



RedAbogacía

ABOGACÍA ESPAÑOLA

Plataforma de Firma Digital

Tipo de Documento

Integración de servicios de firma y validación en entornos J2EE

Código de Documento

IntegracionPlataformaFirma

Versión

2.4

Paginas

52

Autor

Ana Vázquez Núñez

Cambios desde la última versión

Lista de Distribución

ITCGAE - Dep nformática Corporativa

Revisado por: David Barbáchano González

Aprobado por

Fernando Cuesta

Firma

Firma

Fecha

Fecha

CONTROL DE VERSIONES

Versión	Fecha	Autor	Descripción / Cambios Relevantes
1.0	26/02/2008	Ana Vázquez Núñez	Elaboración inicial del documento
1.1	23/04/2008		Ampliación del documento
1.10	26/09/2011		Actualización WSDL
1.11	12/12/2011		Modificaciones en la firma de correo en cliente
1.12	22/02/2012		Servicio de asociación de firmas de pdf (servidor-cliente)
1.13	05/06/2012		Modificaciones en la firma de correo en cliente y servidor
1.14	20/06/2012		Se añade que las direcciones de correo deben ir separadas por punto y coma
1.15	05/07/2012		Firma en cliente accede mediante webService.
1.2	23/04/2008		Ampliación del documento
1.3	24/03/2009		Cambios en la interfaz de solicitud del applet. Se incluye el navegador
1.4	21/09/2009		Ampliación del documento: Firma XML
1.5	19/10/2009		Rev Acrónimos y Definiciones – Rev Gráfico Arquitectura
1.6	30/11/2009		Ampliación del documento: Web Service de solicitud de Applet de firma
1.7	22/03/2010		Ampliación del documento: Applets con interfaz gráfica
1.8	28/06/2010		Integración de firma con dni-e y servicio de recuperación de pdf firmado y validado
1.9	16/12/2010		Integración de la plataforma de firma en entornos no J2EE
2.00	10/07/2012		Identificadores de procesos y de aplicaciones cambian.
2.1	06/08/2012		Se actualiza la integración del servicio de recuperación de un documento a partir del CSV
2.2	17/10/2012		Se actualiza la versión de las librerías core
2.3	14/11/2012		Actualización requisito versión Java
2.4	14/11/2016		Cambio de formato

INDICE

1.	Introducción.....	5
1.1.	OBJETIVO	6
1.2.	ALCANCE	6
1.3.	ABREVIATURAS Y DEFINICIONES.....	6
1.3.1.	Abreviaturas.....	6
1.3.2.	Definiciones	7
2.	REQUISITOS PREVIOS.....	8
2.1.	VERSIÓN JAVA.....	8
2.2.	ALTA APLICACIÓN CLIENTE.	8
2.3.	SERVICIOS DE FIRMA	8
2.4.	ALTA DE LA APLICACIÓN EN SERVICIOS DE FIRMA.	9
3.	INTEGRACION DE SERVICIOS	10
3.1.	INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE CORREOS	10
3.1.1.	Establecimiento de datos componentes del correo.....	10
3.1.2.	Solicitud de firma de correo corporativo a través del servicio del Webservice.....	12
3.1.2.1.	Tratamiento del resultado	13
3.2.	INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE FICHEROS PDF.....	13
3.2.1.	Establecimiento de datos componentes de la firma del pdf.....	14
3.2.2.	Solicitud de firma de pdf corporativo a través del servicio del Webservice	15
3.2.2.1.	Tratamiento del resultado	16
3.3.	INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE TEXTO	17
3.3.1.	Establecimiento de datos	17
3.3.2.	Solicitud de firma texto a través del servicio del Webservice.....	17
3.3.2.1.	Tratamiento del resultado	18
3.4.	INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA XML SIGNATURE	18
3.4.1.	Establecimiento de datos	19
3.4.2.	Solicitud de firma XML SIGNATURE a través del servicio del Webservice	19
3.4.2.1.	Tratamiento del resultado	20
3.5.	SOLICITUD DEL APPLLET DE FIRMA VÍA WEB SERVICE	20
3.6.	INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE TEXTO	23
3.6.1.	Establecimiento de datos necesarios para la validación de la firma de PDF.....	24
3.6.2.	Solicitud de validación de firma de texto a través del servicio del webservice.....	24
3.6.2.1.	Tratamiento del resultado	25
3.7.	INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE CORREO	25
3.7.1.	Establecimiento de datos necesarios para la validación de la firma de correo	25
3.7.2.	Solicitud de validación de firma de correo a través de un servicio del webservice	25
3.7.2.1.	Tratamiento del resultado	26
3.8.	INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE FICHERO PDF	26
3.8.1.	Establecimiento de datos necesarios para la validación de la firma de pdf.....	27
3.8.2.	Solicitud de validación de firma de pdf a través del servicio del webservice.....	27

3.8.2.1.	Tratamiento del resultado	28
3.9.	INTEGRACIÓN DEL SERVICIO DE VALIDACION DE CERTIFICADO	28
3.9.1.	Establecimiento de datos necesarios para la validación de la firma de correo	28
3.9.2.	Solicitud de validación de certificado de firma a través de un servicio de webservice..	29
3.9.2.1.	Tratamiento del resultado	30
3.10.	RECUPERACIÓN EN LOCAL DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE...	30
3.11.	RECUPERACIÓN DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE VÍA WEBSERVICE	31
3.11.1.	Solicitud del fichero de evidencia a través de un servicio de webservice	31
3.11.1.1.	Tratamiento del resultado	32
3.12.	ASOCIACIÓN DE FIRMAS VÍA WEBSERVICE.....	32
3.12.1.	Solicitud de asociación de firmas a través de un servicio de webservice.....	33
3.12.1.1.	Tratamiento del resultado	34
4.	INTEGRACIÓN DE LA PFD EN ENTORNOS NO J2EE	34
5.	ANEXOS.....	35
5.1.	PROCESOS DE FIRMA	35
5.2.	RESULTADO DEL PROCESO DE FIRMA	37
5.3.	RESULTADO DEL PROCESO DE VALIDACION DE FIRMA.....	37
5.4.	RESULTADO DEL PROCESO DE VALIDACION DE CERTIFICADO de FIRMA.....	38
5.5.	RESULTADO DEL PROCESO DE OBTENCIÓN DE DOCUMENTO	38
5.6.	RESULTADOS SOLICITUD APPLET FIRMA	38
5.7.	RESULTADOS A COMUNICAR A LA INTERFAZ JAVASCRIPT	38
5.8.	WSDL.....	39
5.9.	ACCESO HTTPS: librerías	48
5.10.	ACCESO WEBSERVICE: Librerías.....	49
5.11.	URLS DE ACCESO AL SERVICIO	49

1. INTRODUCCIÓN

En la dinámica diaria surgen múltiples situaciones en las que es necesario firmar los datos (documentos) que se están manipulando. Esta misma necesidad se produce también en procesos informáticos automatizados.

Para paliar esta situación, existe una plataforma unificada y centralizada de firma, que ofrece mediante una batería de servicios diferentes funcionalidades, adaptadas a las necesidades que surjan en cada momento.

Así mismo se consigue liberar a las aplicaciones de conocimientos de seguridad y criptografía, haciendo transparente el uso de servicios de firma ofrecidos.

La firma digital es un método criptográfico que asegura la identidad del remitente. En función del tipo de firma, puede, además, asegurar la integridad del documento o mensaje.

Existen varios tipos de firma:

- Simple: Identificación del firmante.
- Avanzada: garantiza la integridad de los datos.
- Reconocida: Es la firma avanzada ejecutada con un DSCF (dispositivo seguro de creación de firma) y amparada por un certificado reconocido (certificado que se otorga tras la verificación presencial de la identidad del firmante).

Desde la plataforma de firma se ofrecen servicios de firma reconocida.

La plataforma de firma forma parte de una arquitectura genérica de componentes que permiten ofrecer una batería de servicios a terceros.

La plataforma pone a disposición de los usuarios (aplicaciones y usuarios finales) la posibilidad de registrarse como clientes de los servicios que van a consumir. No se permite acceder a clientes no autorizados ni consumir servicios no autorizados, por lo que se ofrece una gestión de clientes y servicios de firma para controlar el acceso.

Las aplicaciones cliente tienen a su disposición una serie de servicios (webservices, applets, etc.) que les permite realizar firmas en los formatos soportados por la plataforma, así como validar los certificados usados para firmar y validar las firmas realizadas.

En la siguiente figura se muestra la arquitectura.

Arquitectura

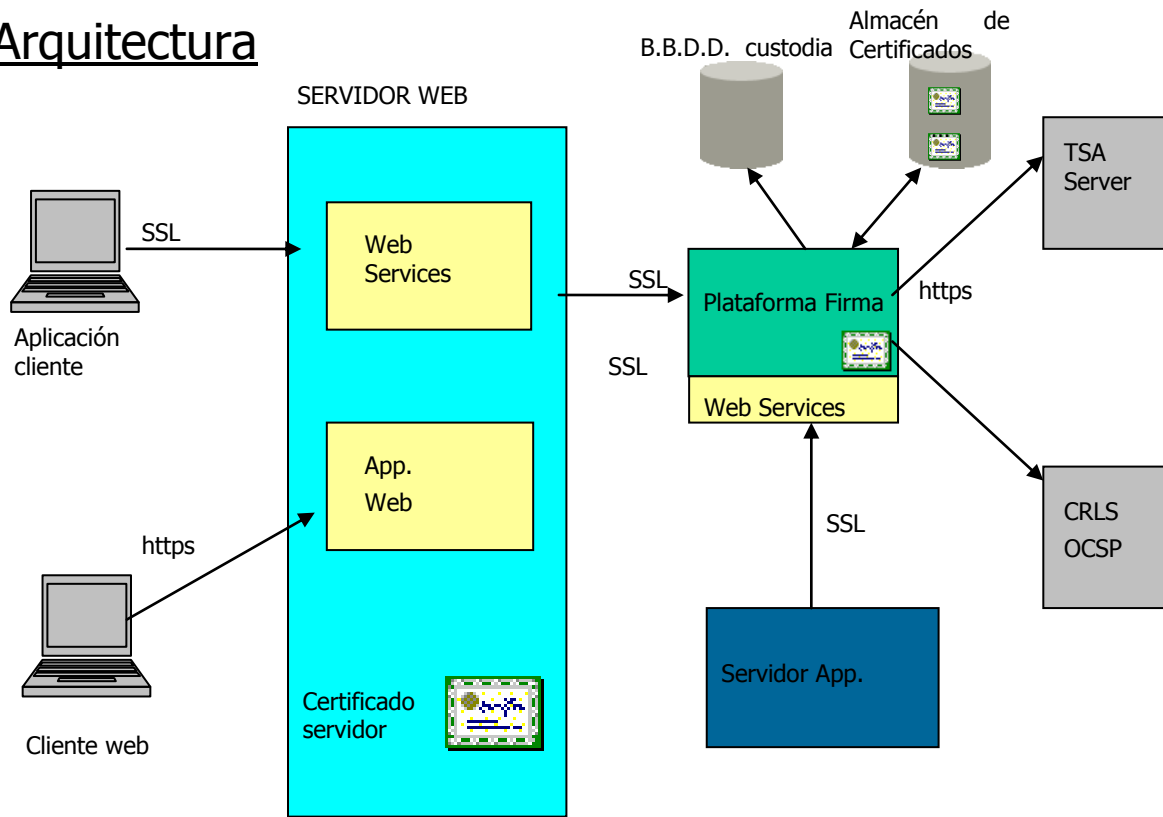


Ilustración 1. Arquitectura cliente servidor IHM.

1.1. OBJETIVO

El objetivo de este documento es definir los pasos para la integración de los diferentes servicios de firma ofrecidos por la plataforma en entornos J2EE.

1.2. ALCANCE

El documento va dirigido al CGAE con el fin de definir los pasos de integración de los diferentes servicios de firma dentro de las aplicaciones cliente en entornos J2EE.

1.3. ABREVIATURAS Y DEFINICIONES

1.3.1. Abreviaturas

ABREVIATURA	DESCRIPCIÓN
CGAE	Consejo General de la Abogacía Española.

CA	Autoridad de certificación
SSL	Security Socket Layer – Conexión segura cifrada
OCSP	Online Certificate Status Protocol – Servidor de validación de certificados
CRL	Certificate Revocation List – Lista de Certificados Revocados
TSA	Time Stamping Authority – Servidor de sellado de tiempo
PFD	Plataforma de firma digital

1.3.2. Definiciones

Concepto	Descripción
MODULO FUNCIONAL	Cada uno de los subsistemas, con funcionalidad propia, en que se subdivide el sistema principal (Portal Red Abogacía).
FUNCIONALIDAD	Cada una de las subdivisiones básicas e indivisibles de las que se compone un módulo funcional.

2. REQUISITOS PREVIOS

2.1. VERSIÓN JAVA

Los ficheros jar generados por la plataforma son compatibles con Java 1.6 y superior.

2.2. ALTA APLICACIÓN CLIENTE.

Para poder utilizar los servicios de la Plataforma es necesario que las aplicaciones cliente se den de alta en la misma y que se encuentren en estado de alta (A).

Siempre que un cliente solicite algún servicio de firma, la plataforma comprobará que la aplicación está autorizada (existe en la base de datos y está en estado de alta), y solo en caso afirmativo aceptará la petición.

La PFD facilitará a cada cliente el identificador para que puedan usarlo en las peticiones de servicios.

2.3. SERVICIOS DE FIRMA

La plataforma de firma ofrece los siguientes servicios de firma

- PDF_Corporativo
- JAR_Corporativo
- TEXTO_Cliente
- PDF_Remoto
- FICHERO_Cliente
- PDF_Local
- CORREO_Corporativo
- CORREO_Cliente
- TEXTO_Corporativo
- VALIDAR_Firma_Texto
- VALIDAR_Firma_PDF
- VALIDAR_Firma_Correo
- VALIDAR_Certificado
- XML_Corporativo
- XML_Cliente
- VALIDAR_Firma_XML
- VALIDAR_CERTIFICADO_FIRMA
- ASOCIAR_FIRMAS
- VALIDACION_WEB_FIRMA_PDF

La PFD facilitará a cada los identificaores de los servicios requeridos para que puedan usarlo en las peticiones de servicios.

2.4. ALTA DE LA APLICACIÓN EN SERVICIOS DE FIRMA.

Las aplicaciones cliente deberán darse de alta en los diferentes servicios de firma disponibles en la plataforma.

Los cliente podrán determinar si quieren dejar evidencia de la firma, en este caso solo tiene sentido si el proceso de firma tiene posibilidad de dejar evidencia (campo 'evidencia' igual a 1).

Si el proceso y el cliente requieren evidencia, la evidencia guardará la firma. Si solo el proceso requiere evidencia, se registrará los datos de la firma pero no la firma en si. Si el proceso no guarda evidencia, se ignorará el valor del cliente.

Se permite a las aplicaciones seleccionar el certificado de firma para cada uno de los procesos de firma corporativa a los que tengan acceso. Para ello se debe facilitar a la plataforma el certificado de firma en software y la contraseña, que la plataforma almacena en su sistema de ficheros y la referencia a su ubicación en la base de datos .

Para que una aplicación cliente pueda hacer uso de un servicio de firma, debe estar dada de alta en el sistema, estar la misma en estado de Alta, y debe tener asociado el servicio de firma solicitado y estar este servicio disponible (en estado de alta).

3. INTEGRACION DE SERVICIOS

En este apartado se indican los pasos a seguir para la integración de los diferentes servicios que facilita la PFD.

Para los servicios proporcionados via Web service se puede obviar el uso de las librerías facilitadas y generar el cliente de web service a partir del [WSDL](#).

3.1. INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE CORREOS

Para la firma corporativa de correos la plataforma ofrece a sus clientes una librería que facilita la integración en las aplicaciones clientes: **plataformaFirmaCore.x.x.x.jar**.

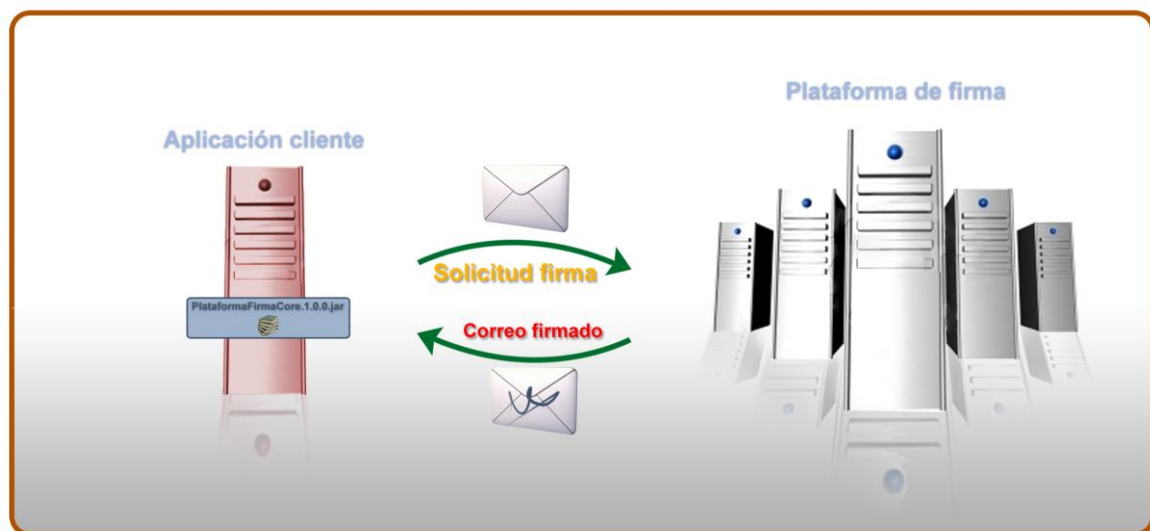


Ilustración 2: Firma corporativa de correo

3.1.1. Establecimiento de datos componentes del correo

Para la firma de correo corporativo, la aplicación cliente deberá facilitar los datos propios del correo:

- Texto: texto del mensaje. Se deben controlar los caracteres 'salto de línea' para que en el caso de que se recojan desde una campo textarea del formulario, se sustituyan por '
' haciendo uso de la función tratarSaltosLineaTextArea(campo) que se encuentra en el fichero 'comun.js'.
- Subject: subject del mensaje. En caso de no llegar nada, se pondrá el subject por defecto configurado en la plataforma.
- From: Este dato deberá ir en blanco, ya que para que no falle la verificación de la firma, el from debe ser el que aparezca como campo 'email' en el certificado de firma.
- To: lista de direcciones de correo separadas por ';'.
- CC: lista de direcciones de correo separadas por ';'.
- Bcc: lista de direcciones de correo separadas por ';'.

Se permite adjuntar ficheros en el correo, si es más de uno, deberán incluirse en un zip previamente, y será este fichero el que aparezca como 'attach' en el correo.

En el paquete **com.sis.firma.core.compresionFicheros** se encuentra una clase que facilita la compresión del fichero:

```
utilidades.comprimirFicheros(lista,result);
```

Donde 'lista' es un ArrayList de strings, conteniendo las rutas absolutas de los ficheros a incluir en el 'zip', y 'result' el nombre del fichero resultante con su ruta absoluta.

```
Ejemplo:  
import com.sis.firma.core.compresionFicheros;  
  
String file1= "c:\\Temp\\img15.png";  
String file2= "c:\\Temp\\img5.png";  
String file3= "c:\\Temp\\img4.png";  
  
ArrayList lista=new ArrayList<String>();  
  
lista.add(file1);  
lista.add(file2);  
lista.add(file3);  
  
String result="c:\\Temp\\ficheroZip.zip";  
  
utilidades.comprimirFicheros(lista,result);  
  
System.out.println("ficheros comprimidos");
```

En el caso de adjuntar ficheros al correo a firmar, se deberá facilitar el nombre del fichero comprimido, sin ruta, y el contenido del mismo en base 64.

```
import com.sis.firma.core.Base64;  
.....  
byte [] ficheroZip=.....  
String b64Attach=new String(Base64.encode(ficheroZip));
```

Una vez recopilada toda la información que compone el correo a firmar, se guardará en un objeto DatosEnvioCorreoTO (com.sis.firma.core.TO.DatosEnvioCorreoTO)

```
DatosEnvioCorreoTO datosEnvioCorreo=new DatosEnvioCorreoTO();

datosEnvioCorreo.setTextoCorreo(.....);
datosEnvioCorreo.setFrom(.....);
datosEnvioCorreo.setTo(.....);
datosEnvioCorreo.setCc(.....);
datosEnvioCorreo.setBcc(.....);
datosEnvioCorreo.setSubject (.....);

// Se indica el tipo de correo HTML o TEXTO_PLANO
// TipoCorreoFirmado.HTML
// TipoCorreoFirmado.TEXTO_PLANO
datosEnvioCorreo.setTipoCorreo((TipoCorreoFirmado.HTML));

//Se indica si la plataforma debe o no enviar el correo
// ConstantesCorreo.ENVIA_CORREO_PFD
// ConstantesCorreo.NO_ENVIA_CORREO_PFD
datosEnvioCorreo.setEnvio(ConstantesCorreo.....);

// Si hay attach se añade la info
if (HayAttach()){
    datosEnvioCorreo.setB64Attach(b64Attach);
    datosEnvioCorreo.
    setNombreFicheroAttach(nombreFicheroAttach);
}
else{
    datosEnvioCorreo.setB64Attach(null);
    datosEnvioCorreo.setNombreFicheroAttach(null);
}
```

3.1.2. Solicitud de firma de correo corporativo a través del servicio del Webservice.

Para realizar la solicitud de la firma corporativa de un correo es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma a través del webservice y recoge el resultado de la firma en un objeto FirmaResultadoTO . En esta clase hay varios métodos para llamar al applet de firma, de acuerdo a la información de partida:

```
public static FirmaResultadoTO solicitarFirmaCorporativaCorreo(String idCliente, DatosEnvioCorreoTO datosEnvioCorreo, String urlWSPFD) (NOTA 2)
```

```
public static FirmaResultadoTO solicitarFirmaCorporativaCorreo(String idCliente, DatosEnvioCorreoTO datosEnvioCorreo)
```

Ejemplo:

```
import com.sis.firma.coreServer.accesoWS;

try {

FirmaResultadoTO firma;
firma=accesoWS.solicitarFirmaCorporativaCorreo(idCliente,datosEnvioCorreo);

if (firma.getResultado().toUpperCase().equals(TipoResultadoFirma.FIRMA_OK))
{
// realizamos la tareas asociadas a firma correcta
Base64CODEC.decodeToFile(firma.getB64(), "firmaCorreo.eml");
}
else
{//hAY ERROR EN LA FIRMA DE CORREO
}

} catch (ProblemasComunicacionWSException e)
{
}
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso0 [ver Anexo 6.11](#)

3.1.2.1. Tratamiento del resultado

La plataforma devuelve un objeto FirmaResultadoTO, con cuatro campos. En el campo <resultado> se indica si ha habido problemas o si se ha realizado correctamente el proceso de firma.

En caso de que la firma se haya realizado correctamente el campo <B64> contienen el mensaje firmado (javax.mail.Message) en base 64.

El nodo <RESULTADO_ID_VALIDACION> se encuentra el código de validación asociado al fichero de custodia de firma, el cual servirá para acceder al fichero de evidencia de firma desde la interfaz web

El nodo <RESULT_RUTA> se encuentra la ruta de acceso al fichero de evidencia xml, util para recuperar los datos firmados en el caso de que la aplicación cliente tenga accesible el disco de la PFD. (Ver [RECUPERACIÓN DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE](#))

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE FIRMA](#).

3.2. INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE FICHEROS PDF

Para la firma corporativa de ficheros PDF la plataforma ofrece a sus clientes una librería que facilita la integración en las aplicaciones clientes: **plataformaFirmaCore.x.x.x.jar**.

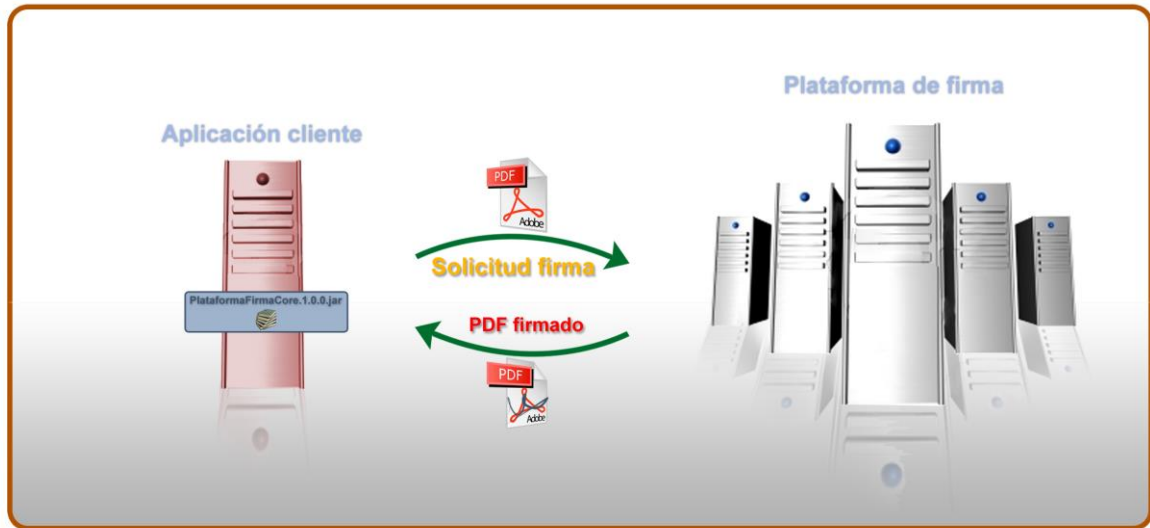


Ilustración 3: Firma corporativa de PDF

3.2.1. Establecimiento de datos componentes de la firma del pdf

Para la firma de corporativa de un fichero pdf, la aplicación cliente deberá facilitar los datos propios del sello de la firma:

- FechaFirma: fecha de la firma, si no se indica se pondrá la fecha y hora de la plataforma
- Location: indica la entidad o ubicación donde se realiza la firma. Por defecto Consejo General de la Abogacía Española.
- Razón: Texto libre indicando la razón de la firma.
- Visible: Indica si el sello se va a visualizar el documento PDF.

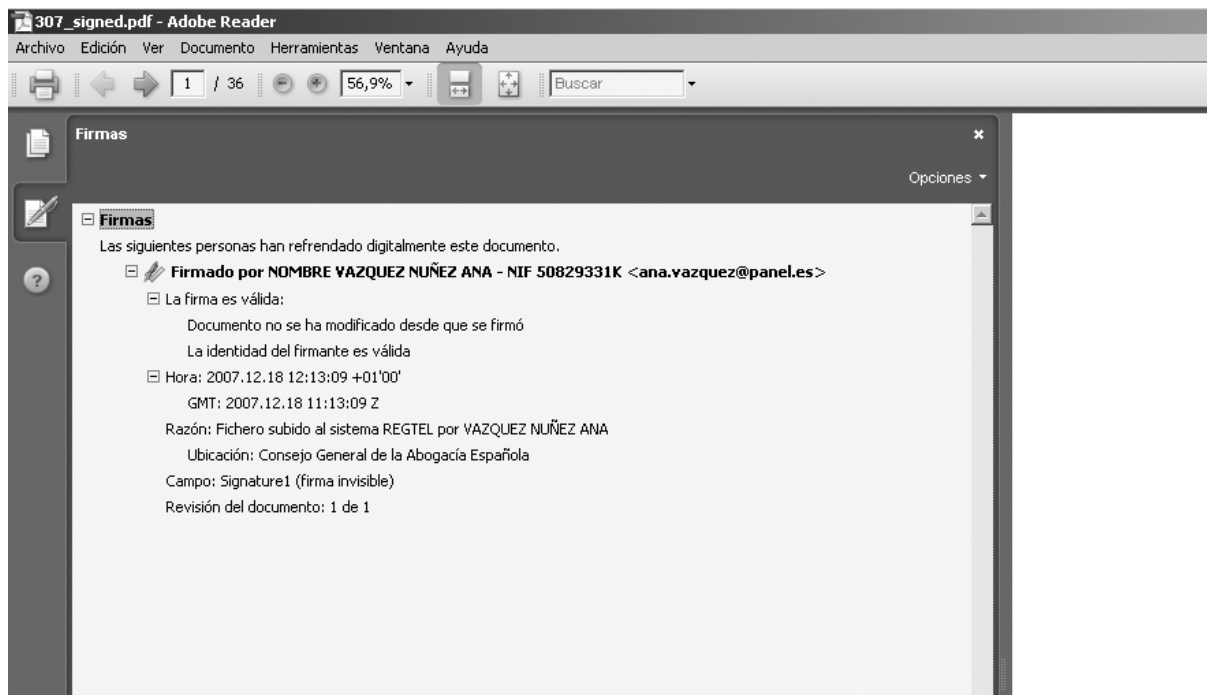


Ilustración 4: Sello de un pdf

Wed Dec 01 10:58:10 CET 2004

Firma válida
Digitally signed by NOMBRE VAZQUEZ NUÑEZ ANA - M...50829331K
Date: 2007.12.21 13:10:22 CET
Reason: Fichero registrado en el sistema REGTEL por VAZQUEZ NUÑEZ ANA
Location: Consejo General de la Abogacía Española

Ilustración 5: Sello visible de un pdf firmado

El fichero a firmar debe codificarse en base 64.

```
import com.sis.firma.core.pdf.util.UtilidadesPdf;  
String pdfB64=UtilidadesPdf.PdfToB64(rutaFichero);
```

También cabe la posibilidad de enviar la ruta local del fichero en lugar del contenido del mismo, esto solo tiene sentido cuando la aplicación cliente comparte rutas con el servidor de la plataforma de firma, en ese caso se indica la ruta del fichero a firmar y la ruta del fichero destino (pdf firmado)

```
import com.sis.firma.core.pdf.util.UtilidadesPdf;  
  
String urlorigen=".../.../.../fichero.pdf  
String urlDestino=".../.../.../ficheroFirmado.pdf
```

Una vez recopilada toda la información que compone el sello de la firma, se guardará en un objeto DatosSelloFirmaPDFTO (com.sis.firma.core.TO.DatosSelloFirmaPDFTO)

```
DatosSelloFirmaPDFTO datosSelloFirma =new DatosSelloFirmaPDFTO();  
  
//La firma puede verse o no  
// DatosSelloFirmaPDFTO.SELLO_PDF_NO_VISIBLE  
// DatosSelloFirmaPDFTO.SELLO_PDF_VISIBLE  
datosSelloFirma.setVisible(DatosSelloFirmaPDFTO.SELLO_PDF_NO_VISIBLE);  
datosSelloFirma.setRazon(".....");  
datosSelloFirma.setFechaFirma("...");  
datosSelloFirma.setLocation("...");
```

3.2.2. Solicitud de firma de pdf corporativo a través del servicio del Webservice

Para realizar la solicitud de la firma corporativa de un fichero pdf es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma a través del webservice y recoge el resultado de la firma en un objeto FirmaResultadoTO. Dentro de esta clase hay varios métodos para llamar al applet de firma, de acuerdo a la información de partida:

```
public static FirmaResultadoTO solicitarFirmaCorporativaPDF(String idCliente,String b64Pdf,  
DatosSelloFirmaPDFTO datosSelloFirma)
```

```
public static FirmaResultadoTO solicitarFirmaCorporativaPDF(String idCliente,String  
b64Pdf,DatosSelloFirmaPDFTO datosSelloFirma, String conCsv)
```

```
public static FirmaResultadoTO solicitarFirmaCorporativaPDF(String idCliente,String  
b64Pdf,DatosSelloFirmaPDFTO datosSelloFirma,String conCsv, String urlWSPFD) (NOTA 2)
```

Ejemplo:

```
import com.sis.firma.coreServer.accesoWS;  
  
FirmaResultadoTO firma;  
firma=accesoWS.solicitarFirmaCorporativaPDF(idCliente, firmaB64,datosSelloFirma);  
  
if (firma.getResultado().toUpperCase().equals(TipoResultadoFirma.FIRMA_OK))  
{  
    //tratamos la firma  
    Firma.getB64();  
}  
else  
{//HAY ERROR EN LA FIRMA DE CORREO  
}  
  
}catch (ProblemasComunicacionWSEException e){  
    e.printStackTrace();  
} catch (PDFKOEException e)  
{e.printStackTrace();  
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso0 [ver Anexo 6.11](#)

3.2.2.1. Tratamiento del resultado

La plataforma devuelve un objeto FirmaResultadoTO, con cuatro campos: resultado, b64, idValidacion y rutaFicheroEvidencia

En el campo <resultado> se indica si ha habido problemas o si se ha realizado correctamente el proceso de firma.

En caso de que la firma se haya realizado correctamente el campo <B64> contienen el fichero pdf firmado en base 64.

En el campo <idValidacion> se encuentra el código de validación asociado (CSV) al fichero de custodia de firma.

En el campo <rutaFicheroEvidencia> se encuentra la ruta de acceso al fichero de evidencia xml, util para recuperar los datos firmados en el caso de que la aplicación cliente tenga accesible el disco de la PFD. (Ver [RECUPERACIÓN DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE](#))

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE FIRMA](#).

3.3. INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA DE TEXTO

Para la firma corporativa de texto la plataforma ofrece a sus clientes una librería que facilita la integración en las aplicaciones clientes: **plataformaFirmaCore.x.x.x.jar**.

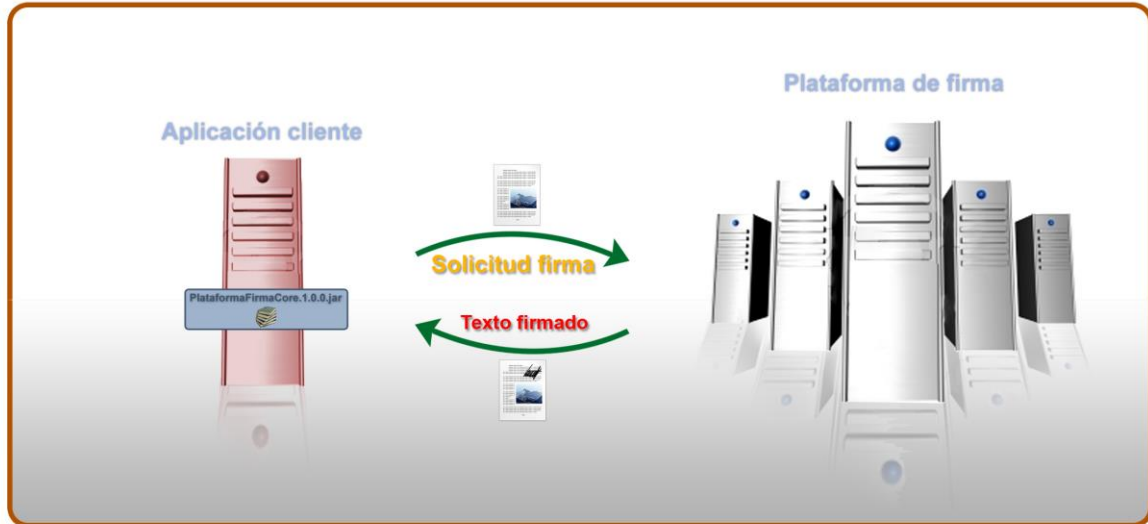


Ilustración 6: Firma corporativa de texto

3.3.1. Establecimiento de datos

Para la firma de texto corporativo, la aplicación cliente deberá facilitar el texto a firmar:

- Texto: texto a firmar

3.3.2. Solicitud de firma texto a través del servicio del Webservice

Para realizar la solicitud de la firma corporativa de un texto es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la firma en formato XML. En esta clase hay varios métodos para llamar al applet de firma, de acuerdo a la información de partida:

```
public static FirmaResultadoTO solicitarFirmaCorporativaTexto(String idCliente,String texto)
```

```
public static FirmaResultadoTO solicitarFirmaCorporativaTexto(String idCliente,String texto, String urlWSPFD) (NOTA 2)
```

Ejemplo:

```
import com.sis.firma.coreServer.accesoWS;  
  
//Llamamos al webservice  
try  
{  
    FirmaResultadoTO firma;
```

```
//el texto debe estar en b64
textoB64=new String(Base64.encode(texto.getBytes()));
firma=accesoWS.solicitarFirmaCorporativaTexto(idCliente, textoB64);
if (firma.getResultado().toUpperCase().equals(TipoResultadoFirma.FIRMA_OK))
{
}
else
{//HAY ERROR EN LA FIRMA DE CORREO
}
} catch (ProblemasComunicacionWSEException e)
{
e.printStackTrace();
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso0 [ver Anexo 6.11](#)

3.3.2.1. Tratamiento del resultado

La plataforma devuelve un objeto FirmaResultadoTO, con dos campos. En el campo < resultado> se indica si ha habido problemas o si se ha realizado correctamente el proceso de firma.

En caso de que la firma se haya realizado correctamente el campo <B64> contienen el fichero pdf firmado en base 64.

Los posibles valores del campo ‘resultado’ se muestran en el apartado [RESULTADO DEL PROCESO DE FIRMA](#).

3.4. INTEGRACIÓN DEL SERVICIO DE FIRMA CORPORATIVA XML SIGNATURE

Para la firma corporativa de xml (XML signatura) la plataforma ofrece a sus clientes una librería que facilita la integración en las aplicaciones clientes: **plataformaFirmaCore.x.x.x.jar**.

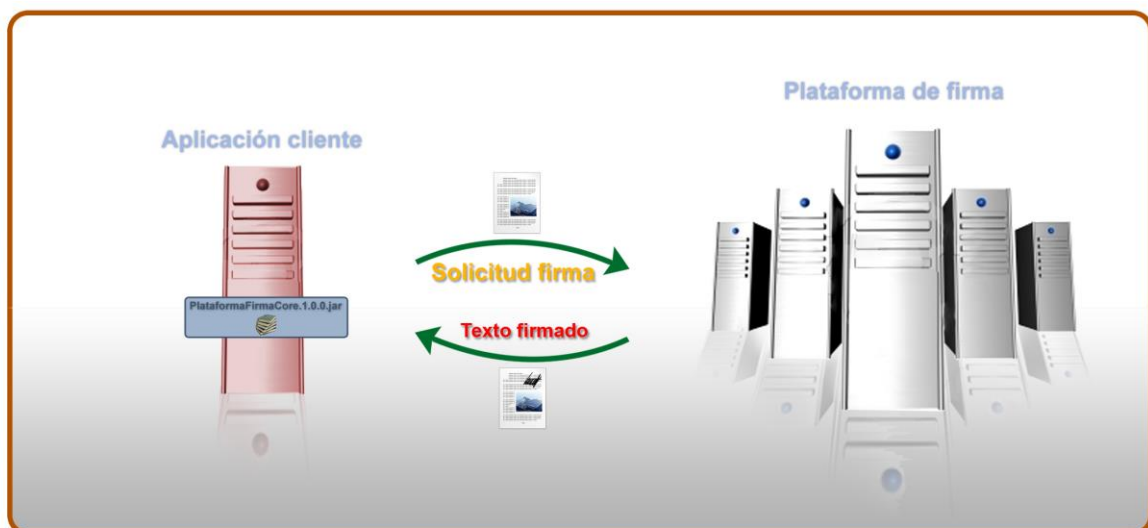


Ilustración 7: Firma corporativa de XML signature

3.4.1. Establecimiento de datos

Para la firma xml signature corporativo, la aplicación cliente deberá facilitar los datos a firmar:

- Tipo de firma: se soporta 4 tipos de firma: texto , texto XML, fichero, fichero XML
- datosB64: texto a firmar. En el caso de que el tipo sea de fichero o de fichero XML, será el contenido del fichero. Este dato deber codificarse en base 64

```
if(tipoDeFirma.equals("3") || tipoDeFirma.equals("4")){  
    //cojemos el contenido del fichero  
    datosB64= new String(Base64.encode(<contenidoFichero>));  
}  
else  
{  
    datosB64= new String(Base64.encode(texto.getBytes()));  
}
```

3.4.2. Solicitud de firma XML SIGNATURE a través del servicio del WebService

Para realizar la solicitud de la firma corporativa de un texto es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la firma en formato XML. En esta clase hay varios métodos para llamar al applet de firma, de acuerdo a la información de partida:

```
public static FirmaResultadoTO solicitarFirmaCorporativaXML(String idCliente,String textoB64,String tipoDeFirma)
```

```
public static FirmaResultadoTO solicitarFirmaCorporativaXML(String idCliente,String textoB64,String tipoDeFirma, String urlWSPFD) (NOTA 2)
```

Ejemplo:

```
import com.sis.firma.coreServer.accesoWS;  
  
//Llamamos al webservice  
try{  
    FirmaResultadoTO firma;  
  
    // el texto debe estar en b64  
    textoB64=new String(Base64.encode(texto.getBytes()));  
  
    accesoWS.solicitarFirmaCorporativaXML(idCliente, datosB64,tipoDeFirma);  
  
    if (firma.getResultado().toUpperCase().equals(TipoResultadoFirma.FIRMA_OK))  
    {  
        System.out.println("XML Signature :::>> " + new String (Base64.decode(firma.getB64()).getBytes()));  
    }  
}
```

```
else
  { //hAY ERROR EN LA FIRMA DE CORREO
  }

} catch (ProblemasComunicacionWSEException e)
{
  e.printStackTrace();
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso [ver Anexo 6.11](#)

3.4.2.1. Tratamiento del resultado

La plataforma devuelve un objeto FirmaResultadoTO, con dos campos. En el campo < resultado> se indica si ha habido problemas o si se ha realizado correctamente el proceso de firma.

En caso de que la firma se haya realizado correctamente el campo <B64> contienen el fichero pdf firmado en base 64.

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE FIRMA](#).

3.5. SOLICITUD DEL APLET DE FIRMA VÍA WEB SERVICE

La plataforma dispone de un web service para la solicitud del applet de firma, devolviendo la página html que contiene el applet codificada en base 64.

Los datos necesarios para la solicitud del applet son:

- Número de serie del certificado de firma: número de serie del certificado con el que se ha de firmar. Si no se indica el applet permitirá seleccionar entre los disponibles en la tarjeta.
- Identificador de cliente: identifica al cliente que solicita la firma
- Tipo de almacén: En este caso solo se admitirá la firma con certificados ubicados en tarjeta criptográfica. (TARJETA)
- Identificador del proceso de firma (Ver [PROCESOS DE FIRMA](#))
- Navegador

```
NAVEGADOR_FIREFOX = "FIREFOX";
NAVEGADOR_IE = "IE";
NAVEGADOR_UNDEFINED = "UNDEFINED";
```
- SO : sistema Operativo (1-> Window, 2-> Mac, 3-> MAC)

```
ListaParametrosApplet listaParametros=new ListaParametrosApplet();
listaParametros.addParametro(ConstantesSIS.PARAM_APPLET_ALMACEN_CERTIFICADOS,
TipoAlmacen.TARJETA);
listaParametros.addParametro(ConstantesSIS.PARAM_IDCLIENTE,CLIENTE_PRUEBAS);
listaParametros.addParametro(ConstantesSIS.PARAM_APPLET_PROCESO_FIRMA,
tipoProcesoFirma);
listaParametros.addParametro(ConstantesSIS.PARAM_APPLET_NUM_SERIE_CERT,numSerie);
listaParametros.addParametro(ConstantesSIS.PARAM_APPLET_NAVEGADOR,navegador);

String ua = request.getHeader("User-Agent");
int so= utilidades.getSistemaOperativo(ua.toUpperCase());

listaParametros.addParametro(ConstantesSIS.SISTEMA_OPERATIVO, so);
```

Dependiendo del proceso de firma a solicitar se deberán incluir, además, los siguientes parametros:

- **Firma de PDF**

- Razón: información a asociar a la firma. Si no se indica la PDF pone "FIRMA PDF ELECTRONICA";
- Visible: Indica si la firma se va a ver en el documento PDF o va a permanecer oculta
 - SELLO_PDF_VISIBLE=1;
 - SELLO_PDF_NO_VISIBLE=0;
- Location: Dato que indica donde se ha producido la firma. Si no se indica, la PFD pone: "CGAE"
- FormatoFecha: 'dd/mm/yyyy'
- fecha Firma: fecha de la firma en el formato indicado en el parámetro anterior. Si no se indica la PFD pone la fecha del sistema.
- Fichero a firma: ruta del fichero remoto a firmar
- Fichero de salida: ruta local del fichero firmado. Es opcional y en caso de no indicarse no se graba el pdf firmado en el cliente.

```
listaParametros.addParametro (ConstantesSIS.PARAM_APPLET_RAZON_FIRMA,datosSello.getRazon());

listaParametros.addParametro (ConstantesSIS.PARAM_APPLET_FIRMA_VISIBLE,Integer.toString
(datosSello.getVisible()));

listaParametros.addParametro
(ConstantesSIS.PARAM_APPLET_LOCATION_FIRMA,datosSello.getLocation());

listaParametros.addParametro
(ConstantesSIS.PARAM_APPLET_FORMATO_FECHA_FIRMA,datosSello.getFormatoFecha());

listaParametros.addParametro (ConstantesSIS.PARAM_APPLET_FECHA_FIRMA,
datosSello.getFechaFirmaStr() );

listaParametros.addParametro (ConstantesSIS.PARAM_APPLET_FECHA_FIRMA,
datosSello.getFechaFirmaStr() );

listaParametros.addParametro (ConstantesSIS. PARAM_APPLET_FICHERO_AFIRMAR, rutaPdfRemoto);
```

```
listaParametros.addParametro (ConstantesSIS. PARAM\_APPLET\_FICHERO\_SALIDA, rutaDestino);
```

- **Firma de Texto**

- Texto : texto a firmar.

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_TEXTO\_AFIRMAR, textoAfimar);
```

- **Firma de XML**

- XMLAFIRMAR: texto o nombre del fichero local a firmar.
- TIPO: inidica si los datos a firmar son texto (TEXTO), texto XML(XML), fichero(FICHERO) o un fichero xml (FICHERO_XML)
- Salvar:indica si el fichero firmado hay que salvarlo en el cliente

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_TEXTO\_AFIRMAR, "");
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_TIPO\_XML, "FICHERO_XML");
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_SALVAR\_FICHERO, "1");
```

- **Firma de correo**

- textoAfimar: texto del correo.
- listaAttach: lista de los ficheros a enganchar al correo.
- SUBJECT: asunto. Opcional.
- FROM: remitente. Opcional.
- TO: destinatario. Opcional; lista de direcciones separdos por ;.
- CC: con copia. Opcional; lista de direcciones separdos por ;.
- BCC: con copia oculta. Opcional; ; lista de direcciones separdos por ;.

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_TEXTO\_AFIRMAR, textoAfimar);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_ATTACH, attach);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_SUBJECT, subject);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_FROM, from);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_TO,to);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_CC,cc);
```

```
listaParametros.addParametro(ConstantesSIS.PARAM\_APPLET\_CORREO\_BCC,bcc);
```

Una vez establecidos los parámetros se llama al método que recubre la llamada al web service, usando una de estos dos métodos:

```
public static ResultSolAppletFirmaTO solicitarAppletFirma(String listaStrParametros) throws ProblemasComunicacionWSException
```

```
public static ResultSolAppletFirmaTO solicitarAppletFirma(String listaStrParametros, String urlWSPFD) throws ProblemasComunicacionWSException (NOTA 2)
```

```
ResultSolAppletFirmaTO firma = accesoWS.solicitarAppletFirma(listaParametros.toStringHashTable());
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

Se obtiene un objeto de tipo ResultSolAppletFirmaTO con los siguientes campos

- **resultado:** indica si se ha obtenido la página con el applet. Los posibles valores revueltos se muestran en RESULTADOS SOLICITUD APPLLET FIRMA
- **paginaAppletB64:** Si todo es correcto, se recibe la página html en base64.

```
if (firma.getResultado().toUpperCase().equals(TipoResultadoSolApplet.SOLICITUD_APPLET_OK)){
    String pkcs7= firma.getPaginaAppletB64();
    try{
        System.out.println("pagina 64 es: "+new String (Base64.base64ANormal(pkcs7.getBytes())));
    }
    catch (B64Exception e){
        e.printStackTrace();
    }
    session.setAttribute("PKCS7", pkcs7);
    return;
}
else{
    //hAY ERROR
    session.setAttribute("ERRORFIRMA", firma.getResultado().toUpperCase());
    return;
}
```

3.6. INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE TEXTO

La plataforma dispone de un servicio de validación de firma de texto. Los clientes que necesiten validar una firma podrán hacer uso del mismo utilizando el método [solicitarValidacionFirmaTexto](#) incluido en la librería **plataformaFirmaCore.x.x.x.jar**.

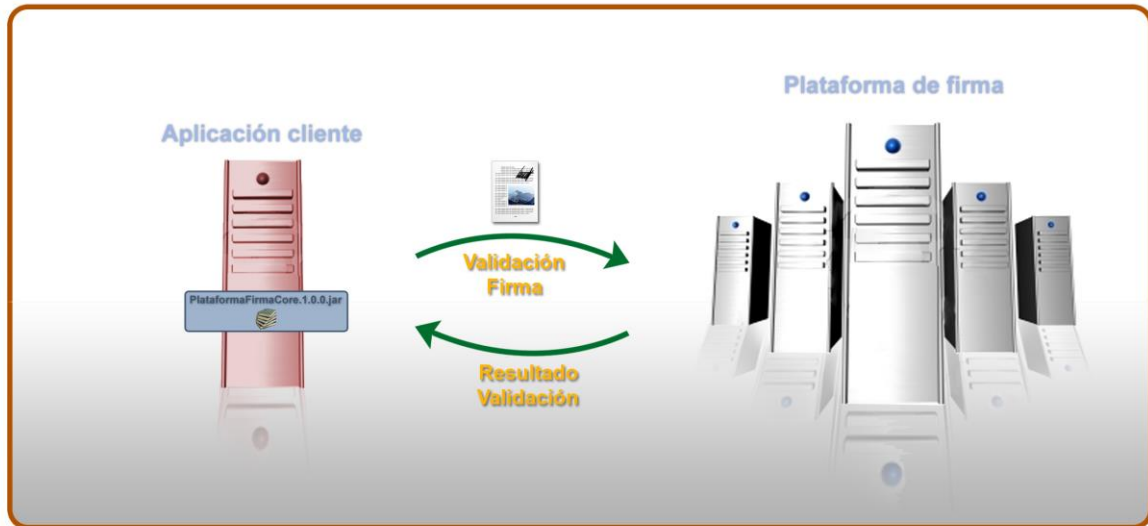


Ilustración 25: Validación de firma de texto

3.6.1. Establecimiento de datos necesarios para la validación de la firma de PDF

Para la validación de la firma de texto es necesario conocer:

- texto firmado: el texto fuente de la firma.
- PKCS7: pkcs7 conteniendo la firma del texto.

3.6.2. Solicitud de validación de firma de texto a través del servicio del webservice

Para realizar la solicitud de validación de firma de texto es necesario identificar la aplicación cliente. Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultValidacionFirmaTO . En esta clase hay dos métodos:

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaTexto(String idCliente,String texto, String firmaB64)
```

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaTexto(String idCliente, String texto, String firmaB64, String urlWSPFD) (NOTA 2)
```

```
import com.sis.firma.core.utilidades;

String texto="..." // Texto firmado
String firmaB64="..." // PKCS7

//Obtenemos la ruta de action dentro de la plataforma de firma
try {

    ResultValidacionFirmaTO result;
    String textoB64=new String(Base64.encode(texto.getBytes()));
    result=accesoWS.solicitarValidacionFirmaTexto(idCliente, textoB64, firmaB64);
    String resultado=result.getResultado();
} catch (ProblemasComunicacionWSEException e) {}
```


NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.6.2.1. Tratamiento del resultado

La plataforma devuelve un objeto ResultValidacionFirmaTO , con dos campos. En el campo < resultado > se indica si ha habido problemas o si se ha realizado correctamente el proceso de validación de firma.

El campo < informe > contiene un informe de la validación. En esta versión no se utiliza.

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE VALIDACION DE FIRMA](#)

3.7. INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE CORREO

La plataforma dispone de un servicio de validación de firma de correos. Los clientes que necesiten validar una firma podrán hacer uso del mismo utilizando el método [solicitarValidacionFirmaCorreo](#) incluido en la librería **plataformaFirmaCore.x.x.x.jar**.

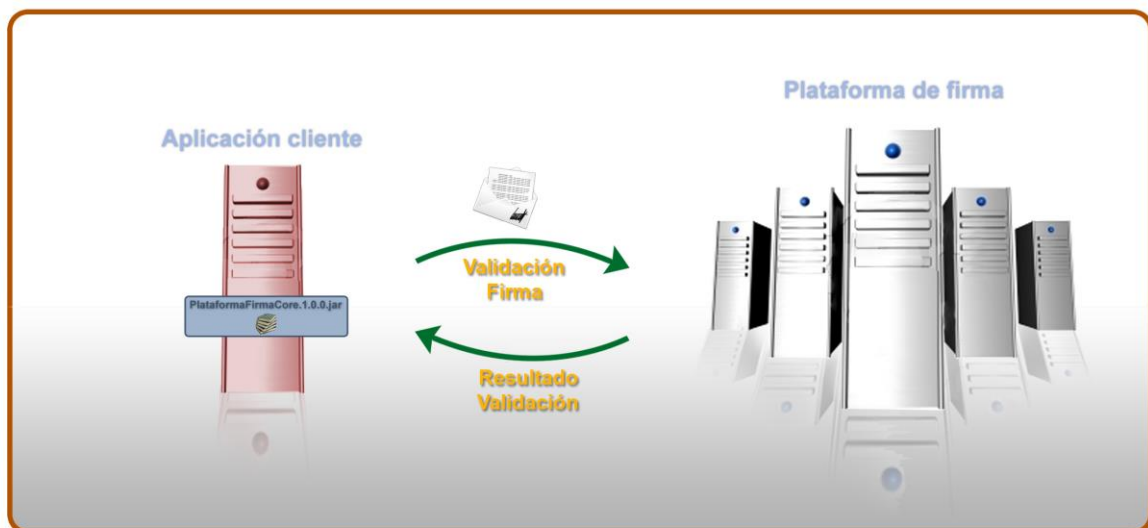


Ilustración 26: Validación de firma de correo

3.7.1. Establecimiento de datos necesarios para la validación de la firma de correo

Para la validación de la firma de correo es necesario conocer:

- correo firmado: Contenido del correo firmado.

3.7.2. Solicitud de validación de firma de correo a través de un servicio del webservice

Para realizar la solicitud de validación de firma de correo es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultValidacionFirmaTO . En esta clase hay varios métodos:

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaCorreo (String idCliente,String firmaB64)
```

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaCorreo (String idCliente, String firmaB64,  
String urlWSPFD) (NOTA 2)
```

```
import com.sis.firma.core.utilidades;  
  
String firmaB64="..." // Contenido del correo firmado en base64  
  
//Obtenemos la ruta de action dentro de la plataforma de firma  
ResultValidacionFirmaTO result;  
try{  
    result=accesoWS.solicitarValidacionFirmaCorreo(idCliente, firmaB64);  
    String resultado=result.getResultado();  
}  
catch (ProblemasComunicacionWSExeption e){  
    e.printStackTrace();  
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.7.2.1. Tratamiento del resultado

La plataforma devuelve un objeto ResultValidacionFirmaTO, con dos campos. En el campo < resultado > se indica si ha habido problemas o si se ha realizado correctamente el proceso de validación de firma.

El campo <informe> contiene un informe de la validación. En esta versión no se utiliza.

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE VALIDACION DE FIRMA](#)

3.8. INTEGRACIÓN DEL SERVICIO DE VALIDACION DE FIRMA DE FICHERO PDF

La plataforma dispone de un servicio de validación de firma de ficheros pdf. Los clientes que necesiten validar una firma podrán hacer uso del mismo utilizando el método [solicitarValidacionFirmaPDF](#) incluido en la librería **plataformaFirmaCore.x.x.x.jar**.

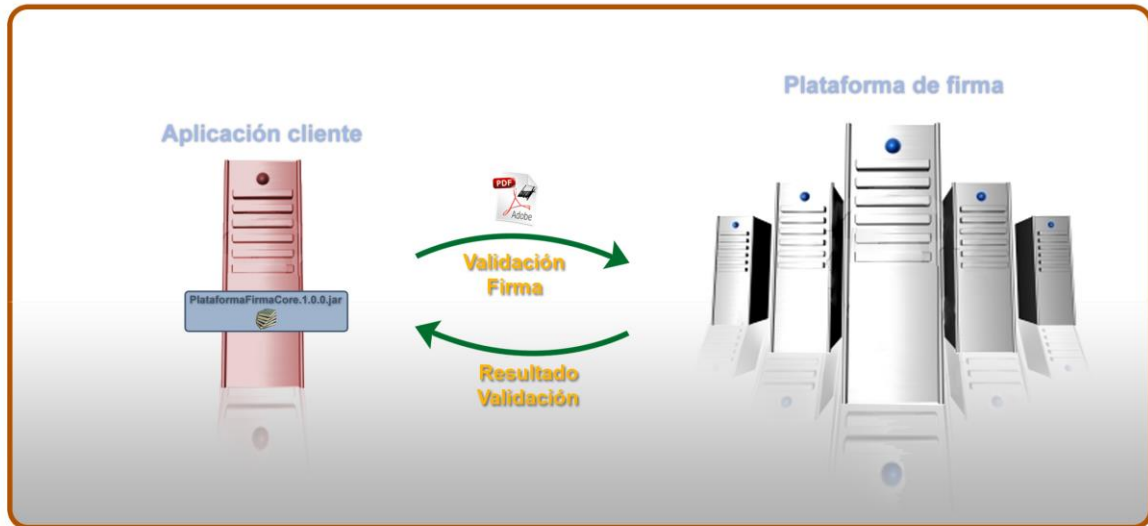


Ilustración 27: Validación de firma de fichero pdf

3.8.1. Establecimiento de datos necesarios para la validación de la firma de pdf

Para la validación de la firma de pdf es necesario conocer:

- pdf firmado: Contenido del fichero pdf firmado.

3.8.2. Solicitud de validación de firma de pdf a través del servicio del webservice

Para realizar la solicitud de validación de firma de correo es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultValidacionFirmaTO. En esta clase hay varios métodos:

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaPDF(String idCliente, String firmaB64)
```

```
public static ResultValidacionFirmaTO solicitarValidacionFirmaPDF(String idCliente, String firmaB64, String urlWSPFD) (NOTA 2)
```

```
import com.sis.firma.core.utilidades;

String firmaB64="..." // Contenido del pdf firmado en base64
ResultValidacionFirmaTO result;
try{
    result=accesoWS.solicitarValidacionFirmaPDF(idCliente, firmaB64);
    String resultado=result.getResultado();
}
catch(ProblemasComunicacionWSEException e)
{
    e.printStackTrace();
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.8.2.1. Tratamiento del resultado

La plataforma devuelve un objeto ResultValidacionFirmaTO, con dos campos. En el campo < resultado > se indica si ha habido problemas o si se ha realizado correctamente el proceso de validación de firma.

El campo < informe > contiene un informe de la validación. En esta versión no se utiliza.

Los posibles valores del campo 'resultado' se muestran en el apartado [RESULTADO DEL PROCESO DE VALIDACION DE FIRMA](#)

3.9. INTEGRACIÓN DEL SERVICIO DE VALIDACION DE CERTIFICADO

La plataforma dispone de un servicio de validación certificado de firma. Los clientes que necesiten validar una firma podrán hacer uso del mismo utilizando el método solicitarValidacionCertificadoFirma incluido en la librería **plataformaFirmaCore.x.x.jar**.

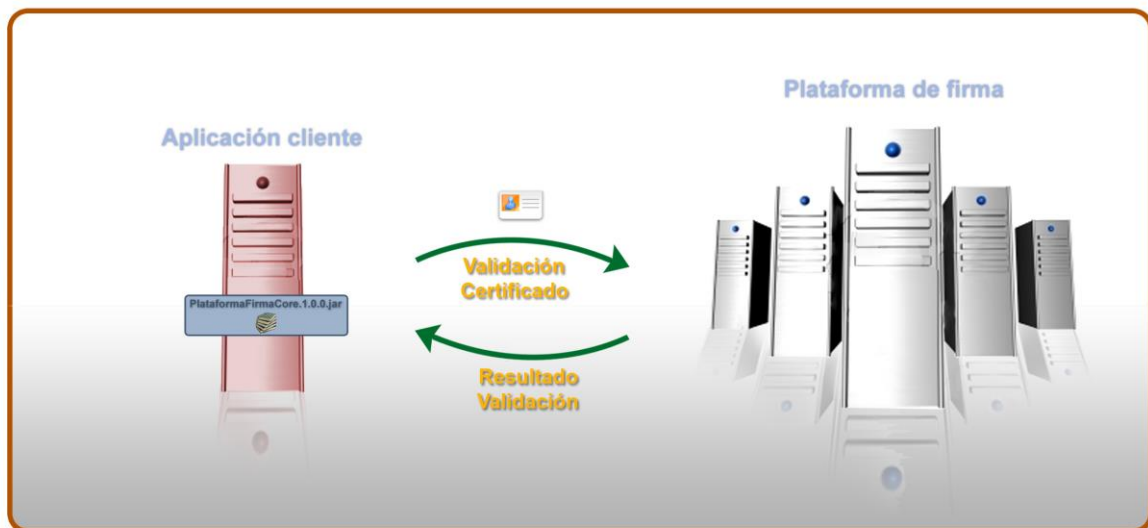


Ilustración 28: Validación de certificado de firma

3.9.1. Establecimiento de datos necesarios para la validación de la firma de correo

Para la validación de un certificado es necesario conocer:

- Identificador de la cliente.
- certificado: certificado X509 en base 64.
- validaciones: lista de las validaciones a realizar (TipoValidación) separadas por el carácter '#'.

Existen los siguientes tipos de validación:

- VIGENCIA: comprueba que el certificado está vigente.
- CADENA_CAS: comprueba que el certificado pertenece a unas CAS autorizadas en la PFD
- REVOCACION: comprueba que el certificado no está revocado

- **TOTAL:** realiza los tres tipos de validación anteriores.

Si se indica más de una validación, deberán venir separadas por el carácter '#'.

```
public class TipoValidacion {

    public static final String VALIDACION_VIGENCIA="VIGENCIA";
    public static final String VALIDACION_CADENA_CAS="CADENA_CAS";
    public static final String VALIDACION_REVOCACION="REVOCACION";
    public static final String VALIDACION_TOTAL="TOTAL";
    public static final String SEPARADOR="#";

}
```

3.9.2. Solicitud de validación de certificado de firma a través de un servicio de webservice

Para realizar la solicitud de validación de de certificado de firma es necesario identificar la aplicación cliente.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultValidacionCertificadoTO . En esta clase hay varios métodos:

```
public static ResultValidacionCertificadoTO solicitarValidacionCertificadoFirma(String idCliente,String
b64Certificado, String validaciones, String urlWSPFD) (NOTA 2)
```

```
public static ResultValidacionCertificadoTO solicitarValidacionCertificadoFirma(String idCliente,String
b64Certificado, String validaciones)
```

```
import com.sis.firma.core.utilidades;

.....

String certBase64="..." // Contenido del certificado de firma en base64
ResultValidacionCertificadoTO resultado
try {

    ResultValidacionCertificadoTO result;
    CertificadoFirma cf = new CertificadoFirma();
    cf.establecerValidacionTotal();
    result=accesoWS.solicitarValidacionCertificadoFirma(idCliente, certB64, cf.toStringTiposValidacion());
    resultado=result.getResultado();
    session.setAttribute("RESULTADO" , resultado);
}catch(ProblemasComunicacionWSExeption e)
{
    e.printStackTrace();
    session.setAttribute("RESULTADO","Problemas de acceso al proceso de validacion");
}catch (urlHttpNotFoundException e)
{
    e.printStackTrace();
}
```

```
public ResultValidacionCertificadoTO() {

infoValidacion = new InfoValidacionTO();
```

```
infoCertificado= new InfoCertificadoTO();
}

...

public class InfoValidacionTO implements java.io.Serializable {
    private java.lang.String tipo;

    private java.lang.String url;

    private java.lang.String tokenOCSP;

    public InfoValidacionTO() {
    }
}

...

public class InfoCertificadoTO implements java.io.Serializable {
    private java.lang.String certB64;

    private com.pfd.ws.validacionCertificado.CampoTO[] campos;

    public InfoCertificadoTO() {
    }
}

...
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.9.2.1. Tratamiento del resultado

La plataforma devuelve un objeto ResultValidacionCertificadoTO, donde en el campo <resultado> indica el resultado de la validación ([RESULTADO DEL PROCESO DE VALIDACION DE CERTIFICADO de FIRMA](#))

3.10. RECUPERACIÓN EN LOCAL DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE

Entre los datos recibidos como resultado de una operación de firma se recibe la ruta local del fichero xml de evidencia. Si el cliente de la PFD tiene accesible esta ruta podrá recuperarlo y tratar la información del mismo, haciendo uso de clases auxiliares facilitadas en la librería **plataformaFirmaCore.x.x.x.jar**

```
String rutaEvidenciaPDFxml=.....

NodeList listaNodos=obtenerListaNodos(rutaEvidenciaPDFxml);
AtributosXMLTO informacionXML=new AtributosXMLTO(listaNodos);

pdfB64Firmado=informacionXML.getFirmaB64();

informacionXML.getIdValidacion();
informacionXML.getCiente();
```

```
informacionXML.getProcesoFirma();  
informacionXML.getFecha();  
informacionXML.getResultado();  
informacionXML.getCN();  
informacionXML.getCertB64();  
informacionXML.getIdsAuditoriasAsociadas();
```

3.11. RECUPERACIÓN DEL FICHERO DE EVIDENCIA DESDE LA APLICACIÓN CLIENTE VÍA WEBSERVICE

Las aplicaciones externas, que no tengan posibilidad de acceder al disco directamente a través de la ruta recibida en la respuesta, pueden solicitar via webservice el documento firmado usando el servicio que facilita el fichero xml de custodia para que una vez en el cliente, este pueda extraer el documento firmado y la información relacionada con la firma, ayudandose de las utilidades incluidas en la librería **plataformaFirmaCore.x.x.x.jar** para tal fin.

3.11.1. Solicitud del fichero de evidencia a través de un servicio de webservice

Para realizar la solicitud del fichero de evidencia es necesario identificar la aplicación cliente (opcional) y el código de validación asociado al documento de firma custodiado.

Existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultSolicitudDocumentoTO.

Los métodos son:

```
public static ResultSolicitudDocumentoTO obtenerDocumento(SolicitudDocumentoTO solDoc, String url) throws ProblemasComunicacionWSEException (NOTA 2)
```

```
public static ResultSolicitudDocumentoTO obtenerDocumento(SolicitudDocumentoTO solDoc) throws ProblemasComunicacionWSEException
```

```
SolicitudDocumentoTO solDocTO = new SolicitudDocumentoTO();  
  
solDocTO.setIdAppCliente(idCliente);  
solDocTO.setIdValidacion(idValidacion);  
ResultSolicitudDocumentoTO obtDoc = accesoWS.obtenerDocumento(solDocTO);  
if (obtDoc != null) {  
    if (obtDoc.getResultado().toUpperCase().equals  
(TipoResultadoObtencionDoc.SOLICITUD_DOCUMENTO_OK))  
    {  
        IdValidacion=(obtDoc.getDocumento().getIdValidacion());  
        Aplicación=(obtDoc.getDocumento().getCliente());  
        ProcesoFirma=(obtDoc.getDocumento().getProcesoFirma());  
        Fecha=(obtDoc.getDocumento().getFecha());
```

```
Resultado=(obtDoc.getDocumento().getResultado());
Cn=(obtDoc.getDocumento().getCn());
?CertificadoB64=(obtDoc.getDocumento().getCertificadoB64().toString());
ContenidoPDFB64=(obtDoc.getDocumento().getFirmab64().toString());

}else if (obtDoc.getResultado().toUpperCase().equals
(TipoResultadoObtencionDoc.SOLICITUD_DOCUMENTO_KO)){

    .....
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.11.1.1. Tratamiento del resultado

La plataforma devuelve un objeto com.pfd.ws.validacionFirma.ResultSolicitudDocumentoTO donde en el campo <resultado> indica el resultado de la validación ([RESULTADO DEL PROCESO DE OBTENCIÓN DE DOCUMENTO](#)).

En el campo <documento>, com.pfd.ws.validacionFirma.DocumentoTO, se encuentra la información del documento, entre ellos el documento pdf docificado en base 64, por lo que deberá ser decodificado para su visualización o almacenamiento.

```
public class DocumentoTO implements java.io.Serializable {
    private java.lang.String idValidacion;

    private java.lang.String cliente;

    private java.lang.String procesoFirma;

    private java.lang.String fecha;

    private java.lang.String resultado;

    private java.lang.String cn;

    private java.lang.String certificadoB64;

    private java.lang.String firmab64;

    private java.lang.String idsAuditoriasAsociadas;
```

3.12. ASOCIACIÓN DE FIRMAS VÍA WEBSERVICE

Existe la posibilidad de asociar firmas cuando se realizan secuencialmente sobre el mismo documento, a partir de una firma inicial en servidor en la que se ha asignado un identificador de validación, de forma que el resto de las firmas se asocien al mismo identificador de validación, y el documento resultante esté asociado al identificador de validación inicial.

El servicio de asociación de firmas se pueden solicitar via webservice ayudandose de las utilidades incluidas en la librería **plataformaFirmaCore.x.x.jar** para tal fin.

3.12.1. Solicitud de asociación de firmas a través de un servicio de webservice

Para realizar la solicitud de asociación de firmas existe en la librería una clase de utilidades que realiza la solicitud a la plataforma y recoge el resultado de la validación en un objeto ResultSolicitudDocumentoTO. Los métodos son:

```
public static ResultadoAsociacionFirmaTO asociarFirmas(DatosAsociacionFirmaTO datos)
```

```
public static ResultadoAsociacionFirmaTO asociarFirmas(DatosAsociacionFirmaTO datos, String url) (NOTA 2)
```

Los datos de entrada son:

- Pdf64firmado: fichero pdf firmado
- idAuditoria: identificador de la auditoria correspondiente a la última firma realizada en cliente.
- idCliente: identificador de cliente.
- idValidación: identificador de validación de la firma inicial (firma en servidor) a la que asociar la firma identificada por el identificador de auditoria.

```
DatosAsociacionFirmaTO datos = new DatosAsociacionFirmaTO();
    datos.setFirmaPFDb64(pdfB64Firmado);
    datos.setIdAuditoriaFirmaCliente(idAuditoria);
    datos.setIdCliente(idCliente);
    datos.setIdValidacion(idValidacion);

    ResultadoAsociacionFirmaTO resultado = accesoWS.asociarFirmas(datos, url);
    resultado.getDescResultado();
    log.debug("resultado de asociacion firmas "+resultado.getDescResultado());

    if(resultado.getIdResultado().equals(TipoResultadoFirma.FIRMA_OK)
        && resultado.getDescResultado().equals("")){
        getContext().getMessages().add(new SimpleMessage("Asociacion de firmas correcta
"));
    }else{
        new Exception(resultado.getIdResultado()+" "+resultado.getDescResultado());
    }
}
```

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

3.12.1.1. Tratamiento del resultado

La plataforma devuelve un objeto ResultadoAsociacionFirmaTO, donde en el campo <resultado> indica el resultado de la asociación de firmas ([RESULTADO DEL PROCESO DE FIRMA](#)):

- *FIRMA_OK*
- *ASOCIACION_FIRMAS_KO*
- *RECUPERACION_CON_IDVALIDACION_KO*
- *PROCESO_FIRMA_NO_AUTORIZADO*
- *CLIENTE_NO_AUTORIZADO*
- *PROCESO_FIRMA_NO_DISPONIBLE*

4. INTEGRACIÓN DE LA PFD EN ENTORNOS NO J2EE

Las librerías que ofrece la PFD está diseñadas para entornos J2EE por lo fuera de estos entornos no se garantiza su funcionamiento, debido a las dependencias entre librerías.

Si se desea hacer uso de los servicios de la Plataforma de Firma desde aplicaciones no incluidas en entornos J2EE, se puede utilizar la librería **web-services-clientePFD.jar**, en el caso de aplicaciones Java, o bien generar un cliente del webservice a partir del esquema [WSDL](#), esto último es lo más recomendable.

Se deberá tener en cuenta que la solicitud de servicios a la PFD se realiza via SSL, por lo que es necesario que la aplicación cliente se configure para que tenga como certificado de confianza el certificado SSL del servidor de la plataforma de firma.

NOTA 1 : Librerías a incluir [ver Anexo 6.10](#)

NOTA 2: Urls de acceso: [ver Anexo 6.11](#)

5. ANEXOS

5.1. PROCESOS DE FIRMA

```
public class TipoProcesoFirma {

    /** Identificador del proceso de firma corporativa de un pdf */
    public static final String FIRMA_PDF_CORPORATIVO="1";
    /** Descripción del proceso de firma corporativa de un pdf */
    public static final String FIRMA_PDF_CORPORATIVOSTR="PDF_Corporativo";

    /** Identificador del proceso de firma corporativa de un jar */
    public static final String FIRMA_JAR_CORPORATIVO="2";

    /** Descripción del proceso de firma corporativa de un jar */
    public static final String FIRMA_JAR_CORPORATIVOSTR="JAR_Corporativo";

    /** Identificador del proceso de firma de texto en cliente */
    public static final String FIRMA_TEXTO="3";
    /** Descripción del proceso de firma de texto en cliente */
    public static final String FIRMA_TEXTOSTR="TEXTO_Cliente";

    /** Identificador del proceso de firma de pdf remoto en cliente */

    public static final String FIRMA_PDF_REMOTO="4";
    /** Descripción del proceso de firma de pdf remoto en cliente */
    public static final String FIRMA_PDF_REMOTOSTR="PDF_Remoto";

    /** Identificador del proceso de firma de un fichero local en cliente */
    public static final String FIRMA_FICHERO_CLIENTE="5";

    /** Descripción del proceso de firma de un fichero local en cliente */
    public static final String FIRMA_FICHERO_CLIENTESTR="FICHERO_Cliente";

    /** Identificador del proceso de firma de pdf local en cliente */
    public static final String FIRMA_PDF_LOCAL="6";
    /** Descripción del proceso de firma de pdf local en cliente */
    public static final String FIRMA_PDF_LOCALSTR="PDF_Local";
```

```
/** Identificador del proceso de firma de correo corporativo */
public static final String FIRMA_CORREO_CORPORATIVO="7";
/** Descripcion del proceso de firma de correo corporativo */
public static final String FIRMA_CORREO_CORPORATIVOSTR="CORREO_Corporativo";
/** Identificador del proceso de firma de correo en cliente */
public static final String FIRMA_CORREO_CLIENTE = "8";
/** Descripcion del proceso de firma de correo en cliente */
public static final String FIRMA_CORREO_CLIENTESTR = "CORREO_Cliente";
public static final String FIRMA_TEXTO_CORPORATIVO = "9";
public static final String FIRMA_TEXTO_CORPORATIVO_STR = "TEXTO_Corporativo";

public static final String VALIDAR_FIRMA_TEXTO = "10";
public static final String VALIDAR_FIRMA_TEXTO_STR = "VALIDAR_Firma_Texto";
public static final String VALIDAR_FIRMA_PDF = "11";
public static final String VALIDAR_FIRMA_PDF_STR = "VALIDAR_Firma_PDF";
public static final String VALIDAR_FIRMA_CORREO = "12";
public static final String VALIDAR_FIRMA_CORREO_STR = "VALIDAR_Firma_Correo";

public static final String VALIDAR_CERTIFICADO = "13";
public static final String VALIDAR_CERTIFICADO_STR = "VALIDAR_Certificado";

public static final String VALIDAR_FIRMA_WEB_PDF = "100";
public static final String VALIDAR_FIRMA_WEB_PDF_STR = "VALIDAR_Firma_Web_PDF";

public static final String FIRMA_XML_CORPORATIVO = "14";
public static final String FIRMA_XML_CORPORATIVO_STR = "XML_Corporativo";

public static final String FIRMA_XML_CLIENTE = "15";
public static final String FIRMA_XML_CLIENTE_STR = "XML_Cliente";

public static final String VALIDAR_FIRMA_XML = "16";
public static final String VALIDAR_FIRMA_XML_STR = "VALIDAR_Firma_XML";
public static final String VALIDAR_CERTIFICADO_FIRMA = "17";
public static final String VALIDAR_CERTIFICADO_FIRMA_STR = "VALIDAR_CERTIFICADO_FIRMA";

public static final String ASOCIAR_FIRMAS = "18";
public static final String ASOCIAR_FIRMAS_STR = "ASOCIAR_FIRMAS";

}
```

5.2. RESULTADO DEL PROCESO DE FIRMA

```
public class TipoResultadoFirma {  
  
    public static final String FIRMA_OK="FIRMA_OK";  
    public static final String CLIENTE_NO_ESPECIFICADO="CLIENTE_NO_ESPECIFICADO";  
    public static final String TEXTO_NO_ESPECIFICADO="TEXTO_NO_ESPECIFICADO";  
    public static final String PROCESO_FIRMA_NO_AUTORIZADO =  
"PROCESO_FIRMA_NO_AUTORIZADO";  
    public static final String CLIENTE_NO_AUTORIZADO = "CLIENTE_NO_AUTORIZADO";  
    public static final String PROCESO_FIRMA_NO_DISPONIBLE = "PROCESO_FIRMA_NO_DISPONIBLE";  
    public static final String FIRMA_KO = "FIRMA_KO";  
    public static final String FIRMA_KO_FILE_NO_PDF = "FIRMA_KO_FILE_NO_PDF";  
    public static final String FIRMA_KO_FORMATO_FECHA = "FIRMA_KO_FORMATO_FECHA";  
    public static final String FIRMA_CERTIFICADO_CADUCADO = "FIRMA_CERTIFICADO_CADUCADO";  
    public static final String FIRMA_CERTIFICADO_CASKO = "FIRMA_CERTIFICADO_CASKO";  
    public static final String FIRMA_CERTIFICADO_REVOCADO = "FIRMA_CERTIFICADO_REVOCADO";  
    public static final String FIRMA_CERTIFICADO_AUN_SIN_VIGENCIA =  
"FIRMA_CERTIFICADO_AUN_SIN_VIGENCIA";  
    public static final String FIRMA_CRLS_KO = "CRLS_KO";  
    public static final String CERTIFICADO_NOT_FOUND="CERTIFICADO_NOT_FOUND";  
    public static final String PTE_RESULTADO = "PTE_RESULTADO";  
    public static final String FIRMA_CERTIFICADO_NO_VERIFICADO =  
"FIRMA_CERTIFICADO_NO_VERIFICADO";  
    public static final String FIRMA_CERTIFICADO_KO="FIRMA_CERTIFICADO_KO";  
    public static final String  
FIRMA_FICHERO_MAX_TAMANO_KO="FIRMA_FICHERO_MAX_TAMANO_KO";  
    public static final String FIRMA_KO_FILE_NO_EXIST = "FIRMA_KO_FILE_NO_EXIST";  
    public static final String FIRMA_KO_FILE_DESTINO_NO_EXIST =  
"FIRMA_KO_FILE_DESTINO_NO_EXIST";  
    public static final String FIRMA_KO_TIPO_DE_FIRMA = "FIRMA_KO_TIPO_DE_FIRMA";  
    public static final String PDF_VERY_LARGE="PDF_VERY_LARGE";  
  
    public static final String  
RECUPERACION_CON_IDVALIDACION_KO="RECUPERACION_CON_IDVALIDACION_KO";  
    public static final String ASOCIACION_FIRMAS_KO = "ASOCIACION_FIRMAS_KO";  
}
```

5.3. RESULTADO DEL PROCESO DE VALIDACION DE FIRMA

```
public class TipoResultadoValidacionFirma {  
  
    public static final String FIRMA_OK="FIRMA_OK";  
    public static final String FIRMA_CERTIFICADO_CADUCADO="CERTIFICADO_CADUCADO";  
    public static final String FIRMA_CERTIFICADO_REVOCADO="CERTIFICADO_REVOCADO";  
    public static final String FIRMA_CERTIFICADO_CASKO="CERTIFICADO_CASKO";  
    public static final String FIRMA_CERTIFICADO_AUN_SIN_VIGENCIA = "CERTIFICADO_AUN_SIN_VIGENCIA";  
    public static final String FIRMA_CRLS_KO = "CRLS_KO";  
    public static final String FIRMA_KO="FIRMA_KO";  
    public static final String CLIENTE_NO_AUTORIZADO = "CLIENTE_NO_AUTORIZADO";  
    public static final String PROCESO_NO_AUTORIZADO = "PROCESO_NO_AUTORIZADO";  
    public static final String VALIDACION_NO_DISPONIBLE = "VALIDACION_NO_DISPONIBLE";  
}
```

}

5.4. RESULTADO DEL PROCESO DE VALIDACION DE CERTIFICADO DE FIRMA

```
public class TipoResultadoValidacionCertificado {  
  
    public static final String CERTIFICADO_OK="CERTIFICADO_OK";  
    public static final String CERTIFICADO_CADUCADO="CERTIFICADO_CADUCADO";  
    public static final String CERTIFICADO_REVOCADO="CERTIFICADO_REVOCADO";  
    public static final String CERTIFICADO_CASKO="CERTIFICADO_CASKO";  
    public static final String CERTIFICADO_AUN_SIN_VIGENCIA = "CERTIFICADO_AUN_SIN_VIGENCIA";  
    public static final String CRLS_KO = "CRLS_KO";  
}
```

5.5. RESULTADO DEL PROCESO DE OBTENCIÓN DE DOCUMENTO

```
public static final String SOLICITUD_DOCUMENTO_OK = "SOLICITUD_DOCUMENTO_OK";  
    public static final String SOLICITUD_DOCUMENTO_KO = "SOLICITUD_DOCUMENTO_KO";  
    public static final String URL_HTTP_NO_ENCONTRADA = "URL_HTTP_NO_ENCONTRADA";
```

5.6. RESULTADOS SOLICITUD APLET FIRMA

```
public class TipoResultadoSolApplet {  
  
    public static final String SOLICITUD_APPLET_OK = "SOLICITUD_APPLET_OK";  
    public static final String SOLICITUD_APPLET_KO = "SOLICITUD_APPLET_KO";  
    public static final String URL_HTTP_NO_ENCONTRADA = "URL_HTTP_NO_ENCONTRADA";  
}
```

5.7. RESULTADOS A COMUNICAR A LA INTERFAZ JAVASCRIPT

```
public class TipoResultadoValidacionCertificado {  
  
    public static final String CERTIFICADO_FOUND = "CERTIFICADO_FOUND";  
    public static final String CERTIFICADO_NOT_FOUND = "CERTIFICADO_NOT_FOUND";  
    public static final String NO_DLL="NO_DLL";  
    public static final String DDL_YA_CARGADA="DDL_YA_CARGADA";  
    public static final String NO_TARJETA="NO_TARJETA";  
    public static final String TARJETA_INCORRECTA="TARJETA_INCORRECTA";  
    public static final String NO_TOKEN="NO_TOKEN";  
    public static final String PIN_TARJETA_INCORRECTA = "PIN_TARJETA_INCORRECTA";  
    public static final String ERROR_FIRMA = "ERROR_FIRMA";  
    public static final String PIN_TARJETA_BLOQUEADA = "PIN_TARJETA_BLOQUEADA";  
    public static final String FICHERO_HTTP_NOT_FOUND="FICHERO_HTTP_NOT_FOUND";  
    public static final String FICHERO_NOT_FOUND="FICHERO_NOT_FOUND";  
    public static final String URL_HTTP_NOT_FOUND_EXCEPTION = "URL_HTTP_NOT_FOUND_EXCEPTION";  
    public static final String UNSUPPORTED_PROTOCOL="UNSUPPORTED_PROTOCOL";
```

```

public static final String CERTIFICADO_CADUCADO="CERTIFICADO_CADUCADO";
public static final String CERTIFICADO_NO_ACEPTADO="CERTIFICADO_NO_ACEPTADO";
public static final String VERIFICACION_NO_POSIBLE="VERIFICACION_NO_POSIBLE";
public static final String CERTIFICADO_REVOCADO="CERTIFICADO_REVOCADO";
public static final String CERTIFICADO_AUN_SINVIGENCIA="CERTIFICADO_AUN_SINVIGENCIA";
public static final String CERTIFICADO_KO="CERTIFICADO_KO";
public static final String FORMATO_FECHA_FIRMA_KO="FORMATO_FECHA_FIRMA_KO";
public static final String OUT_OF_MEMORY="OUT_OF_MEMORY";
public static final String ACCESOALMACEN_KO = "ACCESOALMACEN_KO";
public static final String CORREO_SIN_INFO = "CORREO_SIN_INFO";
public static final String ERROR_BORRAR_CTRLVERSIONPFD = "ERROR_BORRAR_CTRLVERSIONPFD";
public static final String OK="OK";
public static final String KO="KO";
public static final String WORKING="WORKING";
public static final String XML_PARSER_KO = "XML_PARSER_KO";
public static final String ERROR_SISTEMA = "ERROR_SISTEMA";
}

```

5.8. WSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://ServiciosPFD.service" xmlns:tns1="http://validacionCertificado.ws.pfd.com"
xmlns:tns2="http://validacionFirma.ws.pfd.com" xmlns:tns3="http://firma.ws.pfd.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="ServiciosPFDService" targetNamespace="http://ServiciosPFD.service">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://validacionCertificado.ws.pfd.com">
      <xsd:complexType name="validarCertificadoTO">
        <xsd:sequence>
          <xsd:element name="idCliente" type="xsd:string"/></xsd:element>
          <xsd:element name="certBase64" type="xsd:string"/></xsd:element>
          <xsd:element name="validacion" type="xsd:string"/></xsd:element>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="resultValidacionCertificadoTO">
        <xsd:sequence>
          <xsd:element name="resultado" type="xsd:string"/></xsd:element>
          <xsd:element name="infoValidacion"
type="tns1:infoValidacionTO" maxOccurs="1" minOccurs="0">
            </xsd:element>
          <xsd:element name="infoCertificado"
type="tns1:infoCertificadoTO" maxOccurs="1" minOccurs="0">
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      <xsd:complexType name="infoValidacionTO">
        <xsd:sequence>

```

```

        <xsd:element name="tipo" type="xsd:string" maxOccurs="1"
            minOccurs="1">
        </xsd:element>
        <xsd:element name="url" type="xsd:string" maxOccurs="1"
            minOccurs="0">
        </xsd:element>
        <xsd:element name="tokenOCSP" type="xsd:string" maxOccurs="1"
minOccurs="0"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="infoCertificadoTO">
    <xsd:sequence>
        <xsd:element name="certB64" type="xsd:string" maxOccurs="1"
            minOccurs="0">
        </xsd:element>

        <xsd:element name="campos" type="tns1:campoTO"
            maxOccurs="unbounded" minOccurs="0">
        </xsd:element>

    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="campoTO">
    <xsd:sequence>
        <xsd:element name="shortName" type="xsd:string"></xsd:element>
        <xsd:element name="oid" type="xsd:string"></xsd:element>
        <xsd:element name="descripcion" type="xsd:string"></xsd:element>
        <xsd:element name="valor" type="xsd:string" maxOccurs="1" minOccurs="0"
nillable="true" ></xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://validacionFirma.ws.pfd.com">
    <xsd:complexType name="validarFirmaCorreoTO">
        <xsd:sequence>
            <xsd:element name="idCliente" type="xsd:string"></xsd:element>
            <xsd:element name="emIB64" type="xsd:string"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="resultValidacionFirmaTO">
        <xsd:sequence>
            <xsd:element name="resultado" type="xsd:string"></xsd:element>

```



```

        <xsd:element name="informe" type="xsd:string"></xsd:element>
    </xsd:sequence>
</xsd:complexType></xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://validacionFirma.ws.pfd.com">
    <xsd:complexType name="validarFirmaTextoTO">
        <xsd:sequence>
            <xsd:element name="idCliente" type="xsd:string"></xsd:element>
            <xsd:element name="texto" type="xsd:string"></xsd:element>
            <xsd:element name="firmaB64" type="xsd:string"></xsd:element>
        </xsd:sequence>
    </xsd:complexType></xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://validacionFirma.ws.pfd.com">
    <xsd:complexType name="validarFirmaPDFTO">
        <xsd:sequence>
            <xsd:element name="idCliente" type="xsd:string"></xsd:element>
            <xsd:element name="pdfB64" type="xsd:string"></xsd:element>
        </xsd:sequence>
    </xsd:complexType></xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">
    <xsd:complexType name="firmaCorporativaCorreoTO">
        <xsd:sequence>
            <xsd:element name="idCliente" type="xsd:string"></xsd:element>
            <xsd:element name="datosCorreo"
                type="tns3:datosCorreoTO">
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="datosCorreoTO">
        <xsd:sequence>
            <xsd:element name="textoCorreo" type="xsd:string"></xsd:element>
            <xsd:element name="from" type="xsd:string"></xsd:element>
            <xsd:element name="to" type="xsd:string"></xsd:element>
            <xsd:element name="cc" type="xsd:string"></xsd:element>
            <xsd:element name="bcc" type="xsd:string"></xsd:element>
            <xsd:element name="subject" type="xsd:string"></xsd:element>
            <xsd:element name="b64Attach" type="xsd:string"></xsd:element>
            <xsd:element name="nombreFicheroAttach"
                type="xsd:string">
            </xsd:element>
            <xsd:element name="tipoCorreo" type="xsd:int"></xsd:element>
            <xsd:element name="envio" type="xsd:string"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="resultFirmaTO">
        <xsd:sequence>
            <xsd:element name="resultado" type="xsd:string"></xsd:element>
            <xsd:element name="firmaB64" type="xsd:string"></xsd:element>
            <xsd:element name="idValidacion" type="xsd:string"

```

```

maxOccurs="1" minOccurs="0">
  </xsd:element>
  <xsd:element name="rutaFichValidacion" type="xsd:string" maxOccurs="1"
minOccurs="0"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">
  <xsd:complexType name="firmaCorporativaPDFTO">
    <xsd:sequence>
      <xsd:element name="idCliente" type="xsd:string"></xsd:element>
      <xsd:element name="pdfB64" type="xsd:string"></xsd:element>
      <xsd:element name="selloFirma"
        type="tns3:selloFirmaPDFTO">
      </xsd:element>
      <xsd:element name="CSV" type="xsd:string" maxOccurs="1"
minOccurs="0"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="selloFirmaPDFTO">
    <xsd:sequence>
      <xsd:element name="localidad" type="xsd:string"></xsd:element>
      <xsd:element name="razon" type="xsd:string"></xsd:element>
      <xsd:element name="visible" type="xsd:int"></xsd:element>
      <xsd:element name="fechaFirma" type="xsd:string"></xsd:element>
      <xsd:element name="formatoFecha" type="xsd:string"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">
  <xsd:complexType name="firmaCorporativaTextoTO">
    <xsd:sequence>
      <xsd:element name="idCliente" type="xsd:string"></xsd:element>
      <xsd:element name="texto" type="xsd:string"></xsd:element>
    </xsd:sequence>
  </xsd:complexType></xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">
  <xsd:complexType name="firmaCorporativaXMLTO">
    <xsd:sequence>
      <xsd:element name="idCliente" type="xsd:string"
maxOccurs="1" minOccurs="1">
      </xsd:element>
      <xsd:element name="texto" type="xsd:string"
maxOccurs="1" minOccurs="1">
      </xsd:element>
      <xsd:element name="tipoFirma" type="xsd:string" maxOccurs="1"
minOccurs="1"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="resultFirmaTO"></xsd:complexType></xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">

```

```

<xsd:complexType name="resultSolAppletFirmaTO">
  <xsd:sequence>
    <xsd:element name="resultado" type="xsd:string" maxOccurs="1"
minOccurs="1"></xsd:element>
    <xsd:element name="paginaAppletB64" type="xsd:string" maxOccurs="1"
minOccurs="1"></xsd:element>
  </xsd:sequence>
</xsd:complexType><xsd:complexType
name="ListaParametrosApplet"></xsd:complexType></xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://validacionFirma.ws.pfd.com">
  <xsd:complexType name="solicitudDocumentoTO">
    <xsd:sequence>
      <xsd:element name="idAppCliente" type="xsd:string"></xsd:element>
      <xsd:element name="idValidacion" type="xsd:string"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="resultSolicitudDocumentoTO">
    <xsd:sequence>
      <xsd:element name="resultado" type="xsd:string"></xsd:element>
      <xsd:element name="documento" type="tns2:documentoTO"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="documentoTO">
    <xsd:sequence>
      <xsd:element name="idValidacion"
        type="xsd:string">
      </xsd:element>
      <xsd:element name="cliente" type="xsd:string"></xsd:element>
      <xsd:element name="procesoFirma"
        type="xsd:string">
      </xsd:element>
      <xsd:element name="fecha" type="xsd:string"></xsd:element>
      <xsd:element name="resultado" type="xsd:string"></xsd:element>
      <xsd:element name="cn" type="xsd:string"></xsd:element>
      <xsd:element name="certificadoB64"
        type="xsd:string">
      </xsd:element>
      <xsd:element name="firmab64" type="xsd:string"><xsd:annotation>
        <xsd:documentation>Contiene el documento en base64, Será
necesario decodificarlo para visualizar o almacenar el documento.</xsd:documentation>
      </xsd:annotation></xsd:element>
      <xsd:element name="idsAuditoriasAsociadas" type="xsd:string" maxOccurs="1"
minOccurs="0"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://firma.ws.pfd.com">
  <xsd:complexType name="datosAsociacionFirmaTO">
    <xsd:sequence>
      <xsd:element name="idCliente" type="xsd:string" maxOccurs="1" minOccurs="1"></xsd:element>
      <xsd:element name="idValidacion" type="xsd:string"
        maxOccurs="1" minOccurs="1">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

        <xsd:element name="firmaPFDb64" type="xsd:string"
            maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="idAuditoriaFirmaCliente"
            type="xsd:string" maxOccurs="1" minOccurs="1">
        </xsd:element>

    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="resultadoAsociacionFirmaTO">
    <xsd:sequence>
        <xsd:element name="idResultado" type="xsd:string"
            maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="descResultado" type="xsd:string" maxOccurs="1"
minOccurs="0"></xsd:element>
    </xsd:sequence>
</xsd:complexType></xsd:schema>
</wsdl:types>
<wsdl:message name="validarCertificadoRequest">
<wsdl:part name="validarCertificadoRequest" type="tns1:validarCertificadoTO"/>
</wsdl:message>
<wsdl:message name="validarCertificadoResponse">
<wsdl:part name="validarCertificadoResponse" type="tns1:resultValidacionCertificadoTO"/>
</wsdl:message>

<wsdl:message name="validarFirmaCorreoRequest">
    <wsdl:part name="validarFirmaCorreoRequest" type="tns2:validarFirmaCorreoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="validarFirmaCorreoResponse">
    <wsdl:part name="validarFirmaCorreoResponse" type="tns2:resultValidacionFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="validarFirmaTextoRequest">
    <wsdl:part name="validarFirmaTextoRequest" type="tns2:validarFirmaTextoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="validarFirmaTextoResponse">
    <wsdl:part name="validarFirmaTextoResponse" type="tns2:resultValidacionFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="validarFirmaPDFRequest">
    <wsdl:part name="validarFirmaPDFRequest" type="tns2:validarFirmaPDFTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="validarFirmaPDFResponse">
    <wsdl:part name="validarFirmaPDFResponse" type="tns2:resultValidacionFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaCorreoRequest">
    <wsdl:part name="firmaCorporativaCorreoRequest"
type="tns3:firmaCorporativaCorreoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaCorreoResponse">
    <wsdl:part name="firmaCorporativaCorreoResponse" type="tns3:resultFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaPDFRequest">
    <wsdl:part name="firmaCorporativaPDFRequest" type="tns3:firmaCorporativaPDFTO"></wsdl:part>
</wsdl:message>

```

```

<wsdl:message name="firmaCorporativaPDFResponse">
  <wsdl:part name="firmaCorporativaPDFResponse" type="tns3:resultFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaTextoRequest">
  <wsdl:part name="firmaCorporativaTextoRequest"
type="tns3:firmaCorporativaTextoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaTextoResponse">
  <wsdl:part name="firmaCorporativaTextoResponse" type="tns3:resultFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaXMLRequest">
  <wsdl:part name="firmaCorporativaXMLRequest" type="tns3:firmaCorporativaXMLTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="firmaCorporativaXMLResponse">
  <wsdl:part name="firmaCorporativaXMLResponse" type="tns3:resultFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="solicitarAppletFirmaRequest">
  <wsdl:part name="solicitarAppletFirmaRequest" type="xsd:string"></wsdl:part>
</wsdl:message>
<wsdl:message name="solicitarAppletFirmaResponse">
  <wsdl:part name="solicitarAppletFirmaResponse" type="tns3:resultSolAppletFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="obtenerDocumentoRequest">
  <wsdl:part name="obtenerDocumentoRequest" type="tns2:solicitudDocumentoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="obtenerDocumentoResponse">
  <wsdl:part name="obtenerDocumentoResponse"
type="tns2:resultSolicitudDocumentoTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="asociarFirmaClienteRequest">
  <wsdl:part name="asociarFirmaClienteRequest" type="tns3:datosAsociacionFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:message name="asociarFirmaClienteResponse">
  <wsdl:part name="asociarFirmaClienteResponse"
type="tns3:resultadoAsociacionFirmaTO"></wsdl:part>
</wsdl:message>
<wsdl:portType name="ServiciosPFDSservice">
<wsdl:operation name="validarCertificado">
<wsdl:input message="tns:validarCertificadoRequest"/>
<wsdl:output message="tns:validarCertificadoResponse"/>
</wsdl:operation>

```

```

<wsdl:operation name="validarFirmaCorreo">
  <wsdl:input message="tns:validarFirmaCorreoRequest"></wsdl:input>
  <wsdl:output message="tns:validarFirmaCorreoResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="validarFirmaTexto">
  <wsdl:input message="tns:validarFirmaTextoRequest"></wsdl:input>
  <wsdl:output message="tns:validarFirmaTextoResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="validarFirmaPDF">
  <wsdl:input message="tns:validarFirmaPDFRequest"></wsdl:input>

```

```

<wsdl:output message="tns:validarFirmaPDFResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="firmaCorporativaCorreo">
  <wsdl:input message="tns:firmaCorporativaCorreoRequest"></wsdl:input>
  <wsdl:output message="tns:firmaCorporativaCorreoResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="firmaCorporativaPDF">
  <wsdl:input message="tns:firmaCorporativaPDFRequest"></wsdl:input>
  <wsdl:output message="tns:firmaCorporativaPDFResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="firmaCorporativaTexto">
  <wsdl:input message="tns:firmaCorporativaTextoRequest"></wsdl:input>
  <wsdl:output message="tns:firmaCorporativaTextoResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="firmaCorporativaXML">
  <wsdl:input message="tns:firmaCorporativaXMLRequest"></wsdl:input>
  <wsdl:output message="tns:firmaCorporativaXMLResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="solicitarAppletFirma">
  <wsdl:input message="tns:solicitarAppletFirmaRequest"></wsdl:input>
  <wsdl:output message="tns:solicitarAppletFirmaResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtenerDocumento">
  <wsdl:input message="tns:obtenerDocumentoRequest"></wsdl:input>
  <wsdl:output message="tns:obtenerDocumentoResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="asociarFirmaCliente">
  <wsdl:input message="tns:asociarFirmaClienteRequest"></wsdl:input>
  <wsdl:output message="tns:asociarFirmaClienteResponse"></wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ServiciosPFDSserviceSOAP" type="tns:ServiciosPFDSservice">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

```

```

<wsdl:operation name="validarCertificado">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

```

<wsdl:operation name="validarFirmaCorreo">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

```
<wsdl:operation name="validarFirmaTexto">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="validarFirmaPDF">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="firmaCorporativaCorreo">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="firmaCorporativaPDF">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="firmaCorporativaTexto">
  <soap:operation/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="firmaCorporativaXML">
  <soap:operation/>
  <wsdl:input>
```

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="solicitarAppletFirma">
    <soap:operation/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtenerDocumento">
    <soap:operation/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="asociarFirmaCliente">
    <soap:operation/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="ServiciosPFDSservice">
<wsdl:port binding="tns:ServiciosPFDSserviceSOAP" name="ServiciosPFDSservice.service">
<soap:address location="http://www.example.org"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

5.9. ACCESO HTTPS: LIBRERIAS

Es necesario incluir en el lib las librerías:

- commons-httpclient-2.0.2.jar
- plataformaFirmaCore.x.x.x.jar
- commons-logging.jar
- jdom.jar

5.10. ACCESO WEBSERVICE: LIBRERIAS

Es necesario incluir en el cacert el certificado SSL del servidor de la plataforma de firma

Es necesario para un correcto funcionamiento en ejecución, incluir en el lib las librerías:

- jdom.jar
- axis.jar
- plataformaFirmaCore.x.x.x.jar
- commons-logging.jar
- commons-httpclient-2.0.2.jar
- web-services-clientePFD.x.x.jar

5.11. URLS DE ACCESO AL SERVICIO

La URL de acceso al servicio es:

Entorno demo:

<https://demo.redabogacia.org/plataformafirma/services/ServiciosPFDSERVICE.service>

Entorno producción:

<https://www.redabogacia.org/plataformafirma/services/ServiciosPFDSERVICE.service>