

Laboratorio. Formas dinámicas.

Objetivo. Desarrollar una aplicación utilizando formas dinámicas que despliegue opciones y elementos en pantalla que no se registran directamente en el código de las vistas.

Pasos a seguir.

1. Crear un nuevo proyecto y configurar el entorno de desarrollo.
 2. Desarrollar una vista de validación.
 3. Desarrollar una vista y clases para cargar y desplegar la información de UEAS.
 4. Desarrollar una vista y clases para cargar y desplegar listas de alumnos.
 5. Desarrollar una vista y clases para cargar y desplegar la información de las calificaciones de alumnos.
1. Crear un nuevo proyecto y configurar el entorno de desarrollo.
 - Crear un nuevo proyecto llamado **LaboratorioFormasDinamicas**
 - Configurar el entorno de desarrollo
 - Crear los siguientes paquetes:
 - **ts.struts.servlets**
 - **ts.struts.beans**
 - **ts.struts.DTO**
 - **ts.struts.modelo**
 2. Desarrollo de la vista de validación
 - Crear la clase ValidarUsuarioForm en el paquete **ts.struts.beans**
 - Crear la vista **login.jsp** en la carpeta **WebContent/paginas/**
 - Crear la clase ValidarUsuarioAction en el paquete **ts.struts.servlets**
 - Realizar un forward a la página para seleccionar uea
 - Registrar la información necesaria en el archivo **struts-config.xml**

ValidarUsuarioForm.java

```
package ts.struts.beans;

import org.apache.struts.action.ActionForm;

public class ValidarUsuarioForm extends ActionForm{

    private String login;
    private String passwd;

    public String getLogin() {
        return login;
    }
    public String getPasswd() {
        return passwd;
    }

    public void setLogin(String login) {
        this.login = login;
    }
}
```

```
public void setPasswd(String passwd) {
    this.passwd = passwd;
}
}
```

login.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>Validación de usuario</title>
</head>
<body>
<h1>Introduce tus datos:</h1>

<html:form action="/validarUsuario" method="POST">

Login: <html:text property="login"></html:text><br>
Password: <html:password property="passwd"></html:password><br><br>

<html:submit>Aceptar</html:submit> <html:reset>Limpiar</html:reset>
</html:form>

</body>
</html>
```

seleccionarUea.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">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Selección de UEA</title>
</head>
<body>
<h1>Selecciona la UEA </h1>
</body>
</html>
```

ValidarUsuarioAction.java

```
package ts.struts.servlets;

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

import ts.struts.beans.ValidarUsuarioForm;

public class ValidarUsuarioAction extends Action{

    public ActionForward execute(ActionMapping mapping, ActionForm
form,

    HttpServletRequest request,

    HttpServletResponse response){

        ValidarUsuarioForm uf = new ValidarUsuarioForm();
        uf=(ValidarUsuarioForm) form;

        return mapping.findForward("tablaUeas");
    }
}
```

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="ValidarUsuarioForm"
type="ts.struts.beans.ValidarUsuarioForm"></form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards />
    <action-mappings>
        <action path="/validarUsuario" name="ValidarUsuarioForm"
scope="request" type="ts.struts.servlets.ValidarUsuarioAction">
            <forward name="tablaUeas"
path="/paginas/seleccionarUea.jsp"></forward>
        </action>
    </action-mappings>
    <controller bufferSize="4096" debug="0" />
    <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

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

3. Desarrollo de la vista y clases para cargar y desplegar la información de UEA

Antes del desarrollo de la vista, se deben crear clases de apoyo. La primera es la clase UEA, en la que se registrarán los datos que se encuentran en una uea. Esta clase se crea en el paquete ts.struts.DTO

UEA.java

```
package ts.struts.DTO;

public class UEA {

    private String nombre;
    private String clave;
    private String grupo;
    private String id;
    private String completa;

    public UEA(String nombre, String clave, String grupo){

        this.clave=clave;
        this.nombre=nombre;
        this.grupo=grupo;
        this.id=clave + ','+grupo;
        this.completa=nombre+" (" +grupo +)";
    }

    public String getCompleta() {
        return completa;
    }
    public void setCompleta(String completa) {
        this.completa = completa;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getGrupo() {
        return grupo;
    }
    public void setGrupo(String grupo) {
        this.grupo = grupo;
    }
    public String getNombre() {
        return nombre;
    }
    public String getClave() {
        return clave;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public void setClave(String clave) {
        this.clave = clave;
    }
}
```

Crear un bean llamado ListaUeasForm que contendrá un objeto tipo LinkedList con la lista de las uea a desplegar, esta clase se crea en el paquete **ts.struts.beans**

ListaUeasForm.java

```
package ts.struts.beans;

import java.util.LinkedList;
import org.apache.struts.action.ActionForm;

public class ListaUeasForm extends ActionForm{

    private LinkedList listaDeUeas;

    public LinkedList getListaDeUeas() {
        return listaDeUeas;
    }
    public void setListaDeUeas(LinkedList listaDeUeas) {
        this.listaDeUeas = listaDeUeas;
    }
}
```

Crear una clase en el paquete **ts.struts.modelo** encargada de recuperar toda la información, por el momento solo se tiene un método para llenar la lista de uea.

