

8.13. Valves

- Implementa y prueba el *Valve* que se propone en la práctica.
- Observa el funcionamiento del **AccessLogValve** definido por defecto en *Tomcat*.
- Configura el *Valve* predefinido **RemoteAddrValve** para que solo se pueda acceder a las aplicaciones desde **DesarrolloW7XX** (192.168.1.X6).

1. LogginValve.

- 1.1. Inicia sesión en **DesarrolloW7XX**.
- 1.2. Crea un proyecto Java (**File, New, Project, Java, Java Project**).
- 1.3. Llámalo *valve* y pincha en **Next**.
- 1.4. Accede a la pestaña **Library** y añade (**Add External Jar**) las librerías **catalina.jar** y **servlet-api.jar** (la puedes encontrar en la instalación local de *Tomcat*), Figura 8.100. Pincha en **Finish** para crear el proyecto.

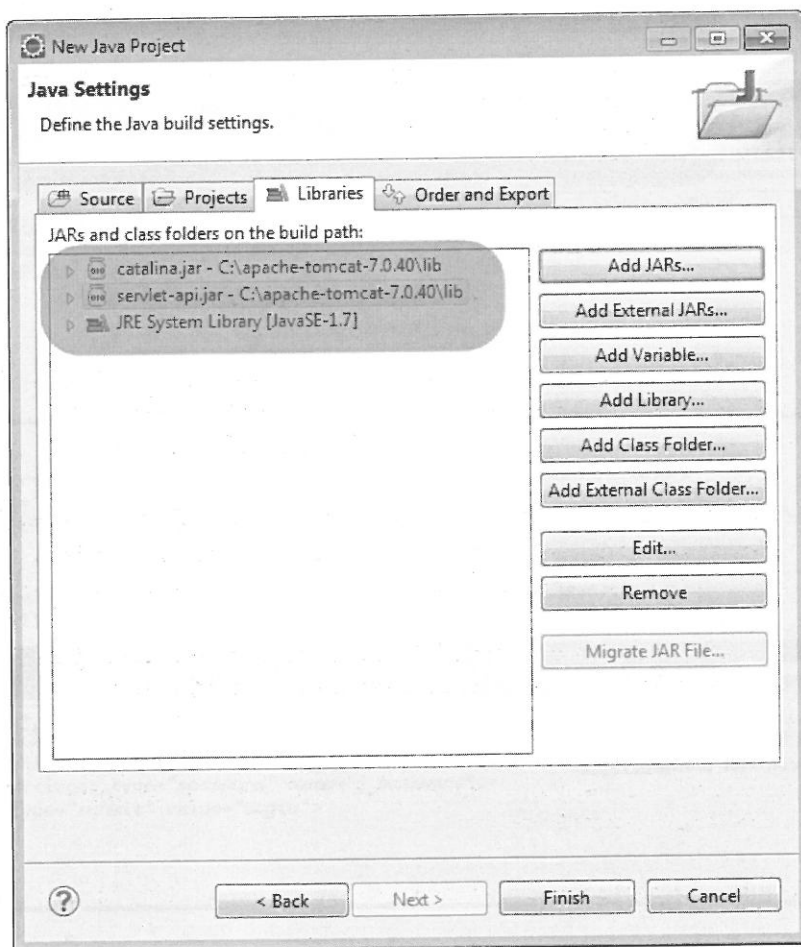


Figura 8.100: Librerías en el proyecto **valve**

- 1.5. Crea la clase **LogginValve** en el paquete **net.daw01.valves** con el siguiente contenido, Figura 8.101.

```
package net.daw01.valves;

import java.io.IOException;

import javax.servlet.*;

import org.apache.catalina.Valve;
import org.apache.catalina.valves.*;
import org.apache.catalina.connector.Request;
import org.apache.catalina.connector.Response;

public class LogginValve extends ValveBase {

    @Override
    public void invoke(Request request, Response response)
        throws IOException, ServletException {

        String IPRemota = request.getRemoteAddr();
        String URI = request.getRequestURI();
        System.out.println("URI " + URI + " accedida desde: "+IPRemota);
        Valve nextValve = getNext();
        if(nextValve!=null){
            nextValve.invoke(request, response);
        }
    }
}
```

