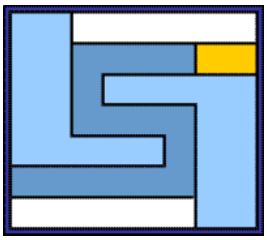

Ítem 6 - Report

Diseño y Pruebas

Grado de Ingeniería del Software

Curso 3

Armando Garrido Castro
Jorge Puente Zaro
Manuel Enrique Pérez Carmona
Cesar García Pascual
Pablo Tabares García
Rafael Trujillo González

Fecha: 27 de febrero de 2018

1. Introducción

Para este ítem se nos pide implementar el protocolo HyperText Transfer Protocol Secure, en adelante HTTPS. Utiliza un cifrado basado en SSL/TLS para crear un canal cifrado, este protocolo es el más apropiado para el tráfico de información sensible que el protocolo HTTP. El puerto estándar que utiliza este protocolo es el 443.

2. Creación de claves

En primer lugar es necesario generar las claves RSA necesarias para establecer una conexión segura. Para ello, ejecutamos la siguiente línea de comando en nuestro entorno de desarrollo.
En esta línea de comando indicaremos la contraseña que luego indicaremos en el fichero server.xml.

```
keytool -genkey -alias tomcat -keyalg RSA -storepass p@$$w0rD! -keypass p@$$w0rD! -dname CN=tomcat
```

Al ejecutar este comando generará un nuevo archivo en el directorio home del usuario que lo ejecuta, con el nombre ".keystore".

