



Report Item 6

Grupo: 20

Miembro:

José Ángel Domínguez Espinaco

Daniel Lozano Portillo

José Joaquín Rodríguez Pérez

María Ruíz Gutiérrez

Miguel Ternero Algarín

Laura Vera Recacha

Contenido

DEVELOPERS CONFIGURATION3

PRE-PRODUCTION.....11

BIBLIOGRAFÍA13

DEVELOPERS CONFIGURATION

Nos creamos una carpeta en "C:\\" llamada key, que será donde generaremos nuestro almacén de claves.

Una vez creada abrimos una terminal de cmd como "Boss" y ejecutamos el siguiente comando:

keytool -genkey -alias tomcat -keyalg RSA -keystore "C:\key\keys"

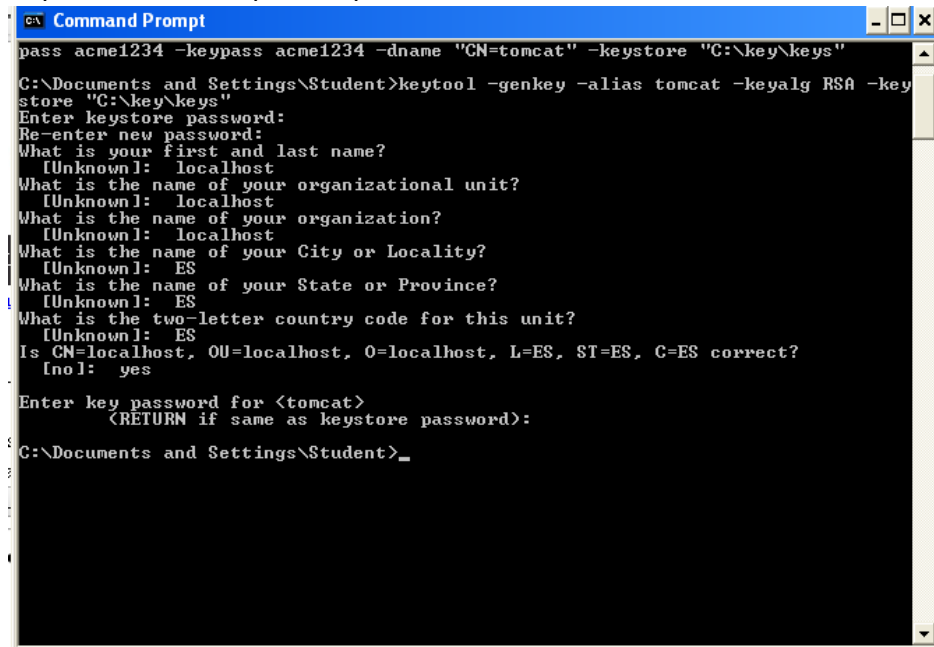
Algunos parámetros son:

-alias: alias con el que llamaremos a nuestro almacen de claves.

-keyalg: algoritmo de encriptación; RSA en este caso.

-keystore: donde se generará el almacén de claves. (Carpeta creada previamente).

Al pulsar intro nos pedirá que introduzcamos varios datos:



```
pass acme1234 -keypass acme1234 -dname "CN=tomcat" -keystore "C:\key\keys"
C:\Documents and Settings\Student>keytool -genkey -alias tomcat -keyalg RSA -key
store "C:\key\keys"
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: localhost
What is the name of your organizational unit?
[Unknown]: localhost
What is the name of your organization?
[Unknown]: localhost
What is the name of your City or Locality?
[Unknown]: ES
What is the name of your State or Province?
[Unknown]: ES
What is the two-letter country code for this unit?
[Unknown]: ES
Is CN=localhost, OU=localhost, O=localhost, L=ES, ST=ES, C=ES correct?
[no]: yes
Enter key password for <tomcat>
<RETURN if same as keystore password>:
C:\Documents and Settings\Student>
```

Establecemos una contraseña y la confirmamos escribiéndola de nuevo.

