Hadrián
Villar Cuadrado

Tema 2: Aspectos de seguridad en los servidores de aplicaciones

Condiciones de entrega:

La forma de entregar los ejercicios será en un fichero ZIP con el formato **ApellidosNombreTarea2.pdf**.

Puntuación:

El boletín está puntuado sobre 10 puntos. En caso de no entregarlo en fecha, hay una prórroga de 3 días en las que se puede entregar con penalización de 1 punto por día hasta un mínimo de 3.5. A partir del tercer día la tarea será puntuada con un cero.

El ejercicio debe funcionar cuando se ejecute.

La nota mínima para considerar aprobado el boletín es de 3.5 puntos.

Ejercicios:

1. Crearemos una aplicación en Java Web con las siguientes características:
    a. Importaremos el servlet **servletFactorial.java** adjunto a nuestro proyecto.
    (1 punto)

    b. Debemos crear un rol “alumno” y asignárselo a un usuario con vuestro nombre. Además, crearemos un rol “profesor” y se lo asignareis al usuario “martin”. **Debéis hacer una captura del fichero donde hayáis hecho esos cambios.**
    (3 puntos)

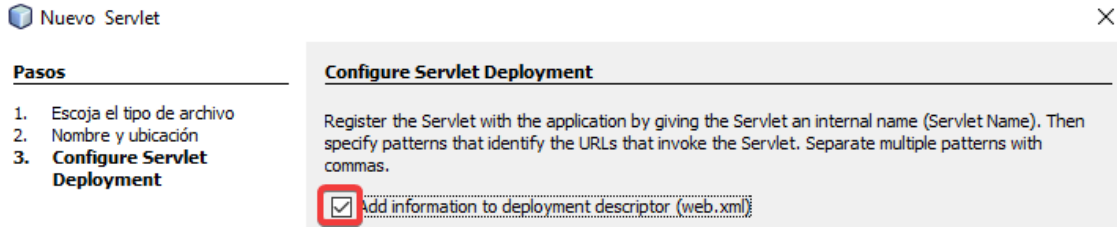
    c. Configuraréis esa aplicación para que **SOLAMENTE** se pueda acceder utilizando el protocolo HTTPS. Además, el único usuario que podrá acceder será el que tenga el rol de “profesor”.
    (3 puntos)

    d. Crearéis una Valve **PARA TODO EL SERVIDOR TOMCAT** donde:
        i. Se permita el acceso a la IP 192.168.0.2
        ii. Se deniegue el acceso a la IP 192.168.0.3

    (3 puntos)