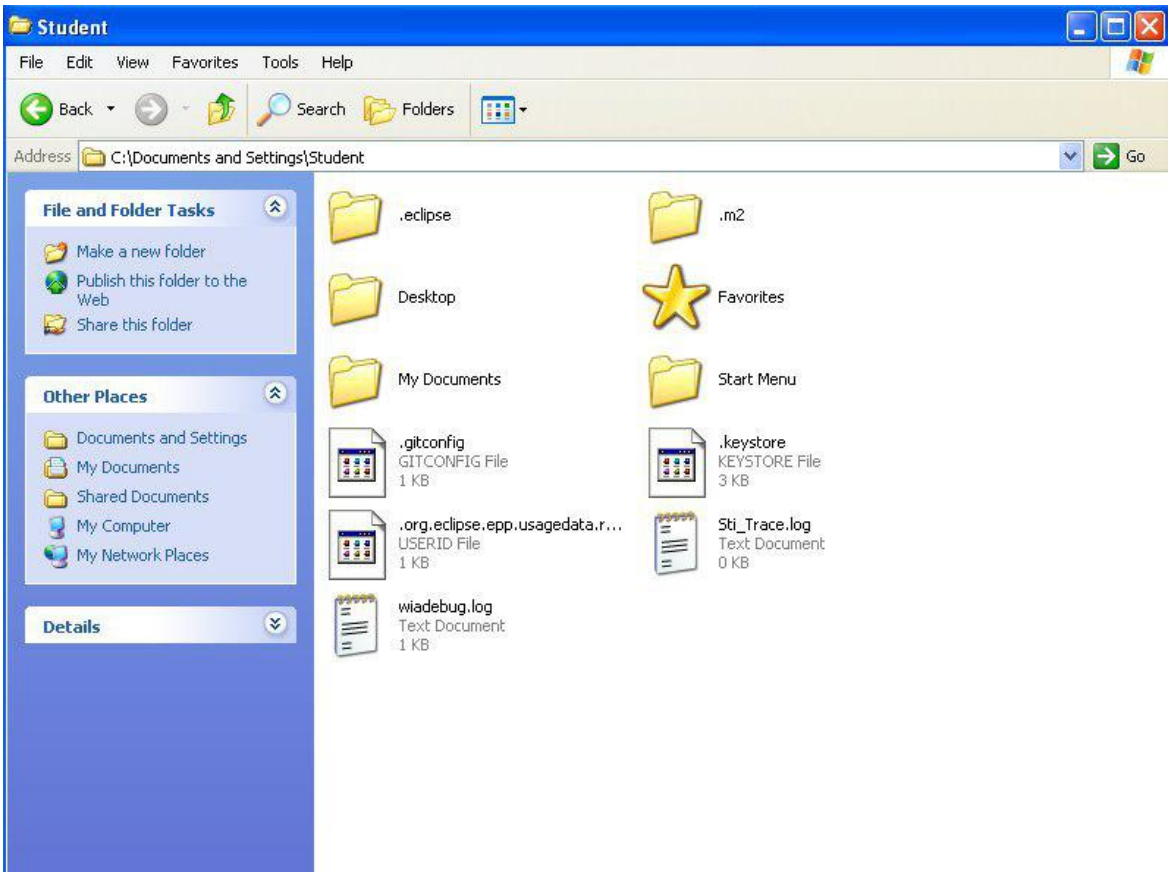

Ilustración 1: Carpeta home

De este modo ya habremos finalizado el primer paso para la implementación del protocolo HTTPS.

3. Configuración del Servidor

Una vez generado el archivo .keystore pasamos a la configuración de nuestro servidor Tomcat dentro de nuestro de desarrollo.

Para llevar a cabo la configuración del servidor en primer lugar tenemos que localizar el archivo server.xml. Para ello accederemos dentro de nuestro workspace.

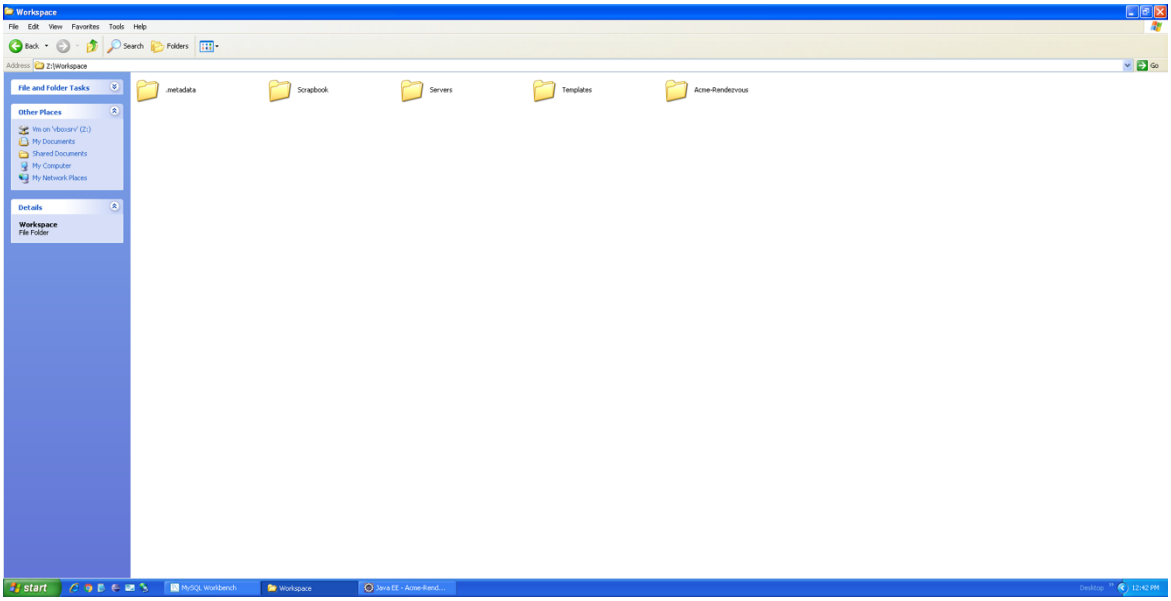

Ilustración 2:Workspace

Una vez dentro del workspace accedemos a la carpeta Servers.