ObtenerDatos.java

```
package ts.struts.modelo;

import java.util.LinkedList;
import ts.struts.DTO.UEA;

public class ObtenerDatos {

    public LinkedList llenarLista(){

        LinkedList lista = new LinkedList();

        lista.add(new UEA("Almacenamiento y recuperacion de informacion", "1151012", "CCT01"));
        lista.add(new UEA("Software de base", "1151013", "CCT02"));
        lista.add(new UEA("Software de base", "1151013", "CCT81"));
        lista.add(new UEA("Introduccion a la programacion", "1151006", "CCT01"));
        lista.add(new UEA("Introduccion a la computacion", "1151010", "CCT51"));
        lista.add(new UEA("Temas selectos de ingenieria en computacion I", "1151032", "CCT01"));

        return lista;
    }
}
```

Ahora se debe cargar la información de la lista de ueas para pasársela a la vista correspondiente. Esto se realiza en la clase ValidarUsuarioAction que es la clase que realiza el forward hacia la vista seleccionarUea.jsp. Para poder desplegar una lista en una caja de selección, se le debe pasar un objeto tipo Collection, en este caso una LinkedList.

ValidarUsuarioAction.java

```
package ts.struts.servlets;

import java.util.LinkedList;

import javax.servlet.http.*;
import org.apache.struts.action.*;
import ts.struts.beans.*;
import ts.struts.modelo.ObtenerDatos;

public class ValidarUsuarioAction extends Action{

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

        ValidarUsuarioForm uf = new ValidarUsuarioForm();
        uf=(ValidarUsuarioForm) form;

        LinkedList listaUeas = new LinkedList();
        ObtenerDatos obtener = new ObtenerDatos();

        listaUeas = obtener.llenarLista();
        request.setAttribute("lista", listaUeas);

        return mapping.findForward("tablaUeas");
    }
}
```

Modificaciones a la vista seleccionarUea.jsp

Es necesario desarrollar una clase para recibir la opción seleccionada en las diferentes opciones, en este caso la clase **OpcionSeleccionadaForm** en el paquete **ts.struts.beans**.

OpcionSeleccionadaForm.java

```
package ts.struts.beans;

import org.apache.struts.action.ActionForm;

public class OpcionSeleccionadaForm extends ActionForm{

    private String claveSeleccionada;

    public String getClaveSeleccionada() {
        return claveSeleccionada;
    }

    public void setClaveSeleccionada(String claveSeleccionada) {
        this.claveSeleccionada = claveSeleccionada;
    }
}
```

Para desplegar una tabla de selección, en principio se utiliza la misma instrucción que cuando se construye de manera "estática".

```
<html:select property="atributoQueRecibeLaOpcion">
```

Para que cargue la información de la lista, se debe utilizar la instrucción:

```
<html:optionsCollection name="nombre" label="at_1" value="at_2"/>
```

Donde:

- nombre: Nombre del objeto tipo Collection que se le pasó a través del setAttribute() desde el ActionServlet.
- at_1: Nombre del atributo del objeto que se encuentra en la colección que se desea aparezca el nombre.
- at_2: Nombre del atributo del objeto que se encuentra en la colección que se utilizará como identificador.

seleccionarUea.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>Selección de UEA</title>
</head>
<body>
<html:form action="/administrarUea" method="POST">
<h1>Selecciona la UEA </h1>
<html:select property="claveSeleccionada">
<html:optionsCollection name="lista" label="completa" value="id"/>
</html:select>
<html:submit>Aceptar</html:submit>
</html:form>
</body>
</html>
```

Crear la clase **AdministrarUeaAction** en el paquete **ts.struts.servlets**, de momento solo imprime la clave seleccionada.

AdministrarUeaAction.java

```
package ts.struts.servlets;

import javax.servlet.http.*;
import org.apache.struts.action.*;
import ts.struts.beans.OpcionSeleccionadaForm;

public class AdministrarUeaAction extends Action{

    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response){
        OpcionSeleccionadaForm of = new OpcionSeleccionadaForm();
        of = (OpcionSeleccionadaForm)form;
        System.out.println(of.getClaveSeleccionada());
        return null;
    }
}
```

Registrar los nuevos beans y la acción “/administrarUea” en el archivo struts-config.xml.

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="ValidarUsuarioForm"
type="ts.struts.beans.ValidarUsuarioForm"></form-bean>
    <form-bean name="ListaUeasForm"
type="ts.struts.beans.ListaUeasForm"></form-bean>
    <form-bean name="OpcionSeleccionadaForm"
type="ts.struts.beans.OpcionSeleccionadaForm"></form-bean>
  </form-beans>
  <global-exceptions />
  <global-forwards />
  <action-mappings>
    <action path="/validarUsuario" name="ValidarUsuarioForm"
scope="request" type="ts.struts.servlets.ValidarUsuarioAction">
      <forward name="tablaUeas"
path="/paginas/seleccionarUea.jsp"></forward>
    </action>
    <action path="/administrarUea" name="OpcionSeleccionadaForm"
scope="request"
type="ts.struts.servlets.AdministrarUeaAction"></action>
  </action-mappings>
  <controller bufferSize="4096" debug="0" />
  <message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

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

4. Creación de la vista para elegir alumno.

Es necesario crear una clase que contenga la información de los alumnos, esta clase se creará en el paquete **ts.struts.DTO**.

Alumno.java

