

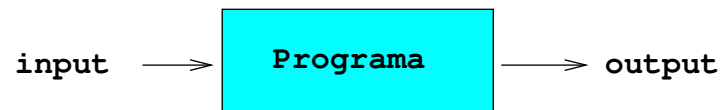
Java Servlets

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

10 de mayo de 2007

¿Qué es un programa?



En pascal:

```
program holaMundo (input,output);  
var  
  nombre : String;  
begin  
  readLn(nombre);  
  writeLn('Hola ',nombre);  
end.  
1  
2  
3  
4  
5  
6  
7
```

Un servlet

- **input** → `HttpServletRequest`
- **output** → `HttpServletResponse`

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class HolaMundo extends HttpServlet {
    public final void doGet(final HttpServletRequest request,
                           final HttpServletResponse response)
        throws ServletException, IOException {
        ServletOutputStream out = response.getOutputStream();
        String nombre = request.getParameter("nombre");
        response.setContentType("text/html");
        if (nombre==null) {
            out.println(" <p>Hola Mundo</p>");
        } else {
            out.println(" <p>Hola "+nombre+" </p>");
        }
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

Servidor de aplicaciones: Tomcat

Los servlet son controlados por servidor de aplicaciones.

Tomcat: <http://jakarta.apache.org/tomcat/index.html>

Cuando una petición llega

- 1 Si el objeto de la clase no ha sido creado
 - 1 Carga la clase
 - 2 Crea el objeto
 - 3 Invoca el método `init`.
- 2 Llama al método `service`.

Si debe eliminar al servlet llama al método `destroy`.

Interfaz `javax.servlet.Servlet`

- `void destroy()`
- `void init(ServletConfig config)`
- `void service(ServletRequest req, ServletResponse res)`

Clase `javax.servlet.http.HttpServlet`

- `void doGet(HttpServletRequest req, HttpServletResponse resp)`
- `void doPost(HttpServletRequest req, HttpServletResponse resp)`

Servlet contador I

```
public class Contador extends HttpServlet {
    private int contador;

    /*
     * Metodos para guardar el contador antes de
     * finalizar y leerlo al empezar.....
     */

    private synchronized int incrementa() {
        contador++;
        return contador;
    }

    public final void doGet(final HttpServletRequest request,
        final HttpServletResponse response)
        throws ServletException, IOException {
        int n=incrementa();
        ServletOutputStream out = response.getOutputStream();
        response.setContentType("text/html");
        out.println(" <p>Numero de accesos:"+n+" </p>");
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Servlet contador II

```
private ServletConfig config;
private final static String strFichero="fichero";

public final void init(final ServletConfig c)
    throws ServletException {
    config = c;
    ServletContext ctx = config.getServletContext();
    String fichero = config.getInitParameter(strFichero);
    try {
        BufferedReader f
            = new BufferedReader(
                new FileReader(ctx.getRealPath(fichero)));
        contador = Integer.parseInt(f.readLine());
        f.close();

    } catch (FileNotFoundException e) {
        contador = 0;
    } catch (IOException e) {
        contador = 0;
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Servlet contador III

```
public final void destroy() {
    ServletContext ctx = config.getServletContext();
    String fichero = config.getInitParameter(strFichero);
    try {
        FileWriter f = new FileWriter(ctx.getRealPath(fichero));
        f.write(contador+"\n");
        f.close();
    } catch (IOException e) {
        throw new RuntimeException(e.getLocalizedMessage());
    }
}
```

1
2
3
4
5
6
7
8
9
10
11

Aplicación Hola I

```
TOMCAT_HOME/webapps/hola
|-- build.xml
|-- contador.txt
|-- html
|  |-- cabecera.html
|  |-- pie.html
|-- prj.el
|-- src
|  |-- Contador.java
|  |-- Debug.java
|  |-- EnvoltorioHTML.java
|  |-- Fecha.java
|  |-- HolaMundo.java
|  |-- IncluyeFichero.java
|  |-- Login.java
|  |-- Sesion.java
|-- WEB-INF
|  |-- classes
|  |  |-- Contador.class
|  |  |-- Debug.class
|  |  |-- EnvoltorioHTML.class
|  |  |-- Fecha.class
|  |  |-- HolaMundo.class
|  |  |-- IncluyeFichero.class
|  |  |-- Login.class
|  |  |-- Sesion.class
|-- lib
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

Aplicación Hola II

