Secure communications report (HTTPS)

Este documento incluye toda la información relacionada con la implementación y despliegue de un proyecto con comunicaciones seguras haciendo uso del protocolo HTTPS. Para una explicación más clara y completa, se mostrarán los ejemplos implementándolos sobre el proyecto “Sample Project”, probándolos sobre el proyecto “Acme Six Pack” y configurando el servicio Tomcat sobre la máquina virtual “Pre-Production Configuration”.

Debido a la longitud del documento, este queda dividido en 5 secciones principales que se enumeran y enlazan a continuación:

---

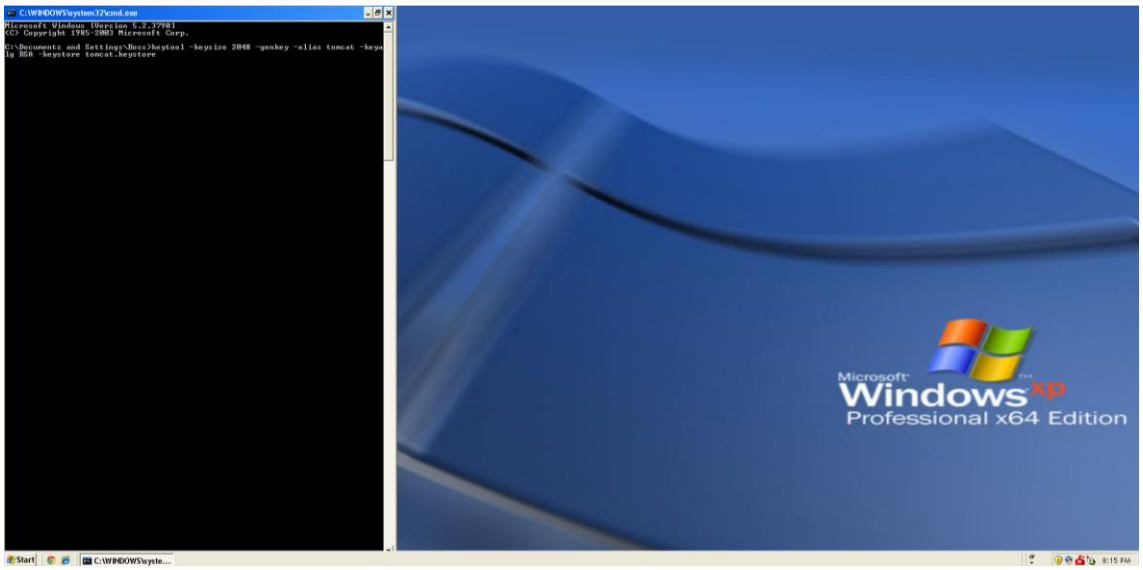
1. Getting a SSL certificate

Antes de empezar a configurar el servicio Tomcat o el proyecto hay que saber que necesitamos un certificado de seguridad valido. Para ello en este caso hemos obtenido un certificado SSL de uso temporal (3 meses desde su creación) que hemos adjuntado a la carpeta “Item 7” de la entrega, aunque a continuación detallamos cómo se puede obtener dicho certificado:

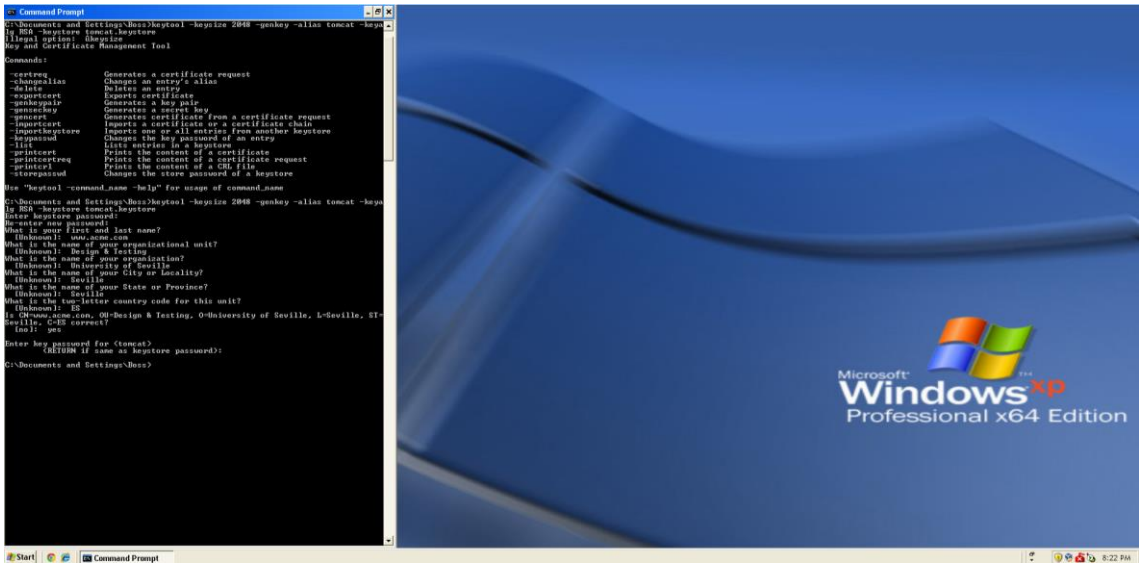
1.1. Primero accedemos a la consola de comandos del sistema, para ello en primer lugar presionamos el atajo de teclado Win + R. Aparecerá una ventana en la que escribiremos “cmd” y le daremos a “OK”.

1.2. Aparecerá la consola de comandos, en la que deberemos escribir el siguiente texto “keytool –keysize 2048 –genkey –alias tomcat –keyalg RSA –keystore tomcat.keystore” y presionar la tecla “Intro” del teclado (ejecutar el comando). NOTA:

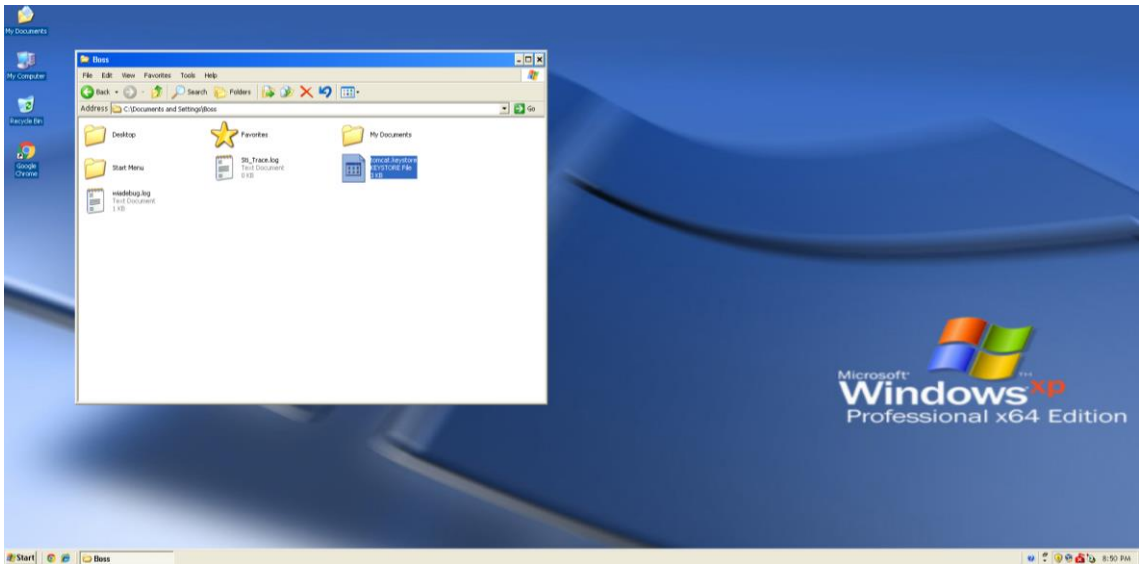
Para este paso evite copiar el texto directamente de este documento, la codificación del carácter “-” puede causar un error del tipo: “Illegal option: ûkeysize”.



