

Arquitectura y Diseño de Sistemas Web y C/S



Práctica 5: Java Servlets y Java Server Pages

D. Roberto Barchino y D. Javier Albert

Curso 2021/2022



Índice del Documento

Introducción	3
1. Práctica de Servlets	4
1.1. Hola Mundo	4
1.2. Acceso Básico a Formularios	5
1.3. Calculadora	6
1.4. Primitiva	8
1.5. Acceso a ficheros de texto plano	10
1.6. Acceso a datos	12
2. Práctica de Java Server Pages	14
2.1. Acceso a datos (I)	14
2.2. Acceso a datos (II)	15
2.3. Sesiones.....	16



Introducción

Este documento presenta la quinta práctica de la asignatura “Arquitectura y Diseño de Sistemas Web y C/S” optativa de 4º curso de las titulaciones de Grado en Ingeniería Informática, Grado en Ingeniería de Computadores y Grado en Sistemas de Información.

El objetivo de esta práctica es el manejo de la API de los Servlets y de la tecnología Java Server Pages del lenguaje de programación Java.

La forma y/o procedimiento de entrega del trabajo será el siguiente:

- Se realizarán en los grupos establecidos alumnos.
- Se confeccionará un documento Word donde se ofrecerán las explicaciones y las capturas de pantallas que acrediten la ejecución de cada uno de los Servlets y las páginas JSP's.
- La fecha tope de entrega será el próximo 29 de Noviembre y para la evaluación el 6 de Diciembre.
- Entrega: Mediante la plataforma Blackboard, a través de la utilidad de actividades los líderes de cada grupo podrán subir vuestro el trabajo realizado y evaluar posteriormente los trabajos de otros grupos asignados. Se entregará el trabajo como un único fichero zip que contendrá el pdf (transformando la memoria de formato Word a pdf) y los proyectos (código fuente) realizados.

Roberto Barchino y Javier Albert
Alcalá de Henares a 11 de Noviembre de 2021



1. Práctica de Servlets

Como se ha comentado anteriormente, el objetivo de esta práctica es utilizar el API de los Servlets, para ello se proponen distintos ejemplos que deberán ser ejecutados, además de explicar someramente su funcionamiento. Por tanto, se ofrecerá al alumno el código fuente del Servlet y la página en formato HTML que le dará acceso.

1.1. *Hola Mundo*

Página HTML

```
<html>
<head>
<title>Primer Servlet</title>
</head>
<body>
<center><h2>Ejemplo de Ejecución de un Servlet</h2></center><br><br>
<form action=/PrimerServlet/PrimerServlet method=GET>
<Center>
<BR><input type=submit value="Ejecutar el Servlet"></form>
</center>
</body>
</html>
```

Código del Servlet

```
//Primer Servlet.
//Muy sencilllo.
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class PrimerServlet extends HttpServlet
{
    public void doGet (HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter out = new PrintWriter(res.getOutputStream());
        out.println("<html>");
        out.println("<head><title>HolaMundoServlet</title></head>");
        out.println("<body>");
        out.println("<h1><center>Hola Mundo desde el servidor
        WEB</center></h1>");
        out.println("</body></html>");
        out.close();
    }
    public String getServletInfo()
    {
        return "Crea una página HTML que dice HolaMundo";
    }
}
```



1.2. Acceso Básico a Formularios

Página HTML

```
<html>
<head>
<title>Segundo Servlet</title>
</head>
<body>
<h2><center>Segundo Servlet</center></h2>
<form action=/SegundoServlet/SegundoServlet method=POST>
<BR>
<BR>
<center>
Introduzca su nombre y pulse el botón de enviar<BR>
<BR><input type=text name=NOMBRE>
<BR>
<BR><input type=submit value="Enviar Nombre">
<input type=reset value="Borrar">
</center>
</form>
</body>
</html>
```

