Títol: Signatura digital de documents machine-readable

Volum:1/1

Alumne: Albert Fontquerni i Tomàs

Director/Ponent: Rubén Tous Liesa

Departament: AC

Data: 03 – 07 – 2009

DADES DEL PROJECTE

Títol del Projecte: Signatura digital de documents machine-readable

Nom de l'estudiant: Albert Fontquerni i Tomàs
Titulació:Enginyeria Tècnica en Informàtica de Gestió
Crèdits: 22,5
Director/Ponent: Rubén Tous Liesa
Departament: AC

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President: Jaime M. Delgado Merce

Vocal: Juan Trias Pairo

Secretari: Rubén Tous Liesa

QUALIFICACIÓ

Qualificació numèrica:
Qualificació descriptiva:

Data: 03 – 07 – 2009

Signatura digital de documents machine-readable

PFC Enginyeria tècnica en informàtica de gestió

Albert Fontquerni i Tomàs

Tutor: Rubén Tous i Liesa

Índex

1. Pròleg

Internet és una eina cada vegada més utilitzada; a Catalunya, per exemple, al 2006 ja hi havia un 46,6% de les llars que hi tenien accés, mentre que al 2008 ja eren un 60,1%[1]. Així doncs, cada vegada hi ha més ciutadans que l'utilitzen i que, per tant, en poden treure el màxim de profit.

Per altra banda, cal reduir el consum de paper en la nostra societat. Primerament, perquè cada vegada és més habitual que hi hagi grans quantitats de papers acumulats, que ocupen espai i comporten una pèrdua de temps, i en segon lloc perquè si reduïm la quantitat de paper que utilitzem, també reduirem la quantitat de residus generats per aquest.

La signatura digital és una opció que ens permet, des de l'ordinador, signar arxius. És una eina que està en expansió, per exemple el nou DNI electrònic (DNIe) la permet. El principal i gairebé únic inconvenient és que cal una generalització de la confiança dels ciutadans en les comunicacions telemàtiques.

[1] Font: Institut d'Estadística de Catalunya: http://www.idescat.cat

2. Organització de la memòria

Aquesta memòria està organitzada de forma que reflecteixi les tres etapes de l'enginyeria del software, com són especificació, disseny i implementació.

Això estarà acompanyat per una introducció inicial on s'intentarà descriure de forma resumida però precisa el contingut del projecte i tots els elements que en formen part, a més s'indicaran els objectius inicials del projecte que seran l'eix sobre el que tractarà tota la memòria.

En la primera part, s'hi pot troba una especificació del sistema, incloent casos d'us i models de comportament.

A continuació, es mostra el disseny de l'aplicació, diagrames de classes, esquemes de la base de dades,…

Tot seguit, en l'apartat d'implementació, es descriuen les principals tecnologies usades en el desenvolupament d'aquest i una visió general de com queden situades aquestes un cop implementat tot.

Per acabar, al final de la memòria s'inclou la planificació inicial i el resultat final, acompanyat d'un anàlisi de costos. Finalment, unes conclusions on es revisaran els objectius inicials i s'inclourà la opinió personal de l'autor.

3. Introducció

Tota aplicació nova ha de resoldre algun problema existent, en aquest projecte es vol desenvolupar una aplicació web que permeti la signatura digital de documents; aquesta ens permetria eliminar una bona part de la gran quantitat de papers que s'utilitzen i que després s'acumulen, així com també la necessitat de desplaçar-se a un punt per a fer només una signatura. Una aplicació web que ens permetés fer-ho digitalment ens oferiria un estalvi de paper, d'espai i de temps. La signatura digital és un mecanisme de xifrat que ens permet autenticar informació digital, utilitza un sistema de clau pública; és a dir, que si una persona firma un fitxer digitalment amb la seva clau pública el receptor pot, amb la clau pública del firmant, comprovar que ha estat realment ell qui ha signat el document.

D'altra banda, l'aplicació pot treballar amb documents el format dels quals permeti tractar-los directament amb màquines, de tal manera que pugui ser un programa, sense la intervenció manual, el que realitzi una petició de signatura d'un document o faci la verificació d'aquesta. Així, l'aplicació donarà una pas més, no tan sols incorporarà la firma digital sinó que treballarà amb documents machine-readable. Un format que compleix totes aquestes condicions és el XML. XML (de Extensible Markup Language) és un metallenguatge d'etiquetes desenvolupat pel W3C (World Wide Web Consortium) que es proposa com un estàndard per a l'intercanvi d'informació estructurada. Aquest format no conté informació sobre la presentació, i és simple tan per un programa com per una persona obtenir-ne la informació. S'utilitzaran com a exemple, doncs, uns contractes XML.

Tenim doncs, una aplicació que permetrà la firma digital de documents machine-readable; per acabar-ho de completar, s'implementarà un lloc web que permeti als usuaris comunicar-se per tal de fer peticions de signatura a altres usuaris, aquest inclourà un Applet per a realitzar signatures i verificar-les.

A més, permetrà veure els contractes en un format més fàcil de llegir per un humà que XML. Com que XML és un format fàcil de llegir per a màquines, podem transformar-lo en un document PDF a través d'un programa; demostrant així una de les moltes possibilitats que ofereix tractar amb aquest tipus de documents. D'aquesta manera, obtindríem una representació del fitxer XML signat més propera als documents en paper que s'utilitzen actualment. Es podria desenvolupar també una aplicació que generés aquest tipus de documents, o bé una que a través d'una aplicació com ara la tractada en aquest projecte fes una petició de signatura d'un document, ell mateix la verifiqués un cop feta i n'extragués la informació necessària.

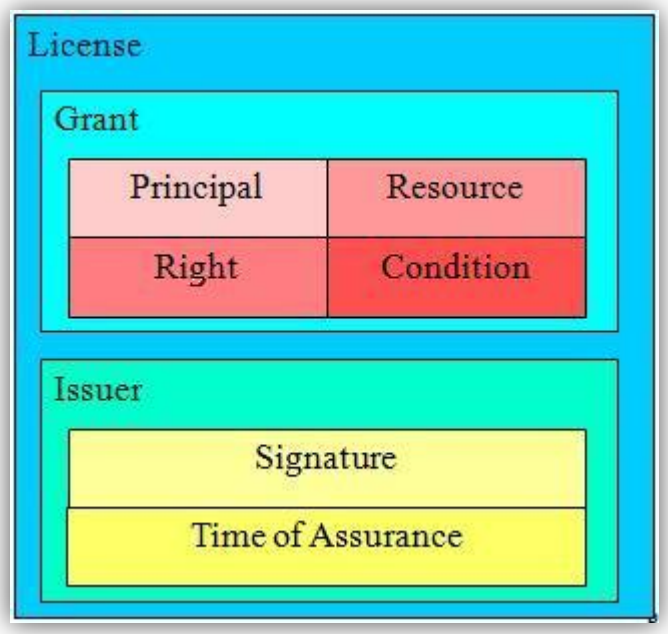
3.1. Formats existents de contractes machine-readable

L'aplicació no està encarada a cap tipus de document en concret, fer-ho tancaria la porta a molts possibles usos de l'aplicació. Així doncs, l'aplicació permet tractar amb qualsevol tipus de documents, s'ha volgut utilitzar, però, un contracte d'exemple amb l'objectiu de demostrar les capacitats de l'aplicació, ja que per exemple en la funcionalitat de transformar el contracte a un document a PDF cal tractar amb un format concret. Així doncs, farem un especial èmfasi en els contractes REL (Rights Expression Languages) a mode d'exemple.

3.1.1. Rights Expression Language (REL)

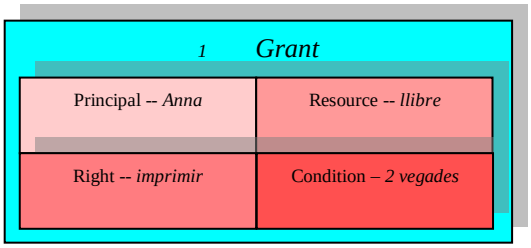
REL és un llenguatge que expressa els drets que una persona té respecte un arxiu. Aquest llenguatge és una tecnologia per a la gestió dels drets digitals i es diferencia del llenguatge jurídic perquè és un llenguatge formal que el poden interpretar computadores. D'aquesta forma, una màquina pot llegir i gairebé garantir els termes de la llicència; no pot, però, garantir-ne un ús just.

L'element bàsic d'un contracte REL és la llicència, que conté: un o més drets ("grants"), l'emissor de la llicència ("issuer"), que és qui dóna els drets; i altra informació administrativa.