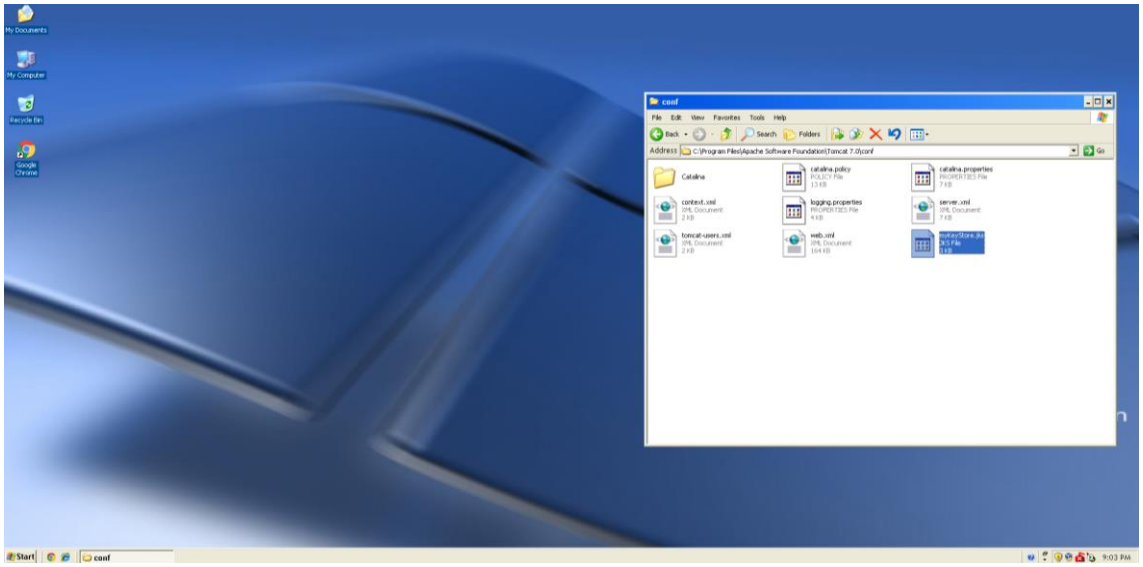
1.3. A partir de aquí se le preguntará por una serie de datos que deberá ir rellenando convenientemente y pulsando la tecla “Intro” una vez completado cada campo. Debe prestar especial atención en algunos de ellos, por lo que a continuación se detalla cómo debe rellenarlos correctamente:
    - “Enter keystore password”: Por defecto la contraseña es “changeit”.
    - “Re-enter keystore password”: De nuevo, por defecto al contraseña es “changeit”.
    - “What is your first and last name?”: En este caso, a pesar de por lo que puede parecer que se le pregunta, debe introducir la url principal de su dominio, en este caso “www.acme.com”. NOTA: Si hiciera falta usar subdominios alternativos debería especificarse explícitamente de la siguiente forma “*.acme.com”, aunque este no es el caso.
    - “What is the name of your organizational unit?”, “What is the name of your organization?”, “What is the name of your City or Locality?”, “What is the name of your State or Province?” y “What is the two-letter country code for this unit?”: Éstas preguntas deben responderse con total naturalidad, en nuestro caso introducimos los datos que se muestran en la captura de pantalla.
    - “Enter key password for <tomcat>”: Aquí debe introducir la misma contraseña que especificó anteriormente (pulsando la tecla “Intro” directamente, se introducirá de manera automática).



1.4. Una vez completado el proceso, ya puede cerrar la consola de comandos. Si todo ha ido bien, en la carpeta raíz del usuario con el que estamos identificados en el sistema debería aparecer el certificado que acabamos de obtener con el nombre "tomcat.keystore". Para comprobarlo, en nuestro caso accedemos a dicha carpeta haciendo clic en el icono "My Computer" del escritorio y navegando por las carpetas y directorios: "STORE (C:)" → "Documents and Settings" → "Boss".



1.5. El siguiente paso es cortar este archivo que se ha generado y llevarlo a la carpeta de configuración de nuestro servicio Tomcat. En nuestro caso se puede acceder, desde "My Computer", a través de las carpetas: "STORE (C:)" → "Program Files" → "Apache Software Foundation" → "Apache 7.0" → "conf". Una vez allí lo renombramos con el siguiente título: "myKeyStore.jks".

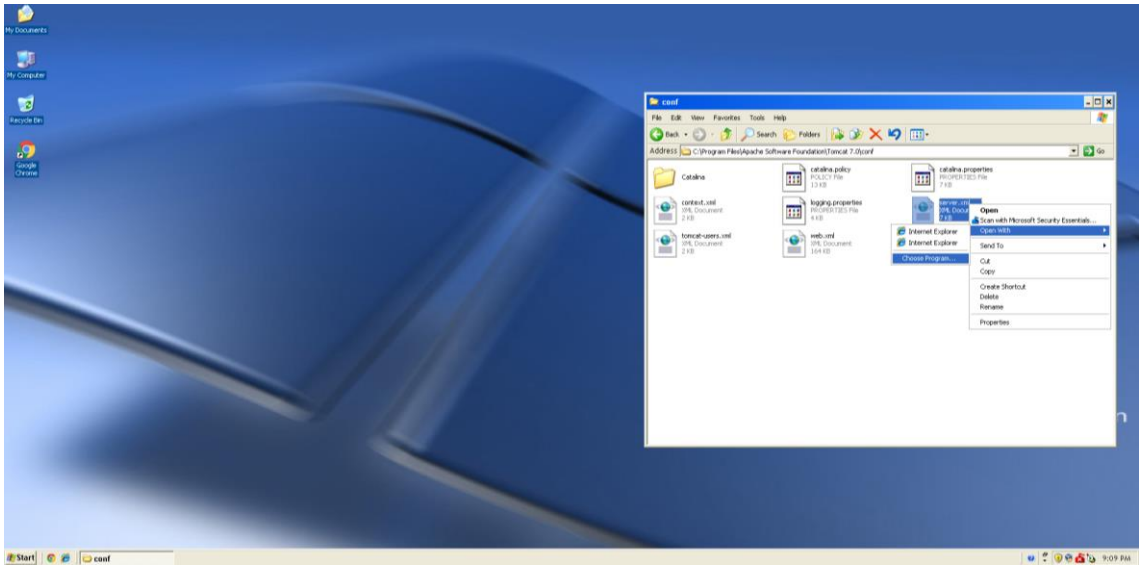


Ya tenemos creado nuestro certificado SSL válido.

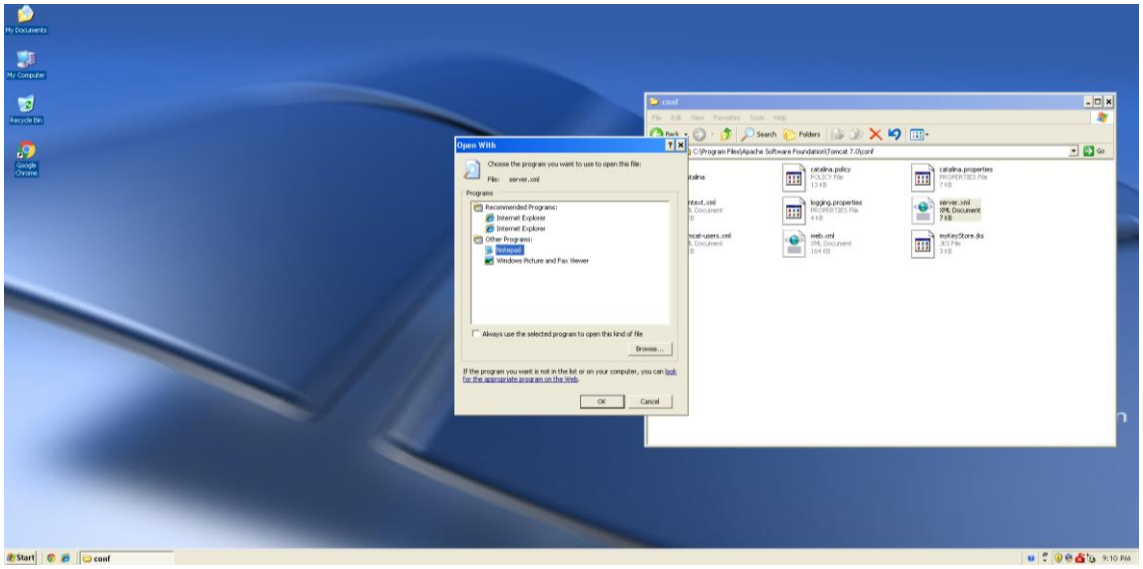
2. Configuring the Tomcat Service

Para poder hacer uso del protocolo HTTPS, y por lo tanto de comunicaciones seguras, debemos configurar también el servidor sobre el que irá montada la aplicación para que soporte dichas comunicaciones. Para ello, en la misma ventana donde nos encontrábamos anteriormente:

2.1. Abrimos el archivo “server.xml” con un editor de texto plano. En este caso hacemos clic derecho sobre el archivo → “Open With” → “Choose Program…”.



