Manual del Programador del Módulo de Autenticación de @Firma

Versión 4.0

Documento nº:	TI-20-1074-MPA-001
Revisión:	04
Fecha:	24/02/2005
Período de retención:	Permanente durante su período de vigencia + 3 años después de su anulación

CONTROL DE COMPROBACIÓN Y APROBACIÓN

Documento nº: TI-20-1074-MPA-001

Revisión: 4

Fecha: 24/02/2005

REALIZADO

06/08/2004

Moisés Manuel

Infante

Gómez

Analista Programador

COMPROBADO

17/09/2004

José Antonio

Márquez

Contreras

Director @firma

APROBADO

17/09/2004

José Antonio

Márquez

Contreras

Director @firma

CONTROL DE MODIFICACIONES

Documento nº: TI-20-1074-MPA-001

Revisión: 4

Fecha: 24/02/2005

Rev. 1

Fecha 06/08/2004

Autor/es MMIG

Descripción Documentación inicial

Rev. 2

Fecha 17/09/2004

Autor/es MMIG

Descripción Actualización API's y diagramas UML

Rev. 3

Fecha 21/01/2005

Autor/es MMIG

Descripción Se añade un código de error nuevo

Rev. 4

Fecha 24/02/2005

Autor/es MMIG

Descripción Se añade el punto 6.3 Configurar la Página de Error por Aplicación

CONTROL DE DISTRIBUCIÓN

Documento nº: TI-20-1074-MPA-001

Revisión: 4

Fecha: 24/02/2005

Copias Electrónicas:

La distribución de este documento ha sido controlada a través del sistema de información.

Copias en Papel:

La vigencia de las copias impresas en papel está condicionada a la coincidencia de su estado de revisión con el que aparece en el sistema electrónico de distribución de documentos.

El control de distribución de copias en papel para su uso en proyectos u otras aplicaciones es responsabilidad de los usuarios del sistema electrónico de información.

Fecha de impresión 24/02/2005 13:28

Distribución en Papel:

Nombre o Cargo y (Organización)	Nº de Ejemplares	Referencia de la carta de transmisión y fecha

Índice

1 Objeto

El objeto de este documento es describir la utilización del Módulo de Autenticación de @Firma válido para cualquier aplicación, ya sea web o de consola, mediante la utilización de certificados digitales X.509.

El módulo de Autenticación contiene las interfaces y componentes web necesarios para la realización de los siguientes procesos:

- Autenticación/Reautenticación Web.
- Autenticación mediante RMI-IIOP y WebServices.

2 Alcance

El presente documento recoge la utilización del módulo de Autenticación de @Firma, y contempla los siguientes objetivos:

- Describir los pasos necesarios para desarrollar una aplicación web que utilice la autenticación web.
- Describir los pasos necesarios para desarrollar una aplicación que utilice las interfaces RMI-IIOP y WebServices disponibles en la plataforma @Firma.
- Describir los métodos disponibles en las interfaces anteriores y la lógica de utilización de los mismos.
- Describir detalladamente el ejemplo de utilización de dichas interfaces y componentes web que se adjunta en la plataforma para facilitar el trabajo al nuevo desarrollador de aplicaciones, tanto RMI-IIOP como WebServices.
- Especificar los diferentes tipos de errores que se pueden dar al integrar una aplicación con la plataforma, y su resolución.

3 Siglas

AC	Autoridad de Certificación
CPD	CRL Distribution Point
CRL	Lista de Revocación de Certificados
DP	Punto de Distribución
FNMT-RCM	Fábrica Nacional de Moneda y Timbre, Real Casa de la Moneda
JSP	JavaServer Pages
LDAP	Lightweight Directory Access Protocol
PC	Ordenador Personal
RSA	Rivest Shamir Adleman
JRE	Java Runtime Environment
JDK	Java Development Kit
PKCS#7	Public Key Cryptography Standard Number 7
ASN.1	Abstract Syntax Notation One
EJB	Enterprise Java Bean
SSL	Secure Socket Layer
Keystore	Almacén de certificados digitales X.509

4 Documentos de Referencia

- Documento TI-20-1074-ADM-001, Manual de Administración de @firma.
- Documento TI-20-1074-INS-001, Manual de Instalación de @firma.

5 Introducción

El módulo de Autenticación es un sistema genérico y centralizado basado en certificados digitales, que permite a cualquier aplicación ya desarrollada o por desarrollar delegar el proceso de autenticación en este sistema.

El módulo de Autenticación contiene las interfaces y componentes web necesarios para la realización de los siguientes procesos:

- Autenticación/Reautenticación Web.
- Autenticación mediante RMI-IIOP y WebServices.

Las interfaces se encuentras disponibles mediante tecnología RMI-IIOP y WEBSERVICES, ambas securizadas mediante SSL y JAAS, lo cual proporciona un doble nivel de seguridad.

5.1 Sobre módulo Autenticación

En el caso de Autenticación/Reautenticación Web, se utiliza un componente de llamada y otro de retorno, ambos componentes proporcionados en la tecnología de la aplicación a integrar (paginas jsp, php, etc) para que sea más fácil su manipulación y adaptación para cada aplicación en concreto. El componente de llamada básicamente hace una llamada al Servidor SSL desplegado en la Fachada @Firma, éste realiza las comprobaciones pertinentes al certificado y devuelve la información del certificado al componente de retorno.

En el caso de Autenticación mediante RMI-IIOP o WebServices simplemente se obtiene una referencia a la interfaz de programación y se utiliza uno de sus tres métodos para realizar la autenticación.

Con la plataforma @Firma se distribuyen ejemplos que muestran la utilización todas las interfaces y componentes mencionados. A lo largo de este documento se describen todas las interfaces disponibles y los ejemplos de utilización de las mismas.

Adicionalmente, para facilitar la tarea del desarrollador / integrador de aplicaciones se distribuye el "javadoc" correspondiente a todas las interfaces, en el cual se detallan los métodos, parámetros, excepciones, etc. Esto permitirá que en el presente manual nos centremos principalmente en el funcionamiento omitiendo detalles específicos.

5.2 Identificación y Autenticación

Para el establecimiento de cualquier aplicación como un sistema confiable y seguro en el entorno de funcionamiento éste debe asegurar un conjunto de requerimientos aceptables por los usuarios finales de la aplicación.

Identificación:

En principio, y es el objeto de este documento, el usuario debe estar identificado para acceder a la aplicación. El uso de certificado digital proporcionará la garantía de este requerimiento.

Autenticación:

Garantiza que las entidades involucradas en la transacción son realmente lo que aparentan ser y el certificado digital garantiza este requerimiento.

