Universidad Zaragoza
1542

Uso de sesiones y cookies con servlets y JSP Ejercicios

Grado en Ing. en Informática e Ing. de Tecnologías y Servicios de Telecomunicación

Curso 2023-2024

Raquel Trillo Lado (raqueltl@unizar.es)
Carlos Tellería Orriols (telleria@unizar.es)
Dpto. Informática e Ingeniería de Sistemas

Universidad Zaragoza
1542

Introducción

Servlets y JSP

Universidad Zaragoza
1542

Guión

- **Introducción**
  - Sesiones en aplicaciones Web
  - Cookies en aplicaciones Web
- **Ejercicio 1**
- **Ejercicio 2**

Universidad Zaragoza
1542

Introducción

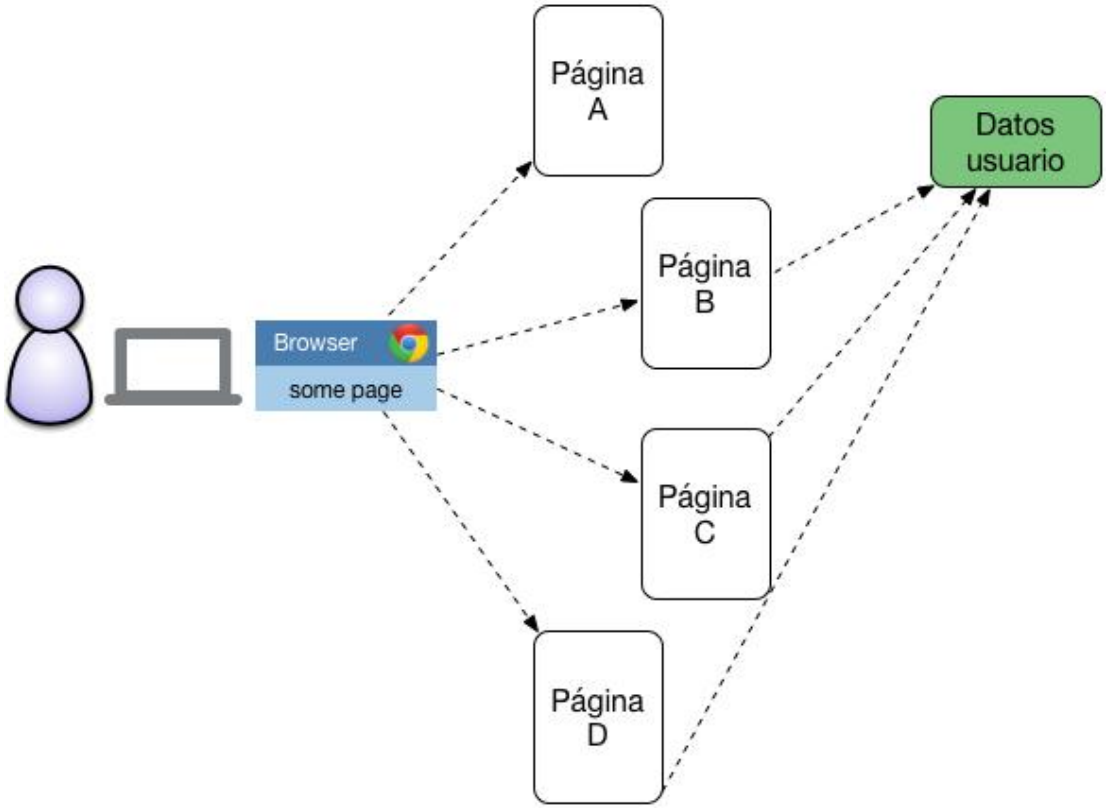
Sesiones en aplicaciones Web

- En un gran porcentaje de aplicaciones existe un proceso de **autenticación/validación de usuario**. Una vez el usuario ha accedido al sistema con su nombre de usuario y palabra clave, en función de sus permisos se le permiten unas operaciones u otras.
- **Concepto de sesión**: Espacio de **tiempo ocupado por la actividad del usuario desde que accede al sistema hasta que lo abandona**
- **Problema**: El protocolo **HTTP no tiene estado**, cada petición es independiente. Sin embargo en las aplicaciones web que conocemos:
  - El usuario no inserta su clave y nombre de usuario cada vez que realiza una operación sino que la realiza una sóla vez al inicio

Introducción

Sesiones en aplicaciones Web

- Problema: cómo compartir estado entre las distintas páginas de la aplicación web



Universidad Zaragoza
1542

Introducción

Sesiones en aplicaciones Web

- **La mayor parte de las APIs para desarrollo de aplicaciones Web proporcionan mecanismos para gestionar sesiones**