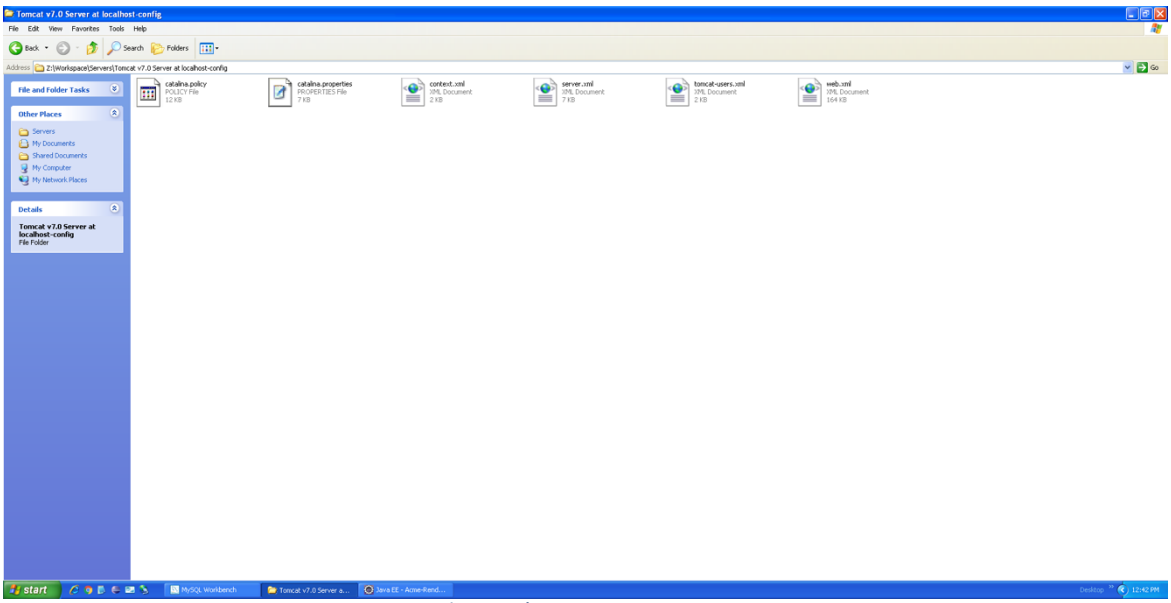

Ilustración 3: Carpeta Servers

En ella podemos observar que se encuentra el fichero server.xml el cual editaremos para añadir el siguiente fragmento de código, junto a los demás conectores que ya contiene el fichero.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="${user.home}/.keystore" keystorePass="p@$$w0rD!" />
```

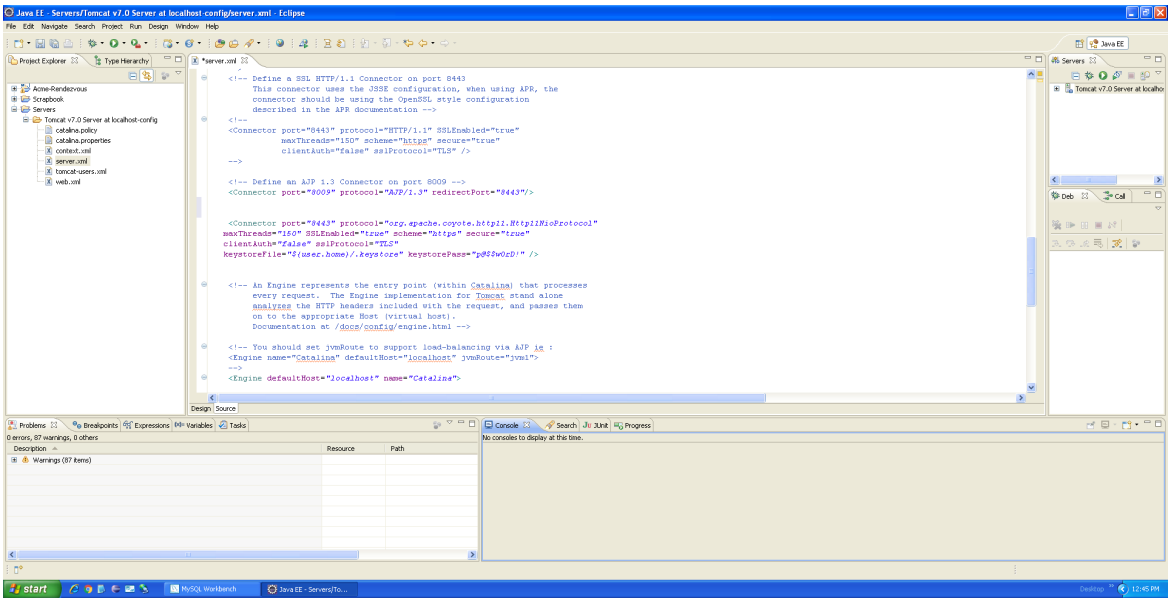

Ilustración 4: Fichero Server.xml editado

En este conector indicaremos el puerto de conexión, en este caso el 8433, habilitaremos SSL, indicaremos que el esquema utilizado es el https y que utilizaremos el TLS como protocolo SSL. Finalmente, indicamos la ruta donde podemos encontrar el archivo .keystore y la clave necesaria para su utilización. Es de suma importancia cerciorarse de que las claves introducidas en la línea de comando y en el conector son exactamente las mismas.