Dins de cada dret, s'hi poden distingir diferents elements: el principal, on es fa referència a la persona a qui s'atorga el permís; el recurs ("resource"), que especifica sobre què se li dóna permís; el dret ("right") que s'atorga i a condició les condicions d'aquest.

D'aquesta manera, es pot donar permís a l'Anna perquè imprimeixi un llibre dues vegades:



3.1.2. Altres formats

Hi ha altres llenguatges REL com Creative Commons i ODRL. Creative Commons funciona específicament per l'entorn d'accés a la World Wide Web. En canvi ODRL, com MPEG-21, és un llenguatge de propòsit general, ambdós tenen un vocabulari que pot ser reduït o ampliat per a què tinguin un propòsit específic.

3.2. Validesa legal

La signatura digital a l'estat espanyol està regulada per la llei 34/2002, de 11 de juliol, de "Servicios de la Sociedad de la Información y de Comercio Electrónico", més coneguda com a Llei d'Internet. Els seus punts més importants són:

- Permet la possibilitat, en l'article 25, de que intervinguin tercers de confiança per arxivar les declaracions de voluntat que integren els contractes electrònics. És admissible en un judici utilitzar com a prova documental el fitxer electrònic en el que s'arxivin aquests contractes.
- A l'article 23: "Validez y eficacia de los contratos celebrados por vía electrónica", la llei recull que "Siempre que la Ley exija que el contrato o cualquier información relacionada con el mismo conste por escrito, este requisito se entenderá satisfecho si el contrato o la información se contiene en un soporte electrónico".

4. Objectius del projecte

En la introducció s'ha explicat la problemàtica del sistema actual de signatura manuscrita i com es podria solucionar aplicant la signatura digital. Un canvi que sembla inevitable, tot i algunes reticències de molts usuaris.

En aquest projecte doncs, es pretén crear un entorn per a aplicar aquest últim mètode, però cal, primerament, definir de forma clara en quins punts es treballarà, fixant així uns objectius concrets. Aquests seran:

- **Signatura:** Programar una applet Java que permeti signar digitalment un document XML (l'usuari entra el document i la clau privada). El resultat és un document signat que l'usuari podrà emmagatzemar al seu disc.
- **Validació de la signatura:** Programar un segon applet que rebi un document XML signat i amb la clau pública corresponent digui si la signatura és vàlida o no.
- **Entorn web:** Desenvolupar un entorn web on estaran inclosos els dos applets i que permetrà als usuaris comunicar-se entre ells fent peticions de signatura.
- **Mostrar un document:** Mitjançant la utilització de stylesheets XSLT, mostrar el document XML validat en un format human-readable, en concret en PDF.
- **Gestió d'esquemes XML (*objectiu opcional):** Permetre administrar un catàleg de XML schemas que permetran a l'usuari comprovar que el document XML és vàlid abans de signar-lo.

4.1. Motivacions

En aquest apartat s'explicarà els motius de l'elecció d'aquest projecte par al seu desenvolupament.

Un cop acabades les assignatures obligatòries de la carrera, només falta el projecte per a tenir el títol. Com que no tenia una idea concreta de quin projecte fer, vaig contactar amb diversos professors, per veure quins tipus de projecte portaven i així poder-ne escollir un que m'agradés i em motivés.

Així, vaig contactar amb el Rubén que em va oferir entre altres aquest projecte. De seguida em va cridar l'atenció, tot i que no sabia gaire sobre què tractava la signatura digital. Vam concertar una reunió per aclarir més bé el tema del projecte i sobre què tractava exactament. Em va semblar una opció molt interessant, ja que crec que és una eina de futur. A més, tractava amb documents XML i com transformar-los, sobre els que tenia ganes d'aprendre més.

Per tant, vaig escollir aquest projecte sobretot perquè tractava temes interessants per a mi i sobre els quals tenia ganes de saber més, i també perquè desenvolupa una eina que segurament utilitzarem en un futur.

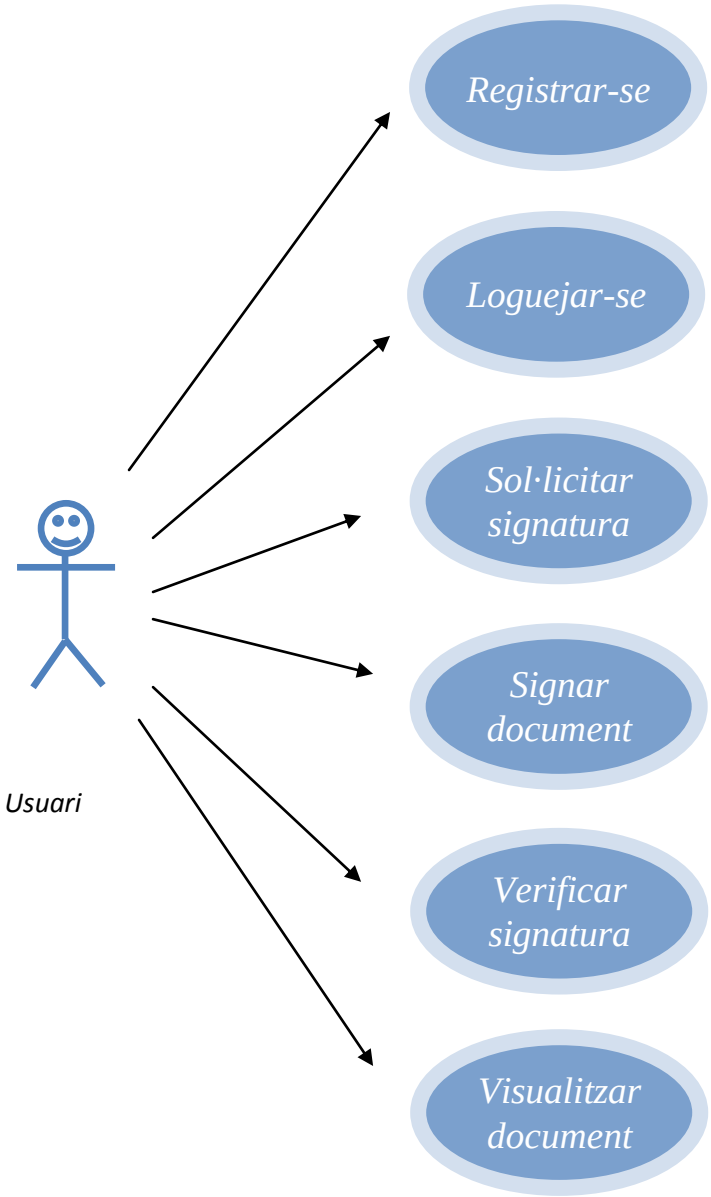
4.2. Mètode de treball

El mètode seguit en del desenvolupament del projecte és una part essencial, ja que l'elecció d'un bon mètode pot repercutir de forma clara en l'assoliment o no dels objectius fixats.

En el cas que ens ocupa, vam acordar amb el tutor de fer unes reunions aproximadament quinzenals; en les quals parlàvem i discutíem sobre la feina feta en aquestes dues setmanes i fixàvem uns objectius a curt termini per a les dues pròximes. Aquest mètode permetia veure com anava el desenvolupament del projecte, ja que si en una trobada no s'havien complert els objectius s'intentaven complir per la següent. De la mateixa manera, fixar objectius a curt termini, permet centrar-se en una feina concreta i no treballar en un objectiu massa global.

5. Especificació

En aquest sistema intervé, només, un actor; l'usuari. Aquest és l'actor que entrarà al lloc web per tal de pujar-hi contractes, signar-ne, verificar-ne i visualitzar-ne.



5.1. Model de casos d'ús

5.1.1. Un usuari es vol registrar

L'usuari entra al lloc web.

El sistema li mostra una pantalla de benvinguda on, a l'esquerra, té un menú amb les seccions on pot entrar.

L'usuari clica a l'apartat "Registra't".

El sistema li mostra una pantalla amb diferents camps per omplir: Nom d'usuari, contrasenya i un fitxer que sigui un certificat legal.

L'usuari introdueix aquests camps i clica al botó "Registra't".

5.1.2. Un usuari que es vol loguejar

Si l'usuari no està registrat -> veure punt 1.1.1, registrar-se.

L'usuari clica a la secció "Login".

El sistema li mostra una pantalla on ha d'omplir dos camps, nom d'usuari i contrasenya.

L'usuari omple els camps amb els mateixos valors que hi va posar a l'hora de registrar-se i prem el botó "Entra".

Si el logueig ha estat correcte, el sistema li mostra un missatge confirmant-ho i mostra al menú esquerre totes les seccions a les que pot accedir un usuari loguejat. En cas que no sigui correcte mostra un missatge d'error.