Código del Servlet

```
//
// SegundoServlet.java
//
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class SegundoServlet extends HttpServlet
{
    String nombre;
    public void service( HttpServletRequest petition, HttpServletResponse
    respuesta )
    throws ServletException, IOException
    {
        nombre = petition.getParameter("NOMBRE");
        ServletOutputStream out = respuesta.getOutputStream();
        out.println("<html>");
        out.println("<head><title>HolaTalServlet</title></head>");
        out.println("<body>");
        out.println("<p><h1><center>Su nombre es: <B>"+nombre+"</B></center></h1></p>");
        out.println("</body></html>");
        out.close();
    }
}
```





Código del Servlet

```
//  
// CalculadoraServlet  
//  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class CalculadoraServlet extends HttpServlet  
{  
    public void service (HttpServletRequest req, HttpServletResponse  
    res)  
    throws ServletException, IOException  
    {  
        double op1, op2, result;  
        int operacion;  
        String simb_op[] = {"+", "-", "*", "/"};  
        ServletOutputStream out = res.getOutputStream();  
        op1 = Double.parseDouble(req.getParameter("operando1"));  
        op2 = Double.parseDouble(req.getParameter("operando2"));  
        operacion = Integer.parseInt(req.getParameter("operacion"));  
        result = calcula(op1, op2, operacion);  
        out.println("<html>");  
        out.println("<head><title>Resultado de calcular con  
Servlet</title></head>");  
        out.println("<body BGCOLOR = \"#E0E0FF\" TEXT= \"blue\">");  
        out.println("<h1><center>La operacion efectuada  
es:</center></h1>");  
        out.println("<h2> <b><center>"+ op1+" "+ simb_op[operacion-  
1] +  
" "+ op2 +  
" = "+ result +  
"</center></b></h2>");  
        out.println("</body>");  
        out.println("</html>");  
        out.close();  
    }  
    public double calcula(double op1, double op2, int operacion)  
    {  
        double result = 0;  
        switch (operacion)  
        {  
            case 1:  
                return op1 + op2;  
            case 2:  
                return op1 - op2;  
            case 3:  
                return op1 * op2;  
            case 4:  
                return op1 / op2;  
        }  
        return result;  
    }  
}
```



1.4. Primitiva

Página HTML

```
<html>
<head>
<title>Primitiva Servlet</title>
</head>
<body>
<h2><center>Primitiva Servlet</center></h2>
<form action=/PrimitivaServlet/PrimitivaServlet method=POST>
<BR>
<BR>
<center>
Introduce tu combinación y pulsa el botón de enviar<BR>
<BR>NUM1:<input type=text name=NUM1>
<BR>NUM2:<input type=text name=NUM2>
<BR>NUM3:<input type=text name=NUM3>
<BR>NUM4:<input type=text name=NUM4>
<BR>NUM5:<input type=text name=NUM5>
<BR>NUM6:<input type=text name=NUM6>
<BR>
<BR><input type=submit value=Enviar Combinación>
<input type=reset value=Borrar>
</center>
</form>
</body>
</html>
```

