

Tipo : Guía de laboratorio
Capítulo : Java Server Pages (JSP)
Duración : 50 minutos

I. OBJETIVO

Elaborar un JSP usando objetos implícitos.

II. REQUISITOS

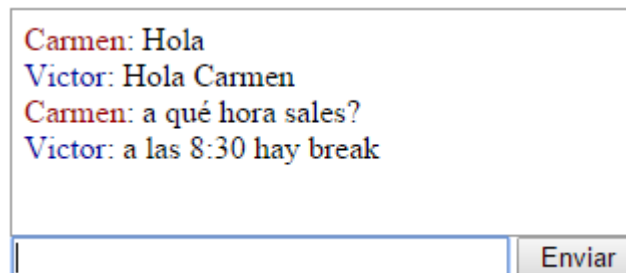
Los siguientes elementos de software son necesarios para la realización del laboratorio:

- NetBeans
- Navegador de Internet

III. DESCRIPCIÓN

Ejercicio:

Crear un proyecto web que mediante objetos implícitos de JSPs los usuarios puedan comunicarse en una sala de chat, como muestra la figura:



1. Crear la página **index.jsp** con la siguiente implementación:

```
<html>
<head>
  <meta charset="UTF-8">
  <title>JSP Page</title>
</head>
<body>
  <form action="Ingreso" method="post">
    <table style="margin: auto">
      <tr>
        <td>Nick</td>
        <td>
```

```
<input type="text" name="nick"/>
</td>
</tr>
<tr>
<td>Color</td>
<td>
<select name="color">
<option value="#900">Rojo</option>
<option value="#090">Verde</option>
<option value="#009">Azul</option>
</select>
</td>
</tr>
<tr>
<td colspan="2">&nbsp;   </td>
</tr>
<tr>
<td colspan="2" style="text-align: center">
<input type="submit" value="Enviar Datos"/>
</td>
</tr>
</table>
</form>
</body>
</html>
```

2. Crear **chat.jsp** donde se muestre el chat según lo que se vaya ingresando por los usuarios que se conectan:

```
<%
String sala = (String) application.getAttribute("sala");
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="refresh" content="20"/>
<title>JSP Page</title>
<style type="text/css">
    #sala {
        margin: auto;
        border: 1px solid #999;
        padding: 6px;
        overflow: auto;
        width: 300px;
        height: 100px;
    }
</style>

<script type="text/javascript">
    function load() {
        document.getElementById("msg").focus();
    }
</script>
</html>
```

```

</script>
</head>
<body onload="load()">
  <div id="sala"><%=sala%></div>

  <form action="Chat" method="post" style="text-align: center">
    <input type="text" name="msg" style="width: 244px"
      id="msg"/>
    <input type="submit" value="Enviar" style="width: 60px"/>
  </form>
</body>
</html>

```

3. Crear el Servlet que permitirá el ingreso de los usuarios a la sala de chat. El nombre del Servlet será **ServletIngreso**:

```

package web.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet(name = "ServletIngreso", urlPatterns = {"/Ingreso"})
public class ServletIngreso extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        String nick = request.getParameter("nick");
        String color = request.getParameter("color");
        String target = "index.jsp";

        if ((nick != null) && nick.trim().length() > 0) {
            HttpSession session = request.getSession(true);
            session.setAttribute("nick", nick);
            session.setAttribute("color", color);

            target = "Chat";
        }

        response.sendRedirect(target);
    }

    // <editor-fold defaultstate="collapsed" desc="doGet y doPost">
    /**
     * Handles the HTTP
     * <code>GET</code> method.

```

```

*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

4. Por último, crearemos el Servlet que permita manejar los mensajes que se muestran/crean en la sala de chat. **ServletChat:**

```
package web.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.annotation.WebInitParam;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet(name = "ServletChat", urlPatterns = {"/Chat"},
initParams = {
    @WebInitParam(name = "sala", value = "")})
public class ServletChat extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");

        String msg = request.getParameter("msg");
        String sala = (String) getServletContext().getAttribute("sala");
        if (sala == null) {
            sala = "";
        }

        if ((msg != null) && (msg.trim().length() > 0)) {
            HttpSession session = request.getSession();
            String nick = (String) session.getAttribute("nick");
            String color = (String) session.getAttribute("color");

            sala += "<span style=\"color:\" + color + \">\"
                + nick + "</span>\" + \": \"
                + msg + "<br/>";
        }

        getServletContext().setAttribute("sala", sala);
        response.sendRedirect("chat.jsp");
    }

    // <editor-fold defaultstate="collapsed" desc="doGet y doPost">
    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     */
}
```

```

    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

IV. EVALUACIÓN

1. Mencionar y explicar 3 objetos implícitos.
 - a. El objeto **application** para acceder a variables de ámbito aplicación.
 - b. El objeto **session** para acceder a variables de ámbito de la sesión.
 - c. El objeto **param** para recoger parámetros enviados por otros documentos.