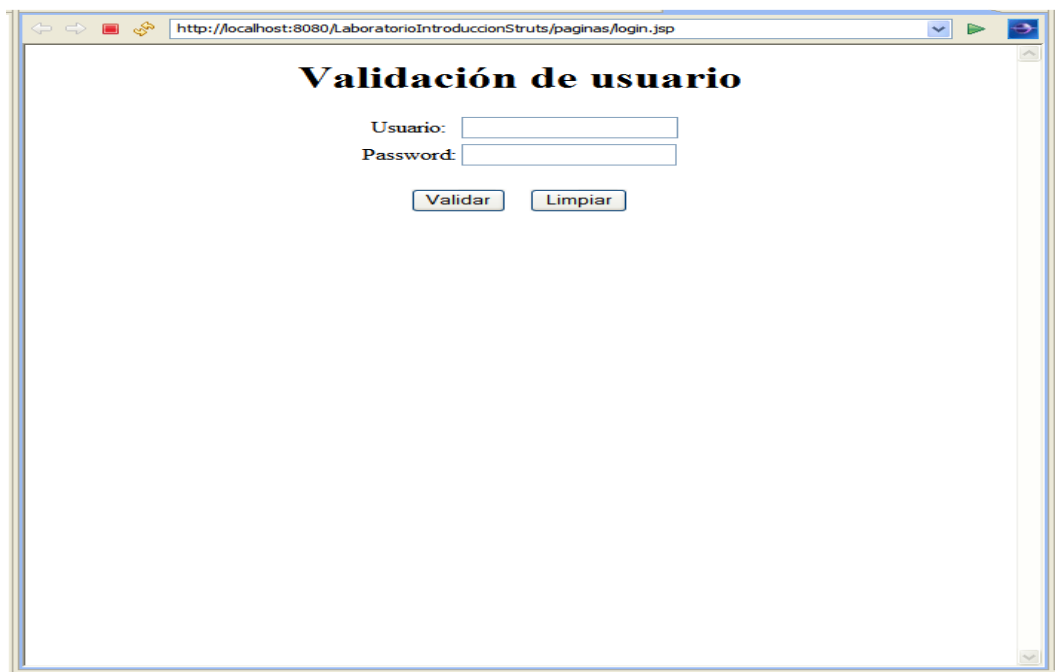
Pasos a seguir.

1. Utilizar el proyecto de validación con el que se trabajó en el Laboratorio anterior
2. Programar y registrar las clases necesarias y desarrollar las vistas necesarias

## 1. Uso del proyecto de validación

Hasta el momento, solo se cuenta con una funcionalidad sencilla para la validación, al finalizar este laboratorio se contará con una aplicación que permita validar y registrar a un nuevo usuario, para este laboratorio, tampoco se considera el uso de base de datos para estas operaciones.

Por el momento, solo se cuenta con la siguiente ventana



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/LaboratorioIntroduccionStruts/paginas/login.jsp`. The main content area features a form titled **Validación de usuario**. The form includes two input fields: 'Usuario:' and 'Password:', each followed by a text box. Below these fields are two buttons: 'Validar' and 'Limpiar'.

## Llamada a una nueva página

Hasta el momento, el resultado de la validación solo se muestra en la Consola, el siguiente paso es crear dos nuevos JSP para la validación, uno llamado **errorValidacion.jsp** y otro **bienvenida.jsp**.

- Agregar en WebContent/paginas/ los archivos **errorValidacion.jsp** y **bienvenida.jsp**

Por el momento, el contenido será muy sencillo

### errorValidacion.jsp

```
<%@ 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>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pantalla de error</title>
</head>
<body>
<h1>Lo siento, tus datos son incorrectos</h1>
</body>
</html>
```

### bienvenida.jsp

```
<%@ 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>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Página de bienvenida</title>
</head>
<body>
<h1>Bienvenido al sistema</h1>
</body>
</html>
```

Hay que indicar en el servlet **ValidarAction** que se solicitará a las diferentes vistas dependiendo el resultado de la validación modificando el código.