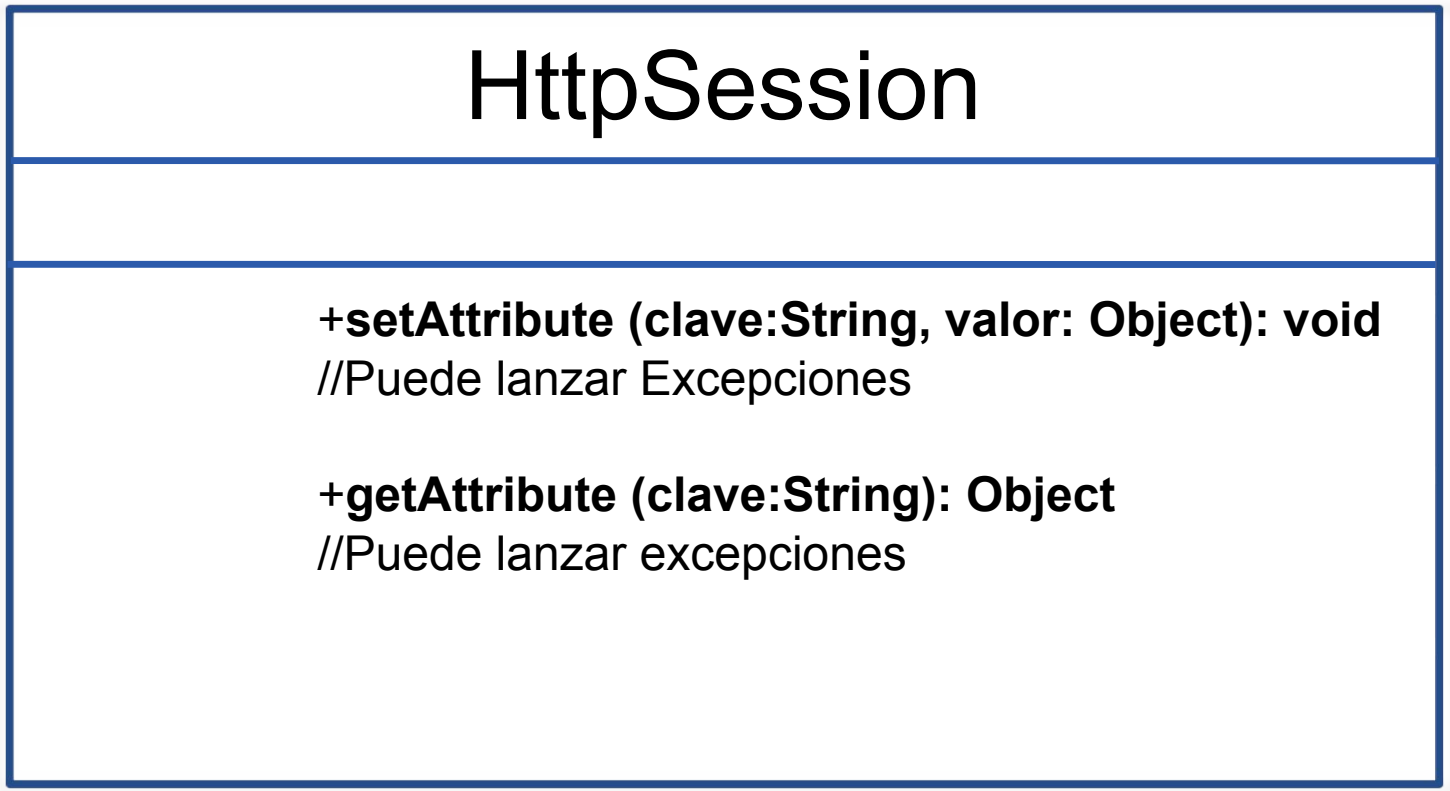
- API Servlets:
  - Crea una sesión en el servidor por cada navegador que accede a la aplicación Web, la cual se modela con el objeto **javax.servlet.http.HttpSession**
  - Cada petición del cliente (request) tiene asociada una sesión, para acceder a ella se realiza una llamada al método **getSession() de los objetos HttpServletRequest**
  - **request.getSession()** devuelve el objeto de la sesión actual, si este existe. Si no existe crea el objeto sesión antes