```
package ts.struts.DTO;

public class Alumno {

    private String nombre;
    private String apellidoPaterno;
    private String apellidoMaterno;
    private String matricula;
    private String nombreCompleto;

    public Alumno(String nombre, String apellidoPaterno,
        String apellidoMaterno, String matricula){
        this.nombre=nombre;
        this.apellidoPaterno=apellidoPaterno;
        this.apellidoMaterno=apellidoMaterno;
        this.matricula=matricula;
        this.nombreCompleto=nombre + " " + apellidoPaterno + " " +
apellidoMaterno;
    }
    public String getNombreCompleto() {
        return nombreCompleto;
    }
    public void setNombreCompleto(String nombreCompleto) {
        this.nombreCompleto = nombreCompleto;
    }
    public String getNombre() {
        return nombre;
    }
    public String getApellidoPaterno() {
        return apellidoPaterno;
    }
    public String getApellidoMaterno() {
        return apellidoMaterno;
    }
    public String getMatricula() {
        return matricula;
    }
    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 setMatricula(String matricula) {
        this.matricula = matricula;
    }
}
```

Antes de desarrollar las clases, es necesario considerar los elementos a desplegar en pantalla, para elegir un alumno se requiere desplegar:

- El nombre y el grupo de la uea
- La lista de alumnos
- La clave que identifique al alumno seleccionado.
- Adicionalmente es necesario tener presente la clave de la uea (formada por el grupo y la clave misma de la uea)

De esta manera la forma (bean) que se manejará en la vista debe contener estos elementos. En especial se debe considerar que para la consulta de calificaciones, se necesita en particular:

- La clave y el grupo de la uea
- La clave del alumno elegido

Crear la clase **SeleccionAlumnoForm** en el paquete **ts.struts.beans**

SeleccionAlumnoform.java

```
package ts.struts.beans;

import org.apache.struts.action.ActionForm;

import ts.struts.DTO.UEA;

public class SeleccionAlumnoForm extends ActionForm{

    private UEA uea;
    private String alumnoSeleccionado;
    private String claveUea;

    public String getClaveUea() {
        return claveUea;
    }
    public UEA getUea() {
        return uea;
    }
    public String getAlumnoSeleccionado() {
        return alumnoSeleccionado;
    }
    public void setAlumnoSeleccionado(String alumnoSeleccionado) {
        this.alumnoSeleccionado = alumnoSeleccionado;
    }
    public void setUea(UEA uea) {
        this.uea = uea;
    }
    public void setClaveUea(String claveUea) {
        this.claveUea = claveUea;
    }
}
```

Lo siguiente es cargar la lista de alumnos dependiendo la clave y grupo de la uea recibida, estos datos se tienen en el servlet AdministrarUeaAction. Es necesario cargar la lista de

alumnos y pasarle a la vista correspondiente esta lista y la información de los otros elementos a desplegar.

Para realizar la carga, es necesario realizar modificaciones a la clase ObtenerDatos

ObtenerDatos.java

```
package ts.struts.modelo;

import java.util.LinkedList;
import java.util.StringTokenizer;
import ts.struts.DTO.Alumno;
import ts.struts.DTO.UEA;

public class ObtenerDatos {

    public LinkedList llenarLista(){

        LinkedList lista = new LinkedList();
        lista.add(new UEA("Almacenamiento y recuperacion de
informacion", "1151012", "CCT01"));
        lista.add(new UEA("Software de base", "1151013", "CCT02"));
        lista.add(new UEA("Software de base", "1151013", "CCT81"));
        lista.add(new UEA("Introduccion a la
programacion", "1151006", "CCT01"));
        lista.add(new UEA("Introduccion a la
computacion", "1151010", "CCT51"));
        lista.add(new UEA("Temas selectos de ingenieria en
computacion I", "1151032", "CCT01"));

        return lista;

    }

    public UEA obtenerUea(String clave){

        StringTokenizer st = new StringTokenizer(clave, ",");
        UEA uea;
        String c = st.nextToken();
        String g = st.nextToken();
        if(c.compareTo("1151012")==0)
            return new UEA("Almacenamiento y recuperacion de
informacion", c, g);
        else if (c.compareTo("1151013")==0)
            return new UEA("Software de base", c, g);
        else if (c.compareTo("1151006")==0)
            return new UEA("Introduccion a la programacion", c, g);
        else if (c.compareTo("1151010")==0)
            return new UEA("Introduccion a la computacion", c, g);
        else
            return new UEA("Temas selectos de ingenieria en
computacion I", c, g);
    }

    public LinkedList llenarListaAlumnos(String claveCompleta){

        LinkedList listaAlumnos = new LinkedList();
        StringTokenizer st = new StringTokenizer(claveCompleta, ",");
        String clave = st.nextToken();
        String grupo = st.nextToken();
```

```

        if(clave.compareTo("1151012")==0){
            listaAlumnos.add(new Alumno("ADRIANO", "RIVAS", "CLAUDIA
MARIBEL", "204317362"));
            listaAlumnos.add(new Alumno("ALVARADO", "VILLAFRANCA",
"USIEL", "204305454"));
            listaAlumnos.add(new Alumno("ALVAREZ", "HERNANDEZ",
"ALBERTO", "204300022"));
        }

        else if(clave.compareTo("1151013")==0 &&
grupo.compareTo("CCT02")==0){
            listaAlumnos.add(new Alumno("AREVALO", "DAVILA", "VERONICA
VERENA", "204300608"));
            listaAlumnos.add(new Alumno("CABRAL", "GUZMAN", "LUIS
MIGUEL", "205300732"));
            listaAlumnos.add(new Alumno("CANTOR", "PERALTA", "JOSE
LUIS", "205200932"));
        }