## ValidarAction.java

```
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)
            return mapping.findForward("bienvenida");
        else if(v.validar(vf)==1)
            return mapping.findForward("bienvenida");
        else
            return mapping.findForward("errorValidacion");
    }
}
```

Estas acciones, deben registrarse como parte del servlet ValidarAction en el archivo **struts-config.xml**

```
<?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">
            <forward name="bienvenida"
path="/paginas/bienvenida.jsp"></forward>
            <forward name="errorValidacion"
path="/paginas/errorValidacion.jsp"></forward>
        </action>
    </action-mappings>
    <controller bufferSize="4096" debug="0" />
    <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

Probar la aplicación (ejecutando en el servidor el archivo login.jsp)

## Agregar una liga para llamar a una nueva ventana

Se agregará una liga que permita llamar a la ventana que servirá para registrar a un nuevo usuario.

- Abrir el contenido del archivo **login.jsp**

Para colocar una liga hacia otra pantalla, es necesario modificar el código a:

```
<%@ 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>
<html:link forward="registro">Registrarse como nuevo
usuario</html:link>
</center>
</body>
</html:html>
```

Antes de probar la aplicación, es necesario registrar el nombre *registro* para que se sepa hacia donde se va a dirigir la aplicación. Esto se realiza en el archivo **struts-config.xml**.

A pesar de que esta liga se encuentra en el archivo relacionado con el servlet ValidarAction, no está contemplado dentro de sus llamadas (*errorValidacion* y *bienvenida*), por esta razón es que debe registrarse dentro de las etiquetas **<global-forwards>** y **</global-forwards>**

De esta manera 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>
    <forward name="registro"
path="/paginas/nuevoRegistro.jsp"></forward>
  </global-forwards>
  <action-mappings>
    <action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction">
      <forward name="bienvenida"
path="/paginas/bienvenida.jsp"></forward>
      <forward name="errorValidacion"
path="/paginas/errorValidacion.jsp"></forward>
    </action>
  </action-mappings>
  <controller bufferSize="4096" debug="0" />
  <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

Lo siguiente es crear el JSP llamado **nuevoRegistro.jsp**

```
<%@ 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>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Página de registro</title>
</head>
<body>
<h1>Por favor introduce tus datos</h1>
</body>
</html>
```

Probar la aplicación (ejecutando en el servidor el archivo login.jsp)

## Registro de un nuevo usuario

Se debe crear una clase que contenga los datos a leer en la pantalla de registro de usuario, esta clase será un JavaBean por lo que se creará en el paquete correspondiente.

- Crear una clase llamada **RegistroUsuarioForm** en el paquete **ts.struts.javabeans**
- Agregar los datos miembros nombre, apellidoPaterno, apellidoMaterno, teléfono, email, RFC, login y password y sus correspondientes métodos get/set
- Al ser una clase que se utilizará como JavaBean es necesario que herede de la clase **ActionForm** (será necesario importar las bibliotecas necesarias)

### RegistroUsuarioForm.java

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

public class RegistroUsuarioForm extends ActionForm{

    private String nombre;
    private String apellidoPaterno;
    private String apellidoMaterno;
    private String telefono;
    private String email;
    private String login;
    private String password;
    private String RFC;

    public String getNombre() {
        return nombre;
    }
    public String getApellidoPaterno() {
        return apellidoPaterno;
    }
    public String getApellidoMaterno() {
        return apellidoMaterno;
    }
    public String getTelefono() {
        return telefono;
    }
    public String getEmail() {
        return email;
    }
    public String getLogin() {
        return login;
    }
    public String getPassword() {
        return password;
    }
    public String getRFC() {
        return RFC;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public void setApellidoPaterno(String apellidoPaterno) {
        this.apellidoPaterno = apellidoPaterno;
    }
}
```

```

    public void setApellidoMaterno(String apellidoMaterno) {
        this.apellidoMaterno = apellidoMaterno;
    }
    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public void setLogin(String login) {
        this.login = login;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    public void setRFC(String RFC) {
        this.RFC = RFC;
    }
}

```

Al tratarse de una clase JavaBean, será necesario registrarla en el archivo **struts-config.xml**

```

<?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-bean name="RegistroUsuarioForm"
type="ts.struts.javabeans.RegistroUsuarioForm"></form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards>
        <forward name="registro"
path="/paginas/nuevoRegistro.jsp"></forward>
    </global-forwards>
    <action-mappings>
        <action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction">
            <forward name="bienvenida"
path="/paginas/bienvenida.jsp"></forward>
            <forward name="errorValidacion"
path="/paginas/errorValidacion.jsp"></forward>
        </action>
    </action-mappings>
    <controller bufferSize="4096" debug="0" />
    <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>

```

Lo siguiente es crear la clase que manejará los datos que introduzca el usuario, en este caso la clase **RegistrarUsuarioAction.java** en el paquete **ts.struts.servlets**, al recibir un **JavaBean** enviado por un **JSP**, debe extender a la clase **Action** para heredar el método **execute()**