Una vez hemos añadido el conector al fichero server.xml tenemos comprobar que el servidor Tomcat está configurado de manera correcta. Para comprobarlo pulsaremos dos veces sobre nuestro servidor, este se encuentra la ventana Servers en el entorno de desarrollo Eclipse, como podemos observar en la siguiente ilustración.

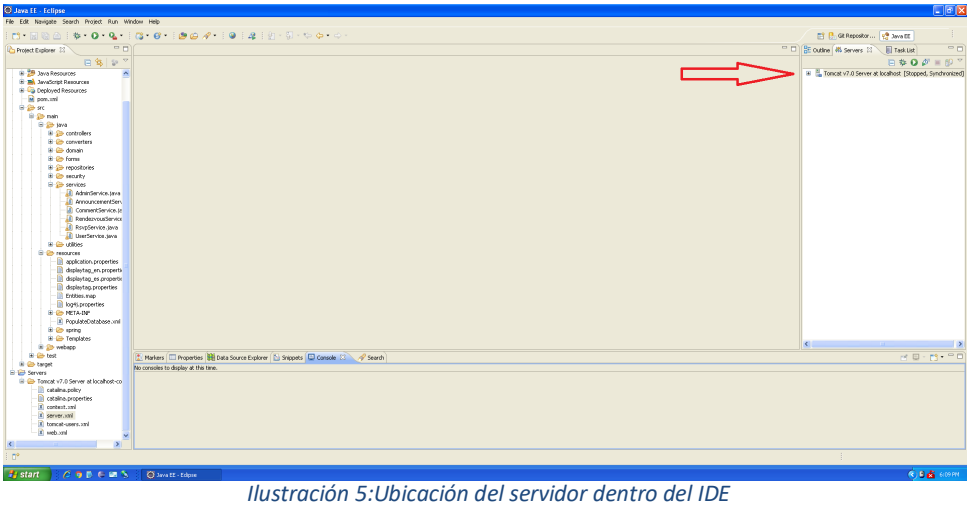

Ilustración 5:Ubicación del servidor dentro del IDE

Al pulsar dos veces sobre el nombre del servidor se abrirá un panel de mandos divididos en varios apartados. Hay que comprobar que la sección llamada Ports tenga los siguientes puertos configurados de la siguiente forma.

Tomcat admin port: 8005 **http/1.1: 8080** **SSL:8443** **AJP/1.3:8009**

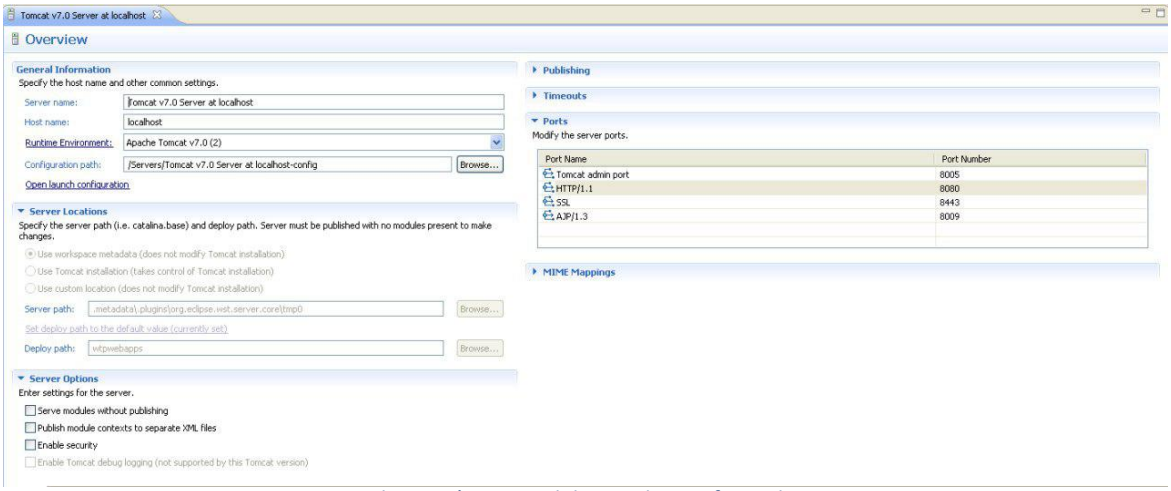

Ilustración 6: Panel de mandos configurado

4. Configuración del proyecto

Por último, debemos configurar nuestro proyecto para que soporte el protocolo HTTPS. Para ello, tenemos que acceder al fichero security.xml ubicado en la siguiente ruta Acme-Rendezvous > src > resources > spring > config > security.xml.

En este archivo tenemos que hacer un único cambio indicar que el canal requerido para realizar el login en el sistema es https.