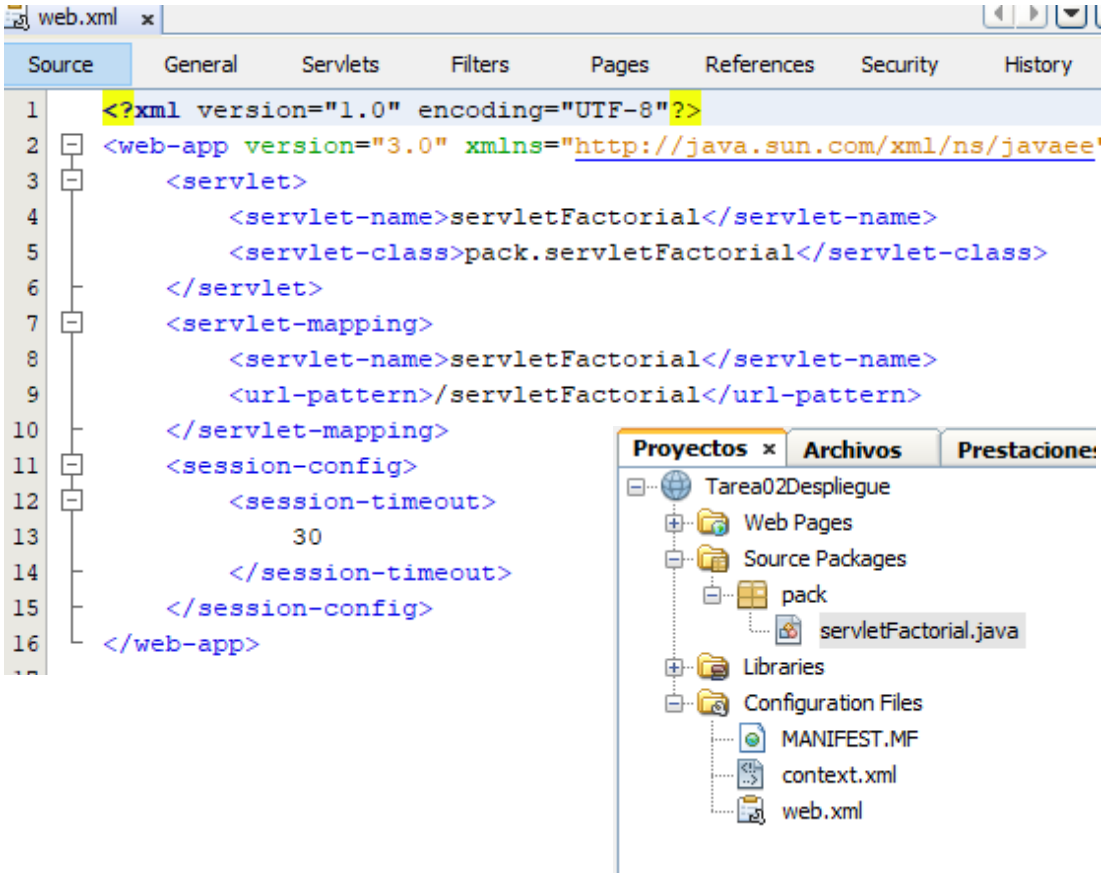
a. Como importar servlet “servletFactorial”.

Para ello debemos hacer **click derecho** sobre el proyecto > **Nuevo** > **Servlet** y completar los campos con la información necesaria, es muy importante dejar este **checkbox** como se muestra a continuación, ya que gracias a esto Netbeans nos ahorrará trabajo en el



web.xml.

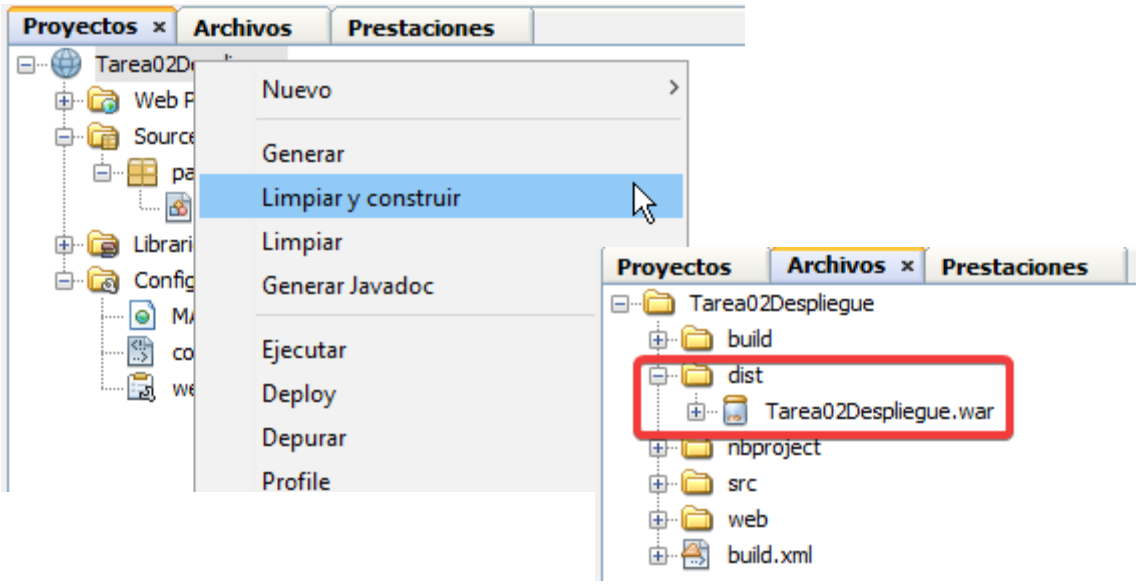
Netbeans generará el archivo **web.xml** que se mostrará a continuación. Lo encontraremos dentro del apartado “**Proyectos**”, en los archivos de configuración.