5.1.3. Un usuari vol sol·licitar una signatura

Si l'usuari no ho ha fet prèviament, entra a la pàgina (es logueja, veure punt 1.1.2).

L'usuari prem a la secció "Pujar contracte".

El sistema li mostra una pàgina amb un camp on ha d'introduir la ruta del fitxer.

L'usuari introdueix la ruta del contracte XML que vol que sigui signat i prem al botó "Pujar".

Si el procés és satisfactori el sistema li mostra un missatge de confirmació (en cas contrari mostra missatge d'error).

L'usuari va a la secció "Veure contractes pujats".

El sistema li mostra una llista de tots els contractes pujats per ell al servidor.

L'usuari selecciona el contracte que acaba de pujar i prem el botó "Demanar signatura".

El sistema li mostra un camp on ha d'introduir el nom d'usuari a qui vol demanar la signatura.

L'usuari introdueix l'identificador (o nom d'usuari) de l'usuari corresponent i prem el botó "acceptar".

5.1.4. Un usuari que vol realitzar una signatura (que se li ha demanat prèviament)

Si l'usuari no ho ha fet prèviament, entra a la pàgina (es logueja, veure punt 1.1.2).

L'usuari selecciona la secció "Veure signatures pendents".

El sistema li mostra una llista de tots els contractes dels quals se li ha demanat una signatura i aquesta encara no ha estat realitzada.

L'usuari selecciona el contracte que vol signar i prem el botó signar.

L'aplicació li mostra l'"applet" des del qual amb la seva clau privada i la seva clau pública l'usuari pot realitzar la signatura.

L'usuari introdueix les rutes corresponents als fitxers que contenen les claus i prem el botó "Signa el document".

El sistema li mostrarà un missatge de confirmació o d'error segons si la signatura ha estat correcta o no respectivament.

5.1.5. Un usuari que vol verificar una signatura que prèviament havia demanat

Si l'usuari no ho ha fet prèviament, entra a la pàgina (es logueja, veure punt 1.1.2).

L'usuari selecciona la secció "Veure peticions fetes signades".

El sistema li mostra una llista amb tots els documents dels quals ha demanat una signatura i aquesta ha estat realitzada.

L'usuari selecciona el document signat del qual vol verificar la signatura i prem el botó "Verificar signatura".

L'aplicació li mostrarà l'"applet" des del qual amb la clau pública corresponent podrà verificar la signatura.

L'usuari introdueix la clau pública de l'usuari que se suposa que ha signat el document.

El sistema li mostrarà un missatge de confirmació si la verificació ha estat correcta o d'error si no ho ha estat.

5.1.6. Un usuari vol visualitzar un document

L'usuari selecciona una de les seccions on es mostren els documents.

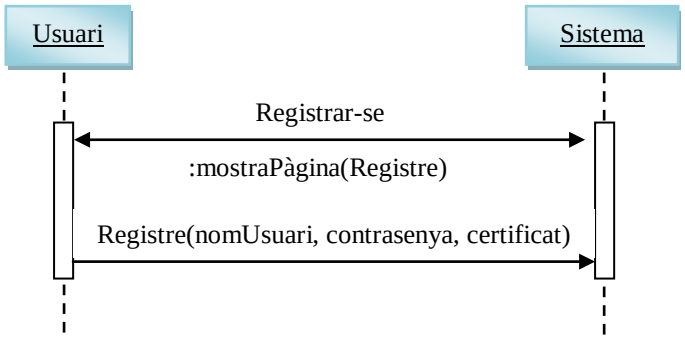
El sistema li mostra una llista dels documents que compleixen els requisits escollits.

L'usuari escull un document i clica el botó "Visualitzar contracte".

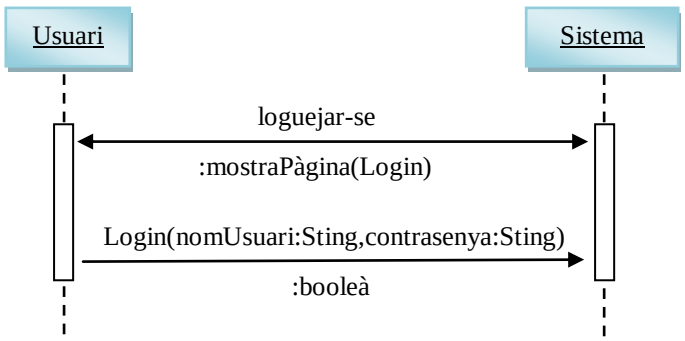
El sistema li mostra el contracte en format PDF. Si el document està signat es mostra en el document.