```
<security:intercept-url pattern="/security/login.do" access="permitAll" requires-channel="https"/>
```

```
security.xml
24    <bean id="passwordEncoder"
25        class="org.springframework.security.authentication.encoding.Md5PasswordEncoder" />
26
27    <!-- Access control -->
28
29    <security:http auto-config="true" use-expressions="true">
30        <security:intercept-url pattern="/" access="permitAll" />
31
32        <security:intercept-url pattern="/favicon.ico" access="permitAll" />
33        <security:intercept-url pattern="/images/**" access="permitAll" />
34        <security:intercept-url pattern="/scripts/**" access="permitAll" />
35        <security:intercept-url pattern="/styles/**" access="permitAll" />
36        <security:intercept-url pattern="/profile/**" access="permitAll" />
37
38        <security:intercept-url pattern="/views/misc/index.jsp" access="permitAll" />
39
40        <security:intercept-url pattern="/security/login.do" access="permitAll" requires-channel="https"/>
41        <security:intercept-url pattern="/security/loginFailure.do" access="permitAll" />
42
43        <security:intercept-url pattern="/welcome/index.do" access="permitAll" />
44
45        <security:intercept-url pattern="/user-**" access="permitAll" />
46        <security:intercept-url pattern="/terms**" access="permitAll" />
47              <security:intercept-url pattern="/cookies**" access="permitAll" />
48        <security:intercept-url pattern="/register/**" access="isAnonymous()" />
49        <security:intercept-url pattern="/user/**" access="hasRole('USER')" />
50        <security:intercept-url pattern="/admin/**" access="hasRole('ADMIN')" />
51
52
53        <security:intercept-url pattern="/rsvp/qa.do" access="permitAll"/>
54        <security:intercept-url pattern="/rendezvous/**" access="permitAll"/>
55        <security:intercept-url pattern="/announcement/**" access="permitAll"/>
```

Ilustración 7:Security.xml configurado

Al realizar este paso ya nuestro entorno y nuestro proyecto estarán configurados para la utilización del protocolo HTTPS.

5. Ejemplo de uso

Para comprobar que nuestro Sistema de información se conecta mediante el protocolo HTTPS. Desplegaremos nuestro proyecto y accederemos a él mediante la ruta localhost:8080:/Acme-Rendezvous y comprobaremos como se nos abre el home de nuestro sistema. Pero al pulsar sobre el botón de Login se nos redirigirá a la ruta correcta que utiliza el puerto 8443.