Una vez configurado el servlet de esta manera solo queda incluir el código a "importar" en el archivo .java, en este caso **servletFactorial.java**.

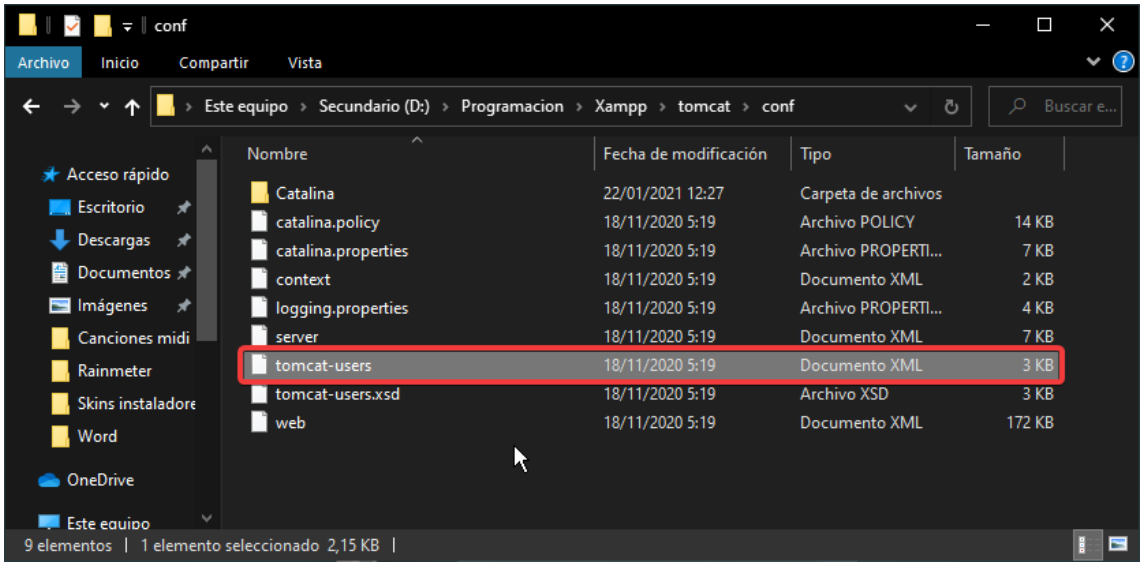
```
58      */
59      private int numero;
60
61      @Override
62      protected void doGet(HttpServletRequest request, HttpServletResponse response)
63          PrintWriter out = response.getWriter();
64
65          numero = Integer.parseInt(request.getParameter("numero"));
66
67          out.println(calculaFactorial());
68      }
69
70      @Override
71      protected void doPost(HttpServletRequest request, HttpServletResponse response)
72          doGet(request, response);
73      }
74
75      private int calculaFactorial() {
76          int resultado = 1;
77          for (int i = numero; i > 0; i--) {
78              resultado *= i;
79          }
80          return resultado;
81      }
```

Lo último que faltaría por hacer sería hacer **click derecho** en el proyecto y presionar "**Limpiar y construir**", esto generará el archivo **.war** (localizado en el apartado "**Archivos**" en la carpeta **dist**) que hemos de introducir en la máquina virtual o en la carpeta "**tomcat/webapps**" de la carpeta en la que tengamos instalado el **XAMPP**.



b. Creación de roles y usuarios.

Para la creación de roles hemos de localizar la carpeta "**tomcat/conf**" y modificar el archivo "**tomcat-users.xml**".