Confidencialidad:

Entendiendo por confidencialidad la capacidad de mantener la información fuera del alcance de usuarios no autorizados. Para ello se utiliza el protocolo estándar integrado en el navegador y servidor Web, Secure Socket Layer (SSL), que mediante técnicas criptográficas oculta el contenido de la información que viaja a través de la Red.

Logado por Certificados

Todo sistema de logado por certificados está compuesto de dos fases:

- **Autenticación**. En esta fase se verifica que un usuario que desea acceder a un sistema o aplicación es quien dice ser. Para ello el usuario presenta su certificado y se verifica que es confiable, válido y que no está revocado.

- **Autorización**. Fase en la que se comprueba si el usuario previamente autenticado pertenece a la aplicación y en caso positivo se le asigna el rol y los permisos correspondientes con el cual accede a la aplicación en cuestión.

5.3 Certificado digital

Un certificado digital es un documento electrónico que establece la identidad del poseedor de una clave pública y una clave privada propias, el plazo de validez y el algoritmo de encriptación, entre otras características. Estas características son establecidas por una Entidad Certificadora o Autoridad de Certificación (AC), que es la encargada de desarrollar los procedimientos y tecnologías para proveer la generación y gestión de certificados digitales.

5.4 Protocolo SSL

El protocolo SSL es el utilizado para la comunicación entre la plataforma @Firma y el navegador del usuario y también en las comunicaciones internas que se realicen.

Al ser iniciada una sesión en Internet utilizando SSL, los dos participantes (servidor Web y navegador del usuario) autentican la identidad del servidor Web y negocian una clave simétrica (DES o similar) que será utilizada para cifrar los siguientes mensajes a ser intercambiados. Esta negociación se realiza en modo cifrado, con la utilización de un sistema de clave pública (RSA o similar), a partir del certificado digital del servidor Web. Este mecanismo garantiza que ningún intruso tome conocimiento de la clave simétrica a ser utilizada en el transcurso de la sesión.

Cada mensaje intercambiado durante esa sesión está protegido, además de la criptografía, por un número de secuencia y por un código de autenticación de mensaje, que refuerza la garantía de integridad y es generado utilizando una función *hash* aplicada sobre el cómputo del mensaje, número de secuencia, la clave utilizada y algunas constantes.

El protocolo SSL es muy simple de activar; basta que el servidor Web disponga de un certificado de autenticación de servidor, y su URL ser accedida utilizando *https*. Los usuarios deben estar en posesión de certificados digitales de autenticación de cliente, que son los que habitualmente proporciona la FNMT-RCM a través de la Agencia Tributaria o Autoridad de Registro especializada, como es la Junta de Andalucía.

El protocolo SSL autentica solamente al servidor Web, la autenticación del usuario es realizada a nivel de aplicación por medio de los certificados digitales.

6 Proceso Autenticación/Reautenticación Web

Las aplicaciones web que empleen tecnología de páginas web dinámicas en el servidor podrán hacer uso del sistema de autenticación/reautenticación por certificados.

La base de este sistema de autenticación se centra en el protocolo SSL. La versión 3 de este protocolo y las versiones TLS permiten la opción del establecimiento de conexiones seguras usándose certificados reales (no generados por el protocolo) en ambos extremos de la conexión. Esto junto a un navegador que acepte conexiones SSL v3 o TLS y un servidor correctamente configurado permite llevar a cabo de forma simple el sistema de autenticación por certificados descrito a continuación.

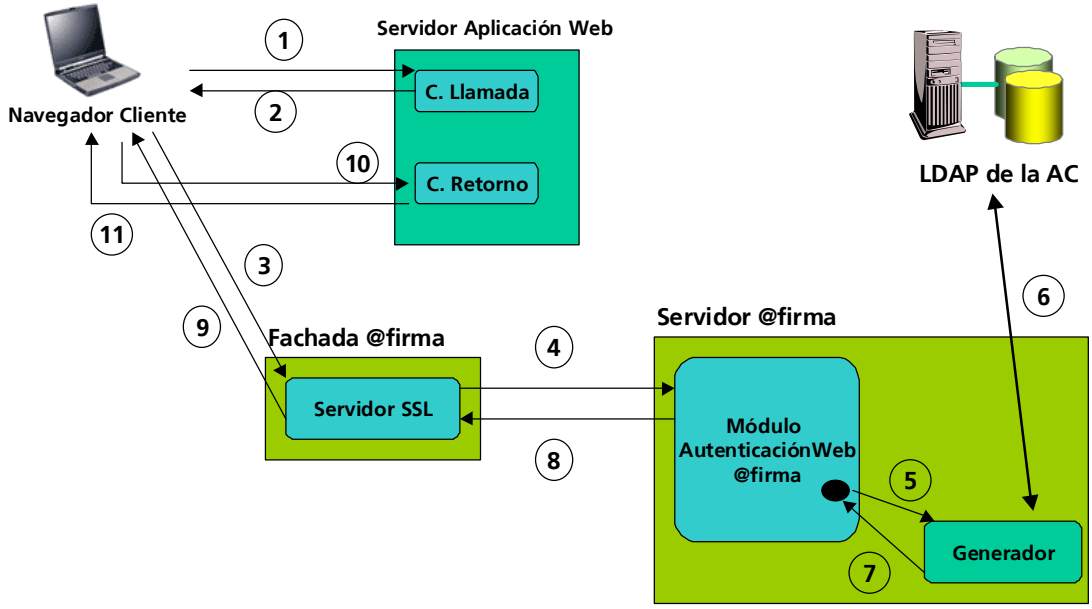
Reautenticación

El concepto de **reautenticación** aparece como consecuencia de la existencia de aplicaciones que necesitan realizar la autenticación varias veces, como puede ser un portal de entrada desde el cual se pueda acceder a distintas funcionalidades cada una de ellas con una autenticación independiente. En una misma sesión web, el protocolo SSL solo realiza el intercambio de certificados una sola vez, con lo cual en el caso anterior el navegador sólo pide el certificado anterior una única vez y no "n veces" como sería aconsejable. Las aplicaciones suelen comprobar que un usuario no accede más de una vez al servidor de @firma en una misma sesión web, para evitar que si un usuario deja abierta la ventana del navegador, otra persona pueda utilizar "fraudulentamente" el certificado del usuario.

Para solucionar este inconveniente una solución podría ser la existencia de varios Servidores SSL para autenticación y que la aplicación llamara a uno distinto cada vez que necesitara la autenticación. Esta solución ha sido adoptada por la AEAT (Agencia Tributaria).

La solución aportada por Telvent es la instalación de nuestro componente de reautenticación, que permite simular la existencia de varios "servidores SSL virtuales" configurados a través de un certificado wildcard.

La siguiente figura resume de forma global el proceso de autenticación:



1. El Navegador Cliente accede al Componente de Llamada.

2. El Componente de Llamada redirecciona al cliente web hacia el ServidorSSL.

3. El navegador establece la conexión https y le pide al usuario que seleccione el certificado que se enviará al ServidorSSL (SSL v 3.0).

4. El ServidorSSL obtiene el certificado y los parámetros (nombre aplicación y valor aleatorio) de la llamada con los que realiza una petición RMI-IIOP al ModuloAutenticacionWeb, securizada mediante SSL y JAAS.

5. El ModuloAutenticacionWeb crea el generador asociado a la aplicación.

6. El generador verifica la validez del certificado y comprueba su estado de revocación.

7. El generador saca la información del certificado y genera con ella un objeto Subject.

8. El ModuloAutenticacionWeb encripta con la clave 3DES el resultado del proceso, lo codifica en Base64 y lo devuelve al ServidorSSL.

9. El ServidorSSL genera la llamada al Componente de Retorno pasando como parámetro los datos encriptados.

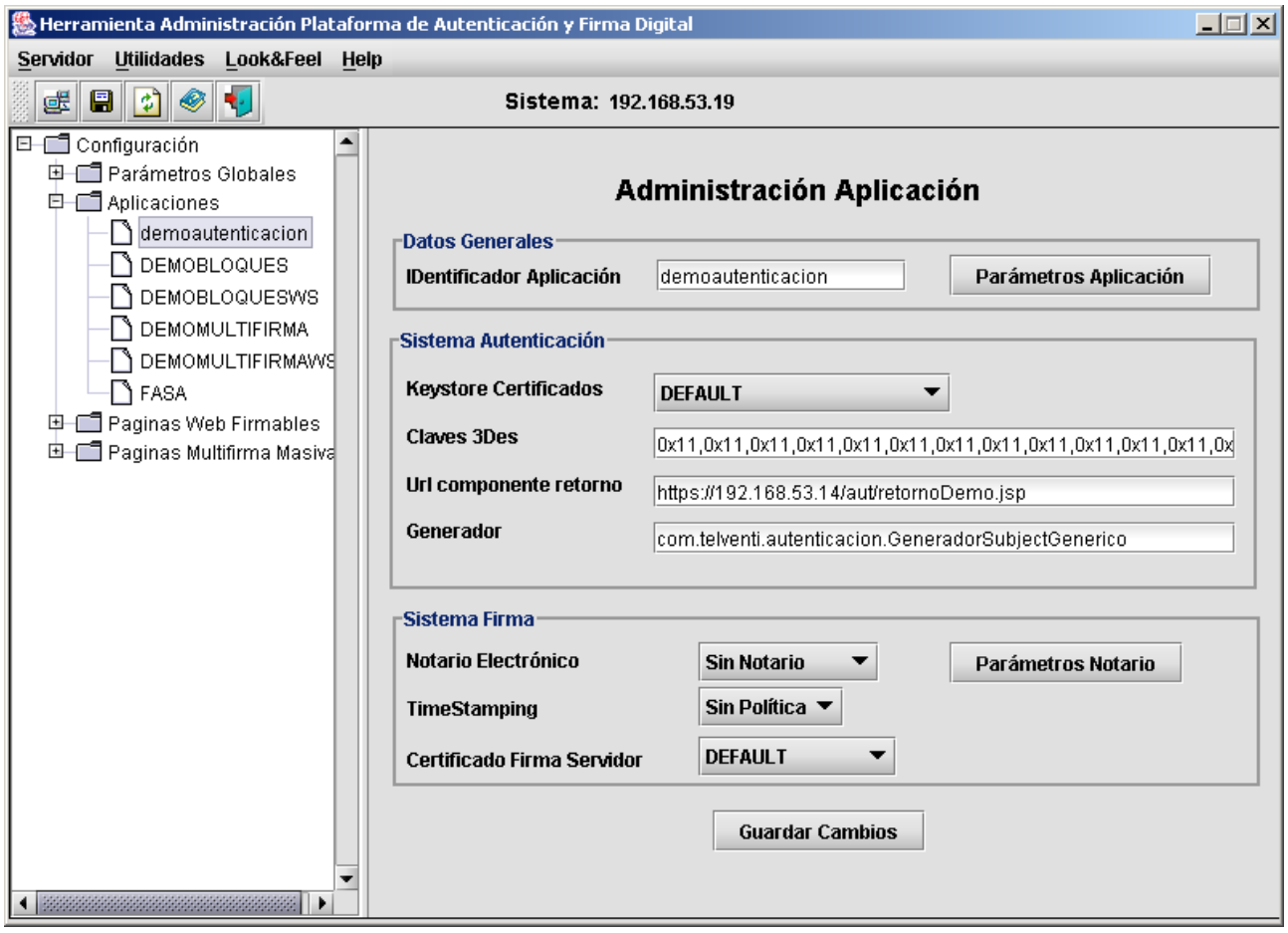
10. EL navegador realiza la llamada al Componente de Retorno el cuál decodifica los datos en Base 64 y los desencripta con la clave 3DES. Con los datos del certificado devueltos se procede a realizar la fase de autorización.

11. En base a las comprobaciones anteriores el Componente de Retorno devuelve el control a la aplicación para que proceda a la fase de autorización, o redirecciona a una página de error indicando el error producido.

Para poder desarrollar una aplicación que utilice la autenticación/reautenticación web es necesario dar alta una aplicación en la herramienta de Administración de la plataforma @Firma.

A continuación se describen los pasos necesarios a llevar a cabo en dicha herramienta. Para información mas detallada sobre la herramienta de Administración véase el manual del Administrador del Sistema.

Una vez inicializada la herramienta seleccionar el nodo "Aplicaciones" del árbol de la izquierda. Esto nos mostrará en la parte izquierda un listado de aplicaciones existentes. A continuación pulsar el botón "Nueva Aplicación" de la parte izquierda de la pantalla. La siguiente pantalla muestra el resultado:

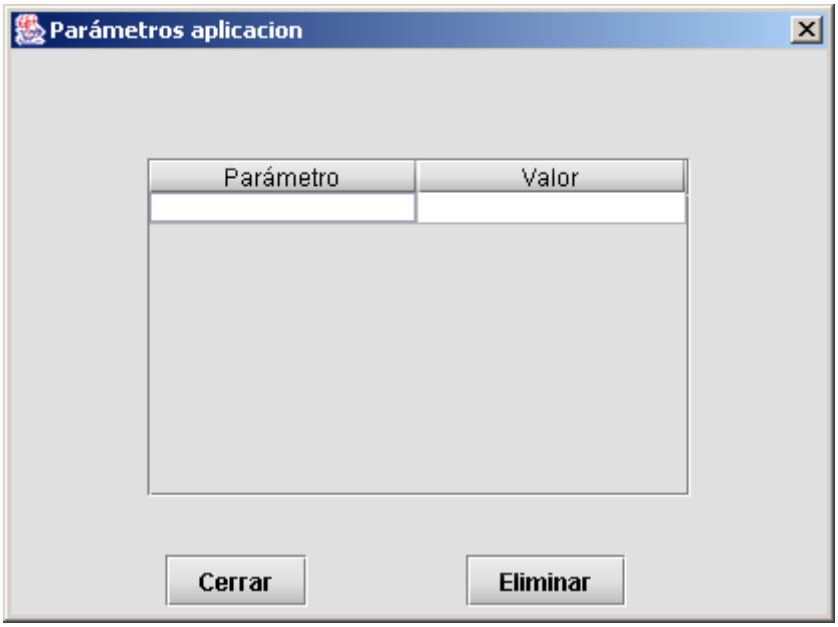


Los parámetros a introducir son:

Datos Generales:

- **Identificador Aplicación:** Identificador de la aplicación.

- **Parámetros Aplicación:** parámetros específicos para cada aplicación. Es decir, las aplicaciones concretas pueden definir sus propios parámetros de configuración bajo la forma de parejas nombre/valor. Estos parámetros son empleados por los generadores de Subjects programados para cada aplicación en concreto. El generador de Subject Genérico utilizado por la mayoría de aplicaciones no tiene definidos ningún par de valores. Pulsando en el botón se muestra la ventana siguiente:



Sistema Autenticación

- **Keystore Certificados:** Indica el keystore utilizado para verificar los certificados digitales de usuario utilizados en la autenticación.

- **Clave 3Des:** Es la clave DES necesaria para encriptar los datos de vuelta en las aplicaciones de autenticación web. Es importante anotar esta clave para distribuirla posteriormente a los desarrolladores que utilicen esta aplicación. La clave puede y ha de ser cambiada con "sensata" frecuencia poniendo especial cuidado es distribuirla a todas aquellas personas que trabajen con la aplicación configurada.

- **URL componente retorno**. Indica la URL del componente que trata los datos de vuelta, y que está ubicado en el servidor de aplicaciones donde se necesite autenticación.

- **Generador**. Es el componente de lógica de negocio desarrollado en Java que ofrece la funcionalidad requerida para una aplicación en concreto, y que ha sido desarrollado según las especificaciones de este manual. Por defecto, se utiliza el componente "com.telventi.autenticacion.GeneradorSubjectGenerico".(Ver el punto 8)

Por otro lado toda aplicación que desee integrar la autenticación web debe instalar dos componentes dentro de su sistema, un modulo para generar la petición de autenticación contra el servidor SSL (componente de llamada) y otro para recoger los datos de vuelta del proceso de autenticación (componente de retorno). Ambos componentes son proporcionados normalmente en forma de plantillas jsp, asp, php, etc para que sea más fácil su adaptación e integración con la plataforma @firma 4.

6.1 Componentes de Llamada y Retorno para Autenticación

A continuación se describen las características de los componentes de llamada y retorno para la utilización de la autenticación web de la plataforma @Firma.

NOTA: En la Fachada @Firma debe estar instalado el ServidorSSL para Autenticación.

- Componente de Llamada. Este elemento captura la sesión del usuario y redirige la petición de autenticación hacia la Fachada @Firma mediante la realización de una llamada por HTTPS al puerto donde esté desplegado el ServidorSSL, pasando como parámetros el "Identificador Aplicación" registrado en @firma y la Sesión capturada. La llamada quedaría así:

https://<fachada_afirma>:<puertoAutenticacion>/?ap=<id_aplicación>&sesion=<id_sesion>

- Componente de Retorno. Este elemento recibe el resultado del proceso de autenticación procedente de la Fachada @Firma. Lo decodifica en Base64, lo desencripta con la clave 3DES proporcionada por el Administrador del Servidor de @firma, lo trata según la aplicación, y compara la Sesión actual con la sesión pasada como parámetro para verificar que es la misma sesión y que nadie está intentando acceder copiando la URL directamente del navegador. También comprobará que no se ha llamado más de una vez en la misma sesión del navegador. Los datos desencriptados serán de la siguiente forma:

  <datos según aplicación>;<id_session>

NOTA: Para información detallada ver los ejemplos.

6.2 Componentes de Llamada y Retorno para Reautenticación

A continuación se describen las características de los componentes de llamada y retorno para la utilización de la reautenticación web de la plataforma @Firma.

NOTA: En la Fachada @Firma debe estar instalado el ServidorSSL para Reautenticación.

- Componente de Llamada. Este elemento redirige la petición de autenticación hacia la Fachada @Firma mediante la realización de una llamada por HTTPS al puerto donde esté desplegado el ServidorSSL, pasando como parámetros el "Identificador Aplicación" registrado en @firma y un valor numérico aleatorio, dicho valor hay que guardarlo como un parámetro de la sesión actual con el nombre "_Fielaut_Sesion_llamada". La llamada quedaría así:

https://<fachada_afirma>:<puertoAutenticacion>/?ap=<id_aplicación>&sesion=<valor_aleat>

- Componente de Retorno. Este elemento recibe el resultado del proceso de autenticación procedente de la Fachada @Firma. Lo decodifica en Base64, lo desencripta con la clave 3DES proporcionada por el Administrador del Servidor de @firma, la trata según la aplicación, y compara el valor guardado como parámetro de la sesión con el nombre "_Fielaut_Sesion_llamada" valor pasado como parámetro para verificar que es el mismo valor y que nadie está intentando acceder copiando la URL directamente del navegador, una vez comprobado que son iguales se debe de eliminar el parámetro "_Fielaut_Sesion_llamada" de la sesión. Los datos una vez desencriptados serán de la siguiente forma:

  <datos_según_aplicación>;<valor_aleatorio_pasado>

NOTA: Para información detallada ver los ejemplos.

6.3 Configurar la Página de Error por Aplicación.

Las aplicaciones de autenticación/reautenticación web podrán tener una página de error propia para los casos de error siguientes:

- Error de conexión RMI con @firma. (Error 1)
- Error en la obtención de certificado. (Error 2)
- Error en los parámetros de entrada. (Error 6)

Para ello deberán de incluir un nuevo parámetro en la llamada a @firma llamado "zona" que servirá para identificar a la pagina de error, normalmente tendrá el mismo valor que el parámetro "ap" siendo otro parámetro distinto para el caso de que varias aplicaciones compartan la misma página de error. La llamada quedaría así:

https://<fachada_afirma>:<puertoAutenticacion>/?ap=<id_aplicación>&sesion=<id_sesion>&zona=<id_zona>

Además el administrador de @firma deberá incluir en el fichero "errorCertificadoAut.jsp", que es la página de error por defecto de autenticación/reautenticación ubicada en la Fachada de Firma (%JBOSS_HOME%\server\all\deploy\firmadigital.war\servcio\formsBackEnd), una nueva entrada del tipo:

```
if(strZona.equals("<id_zona>")){

  respose.sendRedirect("<URL_pagina_error>?error="+strError);

}
```

7 Proceso Autenticación RMI-IIOP y WebServices

El proceso de Autenticación es implementado por la interfaz com.telventi.autenticacion. ModuloAutenticacion.

En el proceso de autenticación de usuario intervienen tres agentes:

- Usuario (cliente).
- Aplicación que utiliza la interfaz.
- Plataforma de Firma (interfaz ModuloAutenticacion).

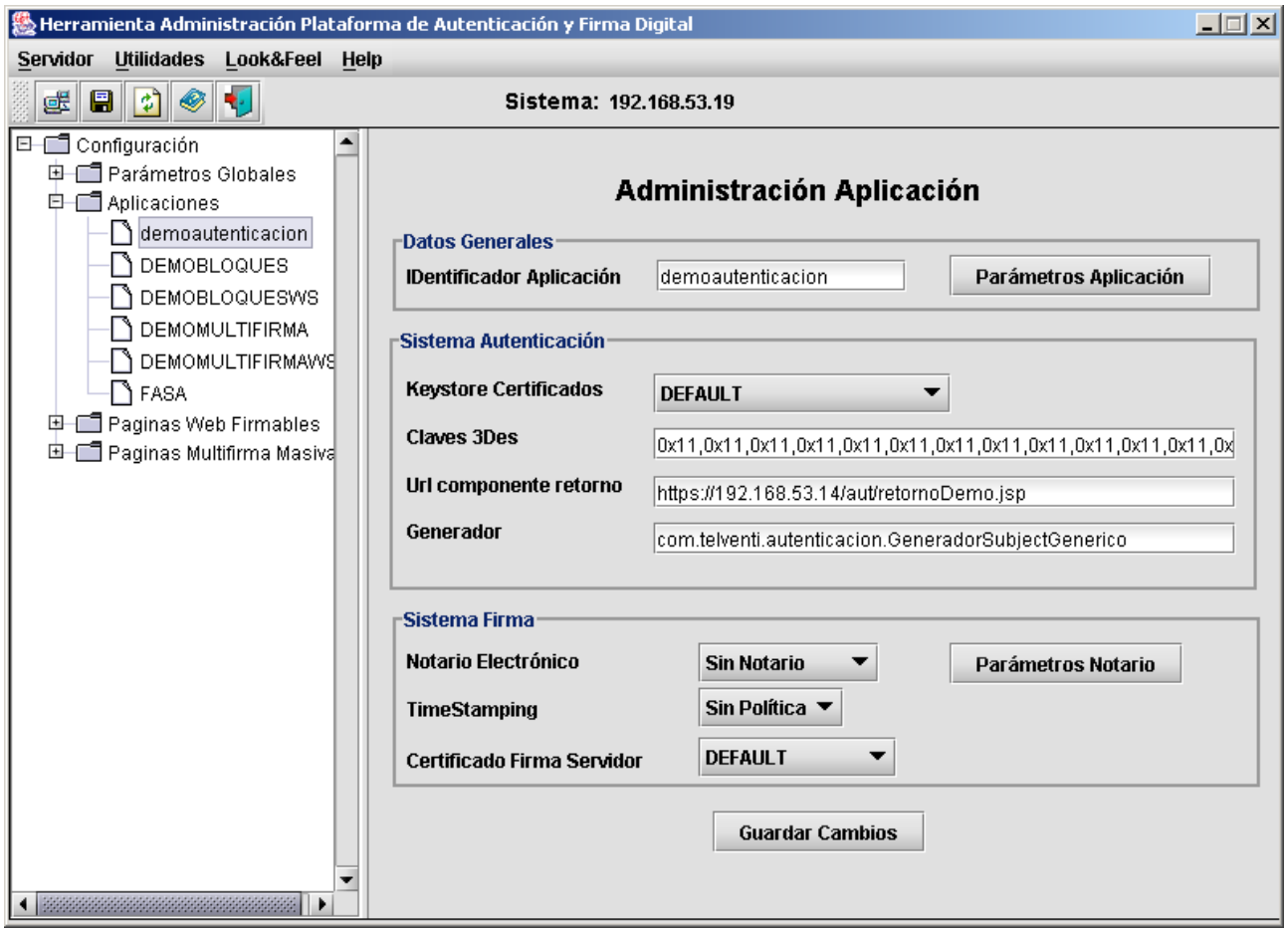
Este proceso se puede describir como un procedimiento en 3 pasos:

1) La aplicación solicita al usuario el certificado a utilizar.
2) El usuario selecciona el certificado.
3) La aplicación solicita a la plataforma @Firma la autenticación del certificado, utilizando cualquiera de los tres métodos disponibles.

Para poder desarrollar una aplicación que utilice la interfaz com.telventi.autenticacion. ModuloAutenticacion es necesario dar alta una aplicación en la herramienta de Administración de la plataforma de Firma.

A continuación se describen los pasos necesarios a llevar a cabo en dicha herramienta. Para información mas detallada sobre la herramienta de Administración véase el manual del Administrador del Sistema.

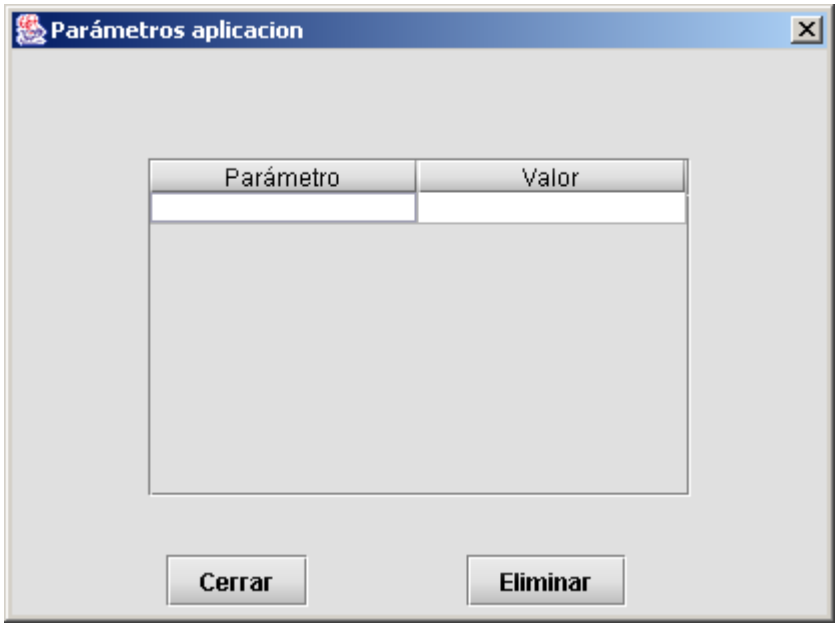
Una vez inicializada la herramienta seleccionar el nodo "Aplicaciones" del árbol de la izquierda. Esto nos mostrará en la parte izquierda un listado de aplicaciones existentes. A continuación pulsar el botón "Nueva Aplicación" de la parte izquierda de la pantalla. La siguiente pantalla muestra el resultado:



Los parámetros a introducir son:

Datos Generales:

- **Identificador Aplicación:** Identificador de la aplicación.
- **Parámetros Aplicación:** parámetros específicos para cada aplicación. Es decir, las aplicaciones concretas pueden definir sus propios parámetros de configuración bajo la forma de parejas nombre/valor. Estos parámetros son empleados por los generadores de Subjects programados para cada aplicación en concreto. El generador de Subject Genérico utilizado por la mayoría de aplicaciones no tiene definidos ningún par de valores. Pulsando en el botón se muestra la ventana siguiente:



Sistema Autenticación

- Keystore Certificados: Indica el keystore utilizado para verificar los certificados digitales de usuario utilizados en la autenticación.

- Generador. Es el componente de lógica de negocio desarrollado en Java que ofrece la funcionalidad requerida para una aplicación en concreto, y que ha sido desarrollado según las especificaciones de este manual. Por defecto, se utiliza el componente"com.telventi.autenticacion.GeneradorSubjectGenerico".(Ver el punto 8)

7.1 Acceso mediante RMI-IIOP

Para utilizar la interfaz RMI-IIOP ModuloAutenticacion se requieren clientes con máquina virtual de JAVA JDK 1.4 o superior.

En el "CD Desarrollo" se proporcionan las librerías necesarias para acceder a la interfaz. Se encuentran ubicadas en el directorio "\ Modulo Autenticacion \ Ejemplos \ RMI-IIOP \ ". El directorio contiene los siguiente ficheros:

- lib \ ModuloAutenticacionTelvent.jar : Clases e Interfaces de acceso a PKI

- auth.conf : Fichero de para configuración de acceso a interfaz mediante Jaas

- lib \ jbossall-client.-jar : Librerías cliente acceso Jboss

Las clases necesarias en este caso son las siguientes:

- com.telventi.conexion.ConexionAutenticacion

- com.telventi.autenticacion.ModuloAutenticacion

- com.telventi.autenticacion.Subject

- com.telventi.autenticacion.SubjectGenerico

A continuación se muestra un pequeño extracto de código JAVA que obtiene una referencia a la interfaz para poder utilizar sus métodos.

```
// IMPORTAMOS LAS CLASES NECESARIAS EN LA CABECERA
import com.telventi.autenticacion.*;
import com.telventi.conexion.ConexionAutenticacion;
        ……
        ……
try {
// Creamos un OBJETO ConexionAutenticacion (SERVIDOR, USUARIO, PASSWORD)
String host = "192.168.53.18";
String usuario = "user01";
String password = "12345";
ConexionAutenticacion conexion = new ConexionAutenticacion(host,usuario,password);

// OBTENEMOS UNA REFERENCIA A LA INTERFAZ A PARTIR DE LA CONEXION
ModuloAutenticacion api = conexion.getModuloAutenticacion();
// AHORA PODEMOS UTILIZAR CUALQUIERA DE SUS METODOS
Subject s  = api.autenticar(idAplicacion, certificado);
}catch(java.lang.Exception ex)  {}
```

El fichero auth.conf debe copiarse en el directorio desde el cual se ejecuta la aplicación.

7.2 Acceso mediante WEBSERVICES

La plataforma de Autenticacion proporciona los Ficheros de Descripción de los Servicios Web (WSDL) publicados en la siguiente URL:

https://<servidor_firma>:<puerto>/axis/servlet/AxisServlet

(Será necesario introducir el usuario/password para Webservices. Consultar con el administrador del sistema.)

Desarrollando una aplicación, para poder comunicarse con el Servicio Web que se desee es necesario generar las clases de acceso al mismo a partir de su fichero descriptor WSDL.

Existen una serie de herramientas que facilitan este trabajo, entre ellas se citan las siguientes:

- Paquete AXIS JAVA: La clase "org.apache.axis.wsdl.WSDL2Java", dado un fichero descriptor WSDL permite generar las clases cliente en tecnología JAVA.

- Paquete AXIS C/C++: La clase "org.apache.axis.wsdl.wsdl2ws.WSDL2Ws", dado un fichero descriptor WSDL permite generar las clases cliente en tecnología C/C++.

- GSoap. Permite generar clases cliente C/C++ a partir de un fichero descriptor WSDL.

- Etc.

A los clientes generados por las herramientas anteriores posiblemente será necesario añadir el código necesario para realizar la comunicación SSL y la autenticación JAAS con la plataforma de firma. En los ejemplos proporcionados con la plataforma (JAVA) se muestran claramente los mecanismos adicionales incorporados.

Como ayuda adicional puede consultarse el "Javadoc" proporcionado en el directorio "Documentación / JavaDoc / Modulo Autenticacion" del CD Desarrollo y los ejemplos desarrollados en JAVA (directorio "Modulo Autenticacion / Ejemplos / WebServices" del CD Desarrollo)

Concretamente, para obtener el fichero descriptor WSDL correspondiente a la interfaz ModuloAutenticacion conectarse a la siguiente URL:

https://<servidor_firma>:<puerto>/axis/services/ModuloAutenticacion?wsdl

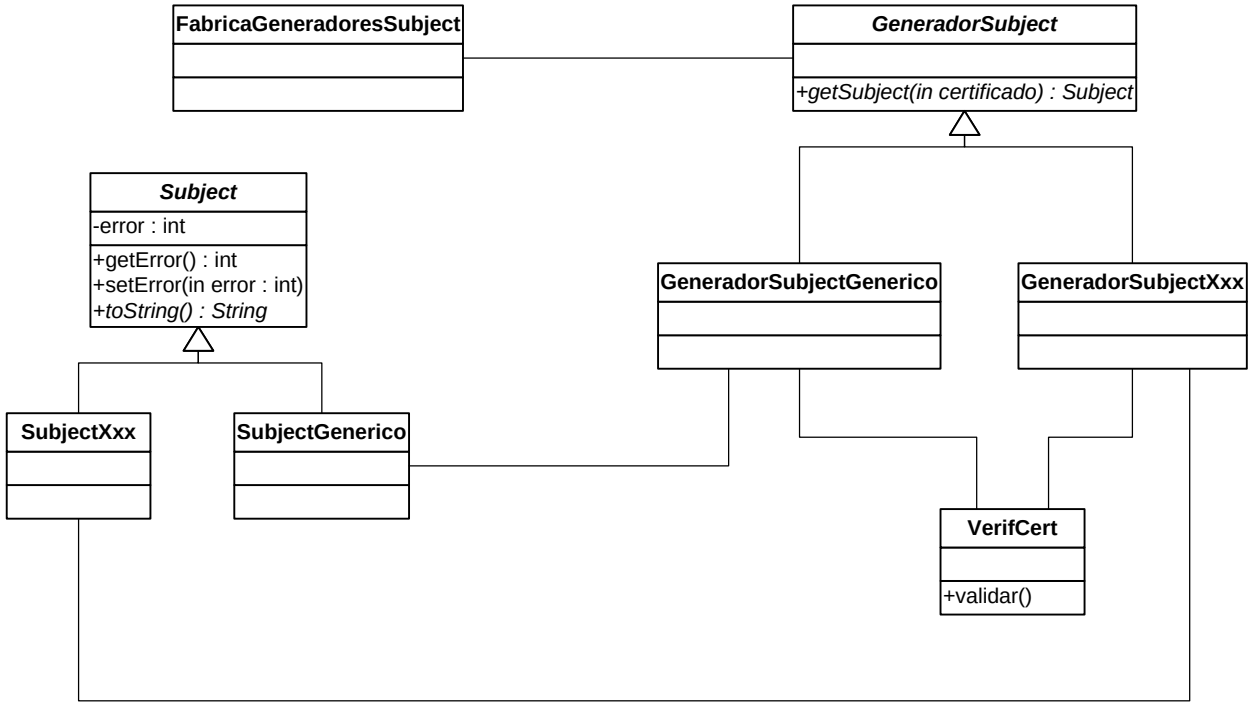
En el menú del navegador *Archivo*, seleccionar *Guardar como...* indicando el nombre multifirma.wsdl.

Como puede verse de la interfaz ModuloAutenticacion sólo se ha publicado por WebServices el método autenticarWS, esto se ha hecho para facilitar la tarea a los integradores de aplicaciones que vayan a utilizar los WebServices ya que todos los métodos realizan lo mismo y este tiene como parámetros de entrada y como valor devuelto el tipo String, tratado como un tipo básico por los WebServices.

8 Subjects y Generadores de Subjects

Como se indicó anteriormente el proceso de logado mediante certificados consta de dos fases: autenticación y la autorización. En principio el Módulo de Autenticación ofrece los componentes necesarios para la autenticación, dejando la fase de autorización en manos de las aplicaciones, aunque ofrece la posibilidad a los desarrolladores de aplicaciones de construir sus propios componentes "generadoresSubjects", con lo cual pueden incluir en ellos la fase de autorización.

Todo ello ha sido diseñado bajo el patrón fábrica como se puede apreciar en el diagrama UML siguiente:



Nota: todas estas clases se encuentran en el paquete com.telventi.autenticacion.

TIPO	FabricaGeneradoresSubjet
Descripción	Clase principal del patrón método fábrica. Esta clase proporciona un generador de Subject en función del parámetro Id de aplicación, el que se halla indicado en la Herramienta de Administración.
Comentarios	Ver javadoc

TIPO	GeneradorSubject
Descripción	Clase abstracta base de los generadores de Subject utilizada para la implementación del patrón Método Fábrica. Las subclases deben de verificar el certificado y extraer de él los datos necesarios para crear el Subject correspondiente (fase de autenticación). Además pueden realizar la fase de autorización.
Comentarios	Ver javadoc

TIPO	GeneradorSubjectGenerico
Descripción	Esta clase representa un generador de Subject genérico para las aplicaciones. El generador realiza la verificación del certificado usando para ello la clase VerifCert, genera un SubjectGenerico con toda la información proporcionada por el certificado y lo devuelve al proceso llamante.
Comentarios	Ver javadoc.

TIPO	Subject
Descripción	Clase abstracta base para los componentes Subject. Tiene métodos para controlar los errores producidos en la verificación y el método abstracto *toString()* necesario para la autenticacion/reautenticacion web. Las subclases deben de tener métodos para la manipulación de la información del certificado necesaria para cada aplicación en concreto.
Comentarios	Ver javadoc

TIPO	SubjectGenerico
Descripción	Clase que implementa un Subject genérico para las aplicaciones. Contiene todos los atributos presentes en el certificado (FNMT, Feste y Notario) al que se refiere y métodos para obtenerlos.
Comentarios	Ver javadoc

El desarrollador que quiera crear su propio Subject y GeneradorSubject deberá hacerlo basándose en el GeneradorSubjectGenerico e incluyendo siempre el siguiente código en el método *getSubject()*:

```
import com.telventi.autenticacion.x509validacion.VerifCert;
…
SubjectXxx subject = new SubjectXxx();
try {
    VerifCert verifCert = new VerifCert();
    verifCert.validar(cert, this.getIdAplicacion());
  } catch (com.telventi.autenticacion.x509validacion.ValidationException e) {
    subject.setError(e.getTipo());
  }
```

9 Aplicaciones de Ejemplo de Autenticación

9.1 Aplicación Autenticación/Reautenticación Web

La aplicación está desarrollada en JSP.

El código de la aplicación se encuentra disponible en el siguiente directorio del CD Desarrollo, respectivamente:

- "Modulo Autenticacion / Ejemplos / Web / Autenticacion"
- "Modulo Autenticacion / Ejemplos / Web / Reautenticacion"

Montar el ejemplo que corresponda según se haya montado el ServidorSSL de la Fachada @Firma.

9.1.1 Poner en marcha la aplicación

Es una aplicación WEB, así pues necesitamos un servidor de aplicaciones, por ejemplo JBOSS.

- Copiar los archivos entradaDemo.jsp y retornoDemo.jsp en el Servidor de Aplicaciones.
- Dar de alta la aplicación en la Herramienta de Administración con el nombre "demoautenticacion". (Si ya existe modificar la configuración para que sea correcta).

Una vez copiado el directorio modificar el siguiente ficheros:

- entradaDemo.jsp: cambiar la dirección ip que aparece por el nombre o dirección ip de la Fachada @Firma.
- RetornoDemo.jsp: cambiar la clave 3DES por la que aparece en la Herramienta de Administración.

Una vez finalizados los pasos anteriores, se accederá mediane la url:

https://<servidor_aplicaciones>:<puerto>/demoautenticacion/entradaDemo.jsp

9.2 Aplicación RMI-IIOP

Se hace uso de la siguiente interfaz de la plataforma:

- com.telventi.autenticacion.ModuloAutenticacion: para realizar la autenticación.

La aplicación está desarrollada en JAVA.