2.2. Seleccionamos “Notepad” y hacemos click en “OK”.



2.3. Una vez abierto el documento bajamos aproximadamente hasta la mitad del mismo, donde encontraremos el siguiente código comentado:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
    This connector uses the JSSE configuration, when using APR, the
    connector should be using the OpenSSL style configuration
    described in the APR documentation -->
```

Bajo el mismo hay otro fragmento de código comentado, el cuál deberemos des-comentar borrando los conjuntos de caracteres “<!--“ y “-->”.

```
<!--
  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
       maxThreads="150" scheme="https" secure="true"
       clientAuth="false" sslProtocol="TLS" />
  -->
```

Una vez des-comentado hay que hacerle una serie de modificaciones, de manera que quede tal y como se muestra a continuación:

```
<Connector port="443" protocol="HTTP/1.1" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75" enableLookups="false"
disableUploadTimeout="true" acceptCount="100" scheme="https" secure="true"
SSLEnabled="true" clientAuth="false" sslProtocol="TLS" keyAlias="tomcat"
keystoreFile="${catalina.home}/conf/myKeyStore.jks" keypass="changeit" />
```

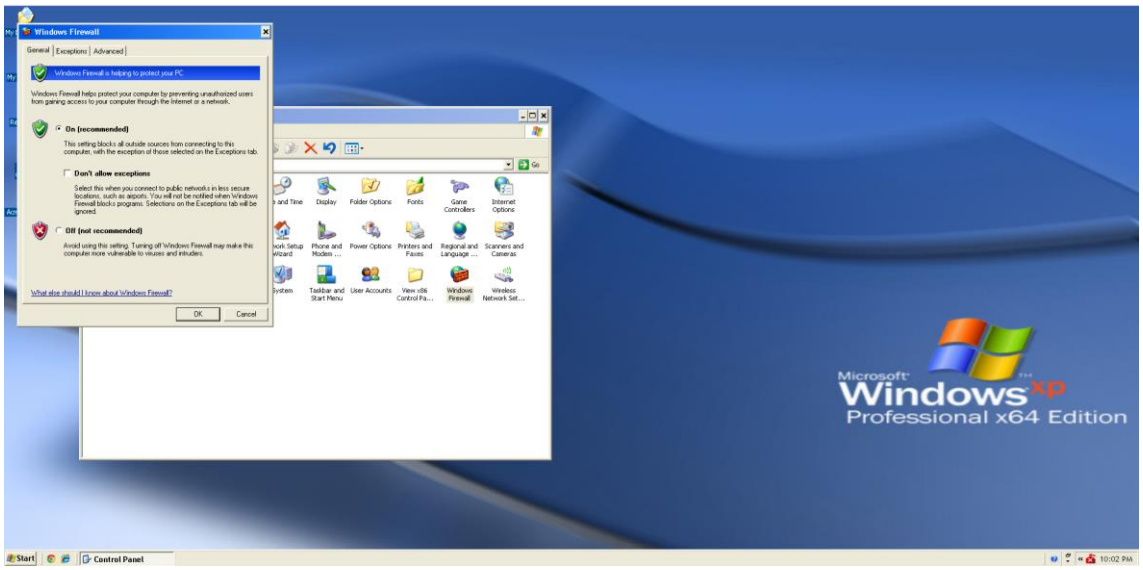
Nuestro servicio Tomcat ya está configurado para soportar conexiones seguras HTTPS.

NOTA: Este archivo de configuración, ya configurado convenientemente, también se proporciona en la carpeta “Item 7” del entregable.

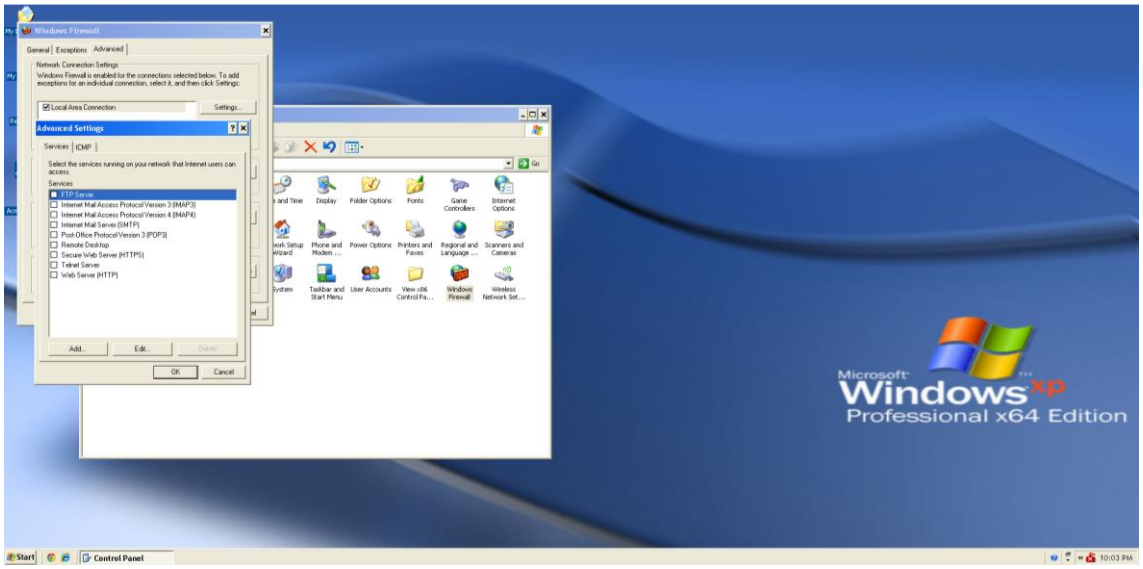
3. Configuring the Operating System

3.1. Ya configurado el servicio Tomcat, pasamos a permitir el paso de las conexiones seguras por nuestro sistema operativo. Para ello el primer paso es acceder al Panel de Control de Windows (“Control Panel”) desde el menú de Inicio (“START”).

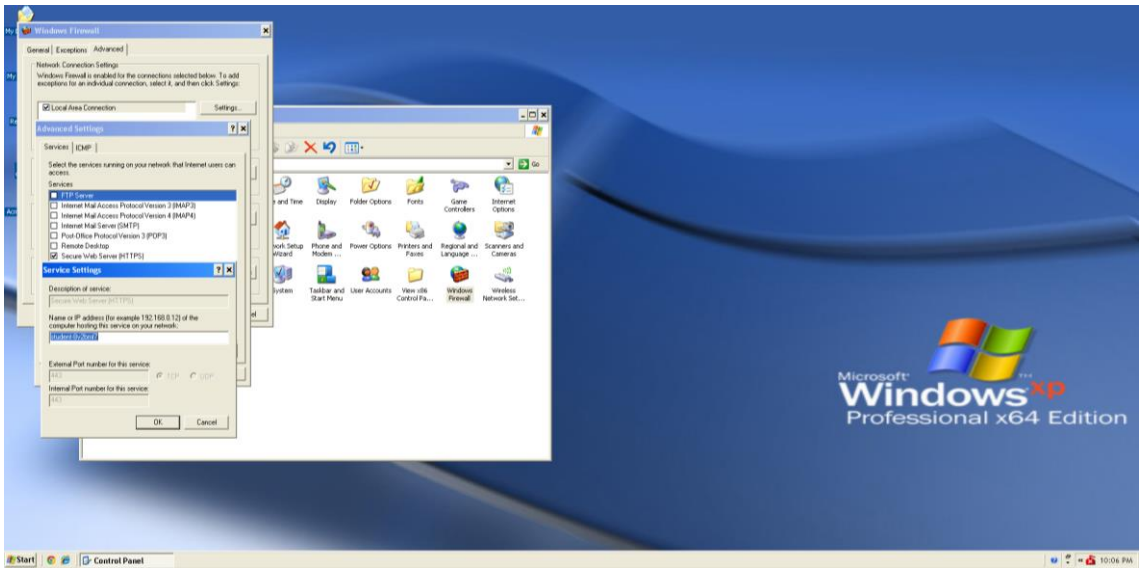
3.2. Una vez dentro seleccionamos la opción “Windows Firewall”.



3.3. Nos dirigimos a la pestaña “Advanced” y en la sección “Network Connection Settings” hacemos clic en el botón “Settings…”.



3.4. Marcamos la opción “Secure Web Server (HTTPS)” y se nos abrirá una ventana en la que deberemos presionar en “OK”.



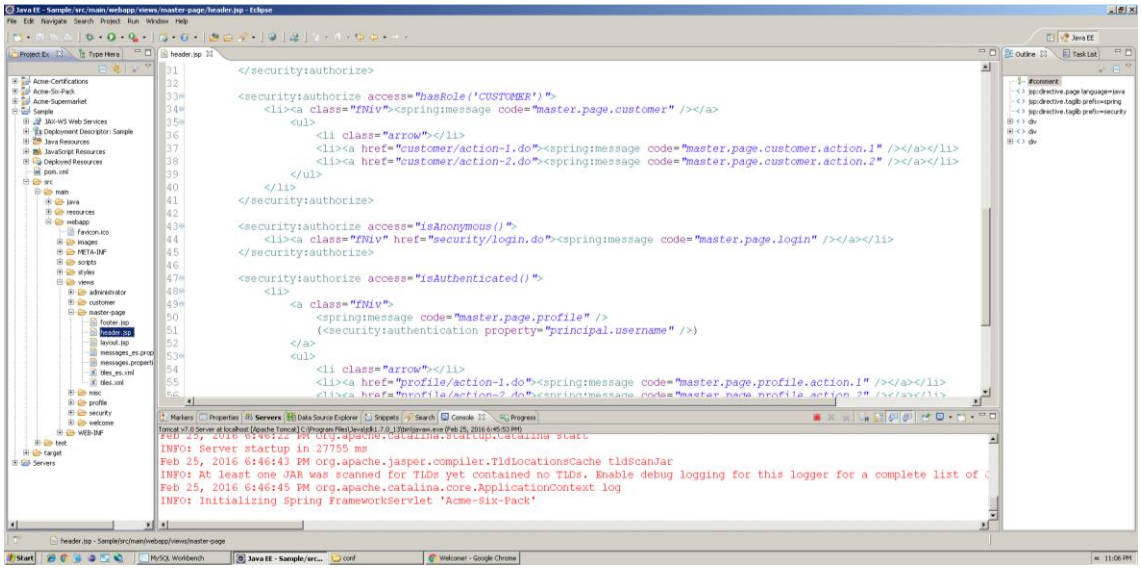
3.5. Ahora cerramos las dos anteriores ventanas de configuración haciendo clic en el botón “OK” de cada una y por último cerramos la ventana del Firewall de Windows.

Nuestro sistema operativo ya está configurado para soportar conexiones seguras HTTPS.

4. Configuring the Project

4.1. Una vez configurado todo el sistema de pre-producción pasamos a realizar algunas modificaciones a nuestro proyecto para que permita el uso de estas conexiones seguras en las condiciones que se nos piden, esto es, que se activen en el momento en el que un usuario pretenda identificarse en el sistema y que se desactiven cuando este cierre su sesión o directamente no tenga una sesión iniciada.

Para ello, en primer lugar, vamos al archivo “header.jsp” situado en la ruta “src” → “main” → “webapp” → “views” → “master-page” del proyecto.



4.2. Una vez dentro tenemos que modificar el enlace de Login para usuarios no identificados en el sistema. Antes de modificarlo tiene el siguiente aspecto:

```
<li><a class="fNiv" href="security/login.do"><spring:message code="master.page.login"
/></a></li>
```

Una vez modificado debe quedar de la siguiente manera:

```
<li> <script>document.write('<a class="fNiv" href="https://' + window.location.hostname +
':443/security/login.do" >');</script> <spring:message code="master.page.login"
/></a></li>
```

4.3. Ahora procedemos a cambiar el enlace para cerrar sesión en el sistema (Logout). Antes del cambio su aspecto era el siguiente:

```
<li><a href="j_spring_security_logout"><spring:message code="master.page.logout" />
</a></li>
```

Tras el cambio, debe quedar tal y como se muestra a continuación:

```
<li><b> <script>document.write('<a class="fNiv" href="http://' +
window.location.hostname + ':80/j_spring_security_logout" >');</script> <spring:message
code="master.page.logout" /></a></b></li>
```

4.4. Es importante destacar que, a pesar de que en el “Sample Project 1.4” no hay más enlaces para usuarios no autenticados que el enlace de inicio de sesión (Login) comentado anteriormente, si hubiera otros enlaces deberían sufrir todos la misma modificación que se describe a continuación.

Si el enlace tuviese el siguiente aspecto:

```
<li><a class="fNiv" href="/entity/action.do"><spring:message
code="master.page.entityAction" /></a></li>
```

Debería quedar implementado de la siguiente forma:

```
<li> <script>document.write('<a class="fNiv" href="http://' + window.location.hostname +
':80/entity/action.do" >');</script> <spring:message code="master.page.entityAction"
/></a></li>
```

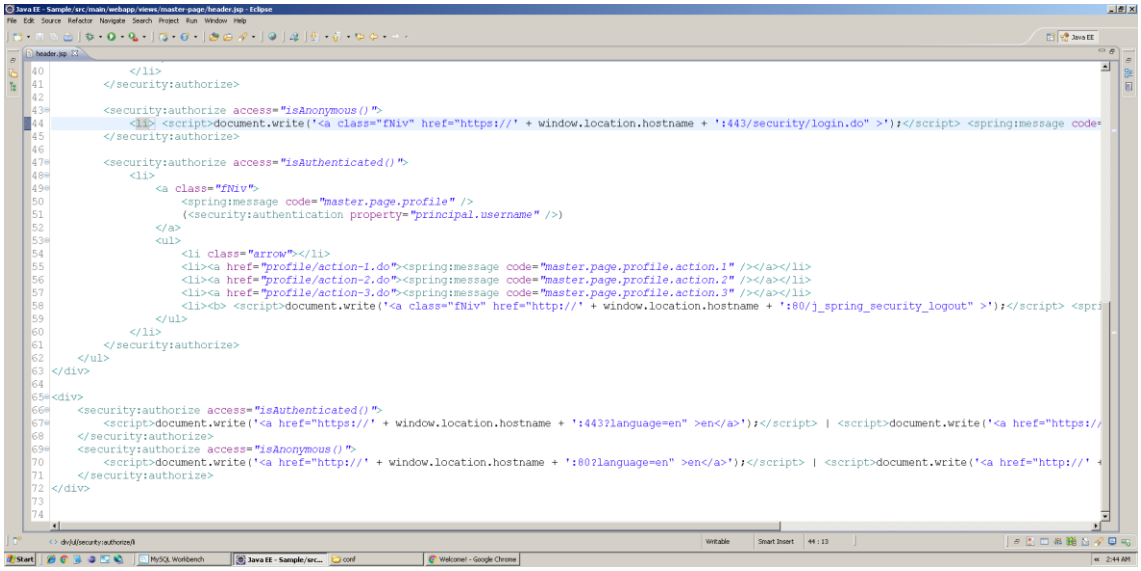
4.5. Además, también es importante añadir que todos aquellos elementos comunes tanto para usuarios autenticados como no autenticados, como pueden ser los enlaces de internacionalización, el enlace a la información de las cookies o el enlace del logo de la aplicación que hemos puesto en nuestro caso, deben ser mostrados personalizados según este hecho y siguiendo el patrón descrito para los elementos anteriores. Poniendo como ejemplo los enlaces de internacionalización, así es como estaban en un principio:

```
<a href="?language=en">en</a> | <a href="?language=es">es</a>
```

Y así es como debería quedar tras sufrir los cambios oportunos:

```
<security:authorize access="isAuthenticated()">

    <script>document.write('<a href="https://' + window.location.hostname +
    ':443?language=en" >en</a>');</script> | <script>document.write('<a
    href="https://' + window.location.hostname + ':443?language=es"
    >es</a>');</script>

</security:authorize>

<security:authorize access="isAnonymous()">

    <script>document.write('<a href="http://' + window.location.hostname +
    ':80?language=en" >en</a>');</script> | <script>document.write('<a href="http://'
    + window.location.hostname + ':80?language=es" >es</a>');</script>

</security:authorize>
```

4.6. Así pues, nuestro archivo “header.jsp” del “Sample Project 1.4” debería quedar de la siguiente manera:



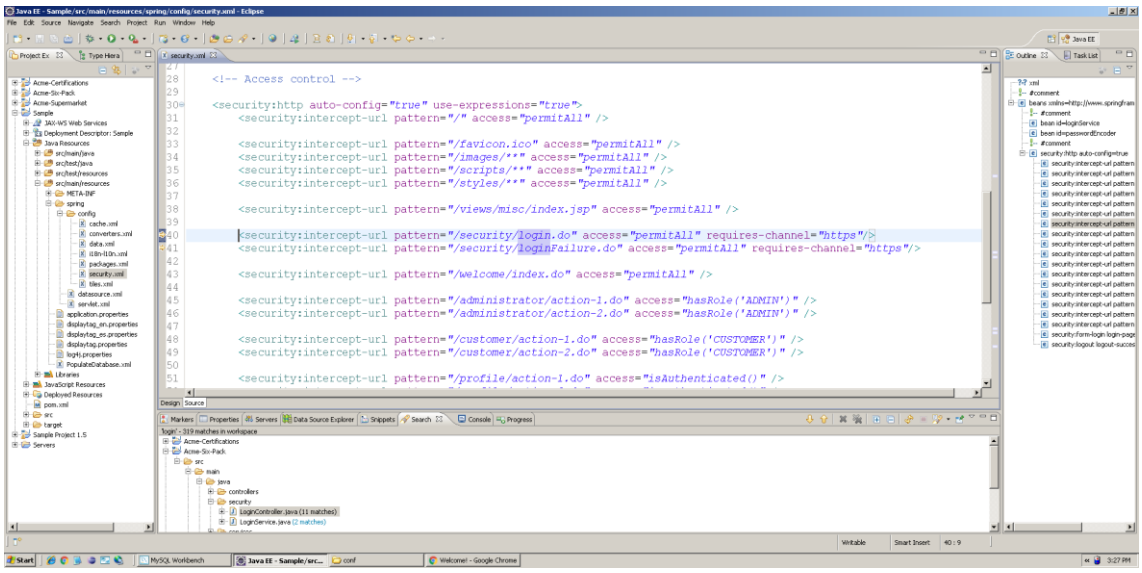
4.7. Pasamos a hacer un par de últimas modificaciones sobre nuestro proyecto, en este caso en el archivo “security.xml” situado en la ruta: “Java Resources” → “src/main/resources” → “spring” → “config”. Aproximadamente a la mitad de este archivo, en la zona “Access control”, encontraremos las siguiente líneas de código:

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

A ambas hay que añadirles la siguiente propiedad: “requires-channel="https"”, de modo que el resultado sea:

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

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

El archivo debería tener el siguiente aspecto:



El proyecto ya está configurado para permitir el uso de conexiones seguras en las condiciones indicadas.