Introducción

Sesiones en aplicaciones Web

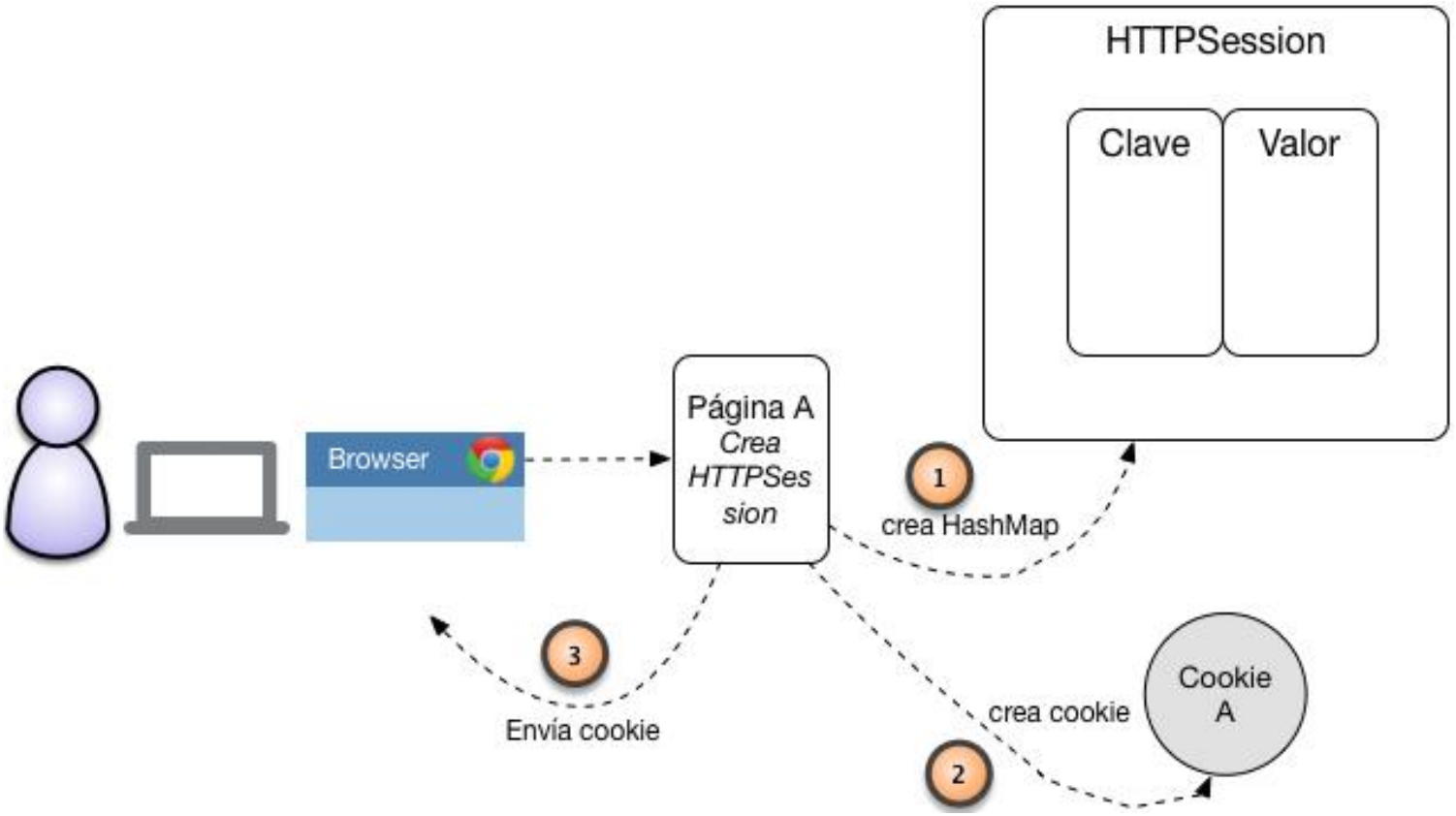
- HttpSession permite almacenar/recuperar objetos:

Es un HashMap



Introducción

- Sesiones en aplicaciones Web



Universidad Zaragoza
1542

Introducción

Cookies en aplicaciones Web

- **La mayor parte de las APIs para desarrollo de aplicaciones Web proporcionan mecanismos para gestionar cookies en el navegador del usuario:**
- **Cookie: Es un par (nombre de parámetro, valor del parámetro), donde tanto el nombre como el valor son cadenas de caracteres**
- **La cookie correspondiente *jsessionid* se gestiona automáticamente**
  - Gestión de cookies en Servlets:
    - Creación: **Cookie cookie = new Cookie ("login", userLogin);**
    - Envío al cliente: **response.addCookie(cookie);**
    - Recuperación en el servidor: **Cookie[] cookies = request.getCookies();**
- **Cada navegador debería soportar alrededor de 20 cookies por cada sitio Web y 300 en total. Además cada navegador suele limitar el tamaño de cada cookie a 4 KBytes**

Universidad Zaragoza
1542

Introducción

Cookies en aplicaciones Web

- **Cookies no persistentes:**
  - Válidas para la sesión, cuando se cierra el navegador se pierden.
- **Cookies persistentes:**
  - Válidas para un cierto periodo de tiempo
    - **public void setMaxAge(int expiry)**
- **Podemos borrarlas en cualquier momento**