Código del Servlet

```
//
// PrimitivaServlet.java
//
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class PrimitivaServlet extends HttpServlet
{
int primi[] = new int[6], combiUsuario[] = new int[6];
int i, contador=0, aux, aciertos=0;
Random rand = new Random();
public void init(ServletConfig config) throws ServletException
{
super.init(config);
//generamos los números
while (contador<6)
{
aux = rand.nextInt(48) + 1;
if (!comprueba(primi,aux))
{
primi[contador] = aux;
contador++;
}
}
}
```

```
//ordenamos el array
```




```
Arrays.sort(primi);
}
private boolean comprueba(int array[], int num)
{
for (int i=0; i<=5; i++)
{
if (primi[i]==num) return true;
}
return false;
}
public void service( HttpServletRequest petition, HttpServletResponse
respuesta )
throws ServletException, IOException
{
aciertos=0;
ServletOutputStream out = respuesta.getOutputStream();
combiUsuario[0] =
Integer.parseInt(petition.getParameter("NUM1"));
combiUsuario[1] =
Integer.parseInt(petition.getParameter("NUM2"));
combiUsuario[2] =
Integer.parseInt(petition.getParameter("NUM3"));
combiUsuario[3] =
Integer.parseInt(petition.getParameter("NUM4"));
combiUsuario[4] =
Integer.parseInt(petition.getParameter("NUM5"));
combiUsuario[5] =
Integer.parseInt(petition.getParameter("NUM6"));
out.println("<h2><center>Primitiva Servlet</center></h2>");
//imprimimos todos los números de la combinación del usuario
out.print("<p>Tu combinación es:</p><B>");
for (i=0; i<6; i++)
{
out.print(" "+combiUsuario[i]);
}
out.print("</B>");
//comprobamos la combinación
for (i=0; i<=5; i++)
{
if (Arrays.binarySearch(primi,combiUsuario[i])>=0)
{
out.println("<p>Número acertado:
<B>"+combiUsuario[i]+"</B></p>");
aciertos++;
}
}
out.println("<p>Números acertados: <B>"+aciertos+"</B></p>");
//imprimimos todos los números de la combinación ganadora
out.print("<p>La combinación ganadora es:</p><B>");
for (i=0; i<6; i++)
{
out.print(" "+primi[i]);
}
out.print("</B>");
out.close();
}
}
```



1.5. Acceso a ficheros de texto plano

Página HTML

```
<html>
<head>
<title>Segundo Servlet</title>
</head>
<body>
<h2><center>Segundo Servlet</center></h2>
<form action=/FicheroServlet/FicheroServlet method=POST>
<BR>
<BR>
<center>
Introduzca su nombre y pulse el botón de enviar<BR>
<BR><input type=text name=TEXTO>
<BR>
<BR><input type=submit value=Enviar Nombre>
<input type=reset value=Borrar>
</center>
</form>
</body>
</html>
```

Código del Servlet

```
//
// FicheroServlet.java
//
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class FicheroServlet extends HttpServlet
{
    StringBuffer mensaje = null;
    FileOutputStream fos = null;
    String[] strTEXT0;
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        FileInputStream fis = null;
        try
        {
            fis = new FileInputStream("c:\\tmp\\texto.txt");
        }
        catch(java.io.FileNotFoundException e)
        {
            e.printStackTrace();
        }
        mensaje = new StringBuffer();
        try
        {
            {
                int caracter;
                while ( (caracter = fis.read()) != -1 )
                {
                    mensaje.append((char) caracter);
                }
            }
        }
        fis.close();
    }
}
```



```
}  
catch(java.io.IOException e)  
{  
e.printStackTrace();  
}  
try  
{  
fos = new FileOutputStream("c:\\tmp\\log.txt");  
}  
catch(java.io.IOException e)  
{  
e.printStackTrace();  
}  
}  
public void service( HttpServletRequest petition,  
HttpServletResponse respuesta )  
throws ServletException, IOException  
{  
strTEXTO = petition.getParameterValues("TEXTO");  
ServletOutputStream out = respuesta.getOutputStream();  
out.println("<p>" + mensaje + "</p>");  
out.println("<p>Su nombre es: " + strTEXTO[0] + "</p>");  
out.close();  
registrar();  
}  
public void destroy()  
{  
try  
{  
fos.close();  
}  
catch(java.io.IOException e)  
{  
e.printStackTrace();  
}  
}  
public synchronized void registrar()  
{  
try  
{  
fos.write(strTEXTO[0].getBytes());  
}  
catch(java.io.IOException e)  
{  
e.printStackTrace();  
}  
}  
}
```



