agesic
agencia de gobierno electrónico y sociedad de la información

PRESIDENCIA
REPÚBLICA ORIENTAL DEL URUGUAY

AGESIC
Área de Tecnología

Configuración de SSL en servidores de aplicaciones JavaEE

Historial de Revisiones

Fecha	Versión	Descripción	Autor	Aprobado Por
27/06/2011	1.0	Versión inicial	Marcelo Caponi	
05/11/2012	1.4	Agregado configuración para JBoss 7	Sergio Pío Alvarez	
01/03/2013	1,5	Agregado errores conocidos para JBoss 7	Sergio Pío Alvarez	

Nombre actual del archivo: Tutorial_Configuración_SSL_servidores_Tomcat_JBoss_v1.5.odt

Liniers 1324 piso 4, Torre Ejecutiva Sur
Montevideo – Uruguay
Tel./Fax: (+598) 2901.2929*
Email: contacto@agesic.gub.uy
www.agesic.gub.uy

Tabla de Contenidos

1 Introducción

En el presente documento se describe como habilitar comunicación vía SSL en servidores de aplicaciones. Los servidores que se presentan son JBoss AS versiones 5.1.0.G.A y 7.1.1.Final, y Apache Tomcat versión 7.0.14.

Por cada servidor se detalla la configuración de SSL con autenticación simple y mutua. Por último, se ilustra un mecanismo sencillo de probar que la configuración realizada funciona correctamente. Para esto, se toma como cliente del servidor un navegador Mozilla Firefox, se le aplican las configuraciones pertinentes, se ejecuta un pedido al servidor y se verifica el funcionamiento esperado.

En todos los casos se hará referencia a “<server_home>” como la ruta de instalación del servidor. En cualquier caso, es deseable que la ruta en donde se instala el servidor no contenga espacios.

En las siguientes dos secciones se ilustra como se habilita la comunicación SSL en cada uno de los servidores, y en la última sección se presenta una forma de verificar que la configuración realizada sea correcta (el procedimiento es el mismo en todos los casos).

1.1 Requerimientos

Para poder configurar cualquier servidor de aplicaciones para admitir conexiones HTTPS es necesario contar con un almacén de certificados, el cual debe contener el certificado que identificará al propio servidor ante los clientes. En el caso de la PGE, este certificado es emitido por AGESIC para cada uno de los organismos que pretendan ofrecer servicios a través de la plataforma de interoperabilidad. Adicionalmente, si se desea autenticación mutua (lo que es un requerimiento en la PGE, también se debe contar con un almacén de certificados de confianza, el cual debe contener al certificado de la CA que firma los certificados de los clientes (en el caso de la PGE, la CA es AGESIC, y el certificado es proporcionado por AGESIC, siendo el mismo para todos los clientes).

2 Tomcat 6.0

En esta sección se detalla cómo configurar el servidor Apache Tomcat con autenticación simple y mutua.

2.1 Configuración de SSL con autenticación simple

Para habilitar SSL en Tomcat (autenticación simple) se debe editar el archivo *server.xml* que encuentra en la ruta *<server_home>/conf/*. En dicho archivo, se deberá descomentar la sección que corresponde al conector SSL. Dicha sección se ilustra en la Figura 1.

```
<Connector port="8443"
           protocol="HTTP/1.1"
           SSLEnabled="true"
           maxThreads="150"
           scheme="https"
           secure="true"
           sslProtocol="TLS"
           keystoreFile="<server_home>/server/default/conf/agesic.keystore"
           keystorePass="xxx"
           clientAuth="false" />
/>
```

Figura 1: Conector SSL en Tomcat

Los atributos que deben configurarse necesariamente son *keystoreFile, keystorePass y clientAuth*, el resto de los atributos pueden dejarse con su valor de fábrica.

A continuación se describen los atributos antes mencionados:

- **keystoreFile**: se debe especificar la ruta al keystore.
- **keystorePass**: se debe especificar la contraseña del keystore.
- **clientAuth**: se debe especificar false para configurar SSL con autenticación simple (sin autenticación mutua).

2.2 Configuración de SSL con autenticación mutua

Para habilitar autenticación mutua, se debe mantener la misma configuración que en autenticación simple excepto por el atributo *clientAuth*, y a su vez, configurar dos atributos más, *truststoreFile* y *truststorePass*.

Los atributos que se deberían tratar además en la configuración son:

- **clientAuth:** El valor de este atributo deberá ser cambiado a *true* para habilitar la autenticación mutua.
- **truststoreFile**: Es la ruta donde se encuentra el archivo truststore.
- **truststorePass**: Es donde se especifica la contraseña del truststore.