Figura 8.101: Clase **LogginValve**

- 1.6. Exporta la clase **LogginValve.class** al fichero **despliegue-valves.jar** (sobre el proyecto **valve**, botón derecho de ratón, *Export, Java, JAR File*).
- 1.7. Sube el fichero **despliegue-valves.jar** a **\$CATALINA_HOME/lib** en **ServidorLinuxXX**.
- Usa *Filezilla* para subir el fichero **despliegue-valves.jar** a **/home/alumno**.
 - En **ServidorLinuxXX** cambia el propietario y el grupo del fichero al usuario **root** y grupo **root** y concédele permisos de lectura y escritura para todos los usuarios.

```
sudo chown root:root /home/alumno/despliegue-valves.jar
sudo chmod 777 //home/alumno/despliegue-valves.jar
```

- Mueve el fichero desde del directorio **home** del usuario **alumno** al directorio **/usr/share/tomcat7/lib**.

```
sudo mv /home/alumno/despliegue-valves.jar /usr/share/tomcat7/lib
```

- 1.8. Edita el fichero **/var/lib/tomcat7/conf/server.xml** y añade los siguientes elementos para configurar el *valve* en todas las aplicaciones de Host **localhost**, Figura 8.102.

```

<Host name="localhost" appBase="webapps"
    unpackWARs="true" autoDeploy="true">

    <!-- SingleSignOn valve, share authentication between web applications
    Documentation at: /docs/config/valve.html -->
    <!--
    <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
    -->

    <!-- Access log processes all example.
    Documentation at: /docs/config/valve.html
    Note: The pattern used is equivalent to using pattern="common" -->
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="$
    prefix="localhost_access_log." suffix=".txt"
    pattern="%h %l %u %t &quot;%r&quot; %s %b" />

    <Valve className="net.daw01.valves.LoginValve" />

```

Figura 8.102: Configuración de **LoginValve** en el fichero **server.xml**

- 1.9. Reinicia *Tomcat*.
- 1.10. Desde **DesarrolloW7XX** accede a varias aplicaciones del servidor.
- 1.11. En **ServidorLinuxXX** consulta el fichero **/var/log/tomcat7/catalina.out** y observa lo que ha escrito el *valve* que hemos configurado, Figura 8.103.

```

alumno@ServidorLinux01:/var/log/tomcat7$ tail -f catalina.out
URI /compras/ accedida desde: 192.168.1.16
URI /favicon.ico accedida desde: 192.168.1.16
URI /compras/productos.jsp accedida desde: 192.168.1.16
URI /favicon.ico accedida desde: 192.168.1.16
URI /curso/ accedida desde: 192.168.1.16
URI /curso/ accedida desde: 192.168.1.16
URI /curso/ accedida desde: 192.168.1.16
URI /favicon.ico accedida desde: 192.168.1.16
URI /curso/buenas.jsp accedida desde: 192.168.1.16
URI /favicon.ico accedida desde: 192.168.1.16

```

Figura 8.103: Fichero **/var/log/tomcat7/catalina.out**

2. LoginValve.

- 2.1. Edita el fichero **var/lib/tomcat7/conf/server.xml** y observa el *valve* creado por defecto. Analiza su configuración, Figura 8.104.
- 2.2. Consulta el directorio **/var/log/tomcat7** y observa los ficheros **localhost_access_log**

3. RemoteAddrValve.

- 3.1. Consulta la documentación del valve http://tomcat.apache.org/tomcat-7.0-doc/config/valve.html#Remote_Address_Filter

```
<!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory=
      prefix="localhost_access_log." suffix=".txt"
      pattern="%h %l %u %t &quot;%r&quot; %s %b" />
```

Figura 8.104: Configuración de **AccessLogValve** en el fichero **server.xml**

- 3.2. Edita el fichero **/var/lib/tomcat7/conf/server.xml** y añade los siguientes elementos para configurar un *RemoteAddrValve* que solo permita las conexiones desde la ip 192.168.1.X6, Figura 8.105.