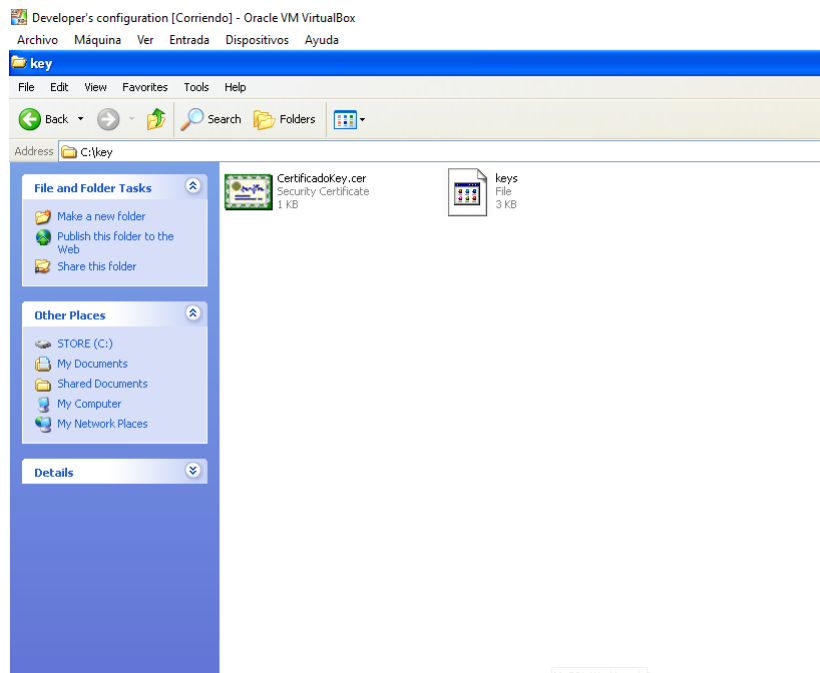
Nos pide un nombre y un apellido que no hace referencia a los nuestros, sino al nombre de nuestra página o el nombre del servidor donde se encuentra nuestra página.

Al ser un proyecto que estamos desarrollando en nuestra máquina pondremos localhost.

Los demás parámetros no son relevantes y escribimos localhost de nuevo.

Por último, ponemos nuestro código de país repetidas veces, escribimos yes y pulsamos intro. Cuando nos pida la contraseña de tomcat no escribimos nada y pulsamos enter.

Se genera nuestro almacén de claves:



Una vez creado nuestro almacén de claves vamos a configurar tomcat para que haga uso del protocolo https.

Buscamos el archivo server.xml y añadimos las siguientes líneas:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\key\keys" keystorePass="acme1234" />
```

Quedaría de la siguiente forma:



El último paso consiste en abrir el fichero security.xml de nuestro proyecto y añadir **requires-channel="https"** a aquellas urls que queremos que hagan uso del protocolo https. En nuestro caso se lo hemos añadido al login y al edit de un user quedando así:

```
*server.xml  security.xml
<security:http auto-config="true" use-expressions="true">
  <security:intercept-url pattern="/" access="permitAll" />

  <security:intercept-url pattern="/favicon.ico" access="permitAll" />
  <security:intercept-url pattern="/images/**" access="permitAll" />
  <security:intercept-url pattern="/scripts/**" access="permitAll" />
  <security:intercept-url pattern="/styles/**" access="permitAll" />
  <security:intercept-url pattern="/views/misc/index.jsp" access="permitAll" />
  <security:intercept-url pattern="/security/login.do" access="permitAll" requires-channel="https"/>
  <security:intercept-url pattern="/security/loginFailure.do" access="permitAll" />
  <security:intercept-url pattern="/welcome/index.do" access="permitAll" />

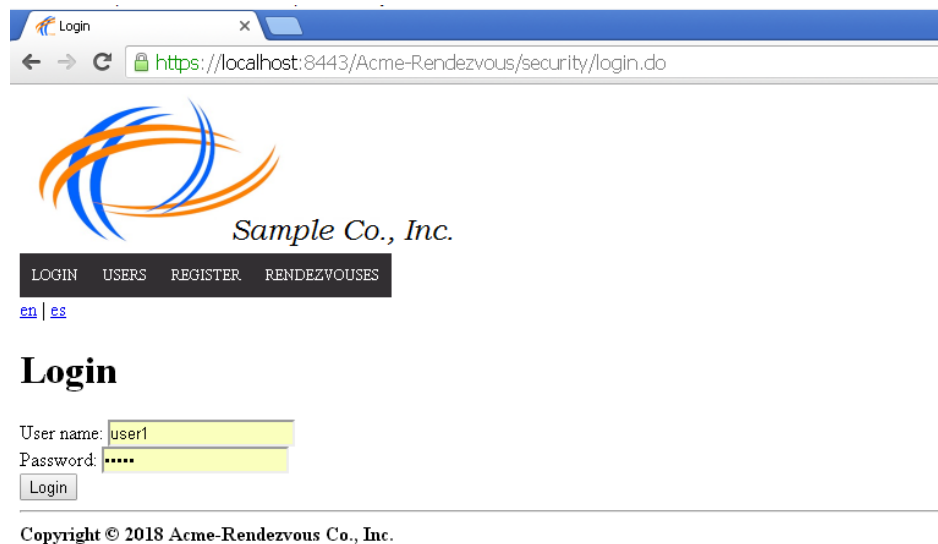
  <security:intercept-url pattern="/user/edit.do" access="permitAll" requires-channel="https"/>
  <security:intercept-url pattern="/user/**" access="permitAll" />