5. Checking the Project

5.2. Una vez realizadas todas las configuraciones y cambios necesarios para que sea posible ejecutar nuestro proyecto con conexiones seguras, pasamos a poner nuestra aplicación en marcha y comprobar que todo funciona correctamente.

Para ello debemos realizar los pasos habituales a la hora de ejecutar nuestro proyecto en el entorno de pre-producción, como son:

En la MV de desarrollo:

-Crear el script de creación de la base de datos MySql.

-Crear el archivo .war de nuestro proyecto.

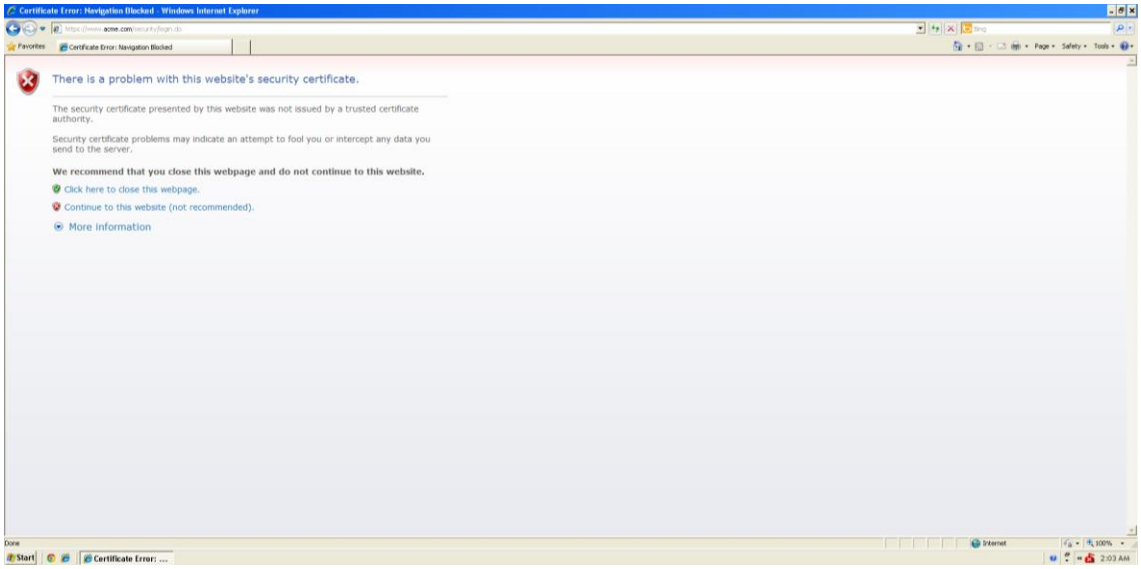
En la MV de pre-producción:

-Ejecutar el script de creación de la base de datos MySql.

-Replegar todas las aplicaciones que permitan ser replegadas en el servidor Tomcat.

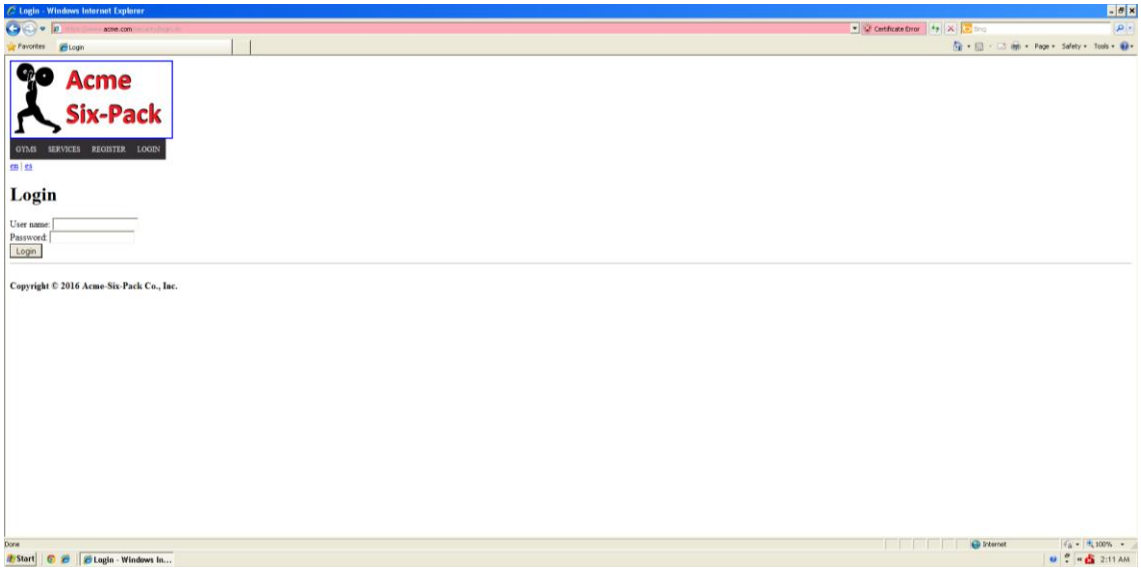
-Desplegar nuestra aplicación a partir del archivo .war previamente generado.

5.3. Una vez desplegada correctamente, abrimos el navegador “Internet Explorer” y accedemos a la url de nuestra aplicación: “www.acme.com”. Se debe mostrar la aplicación sin ningún inconveniente y podremos navegar con total normalidad. Sin embargo, al acceder a la pantalla de Login (donde se requiere una conexión segura) se mostrará un error como el siguiente:

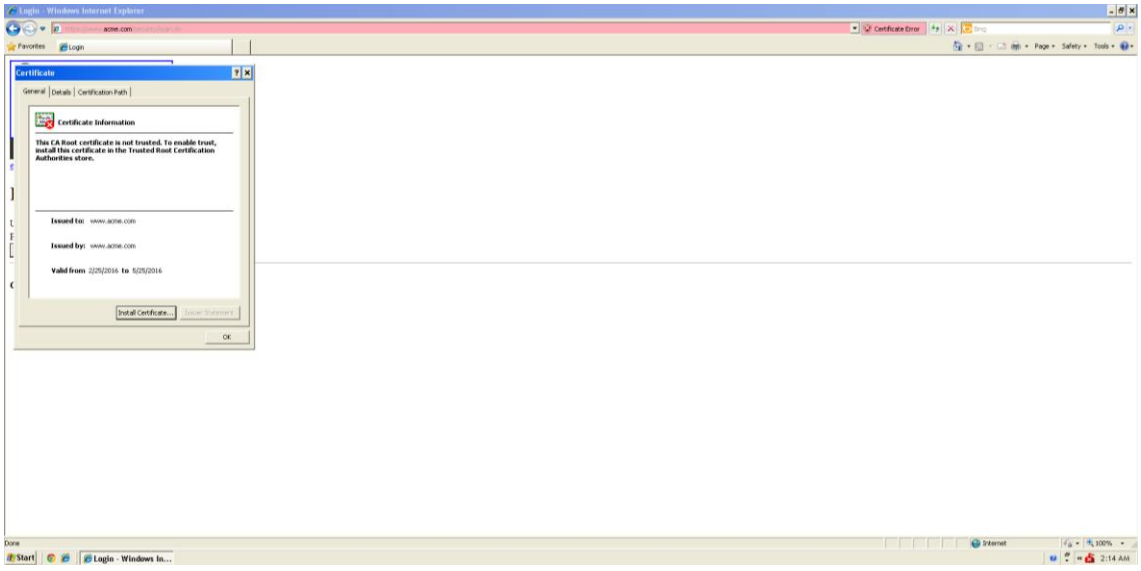


Esto es debido a que nuestro certificado no es “de confianza” para nuestro ordenador, puesto que no ha sido validado expresamente por ninguna empresa y/u organización. Ante este inconveniente, y la imposibilidad de poder comprar un certificado de confianza para la realización de esta entrega, hay una sencilla solución que se debe realizar una única vez y este error no volverá a molestar en el ordenador donde se lleve a cabo.