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

  <!-- SingleSignOn valve, share authentication between web applications
      Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory=
      prefix="localhost_access_log." suffix=".txt"
      pattern="%h %l %u %t &quot;%r&quot; %s %b" />

  <Valve className="net.daw01.valves.LoginValve" />

  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="192.168.1.16" />

</Host>
```

Figura 8.105: Configuración de **RemoteAddrValve** en el fichero **server.xml**

- 3.3. Reinicia *Tomcat*.
- 3.4. Prueba a acceder a alguna aplicación desde **DesarrolloW7XX**.
- 3.5. Prueba a acceder a alguna aplicación desde **ServidorWindowXX**, Figura 8.106.

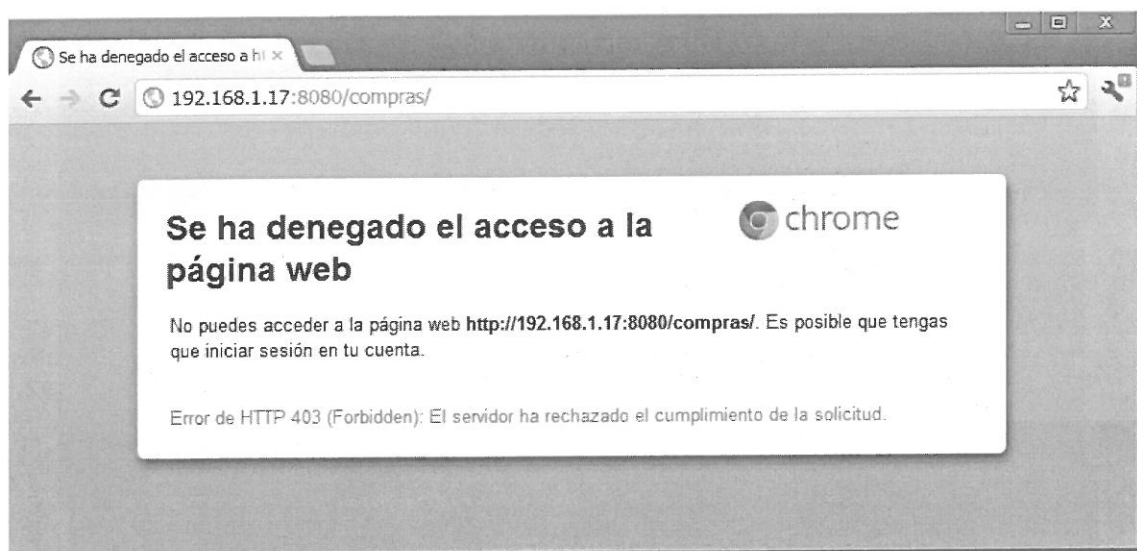


Figura 8.106: Denegación de acceso

3.14. Filtros

Implementa un filtro que cuente el número de accesos a la aplicación **curso** y deje constancia en fichero **/var/log/catalina.out**.

1. Inicia sesión en **DesarrolloW7XX**.
2. Crea un nuevo filtro (FiltroContador) en el proyecto **curso** con el siguiente contenido, Figura 8.107.
3. Edita el descriptor de despliegue de la aplicación **curso** y configura el filtro, Figura 8.108.
4. Elimina la aplicación **curso** del servidor *Tomcat*.
5. Genera el fichero **curso.war** con los cambios.
6. Despliega el nuevo fichero **curso.war**.
7. Desde **DesarrolloW7XX** accede a varias páginas de la aplicación **curso**.
8. En **ServidorLinuxXX** consulta el fichero **/var/log/tomcat7/catalina.out** y observa lo que ha escrito el filtro que hemos configurado, Figura 8.109.