1.6. Acceso a datos

Página HTML

```
<html>
<head>
<title>Ejemplo Encuesta</title>
</head>
<body bgcolor=white>
<center><h2>Por favor rellene todos los datos</h2>
<form action=/EncuestaServlet/EncuestaServlet method=POST>
<BR>Nombre:<BR>
<BR>
<input type=text name=NOMBRE>
<BR>
<BR>E-Mail:<BR>
<BR>
<input type=text name=EMAIL>
<BR>
<BR><B>Pregunta:</B>¿Piensas utilizar a los Servlets para los
proyectos a partir de ahora?<BR>
<BR>
Si<input type=radio name=RESPUESTA value=SI>
No<input type=radio name=RESPUESTA value=NO>
<BR><BR><input type=submit value=Enviar>
<input type=reset value=Borrar>
</form></center>
</body>
</html>
```

Código del Servlet

```
//
// Encuesta.java
//
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class EncuestaServlet extends HttpServlet
{
    Statement mandato = null;
    Connection conexion = null;

    public void init(ServletConfig config) throws ServletException
    {
        try
        {
            conexion = DriverManager.getConnection
            ("jdbc:derby://localhost:1527/sample","app","app");
            mandato = conexion.createStatement();
        }
        catch (Exception e)
        {
            System.out.println("Problemas al conectar con la base de datos");
        }
    }
}
```



```
public void service( HttpServletRequest petition, HttpServletResponse
respuesta )
throws ServletException, IOException
{
    /* creación del flujo de salida hacia el cliente */
    ServletOutputStream out = respuesta.getOutputStream();
    respuesta.setContentType("text/html");
    /* recuperamos los valores que nos manda el cliente */
    String strNombre = petition.getParameter("NOMBRE");
    String strEmail = petition.getParameter("EMAIL");
    String strRespuesta = petition.getParameter("RESPUESTA");
    /* insertamos los datos en la base de datos */
    try
    {
        mandato.executeUpdate("INSERT INTO ENCUESTA VALUES( '" + strNombre
+ "', '" + strEmail + "', '" + strRespuesta + "')");
    }
    catch(SQLException e)
    {
        System.out.println(e);
        return;
    }
    /* leemos todos los registros para crear la estadística */
    try
    {
        int intSI = 0;
        int intNO = 0;
        ResultSet resultado = mandato.executeQuery("SELECT RESPUESTA FROM
ENCUESTA");
        while(resultado.next())
        {
            String resp = resultado.getString("RESPUESTA");
            if(resp.compareTo("SI")==0) intSI++;
            else intNO++;
        }
        out.println("<h2><center>Encuesta Servlet</center></h2>");
        out.println("<BR>Gracias por participar en esta encuesta.");
        out.println("<BR>Los resultados hasta este momento son :");
        out.println("<BR> SI : "+intSI);
        out.println("<BR> NO : "+intNO);
    }
    catch(Exception e)
    {
        System.out.println(e);
        return;
    }
}

public void destroy()
{
    try
    {
        conexion.close();
    }
    catch(SQLException e)
    {
        System.out.println(e);
        return;
    }
}
```



2. Práctica de Java Server Pages

El objetivo de esta práctica es utilizar la tecnología de las páginas JSP, para ello se proponen distintos ejemplos y/o ejercicios que deberán ser ejecutados e implementados, además de explicar someramente su funcionamiento. Por tanto, se ofrecerá al alumno el código fuente en los ejemplos y en los ejercicios se deberá desarrollar las páginas JSP correspondientes.

2.1. Acceso a datos (I)

Se ofrece una página JSP para comprender el funcionamiento básico del acceso a una base de datos.