  <security:intercept-url pattern="/announcement/list.do" access="permitAll" />
  <security:intercept-url pattern="/announcement/user/**" access="hasRole('USER')" />
  <security:intercept-url pattern="/announcement/administrator/**" access="hasRole('ADMINISTRATOR')" />

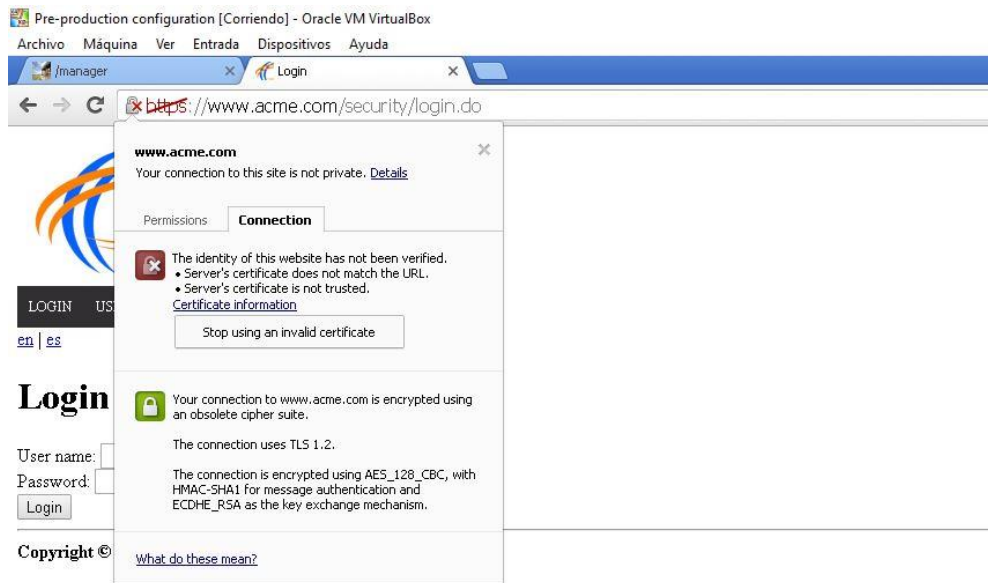
  <security:intercept-url pattern="/question/user/**" access="hasRole('USER')" />
  <security:intercept-url pattern="/rendezvous/user/**" access="hasRole('USER')" />
  <security:intercept-url pattern="/answer/user/**" access="hasRole('USER')" />
  <security:intercept-url pattern="/answer/**" access="permitAll" />
  <security:intercept-url pattern="/rendezvous/**" access="permitAll" />
  <security:intercept-url pattern="/question/**" access="permitAll" />