#### RegistrarUsuarioAction.java

```
package ts.struts.servlets;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.*;

import ts.struts.javabeans.RegistroUsuarioForm;
import ts.struts.modelo.RegistrarUsuario;

public class RegistrarUsuarioAction extends Action{

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

        RegistroUsuarioForm rf = (RegistroUsuarioForm)form;

        RegistrarUsuario registro = new RegistrarUsuario();
        registro.registraUsuario(rf);
        return mapping.findForward("registroCompleto");

    }

}
```

Se debe registrar el servlet **RegistrarUsuarioAction** en el archivo **struts-config.xml** indicando las acciones necesarias y, en caso de existir, las llamadas a otras páginas (en este caso se llama a la página referida por **registroCompleto**)

```
<?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-bean name="RegistroUsuarioForm"
type="ts.struts.javabeans.RegistroUsuarioForm"></form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards>
        <forward name="registro"
path="/paginas/nuevoRegistro.jsp"></forward>
    </global-forwards>
    <action-mappings>
        <action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction">
            <forward name="bienvenida"
path="/paginas/bienvenida.jsp"></forward>
        </action>
    </action-mappings>
</struts-config>
```



```

<forward name="errorValidacion"
path="/paginas/errorValidacion.jsp"></forward>
</action>
<action name="RegistroUsuarioForm" path="/registrarUsuario"
scope="request" type="ts.struts.servlets.RegistrarUsuarioAction">
  <forward name="registroCompleto"
path="/paginas/registroCompleto.jsp"></forward>
</action>
</action-mappings>
<controller bufferSize="4096" debug="0" />
<message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>

```

Es necesario crear la clase **RegistrarUsuario** en el paquete **ts.struts.modelo**, esta clase servirá como auxiliar para el registro del nuevo usuario (de momento solo imprimirá los datos en Consola). Después de haber hecho el registro, se llama a una vista (JSP) que indica que el registro ha sido exitoso.

#### RegistrarUsuario.java

```

package ts.struts.modelo;

import ts.struts.DTO.PersonaDTO;
import ts.struts.javabeans.RegistroUsuarioForm;

public class RegistrarUsuario {

    public void registraUsuario(RegistroUsuarioForm rf){

        PersonaDTO persona=convierte(rf);

        System.out.println(persona.getNombre());
        System.out.println(persona.getApellidoPaterno());
        System.out.println(persona.getApellidoMaterno());
        System.out.println(persona.getTelefono());
        System.out.println(persona.getEmail());
        System.out.println(persona.getLogin());
        System.out.println(persona.getPassword());

    }

    public PersonaDTO convierte(RegistroUsuarioForm rf){

        PersonaDTO persona=new PersonaDTO();

        persona.setApellidoMaterno(rf.getApellidoMaterno());
        persona.setApellidoPaterno(rf.getApellidoPaterno());
        persona.setEmail(rf.getEmail());
        persona.setLogin(rf.getLogin());
        persona.setNombre(rf.getNombre());
        persona.setPassword(rf.getPassword());
        persona.setRFC(rf.getTelefono());
        persona.setTelefono(rf.getTelefono());

        return persona;

    }

}

```

Aunque se puede trabajar directamente con los datos del JavaBean `RegistroUsuarioForm`, se crea una clase **PersonaDTO** para trabajar con ella en lo que serían las operaciones con base de datos.

- Agregar el paquete **ts.struts.DTO**
- Crear una clase **PersonaDTO** con los mismos datos miembros que **RegistroUsuarioForm**

#### PersonaDTO.java

```
package ts.struts.DTO;

public class PersonaDTO {

    private String nombre;
    private String apellidoPaterno;
    private String apellidoMaterno;
    private String telefono;
    private String email;
    private String login;
    private String password;
    private String RFC;

    public String getNombre() {
        return nombre;
    }
    public String getApellidoPaterno() {
        return apellidoPaterno;
    }
    public String getApellidoMaterno() {
        return apellidoMaterno;
    }
    public String getTelefono() {
        return telefono;
    }
    public String getEmail() {
        return email;
    }
    public String getLogin() {
        return login;
    }

    public String getPassword() {
        return password;
    }
    public String getRFC() {
        return RFC;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public void setApellidoPaterno(String apellidoPaterno) {
        this.apellidoPaterno = apellidoPaterno;
    }
    public void setApellidoMaterno(String apellidoMaterno) {
        this.apellidoMaterno = apellidoMaterno;
    }
}
```

```

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public void setLogin(String login) {
        this.login = login;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    public void setRFC(String RFC) {
        this.RFC = RFC;
    }
}

```

Lo siguiente es modificar el archivo **nuevoRegistro.jsp** para que pueda leer los campos que el usuario introducirá, aquí se hará uso de las tags thml proporcionadas por Struts.

