1. Configurar SSL/TLS en Tomcat

Para instalar y configurar soporte de SSL/TLS en Tomcat son necesarios unos simples pasos.

- Crear un contenedor keystore usando java.
- Configurar Tomcat para usar el keystore.
- Configurar nuestra aplicación para que trabaje con SSL (accediendo a traves de https://localhost:8443/tuaplicacion).

1.1 Crear el contenedor keystore

Un Java key Store (JKS) es un repositorio de seguridad de certificados. El JDK provee de una herramienta llamada keytool para manipular el keystore.

Importante: Ejecutar CMD (command windows) **en modo administrador**.

La herramienta keytool la encontramos en cd %JAVA_HOME%/bin

Crear el keystore en formato PKCS12:

keytool -genkey -v -alias seguridadtomcat -keyalg RSA -keysize 2048 -validity 1000 -storetype pkcs12 -keystore fichero.pfx

Crear el keystore en formato jks:

keytool -genkey -v -alias seguridadtomcat -keyalg RSA -keysize 2048 -validity 1000 -storetype jks -keystore fichero.jks

NOTA: - Despues de –keystore debemos poner la ruta y el nombre de repositorio de certificados, terminado en la extensión correspondiente. .pfx o .jks

- Para la contraseña emplear abc123., o la que tu desees.

Con las sentencia anterior creamos un certificado denominado seguridadtomcat, o como tu quieras llamarlo, y a la vez un contenedor de certificados.

Si queremos, podemos realizar la exportación del certificado correspondiente con la siguiente instrucción:

keytool -exportcert -keystore C:\ficherodos.jks -alias seguridadtomcat -file C:\certificado.cer

El nombre del fichero (-file) También puede terminar en .crt

1.2 Configurar Tomcat para emplear el keystore

Vamos a la carpeta de instalación de Tomcat y abrimos la carpeta conf. Dentro de esta carpeta podemos encontrar el archivo server.xml. Lo abrimos y añadimos....

```
<Connector
        protocol="org.apache.coyote.http11.Http11NioProtocol"
        port="8443" maxThreads="200"
        scheme="https" secure="true" SSLEnabled="true"
        keystoreFile="C:/ficherodos.jks" keystorePass="abc123."
        clientAuth="false" sslProtocol="TLS"/>
```

Cada uno tendrá que configurar la linea que está en negrita con los datos correspondientes en cada caso.

1.3 Configurar nuestra aplicación para que use SSL

En el descriptor de despliegue de la aplicación (web.xml), añade el siguiente trodo de código. Como puedes observar, hay una etiqueta <transport-guarantee> , que está comentada. Si toma el valor NONE, la aplicación no iría por protocolo seguro. Si toma valor CONFIDENTIAL, si.

NOTA: en la etiqueta <web-resource-name> pon tu nombre de aplicación.

```
<security-constraint>
        <web-resource-collection>
                <web-resource-name>calculadora</web-resource-name>
                <url-pattern>/*</url-pattern>
        </web-resource-collection>
        <user-data-constraint>
                <!-- <transport-guarantee>NONE</transport-guarantee> -->
                <transport-guarantee>CONFIDENTIAL</transport-guarantee>
        </user-data-constraint>
</security-constraint>
```

Tarea a desarrollar

Implementa la seguridad SSL/TLS para tu aplicación calculadora.