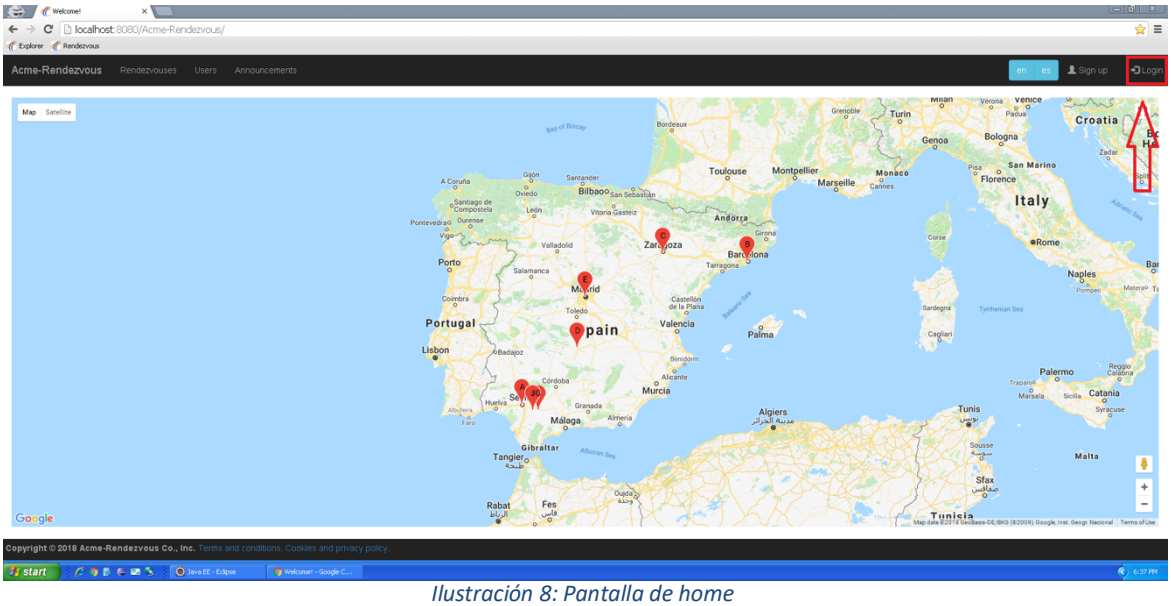

Ilustración 8: Pantalla de home

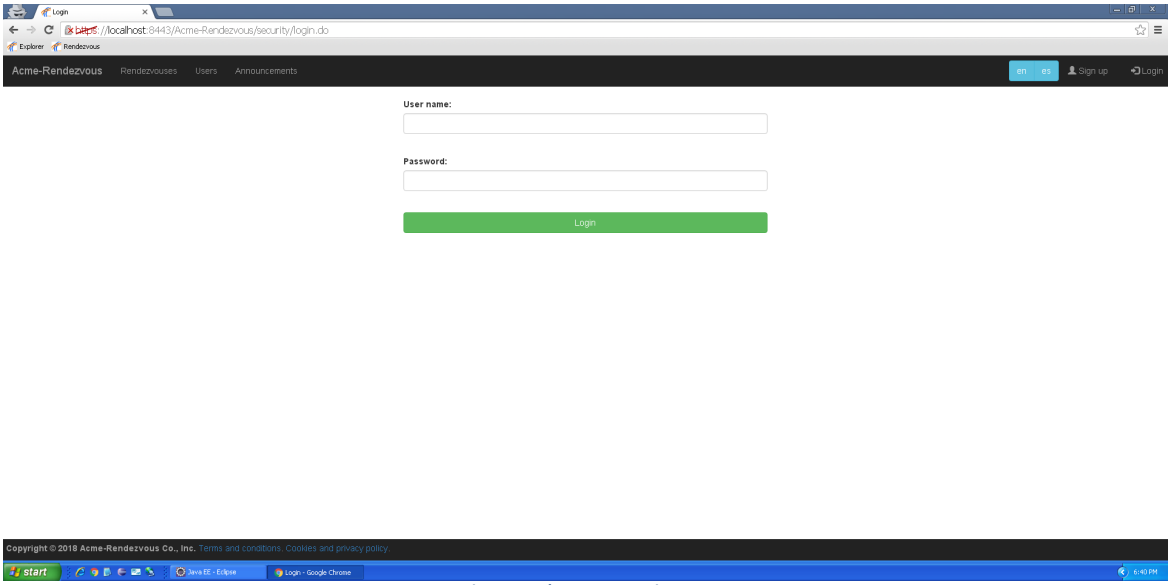

Ilustración 9: Login https

Como podemos ver en la imagen anterior ya utilizamos el protocolo HTTPS.

6. Problemas encontrados

Tras realizar este item recopilamos los problemas encontrados, en nuestro caso solo fue uno.

La primera vez que accedemos a la ventana de login nos muestra el siguiente error:

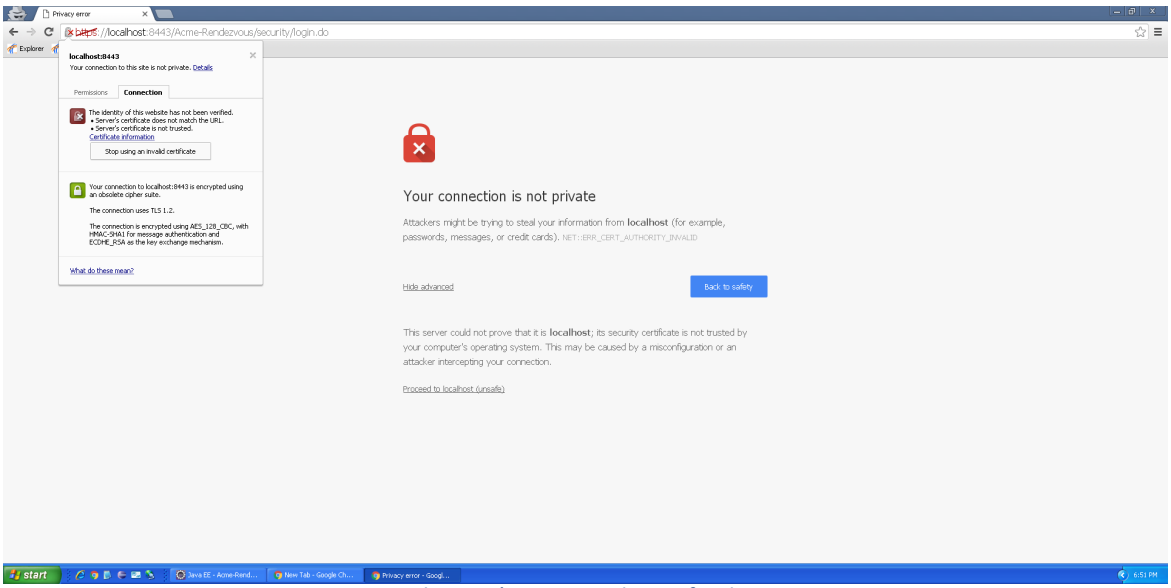

Ilustración 10: Error de certificado.

Este error es debido a que nuestro certificado no está firmado por ningun CA, Autoridad de Certificación.

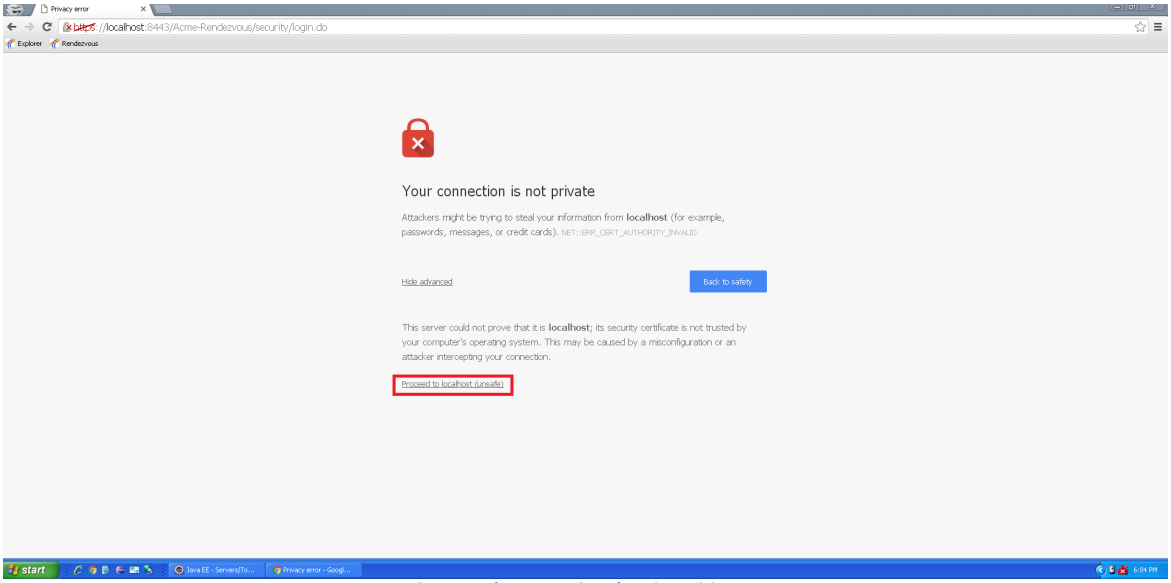

Ilustración 11: Solución al problema

Este problema se soluciona pulsando sobre el enlace “advanced” y luego el enlace “Proceed to Localhost(unsafe)”, como podemos observar en la anterior ilustración, de esta forma el problema quedará resuelto.

7. Conclusiones

Utilizar el protocolo HTTPS nos permite intercambiar información sensible de manera segura. Lo cual hoy en día es un aspecto fundamental en cualquier página web en la que se realicen registros de datos sensibles como números de cuentas bancarias para realizar pagos online o datos de carácter personal.