#### nuevoRegistro.jsp

```

<%@ 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>Página de registro</title>
</head>
<body>
<center>
<h1>Por favor introduce tus datos</h1>
</center>
<html:form action="/registrarUsuario" method="POST">
<table>
<tr>
<td>Nombre:</td>
<td><html:text property="nombre" size="50"></html:text><br><br></td>
</tr>
<tr>
<td>Apellido paterno:</td>
<td><html:text property="apellidoPaterno" size="50"></html:text><br><br></td>
</tr>
<tr>
<td>Apellido materno:</td>
<td><html:text property="apellidoMaterno" size="50"></html:text><br><br></td>
</tr>
<tr>
<td>Teléfono:</td>
<td><html:text property="telefono" size="50"></html:text><br><br></td>

```

```

</tr>
<tr>
<td>e-mail:</td>
<td><html:text property="email" size="50"></html:text><br><br></td>
</tr>
<tr>
<td>RFC:</td>
<td><html:text property="RFC" size="20"></html:text></td>
</tr>
</table>
<br><br><br>
<table>
<tr>
<td>Login:</td>
<td><html:text property="login"></html:text><br><br></td>
</tr>
<tr>
<td>Password:</td>
<td><html:password property="password"></html:password></td>
</tr>
</table>
<br><br>
<html:submit property="submit" value="Registrar"/>
<html:reset value = "Limpiar"/>
</html:form>
</body>
</html:html>

```

Después del registro, se llama a una vista que indica que el resultado fue exitoso y despliega los datos introducidos por el usuario, esta vista es un JSP con el nombre de **registroCompleto.jsp**. Los datos leídos quedan guardados en el objeto ActionForm, por lo que para desplegarlos en otra página, es necesario imprimirlos como datos de un JavaBean, para poder realizar este manejo es necesario incluir la taglib

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

Para poder escribir los datos de un JavaBean, se utiliza la siguiente estructura

```
<bean:write name="NombreBean" property="atributo"/><br>
```

### registroCompleto.jsp

```

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

```

```

<body>
<h1>El registro del usuario ha sido exitoso</h1>
Los datos del usuario son:<br><br>
<bean:write name="RegistroUsuarioForm" property="nombre"/><br>
<bean:write name="RegistroUsuarioForm"
property="apellidoPaterno"/><br>
<bean:write name="RegistroUsuarioForm"
property="apellidoMaterno"/><br>
<bean:write name="RegistroUsuarioForm" property="telefono"/><br>
<bean:write name="RegistroUsuarioForm" property="email"/><br>
<bean:write name="RegistroUsuarioForm" property="RFC"/><br>
<html:link forward="login" >Presionar para volver a la pantalla de
login</html:link>
</body>
</html>

```

Esta vista incluye una llamada a una referencia login, ésta, al no ser parte de operación de algún ActionServlet, debe registrarse en el archivo **struts-config.xml** dentro de los <global-forwards>

```

<?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-bean name="RegistroUsuarioForm"
type="ts.struts.javabeans.RegistroUsuarioForm"></form-bean>
  </form-beans>
  <global-exceptions />
  <global-forwards>
    <forward name="registro"
path="/paginas/nuevoRegistro.jsp"></forward>
    <forward name="login" path="/paginas/login.jsp"></forward>
  </global-forwards>
  <action-mappings>
    <action name="ValidacionUsuarioForm" path="/validarUsuario"
scope="request" type="ts.struts.servlets.ValidarAction">
      <forward name="bienvenida"
path="/paginas/bienvenida.jsp"></forward>
      <forward name="errorValidacion"
path="/paginas/errorValidacion.jsp"></forward>
    </action>
    <action name="RegistroUsuarioForm" path="/registrarUsuario"
scope="request" type="ts.struts.servlets.RegistrarUsuarioAction">
      <forward name="registroCompleto"
path="/paginas/registroCompleto.jsp"></forward>
    </action>
  </action-mappings>
  <controller bufferSize="4096" debug="0" />
  <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>

```

Probar la aplicación (ejecutando en el servidor el archivo login.jsp)

## Validación de un usuario

Hasta ahora la validación solo despliega una pantalla de bienvenida, ahora se modificará esa pantalla para que despliegue los datos de la persona que ha sido validada.