En la Figura 2 se ilustra un conector similar al de la Figura 1 pero con autenticación mutua habilitada.

```
<Connector port="8443"
           protocol="HTTP/1.1"
           SSLEnabled="true"
           maxThreads="150"
           scheme="https"
           secure="true"
           sslProtocol="TLS"
           keystoreFile="<server_home>/server/default/conf/agesic.keystore"
           keystorePass="xxx"
           truststoreFile="<server_home>/server/default/conf/agesic.truststore"
           truststorePass="yyy"
           clientAuth="true" />
/>
```

Figura 2: Conector SSL con autenticación mutua en Tomcat

3 JBoss AS 5.1.0.GA

En esta sección se detalla cómo configurar el servidor JBoss versión 5.1.0.GA con autenticación simple y mutua.

3.1 Configuración de SSL con autenticación simple

La configuración del servidor JBoss AS 5.1.0.GA para habilitar SSL es similar a la requerida para el servidor Apache Tomcat. En el caso de JBoss AS 5.1.0, se debe editar el archivo *server.xml,* el cual se encuentra en la ruta: *<server_home>/<configuracion>/deploy/jbossweb.sar/,*donde <configuracion> debe ser remplazada por el nombre de la configuración que se utilice al iniciar JBoss; por defecto, es "default". En dicho archivo se deberá descomentar la sección que corresponde al conector SSL. Dicha sección se ilustra en la Figura 3.

```
<Connector
    port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    address="${jboss.bind.address}"
    keystoreFile="C:/desarrollo/jboss-5.1.0.GA/server/default/conf/agesic.keystore"
    keystorePass="xxx"
    sslProtocol="TLS"
    clientAuth="false"
/>
```

Figura 3: Conector SSL en JBoss

Los atributos que deben configurarse son *keystoreFile, keystorePass y clientAuth,* el resto de los atributos pueden dejarse con sus valores por defecto.

- **keystoreFile**: debe indicar la ruta donde se encuentra el keystore.
- **keystorePass**: debe especificar la contraseña del keystore.
- **clientAuth**: se debe especificar el valor *false* para configurar SSL con autenticación simple (sin autenticación mutua).

3.2 Configuración de SSL con autenticación mutua

Para habilitar autenticación mutua se debe mantener la misma configuración que en autenticación simple, excepto por el valor del atributo clientAuth, y agregar dos atributos adicionales relativos a la autenticación mutua.

Los atributos que se deben configurar son los siguientes:

- **clientAuth:** debe cambiarse el valor a *true* para habilitar la autenticación mutua.
- **truststoreFile**: debe indicar la ruta donde se encuentra el archivo truststore.
- **truststorePass**: debe especificar la contraseña del truststore.

En la Figura 4 se ilustra un conector similar al de la Figura 3 pero con autenticación mutua.

```
<Connector
    port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    address="${jboss.bind.address}"
    keystoreFile="C:/desarrollo/jboss-5.1.0.GA/server/default/conf/agesic.keystore"
    keystorePass="xxx"
    truststoreFile="C:/desarrollo/jboss-5.1.0.GA/server/default/conf/agesic.truststore"
    truststorePass="yyy"
    sslProtocol="TLS"
    clientAuth="true"
/>
```

Figura 4: Conector SSL en JBoss con autenticación mutua

4 JBoss AS 6.1.0.Final

La configuración de SSL en JBoss AS 6.1.0.Final es idéntica a la que corresponde a JBoss AS 5.1.0.GA. Aunque JBoss AS 6.1.0.Final provee mecanismos que no estaban disponibles en versiones anteriores, los mismos mecanismos de versiones anteriores siguen siendo válidos en esta versión.

5 JBoss AS 7.1.1.Final

En esta sección se detalla cómo configurar el servidor JBoss versión7.1.1.Final con autenticación simple y mutua. En este documento se explica la configuración para el modo standalone del servidor; en modo domain la configuración es exactamente igual, excepto por el archivo de configuración: donde se haga refrencia al archivo *<server-home>/standalone/configuration/*standalone.xml debe considerarse en su lugar el archivo *<server-home>/domain/configuration/domain.xml*.

5.1 Configuración de SSL con autenticación simple

La configuración del servidor JBoss AS 7.1.1.Final es bastante diferente respecto de versiones anteriores. En este caso, el archivo que se debe editar es el llamado standalone.xml y se encuentra en *<server_home>/standalone/configuration/* . En dicho archivo se deberá editar la sección que corresponde al subsistema con nombre *"urn:jboss:domain:web:1.1"* (buscar en el archivo el texto '*subsystem xmlns="urn:jboss:domain:web:1.1"*'). Dentro de la configuración de dicho subsistema, agregar la configuración de un nuevo conector, llamado https, y dentro de él la configuración de SSL como se muestra en la figura 5:

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="false">
    <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
    <connector
      name="https"
      protocol="HTTP/1.1"
      scheme="https"
      socket-binding="https"
      enable-lookups="false"
      secure="true">
      <ssl
        name="ssl"
        protocol="TLSv1"
        certificate-key-file="C:\desarrollo\jboss-7.1.1.Final\standalone\configuration\pge_test_ssl.keystore"
        keystore-type="JKS"
        password="keypassword"
        verify-client="false"/>
    </connector>
    <virtual-server name="default-host" enable-welcome-root="true">
        <alias name="localhost"/>
        <alias name="example.com"/>
    </virtual-server>
</subsystem>
```

Figura 5: Conector SSL en JBoss

La configuración del conector incluye los siguientes atributos:

- **name**: debe ser “https”.
- **protocol**: debe ser "HTTP/1.1".
- **scheme**: debe ser “https”.
- **socket-binding**: debe ser "https".
- **enable-lookups**: debe ser "false".
- **secure**: debe ser "true".

A su vez, la configuración del puerto SSL, que debe estar incluida dentro de la configuración del conector SSL, incluye los siguientes atributos:

- **name**: debe ser “ssl" .
- **protocol**: debe ser "TLSv1" .
- **certificate-key-file**: debe contener la ruta al keystore.
- **keystore-type**: debe ser "JKS" (a menos que en lugar de un keystore se prefiera utilizar un certificado en formato p12 (o pfx) en cuyo caso debe ser “PKCS12”).
- **password**: debe especificar la contraseña del keystore.
- **verify-client**: para no utilizar autenticación mutua, debe ser "false".

5.2 Configuración de SSL con autenticación mutua

Para habilitar autenticación mutua se debe mantener la misma configuración que en el caso de autenticación simple, excepto por el valor del atributo verify-client, y agregar dos atributos adicionales relativos a la autenticación mutua.

Los atributos que se deben configurar son los siguientes:

- **verify-client:** debe cambiarse el valor a *true* para habilitar la autenticación mutua.
- **ca-certificate-file**: debe indicar la ruta donde se encuentra el archivo truststore.
- **ca-certificate-password**: debe especificar la contraseña del truststore.

En la Figura 6 se ilustra un conector similar al de la Figura 5 pero con autenticación mutua.

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="false">
    <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
    <connector
      name="https"
      protocol="HTTP/1.1"
      scheme="https"
      socket-binding="https"
      enable-lookups="false"
      secure="true">
      <ssl
        name="ssl"
        protocol="TLSv1"
        certificate-key-file="C:\desarrollo\jboss-7.1.1.Final\standalone\configuration\pge_test_ssl.keystore"
        keystore-type="JKS"
        password="keypassword"
        verify-client="false"
        ca-certificate-file="C:\desarrollo\jboss-7.1.1.Final\standalone\configuration\pge_test_ssl.trustore"
        ca-certificate-password="trustpassword"
        />
    </connector>
    <virtual-server name="default-host" enable-welcome-root="true">
        <alias name="localhost"/>
        <alias name="example.com"/>
    </virtual-server>
</subsystem>
```

Figura 6: Conector SSL en JBoss con autenticación mutua

5.3 Errores conocidos

Unable to load certificate key /usr/local/jboss-7.1.1/mtss_prod.jks (error:0906D06C:PEM routines:PEM_read_bio:no start line). Si se obtiene este error, verificar que el atributo native del sybsystema

