

**Hoja04_Tomcat_02**

1.- Vamos a crear desde Eclipse un proyecto dinámico (Dynamic Web Project) llamado Tomcat_Saludo. Marca crear un descriptor de despliegue cuando lo crees.

Vamos a crear los siguientes archivos:

- index.html (directamente en la carpeta WebContent)
 - html con un título con tu nombre y dos enlaces (buenas.jsp y hola)
 -

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Tomcat_Saludo</title>
6 </head>
7 <body>
8 <h1>Iván Lorenzo</h1>
9 <ul>
10 <li><a href="buenas.jsp">Buenas (JSP)</a></li>
11 <li><a href="hola">Buenas (Servlet)</a></li>
12 </ul>
13 </body>
14 </html>
```

- buenas.jsp (directamente en la carpeta WebContent)

```
index.html | buenas.jsp | Buenas (JSP)
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@
4   page import="java.util.Calendar"%>
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="UTF-8">
10 <title>Buenas (JSP)</title>
11 </head>
12 <body>
13 <div class="content">
14 <p>
15   Hoy es
16   <%=Calendar.getInstance().getTime()%>
17 </p>
18 <%
19   String saludo;
20   int horaDelDia = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);
21   if (horaDelDia < 12) {
22     saludo = "¡Buenos días!";
23   } else if (horaDelDia >= 12 && horaDelDia < 21) {
24     saludo = "¡Buenas tardes!";
25   } else {
26     saludo = "¡Buenas noches!";
27   }
28 >%
29 <p><%=saludo%></p>
30 </div>
31 </body>
32 </html>
```

**Hoja04_Tomcat_02**

- HolaServlet.java
 - Crear un paquete llamado **com.daw2XX.tomcat_saludo** en la carpeta src de Java Resources siendo XX tu usuario del dominio
 - Y dentro del paquete crear un servlet llamado HolaServlet.java

```
/**
 * Servlet implementation class HolaServlet
 */
@WebServlet("/HolaServlet")
public class HolaServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HolaServlet() {
        super();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        try {
            out.println("<html>");
            out.println("<head><title>Hola (Servlet)</title></head>");
            out.println("<body>");
            out.println("<h1>¡Hola!</h1>");
            out.println("<p>URI: " + request.getRequestURI() + "</p>");
            out.println("<p>Protocolo: " + request.getProtocol() + "</p>");
            out.println("<p>Ip del cliente: " + request.getRemoteAddr() + "</p>");
            out.println("</body></html>");
        } finally {
            out.close();
        }
    }
}
```

2.- Ejecuta ahora la aplicación desde cada uno de los ficheros creados. Se probará en el Tomcat local. Deberían funcionar todos. Haz captura si es así y sino pregunta al profesor para solucionarlo

3.- Ahora desde index.html pulsa en los enlaces. ¿Funcionan? ¿Qué pasa con el enlace hola? Intenta modificar lo necesario para arreglarlo. NOTA: tiene que ver con el descriptor de despliegue (fichero web.xml)

4.- Para ahora el servidor Tomcat desde Eclipse.

5.- Exporta la aplicación a un fichero .war (Export → WAR File) desde el proyecto. Observa el contenido. ¿Se copian los .java?

6.- Si accedemos a http://127.0.0.1:8080/Tomcat_Saludo desde el navegador veremos que no está desplegado en Tomcat (ya que durante el desarrollo se usa un Tomcat local)

7.- Despliega ahora la aplicación copiando el archivo .war en la carpeta webapps y prueba a acceder a la URL anterior