```
-- web.xml
```

28

Fichero web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<web-app>
  <display-name>Servlets de prueba</display-name>
  <servlet>
    <servlet-name>hola</servlet-name>
    <servlet-class>HolaMundo</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>contador</servlet-name>
    <servlet-class>Contador</servlet-class>
    <init-param>
      <param-name>fichero</param-name>
      <param-value>contador.txt</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>invoker</servlet-name>
    <url-pattern>/http/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>invoker</servlet-name>
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Aplicación Hola III

```
<url-pattern>/servlet/*</url-pattern>  
</servlet-mapping>  
</web-app>
```

22
23
24

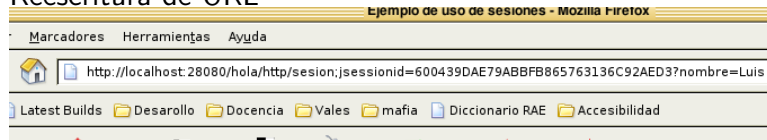
Sesiones

Memoria entre diferentes invocaciones a servlets.

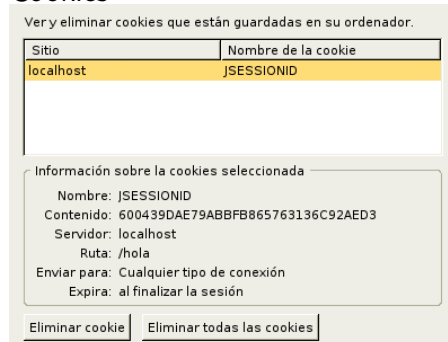
```
HttpSession sesion = request.getSession();  
String nombre =(String)sesion.getAttribute(strNombre);  
Integer numAccesos = (Integer)sesion.getAttribute(strNumAccesos);  
if (numAccesos==null) {  
    numAccesos = new Integer(0);  
}  
sesion.setAttribute(strNumAccesos, new Integer(numAccesos.intValue()+1));  
.....  
.....  
if (borroSesion!=null) {  
    contenido.append(sesionBorrada(response));  
    sesion.removeAttribute(strNumAccesos);  
    sesion.removeAttribute(strNombre);  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14

■ Reescritura de URL



■ Cookies



Toda URL debe ser *codificada*. Funciona tanto si tiene como si no tiene las *cookies* habilitadas.

```
private StringBuffer  
    formularioContinua(HttpServletRequest response) {  
    StringBuffer sb = new StringBuffer();  
    sb.append("<form type=\"get\" action=\"\"")  
    /* Se codifica la URL, para que sea  
       independiente de las cookies */  
    sb.append(response.encodeURL(config.getServletName()));  
    sb.append(">\n");  
    sb.append("<p><button name=\"continuar\" type=\"submit\">");  
    sb.append("Continuar</button></p>\n");  
    sb.append("<p><button name=\"\"");  
    sb.append(strBorraSesion);  
    sb.append("> type=\"submit\">Borrar sesion</button></p>\n");  
    sb.append("</form>\n");  
    return sb;  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Atributos

Se pueden guardar atributos

```
public void setAttribute(String name, Object o); 1
public void removeAttribute(String name); 2
public Object getAttribute(String name); 3
public Enumeration getAttributeNames(); 4
```

ServletRequest Atributos que *sólo viven* en una petición.

HttpSession Atributos que viven mientras dure la sesión.

ServletContext Atributos comunes para todos los servlets.

Filtros

Transformar la salida del servlet



Aplicaciones:

- autenticación,
- cifrado,
- transformaciones XML,
- lo que se quiera....