        else if(clave.compareTo("1151013")==0 &&
grupo.compareTo("CCT81")==0){
            listaAlumnos.add(new Alumno("VARELA", "PEÑA", "STEPHANIE",
"205307831"));
            listaAlumnos.add(new Alumno("VAZQUEZ", "LEAL", "EDGAR
ISMAEL", "206200888"));
            listaAlumnos.add(new Alumno("VIDAL", "MATUS", "PEDRO
ALBERTO", "203304934"));
        }

        else if(clave.compareTo("1151006")==0){
            listaAlumnos.add(new Alumno("CERNA", "LANDA",
"XIMENA", "204305797"));
            listaAlumnos.add(new Alumno("CONTRERAS", "DORIA", "JAVIER
EDUARDO", "206201884"));
            listaAlumnos.add(new Alumno("CRUZ", "RUIZ", "ISABEL",
"204358619"));
        }

        else if(clave.compareTo("1151010")==0){
            listaAlumnos.add(new Alumno("GARCIA", "HERNANDEZ",
"DAVID", "206304373"));
            listaAlumnos.add(new Alumno("GARCIA", "RIOS", "AIDE
WENDY", "205305839"));
            listaAlumnos.add(new Alumno("GODINEZ", "RAMIREZ", "ANGEL
AKBAL", "205202196"));
        }

        else {
            listaAlumnos.add(new Alumno("HARO", "GARCIA", "RICARDO",
"205202154"));
            listaAlumnos.add(new Alumno("HERNANDEZ", "MARTINEZ", "JUAN
CARLOS", "205202340"));
            listaAlumnos.add(new Alumno("HERNANDEZ", "ZAMORANO",
"VICTOR HUGO", "204307278"));
        }

        return listaAlumnos;
    }
}

```

AdministrarUeaAction.java

```
package ts.struts.servlets;

import java.util.LinkedList;
import javax.servlet.http.*;
import org.apache.struts.action.*;
import ts.struts.beans.*;
import ts.struts.modelo.ObtenerDatos;

public class AdministrarUeaAction extends Action{

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

        OpcionSeleccionadaForm of = new OpcionSeleccionadaForm();
        of = (OpcionSeleccionadaForm) form;

        SeleccionAlumnoForm af = new SeleccionAlumnoForm();
        ObtenerDatos obtener = new ObtenerDatos();
        LinkedList listaAlumnos = new LinkedList();

        af.setUea(obtener.obtenerUea(of.getClaveSeleccionada()));
        af.setClaveUea(of.getClaveSeleccionada());
        listaAlumnos =
obtener.llenarListaAlumnos(of.getClaveSeleccionada());

        request.setAttribute("datosUea", af);
        request.setAttribute("listaAlumnos", listaAlumnos);

        return mapping.findForward("seleccionarAlumno");
    }
}
```

Desarrollo de la vista.

Se le envió a través del servlet AdministrarUeaAction un objeto de tipo SeleccionAlumnoForm y una LinkedList a la vista seleccionarAlumno.jsp.

En la vista se desea desplegar la información de la UEA que se ha elegido, esto es con la instrucción:

```
<h1>UEA:</h1><h2><bean:write name="datosUea"
property="uea.nombre"></bean:write></h2>
<h1>Grupo:</h1><h2><bean:write name="datosUea"
property="uea.grupo"></bean:write> </h2>
```

En este caso, la propiedad a desplegar corresponde a un objeto UEA cuyo atributo es nombre, por esa razón se utiliza la notación punto (objeto.atributo)

También se debe considerar la clave de la uea seleccionada, sin embargo, esta no se quiere visualizar, solo que la forma la recuerde, esto se realiza con la instrucción:

```
<h1><html:hidden name="datosUea" property="claveUea"/></h1>
```

Para desplegar la lista de alumnos se utiliza la misma instrucción que para el despliegue de las uea.

seleccionarAlumno.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>Selección de alumno</title>
</head>
<body>
<h1>Elija al alumno del que desea consultar o modificar
calificaciones</h1>
<html:form action="/administrarAlumno" method="POST">

<h1>UEA:</h1><h2><bean:write name="datosUea"
property="uea.nombre"></bean:write></h2>
<h1>Grupo:</h1><h2><bean:write name="datosUea"
property="uea.grupo"></bean:write> </h2>

<h1><html:hidden name="datosUea" property="claveUea"/></h1>
<br><br>
<h1>Lista de alumnos</h1>
<h2>Selecciona el alumno del que deseas consultar o modificar
calificaciones</h2>
<html:select property="alumnoSeleccionado">

<html:optionsCollection name="listaAlumnos" label="nombreCompleto"
value="matricula"/>

</html:select>
<html:submit>Aceptar</html:submit>
</html:form>
</body>
</html>
```

Crear la clase AdministrarAlumnoAction en el paquete **ts.struts.servlets**, de momento solo imprimirá la clave de la uea y el identificador (matrícula) del alumno seleccionado.