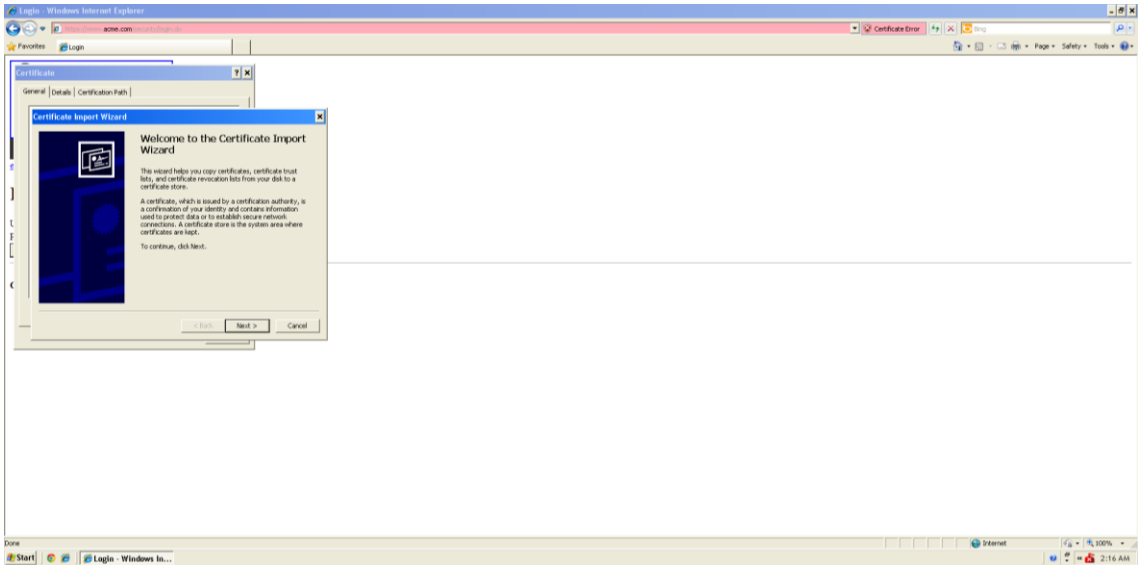
5.4. Haremos clic en el enlace “Continue to this website (not recommended)”. Veremos cómo nos lleva a nuestra aplicación, aunque la url aparece en rojo y a la derecha aparece un mensaje informando sobre un error de certificado (“Certificate Error”).



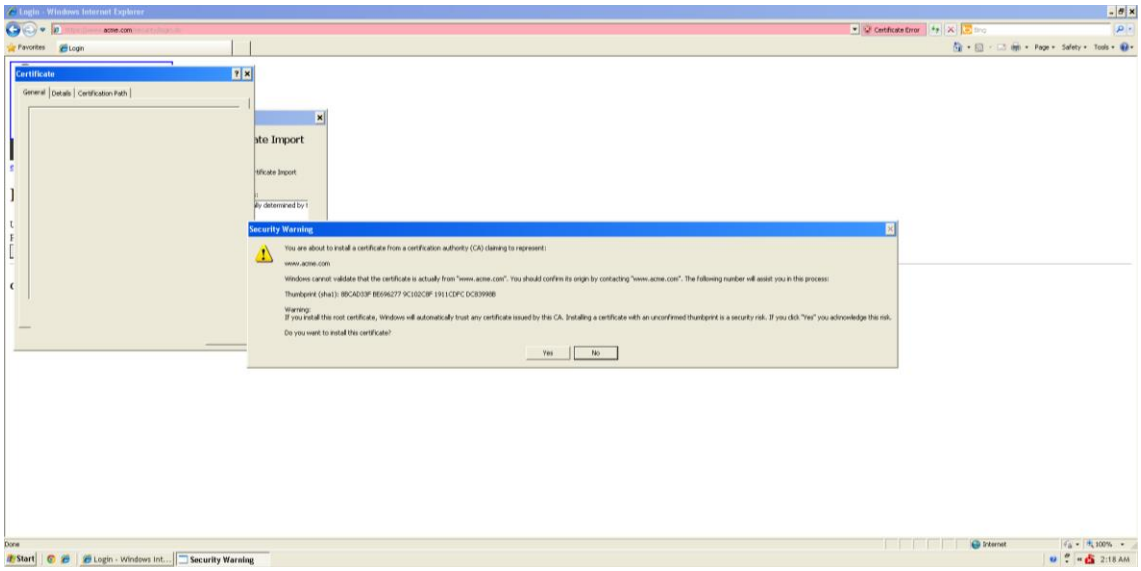
5.5. Hacemos clic en dicho mensaje de error y a continuación en el enlace “View certificates”, con lo que se nos abrirá una ventana como la siguiente:



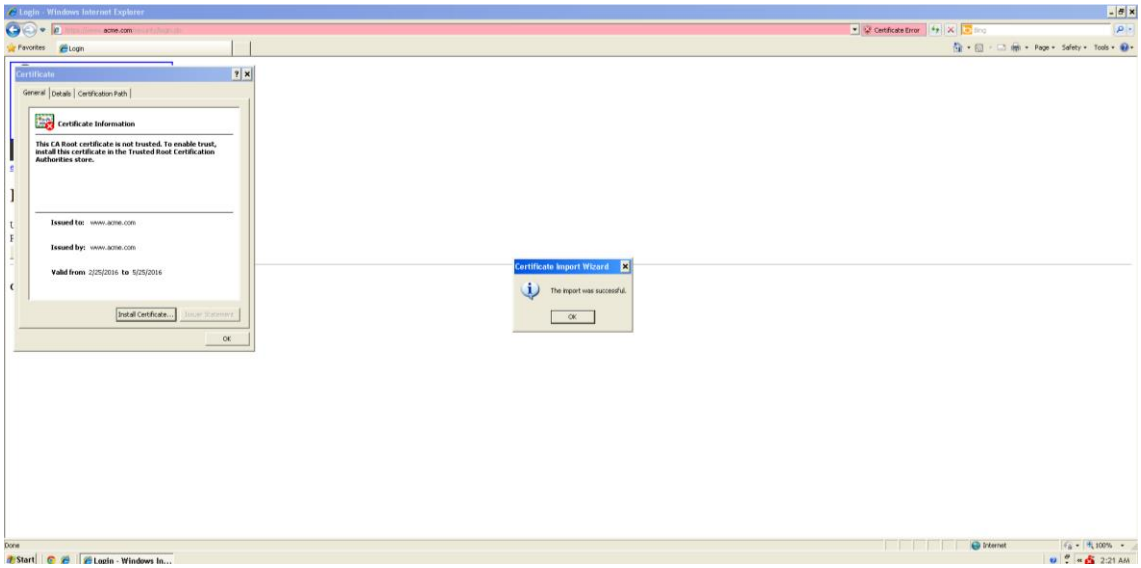
5.6. En esta ventana se nos informa del error comentado anteriormente y cómo solucionarlo, con lo que el siguiente paso sería hacerlo clic en el botón “Install Certificate…”.



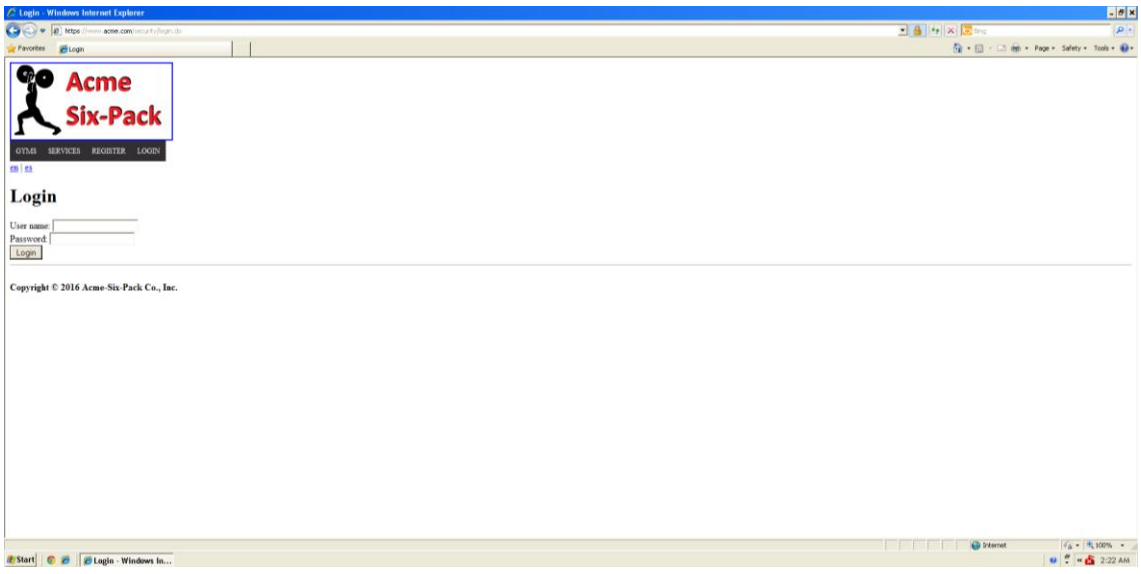
5.7. Aparecerá un instalador que le guiará durante los siguientes pasos, en los que únicamente tiene que hacer clic en el botón “Next >” sin modificar ningún elemento y por último en el botón “Finish”.

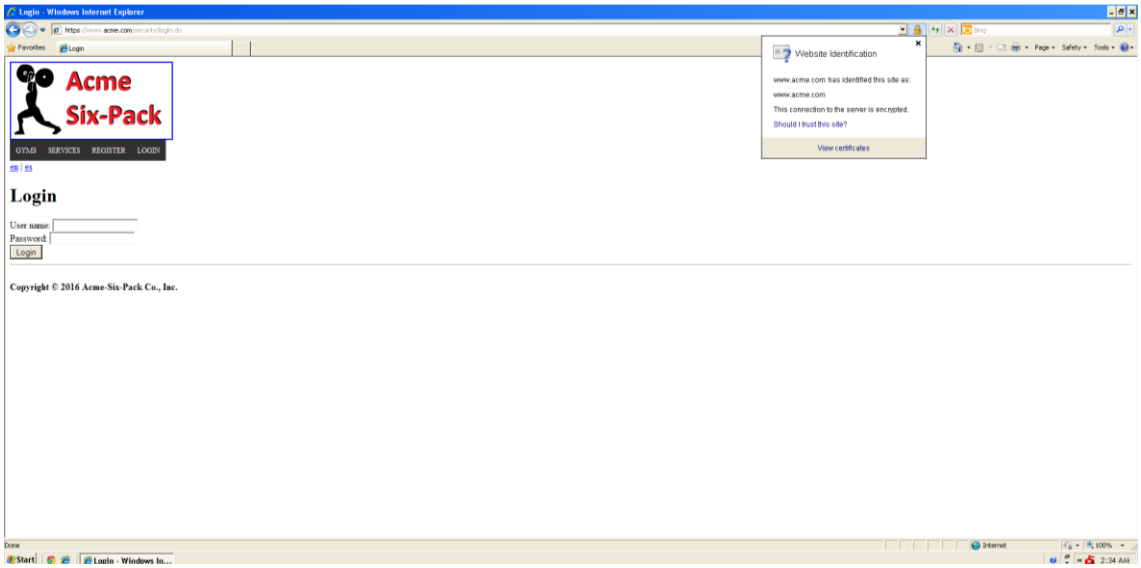


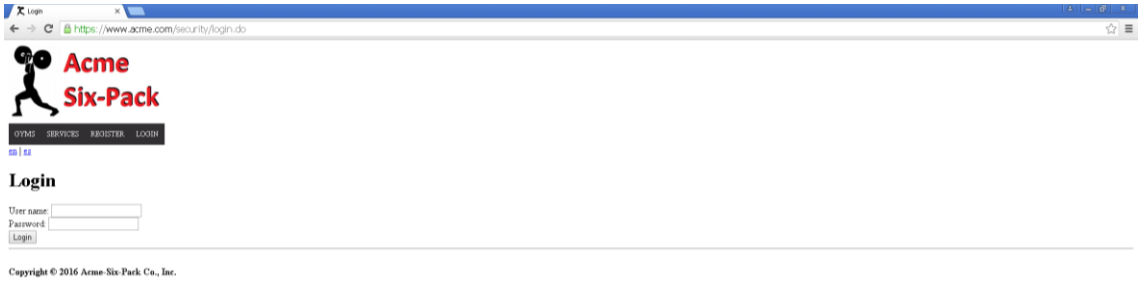
5.8. Como último paso vemos cómo se nos pide confirmación expresa de que el certificado pertenece a la web a la que intentamos acceder, puesto que Windows no puede comprobarlo automáticamente. Haciendo clic en “Yes” habremos instalado/importado el certificado correctamente y se nos informará de ello con una simple ventana emergente que cerraremos pulsando en “OK”.

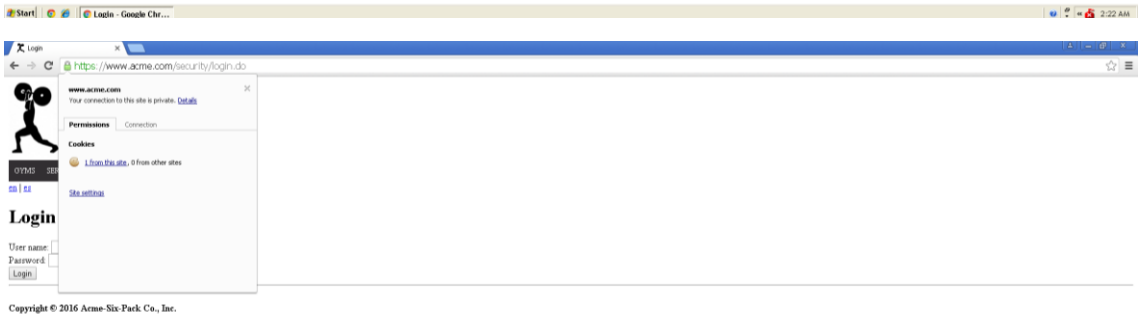


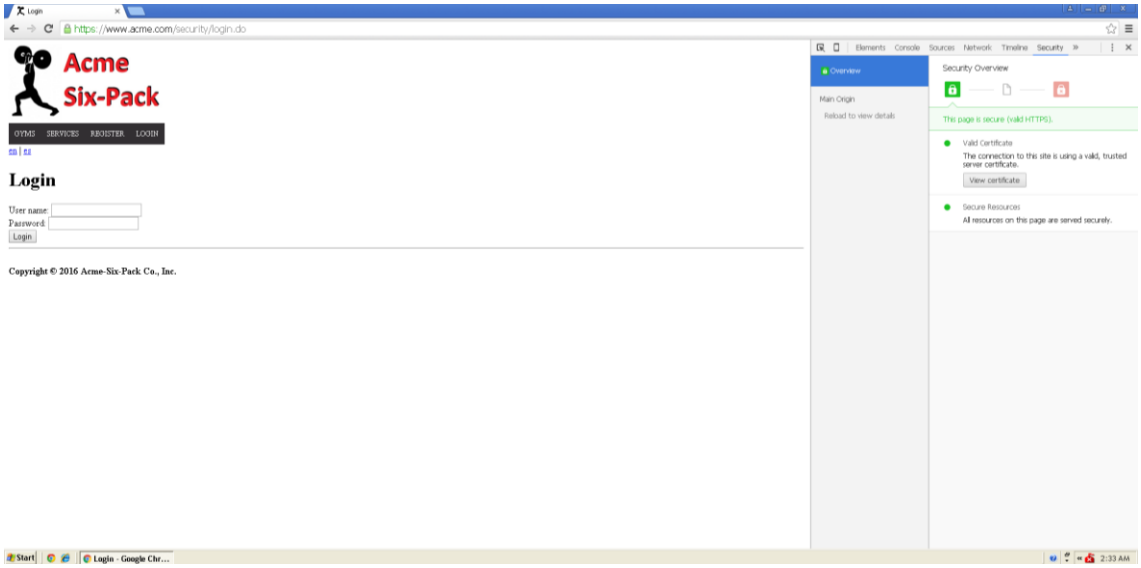