</security:http>
```

En caso de tener arrancado tomcat lo paramos y lo iniciamos de nuevo para cargar la nueva configuración.

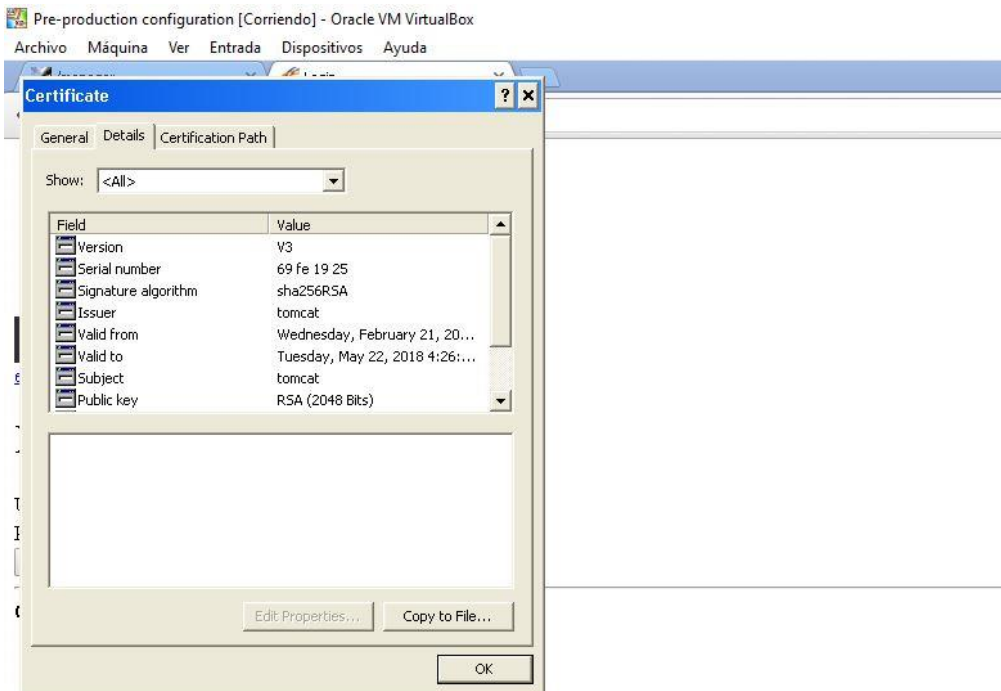
Podemos ver como al hacer login ya hacemos uso del protocolo https.



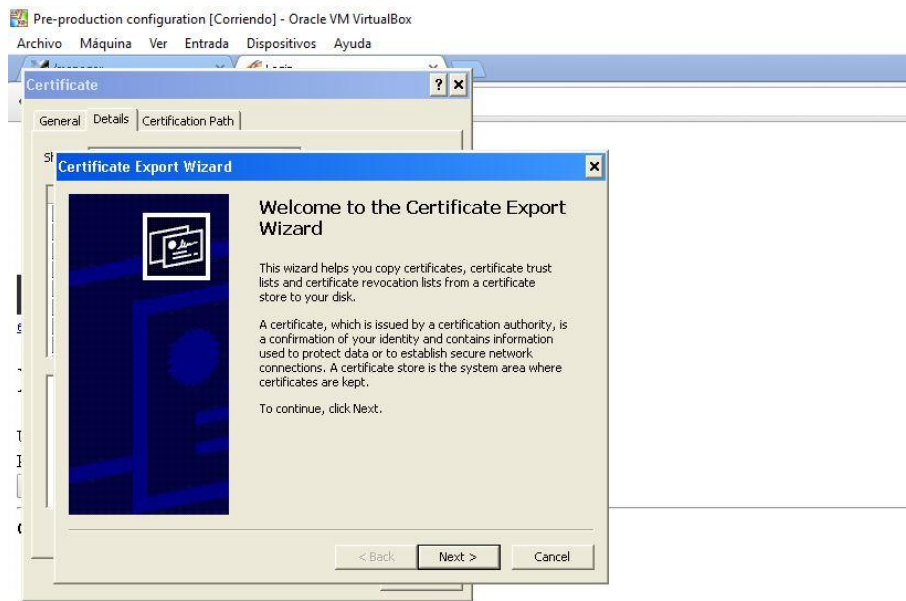
En caso de que nos diga que este sitio no es seguro tenemos que generar nuestro propio certificado de seguridad.
Pinchamos en “Certificate information



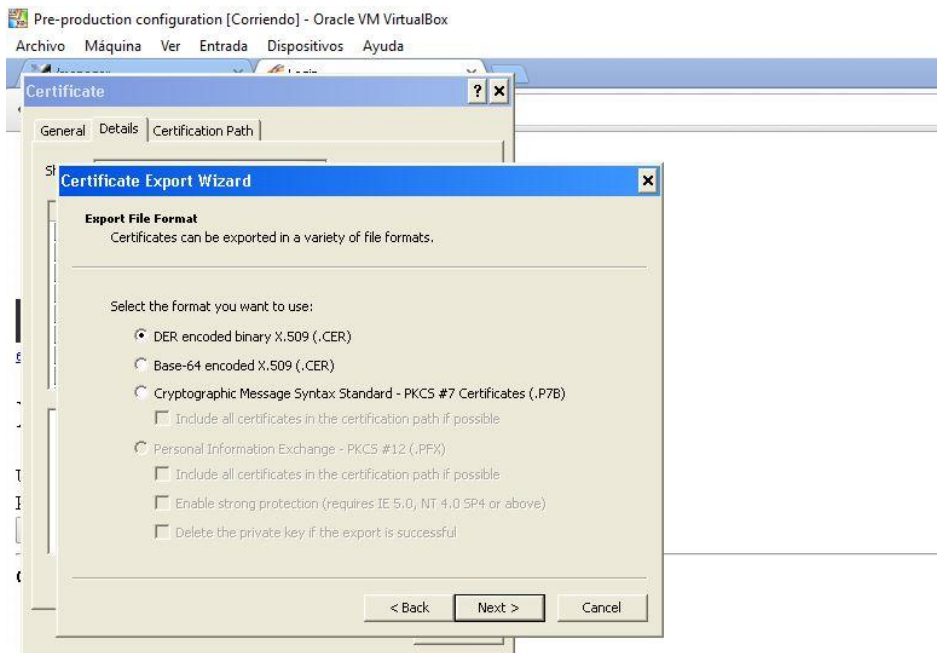
En la pestaña “Details” pinchamos en “Copy to File...”



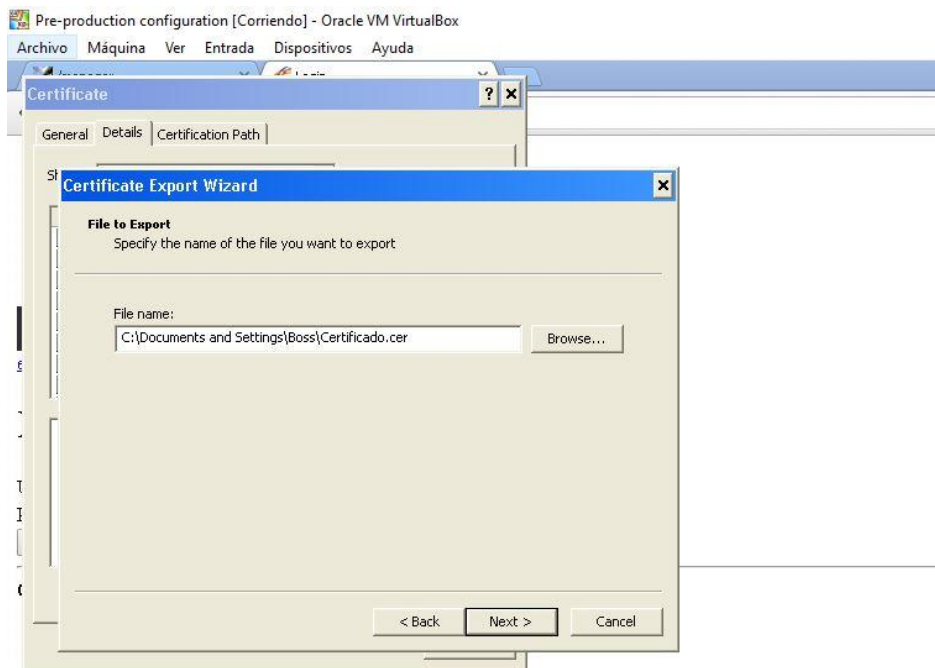
Le damos a “Next”.



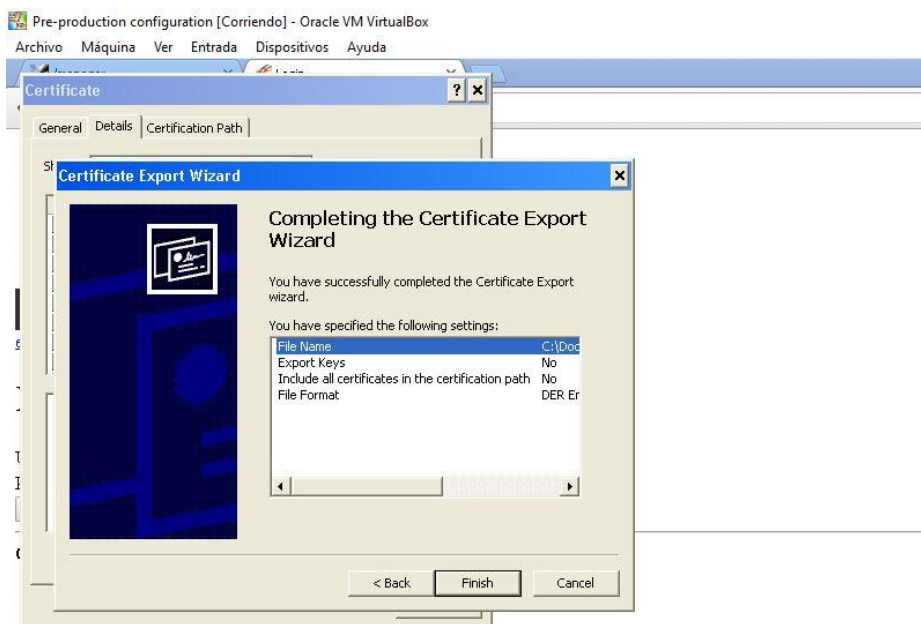
Seleccionamos la primera opción y “Next”.



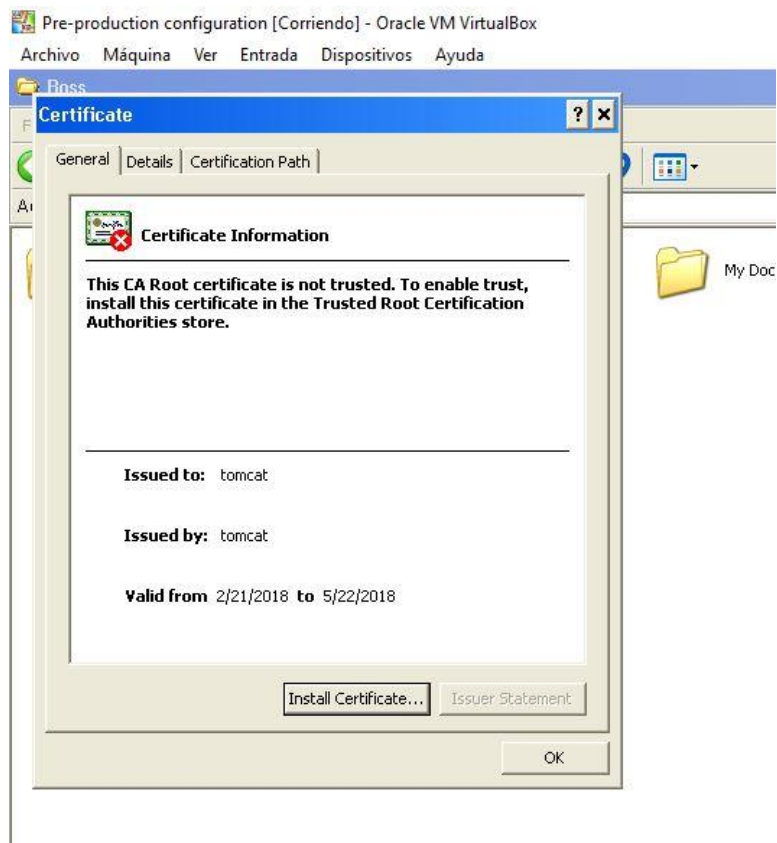