5.2. Model de comportament del sistema

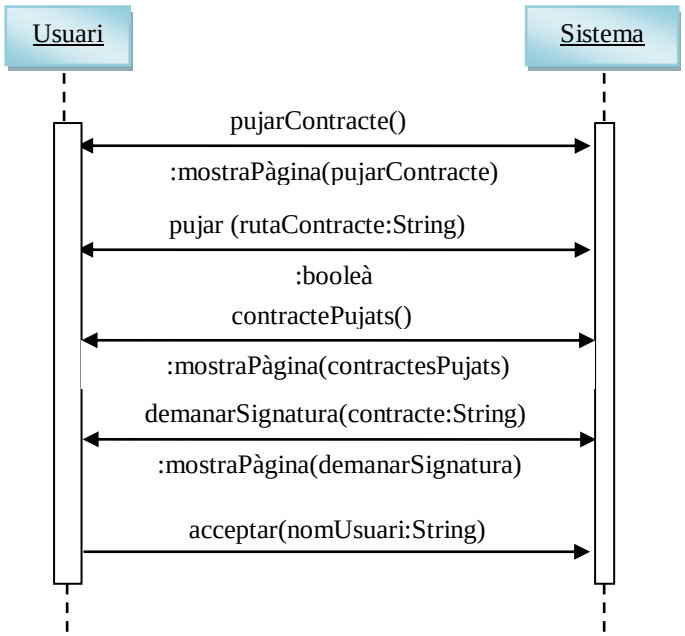
5.2.1. Registrar-se



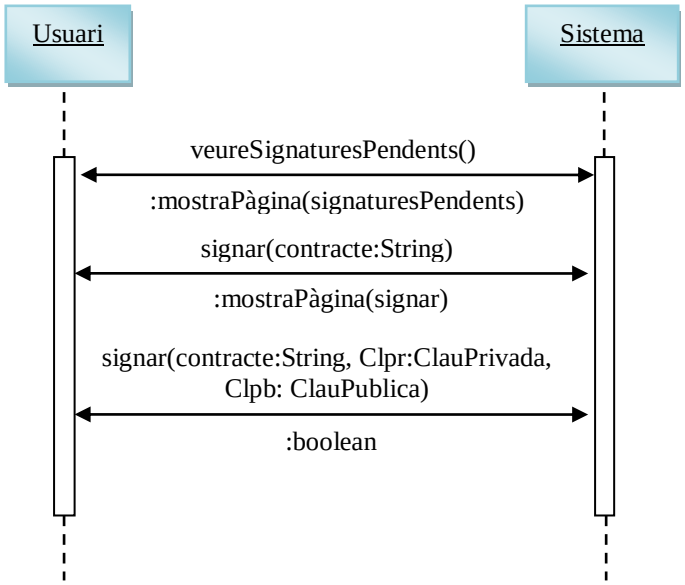
5.2.2. Loguejar-se



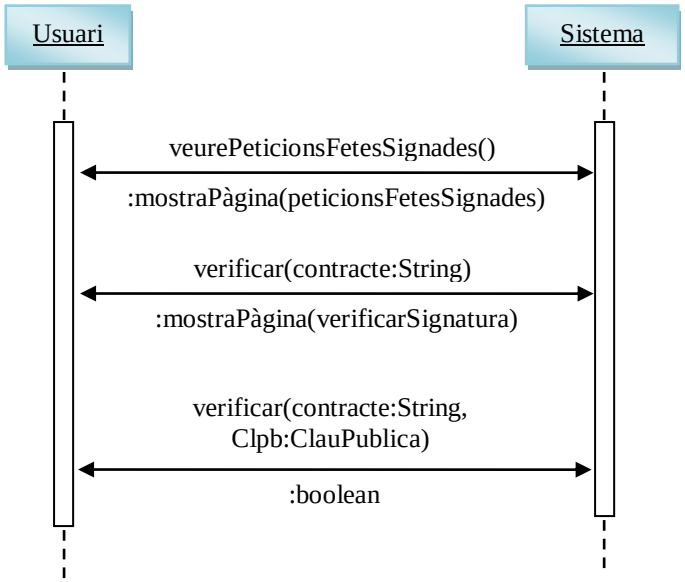
5.2.3. Sol·licitar signatura



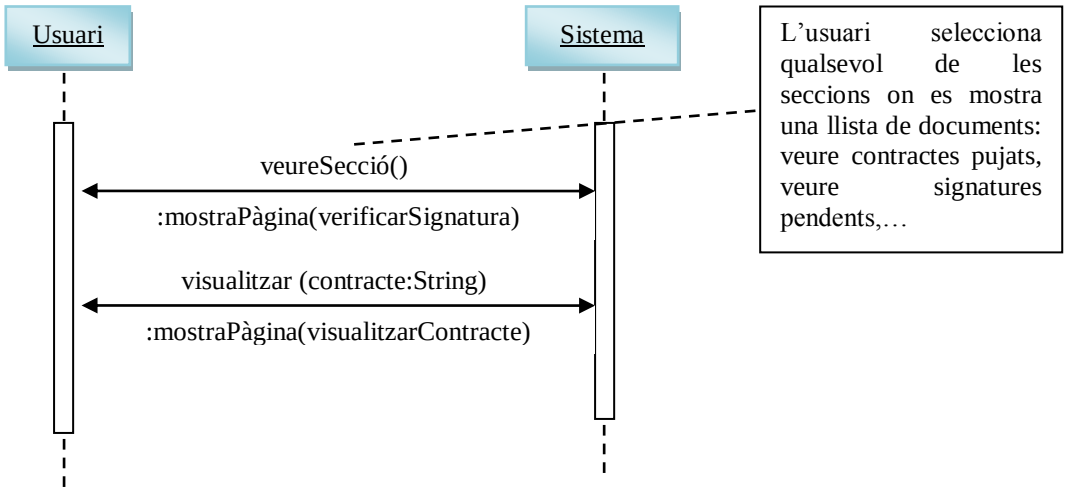
5.2.4. Signar document



5.2.5. Verificar signatura

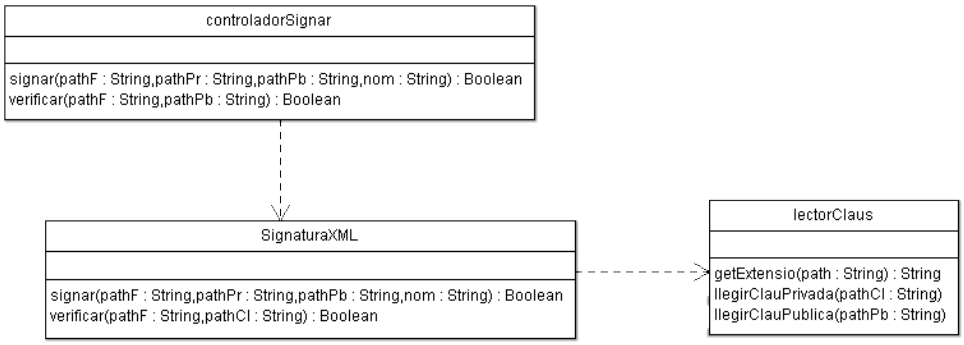


5.2.6. Visualitzar document



6. Disseny

6.1. Diagrama de classes de l'applet



6.2. Descripció del lloc web

El lloc web estarà format per un conjunt de pàgines, aquestes tindran funcionalitats diverses, per això, a continuació es fa una descripció de cadascuna d'elles. A més, hi ha una captura de pantalla de la pàgina resultant un cop implementada:

6.2.1. Índex

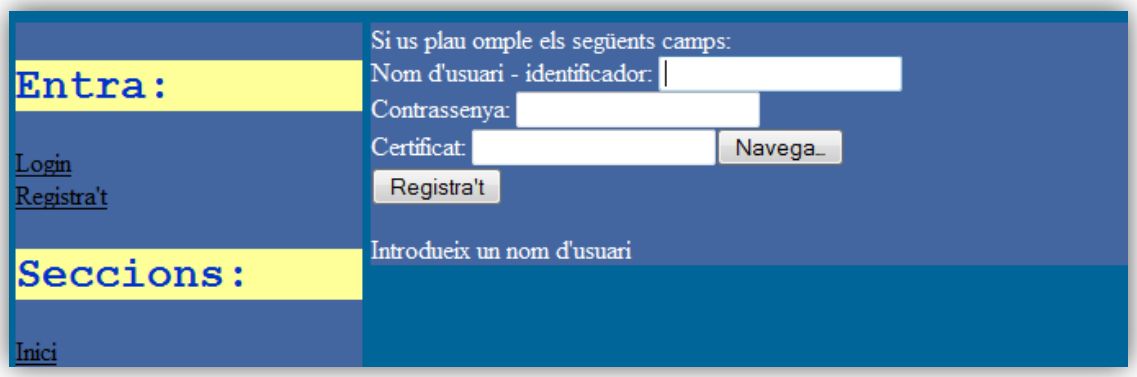
Serà la pàgina de benvinguda al lloc, mostrarà una mica d'informació sobre el que ofereix, si l'usuari està registrat, li mostrarà un missatge de benvinguda.

- Captura de pantalla resultant:

6.2.2. Registre

Com tota pàgina de registre, sol·licitarà a l'usuari unes quantes dades, entre les quals hi haurà el nom d'usuari, una contrasenya i un certificat digital. Aquest últim ens hauria de permetre identificar a l'usuari amb la seva clau pública.

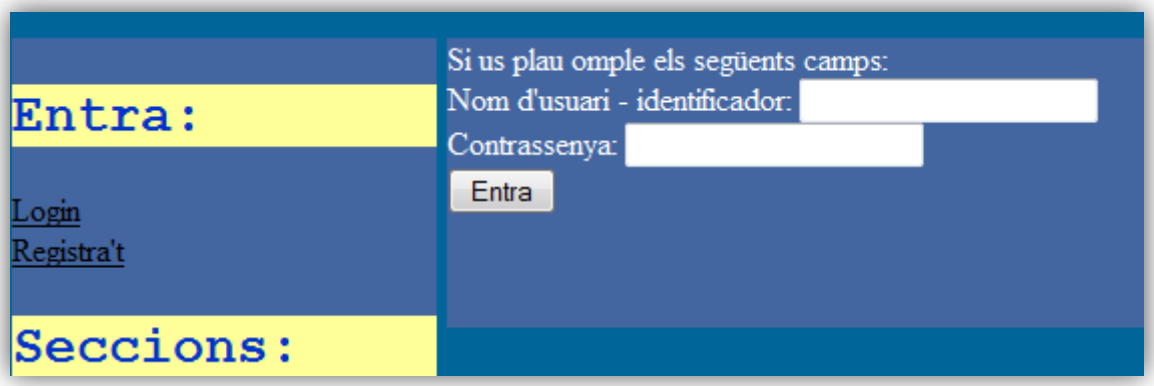
- Captura de pantalla resultant:



6.2.3. Login/entrar

Un cop l'usuari estarà registrat, li caldrà una pàgina per a identificar-se i poder accedir a les possibilitats de la pàgina. Aquesta li demanarà un nom d'usuari i una contrasenya (els que haurà introduït al registre). Un cop l'usuari hagi omplert els camps, i hagi clicat a un botó per acceptar, se li mostrarà un missatge de confirmació en cas d'èxit o d'error en cas contrari.

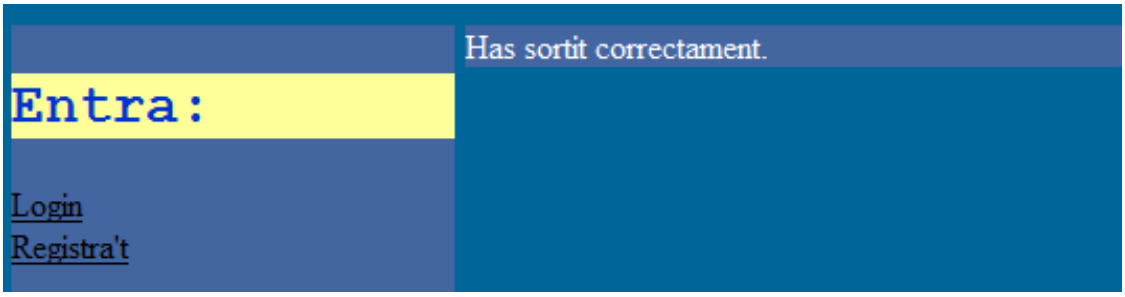
- Captura de pantalla:



6.2.4. Logout/sortir

En aquesta pàgina es permetrà a l'usuari deixar d'estar identificat en el lloc web (i, per tant, no podrà accedir a algunes pàgines), així doncs, només se li mostrarà un missatge de confirmació un cop s'hagi dut a terme aquesta acció.