5.9. La próxima vez que accedamos a la aplicación desde cualquier navegador del ordenador podremos hacer un uso sin ningún tipo de problemas de la misma mediante conexiones seguras a través del protocolo HTTPS, se nos notificará de ello mediante un candado, habitualmente de color verde, en el que haciendo clic se nos confirmará mediante mensajes como “Your connection to this site is private”, “Valid Certificate: The connection to this site is using a valid, trusted server certificate.”, “Secure Resources: All resources on this page are served securely.” o “This page is secure (valid HTTPS).” en Google Chrome, como “The connection to the server is encrypted.” en Internet Explorer o como “The certificate is OK.” en la ventana para ver detalles del certificado donde lo instalamos previamente, concretamente en la pestaña “Certification Path”.

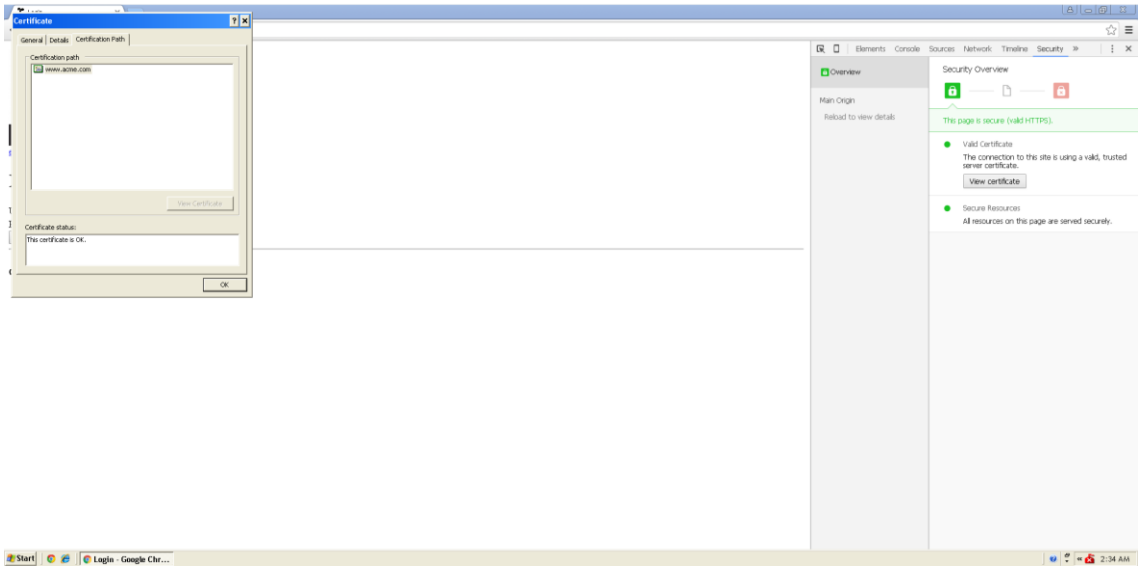



Login - Windows Internet Explorer
Website Identification
www.acme.com has identified this site as:
www.acme.com
This connection to the server is encrypted.
Should I trust this site?
View certificates
Acme
Six-Pack
GYMS SERVICES REGISTER LOGIN
Login
User name:
Password:
Login
Copyright © 2016 Acme-Six-Pack Co., Inc.


Login
https://www.acme.com/security/login.do
Acme
Six-Pack
GYMS SERVICES REGISTER LOGIN
Login
User name:
Password:
Login
Copyright © 2016 Acme-Six-Pack Co., Inc.


Login
https://www.acme.com/security/login.do
www.acme.com
Your connection to this site is private. Details
Permissions
Connection
Cookies
1 from this site, 0 from other sites
Site settings
Login
User name:
Password:
Login
Copyright © 2016 Acme-Six-Pack Co., Inc.





Esto es todo. De esta manera el proyecto estaría desplegado en el entorno de pre-producción y haciendo uso de las conexiones seguras (HTTPS) en las condiciones indicadas.