INSTALAR CERTIFICADO SSL EN WEB SERVER TOMCAT

Este manual consta de cinco secciones:

Crear CSR con KEYTOOL.

El CSR (Certificate Signing Request) es un formato de archivo que contiene un texto cifrado con la información necesaria para la creación de un certificado SSL.
Primero debes ubicarte en el fichero que contendrá el almacén de certificados que utilizarás durante toda la operación. Normalmente se encuentra en:
"**$TOMCAT_HOME/appserver/domains/MiDominio/config/**".

$JBOSS_HOME Fichero home configurado para la instancia de la aplicación tomcat.
MiDominio Fichero correspondiente a la instancia del dominio en el que instalaremos el certificado SSL.

A continuación, seguí estos pasos:

1. Para generar dicho archivo en primera instancia se debe crear el almacén .JKS de certificados, que en un futuro contendrá el certificado SSL.
   El comando para la creación del llamado **KEYSTORE** (almacén de claves) será el siguiente:

```
keytool -genkey -alias midominio -keyalg RSA -keystore almacen.jks -keysize 2048
```

aliasclave Alias que tendrá la clave privada que se genere dentro del almacén (esto se almacena localmente en su servidor).

almacen.jks Nombre del almacén de certificados generados.

IMPORTANTE: No modificar y/o sobreescribir el jks, dado que se perderá la clave privada asociada.

1. Ingresá una contraseña para el almacén de certificados. La contraseña por defecto es **changeit.** (para evitar problemas futuros, elegí una contraseña diferente).
2. Ingresá la información correspondiente a tu organización:

First and Last Name Nombre del dominio en formato FQDN. Recordá que, si solicitas un Wildcard deberás agregar el "*." antes del dominio. Ej. ***.midominio.com**.
Organizational Unit (OU) Sector que solicita dicho certificado. Por ejemplo: Departamento de Informática.
Organization (O) Razón social de la empresa registrada en DGI.
City/Locality Ciudad o localidad en la que se encuentra.
State/Province Estado o Provincia. Ej: Montevideo
Country Code Corresponde al código ISO de dos letras del país en el que se encuentra la empresa. En el caso de Uruguay, ingresá UY.

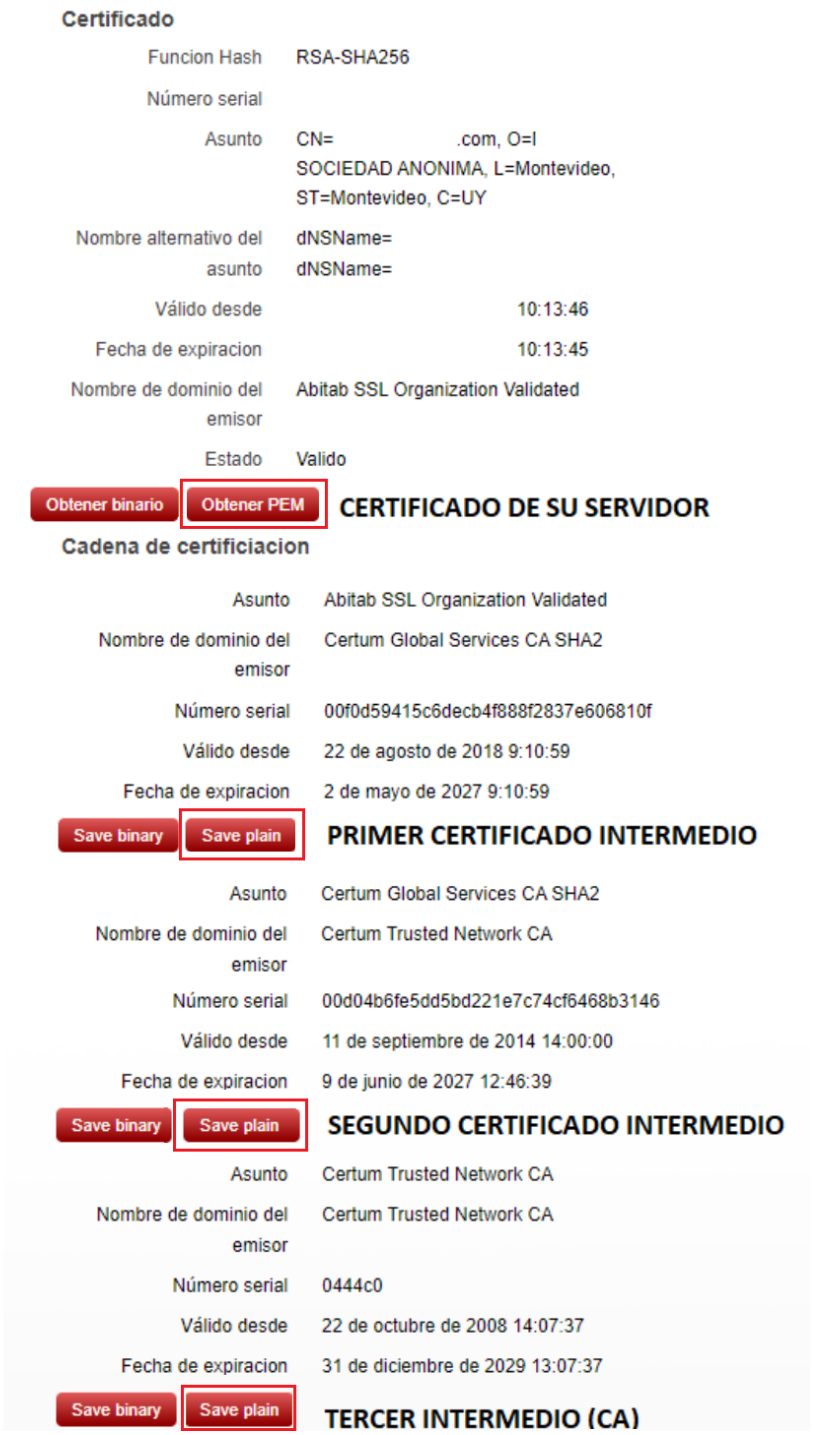
3. Una vez se crea el almacén jks se debe generar csr mediante el siguiente comando:

```
keytool -certreq -alias midominio -keystore almacen.jks -file midominio.csr
```