Introducción

Cookies en aplicaciones Web

Ejemplo (persistente):

```
cookie.Value = value;
cookie.Expires = Convert.ToDateTime(“12/12/2008”);
Response.Cookies.Add(cookie);
```

Ejemplo borrado:

```
Cookie ck=new Cookie("user",""); //deleting value of cookie
ck.setMaxAge(0);//changing the maximum age to 0 seconds
response.addCookie(ck);//adding cookie in the response
```

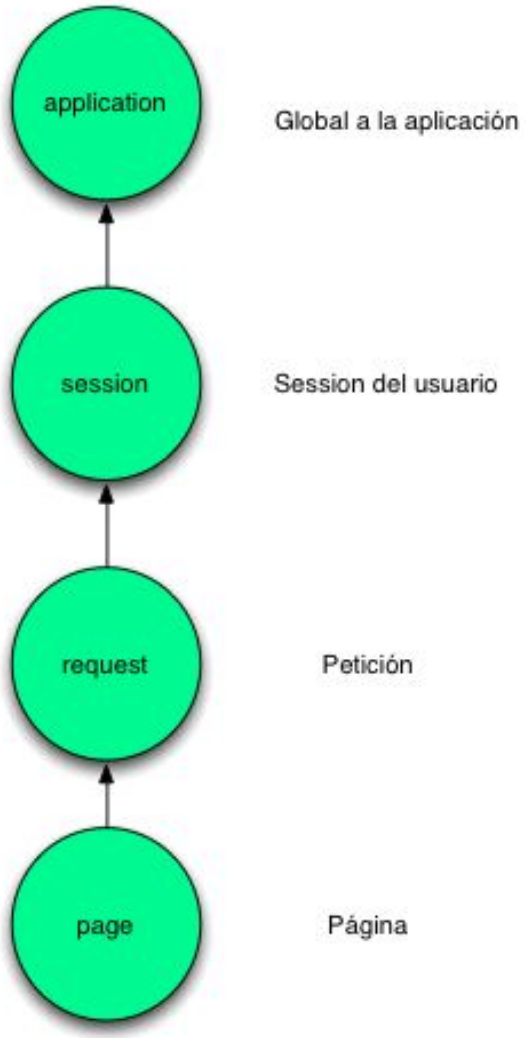
Introducción

Cookies en aplicaciones Web

- **Transcurrido un cierto tiempo (timeout) sin que el navegador acceda a la aplicación Web, el servidor de aplicaciones Web (Tomcat) destruye el correspondiente objeto sesión:**
  - Porque no existen mecanismos que le permitan al servidor saber si el navegador cliente ha dejado de usar la aplicación web
  - Por motivos de escalabilidad
  - También se puede desactivar una sesión:
    - sesion.invalidate();

Universidad Zaragoza

Contextos para gestión de información



Los contextos de Servlet/JSP permiten persistir e intercambiar información entre aplicaciones (**Application**), entre distintas llamadas de un cliente en una misma sesión (**Session**), entre Servlets (y JSPs) que forman parte de una cadena (forward) de ejecución (**Request**), o internamente entre distintas partes de código dentro de una página JSP (**Page**).

Todos los contextos son Maps, que asocian nombres (String) con Objetos

Universidad Zaragoza
1542

Ejercicio 1

Enunciado

- **Dado el siguiente formulario HTML (login.jsp), el fichero de configuración web web.xml y las clases que modelan la lógica de negocio de la aplicación, diseñar los servlets y páginas JSP necesarios para obtener el siguiente comportamiento:**
  - Si el usuario no inserta el nombre de usuario o la clave, el sistema le debe informar de ello y permitirle subsanar el problema.
  - Si el valor de la clave no coincide con el requerido se debe informar al usuario y permitirle que vuelva a teclearla.
  - Si el nombre de usuario no existe en el sistema se debe informar al usuario y permitirle que subsane el problema.
  - Si realiza el proceso de autenticación correctamente se debe informar de ello y saludarlo.

Universidad Zaragoza

Ejercicio 1

Login.jsp

Universidad Zaragoza

Ejercicio 1

Login.jsp

```
<html lang="es">
<head>
	<meta charset="UTF-8"/>
	<title>Registro de usuarios</title>
</head>
<body>
	<h1> Formulario de login de usuarios</h1>
	<form name="login"
		action=""
		method="post">

	  <!- - Campos capturar entrada cliente - - >
	  <!- - Fundamentalmente input  - - >
	    <label for="idNombre">Nombre de usuario*</label>: <input type="text" name="login" id="idNombre"/> <br/><br/>
	    <label for="idApellidos">Clave de acceso*</label>: <input type="text" name="apellido" id ="idApellidos"/><br/> <br/>


	    <p>Los campos marcados con un asterisco deben rellenarse de forma obligatoria</p>
	    <input type="submit" name="Acceder">  <input type="reset" name="Resetear valores"><br/>
	</form>

</body>
</html>
```