```

public class EnvoltorioHTML implements Filter {
private static final String cabecera="html/cabecera.html";
private static final String pie="html/pie.html";
private static final String strTitulo="titulo";
private FilterConfig conf;
public void init(FilterConfig filterConfig)
throws ServletException {
conf = filterConfig;
}
public void destroy() {
conf = null;
}
public void doFilter(ServletRequest request,
ServletResponse response,
FilterChain filterChain)
throws IOException, ServletException {
String fichero =
conf.getServletContext().getRealPath(cabecera);
String cabecera = IncludeFichero.incluye(fichero);
String titulo = conf.getInitParameter(strTitulo);
ServletOutputStream out = response.getOutputStream();
cabecera = cabecera.replaceAll(" <title>",
" <title>"+titulo);
out.println(cabecera);
filterChain.doFilter(request,response);
fichero = conf.getServletContext().getRealPath(pie);
out.println(IncludeFichero.incluye(fichero));
}
}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

html/cabecera.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15">
<title></title>
</head>
<body>

```

1
2
3
4
5
6
7

html/pie.html

```

<hr>
<address>
<a href="mailto:luis@ramonvazquez.net">
Luis Fernando Llana Diaz
</a>
</address>
</body>
</html>

```

1
2
3
4
5
6
7
8

```

..... parametros globales de la aplicacion
<filter>
  <filter-name>envoltorioHola</filter-name>
  <filter-class>EnvoltorioHTML</filter-class>
  <init-param>
    <param-name>titulo</param-name>
    <param-value>Hola Mundo</param-value>
  </init-param>
</filter>
<filter>
  <filter-name>envoltorioContador</filter-name>
  <filter-class>EnvoltorioHTML</filter-class>
  <init-param>
    <param-name>titulo</param-name>
    <param-value>Ejemplo de contador</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>envoltorioHola</filter-name>
  <url-pattern>/http/hola</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>envoltorioContador</filter-name>
  <url-pattern>/http/contador</url-pattern>
</filter-mapping>
definicion de servlets .....

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```

public class Login implements Filter {
.....
  public final void doFilter(final ServletRequest request,
                           final ServletResponse response,
                           final FilterChain filterChain)
    throws IOException, ServletException {
  HttpSession sesion =
    ((HttpServletRequest)request).getSession();
  String nombre=(String)sesion.getAttribute(Sesion.strNombre);
  if (nombre==null) {
    nombre = request.getParameter(strNombre);
    if (nombre==null) {
      pideNombre((HttpServletRequest)request,
                (HttpServletResponse)response);
    } else {
      sesion.setAttribute(Sesion.strNombre,nombre);
      filterChain.doFilter(request, response);
    }
  } else {
    filterChain.doFilter(request, response);
  }
}
}
}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

public class Login implements Filter {
private static final String strServlet="servlet";
private static final String strNombre="nombre";
private FilterConfig config;
public void init(FilterConfig filterConfig)
throws ServletException {
    config = filterConfig;
}
public void destroy() { }
private void pideNombre(final HttpServletRequest request,
final HttpServletResponse response)
throws IOException, ServletException {
    ServletOutputStream out = response.getOutputStream();
    String uri = request.getRequestURI();
    response.setContentType("text/html");
    out.println("<form type=\"get\" action=\""+response.encodeURL(uri)+"\">\n");
    out.println(" <p>\n");
    out.println("    <label for=\""+strNombre+"\">Nombre:</label>");
    out.println("    <input id=\""+strNombre+"\" name=\""+strNombre+"\">\n");
    out.println("    <button type=\"submit\">Login</button>\n");
    out.println(" </p>\n");
    out.println("</form>");
}
public final void doFilter(.....)
}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Invocando otros servlets

```

public class Fecha extends HttpServlet {
public static final String strServlet="/http/fecha";
public static final String strID="id";
public final void doGet(final HttpServletRequest request,
final HttpServletResponse response)
throws ServletException, IOException {
    ServletOutputStream out = response.getOutputStream();
    response.setContentType("text/html");
    Date hoy = new Date();
    DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);

    out.println(" <div id=\""+request.getAttribute(strID)+"\">");
    out.println(df.format(hoy));
    out.println(" </div>");
}
}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Incluir el resultado del servlet /http/fecha en cualquier parte

```
.....
RequestDispatcher rd =
    request.getRequestDispatcher(Fecha.strServlet);
request.setAttribute(Fecha.strID, "fecha");
rd.include(request, response);
.....
```

1
2
3
4
5
6

El servlet /http/sesion I

