

```
1  package opo.informatica.auxiliar;
2
3  /**
4   * Punto
5   */
6  public class Punto {
7
8      private int x;
9      private int y;
10
11     public Punto(){}
12
13     public Punto(int x, int y){
14         this.x = x;
15         this.y = y;
16     }
17
18     public int getX(){
19         return this.x;
20     }
21
22     public void setX(int x){
23         this.x = x;
24     }
25
26     public void setY(int y){
27         this.y = y;
28     }
29
30     public int getY(){
31         return this.y;
32     }
33 }
```

```

1  package opo.informatica.sevlets;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import java.util.Date;
6  import java.util.Enumeration;
7  import java.util.LinkedHashMap;
8  import java.util.Map;
9  import java.util.logging.Logger;
10
11 import javax.servlet.ServletException;
12 import javax.servlet.annotation.WebServlet;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16 import javax.servlet.http.HttpSession;
17
18 import opo.informatica.auxiliar.Punto;
19
20 @WebServlet("/CreaSesion")
21 public class CreaSesion extends HttpServlet{
22
23     private static final Logger LOGGER = Logger.getLogger(CreaSesion.class.getName());
24
25     @Override
26     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
27         response.setContentType("text/html;charset=UTF-8");
28         PrintWriter out = response.getWriter();
29         try {
30             // 1.1) Se vacía la sesión si existe
31             // Llamamos al método para obtener la sesión de la petición, con el flag de
crearla (si no existe) a false
32             HttpSession session = request.getSession(false);
33             // comprobamos si existía la sesión (es decir si la petición ya tenía una)
34             if(session != null) {
35                 LOGGER.info("Antes de invalidar la sesión que existía");
36                 LOGGER.info("Comprobamos el ID que tenía esta > ID: " + session.getId());
37                 // Invalidamos la sesión que desvincula cualquier objeto asociada a ella
38                 session.invalidate();
39             }
40
41             // 1.2) Se crea una sesión asociada a la petición
42             session = request.getSession();
43             LOGGER.info("Comprobamos si creó una nueva sesión");
44             LOGGER.info("Mostrando el ID: " + session.getId());
45
46             // 1.3) Se imprime el contenido de la sesión vacía
47             out.println(this.printSessionContent(session));
48
49             // 1.4) Se registran variables de sesión (objetos de diferentes clases)
50             // Se registran 6 variables de sesión con los siguientes identificadores y tipos:
51             // - entero: un número entero
52             session.setAttribute("entero", new Integer(1));
53             // - real: un número real
54             session.setAttribute("real", new Float(1.0));
55             // - texto: una cadena de texto
56             session.setAttribute("texto", "cadena de texto");
57             // - fecha: una fecha (objeto DateTime)
58             session.setAttribute("fecha", new Date());
59             // - semaforo (array asociativo)
60             Map<String,String> arrayAsociativo = new LinkedHashMap<>();
61             arrayAsociativo.put("rojo", "parar");
62             arrayAsociativo.put("ambar", "cambia");
63             arrayAsociativo.put("verde", "pasar");
64             session.setAttribute("semaforo", arrayAsociativo);
65             // - unPunto (objeto Punto)
66             session.setAttribute("unPunto", new Punto(0, 0));
67

```

```

68         // 1.5) Se crea y visualiza el enlace al script RecuperaSesión
69         out.println("<a href=\"RecuperaSesión\">Recupera Sesión</a>");
70
71     } catch (Exception e) {
72         out.println("Se produce una excepción <br/>");
73         out.println(e.getMessage());
74     }
75 }
76
77 @Override
78 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
79 }
80
81 /**
82  * Mostrará el contenido de todas los atributos de la sesión en formato HTML TABLE
83  */
84 private String printSessionContent(HttpSession session) {
85     String salida = null;
86
87     Enumeration<String> attributeNames = session.getAttributeNames();
88     while(attributeNames.hasMoreElements()) {
89
90         String name = attributeNames.nextElement();
91         salida += name + " - " + session.getAttribute(name) + ",";
92     }
93
94     return this.format2HTMLTable(salida);
95 }
96
97 private String format2HTMLTable(String atributos){
98     String resultado = null;
99
100     if (atributos != null) {
101         resultado = "<h2>Session Info</h2>" +
102             "<table border=\"1\" >" +
103             "    <tr>" +
104             "        <th>Nombre</th><th>Valor</th>" +
105             "    </tr>";
106
107         String[] attr = atributos.split(",");
108         for (String elem : attr) {
109             resultado += "<tr><td>" + elem.split(" - ")[0] + "</td><td>" + elem.split(" - ")
110                 [1] + "</td></tr>";
111         }
112         resultado += "</table>";
113     }
114     return resultado;
115 }
116 }

```

```

1  package opo.informatica.sevlets;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import java.util.Enumeration;
6
7  import javax.servlet.ServletException;
8  import javax.servlet.annotation.WebServlet;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import javax.servlet.http.HttpSession;
13
14 @WebServlet(name = "RecuperaSesion", urlPatterns = {"/RecuperaSesion"})
15 public class RecuperaSesion extends HttpServlet{
16
17     @Override
18     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
19     ServletException, IOException {
20         response.setContentType("text/html;charset=UTF-8");
21         PrintWriter out = response.getWriter();
22         try {
23             // 2.1) Se recuperan las variables de sesión previamente creadas
24             HttpSession session = request.getSession(false);
25
26             if (session != null) {
27                 out.println("<h2>Session Info</h2>");
28                 out.println("<ul>");
29                 Enumeration<String> attributeNames = session.getAttributeNames();
30                 while(attributeNames.hasMoreElements()) {
31
32                     String name = attributeNames.nextElement();
33                     // 2.2) Se imprime el contenido de la sesión.
34                     out.println( "<li>" + name + " - " + session.getAttribute(name) + "</li>"
35 );
36                 }
37                 out.println("</ul>");
38
39                 out.println("<a href=\"CreaSesion\">Crea Sesión</a>");
40
41             } catch (Exception e) {
42                 out.println("Se produce una excepción <br/>");
43                 out.println(e.getMessage());
44             }
45
46         }
47
48         @Override
49         protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
50         ServletException, IOException {}
51     }
52 }

```