urn:jboss:domain:web:1.1" tenga el valor false. Uno de los efectos de que este atributo tenga el valor true es que al momento de especificar, en la configuración de SSL, el certificado digital que deberá usar el servidor, habrá que especificar dos archivos separados, uno conteniendo la clave privada y otro la clave pública, en lugar de un solo archivo de tipo keystore conteniendo el certificado apropiado. Concretamente, el error que puede verse será similar al siguiente:

```
ERROR [org.apache.coyote.http11.Http11AprProtocol] (MSC service thread 1-2) Error inicializando punto final (endpoint): java.lang.Exception: Unable to load certificate key /usr/local/jboss-7.1.1/mtss_prod.jks (error:0906D06C:PEM routines:PEM_read_bio:no start line)

at org.apache.tomcat.jni.SSLContext.setCertificate(Native Method) [jbossweb-7.0.13.Final.jar:]

at org.apache.tomcat.util.net.AprEndpoint.init(AprEndpoint.java:644) [jbossweb-7.0.13.Final.jar:]

at org.apache.coyote.http11.Http11AprProtocol.init(Http11AprProtocol.java:121) [jbossweb-7.0.13.Final.jar:]

at org.apache.catalina.connector.Connector.init(Connector.java:983) [jbossweb-7.0.13.Final.jar:]

at org.jboss.as.web.WebConnectorService.start(WebConnectorService.java:267) [jboss-as-web-7.1.1.Final.jar:7.1.1.Final]

at org.jboss.msc.service.ServiceControllerImpl$StartTask.startService(ServiceControllerImpl.java:1811) [jboss-msc-1.0.2.GA.jar:1.0.2.GA]

at org.jboss.msc.service.ServiceControllerImpl$StartTask.run(ServiceControllerImpl.java:1746) [jboss-msc-1.0.2.GA.jar:1.0.2.GA]

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1110) [rt.jar:1.6.0_18]

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:603) [rt.jar:1.6.0_18]

at java.lang.Thread.run(Thread.java:636) [rt.jar:1.6.0_18]

13:15:31,855 INFO [org.apache.coyote.http11.Http11AprProtocol] (MSC service thread 1-1) Arrancando Coyote HTTP/1.1 en puerto http--0.0.0.0-8080

13:15:31,880 ERROR [org.jboss.msc.service.fail] (MSC service thread 1-2) MSC00001: Failed to start service jboss.web.connector.https: org.jboss.msc.service.StartException in service jboss.web.connector.https: JBAS018007: Error starting web connector

at org.jboss.as.web.WebConnectorService.start(WebConnectorService.java:271)

at org.jboss.msc.service.ServiceControllerImpl$StartTask.startService(ServiceControllerImpl.java:1811) [jboss-msc-1.0.2.GA.jar:1.0.2.GA]

at org.jboss.msc.service.ServiceControllerImpl$StartTask.run(ServiceControllerImpl.java:1746) [jboss-msc-1.0.2.GA.jar:1.0.2.GA]

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1110) [rt.jar:1.6.0_18]

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:603) [rt.jar:1.6.0_18]

at java.lang.Thread.run(Thread.java:636) [rt.jar:1.6.0_18]

Caused by: LifecycleException: Falló la inicialización del manejador de protocolo: java.lang.Exception: Unable to load certificate key /usr/local/jboss-7.1.1/mtss_prod.jks (error:0906D06C:PEM routines:PEM_read_bio:no start line)

at org.apache.catalina.connector.Connector.init(Connector.java:985)

at org.jboss.as.web.WebConnectorService.start(WebConnectorService.java:267)
```

6 Verificación de configuración SSL

En esta sección se detalla cómo verificar que las respectivas configuraciones realizadas anteriormente funcionan de manera correcta.

6.1 Verificación de autenticación simple

Una vez realizadas las configuraciones de SSL tanto para JBoss como para Apache Tomcat, se podrá probar que la misma funciona adecuadamente de la siguiente manera:

- Ejecutar el navegador web Mozilla Firefox[1] e ingresar la URL correspondiente al servidor de aplicaciones:
  - Apache Tomcat: "https://localhost:8443"
  - JBoss AS 5.1.0.GA: "https://localhost:8443/admin-console"
  - JBoss AS 7.1.1.Final: "https://localhost:8443/console"

    Notar que en todos los casos se especifica "https" como esquema y "8433" como puerto; el puerto puede variar según se haya configurado en el respectivo archivo de configuración del servidor de aplicaciones.
- Una vez solicitada la anterior URL, el navegador web intentará autenticar al servidor, para lo cual éste presentará su certificado propio. Si la CA que firmó el mismo está en el almacén de confianza de Mozilla Firefox (truststore), la conexión se establecerá. En caso contrario, se presentará una pantalla indicando que se está presentando un certificado emitido por una CA en la que Mozilla Firefox no confía (Figura 7), dejando a criterio del usuario decidir si de todas maneras acepta la conexión segura utilizando un certificado no confiable, o si cancela la comunicación. Para permitir la comunicación, seguir con los siguientes pasos.