```
..... otros filtros
<filter>
  <filter-name>envoltorioSesion</filter-name>
  <filter-class>EnvoltorioHTML</filter-class>
  <init-param>
    <param-name>titulo</param-name>
    <param-value>Ejemplo de uso de sesiones</param-value>
  </init-param>
</filter>
<filter>
  <filter-name>loginSesion</filter-name>
  <filter-class>Login</filter-class>
  <init-param>
    <param-name>servlet</param-name>
    <param-value>sesion</param-value>
  </init-param>
</filter>
... otros filtros
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

El servlet /http/sesion II

```
..... otras disposiciones de filtros 1
<!-- el orden es importante -->      2
<filter-mapping>                     3
  <filter-name>envoltorioSesion</filter-name> 4
  <url-pattern>/http/sesion</url-pattern>    5
</filter-mapping>                   6
<filter-mapping>                     7
  <filter-name>loginSesion</filter-name>     8
  <url-pattern>/http/sesion</url-pattern>    9
</filter-mapping>                   10
... otras disposiciones de filtros     11
                                        12
```

```
..... otros servlets 1
<servlet>              2
  <servlet-name>sesion</servlet-name>      3
  <servlet-class>Sesion</servlet-class>    4
</servlet>             5
<servlet>              6
  <servlet-name>fecha</servlet-name>       7
  <servlet-class>Fecha</servlet-class>     8
</servlet>             9
... otros servlets    10
```

Fichero hola.xml

```
<Context docBase="${catalina.home}/server/webapps/hola"> 1
  <Logger className="org.apache.catalina.logger.SystemOutLogger" /> 2
  <Logger className="org.apache.catalina.logger.SystemErrLogger" /> 3
  4
  <Resource name="jdbc/DB-valores" auth="Container" 5
    type="javax.sql.DataSource" 6
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" 7
    maxActive="100" maxIdle="30" maxWait="10000" 8
    username="luis" password="patata" 9
    driverClassName="com.mysql.jdbc.Driver" 10
    url="jdbc:mysql://localhost:3306/valores?autoReconnect=true"/> 11
</Context> 12
```

Conexion con la base de datos

```
import java.sql.Connection;
import java.sql.SQLException;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import org.apache.commons.dbcp.BasicDataSource;

public class Conexion {

    public static Connection getConexionValores()
    throws SQLException, NamingException {
        InitialContext ctx = new InitialContext();
        BasicDataSource ds =
            (BasicDataSource)ctx.lookup("java:comp/env/jdbc/DB-valores");
        return ds.getConnection();
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Listado de valores

```
import soporte.html.Tags; //Clase para facilitar el uso de HTML/XML
public class Valores extends HttpServlet {
    public final void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        ServletOutputStream out = response.getOutputStream();
        String nombre = request.getParameter("nombre");
        response.setContentType("text/html");
        try {
            Connection con = Conexion.getConexionValores();
            PreparedStatement psmt = con.prepareStatement("select * from valores");
            StringBuffer lista = new StringBuffer();
            ResultSet rs = psmt.executeQuery();
            while (rs.next()) {
                lista.append(Tags.li("", Tags.a(rs.getString("url"), "", rs.getString("nombre"))));
            }
            out.print(Tags.h(1, "", "Lista de valores"));
            out.print(Tags.ul("", lista.toString()));
            con.close();
        } catch (SQLException e) {
            throw new ServletException(e.getMessage(), e);
        } catch (NamingException e) {
            throw new ServletException(e.getMessage(), e);
        }
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

Los filósofos con servlets I

```
public class Filosofo extends HttpServlet {
    .....
    private static final String strAccion="accion";

    public final void doGet(final HttpServletRequest request, final HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        HttpSession sesion = request.getSession();
        String accion = request.getParameter(strAccion);
        if ( accion==null || accion.equals(strPensar) ) {
            pensar(request,response);
        } else {
            comer(request,response);
        }
    }
    public final void doGet(final HttpServletRequest request, final HttpServletResponse response)
        throws ServletException, IOException {
        .....
    }
    public final void pensar(final HttpServletRequest request, final HttpServletResponse response)
        throws ServletException, IOException {
        .....
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

Los filósofos con servlets II

```
public static final String strFilosofoID="filosofoID";

public final void comer(final HttpServletRequest request, final HttpServletResponse response)
    throws ServletException, IOException {
    ServletOutputStream out = response.getOutputStream();
    HttpSession sesion = request.getSession();
    Integer ID = (Integer)sesion.getAttribute(strFilosofoID);
    try {
        mesa.permisoComer(ID.intValue());
        out.println(Tags.p("", "Filósofo "+ID+" comiendo, cuando quiera pensar el botón"));
        out.println(Tags.p("", "Estado del monitor:"+mesa));