Seleccionamos la ruta donde queremos guardarlo y un nombre y pinchamos en “Next”.



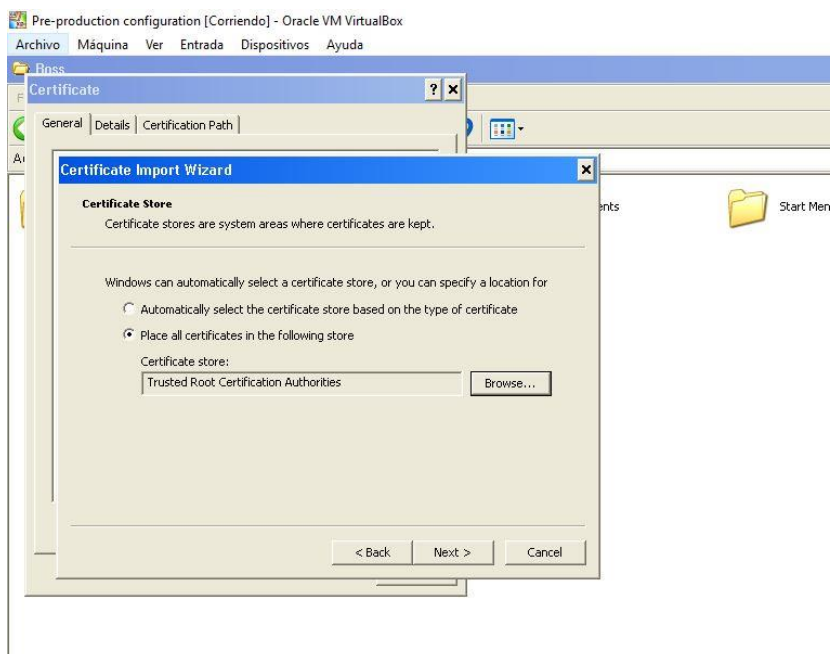
Por último, pinchamos en “Finish” y nos creará nuestro certificado.



El siguiente paso es instalarlo haciendo doble click en “Install Certificate”.

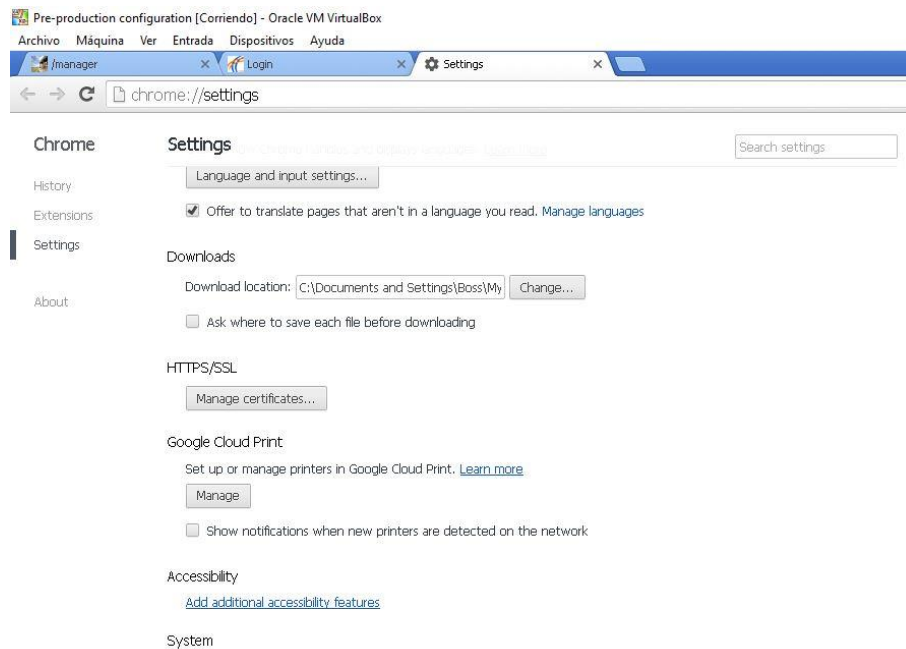


Seleccionamos la opción que aparece en la imagen y finalizamos su instalación.

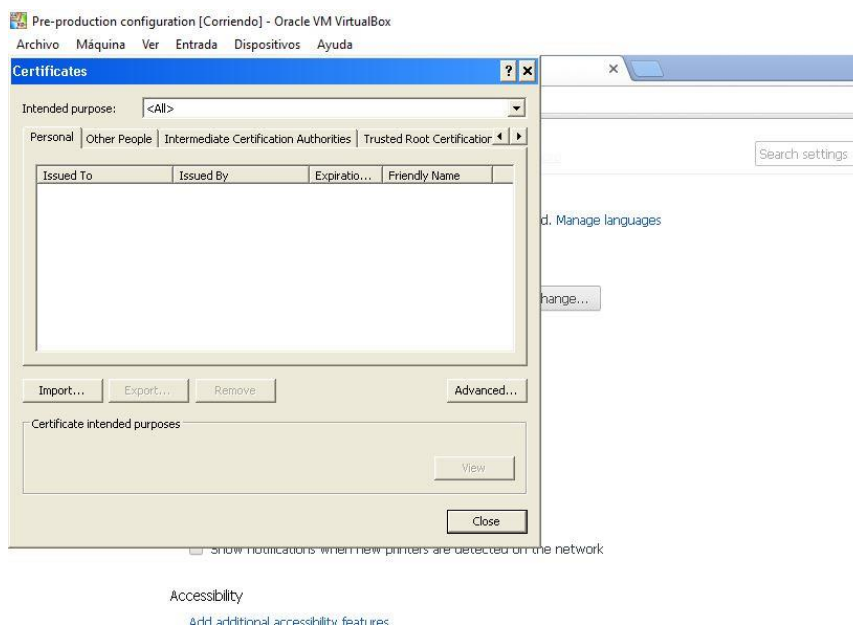


En caso de que nos siga haciendo referencia que el sitio no es seguro pasaremos a importarlo manualmente al navegador que usemos, en nuestro caso Chrome.

No dirigimos a Settings de Chrome y en el apartado HTTPS/SSL seleccionamos “Manage certificates”.



Nos dirigimos a la pestaña Trusted Root Certifications e importamos nuestro certificado que previamente creamos.