Incluyendo a continuación los **usuarios** y **roles** propuestos en la tarea:

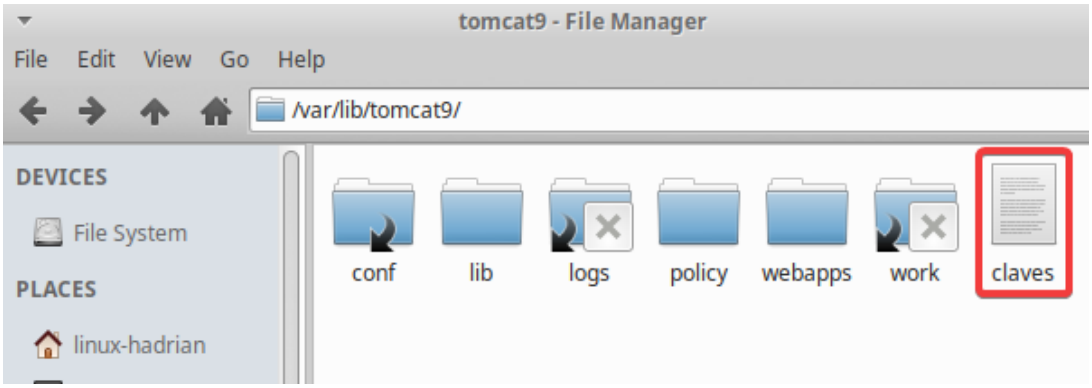
```
37
38
39      <role rolename="tomcat"/>
40      <role rolename="role1"/>
41
42      <role rolename="alumno"/>
43      <role rolename="profesor"/>
44
45      <user username="hadrian" password="tomcat" roles="alumno,manager-gui"/>
46      <user username="martin" password="tomcat" roles="profesor,manager-gui"/>
47
48      <user username="tomcat" password="tomcat" roles="tomcat"/>
49      <user username="both" password="tomcat" roles="tomcat,role1"/>
50      <user username="role1" password="tomcat" roles="role1"/>
51
52  </tomcat-users>
```

c. Configuración SSL para solo utilizar protocolo HTTPS. Y que solamente pueda acceder un rol en específico.

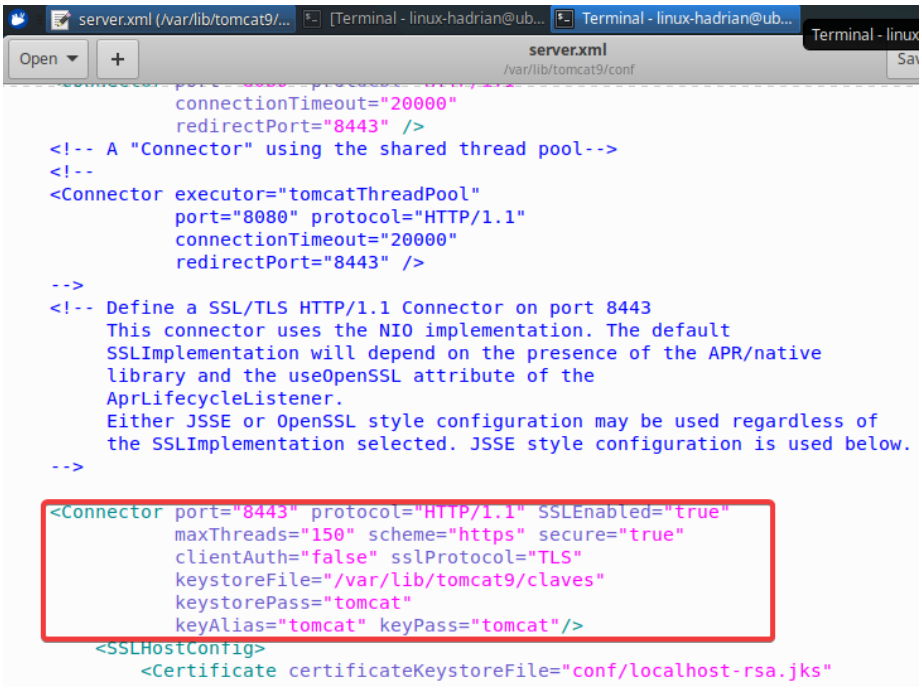
En primer lugar, hemos de crear un **almacén de claves** con un **certificado SSL**, para ello generaremos el certificado con la aplicación "**keytool**" como se muestra a continuación:

```
linux-hadrian@ubuntu:~$ sudo keytool -genkey -alias tomcat -keyalg RSA -keystore /var/lib/tomcat9/claves
[sudo] password for linux-hadrian:
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Hadrian
What is the name of your organizational unit?
  [Unknown]:  Alumno
What is the name of your organization?
  [Unknown]:  Montecastelo
What is the name of your City or Locality?
  [Unknown]:  Vigo
What is the name of your State or Province?
  [Unknown]:  Pontevedra
What is the two-letter country code for this unit?
  [Unknown]:  ES
Is CN=Hadrian, OU=Alumno, O=Montecastelo, L=Vigo, ST=Pontevedra, C=ES correct?
  [no]:  y

Enter key password for <tomcat>
        (RETURN if same as keystore password):

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool
 -importkeystore -srckeystore /var/lib/tomcat9/claves -destkeystore /var/lib/tomcat9/claves -deststoretype pkcs12".
linux-hadrian@ubuntu:~$
```