Universidad Zaragoza

Ejercicio 1

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns=http://java.sun.com/xml/ns/javaee  xmlns:xsi =
http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation =
"http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd”>
    <display-name>Mi primera aplicación Web</display-name>
    <servlet>   <servlet-name>LoginUsuario</servlet-name>
        <servlet-class>miprimeraaplicacion.LoginUsuarioServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginUsuario</servlet-name>
        <url-pattern>/LoginUsuario.do</url-pattern>
    </servlet-mapping>
    <welcome-file-list><welcome-file>login.jsp</welcome-file></welcome-file-list>
</web-app>
```

Universidad Zaragoza

Ejercicio 1

Clases del modelo de la aplicación

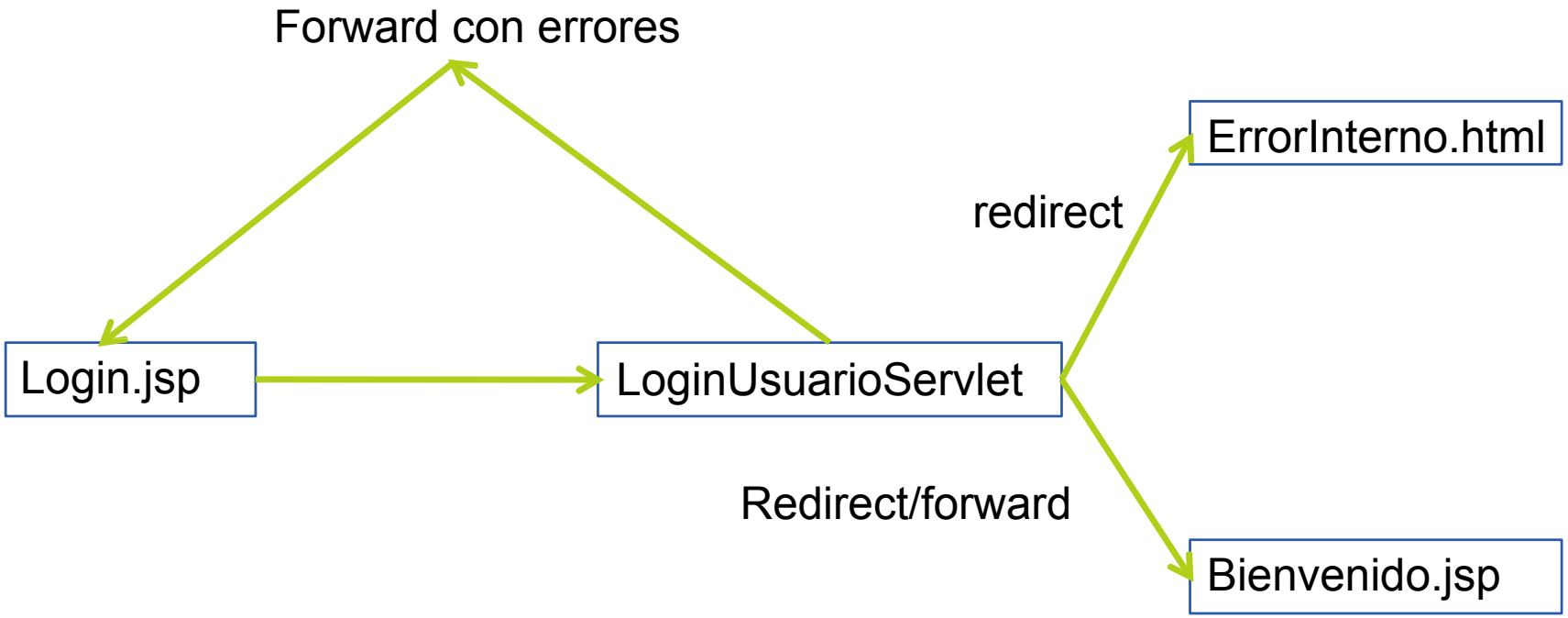
MiAplicacionFachada
+validarUsuario (login:String, clave: String, encriptada:boolean): ResultadoLoginVO //Puede lanzar Excepciones: InvalidPasswordException, InvalidUserException

ResultadoLoginVO
- login: String - claveEncriptada: String - nombre: String
+ getLogin (): String + getClaveEncriptada(): String + getNombre():String

Universidad Zaragoza

Ejercicio 1

Mapa de navegación



Universidad Zaragoza

Ejercicio 1: Servlet

```
package miprimeraaplicacion;
public class LoginUsuarioServlet extends HttpServlet{
    public void doPost (HttpServletRequest request, HttpServletRespond response) throws
    IOException, ServletException{
        Map<String, String> errors = new HashMap <String, String>();