PRE-PRODUCTION

Para usar el protocolo HTTPS en este entorno tenemos que generar el war como hacíamos en la lección del despliegue una vez que hemos guardado las páginas que queremos que usen dicho protocolo.

Antes de desplegarlo repetiremos los pasos que hicimos para el entorno de developer a diferencia que ahora el archivo server.xml se encuentra en:

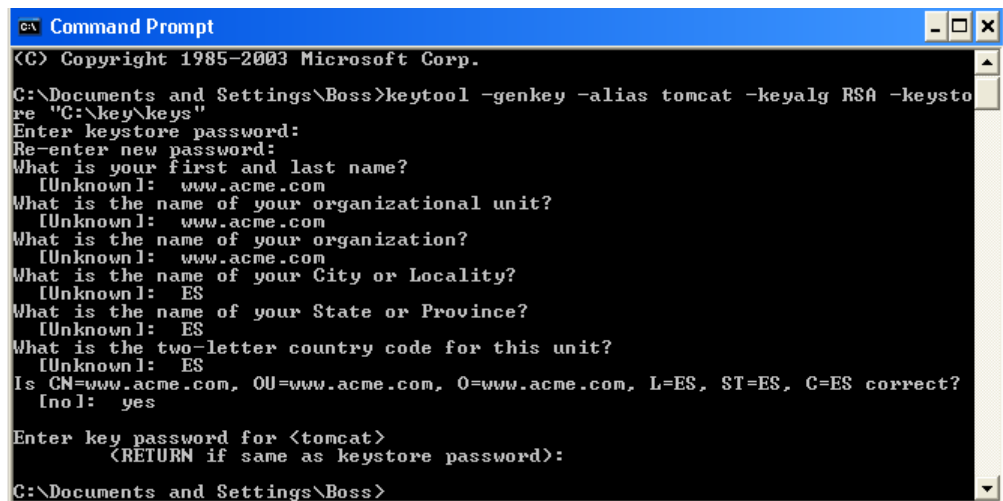
C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf

Primero usamos el comando para crear nuestro almacén de claves:

keytool -genkey -alias tomcat -keyalg RSA -keystore "C:\key\keys"

A diferencia del entorno developer ahora no escribimos localhost, sino www.acme.com que será nuestro dominio. Las iniciales de nuestro país se dejan igual que en la imagen.