1 La versión de Firefox en el cual se verificó, fue la 3.6.22, en otras versiones puede ser igual o diferir sutilmente.

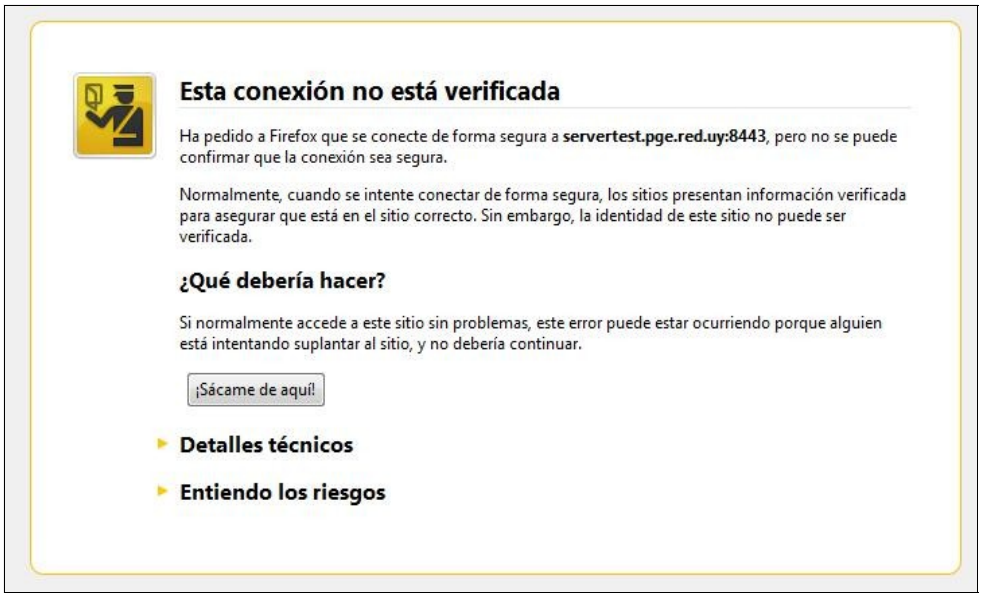


Figura 7: Mensaje que ocurre en una conexión SSL en la que no se confía en la CA

- Luego, expandir la opción “Entiendo los riesgos” tal como se ilustra en la Figura 8.

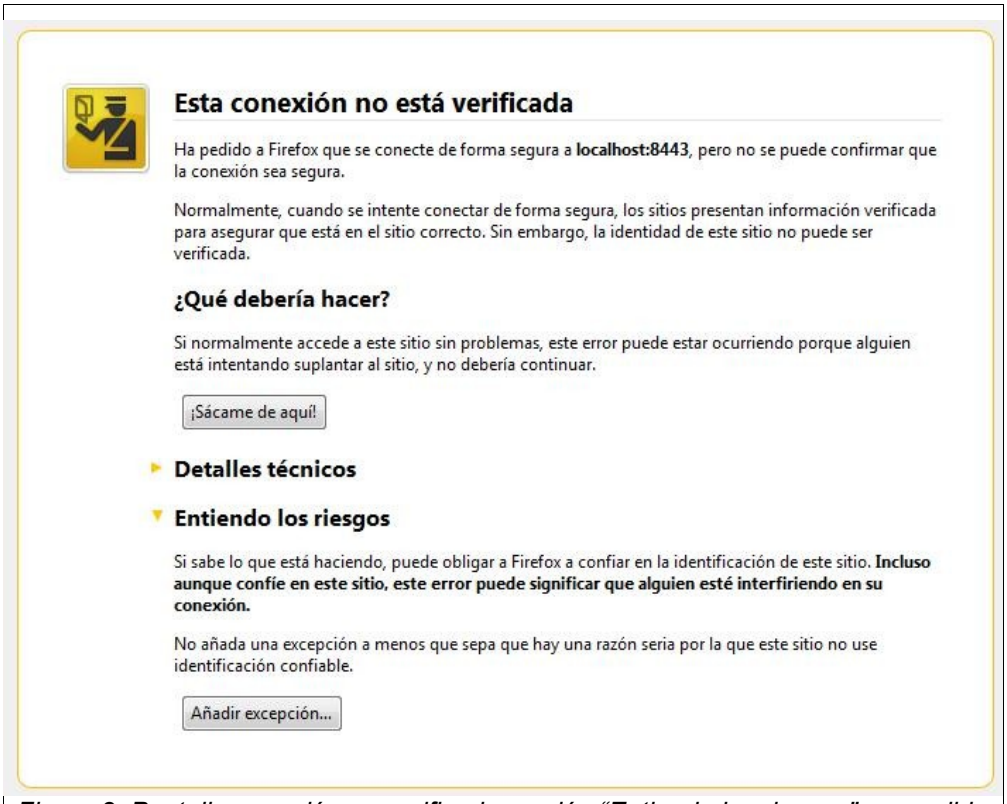


Figura 8: Pantalla conexión no verificada, opción “Entiendo los riesgos” expandida

- Luego, presionar el botón “Añadir excepción...” obteniendo la pantalla que se ilustra en la Figura 9.