        String URI = ((HttpServletRequest)request).getRequestURI();
        out.println(Tags.p("", Tags.form(response.encodeURL(URI), "get", "",
            Tags.button(strAccion, Tags.strSubmit, strPensar, "", "Pensar"))));
    } catch (InterruptedException e) {
        throw new ServletException(e.getMessage(), e);
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

Los filósofos con servlets III

```
private static final int NUM_FILOSOFOS=5;
public static final String strMesa="mesa";
private Mesa mesa;

public final void init(ServletConfig config) throws ServletException {
    ServletContext ctx = config.getServletContext();
    Mesa ctxMesa = (Mesa) ctx.getAttribute(strMesa);
    if (ctxMesa!=null ) {
        mesa = ctxMesa;
    } else {
        mesa = Mesa.getMesa(NUM_FILOSOFOS);
        ctx.setAttribute(strMesa,mesa);
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Los filósofos con servlets IV

```
public class Mesa {
    .....
    private static Mesa mesa=null;

    private Mesa(int n) {
        numFilosofos = n;
        comiendo = new boolean[numFilosofos];
        for (int i = 0; i < numFilosofos; i++) {
            comiendo[i] = false;
        }
    }

    public static Mesa getMesa(int n) {
        if (mesa==null) {
            mesa=new Mesa(n);
        } else {
            if (mesa.numFilosofos!=n) {
                throw new RuntimeException("Monitor de filósofos ya construido con "+
                    mesa.numFilosofos+" filósofos. No puedo darte uno de "+n);
            }
        }
        return mesa;
    }
    .....
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Los filósofos con servlets V

```
public class Login implements Filter {
    .....
    public final void doFilter(final ServletRequest request,
                              final ServletResponse response,
                              final FilterChain filterChain)
        throws IOException, ServletException {
        HttpSession sesion = ((HttpServletRequest)request).getSession();
        Integer filosofoID=(Integer)sesion.getAttribute(Filosofo.strFilosofoID);
        if (filosofoID==null) {
            String ID_str = request.getParameter(Filosofo.strFilosofoID);
            if (ID_str==null) {
                pideNombre((HttpServletRequest)request, (HttpServletResponse)response);
            } else {
                try {
                    Integer ID=new Integer(ID_str);
                    sesion.setAttribute(Filosofo.strFilosofoID, ID);
                    filterChain.doFilter(request, response);
                } catch (NumberFormatException e) {
                    pideNombre((HttpServletRequest)request, (HttpServletResponse)response);
                }
            }
        } else {
            filterChain.doFilter(request, response);
        }
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Los filósofos con servlets VI

.....

26

Los filósofos con servlets VII

```
<?xml version="1.0" encoding="ISO-8859-1"?> 1
<!DOCTYPE web-app 2
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" 3
"http://java.sun.com/dtd/web-app_2_3.dtd"> 4
5
<web-app> 6
  <display-name>Filósofos</display-name> 7
8
  <filter> 9
    <filter-name>envoltorio</filter-name> 10
    <filter-class>EnvoltorioHTML</filter-class> 11
    <init-param> 12
      <param-name>titulo</param-name> 13
      <param-value>Filósofos</param-value> 14
    </init-param> 15
  </filter> 16
17
  <filter> 18
    <filter-name>loginSesion</filter-name> 19
    <filter-class>Login</filter-class> 20
  </filter> 21
22
  <filter> 23
    <filter-name>MierdaExplorer</filter-name> 24
    <filter-class>soporte.servlet.MierdaExplorer</filter-class> 25
```

Los filósofos con servlets VIII