Como contraseña establecemos: acme1234



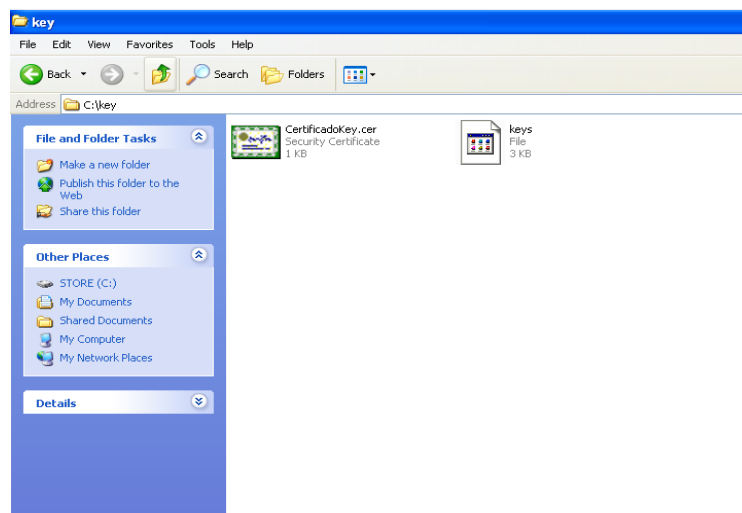
```
C:\> Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Boss>keytool -genkey -alias tomcat -keyalg RSA -keystore "C:\key\keys"
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: www.acme.com
What is the name of your organizational unit?
[Unknown]: www.acme.com
What is the name of your organization?
[Unknown]: www.acme.com
What is the name of your City or Locality?
[Unknown]: ES
What is the name of your State or Province?
[Unknown]: ES
What is the two-letter country code for this unit?
[Unknown]: ES
Is CN=www.acme.com, OU=www.acme.com, O=www.acme.com, L=ES, ST=ES, C=ES correct?
[no]: yes