AdministrarAlumnoAction.java

```
package ts.struts.servlets;

import javax.servlet.http.*;
import org.apache.struts.action.*;
import ts.struts.beans.SeleccionAlumnoForm;
```

```

public class AdministrarAlumnoAction extends Action{

    public ActionForward execute(ActionMapping mapping, ActionForm
form,

    HttpServletRequest request,

    HttpServletResponse response){

        SeleccionAlumnoForm af = new SeleccionAlumnoForm();
        af = (SeleccionAlumnoForm) form;

        System.out.println("La clave es: " + af.getClaveUea());
        System.out.println("La matricula es: " +
af.getAlumnoSeleccionado());

        return null;

    }

}

```

Realizar los mapeos necesarios en el archivo struts-config.xml

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="ValidarUsuarioForm"
type="ts.struts.beans.ValidarUsuarioForm"></form-bean>
        <form-bean name="ListaUeasForm"
type="ts.struts.beans.ListaUeasForm"></form-bean>
        <form-bean name="OpcionSeleccionadaForm"
type="ts.struts.beans.OpcionSeleccionadaForm"></form-bean>
        <form-bean name="SeleccionAlumnoForm"
type="ts.struts.beans.SeleccionAlumnoForm"></form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards />
    <action-mappings>
        <action path="/validarUsuario" name="ValidarUsuarioForm"
scope="request" type="ts.struts.servlets.ValidarUsuarioAction">
            <forward name="tablaUeas"
path="/paginas/seleccionarUea.jsp"></forward>
        </action>
        <action path="/administrarUea" name="OpcionSeleccionadaForm"
scope="request" type="ts.struts.servlets.AdministrarUeaAction">
            <forward name="seleccionarAlumno"
path="/paginas/seleccionarAlumno.jsp"></forward>
        </action>
        <action path="/administrarAlumno" name="SeleccionAlumnoForm"
scope="request" type="ts.struts.servlets.AdministrarAlumnoAction">
        </action>
    </action-mappings>

```

```
<controller bufferSize="4096" debug="0" />
<message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

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

5. Desplegar las calificaciones de los alumnos

Se considera que se registran calificaciones de exámenes y programas para cada alumno, por esto es necesario crear las clases correspondientes en el paquete **ts.struts.DTO**.

Examen.java

```
package ts.struts.DTO;

public class Examen {

    private int numero;
    private float calificacion;
    public Examen(int n, float c){

        this.numero=n;
        this.calificacion=c;
    }
    public int getNumero() {
        return numero;
    }
    public float getCalificacion() {
        return calificacion;
    }
    public void setNumero(int numero) {
        this.numero = numero;
    }
    public void setCalificacion(float calificacion) {
        this.calificacion = calificacion;
    }
}
```

Programa.java

```
package ts.struts.DTO;

public class Programa {

    private int numero;
    private float calificacion;
    public Programa(int n, float c){

        this.numero=n;
        this.calificacion=c;
    }
    public int getNumero() {
        return numero;
    }
    public float getCalificacion() {
        return calificacion;
    }
    public void setNumero(int numero) {
        this.numero = numero;
    }
    public void setCalificacion(float calificacion) {
        this.calificacion = calificacion;
    }
}
```

Es necesario implementar métodos que permitan “cargar” las calificaciones de un alumno, esto se hace en la clase ObtenerDatos del paquete **ts.struts.modelo**.

ObtenerDatos.java

```
package ts.struts.modelo;

import java.util.LinkedList;
import java.util.StringTokenizer;

import ts.struts.DTO.Alumno;
import ts.struts.DTO.Examen;
import ts.struts.DTO.Programa;
import ts.struts.DTO.UEA;

public class ObtenerDatos {

    public LinkedList llenarLista(){

        LinkedList lista = new LinkedList();
        lista.add(new UEA("Almacenamiento y recuperacion de
informacion", "1151012", "CCT01"));
        lista.add(new UEA("Software de base", "1151013", "CCT02"));
        lista.add(new UEA("Software de base", "1151013", "CCT81"));
        lista.add(new UEA("Introduccion a la
programacion", "1151006", "CCT01"));
        lista.add(new UEA("Introduccion a la
computacion", "1151010", "CCT51"));
        lista.add(new UEA("Temas selectos de ingenieria en
computacion I", "1151032", "CCT01"));

        return lista;
    }

    public UEA obtenerUea(String clave){

        StringTokenizer st = new StringTokenizer(clave, ",");
        UEA uea;

        String c = st.nextToken();
        String g = st.nextToken();

        if(c.compareTo("1151012")==0)
            return new UEA("Almacenamiento y recuperacion de
informacion", c, g);
        else if (c.compareTo("1151013")==0)
            return new UEA("Software de base", c, g);
        else if (c.compareTo("1151006")==0)
            return new UEA("Introduccion a la programacion", c, g);
        else if (c.compareTo("1151010")==0)
            return new UEA("Introduccion a la computacion", c, g);
        else
            return new UEA("Temas selectos de ingenieria en
computacion I", c, g);
    }

    public LinkedList llenarListaAlumnos(String claveCompleta){

        LinkedList listaAlumnos = new LinkedList();
        StringTokenizer st = new StringTokenizer(claveCompleta, ",");
        String clave = st.nextToken();
```

```

String grupo = st.nextToken();

    if(clave.compareTo("1151012")==0){
        listaAlumnos.add(new Alumno("ADRIANO", "RIVAS", "CLAUDIA
MARIBEL", "204317362"));
        listaAlumnos.add(new Alumno("ALVARADO", "VILLAFRANCA",
"USIEL", "204305454"));
        listaAlumnos.add(new Alumno("ALVAREZ", "HERNANDEZ",
"ALBERTO", "204300022"));
    }

    else if(clave.compareTo("1151013")==0 &&
grupo.compareTo("CCT02")==0){
        listaAlumnos.add(new Alumno("AREVALO", "DAVILA", "VERONICA
VERENA", "204300608"));
        listaAlumnos.add(new Alumno("CABRAL", "GUZMAN", "LUIS
MIGUEL", "205300732"));
        listaAlumnos.add(new Alumno("CANTOR", "PERALTA", "JOSE
LUIS", "205200932"));
    }

    else if(clave.compareTo("1151013")==0 &&
grupo.compareTo("CCT81")==0){
        listaAlumnos.add(new Alumno("VARELA", "PEÑA", "STEPHANIE",
"205307831"));
        listaAlumnos.add(new Alumno("VAZQUEZ", "LEAL", "EDGAR
ISMAEL", "206200888"));
        listaAlumnos.add(new Alumno("VIDAL", "MATUS", "PEDRO
ALBERTO", "203304934"));
    }

    else if(clave.compareTo("1151006")==0){
        listaAlumnos.add(new Alumno("CERNA", "LANDA",
"XIMENA", "204305797"));
        listaAlumnos.add(new Alumno("CONTRERAS", "DORIA", "JAVIER
EDUARDO", "206201884"));
        listaAlumnos.add(new Alumno("CRUZ", "RUIZ", "ISABEL",
"204358619"));
    }

    else if(clave.compareTo("1151010")==0){
        listaAlumnos.add(new Alumno("GARCIA", "HERNANDEZ",
"DAVID", "206304373"));
        listaAlumnos.add(new Alumno("GARCIA", "RIOS", "AIDE
WENDY", "205305839"));
        listaAlumnos.add(new Alumno("GODINEZ", "RAMIREZ", "ANGEL
AKBAL", "205202196"));
    }

    else {
        listaAlumnos.add(new Alumno("HARO", "GARCIA", "RICARDO",
"205202154"));
        listaAlumnos.add(new Alumno("HERNANDEZ", "MARTINEZ", "JUAN
CARLOS", "205202340"));
        listaAlumnos.add(new Alumno("HERNANDEZ", "ZAMORANO",
"VICTOR HUGO", "204307278"));
    }

    return listaAlumnos;
}