El código de la aplicación se encuentra disponible en el siguiente directorio del CD Desarrollo:

- "Modulo Autenticacion / Ejemplos / RMI-IIOP"

Para ejecutarla seguir los siguientes pasos:

1. Registrar la dll de extracción de Certificados en el sistema. Para ello solo se tiene que copiar el fichero "telventcertfunctions.dll" situado dentro del directorio "Modulo Autenticacion \ Dll certificado" del CD de Desarrollo en el directorio "c:\windows\system32\".

2. Copiar el contenido del directorio al disco duro.

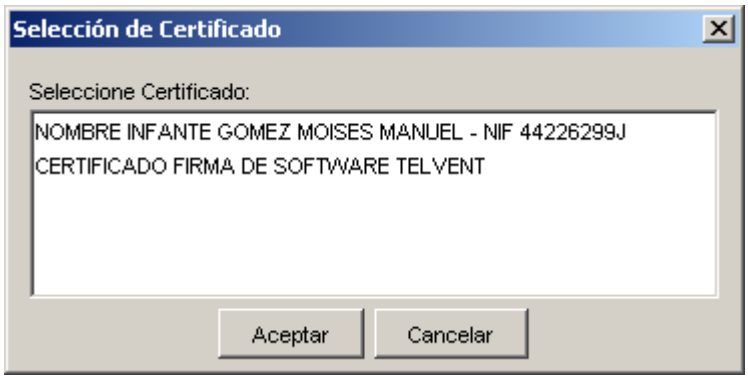
3. Ejecutar el script "run.bat".

Esta Aplicación es una aplicación Java de referencia para integrar aplicaciones similares con el módulo de autenticación de @Firma. En principio presenta una pantalla como la que se indica a continuación:

En la casilla de "Servidor de Autenticación" se ha de introducir el nombre o la dirección IP del Servidor de @Firma contra el cual se desea autenticarse.

En las casillas Usuario/Password se ha de introducir el usuario y el password para poder aceder medianate RMI-IIOP. (Contactar con el Administrador del sistema)

Al seleccionar el botón "Autenticar" nos mostrará una ventana con los certificados que tenemos instalados en el sistema.



Después de seleccionar el certificado para autenticarnos en el Servidor @Firma, la aplicación se conectará con el mismo. El servidor comprobará que el certificado es válido y a continuación devolverá a la aplicación los datos del mismo para que los muestre por pantalla:

9.3 Aplicación Webservices

Se hace uso de la siguiente interfaz de la plataforma:

- com.telventi.autenticacion.ModuloAutenticacion: para realizar la autenticación.

La aplicación está desarrollada en JAVA.

El código de la aplicación se encuentra disponible en el siguiente directorio del CD Desarrollo:

- "Modulo Autenticacion / Ejemplos / WebServices"

Para ejecutarla seguir los siguientes pasos:

1. Registrar la dll de extracción de Certificados en el sistema. Para ello solo se tiene que copiar el fichero "telventcertfunctions.dll" situado dentro del directorio "Modulo Autenticacion \ Dll certificado" del CD de Desarrollo en el directorio "c:\windows\system32\".

2. Copiar el contenido del directorio al disco duro.

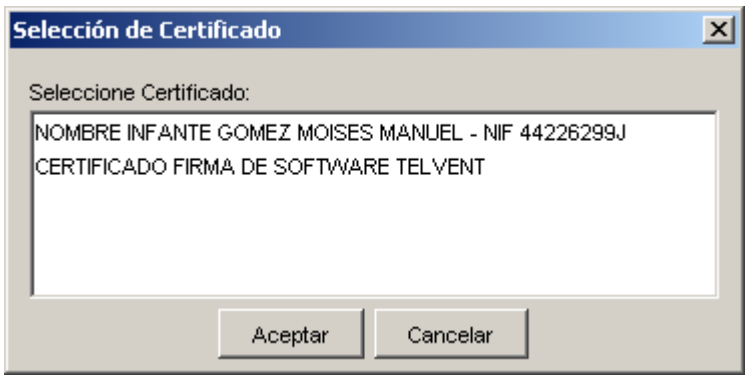
3. Modificar los ficheros:

- demoAutenticacionWS.properties: parámetros para poder ejecutar la aplicación. Los parámetros a configurar son los siguientes:
    - servidorfirma==<nombre o ip del servidor de firma de la plataforma de firma>
    - usuario=<usuario para acceso JAAS WebServices a servidor de firma>
    - password=<password para acceso JAAS WebServices a servidor de firma>
    - trustedstore=<url del fichero trustkeystore que contiene el certificado digital SSL del servidor de firma de la plataforma de firma>Ej:c:\trustkeystore
    - trustedstorepassword=<password el keystore anterior>
- trustKeystore: keystore que contiene la clave pública del certificado digital del servidor de firma de la plataforma de firma. Se utiliza para el acceso mediante SSL al servidor de firma.

4. Ejecutar el script "run.bat".

Esta Aplicación es una aplicación Java de referencia para integrar aplicaciones similares con el módulo de autenticación de @Firma. En principio presenta una pantalla como la que se indica a continuación:

Al seleccionar el botón "Autenticar" nos mostrará una ventana con los certificados que tenemos instalados en el sistema.



Después de seleccionar el certificado para autenticarnos en el Servidor @Firma, la aplicación se conectará con el mismo. El servidor comprobará que el certificado es válido y a continuación devolverá a la aplicación los datos del mismo para que los muestre por pantalla:

10 Códigos de Error en Autenticación

A continuación se incluyen una lista con los códigos de error en autenticación para su identificación:

- Error 0: Todo OK.
- Error 1: Error de conexión RMI contra @firma.
- Error 2: No se ha conseguido el certificado de usuario.
- Error 5: El certificado presentado no corresponde a ninguna CA admitida.
- Error 6: Error al obtener el Identificador o el NIF.
- Error 7: No existe NIF para este certificado.
- Error 8: Sesiones distintas en la página de entrada del usuario y la de la página a la que redirecciona el servidor.
- Error 9: Por motivos de seguridad, sólo se permite una entrada por sesión de navegador.
- Error 11: Algoritmo de firma en certificado de cliente invalido.
- Error 12: Clave en certificado cliente no valida.
- Error 13: Proveedor de certificado cliente no encontrado.
- Error 14: Error al comprobar la firma del certificado de cliente.
- Error 15: Error al obtener los campos obligatorios del certificado.
- Error 16: No se ha encontrado el certificado de la CA de este certificado.
- Error 17: No se ha construido el objeto de forma correcta.
- Error 18: Certificado expirado.
- Error 19: No se ha encontrado la CRL.
- Error 20: Error al descargar la CRL.
- Error 21: Error al verificar la CRL con el certificado de su CA.
- Error 22: No se ha encontrado el certificado de la CA de esta CRL.
- Error 24 hasta Error 32: Certificado revocado.
- Error 33: Este certificado no tiene puntos de distribución de CRLS.

- Error 34: Imposible validar el certificado con sus DP.