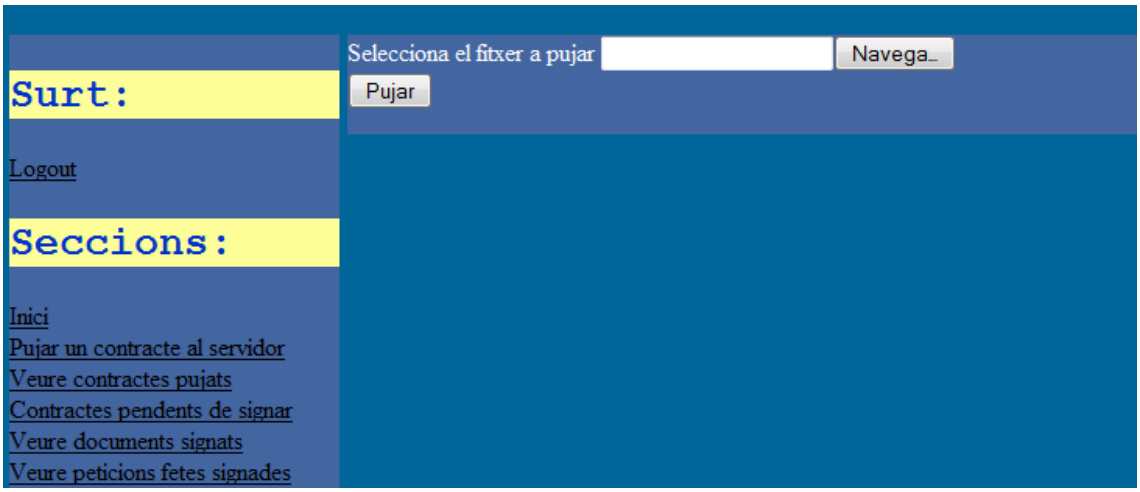
- Captura de pantalla:



6.2.5. Pujar un contracte al servidor

Des d'aquesta pàgina es permetrà a l'usuari pujar un contracte (obligatòriament en format XML) al servidor. Un cop l'usuari l'hagi pujat, podrà sol·licitar-ne signatures. Així, contindrà un camp que haurà d'omplir amb la ruta que contingui el fitxer desitjat.

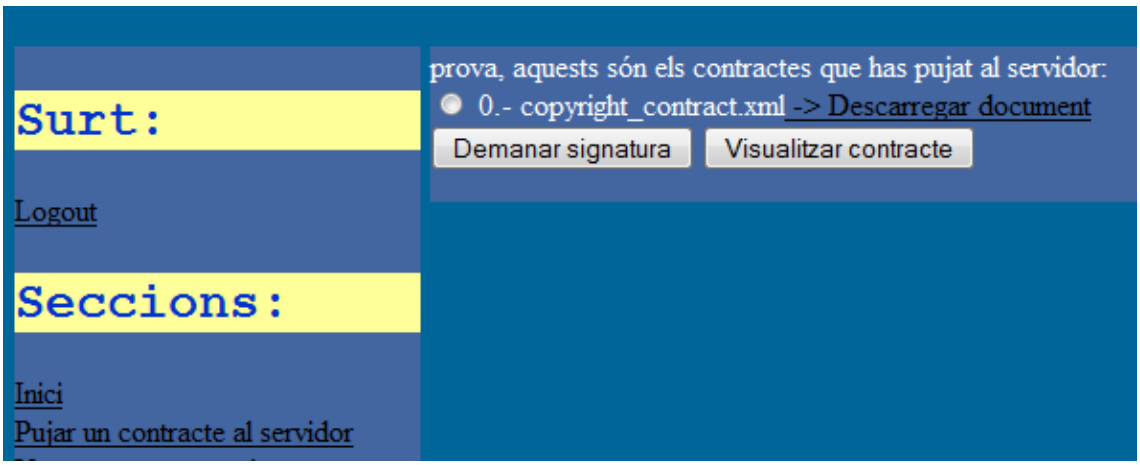
- Captura de pantalla:



6.2.6. Veure contractes pujats

Des d'aquesta pàgina es mostrarà una llista dels contractes que l'usuari ha pujat al servidor. En cas que no n'hagi pujat cap, es mostrarà la llista buida. Des d'aquesta pàgina es podrà seleccionar un dels contractes mostrats per demanar-ne una signatura o bé per visualitzar-lo en format PDF.

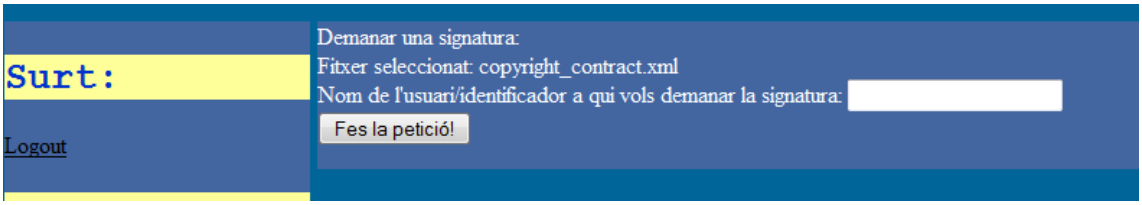
- Captura de pantalla:



6.2.7. Demanar signatura

Aquesta pàgina permetrà fer una petició de signatura del fitxer seleccionat. Es demanarà a l'usuari que introdueixi l'identificador de l'usuari desitjat. Un cop fet això, es comprovarà l'existència de l'usuari i es mostrarà un missatge de confirmació en cas que s'hagi pogut fer la petició o d'error en cas contrari.

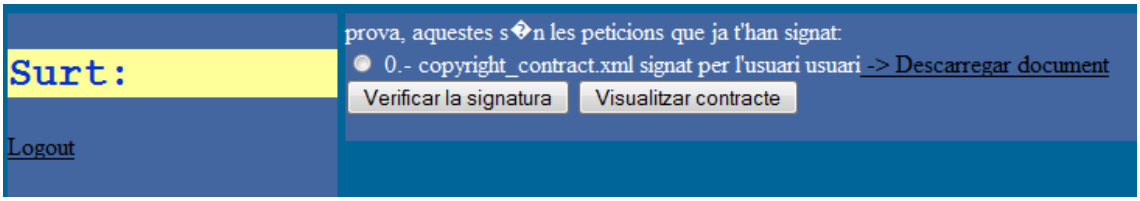
- Captura de pantalla:



6.2.8. Veure peticions signades

Un cop s'hagi fet una petició, l'usuari a qui hagi estat feta podrà signar el document seleccionat en el moment de la petició. Quan ho hagi fet, l'usuari que li havia sol·licitat la signatura podrà veure-ho en aquest apartat. Així, la pàgina li mostrarà una llista amb totes les peticions que ha fet i que han estat signades. L'usuari podrà descarregar qualsevol dels documents de la llista o bé escollir-ne un per visualitzar-lo en format PDF o per verificar-ne la signatura.

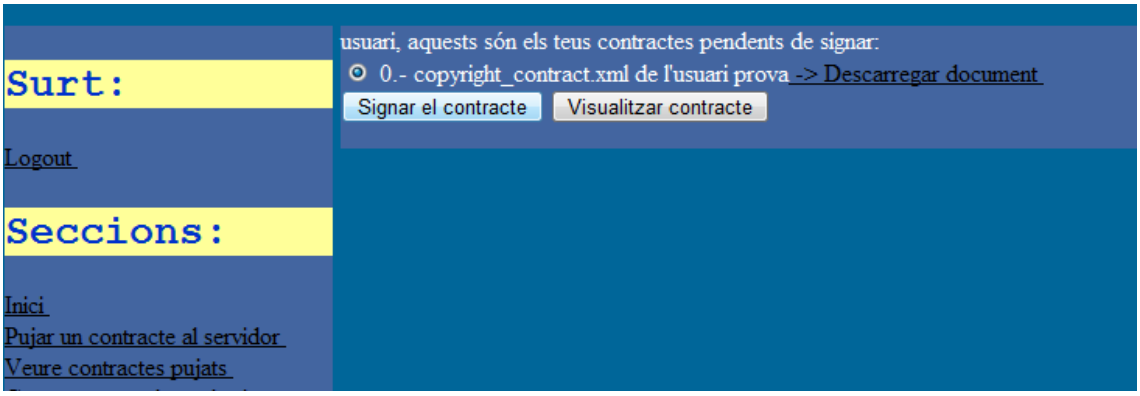
- Captura de pantalla:



6.2.9. Veure signatures pendents

Un usuari podrà, en qualsevol moment, veure els documents pels quals se li ha sol·licitat una signatura. Aquesta pàgina li mostrarà una llista amb tots els documents acompanyats del nom de l'usuari propietari de cadascun d'ells. L'usuari podrà descarregar-los o bé seleccionar-ne un per a signar-lo o per veure'l en format PDF.