```

```

public Alumno obtenerAlumno(String mat){
    Alumno alumno;
    return new Alumno("NOMBRE", "PATERNO", "MATERNO", mat);
}

public LinkedList examenes(String mat, String clave){
    int nExamenes=3;
    LinkedList ex = new LinkedList();
    for(int i=0;i<nExamenes;i++)
        ex.add(new Examen(i+01,0.0F));
    return ex;
}

public LinkedList programas(String mat, String clave){
    int nProgramas=7;
    LinkedList prog = new LinkedList();
    for(int i=0;i<nProgramas;i++)
        prog.add(new Programa(i+1,0.0F));
    return prog;
}
}

```

Crear un form que contenga la información a desplegar en la vista de calificaciones, esta clase debe crearse en el paquete ts.struts.beans y contener la siguiente información:

- Número de exámenes
- Número de programas
- Datos de la UEA
- Datos del alumno
- Clave de la UEA
- Clave del alumno
- Lista de calificaciones de exámenes
- Lista de calificaciones de programas

ListaCalificacionesForm.java

```

package ts.struts.beans;

import java.util.LinkedList;

import ts.struts.DTO.Alumno;
import ts.struts.DTO.Examen;
import ts.struts.DTO.Programa;
import ts.struts.DTO.UEA;

public class ListaCalificacionesForm extends ActionForm{

```

```

private int nExámenes;
private int nProgramas;
private UEA uea;
private Alumno alumno;
private String claveUea;
private String matricula;
private LinkedList listaExámenes;
private LinkedList listaProgramas;

public int getNExámenes() {
    return nExámenes;
}
public int getNProgramas() {
    return nProgramas;
}
public UEA getUea() {
    return uea;
}
public Alumno getAlumno() {
    return alumno;
}
public String getClaveUea() {
    return claveUea;
}
public String getMatricula() {
    return matricula;
}
public LinkedList getListaExámenes() {
    return listaExámenes;
}
public LinkedList getListaProgramas() {
    return listaProgramas;
}
public void setNExámenes(int exámenes) {
    nExámenes = exámenes;
}
public void setNProgramas(int programas) {
    nProgramas = programas;
}
public void setUea(UEA uea) {
    this.uea = uea;
}
public void setAlumno(Alumno alumno) {
    this.alumno = alumno;
}
public void setClaveUea(String claveUea) {
    this.claveUea = claveUea;
}
public void setMatricula(String matricula) {
    this.matricula = matricula;
}
public void setListaExámenes(LinkedList listaExámenes) {
    this.listaExámenes = listaExámenes;
}
public void setListaProgramas(LinkedList listaProgramas) {
    this.listaProgramas = listaProgramas;
}
}

```

```

    public Examen getExamen(int i){

        if(this.listaExamenes==null)
            this.listaExamenes=new LinkedList();

        while(i >=this.listaExamenes.size())
            this.listaExamenes.add(new Examen(0,0));
        return (Examen)listaExamenes.get(i);

    }

    public Programa getPrograma(int i){

        if(this.listaProgramas==null)
            this.listaProgramas=new LinkedList();

        while(i >=this.listaProgramas.size())
            this.listaProgramas.add(new Programa(0,0));
        return (Programa)listaProgramas.get(i);

    }

}

```

Las calificaciones se pasan a la vista a través de la clase AdministrarAlumnoAction, por lo que hay que llenar cada uno de los elementos que se desplegarán en la vista.