```
package net.daw01.curso;

import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;

public class FiltroContador implements Filter {

    private FilterConfig config;

    public void init(FilterConfig config) {
        this.config = config;
    }

    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        // Contexto de sesión de almacenará el contador
        ServletContext contexto = config.getServletContext();
        // Se comprueba si el atributo contador existe dentro del contexto
        Integer contador = (Integer) contexto.getAttribute("contador");
        if (contador == null) {
            contador = new Integer(0);
        }
        contador = new Integer(contador.intValue() + 1);
        contexto.setAttribute("contador", contador);
        System.out.println();
        System.out.println("Número de accesos a curso : " + contador.intValue());
        // Para que se invoque al siguiente filtro
        chain.doFilter(request, response);
    }

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

Figura 8.107: Clase FiltroContador.java

```
<filter>
  <filter-name>FiltroContador</filter-name>
  <filter-class>net.daw01.curso.FiltroContador</filter-class>
</filter>
<filter-mapping>
  <filter-name>FiltroContador</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Figura 8.108: Fichero web.xml

```

alumno@ServidorLinux01:/var/log/tomcat7$ tail -f catalina.out
Número de accesos a curso : 1
URI /favicon.ico accedida desde: 192.168.1.16
URI /curso/hola accedida desde: 192.168.1.16

Número de accesos a curso : 2
URI /favicon.ico accedida desde: 192.168.1.16
URI /curso/buenas.jsp accedida desde: 192.168.1.16

Número de accesos a curso : 3
URI /favicon.ico accedida desde: 192.168.1.16

```

Figura 8.109: Fichero `/var/log/tomcat7/catalina.out`

8.15. Ficheros de registros (*logs*)

Consulta los ficheros de configuración de *Tomcat* y observa las directivas utilizadas para definir la configuración de los ficheros de registros (*logs*). Posteriormente consulta estos ficheros.

1. Inicia sesión en **DesarrolloW7XX**.
2. Consulta el fichero `/var/lib/tomcat7/conf/logging.properties` y busca las directivas que definen la localización de los ficheros de registros (*logs*) de errores, Figura 8.110.

```

# limitations under the License.

handlers = 1catalina.org.apache.juli.FileHandler, 2localhost.org.apache.juli.Fi$
handlers = 1catalina.org.apache.juli.FileHandler, java.util.logging.ConsoleHan$

#####
# Handler specific properties.
# Describes specific configuration info for Handlers.
#####

1catalina.org.apache.juli.FileHandler.level = FINE
1catalina.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.FileHandler.prefix = catalina.

2localhost.org.apache.juli.FileHandler.level = FINE
2localhost.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
2localhost.org.apache.juli.FileHandler.prefix = localhost.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter

```

Figura 8.110: Fichero `/var/lib/tomcat7/conf/logging.properties`

3. Consulta el fichero `/var/lib/tomcat7/conf/server.xml` y busca el *Valve* que definen la localización y el formato de los ficheros de registros (*logs*) de accesos, Figura 8.111.

```
<!-- Access log processes all example.  
Documentation at: /docs/config/valve.html  
Note: The pattern used is equivalent to using pattern="common" -->  
<Valve className="org.apache.catalina.valves.AccessLogValve"  
    directory="logs"  
    prefix="localhost_access_log." suffix=".txt"  
    pattern="%h %l %u %t &quot;%r&quot; %s %b" />
```

Figura 8.111: Fichero `/var/lib/tomcat7/conf/server.xml`

4. Consulta el directorio `/var/lib/tomcat7/logs`, Figura 8.112.

```
alumno@ServidorLinux01:/var/log/tomcat7$ ls  
catalina.2015-06-16.log    localhost.2015-06-17.log  
catalina.2015-06-17.log    localhost_access_log.2015-06-16.txt  
catalina.out              localhost_access_log.2015-06-17.txt  
localhost.2015-06-16.log
```

Figura 8.112: Directorio `/var/lib/tomcat7/logs`

5. Consulta el contenido de los ficheros de *logs*.