Lo primero será modificar el código de la clase **ValidarUsuario.java** para que regrese un tipo de dato **PersonaDTO** en lugar de un entero, por el momento solo se simula que se elige entre dos login diferentes.

### ValidarUsuario.java

```
package ts.struts.modelo;

import ts.struts.javabeans.ValidacionUsuarioForm;
import ts.struts.DTO.*;

public class ValidarUsuario {

    public PersonaDTO validar(ValidacionUsuarioForm vf) {

        PersonaDTO registro = new PersonaDTO();

        if(vf.getUsuario().compareTo("josue")==0) {

            registro.setNombre("Josue");
            registro.setApellidoPaterno("Figueroa");
            registro.setApellidoMaterno("Gonzalez");
            registro.setEmail("josue.figueroa@gmail.com");
            registro.setTelefono("57899810");
            registro.setRFC("FIGJ780718");

            return registro;

        }

        else if(vf.getUsuario().compareTo("ivonne")==0) {

            registro.setNombre("Ivonne");
            registro.setApellidoPaterno("Figueroa");
            registro.setApellidoMaterno("Gonzalez");
            registro.setEmail("ivonne.figueroa@gmail.com");
            registro.setTelefono("57899810");
            registro.setRFC("FIGI780718");

            return registro;

        }

        else

            return null;

    }

}
```

Estos cambios afectan a la clase **ValidarUsuarioAction**, por lo que deben hacerse las modificaciones necesarias. Esta clase llama a la vista **bienvenida.jsp** que desplegará los datos de un usuario registrado, se desea desplegar un objeto de tipo **RegistroUsuarioForm** por lo que es necesario enviarle el objeto completo, esto se hará enviándole el objeto tipo **RegistroUsuarioForm** que contiene los datos que se desean desplegar a un atributo llamado **usuario**. Para esto se utiliza el siguiente código:

```
request.setAttribute("usuario",uf);
```

En donde **uf** representa un objeto tipo **RegistroUsuarioForm**.

De esta manera el código completo es el siguiente:

#### ValidarUsuarioAction.java

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

import ts.struts.DTO.PersonaDTO;
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;
        PersonaDTO r;

        r=v.validar(vf);

        if(r!=null){
            uf.setNombre(r.getNombre());
            uf.setApellidoPaterno(r.getApellidoPaterno());
            uf.setApellidoMaterno(r.getApellidoMaterno());
            uf.setEmail(r.getEmail());
            uf.setRFC(r.getRFC());
            uf.setTelefono(r.getTelefono());
            uf.setLogin(r.getLogin());
            uf.setPassword(r.getPassword());
            request.setAttribute("usuario",uf);
            return mapping.findForward("bienvenida");
        }
        else
            return mapping.findForward("errorValidacion");
    }
}
```

Lo siguiente es modificar la vista **bienvenida.jsp** para que despliegue los datos correspondientes.

Al hacer el llamado desde la clase ValidarUsuarioAction, solo se tienen disponibles los datos que se leyeron como un ActionForm (login y password), estos se desplegarán como parte del JavaBean por lo que es necesario agregar la taglib que permita el manejo

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

Para imprimir los datos contenidos en un JavaBean se utiliza la siguiente instrucción

```
<bean:write name="ValidacionUsuarioForm" property="usuario"/></i>
```

De manera similar, se recibió un objeto tipo **RegistroUsuarioForm** en un atributo llamado **usuario**, por lo que para imprimir sus valores se debe utilizar la instrucción

```
<bean:write name="usuario" property="nombre_atributo_desplegar"/></i>
```

### **bienvenida.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Página de bienvenida</title>
</head>
<body>
<h1>Bienvenido al sistema</h1>
Tu eres el usuario:
<i><bean:write name="ValidacionUsuarioForm" property="usuario"/></i>
<br><br>
Tu password es:
<i><bean:write name="ValidacionUsuarioForm" property="password"/> </i>

<br><br>
<h2>Tus datos son:</h2>
<bean:write name="usuario" property="nombre"/><br>
<bean:write name="usuario" property="apellidoPaterno"/><br>
<bean:write name="usuario" property="apellidoMaterno"/><br>
<bean:write name="usuario" property="RFC"/><br>
<bean:write name="usuario" property="telefono"/><br>
<bean:write name="usuario" property="email"/><br> </body>
</html>
```

Probar la aplicación (ejecutando en el servidor el archivo login.jsp)