- Captura de pantalla:



6.2.10. Signar

Aquí es permetrà a l'usuari signar un document, que prèviament haurà seleccionat d'entre les peticions que li hagin fet. La pàgina contindrà l'applet que li permetrà signar el document XML. Un cop feta la signatura, se li mostrarà un missatge comunicant-li si hi ha hagut èxit o s'ha produït algun error en el procés.

- Captura de pantalla:

6.2.11. Veure documents signats

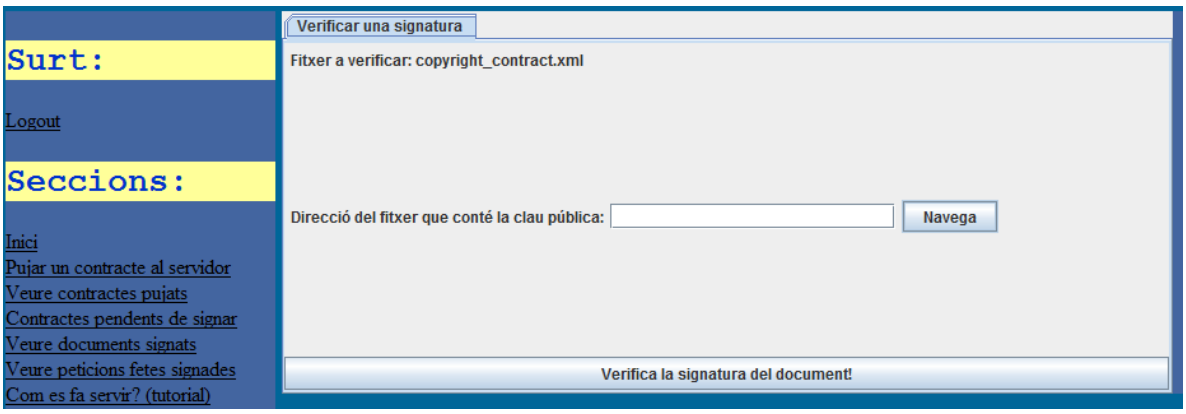
Des d'aquesta pàgina, l'usuari podrà fer un seguiment de tots els documents que ha signat; així, se li mostrarà una llista amb tots aquests, els quals podrà descarregar o bé visualitzar en format PDF.

- Captura de pantalla:

6.2.12. Verificar signatura

Aquesta pàgina permetrà a l'usuari verificar una signatura d'un document que prèviament hagués sol·licitat. Així, podrà verificar la signatura del document que hagi seleccionat a la llista de documents signats. Se li mostrarà doncs, un applet on l'usuari podrà introduir la clau pública de l'usuari que ha realitzat la firma; seguidament se li mostrarà si la verificació de la firma ha estat correcta o no.

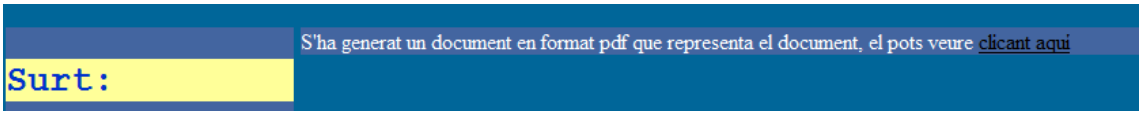
- Captura de pantalla:

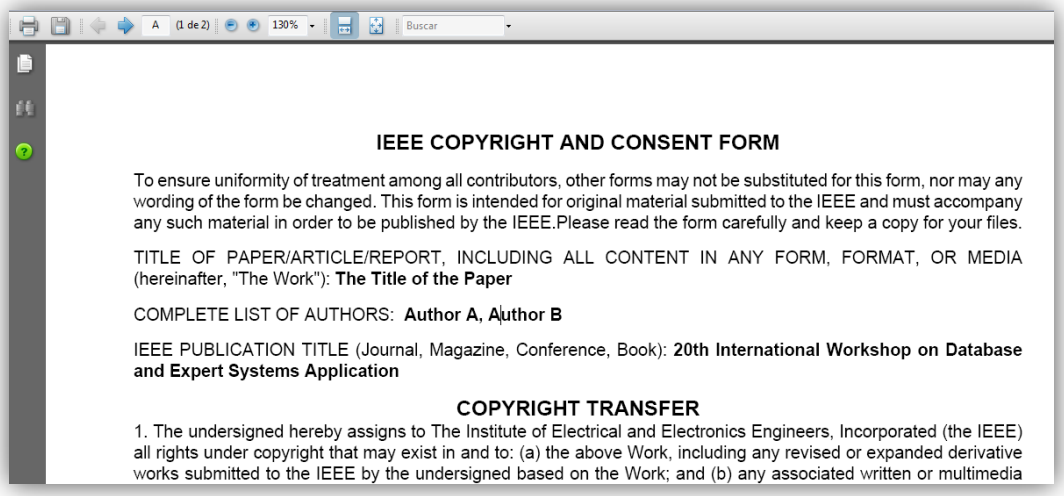


6.2.13. Visualitzar document

Aquesta pàgina permet a l'usuari veure una representació en format PDF del document que prèviament hagi seleccionat. Així, a la pàgina es generarà el document PDF a partir del document XML i es mostrarà l'enllaç al fitxer. Si es clica l'enllaç es podrà veure i descarregar el document PDF.

- Captures de pantalla:





IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE.Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT, INCLUDING ALL CONTENT IN ANY FORM, FORMAT, OR MEDIA (hereinafter, "The Work"): **The Title of the Paper**

COMPLETE LIST OF AUTHORS: **Author A, Author B**

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book): **20th International Workshop on Database and Expert Systems Application**

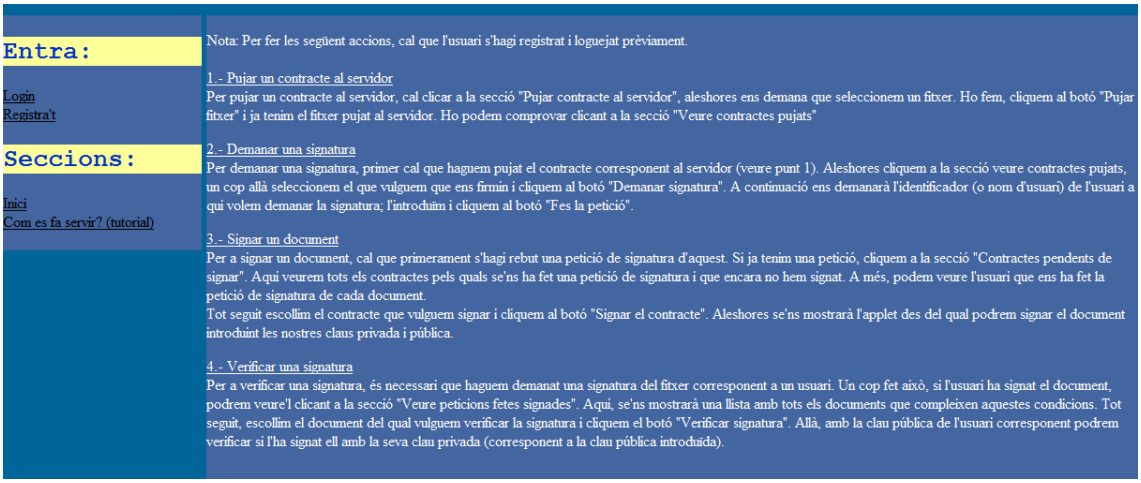
COPYRIGHT TRANSFER

1. The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the IEEE) all rights under copyright that may exist in and to: (a) the above Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia

6.2.14. Tutorial

Pàgina que contindrà una mica d'informació sobre l'ús de la pàgina. En aquesta es mostrarà un text que explicarà com dur a terme les funcionalitats de l'aplicació

- Captura de pantalla:



6.3. Esquema de la Base de Dades

La base de dades conté les següents taules:

Usuari		
Camp	Tipus de dada	Nul?
id	text/"varchar"	No nul
nom	text/"varchar"	No nul
contrasenya	text/"varchar"	No nul
pathCert	text/"varchar"	No nul

Amb clau primària id, l'identificador de l'usuari.

En aquesta taula, es guardaran tots els usuaris que es registrin al lloc web. L'id serà el nom d'usuari que l'usuari introdueixi al registrar-se, el mateix passarà per la contrasenya. Pel què fa al camp nom, contindrà el nom i cognoms reals de l'usuari, que es llegiran del certificat introduït en el moment de registrar-se. PathCert contindrà el tros de text que identifica on està desat el certificat al servidor.

Contracte		
Camp	Tipus de dada	Nul?
idContracte	enter (autoincremental)	No nul
idPropietari	text/"varchar"	No nul
document	text/"varchar"	No nul
nom	text/"varchar"	No nul

Amb clau primària idContracte.

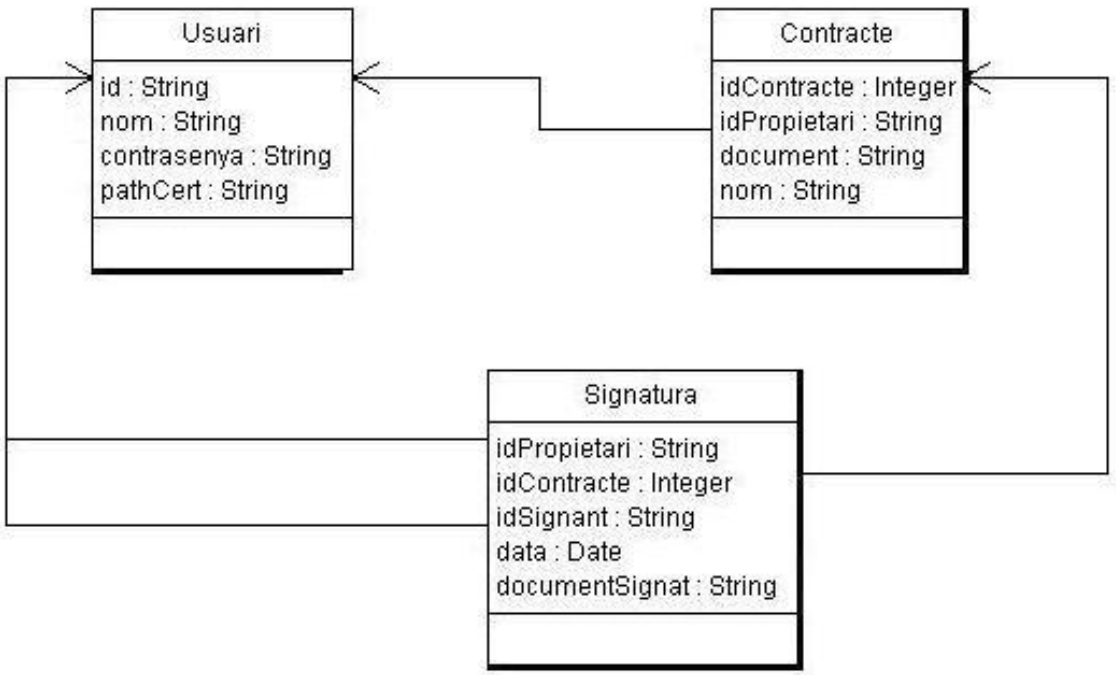
Un contracte estarà identificat internament per un enter, idContracte, que s'assignarà automàticament de forma incremental al introduir-lo a la base de dades. El camp idPropietari farà referència a l'identificador de l'usuari que ha pujat el contracte al servidor, mentre que document serà la ruta al lloc on està guardat el contracte al servidor. Finalment, el nom representarà el nom del fitxer.

Signatura		
Camp	Tipus de dada	Nul?
idPropietari	text/"varchar"	No nul
idContracte	enter	No nul
idSignant	text/"varchar"	No nul
data	date	
documentSignat	text/"varchar"	

Amb clau primària el triplet idPropietari, idContracte i idSignant, de tal manera que un usuari pot demanar una signatura d'un mateix document a diferents usuaris o bé a un mateix usuaris signatures de contractes diferents.

El camp idPropietari fa referència a l'identificador de l'usuari que ha pujat el contracte del qual es demana la signatura al servidor, idContracte és l'identificador d'aquest contracte i idSignant fa referència a l'identificador de l'usuari al qual se li ha demanat la signatura. Finalment els camps data i documentSignat contindran la data de la signatura i la ruta al document XML signat respectivament, si el document no ha estat signat tindran valor nul.

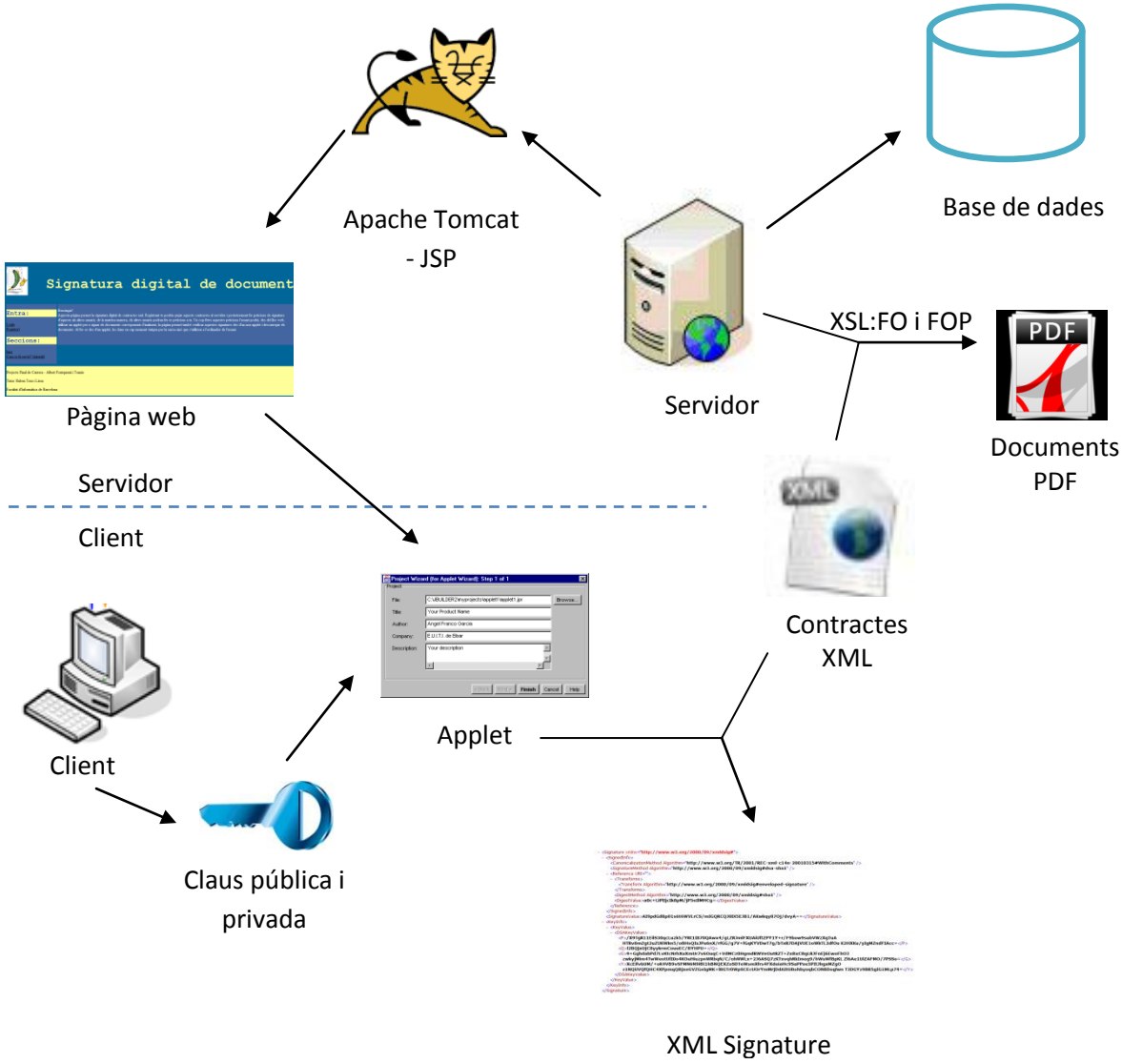
Com que utilitzarem una base de dades relacional, podem introduir claus foranes. Així doncs, en el següent diagrama es mostren les relacions entre aquestes taules. Cada fletxa representa una clau forana, on la columna de l'extrem on comença fa referència l'element de l'extrem on acaba la fletxa.