Te solicitará la contraseña del almacén de claves ingresada anteriormente en el paso 2.
Al finalizar, en el mismo fichero en que se encuentra ubicada tendrás creado el archivo .csr

Instalar certificado

1. **PASO A. DESCARGA DEL CERTIFICADO:** Cuando se emita el certificado , te llegará un mail desde ID Digital Abitab.
   Este mail contendrá un enlace con los certificados necesarios para la instalación del SSL, al ingresar podrás visualizar el certificado (clave pública) con su correspondiente cadena de intermediarios (Chain of Trust):



1. En la imágen anterior podes observar que un certificado se puede descargar como tipo binary o plain. A efectos del tipo de almacén de claves con el que cuentas (jks) descargá los certificados en la opción ***"Obtener PEM " y "Save plain"***.
   Al hacer esto, los archivos descargados tendrán la extensión. pem.
2. **El primer certificado que figura es el certificado solicitado,** luego se visualizan los certificados que representan a la "Chain of Trust".
   Para facilitar su uso en este manual, le pondremos nombres a cada certificado descargado:
   - ➔ El certificado de su servidor (clave pública), le llamaremos: **certificado.pem**
   - ➔ El primer certificado de la "Chain of Trust" será: **intermedio1.pem**
   - ➔ El segundo certificado de la "Chain of Trust" será: **intermedio2.pem**
   - ➔ El tercer certificado (root) de la "Chain of Trust" será: **intermedio3.pem**

3. Una vez descargados y con nombre, ubicá el certificado en la misma carpeta en la que se encuentra el almacén de certificados JKS.

4. **IMPORTACIÓN DEL CERTIFICADO EN EL ALMACÉN DE CERTIFICADOS:** Por consola tendrás que ubicarte en la carpeta donde se encuentra el almacén de certificados.
5. Importá los certificados intermedios en el almacén respetando el orden de la cadena de confianza. Para esto debes ejecutar los siguientes comandos en el orden que se muestran:

   1. keytool -import -alias intermedio1 -file intermedio1.pem -almacen.jks
   2. keytool -import -alias intermedio2 -file intermedio2.pem -almacen.jks
   3. keytool -import -alias intermedio3 -file intermedio3.pem -almacen.jks

El último certificado de estos (CA Root), a veces no lo permite instalar dentro del almacén. En estos casos simplemente ignorá el mensaje de error y no instales dicho certificado (por defecto muchos servidores de aplicaciones ya cuentan con este certificado integrado).

6. Para finalizar, importá el certificado en el almacén de certificados ejecutando el siguiente comando:

   keytool -import -alias certificado -file certificado.pem -almacen.jks

Configuraciones necesarias

Por último, debes asegurarte de que Tomcat acceda al almacén de certificados:

1. Accedé a través de tu editor de texto preferido al archivo ***server.xml*** (generalmente ubicado en el home de Tomcat dentro de la carpeta conf).
2. Dentro del archivo, ubicá el conector a utilizar por el SSL.
3. Dentro del conector, modifícalo de acuerdo a la versión de TOMCAT con la que cuentes:

En Tomcat 4.xx:

```
<clientAuth="false" protocol="TLS"     keystoreFile="/etc/tomcat5/tomcat.keystore"
 keystorePass="changeit" />
```

En Tomcat 5.xx, 6.xx y 7.xx:

```
<-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
      <Connector
           port="8443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"

keystoreFile="$TOMCAT_HOME/appserver/domains/MiDominio/config/almacen.jks"
           keystorePass="changeit"
           clientAuth="false" sslProtocol="TLS"/>
```

keystoreFile Path donde se encuentra tu almacén de certificados.

port Puede ser 8443 o 443

Si tu versión de TOMCAT es anterior a la 7.xx deberás cambiar el keystorePASS por keypass.

4. Luego de realizar el cambio sobre ***server.xml***, guardá los cambios realizados y reinicia el servicio de TOMCAT.
5. Conectate a la aplicación y verificá si se publica correctamente.