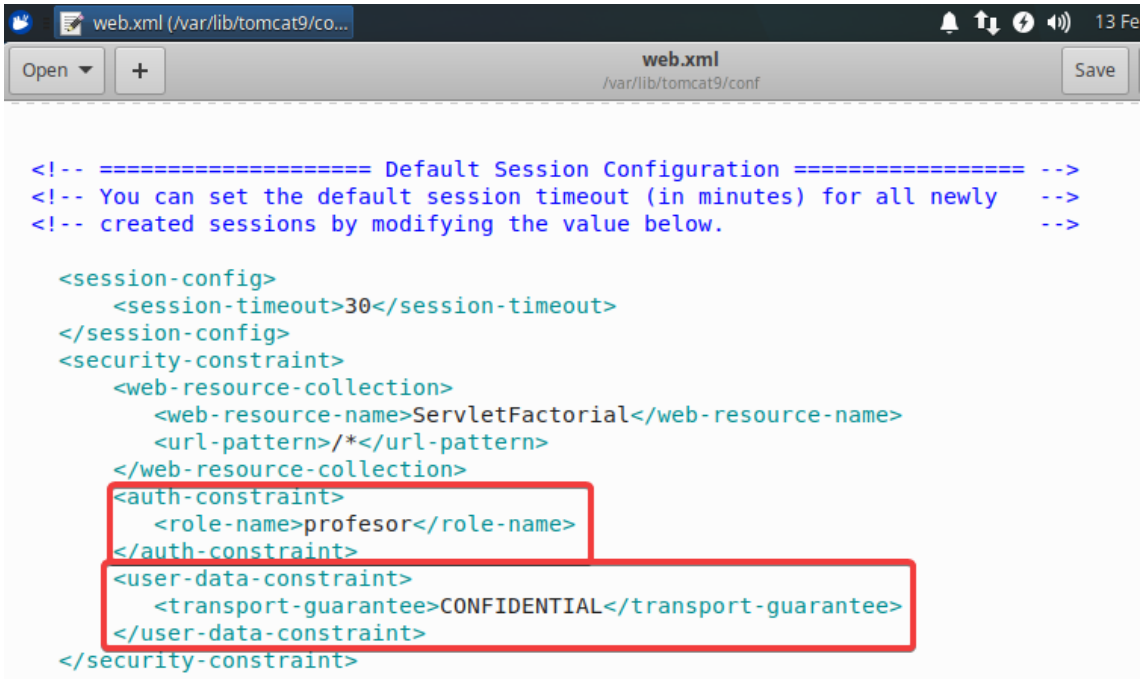
Para comprobar si lo ha creado, debemos ir a la ruta introducida en **-keystore**:



Después de generar el certificado, hemos de crear un **conector SSL** en Tomcat, editando el fichero **server.xml** localizado, en este caso (distribución de **Linux**), en la carpeta "**var/lib/tomcat9/conf**" añadiendo el fragmento de código que se muestra en la imagen inferior:



Por último, para que solo acepte conexiones **HTTPS** de los usuarios con el **rol** "**profesor**", debemos de añadir en el archivo **web.xml** los siguientes elementos dentro del **<security-constraint>:**



d. Creación valves para TODO EL SERVIDOR TOMCAT

En este caso para la creación de la **valve**, como queremos que intercepte **todas las peticiones** que le lleguen al **servidor**, hemos de modificar el archivo **server.xml** dentro de la etiqueta **<Engine>** incluyendo la línea de código destacada en la imagen siguiente: