API de Firma Web, API de Firma En Servidor i Passarel·la de Firma PortaFIB

Emprar el model de plugins i el propi PortaFIB com a Plugin de Firma en aplicacions que requereixen firmes web i firmes en servidor immediates

Govern de les Illes Balears
Vicepresidència i Conselleria
d'Innovació, Recerca i Turisme
Direcció General de Desenvolupament Tecnològic

Govern de les Illes Balears
Fundació Balear d'Innovació i Tecnologia

Govern de les Illes Balears
Fundació Balear d'Innovació i Tecnologia

Informació general del document.

Descripció.

Títol:	API de Firma Web&EnServidor i Passarel·la de Firma PortaFIB
Estat:	Esborrany/Aprovat
Versió:	1.0
Autor/s:	Antoni Nadal Bennasar
Creat:	04/02/2016
Modificat	05/09/2016
Fitxer:	API_Firma_Web&EnServidor_i_ Passarela_de_Firma_PortaFIB.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Millores per XAdES	A. Nadal	23/05/2016
Canvis @firma AutoFirma i Client @firma Mòbil	A. Nadal	08/06/2016
Eliminat Tuckey per gestionar peticions web dels plugins	A. Nadal	06/08/2016
Adaptar Document per incloure API de Firma en Servidor	A. Nadal	02/09/2016

Font documental.

Index de Contingut

1.- Introducció

1.1.- Introducció Firma Web

Existeix una necessitat o requeriment, de cada vegada més important, en les aplicacions de l'Administració Pública que és la de firmar documents per assegurar la no modificació i l'acceptació del mateix en el temps. Aquesta firma actualment la realitzen dos tipus d'aplicacions:

(1) Eines d'escriptori que realitzen signatures dels documents que es troben al computador de l'usuari (Eines de firma d'IBKey d'escriptori, Eines de @firma, ...)
(2) Portafirmes centralitzats via web.

El cas que estam estudiant és el d'un sol·licitant tipus Aplicació i qui ha de firmar una persona. D'aquesta forma la solució (1) no ens serveix i la (2) pareix que si. El cas (2) té un petit inconvenient que és que el proces de firma és asíncron, és a dir, el sol·licitant tipus aplicació envia una petició de firma a PortaFIB i s'espera fins que el destinatari decideix entrar a la compte del PortaFIB i signar els documents pendents (parlam d'hores, dies o fins i tot setmanes).

Un afegit al cas que estam estudiam és que el flux de funcionament de l'aplicació origen requereix un firma **immediata**. Per exemple, un flux web que en algun pas es requereix una firma (aquí ja podem fer referència a aplicacions conegudes, com són Helium i Sistra).

La única alternativa que tenen aquestes aplicacions és emprar algun mecanisme de firma WEB com el Client @firma o el MiniApplet de @firma (o API de IBKey). El gran inconvenient d'aquesta solució és que és molt poc flexible i molt poc mantenible (s'ha d'estar constantment configurant/actualitzant cadascuna de les aplicacions que requereixen firma immediatament).

El que proposem és similar al que passa amb les firmes asíncrones, que s'ha arribat a la conclusió que la centralització en un portafirmes (PortaFIB) de les peticions és la millor opció, és a dir, que es delegui la firma immediata en un servei extern. Ja que el portafirmes és el que s'encarrega de tot el relacionat amb les firmes, aquesta seria l'aplicació candidata perfecte per oferir el servei.

Una altra forma de veure el que és vol solucionar, seria fent un símil amb les passarel·les de pagament. En lloc d'implementar el pagament un mateix dins d'un servidor, passa una petició a un servei extern (passarel·la de pagament o PortaFIB en el nostre cas) i d'aquesta forma ens llevam de damunt la complexitat del proces a realitzar i a més centralitzam en un mateix lloc totes les mateixes peticions (les de pagament o en el nostre cas les peticions de firma).

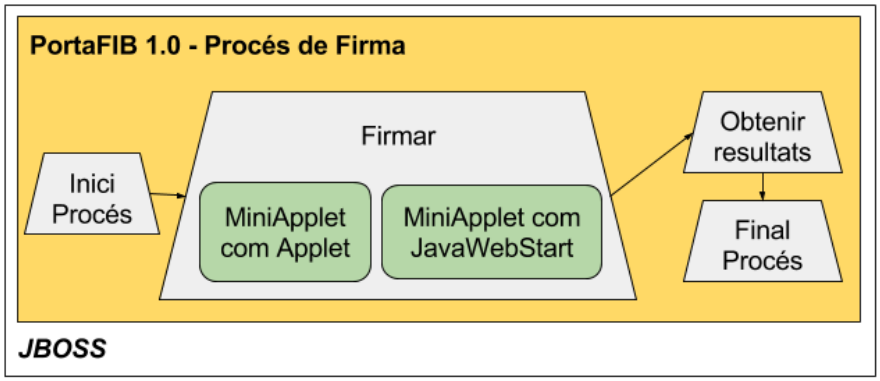
1.2.- Introducció Firma en Servidor

La mateixa problemàtica que s'ha explicat per firma web, també existeix per firma en servidor. Aquest tipus de firma no requereix de cap tipus d'interacció amb l'usuari i es realitza de forma desatesa per part del propi servidor. En aquest cas simplement es tracta de posar una capa que ens aïlli de les diferents implementacions de firma en servidor, així com aïllar-nos de la complexitat i poder canviar d'implementació en qualsevol moment sense que l'aplicatiu que el crida es vegi afectat.

Al ser molt més senzilla aquesta l'API de Firma En Servidor que la Firma Web, es farà més èmfasi en l'API de Firma Web que en la de En Servidor.

2.- PortaFIB 1.1 i Plugins o Mòduls de Firma

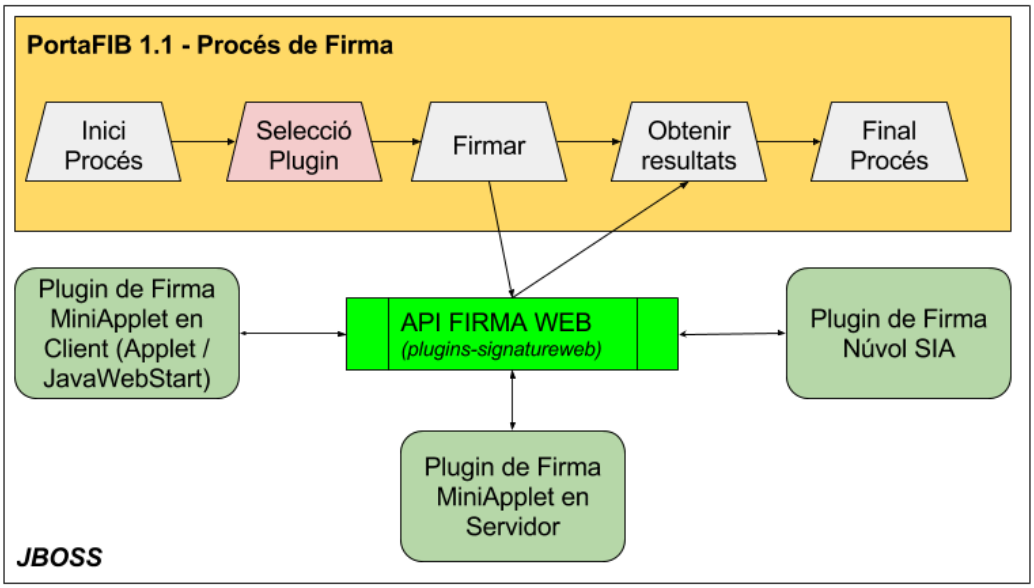
En la primera versió de PortaFIB, la 1.0, només es podia firmar emprant l'Applet de Firma anomenat MiniApplet. Al final se li va afegir que l'applet s'executés com a JavaWebStart a causa de la limitació d'execució d'Applets per part de Chrome. Aquestes dues formes de firma estaven incrustades dins codi. En el següent diagrama es pot observar la firma en un PortaFIB 1.0 on no es fan servir plugins i incrustat en el codi hi ha les rutines necessàries per per funciona el MiniApplet.



En la versió 1.1 de PortaFIB el sistema de firma funciona a través de Plugins, cosa que fa que es flexibilitzi i es doni la llibertat d'afegir sistemes/formes de firma a voluntat, només implementant un nou plugin. El plugins de firma actualment implementats són:

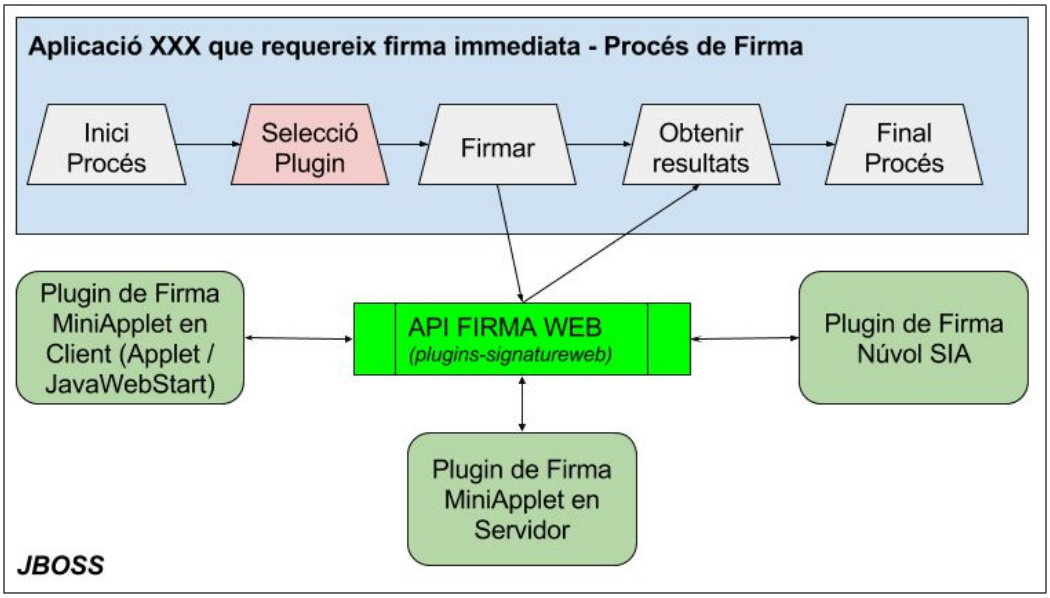
- Plugin de Firma de MiniApplet en Client (Applet/JavaWebStart)
- Plugin de Firma de MiniApplet en Servidor
- Plugin de Firma en Servidor SIA
- Plugin de Firma emprant @firma AutoFirma i Client @firma Mòbil
- Plugin de Firma en Núvol de Cl@ve (encara no implementat)

En el següent gràfic es pot observar el procés de firma dins de PortaFIB 1.1



3.- Primera Aproximació a la Solució

En el diagrama del punt anterior, tot el que es firma està totalment aïllat i separat de PortaFIB. Els plugins que s'han implementat són independents a qualsevol classe o mòdul de PortaFIB. Doncs la solució a simple vista és clonar el model de firma adoptat per PortaFIB i incrustar-ho dins totes les aplicacions que necessitin firma "immediata".



3.1.- Avantatges

Les avantatges són les mateixes que ofereix a PortaFIB i es concentren en una de sola: "Independència de la forma de Firma(verd fort) a través d'un API genèric (verd viu)". Això deriva en altres avantatges que són la intemporalitat de la solució, es a dir, que el model de l'API ens separa de la implementació cosa que fa que l'actualització del Plugin de Firma no sigui cap problema (fins i tot es podrien canviar els plugins sense ni haver de recompilar l'aplicació si els jars d'aquest estan en un ear separat).

3.2.- Inconvenients

Ens hem de plantejar l'escenari en que tenim varis productes que implementen aquest model (per exemple que Sistra i Helium adaptessin aquest sistema com a forma de firma de fitxers). Anem a veure quins inconvenients

3.2.1.- Configuració dels Plugins

Algun plugins com l'Applet de Miniapplet no requereixen configuració en si mateix. D'altres com SIA si que requereixen configuració (Per exemple fitxer de propietats o BBDD)

Per un altre costat l'aplicació Hoste necessita algun sistema per gestionar la llista de plugins disponibles (Per exemple fitxer de propietats o BBDD)

Per finalitzar hi ha d'haver un manteniment de les versions dels plugins a nivell de desenvolupador. Ea a dir, si passam de la versió 1.0.0 del Plugin de MiniApplet com Applet a la 1.0.1 tant dins Helium com dins Sistra hauran d'actualitzar les referències i tres quarts de lo mateix per la resta de plugins.

3.2.2.- Generadors de Segell de Temps

El segells de temps no s'implementen ni dins el Plugin ni dins la cap d'API. Els segells de els genera i proporciona l'aplicació hoste a través d'una Interfície anomenada ITimeStampGenerator. Encara que les implementacions estan disponibles a traves del projecte PluginsIB l'adaptació, gestió i manteniment d'aquest plugins de segellat de temps s'haurien de fer dins cada aplicació hoste.

3.2.3.- Suport de Custòdia

Passa el mateix que el punt anterior. Les custòdies implementen una interfície anomenada IDocumentCustody de PluginsIB, i els inconvenients són els mateixos descrits al punt anterior.

3.2.4.- Generadors de PDF Visible

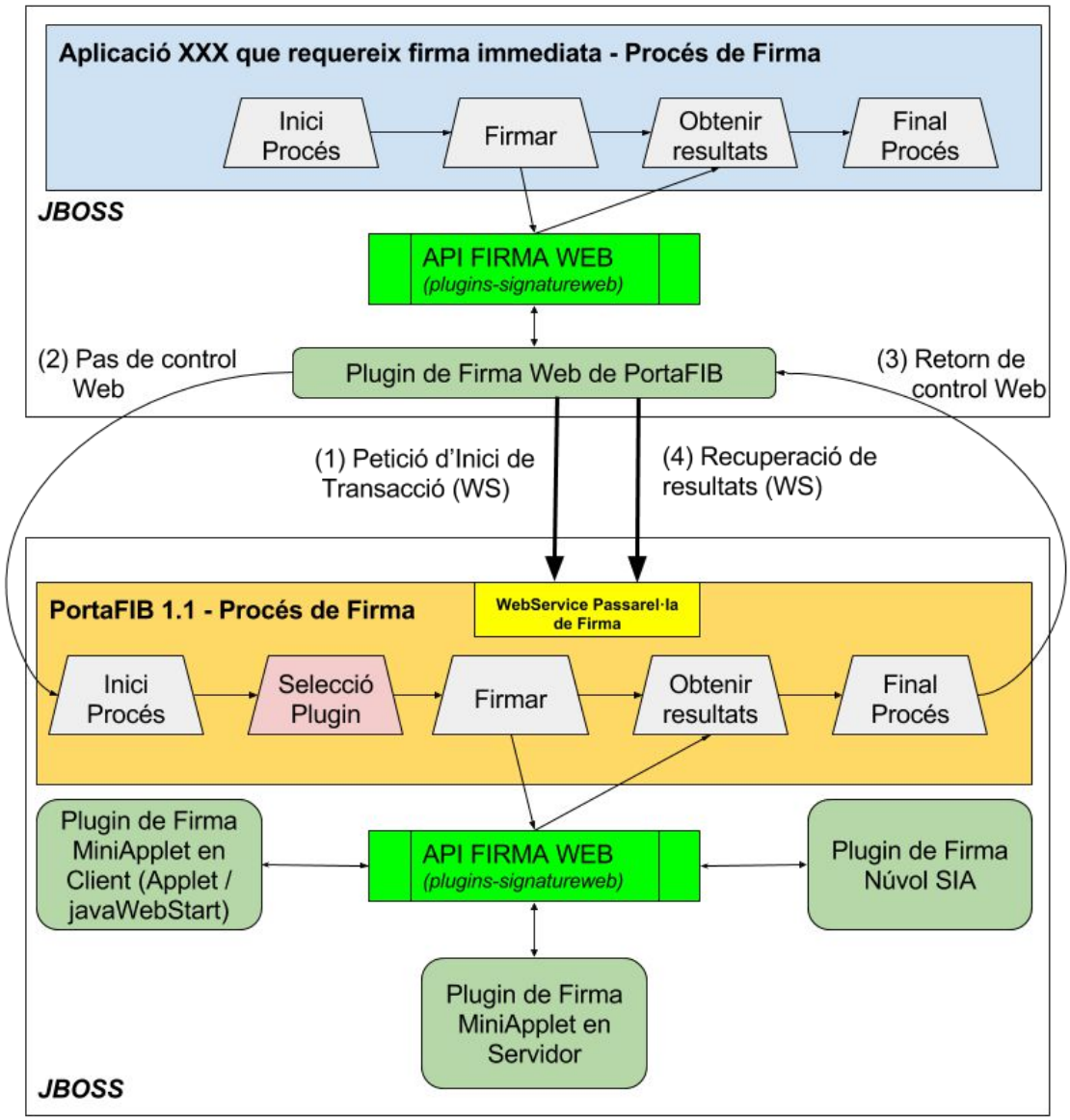
Passa el mateix que el punt anterior o pitjor, ja actualment l'únic generador de la imatge que s'incrusta com a Firma PDF Visible es troba dins del codi de PortaFIB. S'hauria de clonar a totes les aplicacions hoste o extreure a un mòdul dins de PluginsIB.

S'hauria de trobar una altra solució al problema, minimitzant el s inconvenients abans descrits.

3.3.- Com implementar la Passarel·la de Firma PortaFIB

L'API de Firma Web és un API que encapsula una forma/sistema/mecanisme via web de firmar un document. El que requerim per la Passarel·la de Firma és això, o sigui una altra forma/sistema/mecanisme de firma web però que és connectarà a PortaFIB per firmar.

Llavors la solució passa per reutilitzar l'API de Firma Web, però substituir els plugins de Firma per només un que cridi a PortaFIB i que sigui només en aquest darrer que es facin tots els manteniments de tots els plugins (Plugins de Firma, Plugins de Segellat de Temps, generadors de Firma PDF Visible o Plugins de Custòdia):



L'explicació de cada pas és comenta a continuació:

(1) **Petició d'Inici de Transacció (WebServices)**: Serveix per enviar tota la informació de firma: documents a firmar i amb quin tipus de firma, si volem timestamp, si volem custòdia, si volem firma PDF visible, ... a més el següent pas es passar el control web a una plana web de PortaFIb, per la qual cosa s'ha d'enviar la URL de retorn o tornada a l'aplicació XXX (o url de callback). El que retornarà és un id de transacció i una URL de redirecció al Portafirmes. Qualsevol error llançarà un excepció.
(2) **Pas de control Web**: ja s'ha obtingut la URL de salt i farem que vagi a PortaFIB (d'alguna forma se li passarà la id de la transacció). Aquí PortaFIB mostrarà la selecció de plugins si escau i realitzarà la firma.
(3) **Retorn de control Web**: una vegada finalitzat el procés de firma PortaFIB retornarà el control a l'aplicació XXX (enviada la url a través de l'inici de la transacció)
(4) **Recuperació de resultats (WebServices)**: Un cop recuperat el control, el primer que farà

el l'aplicació XXX es reclamar els resultats (documents firmats) a traves de l'API de Firma Web, aquesta cridarà al plugin corersponent (en aquest cas Plugin de PortaFIB) i el plugin emprant de WebServices recuperarà els documents firmats.

Queda pendent veure si els inconvenients descrits en el punt anterior, amb aquesta proposta es solucionen:

(a) **Segellat de Temps**: Segons la configuració l'usuari podria decidir si vol o no segellat de temps, encara que depèn de la configuració establerta a PortaFIB per aquella entitat (Les configuracions possibles serien “Obliga a l'ús de Segell de Temps”, “No emprar Segell de temps” o decideix el peticionari). Aquesta informació es passaria a través de l'Inici de Transacció. No hi hauria cap tipus de configuració en l'hoste.

(b) **Firma PDF Visible**: Cada usuari aplicació està relacionat a una entitat. El contingut d'aquesta regió associada a una firma PADES s'obtindria de la configuració definida en l'entitat. No hi hauria cap tipus de configuració en l'hoste.

(c) **Custòdia**: Aquesta informació es passaria a través de l'Inici de Transacció. També dependria de la configuració definida en el servidor per aquella entitat (Les configuracions possibles en el servidor són “No voler custòdia”, “Custodia predefinida”, “Custòdia de lliure elecció”). No hi hauria cap tipus de configuració en l'hoste.

4.- API de Firma Web i API Firma En Servidor

4.1.- Conceptes

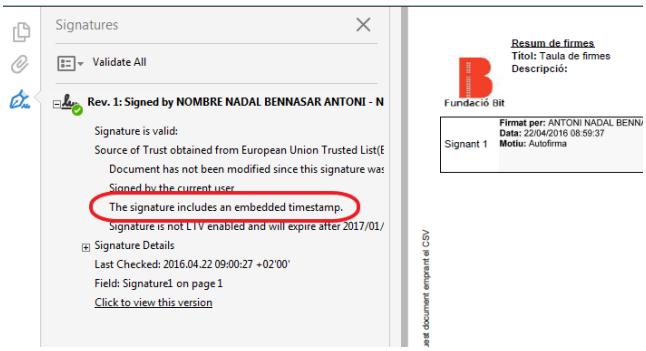
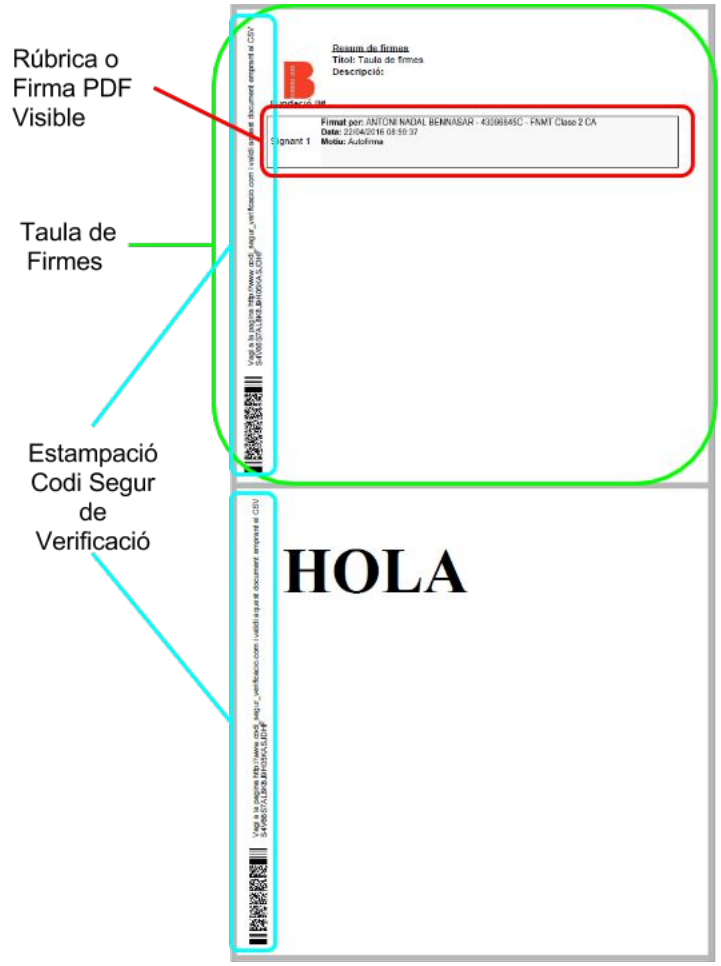
L'API de Firma permet fonamentalment cinc coses:

- Firmar
- Política de Firma
- Segellat de Temps (Veure captura 1 de més abaix)

I pel cas de Firmes PAdES opcionalment permet (Veure captura 2 de més abaix):

- Creació de Taula de Firmes i Estampació de Rúbriques (o Firma PDF Visible)
- Estampació de CSV

A continuació mostrarem d'una forma gràfica els elements que acabam de descriure:

Captura 1 En la captura de la dreta podem observar que en la petició de Firma s'ha demanat la inclusió d'un Segellat de Temps. En el cas de no haver-se demanat, apareixeria un missatge indicant que la data de la firma s'ha obtingut de la màquina on es firmava.	
Captura 2 En la captura de la Dreta podem observar els conceptes de “Taula de Firmes”, Estampació de Rúbriques (o Firma PDF Visible) i Estampació de CSV	

4.2.- Plugins de Firma Web i característiques

Accions suportades per cadascun dels Plugins Web actualment implementats:

Acció	MiniApplet in client	MiniApplet in server	SIA	@firma AutoFirma	Passarel·la PortaFIB
Firma XAdES	√	√		√	√
Firma PAdES	√	√	√	√	√
Segellat de Temps					√
Taula de Firmes i Estampació de Rúbriques (PAdES)					√
Estampació de CSV (PAdES)					√
Número de Firmes a la vegada	N	N	N	1	N

4.3.- Plugins de Firma en Servidor i característiques

Accions suportades per cadascun dels Plugins de Firma En Servidor actualment implementats:

Acció	MiniApplet in server	Passarel·la PortaFIB
Firma XAdES	√	√
Firma PAdES	√	√
Segellat de Temps		√
Taula de Firmes i Estampació de Rúbriques (PAdES)		√
Estampació de CSV (PAdES)		√
Número de Firmes a la vegada	N	N

4.4.- Classes de Api de Firma Web i Api de Firma en Servidor

Tant l'API de Firma Web com l'API de Firma en Servidor utilitzen les mateixes classes excepte la classe arrel que en el primer cas es troba la classe SignaturesSetWeb[1] i en el segon cas SignaturesSet[2] . Aquestes classes són les que engloben tota la informació de firma i contenen tota la informació de quines firmes s'han de realitzar i com.

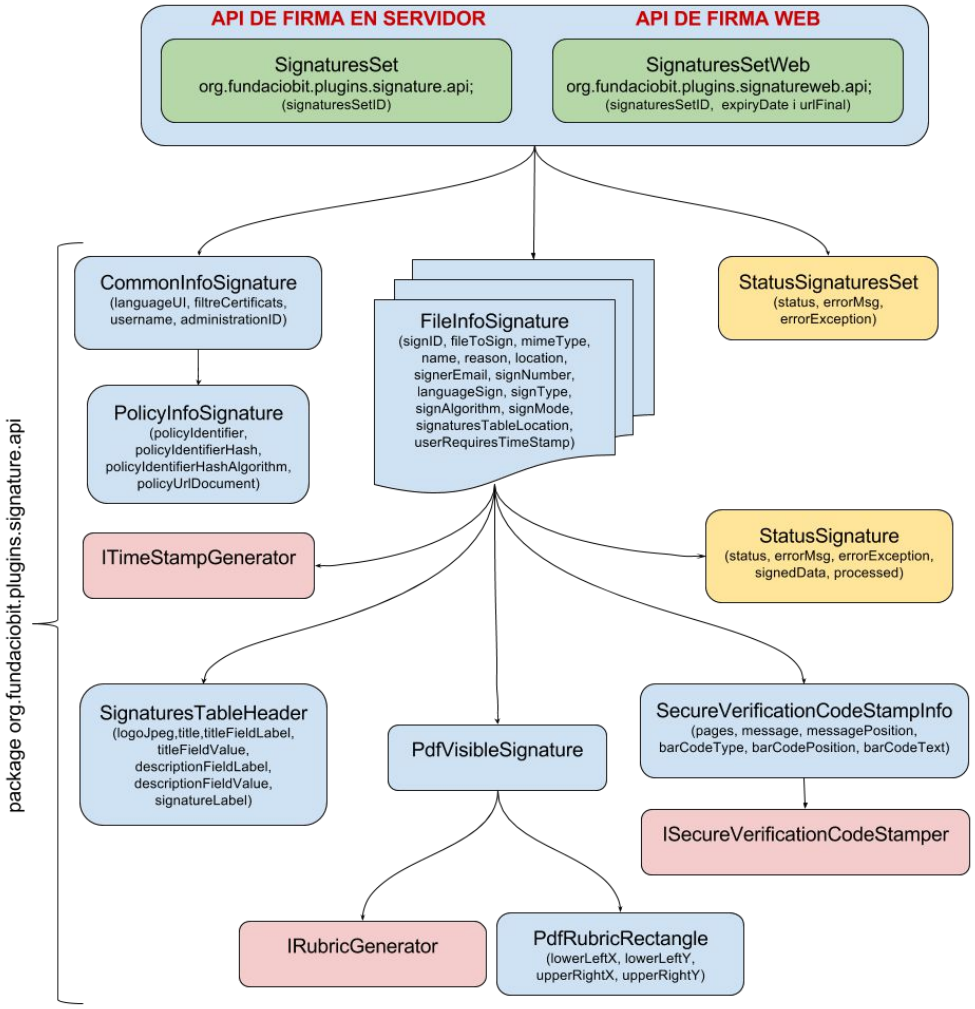
1 org.fundaciobit.plugins.signatureweb.api.SignaturesSetWeb
2 org.fundaciobit.plugins.signature.api.SignaturesSet

4.4.1.- Classes segons Funcionalitat

Les classes i camps de l'API encarregades de cadascuna de les operacions enunciades en l'apartat "4.1.-Conceptes" es descriuen a continuació:

Acció	*Interfícies, Classes i Camps involucrats*
Gestió del Proces de Firma	SignaturesSetWeb(web) o SignaturesSet(en servidor), CommonInfoSignature, StatusSignaturesSet, StatusSignature
Firmar	FileInfoSignature
Política de Firma	PolicyInfoSignature
Segellat de Temps	FileInfoSignature.userRequiresTimeStamp, ITimeStampGenerator
Taula de Firmes i Estampat de Rúbriques	FileInfoSignature.signaturesTableLocation, SignaturesTableHeader, PdfVisibleSignature, PdfRubricRectangle, IRubricGenerator
Estampació de CSV	SecureVerificationCodeStampInfo, ISecureVerificationCodeStamper

A continuació podem veure un esquema de l'estructura de classes (S'ha reutilitzat la primera casella tant per SignaturesSet com per SignaturesSetWeb):



4.5.- Detalls de API de Firma Web

4.5.1.- Interfície ISignatureWebPlugin

Aquesta interfície representa la informació i accions que ofereix el plugin de firma:

- Informació de les característiques que ofereix
- Inici de transacció de Firma
- Recuperació de resultats i d'estat

L'ús dels mètodes següents es pot observar en les diferents classes de l'exemple de passarel·la de PortaFIB (Veure classes EJB SignatureWebPluginManager/SignatureModuleEjb i WEB SignatureModuleController)

```
/**
 * @param locale
 *          idioma amb que es vol el nom del plugin
 * @return Nom del plugin
 */
public String getName(Locale locale);

/**
 * @return Els tipus de firma suportats. Actualment només es suporta PAdES.
 * @see FileInfoSignature.SIGN_TYPE_PADES = "PAdES";
 * @see FileInfoSignature.SIGN_TYPE_XADES = "XAdES";
 * @see FileInfoSignature.SIGN_TYPE_CADES = "CAdES";
 * @see FileInfoSignature.SIGN_TYPE_FACTURAE = "FacturaE";
 * @see FileInfoSignature.SIGN_TYPE_OOXML = "OOXML";
 * @see FileInfoSignature.SIGN_TYPE_ODF = "ODF";
 */
public String[] getSupportedSignatureTypes();

/**
 * @param signType
 *          Tipus de Firma
 * @return Retorna els algorismes suportats segons els tipus de firma passat
 *         per paràmetre
 */
public String[] getSupportedSignatureAlgorithms(String signType);

/**
 * @return Retorna els tipus de Barcode suportats per l'estampació del Codi
 *         Segur de Verificació (CSV). Per exemple, el tipus suportats pel
 *         plugins de PortaFIB són: BarCode128, Pdf417 i QrCode
 */
public List<String> getSupportedBarCodeTypes();

/**
 * Filtre que s'ha de cridar per esbrinar si aquest plugin pot realitzar la
 * firma web. Les següents comprovacions es fan en aquest mètode: tipus de firma,
 * algorismes de firma, segellat de temps, estampacio CSV,
 * taula de firmes i rubrica pdf, codi de barres, ...
 *
 * @param request  Petició de l'API Servlet
 * @param signaturesSet Informació de les firmes a realitzar
 * @return true, si aquest plugin es compatible per realitzar la firma.
 */
public boolean filter(HttpServletRequest request, SignaturesSet signaturesSet);
```

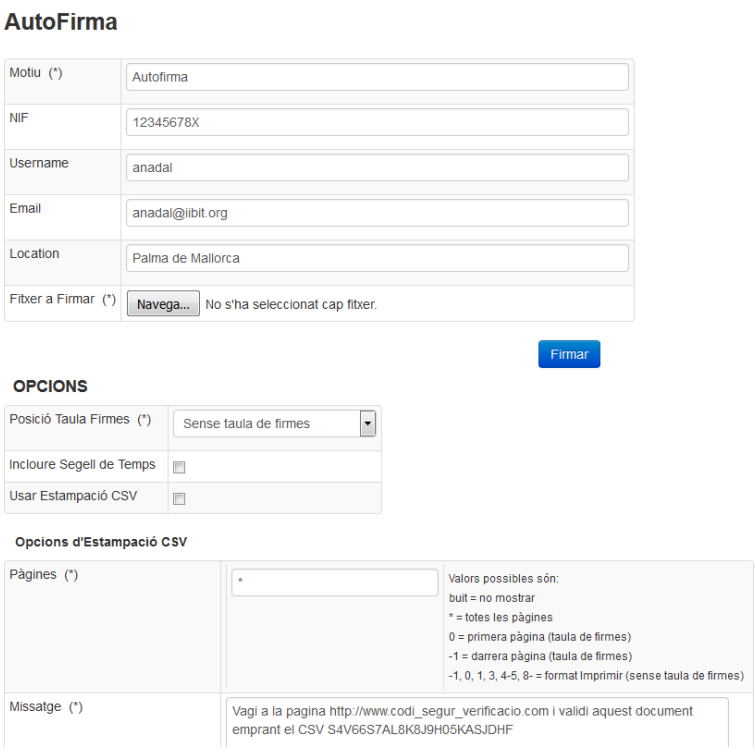
```
/**
 *
 * @param request
 *          Petició de l'API Servlet
 * @param signaturesSetID
 *          Identificació de
 * @throws Exception
 */
public void closeSignaturesSet(HttpServletRequest request, String signaturesSetID)
    throws Exception;

/**
 *
 * @param request
 *          Petició de l'API Servlet
 * @param absolutePluginRequestPath
 *          Base de la Ruta absoluta a aquest plugin
 * @param relativePluginRequestPath
 *          Base de la Ruta relativa a aquest plugin
 * @param signaturesSetWeb
 *          Informació completa del que s'ha de firmar i com
 * @return La URL on s'ha de redireccionar per iniciar el proces de firma
 * @throws Exception
 *           Si hi ha errors
 */
public String signDocuments(HttpServletRequest request, String absolutePluginRequestPath,
    String relativePluginRequestPath, SignaturesSetWeb signaturesSetWeb) throws Exception;

/**
 * Peticio GET
 *
 * @param absolutePluginRequestPath
 *          Base de la Ruta absoluta a aquest plugin
 * @param relativePluginRequestPath
 *          Base de la Ruta relativa a aquest plugin
 * @param query
 *          Resta de la ruta que s'ha critat
 * @param signaturesSetID
 *          Identificador del proces de Firma
 * @param signatureIndex
 *          Indica sobre quina firma s'aplica aquesta operatció. Si val -1
 *          significa que és una operació que s'aplica a tot el proces de
 *          firma
 * @param request
 *          Petició de l'API Servlet
 * @param uploadedFiles
 *          Llistat de Fitxers que venen adjunts a la petició web
 * @param response
 *          Resposta de l'API Servlet
 */
public void requestGET(String absolutePluginRequestPath, String relativePluginRequestPath,
    String query, String signaturesSetID, int signatureIndex, HttpServletRequest request,
    Map<String, IUploadedFile> uploadedFiles, HttpServletResponse response);

/**
 * Peticio POST
 *
 * @param absolutePluginRequestPath
 *          Base de la Ruta absoluta a aquest plugin
 * @param relativePluginRequestPath
 *          Base de la Ruta relativa a aquest plugin
 * @param query
 *          Resta de la ruta que s'ha critat
 * @param signaturesSetID
 *          Identificador del proces de Firma
 * @param signatureIndex
 *          Indica sobre quina firma s'aplica aquesta operatció. Si val -1
 *          significa que és una operació que s'aplica a tot el proces de
 *          firma
 * @param request
```

```
     *          Petició de l'API Servlet
     * @param uploadedFiles
     *          Llistat de Fitxers que venen adjunts a la petició web
     * @param response
     *          Resposta de l'API Servlet
     */
    public void requestPOST(String absolutePluginRequestPath, String relativePluginRequestPath,
        String query, String signaturesSetID, int signatureIndex, HttpServletRequest request,
        Map<String, IUploadedFile> uploadedFiles, HttpServletResponse response);

    /**
     *
     * @param signatureSetID
     *          Identificador del proces de Firma
     * @param signatureIndex
     *          Indica sobre quina firma volem saber l'estat.
     * @return Informació de l'estat
     */
    public StatusSignature getStatusSignature(String signatureSetID, int signatureIndex);

    /**
     * @param signaturesSetID
     *          Identificador del proces de Firma
     * @return Informació total de la petició
     */
    public SignaturesSetWeb getSignaturesSet(String signaturesSetID);

    /**
     * @param signType
     *          Tipus de Firma
     * @return true indica que el plugin accepta generadors de Segell de Temps
     *         definits dins FileInfoSignature.timeStampGenerator
     */
    public boolean acceptExternalTimeStampGenerator(String signType);

    /**
     * @param signType
     *          Tipus de Firma
     * @return true, indica que el plugin internament ofereix un generador de
     *         segellat de temps.
     */
    public boolean providesTimeStampGenerator(String signType);

    /**
     *
     * @return true indica que el plugin accepta generadors del imatges de la
     *         Firma Visible PDF definits dins
     *         FileInfoSignature.pdfInfoSignature.rubricGenerator.
     */
    public boolean acceptExternalRubricGenerator();

    /**
     *
     * @return true, indica que el plugin internament ofereix un generador de
     *         imatges de la Firma Visible PDF.
     */
    public boolean providesRubricGenerator();

    /**
     * @return true indica si el plugin accepta estampadors de Codi Segur de
     *         Verificació (missatge i/o codi de barres).
     */
    public boolean acceptExternalSecureVerificationCodeStamper();

    /**
     * @return true, indica que el plugin internament ofereix estampadors de Codi
     *         Segur de Verificació (missatge i/o codi de barres).
     */
    public boolean providesSecureVerificationCodeStamper();
```

5.- Adaptar una aplicació web a l'API de Firma Web

En es següents pàgines mostrarem pas a pas com adaptar una aplicació web per incorporar-li l'API de Firma i addicionalment la Passarel·la de Firma PortaFIB.

En el codi font de PortaFIB (https://github.com/GovernIB/portafib), concretament en el directori [PORTAFIB]\plugins-signatureweb\exemplepassarela existeix un exemple molt simple de WebApp adaptada per firmar un fitxer rebut des de formulari. Com és lògic la majoria d'aplicacions gestionaran internament el fitxer a firmar per la qual cosa no hi haurà formulari previ sinó que tot es farà de forma programàtica. Llegiu un fitxer install.txt per instal·lar l'aplicació d'exemple.



Captura de pantalla de l'Exemple de Passarel·la

5.1.- Capa de EJB

5.1.1.- Descarregar tot el codi de l'exemple de passarel·la de PortaFIB a un directori temporal (exemplepassarela)

5.1.2.- Afegir totes les classes i EJBs de l'exemple

5.1.3.- Adaptar la classes

Adaptar la classe SignatureWebPluginManager per obtenir la configuració del lloc que sigui. Nosaltres la deixaren tal i com està: a partir d'una propietat de sistema anomenada "signaturewebplugins.path" que conté una ruta a un fitxer de propietat llegirem les propietats dels plugins donats d'alta. Veure fitxer [PORTAFIB]\plugins-signatureweb\exemplepassarela\plugins.properties.

5.1.4.- Afegirem les dependències maven següents:

```
        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugins-api</artifactId>
            <version>1.0.0</version>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugin-signatureweb-api</artifactId>
            <version>2.0.0</version>
            <scope>provided</scope>
        </dependency>
```

5.1.5.- Repositoris d'on obtenir aquestes classes

Afegir les següent entrades al bloc <repositories> del pom.xml global del projecte web:

```
<repositories>
        <repository>
            <id>portafib-maven-repos</id>
            <name>PortaFIB Maven Repository</name>
            <url>http://GovernIB.github.io/portafib/maven/</url>
        </repository>

        <repository>
            <id>pluginsib-maven-repos</id>
            <name>PluginsIB Maven Repository</name>
            <url>http://GovernIB.github.io/pluginsib/maven/</url>
        </repository>
</repositories>
```

5.2.- Capa Web

5.2.1.- Copiar SignatureModuleController

Copiarem la classe SignatureModuleController: aquesta està implementada emprant Spring. En cas de no suportar Spring s'haurà d'adaptar al framework de la webapp (servlets, struts, ...) així com els jsps de exemple-war\src\main\webapp\WEB-INF\views*.jsp al directori de vistes (excepte els autofirma*.jsp).

5.2.2.- En web.xml:

- Afegir la ruta del Controlador a un entorn no autenticat, és a dir públic (<security-constraint>)
- Definir Servlet de peticions dels plugins:

```
<servlet>
```

```
  <servlet-name>PluginSignWebRequests</servlet-name>
  <servlet-class>
    org.fundaciobit.plugins.signatureweb.exemple.controller.SignatureWebModuleController
  </servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>PluginSignWebRequests</servlet-name>
    <url-pattern>/common/signwebmodule/requestPlugin/*</url-pattern>
</servlet-mapping>
```

5.2.3.- Adaptar pom.xml

Afegir les següents dependències:

```
        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugins-api</artifactId>
            <version>1.0.0</version>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugin-signatureweb-api</artifactId>
            <version>2.0.0</version>
            <scope>provided</scope>
        </dependency>
```

5.2.4.- Preparar cridada a API

- Revisar el mètode autofirmaPost() de la classe java exemple-war/plugin-signatureweb-exemplepassarela-war/src/main/java/org/fundaciobit/plugins/signatureweb/exemple/controller/AutoFirmaController.java on es pot observar el muntatge per un inici de transacció de firma: SignatureModuleController.startSignatureProcess(...). En el punt “S'ha produït un error: No s'ha trobat la font de referènciaS'ha produït un error: No s'ha trobat la font de referència“ podeu trobar una explicació de les classes i propietats.
- La API de firma permet enviar a firmar un número indeterminat de fitxers

5.3.- Capa EAR

5.3.1.- Dependències pom.xml

```
        <!-- ===================================== -->
        <!-- =====   PLUGINS DE SIGNATUREWEB  ===== -->
        <!-- ===================================== -->

        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugins-api</artifactId>
            <version>1.0.0</version>
        </dependency>

        <dependency>
            <groupId>org.fundaciobit.plugins</groupId>
            <artifactId>plugin-signatureweb-api</artifactId>
            <version>2.0.0</version>
        </dependency>
```

5.4.- Configuració

5.4.1.- Fitxer de Plugins

- Afegir la propietats de sistema “signaturewebplugins.path” a algun lloc amb la ruta al fitxer de configuració de plugins.
- Fitxer Configuració Plugins: Es pot emprar el fitxer de configuració plugins abans descrit, encara que les rutes s'hauran d'adaptar i alguns com els de SIA o PortaFIB s'hauran de comentar si no es té accés. ES RECOMANA començar les proves amb els plugins de MiniAppletInClient i MiniAppletInServer (aquest darrer requereix configurar un path a un directori)

5.4.2.- Ear de Plugins

Els plugins han d'anar dins un ear separat, d'aquesta forma l'actualització serà molt senzilla. S'ha de copiar el projecte [portafib]\plugins-signatureweb\exemplepassarela\exemple-plugins i adaptar el pom.xml i el fitxer “jboss-classloading.xml” amb el “domain” de l'ear de l'aplicació en qüestió. Aquí s'han de definir els Plugins que es desitgin emprar.

6.- Annexes

6.1.- Implementar Plugin de Firma Web

<<< PENDENT: Conjunt de pautes i normes a seguir a l'hora d'implementar un plugin de firma web. Com que ja n'hi ha 5 de implementats dins el projecte PortaFIB\plugins-signatureweb es poden revisar a veure com s'han implementat aquests >>>

6.2.- Implementar Plugin de Firma En Servidor

<<< PENDENT: Conjunt de pautes i normes a seguir a l'hora d'implementar un plugin de firma en servidor. Com que ja n'hi ha 2 de implementats dins el projecte PortaFIB\plugins-signatureserver es poden revisar a veure com s'han implementat aquests >>>

6.3.- Configuració Plugin de FirmaWeb PortaFIB

A continuació és mostra la classe i propietats necessàries per configurar el Plugin de PortaFIB.

Nom:
Plugin de Firma Web emprant PortaFIB (Passarel·la de Firma PortaFIB)

Classe Java:
org.fundaciobit.plugins.signatureweb.portafib.PortaFIBSignatureWebPlugin

Propietats: [BASE]=package del projecte o package elegit per les propietats.	
[BASE].**plugins.signatureweb.portafib.api_passarela_url**	Ruta al WS de passarela de Firma de PortaFIB Exemple: **http://localhost:8080/portafib/ws/v1/PortaFIBPassarelaDeFirma**
[BASE].**plugins.signatureweb.portafib.api_passarela_username**	Usuari WS emprat per connectar-se als WebServices
[BASE].**plugins.signatureweb.portafib.api_passarela_password**	Contrasenya associada a l'usuari aplicació que realitza la petició WS.
[BASE].**plugins.signatureweb.portafib.filter_by_plugin_ids**	Opcional. Per defecte (nodefinit) mostra tots els plugins de firma web disponibles a PortaFIB. Llistat de identificadors separats per comes de plugins que volem que mostri PortaFIB.
[BASE].**plugins.signatureweb.portafib.use_portafib_certificate_filter=true**	Opcional. Per defecte false. Si val true llavors s'utilitza el filtre de certificats de l'entitat de PortaFIB associat a l'usuari-app. Si val false, llavors s'utilitza el filtre de certificats definit en la propietat SignaturesSet.CommonInfoSignature.getFiltreCertificats()

Un exemple complet de propietats es mostra a continuació:

```
org.fundaciobit.exemple.signaturewebplugins.1.plugins.signatureweb.portafib.api_passarela_url=http://localhost:8080/portafib/ws/v1/PortaFIBPassarelaDeFirma
org.fundaciobit.exemple.signaturewebplugins.1.plugins.signatureweb.portafib.api_passarela_username=fundaciobit_usrapp
org.fundaciobit.exemple.signaturewebplugins.1.plugins.signatureweb.portafib.api_passarela_password=fundaciobit_usrapp
<!-- Opcional -->
#org.fundaciobit.exemple.signaturewebplugins.1.plugins.signatureweb.portafib.filter_by_plugin_ids=165068,23454
#org.fundaciobit.exemple.signaturewebplugins.1.plugins.signatureweb.portafib.use_portafib_certificate_filter=true
```

NOTA: La configuració per la resta de plugins es pot veure al manual d'Usuari de PortaFIB.

6.4.- Exemple Cridada API de Firma en Servidor

Com hem vist abans les dues APIs són quasi idèntiques, En en projecte [portafib]\plugins-signatureweb\exemplepassarela\exemple-war podeu veure exemples de cridades a lAPI. De totes formes aquí vos mostram uns exemples de cridades a l'API de Firma en Servidor.

6.4.1.- Mètode genèric

```java
public static void signFile(String pdfsource, String pdfdest, String signType, int signMode,
      boolean userRequiresTimeStamp, IRubricGenerator rubricGenerator,
      ISignatureServerPlugin plugin) throws Exception, FileNotFoundException, IOException {

    String languageUI = "ca";
    String filtreCertificats = "";
    String username = "anadal"; // configuracio
    String administrationID = null; // No te sentit en API Firma En Servidor
    PolicyInfoSignature policyInfoSignature = null;
    CommonInfoSignature commonInfoSignature = new CommonInfoSignature(languageUI,
        filtreCertificats, username, administrationID, policyInfoSignature);

    String signID = "999";
    File source = new File(pdfsource);
    String name = source.getName();
    String reason = "TEST SIGN";
    String location = "Palma";
    String signerEmail = "anadal@ibit.org";
    int signNumber = 1;
    String languageSign = "ca";

    String signAlgorithm = FileInfoSignature.SIGN_ALGORITHM_SHA1;

    int signaturesTableLocation = FileInfoSignature.SIGNATURESTABLELOCATION_WITHOUT;
    PdfVisibleSignature pdfInfoSignature = null;
    if (FileInfoSignature.SIGN_TYPE_PADES.equals(signType) && rubricGenerator != null) {
```

```
      signaturesTableLocation = FileInfoSignature.SIGNATURESTABLELOCATION_LASTPAGE;
      PdfRubricRectangle pdfRubricRectangle = new PdfRubricRectangle(106, 650, 555, 710);
      pdfInfoSignature = new PdfVisibleSignature(pdfRubricRectangle, rubricGenerator);
    }
    final ITimeStampGenerator timeStampGenerator = null;

    // Valors per defcte
    final SignaturesTableHeader signaturesTableHeader = null;
    final SecureVerificationCodeStampInfo csvStampInfo = null;

    FileInfoSignature fileInfo = new FileInfoSignature(signID, source,
        FileInfoSignature.PDF_MIME_TYPE, name, reason, location, signerEmail, signNumber,
        languageSign, signType, signAlgorithm, signMode, signaturesTableLocation,
        signaturesTableHeader, pdfInfoSignature, csvStampInfo, userRequiresTimeStamp,
        timeStampGenerator);

    final String signaturesSetID = String.valueOf(System.currentTimeMillis());
    SignaturesSet signaturesSet = new SignaturesSet(signaturesSetID, commonInfoSignature,
        new FileInfoSignature[] { fileInfo });

    String timestampUrlBase = null;
    signaturesSet = plugin.signDocuments(signaturesSet, timestampUrlBase);
    StatusSignaturesSet sss = signaturesSet.getStatusSignaturesSet();

    if (sss.getStatus() != StatusSignaturesSet.STATUS_FINAL_OK) {
      System.err.println("Error General MSG = " + sss.getErrorMsg());
      if (sss.getErrorException() != null) {
        sss.getErrorException().printStackTrace();
      }
      throw new Exception(sss.getErrorMsg());
    } else {
      FileInfoSignature fis = signaturesSet.getFileInfoSignatureArray()[0];
      StatusSignature status = fis.getStatusSignature();
      if (status.getStatus() != StatusSignaturesSet.STATUS_FINAL_OK) {
        if (status.getErrorException() != null) {
          status.getErrorException().printStackTrace();
        }
        System.err.println("Error Firma 1. MSG = " + status.getErrorMsg());
        throw new Exception(status.getErrorMsg());
      } else {
        File dest = new File(pdfdest);
        status.getSignedData().renameTo(dest);
        System.out.println();
        System.out.println();
        System.out.println(" Guardada Firma a " + dest.getAbsolutePath());
        System.out.println(" Tamany " + dest);
      }
    }
  }
```

6.4.2.- Cridada firma PAdES i XAdES

Fa ús del mètode anterior per realitzar una firma de tipus PAdES i una de tipus XAdES. També es mostra com emprar plugin de MiniAppletInServer i Passarel·la PortaFIB.

```
      String pdfsource = "sample.pdf";
      String pdfdest = "sample_signed.pdf";

      String propertyKeyBase = "org.fundaciobit.example.";

      ISignatureServerPlugin plugin;
//      Emprant Plugin  PortaFIBSignatureServerPlugin
      {
        Properties prop = new Properties();

        prop.setProperty("org.fundaciobit.example.plugins.signatureserver.portafib.ap
        i_passarela_url","http://localhost:8080/portafib/ws/v1/PortaFIBPassarelaDeFir
        maEnServidor");
        prop.setProperty("org.fundaciobit.example.plugins.signatureserver.portafib.ap
        i_passarela_username","fundaciobit_usrapp");
        prop.setProperty("org.fundaciobit.example.plugins.signatureserver.portafib.ap
        i_passarela_password","fundaciobit_usrapp");
        plugin = new PortaFIBSignatureServerPlugin(propertyKeyBase, prop);
      }

//      Emprant Plugin MiniAppletInServerSignatureServerPlugin
//      {
//        Properties prop = new Properties();
//
prop.setProperty("org.fundaciobit.exemple.signatureserverplugins.2.plugins.signaturese
rver.miniappletinserver.base_dir","C:/tmp/miniappletinserver");
//        plugin = new MiniAppletInServerSignatureServerPlugin(propertyKeyBase,
prop);
//      }

      IRubricGenerator rubricGenerator = null;

      // PAdES SIGN
      {
        String signType = FileInfoSignature.SIGN_TYPE_PADES;
        int signMode = FileInfoSignature.SIGN_MODE_IMPLICIT;
        boolean userRequiresTimeStamp = false;
        signFile(pdfsource, pdfdest, signType, signMode, userRequiresTimeStamp,
            rubricGenerator, plugin);
      }

      // XAdES Attached SIGN
      {
        String signType = FileInfoSignature.SIGN_TYPE_XADES;
        int signMode = FileInfoSignature.SIGN_MODE_IMPLICIT; // Attached
        // FileInfoSignature.SIGN_MODE_EXPLICIT; // Detached
        boolean userRequiresTimeStamp = false;
        String xadesAttachedDest = "hola.pdf.xades_attached.xml";
        signFile(pdfsource, xadesAttachedDest, signType, signMode,
            userRequiresTimeStamp, rubricGenerator, plugin);
      }
```