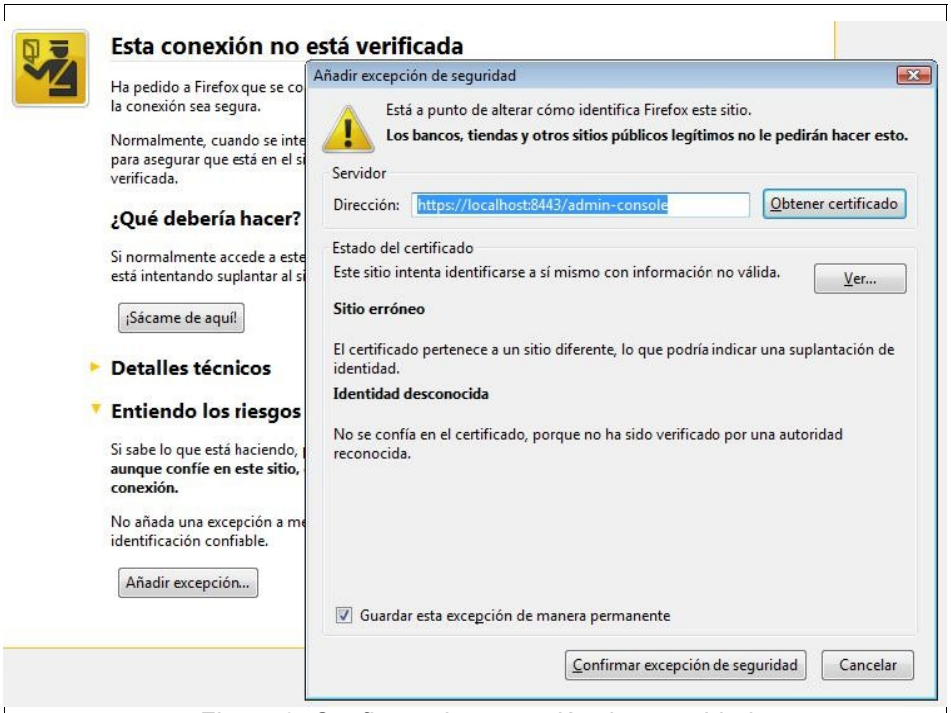


Figura 9: Confirmando excepción de seguridad

- A continuación, si se presiona el botón “Ver..”, se podrá ver el certificado que presenta el servidor, tal como se ilustra en la Figura 10.

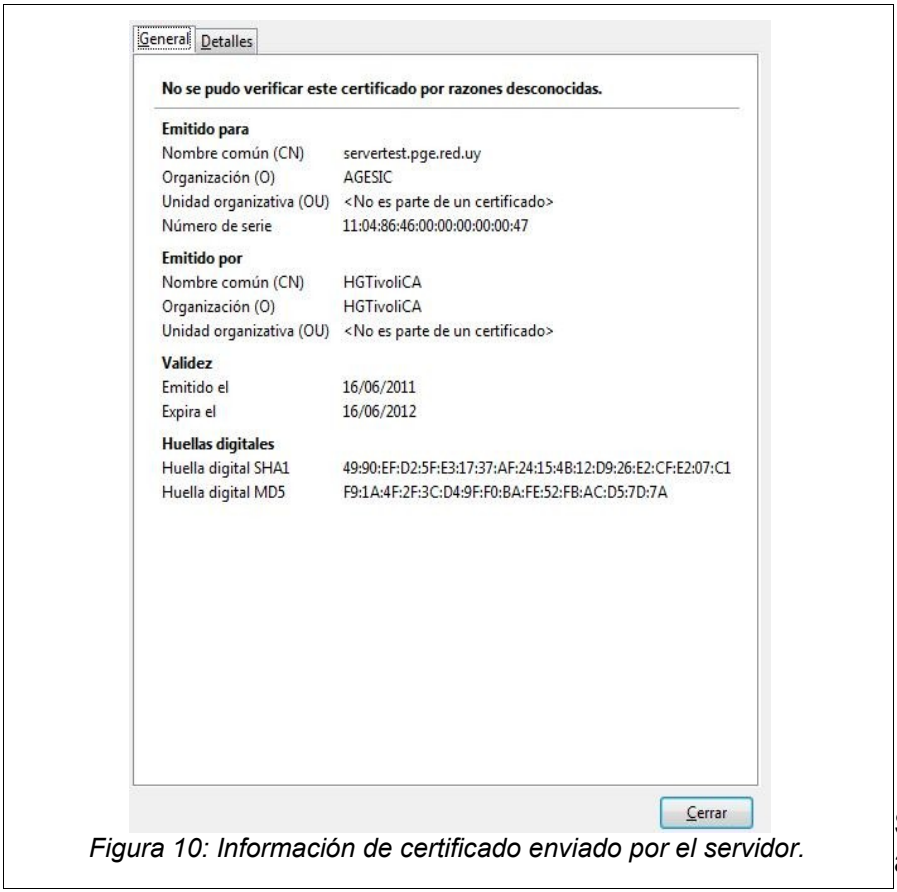


Figura 10: Información de certificado enviado por el servidor.

Se puede apreciar que los datos del certificado corresponden al servidor correcto. En este caso, puede verse que se trata de un certificado de un servidor de testing emitido por AGESIC.

Para continuar, se presiona el botón cerrar, se vuelve a la ventana anterior (Figura 9), y se presiona el botón “Confirmar excepción de seguridad”.

- Por último, se debe verificar que la conexión se estableció correctamente. Si todo es correcto, se deberá ver la consola de administración de los respectivos servidores. Eso será un indicador de que la conexión se efectuó correctamente. La pantalla visualizada depende del servidor de aplicaciones.