        String login = request.getParameter("login");
        String clave = request.getParameter("apellido");
        if ((login == null) || (login.trim().equals(""))) errors.add("Login", "Campo obligatorio");
        if ((clave == null) || (clave.trim().equals(""))) errors.add("Clave", "Campo obligatorio");

        if (!errors.isEmpty()){ //Forward a Login.jsp con el mapa de errores /*Errores*/
       }else{ //Procesamiento del proceso de autenticación /*Lógica negocio*/}
    }
}
```

Universidad Zaragoza
1542

Ejercicio 1: Llamada a la lógica de negocio

```
MiAplicacionFacade f = new MiAplicacionFacade();
try{    ResultadoLoginVO r = f.validarUsuario(login, clave, encriptado);
        if (r == null) { response.sendRedirect("errorInterno.html")}
        else{ HttpSession s = request.getSession();  s.setAttribute ("nombre", r.getNombre());
        response.sendRedirect("bienvenida.jsp");}
} catch (InvalidPasswordException e) {
      errors.add("Clave", "Clave de acceso errónea");  //Forward a Login.jsp
} catch (InvalidUserException e) {
      errors.add("Login", "El usuario no se encuentra registrado"); //Forward a Login.jsp
}

if (!errors.isEmpty()){ //Forward a Login.jsp con el mapa de errores /*Errores*/}
```

Universidad Zaragoza
1542

Ejercicio 1: Redirección a login.jsp con errores

request.setAttribute("errores", errors);

RequestDispatcher dispatcher=request.getRequestDistpatcher("login.jsp");

dispatcher.forwardTo(request, response);

Apartado 1.A: Realizar la página bienvenida.jsp

Universidad Zaragoza
1542

Ejercicio 1.A: bienvenida.jsp

Opción más sencilla:

```
<p>Hola <%= session.getAttribute("Nombre") %></p>
```

Universidad Zaragoza
1542

Ejercicio 1.B:

Ejercicio 1.B: Modificar la página Login.jsp para considerar el tratamiento de errores

Universidad Zaragoza

Ejercicio 1.B: login.jsp con manejo de errores

```
<% @ page import = “java.util.Map” %>
<%
String loginError = “”; String claveError = “”;
String loginValor = “”; String claveValor = “”;
Map <String, String> e=(Map <String, String>) request.getAttribute(“errores”);
if (e != null){
        String cabecera = “<span style=\”color:red\”>”;  String final = “</span>”;
        if (e.containsKey(“login”)) loginError = cabecera + e.get(“login”)+ final;
        if (e.containsKey(“clave”)) claveError = cabecera + e.get(“clave”)+ final;
        loginValor = request.getParameter(“login”);
        claveValor = request.getParameter(“clave”);
}%>
```

Universidad Zaragoza

Ejercicio 1.B: login.jsp con manejo de errores

Añadir el atributo value y los mensajes de error a los campos:

<input type="text" name="login" value="<%= loginValor%>"

id="idLogin"/> <%= LoginError%>

<input type="password" name="apellido" value="<%= claveValor%>"

id="idApellidos"/> <%= claveError%>

Añadir la acción al formulario:

<form action="/LoginUsuario.do">

Universidad Zaragoza
1542

Ejercicio 2

Enunciado

- **Dado el siguiente formulario HTML (login.jsp), el fichero de configuración web web.xml y las clases que modelan la lógica de negocio de la aplicación, diseñar los servlets y páginas JSP necesarios para obtener el siguiente comportamiento:**
  - Si el usuario no inserta el nombre de usuario o la clave el sistema le debe informar de ello y permitirle subsanar el problema.
  - Si el valor de la clave no coincide con el requerido se debe informar al usuario y permitirle que vuelva a teclearla.
  - Si el nombre de usuario no existe en el sistema se debe informar al usuario y permitirle que subsane el problema.
  - Si realiza el proceso de autenticación correctamente se debe informar de ello y saludarlo.
  - Si el usuario tiene almacenada una cookie con la clave de acceso y otra con el nombre de usuario se le debe saludar directamente.

Universidad Zaragoza

Ejercicio 2

Login.jsp

Ejercicio 2

```
<html lang="es">
<head>
	<meta charset="UTF-8"/>
	<title>Registro de usuarios</title>
</head>
<body>
	<h1> Formulario de login de usuarios</h1>
	<form name="login"
		action=""
		method="post">

	  <!- - Campos capturar entrada cliente - - >
	  <!- - Fundamentalmente input  - - >
		<label for="idNombre">Nombre de usuario*</label>: <input type="text" name="login" id="idNombre"/> <br/><br/>
		<label for="idApellidos">Clave de acceso*</label>: <input type="text" name="apellido" id ="idApellidos"/><br/> <br/>
		<label for="recordar">Recordar datos de acceso</label>: <input type="checkbox" name="recordar"  id="recordar"/> <br/>