```
</filter> 26
27
<filter-mapping> 28
  <filter-name>envoltorio</filter-name> 29
  <url-pattern>/http/filosofo</url-pattern> 30
</filter-mapping> 31
<filter-mapping> 32
  <filter-name>MierdaExplorer</filter-name> 33
  <url-pattern>/http/*</url-pattern> 34
</filter-mapping> 35
<filter-mapping> 36
  <filter-name>loginSesion</filter-name> 37
  <url-pattern>/http/filosofo</url-pattern> 38
</filter-mapping> 39
40
<servlet> 41
  <servlet-name>filosofo</servlet-name> 42
  <servlet-class>Filosofo</servlet-class> 43
</servlet> 44
45
<servlet-mapping> 46
  <servlet-name>invoker</servlet-name> 47
  <url-pattern>/http/*</url-pattern> 48
</servlet-mapping> 49
<servlet-mapping> 50
  <servlet-name>invoker</servlet-name> 51
```

Los filósofos con servlets IX

```
<url-pattern>/servlet/*</url-pattern>  
</servlet-mapping>  
</web-app>
```

52
53
54

Los filósofos con servlets X

```
public class MierdaExplorer implements Filter {  
    private static String javascript="javascript/mierdaExplorer.js";  
  
    private static boolean esIE(HttpServletRequest req) {  
        String userAgent = req.getHeader("user-agent");  
        return userAgent.toLowerCase().indexOf("msie")>0;  
    }  
  
    public final void doFilter(final ServletRequest _request,  
                              final ServletResponse response,  
                              final FilterChain filterChain)  
        throws IOException, ServletException {  
        HttpServletRequest request = (HttpServletRequest)_request;  
        HttpServletResponse responseString  
            = new ResponseString((HttpServletResponse)response);  
        ServletOutputStream out = responseString.getOutputStream();  
        Enumeration nombres = request.getHeaderNames();  
        if (esIE(request)) {  
            ServletContext ctx = config.getServletContext();  
            out.println(Tags.script("<script>IncluyeFichero.incluye(ctx.getRealPath(javascript))</script>"));  
            out.println(Tags.p("<p>Mierda de explorer"));  
        } else {  
            out.println(Tags.p("<p>Menos mal que es un navegador decente"));  
        }  
        filterChain.doFilter(request, responseString);  
    }  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Los filósofos con servlets XI

```
        out = response.getOutputStream();
        String res = responseString.toString();
        if (esIE(request)) {
            res = reemplazaBoton(res);
        }
        out.println(res);
        if (esIE(request)) {
            out.println(Tags.p(Tags.clase("disclaimer"),"This pages require an HTML3.01 compliant navigator. Unfortunately Internet Explorer is not, thus we do not guarantee their correct behavior"));
        }
    }

    public static String reemplazaBoton(String codigo) {
        Pattern p = Pattern.compile("<button *name=\"([^\"]*)\" *\"+
            \"type=\"([^\"]*)\" *\"+
            \"value=\"([^\"]*)\" *\"+
            \"([^\"]*)>([^\"]*)</button>");

        String res = codigo;
        Matcher m = p.matcher(res);
        while (m.find()) {
            String tipo=m.group(2);
            String nombre=m.group(1);
            String valor=m.group(3);
            String attrExtra=m.group(4);
            String texto=m.group(5);
            String nuevo =
```

```
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

Los filósofos con servlets XII

```
        Tags.input("",nombre,Tags.type(tipo)+
            Tags.value(texto)+
            Tags.attribute("onclick","cambiaValor(this,'"+valor+"')"+
            attrExtra);
        res = m.replaceFirst(nuevo);
        m = p.matcher(res);
    }
    return res;
}

private FilterConfig config;
public void init(FilterConfig filterConfig) throws ServletException {
    config = filterConfig;
}

public void destroy() {
    config=null;
}
}
```

```
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
```

Los filósofos con servlets XIII

```
public class ResponseString extends HttpServletResponseWrapper {
    private StringOutputStream out;

    public ResponseString(HttpServletResponse response) {
        super(response);
        out = new StringOutputStream();
    }
    public ServletOutputStream getOutputStream() {
        return out;
    }
    public String toString() {
        return out.toString();
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Los filósofos con servlets XIV

```
class StringOutputStream extends ServletOutputStream {
    private ByteArrayOutputStream array;
    public StringOutputStream() {
        array = new ByteArrayOutputStream();
    }
    public void write (int b) {
        array.write(b);
    }
    public String toString() {
        try {
            return array.toString("iso-8859-15");
        } catch (Exception e) {
            throw new RuntimeException(e.getMessage(), e);
        }
    }
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16