AdministrarAlumnoAction.java

```

package ts.struts.servlets;

import javax.servlet.http.*;

import org.apache.struts.action.*;
import ts.struts.beans.*;
import ts.struts.modelo.ObtenerDatos;

public class AdministrarAlumnoAction extends Action{

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

        SeleccionAlumnoForm af = new SeleccionAlumnoForm();
        ListaCalificacionesForm cf = new ListaCalificacionesForm();
        ObtenerDatos obtener = new ObtenerDatos();

        af = (SeleccionAlumnoForm) form;

        cf.setUea(obtener.obtenerUea(af.getClaveUea()));

        cf.setAlumno(obtener.obtenerAlumno(af.getAlumnoSeleccionado()));
        cf.setClaveUea(af.getClaveUea());
        cf.setMatricula(af.getAlumnoSeleccionado());
        cf.setListaExamenes(obtener.examenes(af.getAlumnoSeleccionado(),
        af.getClaveUea()));
    }
}

```

```

        cf.setListaProgramas(obtener.programas(af.getAlumnoSeleccionado(
    ), af.getClaveUea()));
        cf.setNExámenes(cf.getListExámenes().size());
        cf.setNProgramas(cf.getListProgramas().size());
        request.setAttribute("datosForma", cf);
        return mapping.findForward("calificaciones");
    }
}

```

Desarrollo de la vista

calificacionesAlumno.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<!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>Calificaciones del alumno</title>
</head>
<body>

<html:form action="/registrarCalificaciones" method="POST">
<h2>UEA: <bean:write name="datosForma" property="uea.nombre"/></h2>
<h2>Grupo: <bean:write name="datosForma" property="uea.grupo"/></h2>
<h2>Alumno: <bean:write name="datosForma"
property="alumno.nombreCompleto"/></h2>

<html:hidden name="datosForma" property="matricula"/>
<html:hidden name="datosForma" property="claveUea"/>

Calificaciones hasta el momento:<br><br>

Exámenes:<bean:write name="datosForma" property="NExámenes"/><br><br>

<logic:iterate name="datosForma" property="listaExámenes" id="examen">
<html:hidden name="examen" property="numero" indexed="true"/>
Examen <bean:write name="examen" property="numero"/>
<html:text name="examen" property="calificacion" size="3"
indexed="true"></html:text><br>
</logic:iterate>

<br><br>Programas:<bean:write name="datosForma"
property="NProgramas"/><br><br>

<logic:iterate name="datosForma" property="listaProgramas"
id="programa">
<html:hidden name="programa" property="numero" indexed="true"/>

```

```

Programa <bean:write name="programa" property="numero"/>
<html:text name="programa" property="calificacion" size="3"
indexed="true"></html:text><br>
</logic:iterate>

<br><br><html:submit>Aceptar</html:submit>
</html:form>
</body>
</html>

```

Crear la clase RegistrarCalificacionesAction en el paquete **ts.struts.servlets**, de momento solo imprimirá las calificaciones modificadas en consola y desplegará una pantalla que indique que el proceso ha terminado.

RegistrarCalificacionesAction.java

```

import java.util.LinkedList;

import javax.servlet.http.*;
import org.apache.struts.action.*;
import ts.struts.DTO.*;
import ts.struts.beans.ListaCalificacionesForm;

public class RegistrarCalificacionesAction extends Action{

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

        ListaCalificacionesForm cf = (ListaCalificacionesForm)form;
        System.out.println(cf.getClaveUea());
        System.out.println(cf.getMatricula());

        LinkedList listaExamenenes = cf.getListaExamenenes();
        LinkedList listaProgramas = cf.getListaProgramas();

        for(int i=0;i<listaExamenenes.size();i++){
            Examen e = (Examen)listaExamenenes.get(i);
            System.out.println(e.getNumero() + "->" + e.getCalificacion());
        }
        System.out.println("Calificaciones de programas");

        for(int i=0;i<listaProgramas.size();i++){
            Programa p = (Programa)listaProgramas.get(i);
            System.out.println(p.getNumero() + "->" + p.getCalificacion());
        }
        return mapping.findForward("registroCorrecto");
    }
}

```


Finalmente se creará una pantalla sencilla de confirmación.

registroCorrecto.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>Registro correcto</title>
</head>
<body>
<h1>Las calificaciones han sido registradas de manera correcta</h1>
</body>
</html>
```

Realizar los mapeos correspondientes en el archivo **struts-config.xml**

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="ValidarUsuarioForm"
type="ts.struts.beans.ValidarUsuarioForm"></form-bean>
        <form-bean name="ListaUeasForm"
type="ts.struts.beans.ListaUeasForm"></form-bean>
        <form-bean name="OpcionSeleccionadaForm"
type="ts.struts.beans.OpcionSeleccionadaForm"></form-bean>
        <form-bean name="SeleccionAlumnoForm"
type="ts.struts.beans.SeleccionAlumnoForm"></form-bean>
        <form-bean name="ListaCalificacionesForm"
type="ts.struts.beans.ListaCalificacionesForm"></form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards />
    <action-mappings>
        <action path="/validarUsuario" name="ValidarUsuarioForm"
scope="request" type="ts.struts.servlets.ValidarUsuarioAction">
            <forward name="tablaUeas"
path="/paginas/seleccionarUea.jsp"></forward>
        </action>
        <action path="/administrarUea" name="OpcionSeleccionadaForm"
scope="request" type="ts.struts.servlets.AdministrarUeaAction">
            <forward name="seleccionarAlumno"
path="/paginas/seleccionarAlumno.jsp"></forward>
        </action>
        <action path="/administrarAlumno" name="SeleccionAlumnoForm"
scope="request" type="ts.struts.servlets.AdministrarAlumnoAction">
            <forward name="calificaciones"
path="/paginas/calificacionesAlumno.jsp"></forward>
```