6.2 Verificación de autenticación mutua

Para la autenticación mutua, los pasos que deben seguirse son exactamente los mismos que los que se detallaron en la sección 6.1 . La única diferencia que existe es que en el último paso (donde se presiona el botón “Confirmar excepción de seguridad”), el navegador Mozilla Firefox solicitará al usuario que seleccione el certificado que deberá presentarle al servidor de aplicaciones (este certificado debió haber sido cargado previamente en el keystore de Mozilla Firefox).

En este paso, si ya se tiene un certificado importado (en caso de que no, ver sección 6.3), Mozilla Firefox desplegará un popup en donde se podrá seleccionarlo, como lo ilustra la Figura 11.

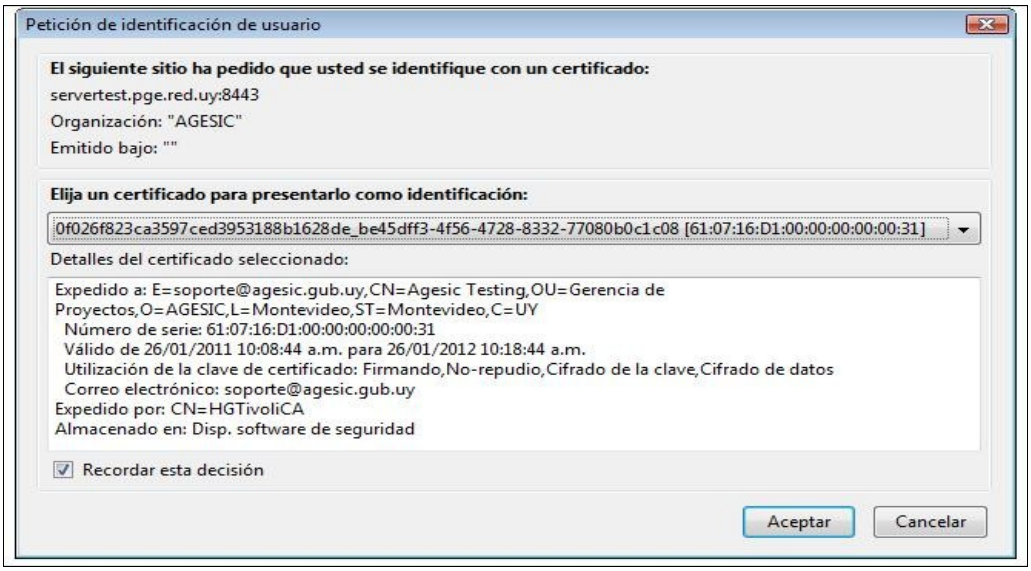


Figura 11: Certificado digital que presenta Firefox al intentar establecer una conexión SSL con autenticación mutua

Al presionar aceptar, se finaliza la conexión y se muestran las consolas administrativas correspondientes.

6.3 Importación de certificados en Mozilla Firefox

Para importar un certificado en Mozilla Firefox[2], se deben completar los siguientes pasos:

- Seleccionar en el menú “Herramientas”, la opción “Opciones...” tal como se ilustra en la Figura 12.

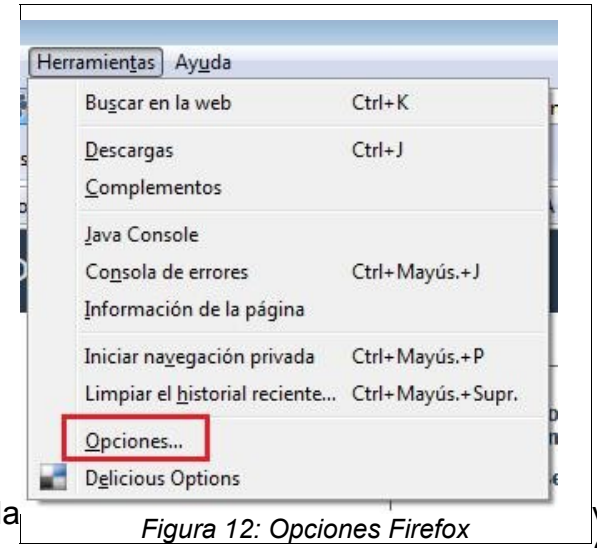


Figura 12: Opciones Firefox

- En la ventana que se despliega (Figura 13), presionar la solapa “Avanzado”, luego la solapa “Certificados” y luego el botón “Ver certificados”.

2 La versión de Firefox en el cual se importaron los certificados fue la 3.6.22, en otras versiones puede ser diferente, aunque el proceso general es el mismo.