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

C:\Documents and Settings\Boss>
```

Almacén de clave generado:



Ahora pasamos a configurar el archivo server.xml que se encuentra en la ruta especificada anteriormente.

Cabe destacar que cambiaremos el puerto que usa el protocolo HTTPS por el 443 ya que el 8443 nos ha dado muchos problemas. También hay que cambiarlo en el conector que usa el puerto 80 en el atributo redirectPort para que haga una correcta redirección.

El archivo server.xml quedaría de la siguiente forma añadiéndole:

```
<Connector port="443" protocol="HTTP/1.1"  
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="C:\key\keys" keystorePass="acme1234" />
```

```
<!-- A "Connector" represents an endpoint by which requests are received  
and responses are returned. Documentation at :  
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)  
Java AJP Connector: /docs/config/ajp.html  
APR (HTTP/AJP) Connector: /docs/apr.html  
Define a non-SSL HTTP/1.1 Connector on port 80  
-->  
<Connector port="80" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="443" />  
  
<Connector port="443" protocol="HTTP/1.1"  
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="C:\key\keys" keystorePass="acme1234" />  
<!-- A "Connector" using the shared thread pool-->  
<!--  
<Connector executor="tomcatThreadPool"  
port="80" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />  
-->
```

En caso de que tuviésemos iniciado Tomcat pasamos a pararlo e iniciarlo de nuevo. Desplegamos nuestro proyecto como ya se explicó en la lección del despliegue y accedemos a la url www.acme.com.

Podemos comprobar como ya hacemos uso del protocolo HTTPS a la hora de iniciar sesión. En caso de decirnos que el sitio no es seguro tendremos que generar nuestro certificado e instalarlo y además si nos sigue saliendo dicho problema importarlo en Chrome siguiendo los mismos pasos que para el entorno developer.

en | es

Login

User name:

Password:

Copyright © 2018 Acme-Rendezvous Co., Inc.

BIBLIOGRAFÍA

<https://www.youtube.com/watch?v=Ur6KzF-L3x8&t=338s&list=LLpVPSI7XhLFBdVSubmIEzBw&index=2>

<http://www.baeldung.com/spring-channel-security-https>

<https://www.digicert.com/es/instalar-certificado-ssl-tomcat.htm>

<https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>