		<p>Los campos marcados con un asterisco deben rellenarse de forma obligatoria</p>
		<input type="submit" name="Acceder">  <input type="reset" name="Resetear valores"><br/>
	</form>

</body>
</html>
```

Universidad Zaragoza

Ejercicio 1

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns=http://java.sun.com/xml/ns/javaee  xmlns:xsi =
http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation =
"http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd”>
    <display-name>Mi primera aplicación Web</display-name>
    <servlet>   <servlet-name>LoginUsuario</servlet-name>
        <servlet-class>miprimeraaplicacion.LoginUsuarioServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginUsuario</servlet-name>
        <url-pattern>/LoginUsuario.do</url-pattern>
    </servlet-mapping>
    <welcome-file-list><welcome-file>login.jsp</welcome-file></welcome-file-list>
</web-app>
```

Universidad Zaragoza

Ejercicio 1

Clases del modelo de la aplicación

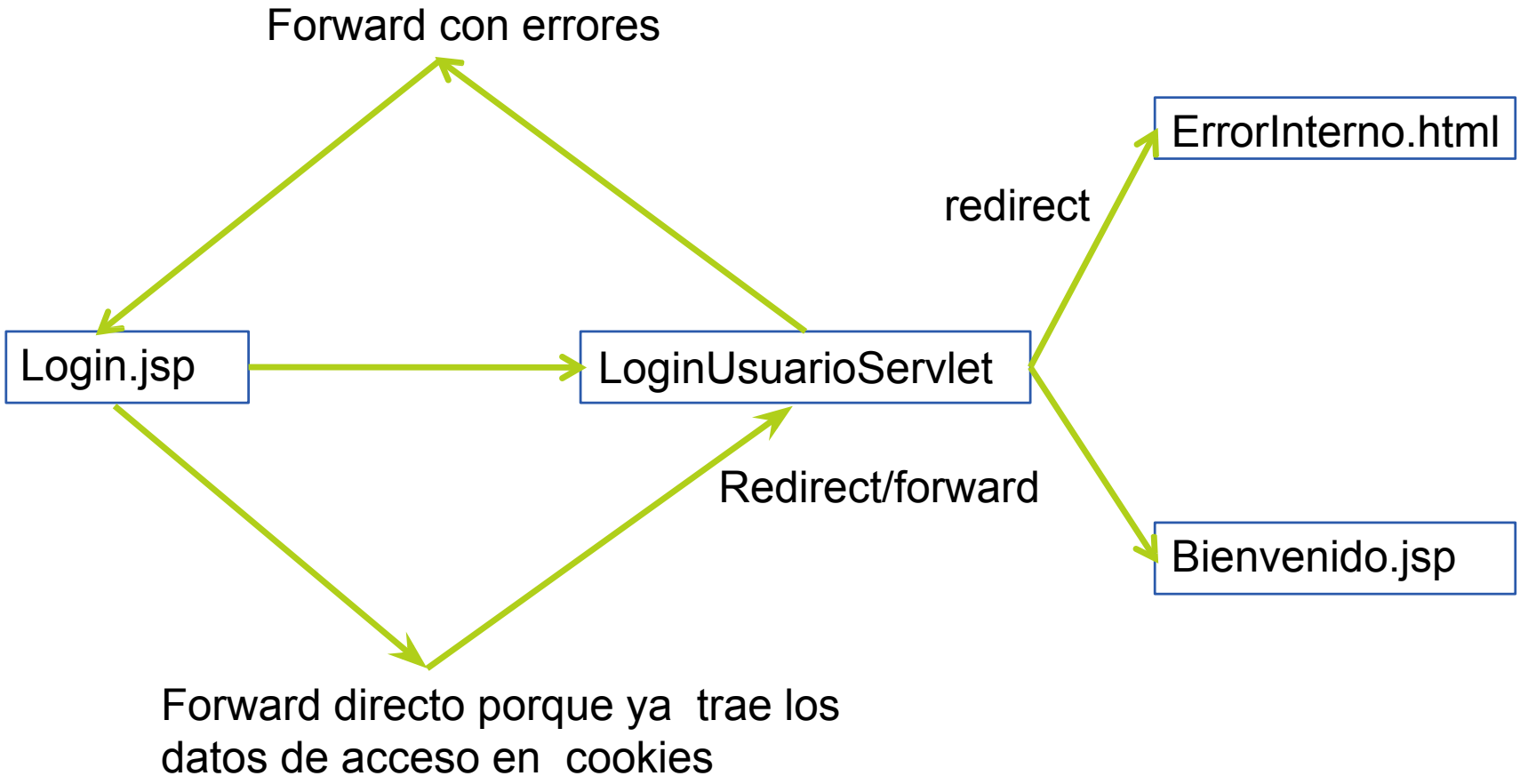
MiAplicacionFachada
+validarUsuario (login:String, clave: String, encriptada:boolean): ResultadoLoginVO //Puede lanzar Excepciones: InvalidPasswordException, InvalidUserException

