Porta Firmas

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Jose Manuel Vives Olaria
i dirigit per
Jordi Pons Aróztegui
Bellaterra, 22 de Juny de 2010

El sotasignat, Jordi Pons Aróztegui,

Professor/a de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jose Manuel Vives Olaria

I per tal que consti firma la present.

Signat: Jordi Pons Aróztegui

Bellaterra, 22 de Juny de 2010

El sotasignat, Juan Carlos Vera Hernández,

de l'empresa UNIT4

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva supervisió mitjançant conveni PROJECTE FINAL DE CARRERA firmat amb la Universitat Autònoma de Barcelona.

I per a que consti firma la present.

Signat: Juan Carlos Vera Hernández

Bellaterra, 18 de juny de 2010

Índex

Índex de figures:

Índex de Taules:

1. Introducció

Aquest primer capítol de la memòria ens servirà per introduir el Projecte Final de Carrera Porta Firmas. S'especificarà quin va ser l'origen del projecte i els seus objectius. Finalment, descriurem l'estructura de la memòria.

1.1. Presentació del projecte

El projecte **Porta Firmas** s'emmarca dins d'un conveni de col·laboració entre la Universitat Autònoma de Barcelona (UAB) i l'empresa CCS Agresso (actualment anomenada Unit4 Iberica) per realitzar el Projecte Final de Carrera (PFC) en l'empresa. CCS Agresso és un fabricant de programari de gestió (ERP, CRM, RRHH, BI, CPM) per a empreses, organismes públics i sanitat, tant en modalitat de venda com de pagament per ús. L'empresa té diferents seus en vàries Comunitats Autònomes d'Espanya i també a l'estranger (Guinea Equatorial i Moçambic). El projecte s'ha desenvolupat a l'empresa ubicada al Polígon Industrial Santiga de Barberà del Vallès. Dins de l'empresa, el projecte s'ha desenvolupant dins de l'àrea de consultoria, supervisat per Juan Carlos Vera Hernández.

1.2. Objectiu general

El projecte a desenvolupar el podem catalogar com un projecte de manteniment/millora d'una aplicació de signatura electrònica i gestió de documents.

Aquest projecte vol millorar certs aspectes que van quedar per desenvolupar en l'aplicació anomenada SIFE (Sistema Integral de Firma Electrònica) desenvolupada per SPAI anteriorment, i que no es van poder arribar a implementar. Tot i que inicialment es deia SIFE (Sistema Integral de Firma Electrónica), el nom que s'ha acabat imposant ha sigut el de Porta Firmas. A partir d'ara s'utilitzarà el terme Porta Firmas per referir-nos a l'aplicació.

Donada la naturalesa d'aquest projecte, no partirem des de zero en el desenvolupament. Inicialment partim d'un projecte desenvolupat mitjançant l'IDE Eclipse, en un entorn JAVA que utilitza varis frameworks que veurem més endavant. D'aquesta manera, s'ha de tenir en compte tant la feina que ja hi ha feta com la manera d'afegir al projecte les millores que es demanen. Així, el que es vol desenvolupar és una millora d'una aplicació Client-Servidor on els usuaris poden gestionar la firma electrònica de documents mitjançant un navegador web de manera fàcil i ràpida. Els clients podran firmar, validar i gestionar els seus documents mitjançant un navegador web. Addicionalment, el servidor també dóna Web Services amb els quals altres aplicacions es poden comunicar amb el sistema.

Algunes de les millores a desenvolupar en l'aplicació Porta Firmes són l'identificació dels usuaris mitjançant certificat (per exemple, el DNI electrònic), afegir més tipus de signatura de documents (XMLDsig, XAdES) amb les seves variants (Implícita, Explicita Attached, Detached, etc), entre d'altres.

1.3. Introducció a l'estat de l'art

Actualment, la llei que regula la firma electrònica a Espanya és la "Ley 59/2003, de 19 de diciembre, de firma electrónica". També hi ha una altra llei de regulació de l'ús de la firma electrònica a les administracions públiques que és la "Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos".

La Llei 59/2003 bàsicament defineix què és la firma electrònica, quines conseqüències legals comporta la firma d'un document i qui pot expedir certificats electrònics, entre d'altres. La llei 11/2007 pretén impulsar la promoció de serveis que utilitzin la identificació i autentificació electrònica per part dels ciutadans a l'hora de fer tràmits amb les Administracions públiques.

Per tant, la signatura electrònica d'un document ha de garantir que té tota la validesa i seguretat necessària. Per tal de poder signar un document necessitem un certificat digital (com ara X.509, el DNI electrònic, IDCat, FNMT, entre altres) que hagi estat expedit en una oficina emissora de certificats digitals reconeguda per la llei 59/2003, i que aquesta mateixa oficina certificadora també ofereixi serveis de validació de certificats.

Per exemple, en l'aplicació Porta Firmas s'haurà d'implementar la plataforma de validació amb @Firma. En canvi, l'aplicació ja té implementada la integració amb CATCert, que ofereix la PSIS (Plataforma de Serveis d'Identificació i Signatura) per a la validació de certificats i signatures digitals.

Pel que fa a aplicacions similars, CATCert ha desenvolupat l'Oficina Virtual de Signatura per donar suport de signatura digital en les administracions públiques mitjançant un portal web. Aquesta aplicació, i similars, podem dir que són la competència del Porta Firmas.

Alguns enllaços amb més informació:

- El servei SIGNA: Sala virtual de signatura. Conté informació del servei SIGNA de CATCert

  Enllaç: http://www.catcert.cat/web/cat/1_4_12_signa.jsp

- La plataforma eaCat. El portal per accedir a l'Oficina Virtual de Signatura

  Enllaç: https://www.eacat.cat/

- Vídeo exemple de signatura amb l'Oficina Virtual de Signatura. L'aplicació Porta Firmes serà molt similar al que el servei SIGNA ofereix.

  Enllaç: http://www.catcert.cat/descarrega/ovs/video_signatura/index.html

1.4. Estructura de la memòria

La memòria s'ha estructurat de la següent manera:

En l'apartat 1, hem vist una introducció al projecte Porta Firmas, objectius i estat de l'art de la signatura electrònica.

En l'apartat 2, farem un repàs del estudi de viabilitat del projecte i la planificació temporal.

En l'apartat 3 veurem l'estat inicial en que es trobava el Porta Firmas. Farem una breu introducció a l'arquitectura i les tecnologies que utilitza.

En l'apartat 4, veurem l'anàlisi de requisits inicials del Porta Firmas per, finalment veure quins requisits s'ha decidit implementar en aquest projecte

En l'apartat 5, explicarem el disseny i d'implementació de cada requisit. També analitzarem la solució i possibles problemes i alternatives que tinguin.

En l'apartat 6, mostrarem un resum del pla de proves per comprovar el funcionament de les solucions aportades.

En l'apartat 7, farem un repàs de quines eines s'han fet servir.

Finalment, en l'apartat 8, veurem les conclusions del projecte, un resum de quins han sigut els objectius assolits, possibles solucions i alternatives de futur i la valoració personal.

La bibliografia conté el llistat amb les referències bibliogràfiques i enllaços a Internet.

En el glossari, tenim un recull dels termes i sigles més utilitzades.

2. Estudi de viabilitat del projecte

2.1. Objectius del Projecte

El projecte que es vol desenvolupar és un projecte que es pot catalogar com de manteniment, millora i/o ampliació. Això implica una certa pèrdua de llibertat, ja que s'han d'implementar les millores mantenint la idea del sistema que s'havia pensat. A més, cal revisar cada pas i mantenir el sistema estable, cada vegada que s'afegeix alguna funcionalitat nova.

2.2. Descripció de la situació inicial

El projecte actual ha estat desenvolupat sobre una pila de frameworks, buscant fer una aplicació de tipus SOA (Service Oriented Service Architecture), utilitzant el paradigma MVC (Model, View, Controller), fent el client compatible amb els principals navegadors web actuals i suportar tant els Sistemes Operatius Windows com Linux.

El projecte es va donar per finalitzat cap a mitjans de 2007. Això fa que moltes de les llibreries i frameworks utilitzats disposin de noves versions i que s'hagi d'estudiar la possibilitat d'actualitzar les dependències d'alguna manera. A més, cal destacar la fluixa documentació (per exemple, en les taules de la Base de Dades n'hi ha que hem de fer suposicions sobre el seu ús) i la poca o fins i tot nul·la utilització de tests de prova i validació de codi com ara amb JUnit.

2.3. Especificacions del sistema i les aplicacions

La part del client ha de poder ser compatible preferiblement amb qualsevol navegador actual (Internet Explorer, Firefox, Chrome, entre d'altres). En la part del servidor, s'utilitza l'Apache Tomcat, branca 5.5 (que no és la branca actual, 6.0) i pel que fa a la BD, pel desenvolupament s'utilitza el SGBD de MS SQL 2000 i/o SQL Express 2005, però hi ha la possibilitat d'utilitzar altres BD (Oracle i MySQL).

2.4. Viabilitat tècnica

Es necessiten coneixements que van des de les Bases de Dades, seguretat, xifratge, arquitectures Client-Servidor SOA, programació Web, manteniment, testing i cal fer també la documentació.
Tècnicament és viable, però la quantitat de coneixements i detalls específics que s'han de tenir en compte poden fer perillar el compliment dels objectius del projecte.

2.5. Viabilitat operativa

L'estada a l'empresa és d'unes 550 hores, temps que s'haurà d'aprofitar bé per assolir els principals objectius del projecte.

2.6. Viabilitat econòmica

En principi, es treballarà amb software lliure i el maquinari informàtic necessari serà donat per l'empresa. El cost, per tant, és el cost de les hores del programador, més el del local, HW i altres serveis (connexió a INTERNET, menjador, etc).

2.7. Viabilitat legal

Complir amb els requisits de la Ley 59/2003, de les entitats certificadores (CATCert, @Firma).

2.8. Alternatives del projecte

El servei SIGNA: Sala virtual de signatura. Aquest servei que ofereix CATCert és molt similar al que es vol implementar. Podem dir que l'aplicació Porta Firmas i el servei SIGNA competeixen pel mateix mercat, el de les administracions públiques i la solució d'una aplicació de gestió i signatura de documents electrònics.

2.9. Riscos i beneficis

El projecte vol millorar certs aspectes del Porta Firmas, per la qual cosa en principi no suposa cap risc elevat. Els principals beneficis són una millora en l'ús i serveis de l'aplicació, amb la que es vol millorar de cara a una possible futura actualització.

2.10. Conclusions sobre la viabilitat

Finalment, després d'analitzar els diferents aspectes de la viabilitat, els riscos i els beneficis, podem concloure que el projecte és viable.

2.11. Planificació temporal del treball

Com es mostra en la figura 1, s'ha fet una planificació temporal molt general, sense definir tasques i dates més concretes. Això és degut a que part de la feina per fer és la revisió del codi i veure com es poden afegir les millores que es demanen. Moltes de les tasques són difícils de planificar per endavant i per això la planificació és molt general.

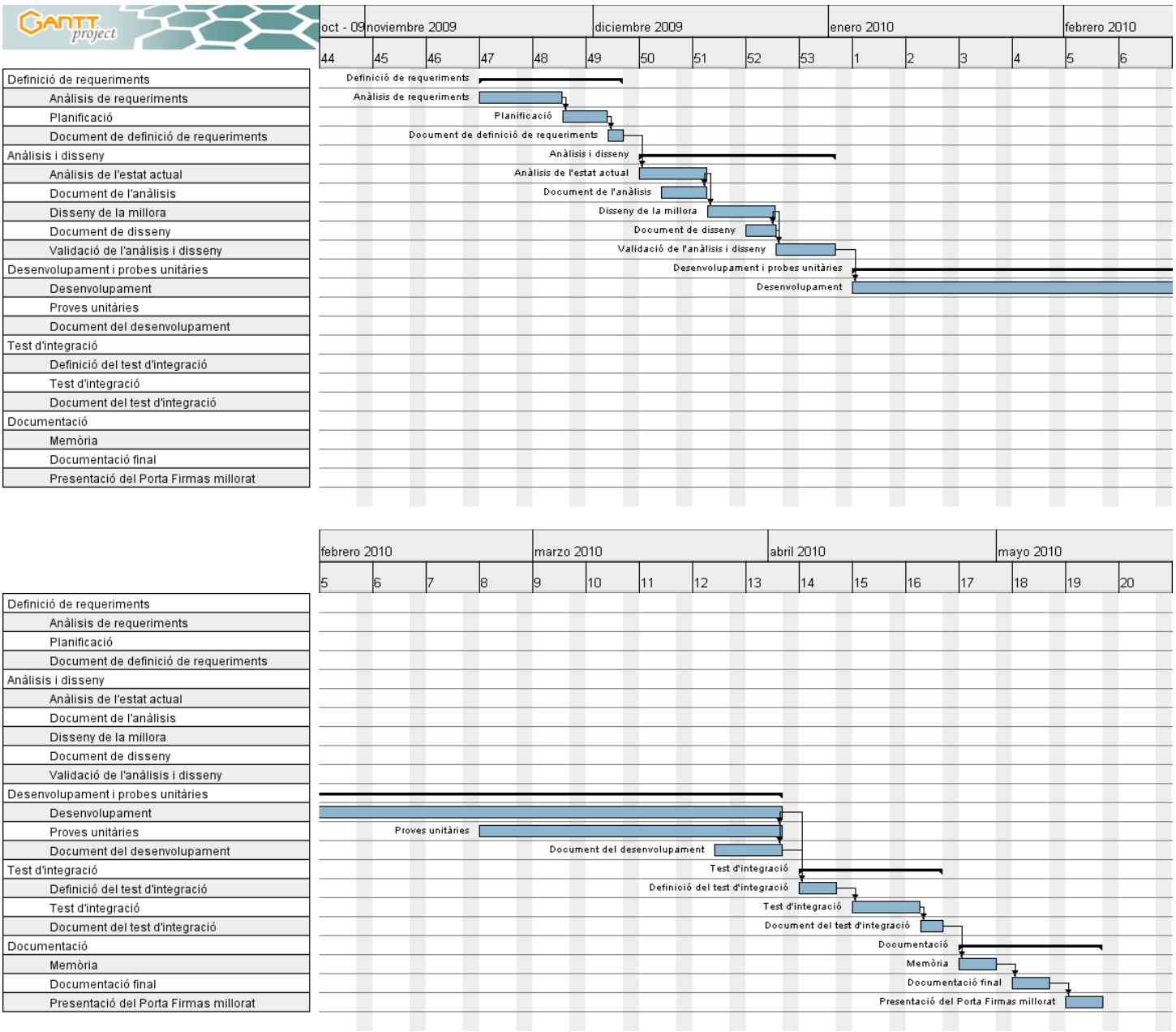

Figura 1 – Planificació del projecte

Inici: 16 de novembre de 2009 **Final:** 14 de maig de 2010

Com podem veure, la data d'inici va ser el 16 de novembre de 2010 i la data prevista de finalització era pel 14 de maig del 2010.

2.12. Altres comentaris

Com ja s'ha comentat, el projecte a desenvolupar el podem emmarcar com a manteniment i/o millora i com a tal, pateix de certes mancances com ara falta de documentació, no actualitzada i incomplerta, baralles amb multitud de detalls no especificats amb les llibreries/frameworks, sobredimensionament de projectes, poc testejat, etc.

Tot i aquests problemes, els projectes de manteniment són una part important necessària en el món de la Informàtica. Tenen les seves pròpies regles i solucions, s'ha d'avaluar què modificar, dissenyar i implementar la modificació i com garantir que la resta continuarà funcionant correctament.

3. Anàlisis de la situació inicial

L'aplicació Porta Firmas està dissenyada entorn a tres grans idees i/o tecnologies. La primera, utilitzar el framework d'Hibernate per tenir un accés independent de la Base de Dades utilitzada. La segona, utilitzar el framework MVC Struts 2 per la creació d'un Portal Web dinàmic amb que els usuaris de l'aplicació Porta Firmas es connectaran a l'hora de gestionar els documents mitjançant un navegador web. La tercera és la utilització de Web Services. En quant als Web Services, els podem tornar a classificar en 2 tipus: Web Services que facilitarà l'aplicació Porta Firmas i els Web Services externs que utilitza el Porta Firmas. Per signar documents, s'utilitzaren el applets de CATCert i @Firma amb els certificats vàlids que el client tingui.

3.1. Visió general de la tecnologia

Com ja hem dit abans, partim des d'un projecte ja iniciat. El projecte Porta Firmas està desenvolupat sobre una pila de frameworks. La següent figura mostra un esquema de l'arquitectura.

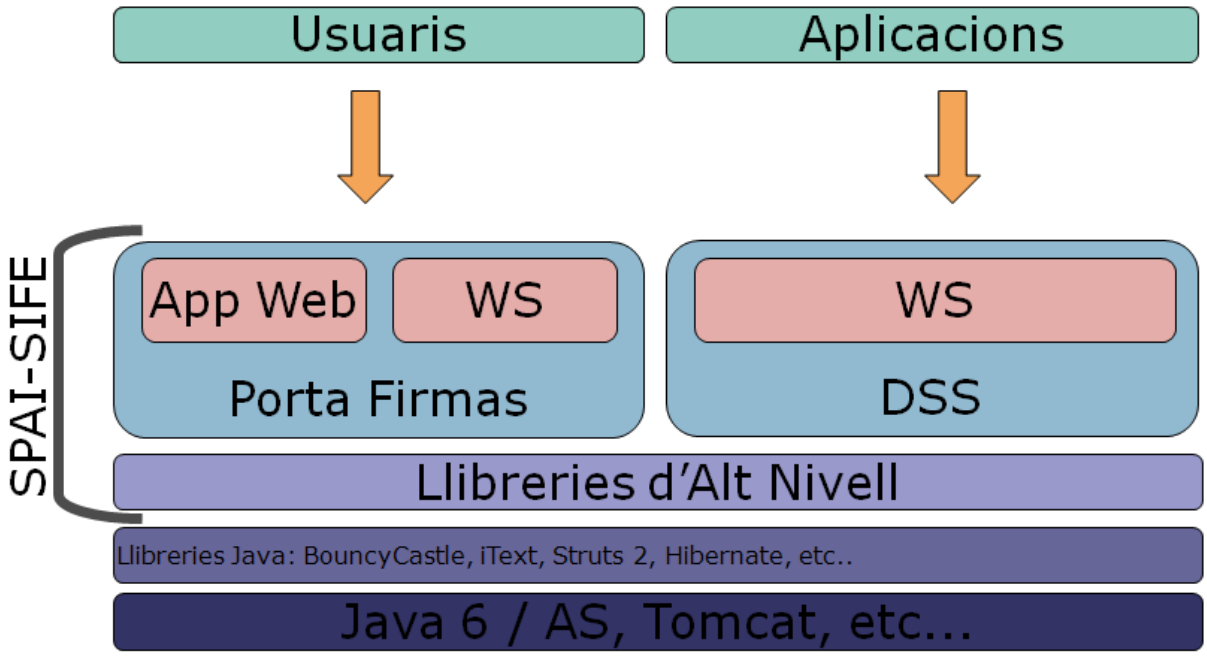

Figura 2 – Arquitectura del Porta Firmas

Com podem veure, hi ha 2 maneres d'interactuar amb la plataforma Porta Firmas.

En la primera, els usuaris accedeixen al portal web per gestionar els seus documents. En la segona les aplicacions es comuniquen amb la plataforma mitjançant Web Services. Hi ha diverses llibreries d'alt nivell desenvolupades per SPAI (com LibreriasSPAI.jar) que ofereixen, entre altres funcionalitats, accés als keystore, enviar XML, etc. El portal web ha estat desenvolupat mitjançant Struts2 i els Web Services mitjançant els frameworks de Codehaus Xfire (pel que fa a DSS) i Apache Axis. La plataforma corre sobre un servidor de servlets que en aquest cas és Tomcat. Per al SGBD, amb la utilització de la llibreria Hibernate, en teoria es pot utilitzar qualsevol SGBD que sigui suportat per aquesta llibreria. A la pràctica, en el Porta Firmas hi ha suport per a MS SQL i Oracle.

3.2. Hibernate i la Base de Dades

Gràcies a la utilització d'Hibernate podem desvincular la Base de Dades utilitzada en concret i fer servir objectes JAVA per fer consultes, en comptes de les habituals sentències SQL. Per fer-ho, primer generarem una sèrie d'arxius declaratius (XML) que permeten establir la relació entre un objecte JAVA i el mapa de la BD relacional. Un altre avantatge és que permet desvincular la BD en concret de les consultes, ja que Hibernate les generarà automàticament per cada BD (com ara per MS SQL, Oracle, PostgreSQL, etc).

3.2.1. Model físic

En aquest apartat veurem el model físic de la Base de Dades del Porta Firmas. En concret, el Porta Firmas disposa de 2 scripts de creació de taules:

1.- Per MS SQL, el fitxer: "[2009-02-16] ScriptCreacionTablasSQLServer.sql"

2.- Per Oracle, el fitxer: ORACLE_Tablas.sql

En l'apartat 3.2.2, veurem l'esquema relacional de la Base de Dades. La **Figura 3** mostra el diagrama de les taules que es genera amb MS SQL 2000.

En l'apartat 3.2.3 veurem un breu resum de les taules que formen la base de dades i quina funció bàsica realitzen.

3.2.2. Diagrama de la base de dades

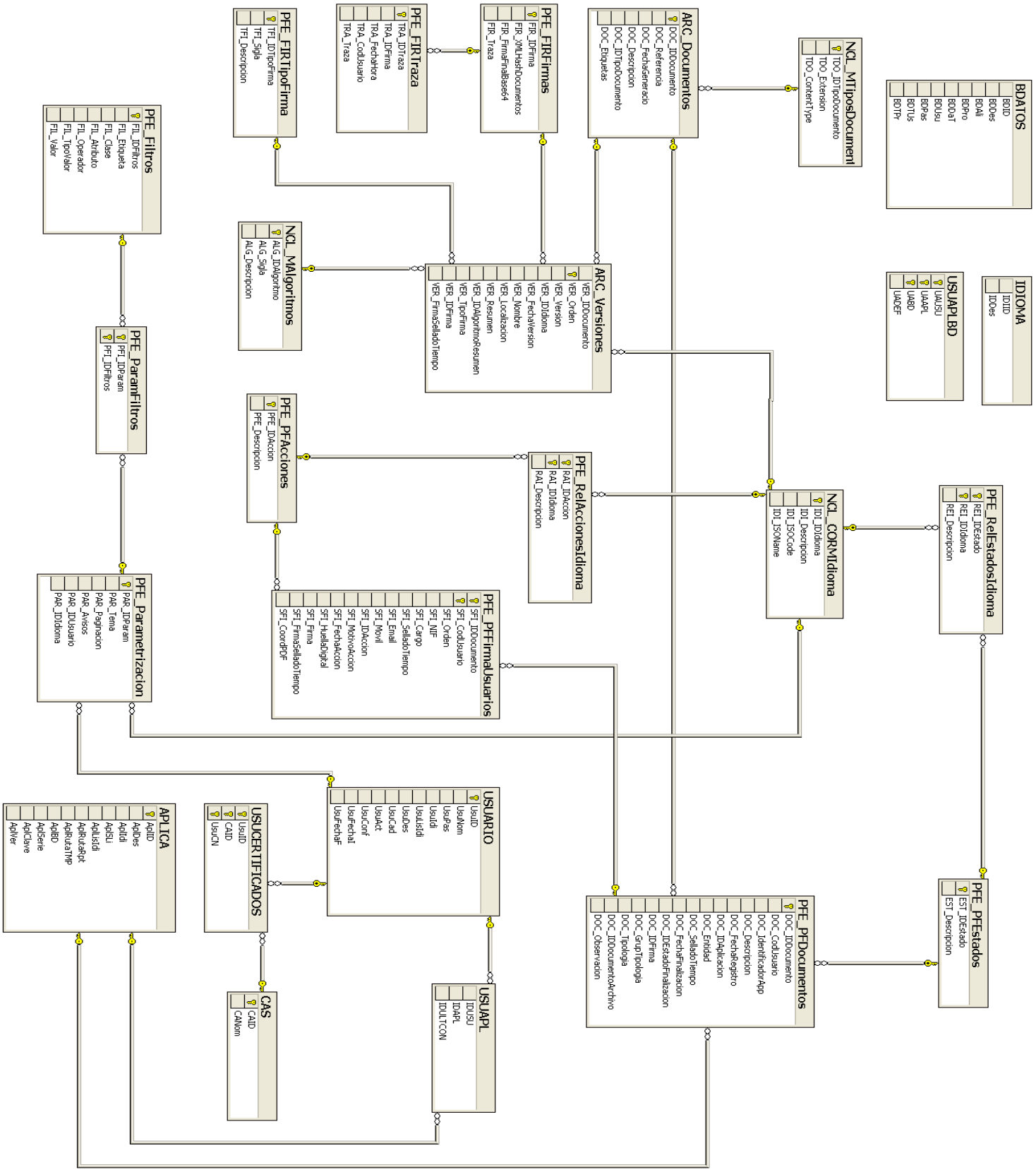

Figura 3 - Diagrama de les Taules de la BD del Porta Firmas

3.2.3. Descripció de les taules de la BD

A continuació es descriuen breument les taules de la Base de dades. Estan dividides en dos grups: Taules de Gestor i Taules de Porta Firmas. Falten per comentar 3 taules que són IDIOMA, BDATOS i USUAPLBD. Aquestes taules són d'ús intern per a l'administrador de la base de dades del PortaFirmas. En total, Porta firma conté 25 taules, però no totes s'utilitzen.

3.2.3.1. Taules del Gestor

En aquest apartat es descriuen les taules del **Gestor d'Aplicacions** que s'utilitzen en el Porta Firmas. El Porta Firmas pot treballar amb altres aplicacions que suportin la signatura electrònica de documents. Aquestes taules, guarden i gestionen aquestes interaccions.

Taula 1 - Descripció de les Taules del Gestor del Porta Firmas

Taula	Descripció
APLICA	Guarda el codi i la descripció de les apliacions. Codi per Porta Firmas = 72, Codi per Gestor = 0.
USUARIO	Guarda la informació relacionada amb els usuaris com ara codi d'usuari, nom, contrasenya i descripció.
USUAPL	Es guarda la relació entre un usuari i quina aplicació utilitza.

3.2.3.2. Taules del Porta Firmas

A continuació es descriuen les taules utilitzades pel **Porta Firmas**. Bàsicament hi ha la gestió dels documents, les signatures, els certificats, els idiomes de l'aplicació, les opcions de configuració (anomenades parametrización), entre d'altres.

Taula 2 - Descripció de les Taules del Porta Firmas

Taula	Descripció
NCL_CORMIDIOMA	Guarda les relacions dels idiomes amb els codis ISO.
NCL_MTIPOSDOCUMENTO	Guarda les extensions de fitxers més comunes.
NCL_MALGORITMOS	No s'utilitza.
PFE_PFESTADOS	Guarda els possibles estats que es pot trobar un document.
PFE_RELESTADOSIDIOMA	No s'utilitza.
PFE_PFACCIONES	No s'utilitza.
PFE_RELACCIONESIDIOMA	No s'utilitza.
PFE_FIRTIPOFIRMA	Tipus de signatura suportats.
ARC_DOCUMENTOS	Guarda informació dels documents.
ARC_VERSIONES	Guarda informació de les versions dels documents.
PFE_FIRFIRMAS	No s'utilitza. Es volia fer servir per guardar els HASH.
PFE_FIRTRAZA	Traces dels processos de signatura.
PFE_PFDOCUMENTOS	Relaciona els documents amb la taula APLICA.
PFE_PFFIRMAUSUARIOS	Guarda la informació de la crida al Web Service registrarXML, que afegeix un nou document per signar.
USUCERTIFICADOS	Relaciona un usuari amb un certificat (el seu CN)
CAS	Guarda una llista d'Autoritats Certificadores
PFE_PARAMETRIZACION	No s'utilitza. Guarda les opcions de configuració de cada usuari.
PFE_PARAMFILTROS	No s'utilitza. Relaciona PFE_PARAMETRIZACION i PFE_FILTROS.
PFE_FILTROS	No s'utilitza. Guarda els filtres.

3.3. Struts 2 i el portal web

El portal web de l'aplicació Porta Firmas està creat amb el framework Struts 2. Desenvolupat per Apache, Struts 2 és un framework d'aplicació web de codi per al desenvolupament d'aplicacions web Java EE.

3.3.1. Arquitectura de Struts 2

Struts 2 és un framework de tipus PULL-MVC. Això vol dir que per poder veure una dada, primer s'ha d'extreure mitjançant una Action (acció). Les sigles MVC corresponent a Model-View-Control, un patró de disseny que està molt estès entre les aplicacions web. Struts 2 també utilitza dins de la seva arquitectura, tecnologies JEE estàndard com Java Filters, JavaBeans, ResourceBundles, Locales, XML, etc. La **Figura 4** mostra més en detall com esta estructurat el framework.

A continuació, detallarem l'arquitectura seguint els passos que es donen quan el client genera una petició:

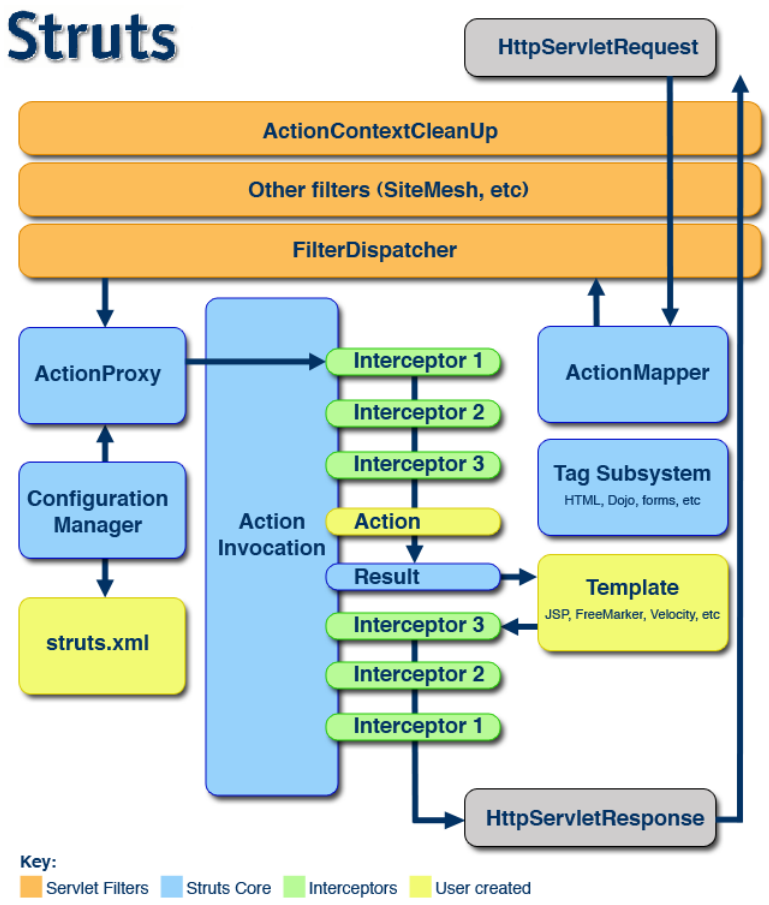

Figura 4 – Arquitectura de Struts 2

1. El cicle normal d'execució de Struts2 comença quan el client sol·licita una petició al servidor. Això dóna lloc a invocar el contenidor de servlets (com ara Tomcat) que al seu torn passa a través de la cadena de filtres estàndard.

2. El filtre anomenat FilterDispatcher consulta el ActionMapper per determinar si l'acció sol·licitada pot ser invocada.

3. Si ActionMapper troba una Action per ser invocada, llavors FilterDispatcher delega el control a ActionProxy.

4. ActionProxy llegeix el fitxer de configuració anomentat struts.xml. ActionProxy crea una instància de la classe ActionInvocation i li passa el control.

5. ActionInvocation és el responsable de l'execució d'ordres segons un patró establert previament. Així, invoca els interceptors un a un (si cal) i després invoca a l'Action.

6. Quan l' Action retorna el seu resultat, ActionInvocation és el responsable de comparar-lo amb els resultats propis associats amb el codi de resultat de l' Action assignada en struts.xml.

En resum, Struts2 mapeja un conjunt d' Actions (que són classes JAVA) amb un conjunt de classes per visualitzar el resultat de l'acció (JSP, Velocity, etc) i aquest resultat, s'envia en codi HTML l'usuari per ser visualitzat. A més a més, Struts2 també disposa d'un conjunt d'interceptors que s'executaràn un a un i ens serviran per fer feines de Loggin, Validació de dades, File Uploading, etc.

Més en detall, les Actions són classes JAVA que implementen una determinada acció en el model de negoci (com pot ser, buscar a la BD les factures del mes, etc) i retornarà una cadena amb el tipus de resultat obtingut (per exemple, INPUT, ERROR, SUCCESS). Llavors, depenent d'aquest tipus retornat i del fitxer de configuració struts.xml que mapeja els resultats de les Actions, es tria amb quina classe s'ha de visualitzar els resultats. Un altre fitxer de configuació important és web.xml, que ens serveix per filtrar les peticions que li arriben al servidor de servlets i enviar-les allà on toca.

Pot semblar un pel complicada, però el fet de separar les accions de negoci a realitzar de la manera de mostrar les dades, permet (si es fa bé) testejar individualment cada Action i cada Interceptor mitjançant, per exemple, Tests d'Unitat.

3.3.2. Mapa de les Actions del Porta Firmas

A continuació es mostra el fitxer web.xml, que mapeja les Actions amb la visualització segons el resultat de cada Action.

```
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
<package name="web" namespace="/web" extends="struts-default">
        <action name="desktop" class="es.spai.portafirmas.web.action.DesktopAction">
                <result>/WEB-INF/web/desktop.jsp</result>
        </action>
        <action name="modelDesktop" class="es.spai.portafirmas.web.action.ModelDesktopAction">
                <result>/WEB-INF/web/index.jsp</result>
        </action>
         <action name="login" class="es.spai.portafirmas.web.action.LoginAction">
                <result name="input">/WEB-INF/web/login.jsp</result>
         <result type="redirect-action">desktop</result>
         </action>
         <action name="logout" class="es.spai.portafirmas.web.action.LogoutAction">
                 <result>/WEB-INF/web/login.jsp</result>
        </action>
        <action name="pendents" class="es.spai.portafirmas.web.action.PendentsAction">
                <result>/WEB-INF/web/pendents.jsp</result>
        </action>
        <action name="rebutjats" class="es.spai.portafirmas.web.action.RebutjatsAction">
                <result>/WEB-INF/web/rebutjats.jsp</result>
        </action>
        <action name="rebutjatsTable" class="es.spai.portafirmas.web.action.TableAction">
                <result>/WEB-INF/web/rebutjatsTable.jsp</result>
        </action>
        <action name="pendentsTable" class="es.spai.portafirmas.web.action.TableAction">
                <result>/WEB-INF/web/pendentsTable.jsp</result>
        </action>
        <action name="signats" class="es.spai.portafirmas.web.action.SignatsAction">
                <result>/WEB-INF/web/signats.jsp</result>
        </action>
```

```
        <action name="signatsTable" class="es.spai.portafirmas.web.action.TableAction">
                <result>/WEB-INF/web/signatsTable.jsp</result>
        </action>
        <action name="configuracio" class="es.spai.portafirmas.web.action.ConfiguracioAction">
                <result>/WEB-INF/web/configuracio.jsp</result>
        </action>
        <action name="modal" class="es.spai.portafirmas.web.action.ModalAction">
                <result>/WEB-INF/web/modal.jsp</result>
        </action>
        <action name="modalRebuig" class="es.spai.portafirmas.web.action.ModalRebuigAction">
                <result>/WEB-INF/web/modalRebuig.jsp</result>
        </action>
        <action name="verify" class="es.spai.portafirmas.web.action.VerifyAction2">
                <result>/WEB-INF/web/verify.jsp</result>
        </action>
         <action name="signedDetails" class="es.spai.portafirmas.web.action.SignedDetailsAction">
                <result>/WEB-INF/web/signedDetails.jsp</result>
        </action>
         <action name="ayuda_pendents" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_pendents.jsp</result>
        </action>
        <action name="ayuda_signats" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_signats.jsp</result>
        </action>
        <action name="ayuda_rebutjats" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_rebutjats.jsp</result>
        </action>
        <action name="ayuda_configuracio" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_configuracio.jsp</result>
        </action>
        <action name="ayuda_newdocument" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_newdocument.jsp</result>
        </action>
        <action name="ayuda_pdf" class="es.spai.portafirmas.web.action.AyudasAction">
                <result>/WEB-INF/web/ayuda_pdf.jsp</result>
        </action>
        <action name="guardaDOCSignat"class="es.spai.portafirmas.web.action.GuardaDOCSignatAction">
                <result>/WEB-INF/web/guardaDOCSignat.jsp</result>
        </action>
        <action name="obtenerObsFirma" class="es.spai.portafirmas.web.action.ObtenerObsFirma">
                <result>/WEB-INF/web/obtenerObsFirma.jsp</result>
        </action>
        <action name="inserirdoc" class="es.spai.portafirmas.web.action.InsertarDocumentoAction">
                <result>/WEB-INF/web/insertarDocumento.jsp</result>
        </action>
</package>

</struts>
```

Com podem veure, totes les accions retornen per defecte *"SUCCESS"*, mostren la pàgina <result> corresponent. Només la *LoginAction* té un altra visualització en cas que l'acció retorni *"INPUT"*, que es el cas en que hi ha hagut algun error a l'hora d'autentificar l'usuari (nom d'usuari i password incorrectes, etc). L'estructura és molt simple, ja que no s'utilitzen casos específics d'error.

3.3.3. Diagrama de casos d'ús de la web

A continuació es mostra el diagrama de casos d'ús del portal web de la plataforma Porta Firmas.

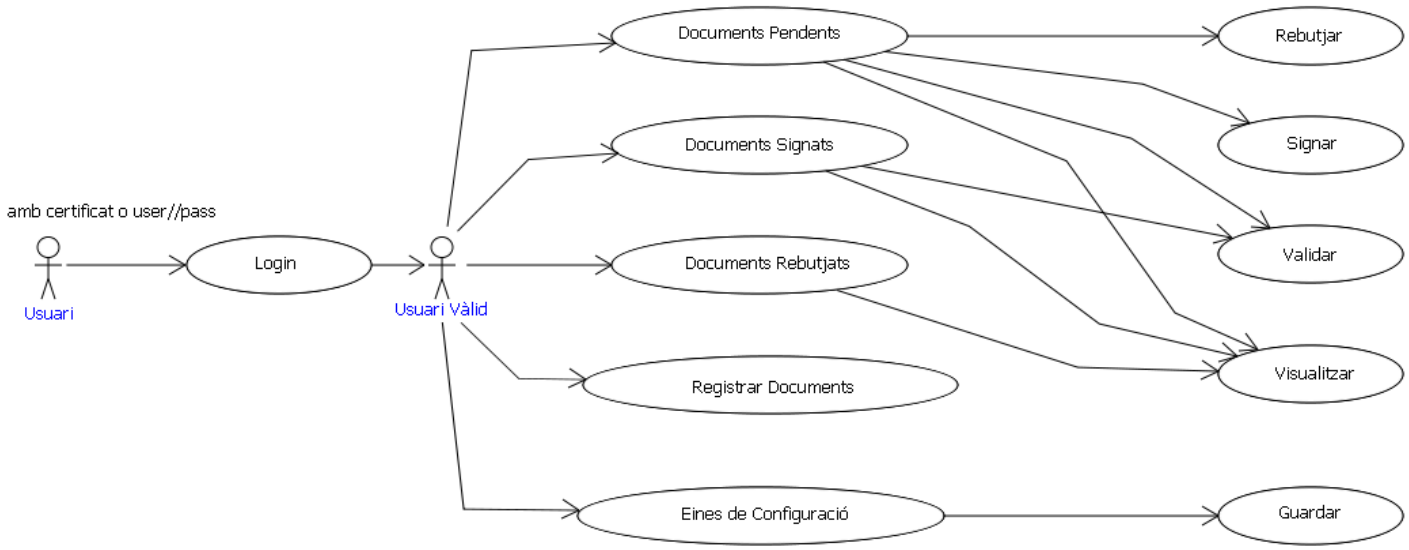

Figura 5 - Diagrama de casos d'ús del Portal Web

Com es pot veure a la figura anterior, l'usuari primer s'autentifica mitjançant un procés de login. Aquest pot entrar el seu nom d'usuari i contrasenya o bé un certificat (per exemple, el DNIe). Una vegada l'usuari és valida, es carrega un menú amb les diferents opcions de que disposa. Així, l'usuari pot consultar els documents pendents de signar, els documents signats, els documents rebutjats, pot registrar nous documents a signar i pot canviar les opcions de configuració.

Quan s'accedeix a opcions de configuració, es mostra un formulari per poder guardar la configuració.

Quan s'accedeix a registrar un nou document a signar, es mostra un formulari per afegir nous fitxers per signar entre un o varis usuaris.

Quan s'accedeix a documents pendents, podem rebutjar, signar, validar i visualitzar els documents. La signatura es realitza mitjançant els applets de CATCert o @Firma.

Quan s'accedeix a documents signats, podem validar i visualitzar els documents. La validació es fa mitjançant Web Servies.

Quan s'accedeix a documents rebutjats, solament podem visualitzar els documents.

L'acció de signar es realitza mitjançant els applets de CATCert o @Firma, la acció de validar es realitza mitjançant Web Services, i la de visualitzar la realitza Struts2

Les accions de signar, validar i visualitzar es realitzen mitjançant applets i Web Services. L'acció de signar, signa un document. L'acció de rebutjar, descarta la signatura d'un document.

3.3.4. El procés de signatura

El procés de signatura es realitza mitjançant un applet, depenent de quina plataforma tingui configurada el Porta Firmas en el fitxer de configuració "pf.config". Segons la plataforma triada, afectarà a la firma i validació de documents. Hi ha 2 plataformes: CATCert i @Firma. Inicialment, al Porta Firmas només estava suportada la signatura de documents PDF.

L'applet de CATCert (també anomenat eina web de signatura-e) ha sigut desenvolupat per l'Agència Catalana de Certificació i permet el seu ús a entitats públiques i administracions públiques de Catalunya. La versió que utilitza el Porta Firmas és actualment la 1.8 i s'han fet una sèrie de modificacions per evitar que demani el certificat cada vegada quan es signen varis documents, l'un darrera l'altre. Això fa més complicat la seva actualització, ja que l'última versió disponible és la 1.9.5, on s'han afegit algunes correccions i més tipus de signatura suportats. Els formats de signatura que suporta la versió 1.8 són PDF, CMS (amb les variants attached i detached), XMLDSig, XAdES-BES, XAdES-T (aquestes tres últimes amb les variants enveloped, enveloping i detached) i CAdES-BES (amb les variants attached, detached i detached en un PDF).

L'applet d'@Firma (també anomenat cliente firma) ha seguit desenvolupat per CSAE (Consejo Superior de Administración Electrónica) i per poder-lo utilitzar, s'ha d'estar donat d'alta a la xarxa SARA (Sistema de Aplicaciones y Redes para las Administraciones). Els formats de signatura que suporta l'applet versió 5.02 són PDF, ODF, CMS, CAdES, XMLDSig, XAdES (els dos últims amb les variants Enveloping, Enveloped, Detached i Externally Detached).

Com que cada applet es configura de manera diferent a l'hora de signar un document, s'han de tenir en compte les seves particularitats. Tota la configuració necessària es realitza mitjançant Javascript.

Per més informació, consultar els fitxers que s'han inclòs al CD adjunt a la memòria:

"Manual d'ús de l'eina web de signatura-e v1.8.pdf"

"Manual_Integrador_1_0_RC17.pdf"

3.4. Els Web Services

Un Web Services (en català, Serveis Web) és una col·lecció de protocols i estàndards que serveix per intercanviar dades entre aplicacions. Així, els Web Services faciliten la comunicació de diferents aplicacions mitjançant l'enviament de documents XML. D'aquesta manera, aplicacions que han estat desenvolupades en llenguatges de programació diferents i executades sobre qualsevol plataforma poden utilitzar els Web Services per intercanviar dades en una xarxa com Internet.

Aquesta gran interoperabilitat s'aconsegueix gràcies a l'adopció d'estàndards oberts. Les organitzacions OASIS i W3C són les responsables de l'arquitectura i reglamentació dels Serveis Web. Per garantir la interoperabilitat entre les diferents implementacions existeix un organisme, el WS-I, que és l'encarregat d'especificar de forma exhaustiva tots els aspectes d'aquests estàndards.

Per definir un Web Service, s'especifica mitjançant un document de tipus WSDL (Web Services Description Language) i que està en format XML. La versió 1.0 és la que existeix com recomanació del W3C. La versió 1.1 no va assolir aquesta condició. En les especificacions de la versió 2.0, hi apareix suport per RESTfull Web Services.

Els missatges que s'utilitzen per enviar i rebre informació dels Web Services són de tipus SOAP (Simple Object Access Protocol o Protocol Simple d'Accés a Objectes) que és un protocol de comunicació dissenyat per intercanviar missatges en format XML en una xarxa d'ordinadors, normalment sobre el protocol HTTP.

La manera ideal de funcionar d'un Web Service és que una aplicació que dóna Web Services, mostra el seu WSDL de manera que una altra aplicació client pot veure quins serveis li interessen. Un cop triat el servei que ens interessa, enviarem missatges en documents XML de tipus SOAP al Web Service. En qualsevol cas, el servidor de Web Services ens tornarà un missatge amb el resultat de l'operació.

Idealment, els Serveis Web que es donen haurien de ser sense estat. A la pràctica, és donen casos en que es necessita guardar una mena d'estat o de sessió per realitzar operacions que requereixin vàries crides a Web Services.

Per facilitar la implementació de Web Services, s'utilitzen frameworks com ara Apache Axis, Apache Axis 2, Xfire Codehaus, Apache CXF, entre d'altres.

Hi ha 2 maneres d'aproximació al disseny de Web Services:

- Bottom up. Primer s'implementa una classe en un llenguatge de programació (com per exemple, JAVA) i després s'utilitza un generador de documents WSDL per exposar els mètodes de la classe com a Web Service. Aquesta aproximació és més simple.
- Top Down: Primer es genera el document WSLD que després s'utilitzarà amb un generador de codi per crear el patró de la classe que s'haurà de completar. Generalment, és una aproximació més difícil però genera dissenys més clars.

Pel que fa als Web Service del Porta Firmas, n'utilitza i en dóna. Els veurem en més detall en els següents apartats.

3.4.1. Web Services del Porta Firmas

Generats a partir de la classe PortaFirmasService, mitjançant el framework de web Service Codehaus Xfire ofereix els següents serveis:

- registrarXML : Afegeix un nou document dintre de la plataforma i defineix la seqüència d'usuaris que han de signar el document. S'utilitza quan afegim un nou document per signar mitjançant el formulari del portal web del Porta Firmas.
- consultarXML : Retorna la informació i l'estat de la signatura del document especificat.
- getDocumentoXML : Recupera físicament un document, inclosa la signatura (si n'hi hagués).

Altres serveis com ara psisTimeStamp o validarXML estan a mig fer i no s'utilitzen. La següent figura mostra una representació gràfica del Web Service definit en el fitxer PortaFirmas.wsdl que genera l'Eclipse.

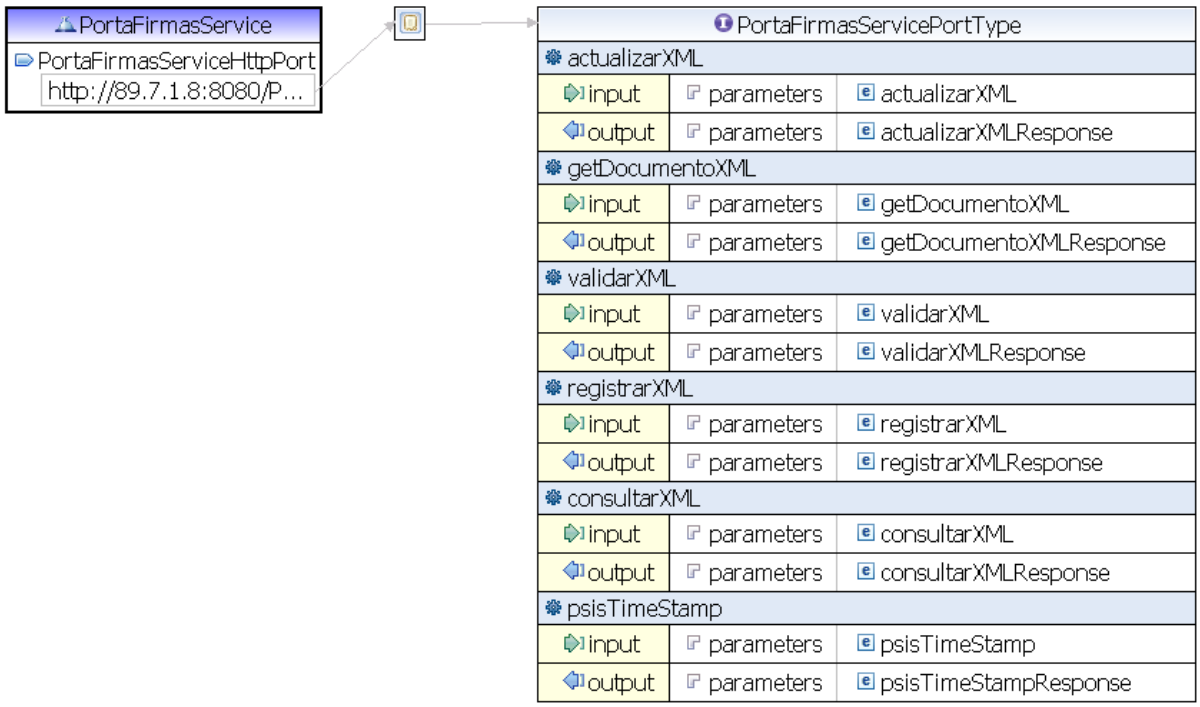

Figura 6 - Representació gràfica simple de PortaFirmas.WSDL

3.4.2. Web Services de la DSS a PSIS de CATCert

A continuació es descriuen els Web Services DSS (Digital Signature Services) amb els que el Porta Firmas dóna suport de signatura i validació amb CATCert (mitjançant la PSIS, la Plataforma de Serveis d'Identificació i Signatura) i també varis serveis d'API Criptogràfica.

Així, els Web Services són:

- PsisProxyService: Actua com a Proxy dels serveis DSS que proveeix la plataforma del PSIS. Així, executa una traducció de l'esquema DSS del Porta Firmas al de la plataforma PSIS per oferir serveis de validació i signatura de documents. En resum, connecta les peticions dels clients amb la PSIS de manera indirecta.
- SpaiDssService: Proporciona serveis DSS contra una API criptogràfica desenvolupada conjuntament amb els serveis web nomenats, sense necessitat d'usar DSS externs.

Ambdós utilitzen el framework de web Service Codehaus Xfire

3.4.3. Web Services de la DSS amb @Firma

Per validar una signatura amb la plataforma d'@Firma, hi ha un Web Service que crida al Web Service ValidarFirma que ofereix @Firma. Aquest Web Service està fet amb el framework Apache Axis. La següent figura mostra una representació gràfica del Web Service definit en el fitxer afirmaValidarFirma.WSDL que genera l'Eclipse.

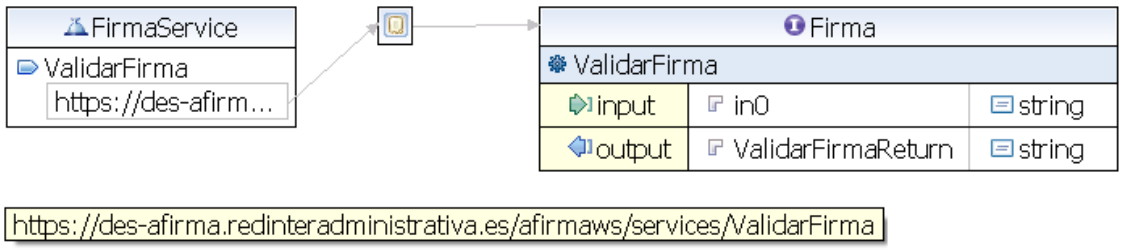

Figura 7 - Representació gràfica simple de afirmaValidarFirma.WSDL

4. Anàlisi de requisits

A continuació veurem la llista de requisits funcionals i no funcionals que inicialment es van voler desenvolupar per part del SIFE al Porta Firmas. Per cada requisit, comentarem la situació en la que es trobava per saber l'estat inicial del Porta Firmas.

D'aquesta manera, en l'últim apartat veurem la llista de requisits que s'han tractat d'implementar i millorar en aquest projecte **PFC: Porta Firmas**.

4.1. Requisits funcionals

Requisit: Ha de permetre l'administració d'usuaris (usuari, contrasenya i altres paràmetres).

Estat: Fet. L'entrada al sistema es controla mitjançant el nom de l'usuari i la seva contrasenya. La contrasenya es xifra mitjançant spaiCrypt/spaiDecrypt de la llibreria SpaiCrypto.dll. Per l'administració d'usuaris, s'ha desenvolupat una altra aplicació que no està inclosa en aquest projecte.

Requisit: Ha de permetre definir circuits de signatura (signatures cosign i countersign predefinides entre varis usuaris).

Estat: A mig fer. Els circuits de signatura estan definits per 3 usuaris, però només es permet la signatura d'un usuari cada vegada.

Requisit: Ha de permetre identificació per certificat X.509 i DNIe.

Estat: Per fer. Un usuari ha de poder entrar al sistema mitjançant un certificat vàlid d'una Agència de Certificació. Per exemple, els certificats que dona la FMNT (Fábrica Nacional de Moneda y Timbre), el IDCat, el DNI electrònic, etc. Cada usuari tindrà un o més certificats associats.

Requisit: Ha d'implementar els tipus de signatura digital més habituals (PDF, XMLDSig, XAdES, CAdES, CMS.. etc.) amb les seves variants (implícita, explicita; attached, detached, etc).

Estat: Per fer, afegir més tipus suportats ja que només signa PDF.

Requisit: Ha de permetre la generació de segells de temps en la signatura.

Estat: A mig fer. No sembla que funcioni, però hi ha parts desenvolupades.

Requisit: Ha de permetre la validació de Signatures, Certificats i segells de temps de qualsevol document signat amb una signatura reconeguda pel sistema.

Estat: Possibles modificacions necessàries. Inicialment, només s'utilitzava la PSIS de CATCert, però s'hi va afegit el Web Service de @Firma per validar quan no es pugui utilitzar CATCert. A l'afegir nous tipus de signatura, es necessitarà fer les modificacions necessàries.

Requisit: Ha d'integrar-se almenys amb les plataformes de validació CATCert i @Firma.

Estat: Fet. Si que està integrat amb CATCert en el DSS, però @Firma està fet d'una altra manera.

Requisit: Ha de definir una bústia de documents per a cada usuari.

Estat: Fet.

Requisit: Ha d'enviar els documents a cada bústia en funció del circuit de signatura definit

Estat: Fet però només està actiu per firmar un usuari cada vegada.

Requisit: Ha d'avisar per email a cada usuari quan un document entra a la seva bústia

Estat: Per fer. Hi ha alguns problemes que fan que aquest requisit de moment es deixi de banda.

Requisit: Ha de permetre signatura múltiple de documents.

Estat: Fet. A la web, ja podem seleccionar varis documents i signar-los tots mitjançant els applets de CATCert i @Firma.

Requisit: Ha de permetre esborrat múltiple de documents.

Estat: Fet. A la web, ja permet descartar varis documents a la vegada.

Requisit: Ha de proveir serveis web (Web Services) d'inserció d'un document.

Estat: Fet. WS del Porta firma anomenat **registrarXML**.

Requisit: Ha de proveir serveis web (Web Services) de consulta d'estat d'un document.

Estat: Fet. WS del Porta firma anomenat **consultarXML**.

Requisit: Ha de proveir serveis web (Web Services) d'obtenció d'un document

Estat: Fet. WS del Porta firma anomenat **getDocumentoXML**.

Requisit: Ha de permetre signatura desatesa de documents.

Estat: Per fer. Pot no ser possible ja que és el client mitjançant els applets de CATCert i @Firma qui signa els documents, no el servidor.

Requisit: Ha de mostrar una interfície gràfica multi idioma.

Estat: Fet. Possibles modificacions si cal afegir nous missatges als fitxers propierties de cada idioma. Actualment disponible en Català i Castellà.

Requisit: Ha de ser multi-entitat i multi-aplicació.

Estat: Fet. El Porta Firmas va se dissenyat per poder ser multi-entitat i multi-aplicació, facilitant que altres aplicacions es comuniquin amb els Web Services.

Requisit: Ha de permetre la configuració de la interfície gràfica per usuari per a establir filtres predefinits de la informació a mostrar, nombre de registres per pantalla etc.

Estat: A mig fer. Hi ha els formularis i les taules per a guardar la configuració, però no funciona bé. Gran part d'aquests requisits ja estan implementat mitjançant una funció AJAX que filtra els documents segons uns paràmetres.

4.2. Requisits no funcionals

Requisit: El client ha de ser compatible amb els navegadors actuals com Internet Explorer, Mozilla Firefox etc.

Estat: A mig fer. Solament funciona bé amb Internet Explorer, però hi ha diversos problemes de visualització.

Requisit: Ha de funcionar tant en sistemes Windows com en altres basats en Linux

Estat: Per fer, de difícil compliment. Pel que fa al servidor amb Tomcat, hi ha diverses dependències de llibreries (com és el cas de spaiCrypto.dll per al tractament criptogràfic de les contrasenyes) que no permeten l'execució en Linux. Pel que fa al client, els applets de signatura estan dissenyats entorn a MS Windows i tampoc funcionarien amb Linux.

4.3. Resum dels requisits a implementar

Després d'estudiar els diferents requisits, el seu estat d'implementació i la viabilitat de dur-los a terme, es van escollir els següents:

Requisits funcionals:

- Identificació de l'usuari mitjançant Certificat Digital.
- Millores a l'hora de registrar un document mitjançant la web.
- Afegir nous tipus de signatura (XAdES, XMLDSig).
- Afegir suport per fer signatures amb una Time Stamp Authority.
- Opcions de configuració.

Requisits no funcionals:

- Millorar la web i que es visualitzi de manera correcta. Així, mirar que funcioni bé amb altres navegadors com ara Mozilla Firefox, Google Chrome, etc.

5. Disseny i implementació

5.1. Login amb certificat

5.1.1. Objectiu

Afegir autentificació d'usuaris mitjançant un Certificat Digital. El certificat ha de ser vàlid i assignat per una Entitat Certificadora reconeguda.

5.1.2. Solució aportada

Veient com l'applet de CATCert mostrava una finestra amb la llista de certificats que Windows té instal·lat, s'ha fet un applet anomenat **AppletCertLogin.jar** que ens serveix per seleccionar un certificat i retornar diverses dades d'interès com el SN (número de sèrie del certificat, generat per cada Autoritat Certificadora) i el CN (Common Name, l'àlies del certificat). Així, hem creat un applet que ens permet obtenir informació del certificat que té el client i a la BD, guardarem una llista amb els seus certificats.

5.1.3. Descripció del codi de la solució

Aquí ens mourem dins del framework Struts2 i utilitzarem Hibernate per comunicar-nos amb la base de dades i obtenir la informació necessària per fer el login.

Primer, veiem con fa el login amb Struts2, dins del fitxer de configuració struts.xml:

```
<action name="login" class="es.spai.portafirmas.web.action.LoginAction">
     <result name="input">/WEB-INF/web/login.jsp</result>
     <result type="redirect-action">desktop</result>
</action>
```

Així, login.jsp ens serveix per mostrar el formulari habitual per demanar el nom d'usuari i password i un cop s'envia, l'acció que tractarà serà la classe LoginAction.java.

En LoginAction.java es fan consultes a la BD per demanar la contrasenya de l'usuari i comparar si es correcta. En el cas del login amb certificat, per cada usuari que vol tenir un certificat associat, hem d'afegir les files a la taula USUCERTIFICADOS, on s'hi guarda la relació entre l'usuari i el número de sèrie del certificat. La Figura 8 - Taules involucrades en Login amb Certificat mostra les 3 taules: USUARIO, USUCERTIFICADOS i CAS.

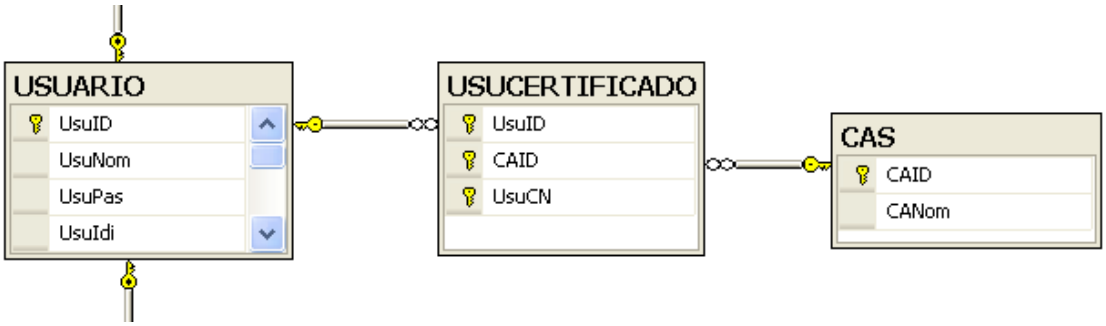

Figura 8 - Taules involucrades en Login amb Certificat

La taula CAS guarda el codi i el nom de la Autoritat Certificadora.

La taula USUCERTIFICADOS guarda en UsuCN del certificat. Per no modificar la taula, en comptes de guardar en UsuCN el CN (Common Name), s'ha canviat per guardar-hi el SN (Número de Sèrie) del certificat.

D'aquesta manera, també hem modificat les classes login.jsp per afegir un botó per cridar l'applet **AppletCertLogin.jar** i un camp ocult per enviar el SN (número de sèrie del certificat seleccionat). En LoginAction.java, hem de tractar un nou cas per quan detectem que l'usuari es vol autentificar amb un certificat, buscant a la Base de Dades si hi ha algun usuari que tingui el certificat associat. Si està associat, llavors es permet l'accés al portal web. En la següent **Figura 9** podem veure a mig el formulari normal per validar un usuari al mig i a la seva dreta, la finestra del AppletCertLogin on ens demana que seleccionem un certificat de la llista. Aquesta llista de certificats retorna tots el certificats que guarda Windows al seu Certificate Store de manera automàtica. Per exemple, si tenim un certificat en una targeta digital (com es el cas del DNI electrònic) també ens el mostra a la llista.

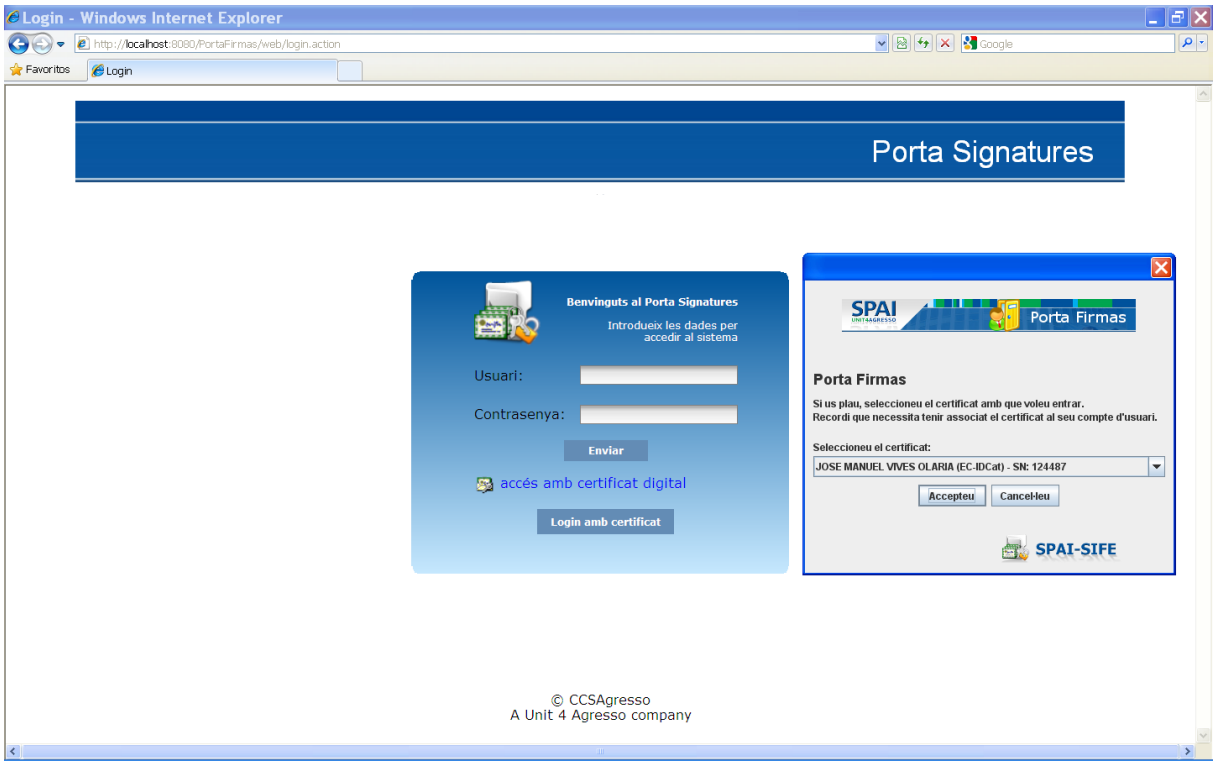

Figura 9 - Captura del Login amb Certificat

5.1.4. Conseqüències del procés

Si l'usuari té associat un certificat i el selecciona a l'hora de fer el login, pot entrar al portal web del Porta Firmas per gestionar els seus documents.

5.1.5. Possibles problemes i alternatives

El principal problema de la solució aportada és la seguretat. No hi ha garanties de que el SN (número de sèrie del certificat) sigui únic, ja que cada Autoritat de Certificació els genera. A més, tampoc és bona solució utilitzar el CN (Common Name) on generalment es guarda per exemple el nom de la persona i el seu DNI, ja que poden crear certificats amb els CN i utilitzar-los per autentificar-nos com a un altre usuari. Encara que en vàries proves fetes amb un self signed certificat, no apareix a la llista al fer el login amb certificat.

El framework Struts2 també té algunes opcions d'autentificació del client mitjançant un certificat digital (Client side certificate authentication), però per no hem pogut trobar gaire informació de com fer-ho, només en el llibre Struts 2: Black book.

Una altra manera mirar de tenir més seguretat amb els certificats és utilitzar un framework per fer Single Sign-on (Autentificació en un sol punt) com ara OpenSSO, JOSSO o JA-SIG CAS. Aquests frameworks ofereixen serveis per poder enviar i rebre certificats per un canal segur SSL per fer accions de login.

Un altre problema és que podem entrar al portat si directament anem a la direcció:

http://localhost:8080/PortaFirmas/web/desktop.action

Una solució es mirar de desenvolupar més la seguretat de Struts 2 mitjançant els seus fitxers de configuració.

Un altre problema és que no hi ha cap eina feta per enllaçar un certificat amb un usuari de manera que s'ha de fer manualment.

5.2. Registrar documents per firmar

5.2.1. Objectiu

En general, millorar l'aspecte i la funcionalitat de la pàgina per registrar documents. L'objectiu principal és automatitzar certs camps del formulari per tenir valors per defecte, afegir la validació necessària i comprovar que la petició d'afegir registre funciona correctament.

5.2.2. Solució aportada

Un formulari ha de mostrar certes opcions per defecte, unes quantes obligatòries i d'altres opcionals i ha de validar que les dades tinguin el format desitjat. Així, s'ha modificat per fer més amigable el formulari fent que alguns camps que abans s'havien de buscar manualment, ara ja tinguin unes opcions per defecte suficient per la majoria dels casos.

5.2.3. Descripció del codi de la solució

La majoria del codi és de la classe insertarDocumento.jsp, utilitzant els tags de la llibreria core de JAVA per JSP. Aquí tenim un exemple:

```
...
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
...
<c:forEach items="${filtrosCombo.usuario}" var="opUsuario">
        <c:if test="${opUsuario.usunom == sUserCode}">
                <option selected value="${opUsuario.usuid}">${opUsuario.usunom}</option>
        </c:if>
        <c:if test="${opUsuario.usunom != sUserCode}">
                <option value="${opUsuario.usuid}">${opUsuario.usunom}</option>
         </c:if>
</c:forEach>
.....
```

En aquest cas, utilitzem un tag forEach per recórrer tota la llista que forma el combo per seleccionar l'usuari i, si l'usuari és el mateix que el que hi ha a la sessió, llavors s'afegeix al combo aquest usuari amb l'opció selected activada. Així, l'usuari actual estarà seleccionat per defecte.

D'igual forma, se n'han fet altres com:

Tipus de Signatura → Per defecte, seleccionat tipus PDF. Extreu els tipus de la taula PFE_FIRTIPOFIRMA. Els tipus de signatura són PDF, XAdES, CAdES, PDFCAdES, XMLDSig, XMLMultDoc, Attached i Detached.

Identificador d'Aplicació → Per defecte, Porta Firmas. Extreu l'identificador de la taula APLICA i els possibles valors són Porta Firmas i Gestor

Codi d'Usuari → Per defecte, el codi de l'usuari de la sessió actual. Extreu els codis d'usuari de la taula USUARIO.

Altres millores han estat afegir al camp Nom, el nom del fitxer del camp Arxiu de manera automàtica i amb una expressió regular, forçar que el DNI sigui de 8 nombres i una lletra al final. Si la lletra del DNI està en minúscula, llavors és canvia a majúscula. En la següent figura es mostra com queda el formulari per afegir un document per signar.

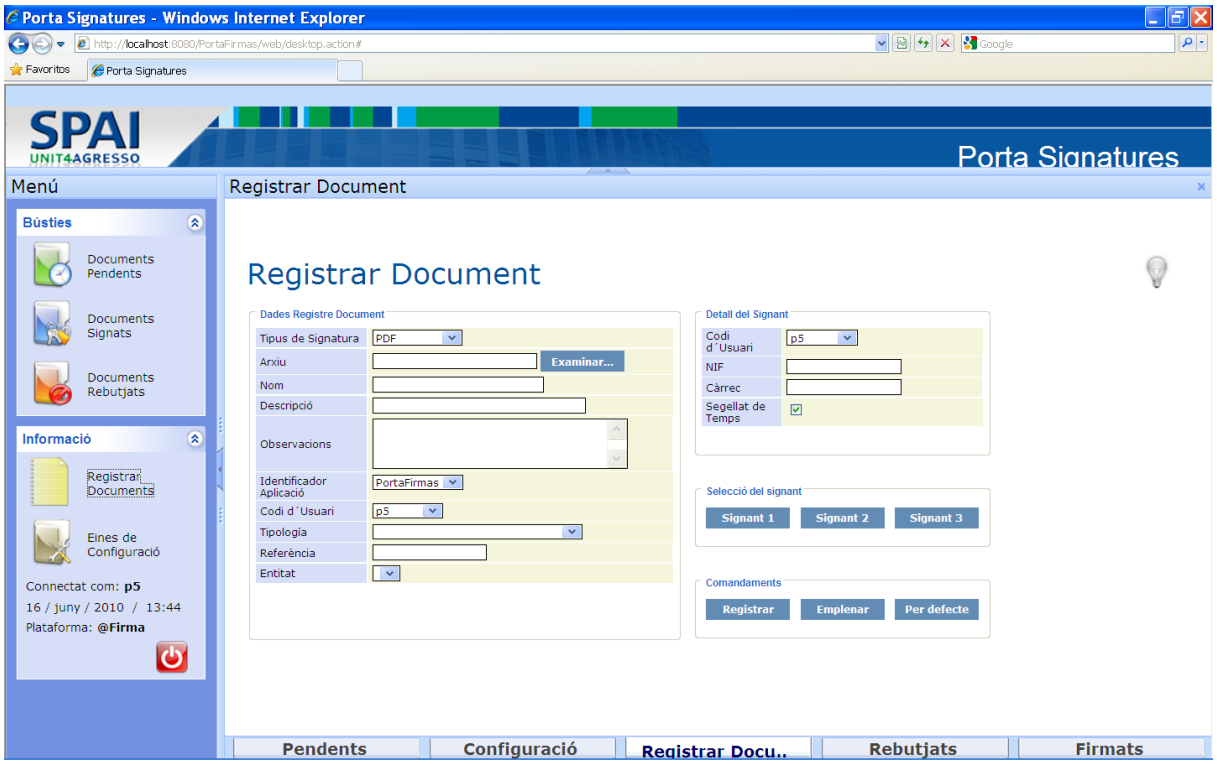

Figura 10 - Captura de Registrar Document amb IE

5.2.4. Conseqüències del procés

Una vegada s'han entrat els camps al formulari, l'usuari registra el document a signar mitjançant el botó Registrar. El botó Registrar, primer valida (amb funcions Javascript) que hi ha tots els camps obligatoris i que són correctes. Després, es crida a la funció RegistraDWR.registrar que ens serveix per cridar al web Service RegistraXML que guardarà a la Base de Dades tota la informació necessària. Es poden afegir fins a 3 usuaris per signar el document, essent el Signant 1 el primer que pot signar el document. Si tot ha anat bé, es mostra una finestra modal amb el nom del fitxer i el número identificador del fitxer a la Base de Dades. Els fitxers es guarden a la carpeta C:\PFE\Documentos.

5.2.5. Possibles problemes i alternatives

DWR(Direct Web Remoting) és una biblioteca Java que permet al codi Java en el servidor i al codi de Javascript en el navegador web interactuar i intercanviar informació de la forma més senzilla possible.

L'ús de la llibreria DWR no funciona correctament amb el navegador Google Chrome. S'ha provat d'actualitzar la llibreria a l'última versió però llavors deixa de funcionar correctament amb altres navegadors com ara el Firefox o Internet Explorer.

A més, creiem que es podem obtenir la mateixa funcionalitat utilitzant les accions del Struts2. Per fer-ho, s'ha de reorganitzar tot el procés de signatura.

Com es pot veure en la **Figura 11**, també tenim problemes de correcta visualització del formulari en altres navegadors diferents del Internet Explorer com es el cas del Firefox. Com es pot apreciar, el requadre Detalls de Signant es dibuixa desplaçat de la posició correcta.

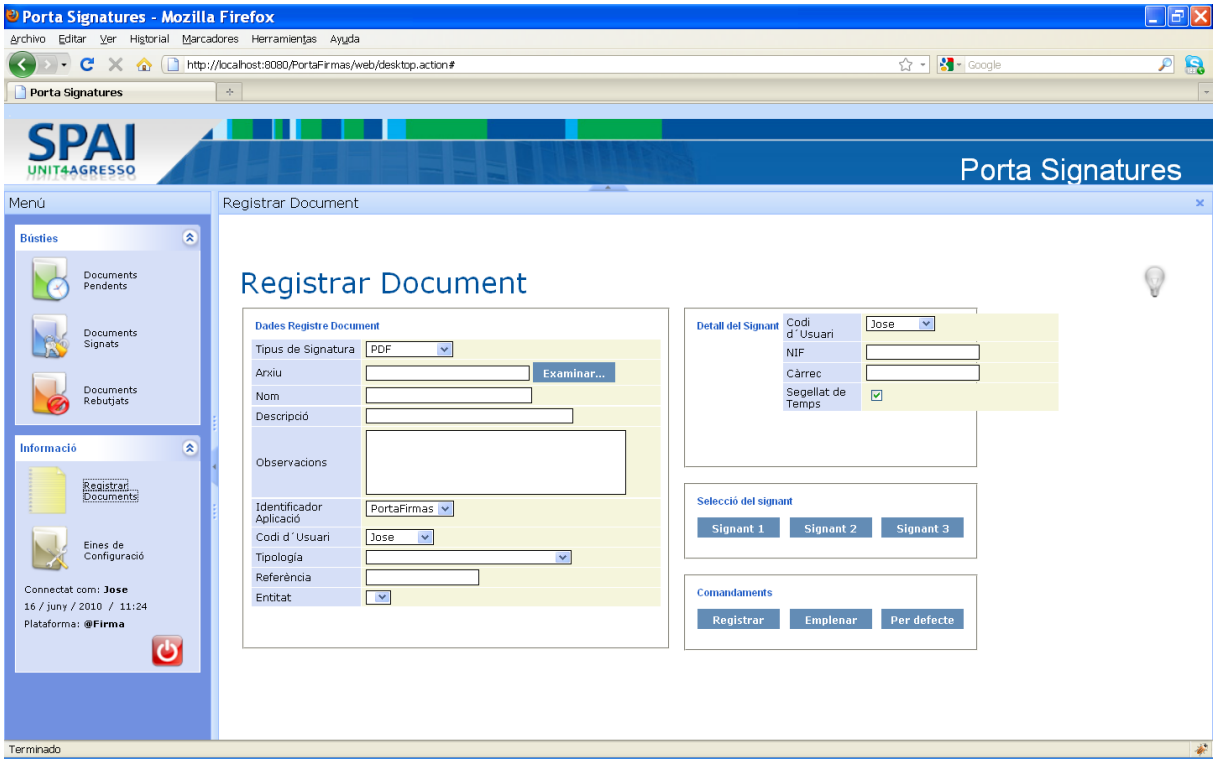

Figura 11 - Captura de Registrar Document amb Firefox

5.3. Afegir nous tipus de signatura

5.3.1. Objectiu

Permetre la signatura de documents mitjançant XAdES i XMLDSig.

5.3.2. Solució aportada

Veient les especificacions dels applets de CATCert i @Firma, ambdós ofereixen serveis per la signatura de documents XML amb el tipus de firma XAdES i XMLDSig. Així, com es mostra a la següent figura hi ha 3 maneres de signar un document XML:

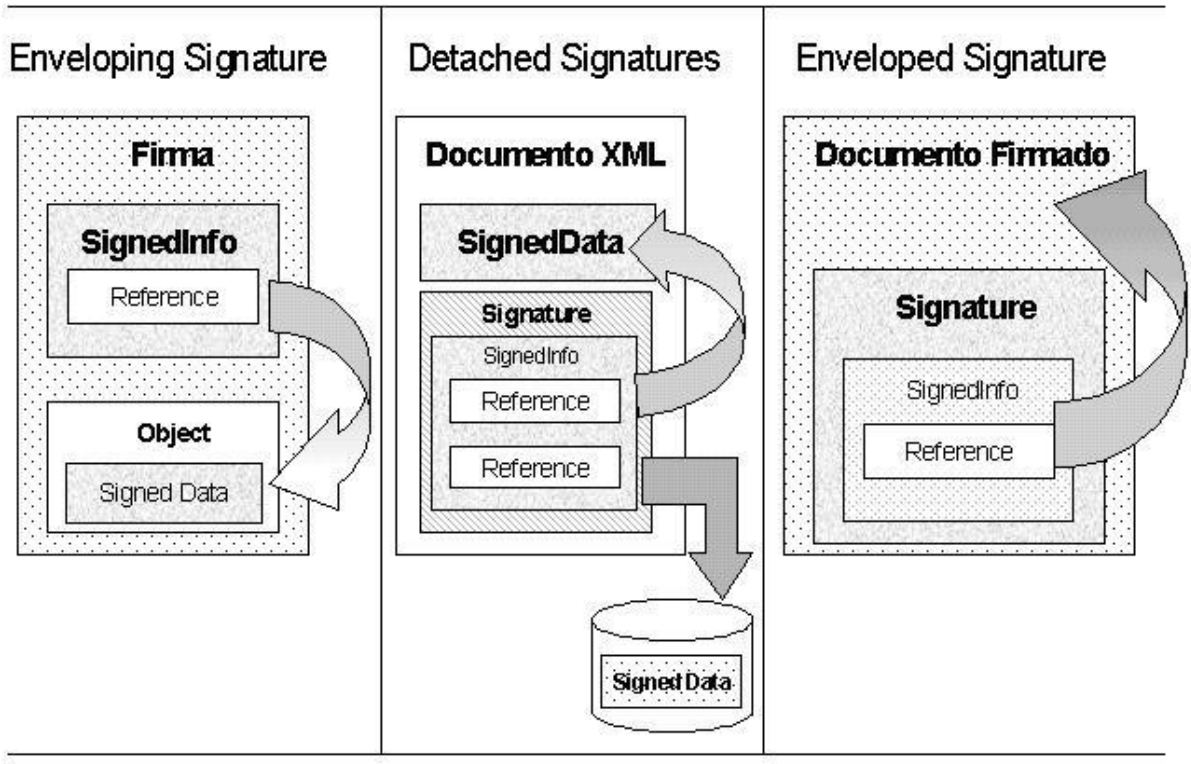

Figura 12 - Tipus de signatura en documents XML

1. Enveloping Signature: La signatura XML embolcalla el contingut que es signa.
2. Detached Signature: L'objecte que s'ha signat està separat de la signatura XML.
3. Enveloped Signature: El contingut que es desitja signar engloba a la signatura.

Per no modificar la taula de tipus de signatura suportats PFE_FIRTIPOFIRMA, quan es signa amb XAdES o XMLDSig es fa de tipus Enveloping Signature per defecte.

5.3.3. Descripció del codi de la solució

La crida als applets es fa dins de la classe DesktopFooter.jsp, amb codi Javascript. Quan un usuari vol signar un document, el selecciona de la llista de documents pendents per signar. Llavors, depenent de quina plataforma utilitzem (CATCert o @Firma), es cridaran les funcions appletCATCertSign o appletArrobaFirmaSign. Cada funció, configura l'applet corresponent segons quin tipus de firma s'ha de fer.

Un cop signat un document XML mitjançat l'applet de CATCert, retorna el document signat no en base64 sinó en clar. Així, s'ha de transformar a Base64 ja que l'acció guardaDOCSignat.action espera que el document estigui en Base64. Guardem el document mitjançant la crida:

```
doAjaxSACK('guardaDOCSignat.action', parametros, GuardDocAfterFunc);
```

L'acció guardaDOCSignat.action s'executa al servidor i guarda el document signat a C:\PFE\Documentos i actualitza la bústia de document del següent signant. Al retornar, s'executa la funció GuardDocAfterFunc que tornarà a iniciar un nou cicle de signatura si l'usuari havia escollit varis documents per signar.

5.3.4. Conseqüències del procés

Els dos applets (tant CATCert com el de @Firma) signen els fitxers que han estat transformats en Base64. Això es fa perquè es vol fer invisible a ulls de l'usuari que se li envia el document al seu ordinador, el signa amb l'applet i després es torna a enviar al servidor per guardar-lo. Com ja s'ha comentat, CATCert no retorna els documents signats amb XAdES i XMLDSig en Base64. Per aquest cas en particular, s'ha de codificar manualment.

5.3.5. Possibles problemes i alternatives

Quan es signa amb l'applet de CATCert (versió 1.8, però també passa amb l'ultima versió 1.9.5) i el tipus de document a signar és XAdES, retorna un document de tipus XMLDSig. S'ha intentat modificar les opcions, però sempre s'ha obtingut el mateix resultat. Això crea un problema a l'hora de validar un document mitjançant el Web Services de @Firma ja que s'envia el document i el tipus de signatura a l'hora de validar-lo i com que són diferents, retorna un error. Tampoc funciona la validació dels nous tipus de signatura afegits amb els Web Services de la DSS, ja que aquests només validen fitxers PDF mitjançant la PSIS de CATCert. Una solució seria afegir més Web Services a la DSS per tal de validar també documents XAdES i XMLDSig amb la PSIS.

Els tipus que falten a XMLDSig i XAdES (Detached i Enveloped) ja estan afegits a les respectives funcions de l'applet, però per simplificar i no modificar la taula PFE_FIRTIPOFIRMA, per defecte només es signa amb Enveloping. De manera experimental, s'ha afegit altres tipus de signatura com PDFCAdES.

5.4. Signatures amb Time Stamp

5.4.1. Objectiu

Afegir el Time Stamp a la signatura d'un document. Quan ens referim a Time Stamp, volem dir que volem que una tercera entitat anomenada TSA (Time Stamp Authority) ens signi el document amb una marca temporal i acrediti que l'hora és la correcta. Generalment, si es signa un document sense Time Stamp, per defecte s'agafa l'hora de l'ordinador que el signa. Així, un document amb Time Stamp ofereix més garanties.

5.4.2. Solució aportada

Aquí tenim 2 problemes, ja que l'applet de CATCert sí que dóna la possibilitat de fer signatures amb Time Stamp (cas PDF). En canvi, l'applet de @Firma no te aquesta funcionalitat.

5.4.3. Descripció del codi de la solució

Primer, hem hagut de solucionar com obtenir el camp Time Stamp de la Base de Dades en la classe DesktopFooter.jsp, ja que necessitem saber si el document es vol signar o no amb Time Stamp. Així, hem modificat la funció getInfoDocumentoRetornoFunc on hi ha una crida que ens permet, entre altres, agafar el tipus de firma, la descripció, el NIF i el Time Stamp i ho guarda en les variables globals _tipoFirma, _descripcion, _nif i _timestamp respectivament.

Un cop tenim el Time Stamp, tenim 2 casos:

1. Cas de CATCert: Modificar la funció appletCATCertSign pel cas de PDF. Llavors, si _timestamp és cert s'activa l'opció del Time Stamp de l'applet de CATCert.
2. Cas de @Firma: Malauradament, s'ignora ja que no és tan trivial. Necessitaríem un Web Service que cridés la funció de Signar del Web Service de @Firma, on si que pot signar documents amb Time Stamp.

Cal afegir que CATCert també té els tipus XAdES-T, que segons les especificacions, permet afegit un Time Stamp a una signatura XAdES. Aquest cas també s'ha afegit.

5.4.4. Conseqüències del procés

Si l'usuari ha sol·licitat firmar un fitxer amb format PDF i amb Time Stamp, si ha signat amb l'applet de CATCert el seu document si que tindrà un Time Stamp. Sinó, cas de @Firma, el document tindrà la marca temporal del rellotge del sistema on s'ha signat en aquell moment. Hem de tenir en compte que l'applet de CATCert necessita una connexió a INTERNET per concertar-se amb la PSIS (Plataforma de serveis d'identificació i signatura). Sinó, tampoc pot signar amb un Time Stamp.

5.4.5. Possibles problemes i alternatives

El problema aquí el tenim amb l'applet de @Firma, ja que no signa amb Time Stamp. Una possible solució, seria cridar al Web Service de @Firma i signar documents. El problema és que no és tan trivial de fer. En general, un Web Service es desitja que no guardi un estat i que les seves accions siguin sense estat. Això no vol dir que no es pugui tenir un estat, i de vegades necessitem la noció de sessió en un Web Service. Aquest és el cas del Web Services que ofereix @Firma per signar un document. Primer hem de pujar el fitxer i guardar-lo en un "Módulo de Custodia". Després, hem de cridar al Web Service Firma Servidor i passar-li entre altres, l'identificador del Modulo de Custodia, el certificat per signar, l'aplicació que ho demana i el tipus/format de firma.

5.5. Opcions de configuració

5.5.1. Objectiu

Poder guardar i restaurar les opcions de configuració de la web de Porta Firmas per a cada usuari.

5.5.2. Descripció de la possible solució

Bàsicament és volia continuar la solució que estava a mig fer. A la pàgina configuracio.jsp es mostra un formulari on podem guardar una sèrie de filtres (entitat, aplicació, dates de registre de document, descripció, tipologia, grup tipologia, referència i etiquetes) i opcions de visualització (com el tema, l' idioma, la paginació i l'avís per mail d'un nou document a la bústia). A la següent figura es mostra el formulari de configuració.

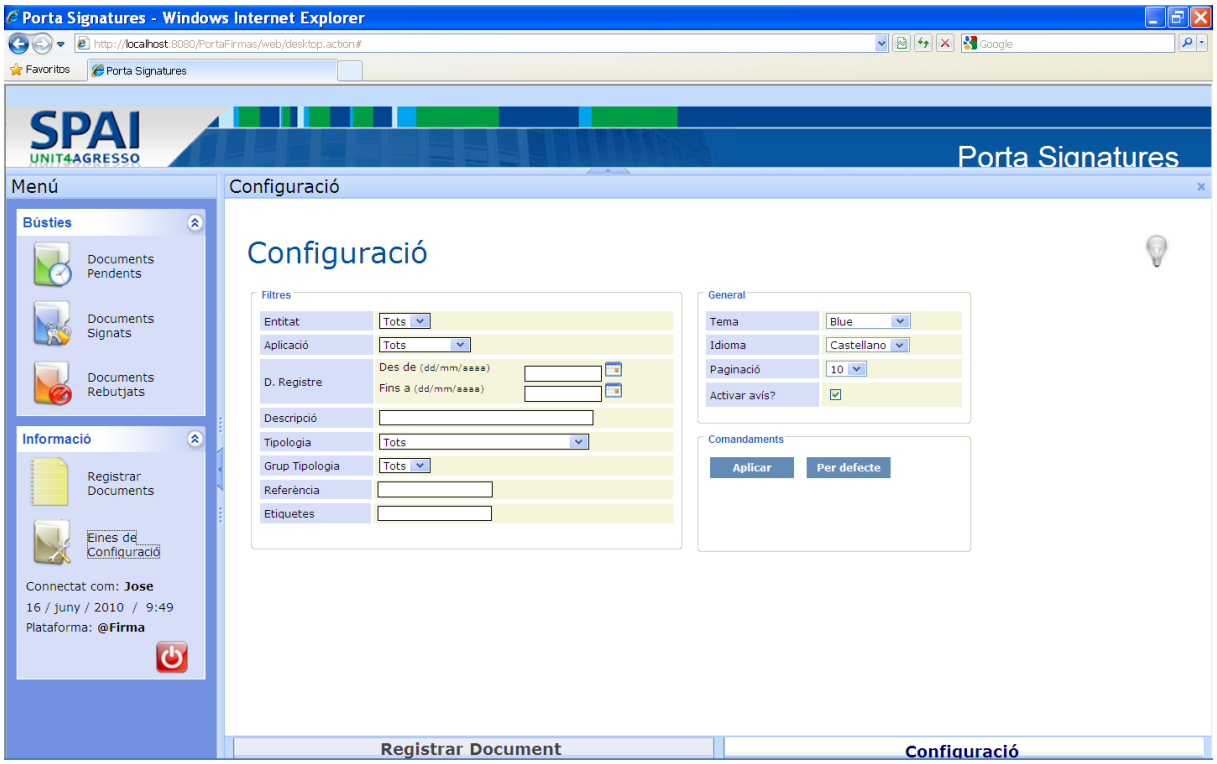

Figura 13 - Formulari de Configuració

Aquesta informació es guarda a la Base de Dades en les taules PFE_Filtros, PFE_ParamFiltros i PFE_Parametrización, estant PFE_Parametrización enllaçada amb la taula USUARIOS.

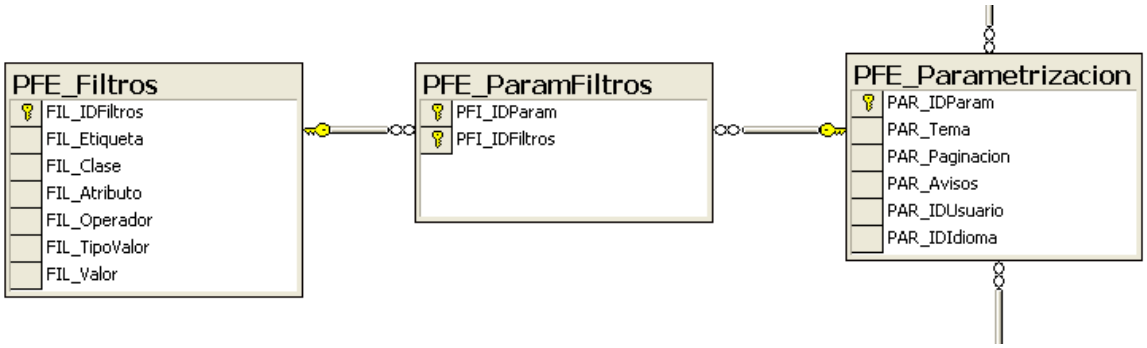

Figura 14 - Taules involucrades en la Configuració

5.5.3. Problemes trobats

Bàsicament, ens hem trobat 2 problemes:

1.- Les taules PFE_Filtros, PFE_ParamFiltros i PFE_Pametrizacion no estaven mapejades amb Hibernate. Una manera de mapejar-les ha estat fent servir el plugin d'Eclipse Hibernate Tools per generar els fitxers de mapeig de la Base de dades que necessita Hibernate.

2.- Utilització d'AJAX per mostrar les taules de pendents, signats i rebutjats, tenint problemes a l'hora de canviar els valors que estaven guardats a les taules.

Veurem el problema amb un exemple de prova que es va voler implementar per comprovar la viabilitat. Es va voler modificar la paginació de documents pendents, mostrant segons la tria de l'usuari, 10, 20, 30 o 40. Es va poder fer funcionar el guardar i restaurar la paginació a la BD, però a l'hora de modificar el codi AJAX, no es va poder aconseguir que canvies el valor. A continuació es mostra el codi:

....

```
<%@ taglib uri="http://displaytag.sf.net" prefix="display"%>
<%@ taglib uri="http://ajaxanywhere.sourceforge.net/" prefix="aa"%>

<s:set name="webDocsPendents" value="webDocs" scope="request" />
<s:set name="sUserCode" value="sUserCode" scope="request" />

<aa:zone name="pendentsTable">
        <form name="formPendents" id="formPendents">
                <display:table requestURI="/web/pendents.action" name="webDocsPendents"
               class="list" id="webDocsListPendents" export="false" sort="list" pagesize="10">
```

….

L'atribut que controla el nombre de registres és pagesize però no funcionava bé quan es va intentar canviar dinàmicament.

5.5.4. Situació final

Per tots aquests problemes i donat que si es vol filtrar els documents ja hi ha implementat el desplegable AJAX que també pot filtrar documents, s'ha descartat modificar res. Per tant, les opcions de configuració continuen sense funcionar.

5.5.5. Possibles problemes i alternatives

Un possible problema, si s'hagués acabat implementant la solució que en el projecte està a mig fer, és que s'augmenta el nombre de consultes a la Base de Dades. Així, tal i com estava proposat, cada vegada que l'usuari vol mostrar la llista de documents pendents per signar, els signats i els rebutjats, s'havien de fer vàries consultes a la BD per carregar les possibles opcions.

Una possible solució podria ser afegir els atributs de configuració necessaris al fer el login. Per exemple, el següent codi extret de loginAction.java mostra com s'obté la sessió actual i s'afegeix un atribut a la sessió per guardar el nom de l'usuari:

....

```
// Guardem el nom del usuario en el context de la sesion
        Map attributes = ActionContext.getContext().getSession();
        attributes.put( USER, username);
        return SUCCESS;
```

....

D'aquesta manera, ens evitaríem carregar de consultes la Base de Dades cada vegada que es vol mostrar una taula.

5.6. Millores en la web del Porta Firmas

5.6.1. Objectiu

L'objectiu és bàsicament, millorar la compatibilitat amb els navegadors, actualitzar l'applet d'@Firma a l'última versió 3.02 i millorar varis problemes de visualització de les pàgines del portal web del Porta Firmas.

5.6.2. Problemes detectats

Hem trobat varis problemes de visualització. Entre ells:

1.- Finestres modals per afegir les observacions que no es mostren correctament

2.- Varis camps de la pàgina per afegir documents que no es mostren correctament

3.- Calia actualitzar l'applet de @Firma a la versió 3.02

4.- No es carreguen correctament els applets de la web en alguns navegadors.

5.6.3. Descripció del codi de la solució

Per solucionar el problema de les finestres modals, s'han modificat les classes modalRebuig.jsp i ObtenerModal.jsp, modificant la taula que mostra el formulari.

Els problemes al afegir documents han sigut varis. Entre ells, s'ha fet:

1.- Problema en la visualització de varis camps

Solució: S'ha reorganitzat i millorat, com s'ha comentat en l'apartat registrar documents per signar

2.- Problema del path: ruta FAKEPATH. Aquest problema ha sorgit perquè per temes de seguretat, es vol evitar que Javascript pugui accedir a la ruta d'un <INPUT>. Internet Explorer retorna "c:\FakePath\nomFitxer" quan s'intenta accedir a la ruta de l'input, mentre que Firefox només retorna el nom del fitxer.

Solució: Fer que l'applet appletleearchivo.jar mostri una finestra d'explorador de Windows per seleccionar el fitxer i retornar la ruta i el nom del fitxer.

3.- Problemes al carregar l'applet appletleearchivo.jar

La solució per carregar els applets s'ha fet de vàries maneres:

A) Amb els tags <APPLET>. S'ha carregat així l'applet de CATCert

B) Amb la funció Javascript de SUN http://java.com/js/deployJava.js, s'han carregat així els applets AppletCertLogin.jar, appletleearchivo.jar i també l'utilitza de manera interna l'última versió de @Firma.

C) Amb el tag recomanat per W3C és <OBJECT>. És el que ha donat més problemes ja que cada navegador el suporta a la seva manera. S'ha descartat utilitzar-lo.

5.6.4. Possibles problemes i alternatives

Tot i haver solucionat varis problemes, la compatibilitat amb els navegadors continua sent baixa. Només funcionen totes les opcions correctament amb Internet Explorer i Mozilla Firefox, tenint més problemes en Opera i Google Chrome.

Com a possibles alternatives per millorar la presentació, es podria utilitzar la llibreria jQuery que és molt utilitzada dins l'empresa. jQuery es una Llibreria Javascript que simplifica el codi HTML, dóna suport pel control d'esdeveniments, animacions i les interacciones amb AJAX. A més facilita la compatibilitat entre navegadors ja que és suportada per la majoria, oferint una molt bona visualització.

6. Pla de proves

Finalment, per comprovar el funcionament del Porta Firmas s'han realitzat les següents proves. En una primera ronda, s'ha utilitzat 4 navegadors web diferents per comprovar la compatibilitat. Els navegadors són: l'Internet Explorer versió 8.0, Mozilla Firefox versió 3.6.3, Google Chrome versió 5.0.375.70 i Opera versió 10.53). Amb tots ells s'ha provat el funcionament del login, registrar documents, signar, validar i descartar. En una segona ronda, s'ha comprovat el funcionament de la validació de les firmes mitjançat els Web Services de cada plataforma.

Les proves han sigut realitzades en la mateixa màquina on hi ha instal·lada una Base Dades MS SQL 2000, l'Apache Tomcat 6.0 i el Porta Firmas, executat dins de l'Eclipse. Veurem un resum en les següents taules.

Taula 3 - Validació del Login

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	OK	No, requadre no es dibuixa	OK
Login normal (user//pass)	OK	OK	OK	OK
Login amb certificat (appletCertLogin)	OK	OK	OK	No carrega amb deployJava.js

Taula 4 - Validació de Registrar Documents

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	No, quadre detall malament	OK	No, els camps sobresurten
Registrar Document	OK	OK	No, error en DWR	OK

Taula 5 - Validació de Documents Pendents (Signatura d'un usuari)

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	OK	OK	OK
Signar amb @Firma	OK	OK	No, error en DWR	OK
Signar amb CATCert	OK	OK	No, error en DWR	OK
Modal Observacions de Signar	OK	OK	OK	OK
Rebutjar	OK	OK	No, error en DWR	OK
Modal Observacions de Rebutjar	OK	OK	OK	OK
Visualitzar el document	OK	OK	OK (el descarrega directament)	OK

Taula 6 - Validació de Documents Signats (Validar firma d'un document)

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	OK	OK	OK
Validar amb @Firma	OK	OK	No, error en DWR	OK
Validar amb CATCert	OK	OK	No, error en DWR	OK
Rebutjar	OK	OK	No, error en DWR	OK
Modal Observacions de Rebutjar	OK	OK	OK	OK
Visualitzar el document	OK	OK	OK (el descarrega directament)	OK

Taula 7 - Validació de Documents Rebutjats

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	OK	OK	OK
Visualitzar el document	OK	OK	OK (el descarrega directament)	OK

Taula 8 - Validació de Opcions de Configuració

	Internet Explorer	Mozilla Firefox	Google Chrome	Opera
Visualització	OK	OK	OK	OK
Guarda Configuració	No implementat	No implementat	No implementat	No implementat

A continuació es mostren proves de validació creuades. L'objectiu és comprovar que documents firmats amb CATCert es validin també correctament amb els Web Services de @Firma i que els documents signats amb @Firma es validin amb els WS de CATCert.

Taula 9 - Validació creuada sense Time Stamp

Signat amb:	Validat amb:	Tipus	Internet Explorer 8.0
CATCert	CATCert	PDF	OK
CATCert	CATCert	XMLDSig	Falla, no implementat a DSS
CATCert	CATCert	XAdES	Falla, no implementat a DSS
CATCert	@Firma	PDF	OK
CATCert	@Firma	XMLDSig	OK
CATCert	@Firma	XAdES	Falla, error de tipus incorrecte
@Firma	CATCert	PDF	OK
@Firma	CATCert	XMLDSig	Falla, no implementat a DSS
@Firma	CATCert	XAdES	Falla, no implementat a DSS
@Firma	@Firma	PDF	OK
@Firma	@Firma	XMLDSig	OK
@Firma	@Firma	XAdES	OK

Taula 10 - Validació creuada amb Time Stamp

Signat amb:	Validat amb:	Tipus	Internet Explorer 8.0
CATCert	CATCert	PDF	OK
CATCert	CATCert	XMLDSig	Falla, no implementat a DSS
CATCert	CATCert	XAdES	Falla, no implementat a DSS
CATCert	@Firma	PDF	OK
CATCert	@Firma	XMLDSig	OK
CATCert	@Firma	XAdES	Falla, error de tipus incorrecte
@Firma	CATCert	PDF	OK
@Firma	CATCert	XMLDSig	Falla, no implementat a DSS
@Firma	CATCert	XAdES	Falla, no implementat a DSS
@Firma	@Firma	PDF	OK
@Firma	@Firma	XMLDSig	OK
@Firma	@Firma	XAdES	OK

Com es pot veure en les validacions creuades, només hi ha actualment validació amb la plataforma de CATCert per PDF. També s'observa el problema que s'ha comentat sobre la signatura en XAdES, que sembla que la fa de tipus XMLDSig. En canvi, @Firma si que pot validar tots els tipus de firma que s'han afegit.

A més d'aquestes proves, també s'han realitzat proves de firma múltiple de documents, guardant correctament els resultats. També, s'han provat els circuits de firma entre 3 usuaris, funcionant també de manera correcta.

En resum, s'ha donat suport a més navegadors però no s'ha aconseguit donar suport complert a Google Chrome i Opera. Els problemes del Google Chrome venen de la llibreria DWR, però al provar d'actualitzar-la la resta de navegadors no funcionen correctament. El tema del Opera en el login amb certificat sembla venir perquè no carrega correctament l'applet amb la llibreria deployJava, tot i que amb la resta d'applets funciona correctament inclòs l'applet d'@Firma que també es carrega amb deployJava. La firma funciona correctament per l'applet de @Firma, però amb l'applet de CATCert hi ha problemes amb els tipus XAdES. L'applet de CATCert és l'únic que pot firmar amb Time Stamp. Pel que fa a la validació, els Web Services d'@Firma poder validar tots els tipus suportats mentre que els Web Services de CATCert només dóna suport per validar PDF (més correctament, els DSS que ofereix el Porta Firmas només hi ha implementat el Proxy amb el servei de validació de PDF de la PSIS de CATCert).

7. Eines utilitzades

Pel desenvolupament del Porta Firmas s'han utilitzat moltes eines programes. Pel que fa al sistema Operatiu utilitzat, s'ha fet servir només Windows XP tant per desenvolupar com per fer les proves.

Pel que fa al programari, s'ha utilitzat:

Pel desenvolupament del codi, s'ha utilitzat l'IDE Eclipse versió Galileo (l'última versió a data de 2010) per desenvolupar i depurar el codi. A més, s'han fet servir els plugins:

- Hibernate Tools → Plugin que genera els XML necessaris per mapejar la base dades en objectes JAVA.
- JADClipse → Plugin per poder descompilar llibreries JAVA, utilitzat per depurar problemes amb algunes llibreries.
- Eclipse CXF plugin → Plugin per generar el codi tant del client com del servidor d'un Web Service amb el framework Apache CXF. Pot generar el codi a partir d'un WDSL (Top down) o bé d'una classe JAVA (Down up).

Per la base de dades, s'ha utilitzat el SLQ 2000 amb el Service Pack 4.

Pel servidor de servlets, s'ha utilitzat l'Apache Tomcat versió 6.0.

En quant a la gestió de codi, s'ha utilitzat l'eina TortoiseHG per crear un repositori al workspace. TortoiseHG és un front-end per facilitar les operacions de Mercurial. Mercurial és un sistema de control de versions distribuït i multi plataforma. D'aquesta manera, com es mostra en la següent figura s'ha tingut un control del codi que s'anava afegint, podent revisar què i quan s'ha modificat.

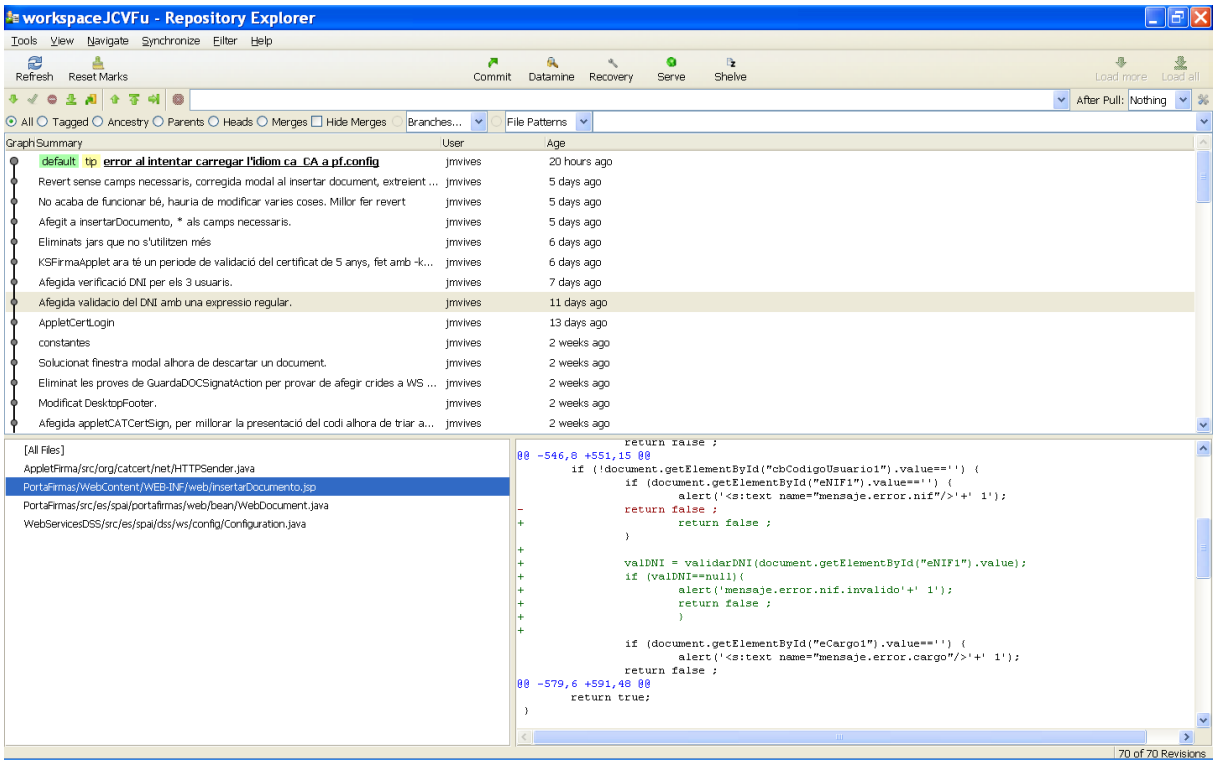

Figura 15 - Repository Explorer de TortoiseHG

Per les proves, s'han utilitzat els navegadors Internet Explorer versió 8.0, Mozilla Firefox versió 3.6.3, Google Chrome versió 5.0.375.70 i Opera versió 10.5.

Amb el Firefox, s'ha utilitzat el complement FireBug per depurar errors en HTML i Javascript de les pàgines.

Pel que fa a documentació, s'ha utilitzat de Microsoft Office 2007, el Word per fer la documentació i la memòria del projecte, l' Excel per guardar informació taules, i el Power Point per les presentacions.

La versió de JDK utilitzada ha estat sempre l'ultima versió disponible (al finalitzar el projecte, la versió és 1.6.0_20).

8. Conclusions

8.1. Resum assoliment d'objectius

S'han completat amb èxit els requisits:

- ✓ Login amb Certificat Digital mitjançant un applet.
- ✓ Millora de la web Registrar Document.
- ✓ Afegit suport pels tipus de signatura XAdES i XMLDSig.
- ✓ Afegit suport per signar amb un Time Stamp en la plataforma de CATCert.
- ✓ Millores per suportar el Portal Web en més navegadors.

No s'han pogut completar els requisits:

- × No tenim suport per signar amb un Time Stamp en la plataforma de @Firma.
- × Guardar i modificar l'aspecte del Portal Web amb Opcions de configuració.
- × Tenir suport de validació per als nous tipus de signatura en la plataforma de CATCert.

Han sorgit els següent problemes que no s'han resolt:

- × Més seguretat en el login amb certificat. La solució aportada pot no ser la millor. En general, caldrà afegir més seguretat al Portal Web.
- × Tenir suport de validació per als nous tipus de signatura XAdES i XMLDSig en la plataforma de CATCert.

Altres requisits que no formaven part del projecte, però que encara no han sigut implementades:

- ✓ Avisar per email als usuaris quan un document entra en la seva bústia. Tampoc es guarda cap informació del mail dels usuaris a la BD.
- ✓ Només es suporta el Sistema Operatiu Windows. Encara no hi ha suport per Linux.

8.2. Ampliacions i línies de millora

Les possibles línies de millora han d'anar encaminades a curt termini a afegir el suport de validació requerit per CATCert i afegir suport per la signatura amb els Webs Services d'@Firma.

El projecte es troba en una situació en que moltes de les llibreries s'han actualitzat i algunes, han deixat de ser desenvolupades. És el cas de, per exemple, el framework de Web Services Codehaus XFire, que s'ha millorat molt i que ara es recomana canviar-se a la nova versió anomenada Apache CXF.

Molts dels problemes de visualització dels navegadors web es poden solucionar utilitzant per exemple la llibreria jQuery. Un altre problema relacionat és que hi ha massa codi Javascript que fa funcions realment importants i que no es pot testejar i depurar de manera adequada. Tot aquest codi és necessari per gestionar els paràmetres que necessiten els applets de signatura i el tractament posterior dels documents signats.

També, l'ús de DWR és un símptoma de que potser el portal Web no està del tot ben dissenyat, i que no s'han tingut en compte algunes opcions que facilita el Struts 2.

8.3. Desviació respecte la planificació inicial

El projecte s'ha desviat respecte la planificació inicial, ja que es pensava que s'acabaria entorn al 14 de Maig de 2010, desviant-se un mes fins a la veritable data del 21 de Juny de 2010, on es va fer la presentació a l'empresa del resultats obtinguts.

Un altre problema en la planificació inicial és que era molt general. Pensada inicialment per tenir una idea molt general de les principals fases del projecte, el cert és que s'ha hagut de modificar i ampliar molt. Així, temes com l'aprenentatge dels diferents frameworks involucrats en el projecte, la comprovació de l'estat real del Porta Firmes, la depuració de les millores i de codi intern relacionat amb aquestes, s'han desviat molt de les optimistes previsions.

8.4. Valoració personal

El projecte desenvolupat ha sigut molt interessant però a la vegada, més complicat del que s'esperava i amb moltes dificultats no planificades a l'inici del projecte.

Aquest projecte es pot classificar com a un projecte de manteniment, on s'han hagut de desenvolupar les millores requerides. Això ha plantejat nous reptes als que no estava acostumat, ja que per exemple la quantitat de codi a controlar és molt gran, mentre que la quantitat de codi per afegir les millores és en general molt petit, però requereix d'un grau de coneixement molt específic.

A l'inici, es tenia una idea molt difusa del funcionament del Porta Firmas. Per exemple, el propi funcionament intern del Porta Firmas (codi pur i dur, no la documentació del que es diu que es fa), els coneixements necessaris per poder modificar el codi dels diferents frameworks que s'utilitzen, la nul·la utilització de proves d'unitat, les tecnologies XML utilitzades en els Web Services, entre altres. A mesura que s'han anat guanyant els coneixements necessaris, s'han anat planificant i implementant les solucions. El punt dèbil potser ha estat el no planificar l'acabament de les tasques, allargant-les de vegades més del convenient.

Un altre tema interessant han sigut les proves, ja que no si havien fet tests d'Unitat. Per comprovar el correcte funcionament del Portal Web, s'havia de fer manualment per exemple comprovant que s'hagi signat un fitxer correctament. Aquest fet ha implicat una quantitat de temps molt gran ja que hi ha autèntics forats negres de temps (com és el cas del codi Javascript o del codi JSP) on la depuració mitjançant l'IDE Eclipse no ha sigut possible. El tema dels tests d'Unitat és el que m'ha decebut més de no poder realitzar. La meva idea inicial era anar desenvolupant les millores i provar-les mitjançant tests d'Unitat. La realitat és que per poder fer-ho d'aquesta manera, l'aplicació ha d'haver estat desenvolupada utilitzant la metodologia TDD (Test-Driven Development).

En quant a coneixements adquirits, potser els que hauria tingut que desenvolupar més són els de Struts 2 i els Web Services. En el cas dels Web Services, són necessaris coneixements de gestió de documents XML ja que pràcticament s'utilitza per tot, tant per enviar i rebre peticions, configuració, etc. Per aquest motiu, també han fet falta coneixement de XML, validació de XML, XML schema, parsers de XML (SAX, DOM,...), XML Binding (XMLBeans, JAXB) per crear un objecte a partir d'un document XML, entre d'altres. Per aquest motiu, algunes de les solucions aportades segurament es podrien millorar i d'altres no s'han pogut arribar a implementar.

També han sorgit qüestions força interessants com ara la quantitat de coneixements que són necessaris per modificar un projecte, seguir o no seguir les pautes ja marcades en el disseny del projecte i que potser són errònies, seria millor tenir la configuració del projecte en fitxers, sense dependre totalment d'un IDE.

Tot i els problemes, l'experiència ha sigut molt positiva. En general, la gestió d'un projecte és en si mateixa, una tasca molt complicada. De vegades, les opcions que es trien són solucions de compromís, donat que no sempre es disposa ni de temps ni dels coneixements necessaris. Pel que fa a l'estada a l'empresa també ha sigut agradable, en un ambient molt bo.

9. Bibliografia

Hibernate

Christian Bauer i Gavin King. Java Persistence with Hibernate. Manning. Any 2006.

SOA

Mark D. Hansen. SOA Using Java Web Services. Prentice Hall. Any 2007.

Servlets i JSP

Bryan Basham, Kathy Sierra, Bert Bates. Head First Servlets and JSP. O'Reilly. Any 2008.

Struts 2

Ian Roughley. Starting Struts 2. InfoQ. Any 2007.

Starting Struts 2, llibre disponible online (Consulta 03-05-2010):

http://www.infoq.com/minibooks/starting-struts2

Kogent Solutions Inc. Struts 2: Black book. Kogent Solutions Inc

Còpia online en Google Books. (Consulta 18-06-2010):

http://books.google.com/books?id=zrUvpI1O53wC&pg=PA453&lpg=PA453&dq=Client+side+certificate+authentication+struts+2+black+book&source=bl&ots=mErODYWyIE&sig=CuF9HMTnxiwpENEa84bs9yccOfI&hl=ca&ei=7rsXTK6VJoTB4gbGnJXkCw&sa=X&oi=book_result&ct=result&resnum=2&ved=0CBoQ6AEwAQ#v=onepage&q&f=false

Arquitectura de Struts 2 (Consulta 18-05-2010):

http://www.roseindia.net/struts/struts2/struts-2-architecture.shtml

Pàgina oficial de Struts2 (Consulta 18-05-2010): http://struts.apache.org/2.x/index.html

Tutorial d'introducció a Struts2 (Consulta 18-05-2010):

http://viralpatel.net/blogs/2009/12/introduction-to-struts-2-framework.html

CATCert (Consulta 15-06-2010): http://www.catcert.cat

Eina web de signature-e (Consulta 27-04-2010):

http://www.catcert.cat/web/cat/6_6_eines.jsp

PSIS de CATCert (Consulta 15-06-2010):

http://www.catcert.cat/web/cat/1_4_3_plataforma.jsp

@Firma

Informació de @Firma (Consulta 07-06-2010): http://www.csi.map.es/csi/pg5a12.htm

Tutorials Web online (Consulta 26-05-2010): http://www.w3schools.com/

Certificats Oficials

Certificat IDCat (Consulta 15-01-2010): http://www.idcat.net

Certificat de la FNMT (Consulta 19-01-2010): http://www.fnmt.es/

DNI Electrònic:

DNI Electrònic (Consulta 11-01-2010): http://www.dnielectronico.es/

Firma Digital

http://firmma.es/

Glossari

Glossari de termes emprats en el document.

@Firma: És el nom del conjunt d'eines i serveis que facilita l'administració publica Espanyola per donar suporta a la signatura electrònica.

A

Apache Software Foundation: És la fundació sense ànim de lucre creada per donar suport als projectes Apache.

Apache Tomcat : Servidor de servlets desenvolupat per Apache.

C

CATCert : L'Agència Catalana de Certificació. Ofereix serveis relacionats amb certificació i signatura electrònica a les administracions publiques Catalanes.

D

DSS: Digital Signature Service són les sigles per referir-nos al conjunt de Web Services que estan desenvolupats al Porta Firmes per donar serveis de signatura.

H

Hibernate : És un framework de mapeig entre els objetes d'un llenguatge de programació com JAVA i una BD relacional, mitjançant fitxers declaratius en format XML.

J

JSP : JavaServer Pages és una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP.

M

MVC : (Mode-View-Controller) és un patró de disseny que separa el model de dades, la interfície usuari i la lògica de control.

P

PSIS : Plataforma de Serveis d'Identificació i Signatura. És la plataforma tecnològica que presta el servei de validació que ofereix CATCert.

S

SIFE : Sistema Integral de Firma Electrònica desenvolupada per SPAI. Actualment anomenada Porta Firmas.

SOAP : Simple Object Access Protocol és un protocol de comunicació dissenyat per intercanviar missatges en format XML entre els Web Services.

SPAI : Empresa que va desenvolupar el Porta Firmas. Va se absorbida per CCS Agresso.

Struts 2: Desenvolupada sobre la plataforma J2EE, es un framework pel desenvolupament de web dinàmiques sota el patró de disseny MVC.

T

Time Stamp: Marca temporal afegida a una signatura electrònica. Té major validesa si ha estat feta amb una TSA.

Time Stamp Autority: És una entitat certificadora que acredita amb una marca temporal que l'hora de signatura d'un document està certificada. Té més validesa.

X

XML: eXtensible Markup Language, és un metallenguatge extensible, d'etiquetes desenvolupat pel W3C.

XML SCHEMA: Utilitzat per a descriure l'estructura i les restriccions dels continguts dels documents XML, d'etiquetes desenvolupat pel W3C.

W

Web Services : Els Serveis Web són una col·lecció de protocols i estàndards que serveix per intercanviar dades entre aplicacions diferents.

WSDL: Lenguatge descriptiu en format XML emprat pels Web Servies per definir la seva interfície pública.

Signat: Jose Manuel Vives Olaria

Bellaterra, 22 de Juny de 2010

Resum

El present projecte de final de carrera ha sigut desenvolupat en el marc d'un conveni de col·laboració entre la UAB i l'empresa Unit4.

Aquest projecte reflecteix les millores que s'han implementat en l'aplicació anomenada Porta Firmas, que és una aplicació web de gestió de signatura digital de documents electrònics de Unit4. Aquesta aplicació utilitza frameworks com ara Struts 2, Hibernate i Web Services per oferir serveis de signatura, validació i gestió de documents.

Resumen

El presente proyecto de final de carrera ha sido desarrollado en el marco de un convenio de colaboración entre la UAB y la empresa Unit4.

Este proyecto refleja las mejoras que se han implementado en la aplicación llamada Porta Firmas, que es una aplicación web de gestión de firma digital de documentos electrónicos de Unit4. Esta aplicación utiliza frameworks como Struts 2, Hibernate y Web Services para ofrecer servicios de firma, validación y gestión de documentos.

Abstract

This final-year project was developed under a collaboration agreement between the UAB and the company Unit4.

This project reflects the improvements that have been implemented in the application called Porta Firmas, which is a web application for managing digital signature of electronic documents of Unit4. This application uses frameworks such as Struts 2, Hibernate and Web Services to offer signature services, validation and document management.