```
</action>
  <action path="/registrarCalificaciones"
name="ListaCalificacionesForm" scope="request "
type="ts.struts.servlets.RegistrarCalificacionesAction">
  <forward name="registroCorrecto"
path="/paginas/registroCorrecto.jsp"></forward>
  </action>
</action-mappings>
<controller bufferSize="4096" debug="0" />
<message-resources
parameter="com.yourcompany.struts.ApplicationResources" />
</struts-config>
```

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

Ligando vistas

En este caso no es tan sencillo colocar ligas entre las diferentes páginas, debido a que se generan de manera dinámica.

En la vista seleccionarUea se requiere una liga sencilla para regresar a la pantalla de login.

seleccionarUea.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>Selección de UEA</title>
</head>
<body>
<html:form action="/administrarUea" method="POST">
<h1>Selecciona la UEA </h1>
<html:select property="claveSeleccionada">
<html:optionsCollection name="lista" label="completa" value="id"/>
</html:select>
<html:submit>Aceptar</html:submit>
</html:form>
<html:link forward = "login">Presiona para volver a la pantalla de
login</html:link>
</body>
</html>
```

Esto se debe registrar en la parte correspondiente en el archivo **struts-config.xml**

```
<global-forwards>
<forward name="login" path="/paginas/login.jsp"></forward>
</global-forwards>
```

En caso de que en la vista **seleccionarAlumno** se quisiera regresar a la vista de selección de uea, no es posible colocar una liga directa a la página, ya que al crearse de manera dinámica, primero es necesario llamar a la acción que llena la información que la vista necesita. Por esa razón es necesario llamar al servlet que llena esa información, en este caso el **ValidarUsuarioAction**.

Para realizar el mapeo, este se debe realizar a una acción, con la instrucción

```
<html:link action="/validarUsuario">Presiona para seleccionar otra
UEA</html:link>
```

seleccionarAlumno.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>Selección de alumno</title>
</head>
<body>
<h1>Elija al alumno del que desea consultar o modificar
calificaciones</h1>
<html:form action="/administrarAlumno" method="POST">

<h1>UEA:</h1><h2><bean:write name="datosUea"
property="uea.nombre"></bean:write></h2>
<h1>Grupo:</h1><h2><bean:write name="datosUea"
property="uea.grupo"></bean:write> </h2>

<h1><html:hidden name="datosUea" property="claveUea"/></h1>
<br><br>
<h1>Lista de alumnos</h1>
<h2>Selecciona el alumno del que deseas consultar o modificar
calificaciones</h2>
<html:select property="alumnoSeleccionado">

<html:optionsCollection name="listaAlumnos" label="nombreCompleto"
value="matricula"/>
</html:select>
<html:submit>Aceptar</html:submit>
</html:form>
<html:link action="/validarUsuario">Presiona para seleccionar otra
UEA</html:link>
</body>
</html>
```

En la página de despliegue/modificación de calificaciones, se colocará una liga para elegir una nueva uea y una para elegir un nuevo alumno.

calificacionesAlumno.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<!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>Calificaciones del alumno</title>
```

```

</head>
<body>

<html:form action="/registrarCalificaciones" method="POST">
<h2>UEA: <bean:write name="datosForma" property="uea.nombre"/></h2>
<h2>Grupo: <bean:write name="datosForma" property="uea.grupo"/></h2>
<h2>Alumno: <bean:write name="datosForma"
property="alumno.nombreCompleto"/></h2>

<html:hidden name="datosForma" property="matricula"/>
<html:hidden name="datosForma" property="claveUea"/>

Calificaciones hasta el momento:<br><br>

Exámenes:<bean:write name="datosForma" property="NExámenes"/><br><br>

<logic:iterate name="datosForma" property="listaExámenes" id="examen">
<html:hidden name="examen" property="numero" indexed="true"/>
Examen <bean:write name="examen" property="numero"/>
<html:text name="examen" property="calificacion" size="3"
indexed="true"></html:text><br>
</logic:iterate>

<br><br>Programas:<bean:write name="datosForma"
property="NProgramas"/><br><br>

<logic:iterate name="datosForma" property="listaProgramas"
id="programa">
<html:hidden name="programa" property="numero" indexed="true"/>
Programa <bean:write name="programa" property="numero"/>
<html:text name="programa" property="calificacion" size="3"
indexed="true"></html:text><br>
</logic:iterate>

<br><br><html:submit>Aceptar</html:submit>
</html:form>

<html:link action="/administrarUea">Presiona para elegir otro
alumno</html:link><br>
<html:link action="/validarUsuario">Presiona para elegir otra
UEA</html:link>

</body>
</html>

```

Como se llaman acciones, no se requiere ningún mapeo adicional, sin embargo, es necesario declara como tipo sesión (session) el Form **OpcionSeleccionadaForm**.

```

<action path="/administrarUea" name="OpcionSeleccionadaForm"
scope="session" type="ts.struts.servlets.AdministrarUeaAction">
<forward name="seleccionarAlumno"
path="/paginas/seleccionarAlumno.jsp"></forward>
</action>

```

Finalmente se colocan tres ligas en la página de confirmación

registroCorrecto.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>Registro correcto</title>
</head>
<body>
<h1> Las calificaciones han sido registradas de manera correcta</h1>

<html:link action="/administrarUea">Presiona para elegir otro
alumno</html:link><br>
<html:link action="/validarUsuario">Presiona para elegir otra
UEA</html:link>
<html:link forward="login">Presiona para salir del sistema</html:link>

</body>
</html>
```

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