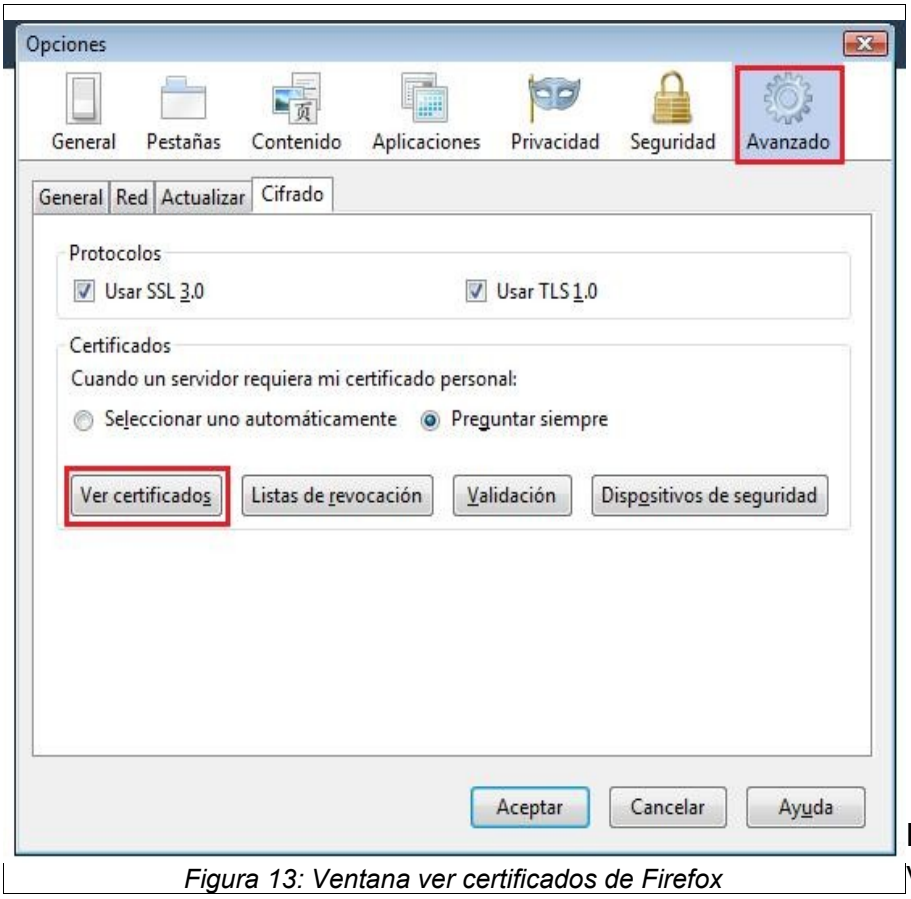
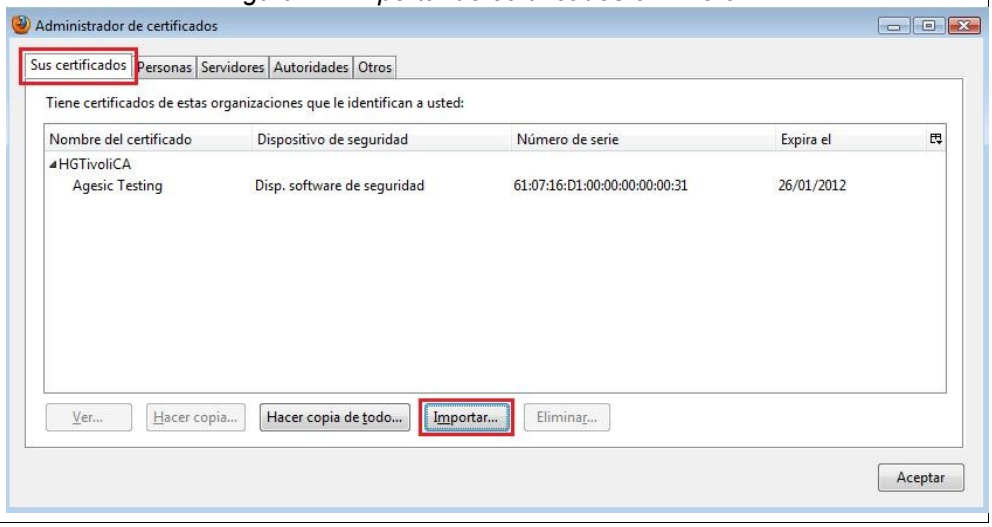


Figura 13: Ventana ver certificados de Firefox

- La ventana que se despliega (Figura 14), muestra los certificados que se tienen ya instalados, permite importar uno nuevo. Pasar a la solapa "Sus certificados".

Figura 14: Importando certificados en Firefox



Para importar un nuevo certificado, presionar el botón importar, seleccionamos el

archivo que contiene el certificado (normalmente, un archivo con extensión pfx o p12), ingresar la contraseña del certificado y confirmar la importación.