ResultadoLoginVO
- login: String - claveEncriptada: String - nombre: String
+ getLogin (): String + getClaveEncriptada(): String + getNombre():String

Universidad Zaragoza

Ejercicio 1

Mapa de navegación



Universidad Zaragoza

Ejercicio 2: Servlet

```
package miprimeraaplicacion;
public class LoginUsuarioServlet extends HttpServlet{
    public void doPost (HttpServletRequest request, HttpServletRespond response) throws  IOException,
    ServletException{
        Map<String, String> errors = new HashMap <String, String>();
        String login = null; String clave = null; String recordar = "No"; boolean claveEncriptada = false;
        if (request.getAttribute ("usoCookies")) {claveEncriptada = true;}
        if (claveEncriptada){       login = request.getAttribute("loginEnCookie");
                        clave = request.getAttribute("claveEnCookie");}
        else{login = request.getParameter("login");  clave = request.getParameter("apellido");
              recordar = request.getParameter("recordar");
        //Tratamiento de errores parámetros
         if (!errors.isEmpty()){ //Forward a Login.jsp con el mapa de errores /*Errores*/
         }else{ //Procesamiento del proceso de autenticación /*Lógica negocio*/}
    }
}
```

Universidad Zaragoza

Ejercicio 2: Llamada a la lógica de negocio

```
MiAplicacionFacade f = new MiAplicacionFacade();
try{        ResultadoLoginVO r = f.validarUsuario(login, clave, encriptado);
    if (r == null) { response.sendRedirect("errorInterno.html")}
    else{   HttpSession s = request.getSession();
            s.setAttribute ("nombre", r.getNombre());
            if ((recordar != null) && (recordar.equals("Si")){
            Cookie cookieLogin = new Cookie ("loginUsuario", login);
            Cookie cookieClave = new Cookie ("claveUsuario", r.getClaveEncriptada());
response.addCookie(cookieLogin);
            response.addCookie(cookieClave);}
            response.sendRedirect("bienvenida.jsp");}
} catch (InvalidPasswordException e) {errors.add("Clave", "Clave de acceso errónea");  //Forward a Login.jsp}
} catch (InvalidUserException e) {errors.add("Login", "El usuario no se encuentra registrado"); //Forward a
Login.jsp}
if (!errors.isEmpty()){ //Forward a Login.jsp con el mapa de errores /*Errores*/}
```

Universidad Zaragoza
1542

Ejercicio 2.B:

Ejercicio 2.B: Modificar la página Login.jsp para considerar el tratamiento de cookies

Universidad Zaragoza

Ejercicio 2.B: login.jsp con manejo de errores

```
<% @ page import = “java.util.Map” %>
<%
String loginError = “”; String claveError = “”;
String loginValor = “”; String claveValor = “”;
Map <String, String> e=(Map <String, String>) request.getAttribute(“errores”);
if (e != null){
    String cabecera = “<span style=\”color:red\”>”;  String final = “</span>”;
    if (e.containsKey(“login”)) loginError = cabecera + e.get(“login”)+ final;
    if (e.containsKey(“clave”)) claveError = cabecera + e.get(“clave”)+ final;
    loginValor = request.getParameter(“login”);
    claveValor = request.getParameter(“apellido”);
}else{//Manejo de cookies}%>
```

Universidad Zaragoza

Ejercicio 2.B: login.jsp con manejo de cookies

```
<% //Manejo de cookies
Cookie [] cookies = request.getCookies();
for (i=0; i<cookies.size(); i++){
        String nombreCookieI = cookies[i].getName();
         if (nombreCookieI.equals("loginUsuario")){ loginValor = cookies[i].getValue();
                        request.setAttribute ("loginEnCookie", loginValor);
        }else if (nombreCookieI.equals("claveUsuario")){ claveValor = cookies[i].getValue();
                        request.setAttribute ("claveEnCookie", claveValor);}
}
if ((loginValor != null) && (claveValor != null) && (!(loginValor.trim().equals("")) &&
(!(claveValor.trim().equals(""))){
        request.setAttribute ("usoCookies", true);
        RequestDispatcher dispatcher = request.getRequestDistpatcher("LoginUsuario.do");
        dispatcher.forwardTo(request, response);
%>
```

Universidad Zaragoza

30224 y 30360-Sistemas de información

Grado en Ingeniería en Informática e Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2023-2024

Raquel Trillo Lado (raqueltl@unizar.es)

Ramón Hermoso Traba (rhermoso@unizar.es)

Carlos Tellería Orriols (telleria@unizar.es)

Dpto. Informática e Ingeniería de Sistemas