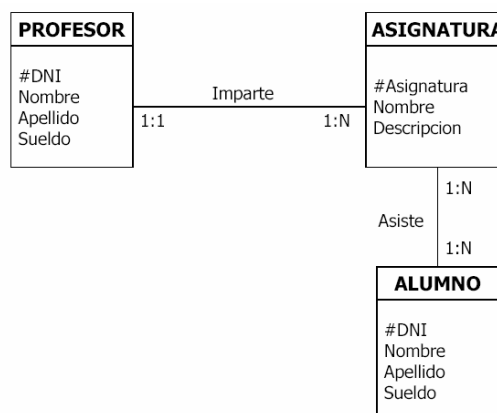
```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- jsp0501.jsp
Esta es la página que permite visualizar la información de los registros que
contiene la base de datos "Libros", y que presenta al lector una forma sencilla de
conexión con bases de datos. En este caso se utiliza el puente Jdbc-Odbc para
acceder a la tabla almacenada en una base de datos Microsoft Access.
CREATE TABLE LIBROS (
TITULO VARCHAR(50) NOT NULL,
AUTOR VARCHAR(30) NOT NULL,
PRECIO INTEGER NOT NULL)
-->
<html>
<head>
<title>Tutorial JSP, Base de Datos</title>
</head>
<body>
<%@ page import="java.sql.*" %>
<%!
// Declaraciones de las variables utilizadas para la
// conexión a la base de datos y para la recuperación de
// datos de las tablas
Connection c;
Statement s;
ResultSet rs;
ResultSetMetaData rsmd;
%>
<%
// Inicialización de las variables necesarias para la
// conexión a la base de datos y realización de consultas
c = DriverManager.getConnection("jdbc:derby://localhost:1527/sample","app","app");
s = c.createStatement();
rs = s.executeQuery("SELECT * FROM LIBROS");
rsmd = rs.getMetaData();
%>
<table width="100%" border="1">
<tr>
```



```
<% for( int i=1; i <= rsmd.getColumnCount(); i++ ) { %>
<%-- Obtenemos los nombres de las columnas y los colocamos
como cabecera de la tabla --%>
<th><%= rsmd.getColumnLabel( i ) %></th>
<% } %>
</tr>
<% while( rs.next() ) { %>
<tr>
<% for( int i=1; i <= rsmd.getColumnCount(); i++ ) { %>
<%-- Recuperamos los valores de las columnas que
corresponden a cada uno de los registros de la
tabla. Hay que recoger correctamente el tipo de
dato que contiene la columna --%>
<% if( i == 3 ) { %>
<td><%= rs.getInt( i ) %></td>
<% } else { %>
<td><%= rs.getString( i ) %></td>
<% } } %>
</tr>
<% } %>
</table>
</body>
</html>
```

2.2. Acceso a datos (II)

Con el siguiente modelo desarrollar las tablas correspondientes en Derby e incorporar una serie mínima de datos. Después crear una página JSP que nos informe de los alumnos que reciben clases del profesor **X**, siendo **X** el nombre de un profesor que se solicitará a través de un formulario.





2.3. Sesiones.

Ejemplo práctico de la gestión de sesiones a través del uso el objeto implícito **session**.

Página HTML: eje3sesion.html

```
<HTML><head>
<title> Ejemplo de Sesión </title></head>
<body>
<h1> Ejemplo de sesión </h1>
<form method="post" action="sesionEje.jsp">
Por favor, introduce tu nombre:
<input type="text" name="nombre">
<input type="submit" value="enviar información"> </form> </body>
</HTML>
```

Página JSP: sesionEje.jsp

```
<HTML>
<head> <title> Ejemplo de Sesión </title> </head>
<body>
<%
String val = request.getParameter("nombre");
if (val != null) session.setAttribute("Nombre",val);
%>
<center> <h1>Ejemplo de Sesión</h1>
Donde quieres ir!!!
<a href="sesionEje1.jsp">Ir a Página 1</a>
<a href="sesionEje2.jsp">Ir a Página 2</a>
</body>
</HTML>
```

Página JSP: sesionEje1.jsp

```
<HTML><head> <title> Ejemplo de Sesión </title> </head> <body>
<center> <h1>Ejemplo de Sesión</h1>
Hola, <%=session.getAttribute("Nombre")%>
Bienvenido a la página 1
</body> </HTML>
```

Página JSP: sesionEje2.jsp

```
<HTML><head> <title> Ejemplo de Sesión </title> </head> <body>
<center> <h1>Ejemplo de Sesión</h1>
Hola, <%=session.getAttribute("Nombre")%>
Bienvenido a la página 2
</body> </HTML>
```