7. Implementació

7.1. Arquitectura del sistema

En aquest apartat es mostrarà una visió global de sistema, quines són les diferents peces que en formen part i com encaixen entre elles. Vegem-ne primer un esquema:



Com es veu en l'esquema, el servidor disposarà d'una pàgina web dinàmica, feta amb JSP. També tindrà una base de dades, per tal de tenir emmagatzemada tota la gestió d'usuaris, de contractes i de signatures de forma ordenada. D'aquesta forma podrem saber consultant a la base de dades quins usuaris s'han registrat al sistema, quins documents ha pujat cadascun

d'ells… La pàgina web permetrà als diferents usuaris fer peticions de signatura a altres i posteriorment verificar-les, de la mateixa manera que permetrà veure les peticions que hagin fet a cada usuari i signar-les. Per a aquest motiu, contindrà l'applet que, amb les claus corresponents, ens permetrà signar i verificar documents XML. Cal remarcar, però, que es fa amb un applet i no directament a través de la pàgina web per tal de que les claus no viatgin per la xarxa en cap moment, ja que els applets s'executen localment i no en el servidor. Així, aquestes claus només estan disponibles pel client i per l'applet. D'aquesta manera, el client podrà signar i verificar signatures de contractes XML fent ús de la tecnologia XML Signature. Aquests documents XML, tant l'inicial com el signat, estaran allotjats al servidor; tot i que l'usuari podrà descarregar-los en qualsevol moment. També podrà visualitzar-los en format PDF; en aquest cas, el servidor, de forma automàtica i utilitzant les tecnologies XSL:FO i FOP podrà generar documents PDF que representin a cadascun d'ells.

7.2. Tecnologies Usades

7.2.1. *Signatura digital*

La signatura digital o signatura digital de clau pública és un mecanisme de xifrat que permet autenticar informació digital. La signatura consisteix en un afegit al missatge que indica autoria, conformitat o altres amb el contingut del missatge signat. Aquests sistemes proporcionen, doncs, serveis d' identificació, d'integritat i de no repudi. A Espanya hi ha una llei que equipara el valor de la signatura digital al de la manuscrita: "*La firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica el mismo valor que la firma manuscrita en relación con los consignados en papel*". Hi ha, però, diferències amb la signatura manuscrita:

- La signatura digital no pot dependre només de qui firma, ha de dependre també del què es firma, del missatge que l'acompanya. Per tant, la verificació d'una signatura digital no es pot fer comparant-la amb la d'un document anterior.
- La signatura digital no permet distingir un document original d'una còpia.

Per a realitzar una signatura digital; primer, cal que la persona que vol signar disposi d'una clau pública i una clau privada. La clau pública ha de dependre de la privada, però de tal forma que no es pugui calcular la privada a partir de pública. Per exemple: (clau pública = $g^{clauPrivada}$) mòdul q; on g i q han de tenir unes característiques especials (g ha de ser generador d'un subgrup d'ordre q).

Un cop l'usuari té la clau es procedeix de la següent manera, com s'explica a la viquipèdia:

"Quan un usuari A vol enviar un missatge a un usuari B i vol que l'usuari B estigui segur que el missatge prové d'ell se segueixen els següents passos: L'usuari A envia el seu missatge a l'usuari B i adjunta una signatura digital. Aquesta signatura es genera usant la clau privada de l'usuari A i pren la forma d'un valor numèric. En rebre el missatge, l'usuari B pot confirmar la procedència del missatge utilitzant la clau pública de l'usuari B, la signatura i el missatge. Si la verificació és correcta l'usuari B pot estar segur que el missatge procedeix de l'usuari A ja que l'algorisme de signat està dissenyat per a que sigui molt difícil crear una signatura que encaixi amb un missatge concret(si no es coneix la clau privada)."

7.2.1.1. Certificats digitals

Un certificat digital és un document electrònic que associa una clau pública amb la identitat del seu propietari. Aquests fitxers els reparteixen les autoritats de certificació (A.C. o C.A. en anglès), que són una tercera part de confiança entre l'emissor i el receptor del missatge; així, si tots dos confien els documents signats per aquesta autoritat poden confiar en els documents que donen fer del lligam entre clau pública i propietari.

L'autoritat certificadora no s'encarrega de generar les claus, en tot cas pot ser responsabilitat d'un sistema associat. Un cop un usuari té les claus, pot acudir a una autoritat perquè li faci un certificat seguint els següents passos:

- L'usuari envia la clau pública a una autoritat certificadora, on queda registrada.
- Aquesta autoritat comprova la identitat de l'usuari. L' única forma vàlida de comprovar la identitat de l'usuari és exigint la presència física i la documentació (DNI).
- L'autoritat genera i firma amb la seva clau privada un certificat que envia a l'usuari i que opcionalment es pot guardar també.

Una autoritat certificadora ofereix els següents serveis:

- Emissió i distribució de certificats.
- Revocació, suspensió i renovació de certificats.
- Cerca de certificats.
- Informació sobre l'estat d'un certificat (llistes de revocació CRL).
- Recuperació de claus.
- Verificació de signatura digital.
- Replicació del contingut en directoris d'accés públic.
- Datació digital.
- Confirmacions d'enviament, entrega i recepció.
- Certificats de continguts.

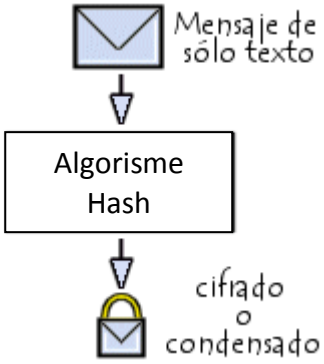
Un format estàndard per als certificats és el format X.509 i té els següents camps:

- **Versió:** Número de versió del certificat, pot ser 1, 2 o 3.
- **Número de sèrie del certificat:** Un enter assignat per l'autoritat certificadora. Cada certificat emès per una mateixa autoritat té un número de sèrie únic.
- **Identificador de l'algorisme de signatura:** Especifica l'algorisme usat per la autoritat per a signar el certificat.
- **Emissor:** Identifica la AC que ha signat i emès el certificat.
- **Període de validesa:** Interval de temps durant el qual la AC garanteix que mantindrà informació sobre l'estat del certificat.
- **Subjecte:** Identifica l'entitat associada amb la clau pública emmagatzemada al certificat.

- **Informació de la clau pública del subjecte:** Conté la clau pública i la identificació de l'algorisme per al qual es destina.
- **Identificador únic de l'emissor/subjecte:** Per permetre reutilitzar noms.
- **Extensions:** Proporcionen una manera d'associar informació addicional a subjectes, claus públiques…

7.2.1.2. Algorismes de hash

Per a realitzar una signatura, el què signem és un hash del missatge. Un hash és un resum; per obtenir-lo, s'utilitza una funció matemàtica. En general, la sortida en aplicar una funció de hash serà més petita que l'entrada. Un algorisme de hash ha de complir una sèrie de propietats:



- Compressió: Normalment cal que els resultats d'una funció hash tinguin una longitud fixada, es pot pensar doncs que la funció hash fa la compressió de les dades en un bloc d'una mida justa.
- Unidireccionalitat: Donat un missatge és fàcil/ràpid calcular el hash, en canvi donat un hash és impossible reconstruir el text.
- Difusió: Una funció hash ha de ser una funció complexa de tots els bits del missatge.
- Feblement lliure de col·lisions: Donat m, és computacionalment impossible trobar m' ≠ m tal que H(m') = H(m).
- Fortament lliure de col·lisions: És computacionalment impossible trobar un parell m i m' tal que H(m) = H(m').

Actualment un dels algorismes més usats i que s'utilitzarà en el projecte és el sha1, del qual no se'n a trobat mai cap col·lisió; tot i això es comença posar en dubte la seva seguretat i es comença a recomanar l'ús del sha-256.

7.2.1.3. Algorisme de signatura (DSA: Digital Signature Algorithm)

Un cop obtingut el hash del missatge es fa la firma d'aquest. Com s'ha dit, cal que l'usuari disposi de una clau privada i una de pública. Per a fer-ho cal tenir un primer p de 1024 bits i un primer q de 160 bits divisor de p. Per altra banda, ens cal un element g, generador d'un subgrup d'ordre q a F^*_p. Aquest valors seran públics i comuns per un grup d'usuaris a qui es reparteixin les claus. Amb aquests paràmetres podem generar la clau privada com a un enter r aleatori mòdul q-1 i la pública com a un valor $u = g^r$ mòdul p. El problema d'obtenir la clau privada a partir de la pública és equivalent al problema del logaritme discret. Amb les claus i el hash del missatge, ja es pot procedir a la signatura, es procedeix de la següent forma:

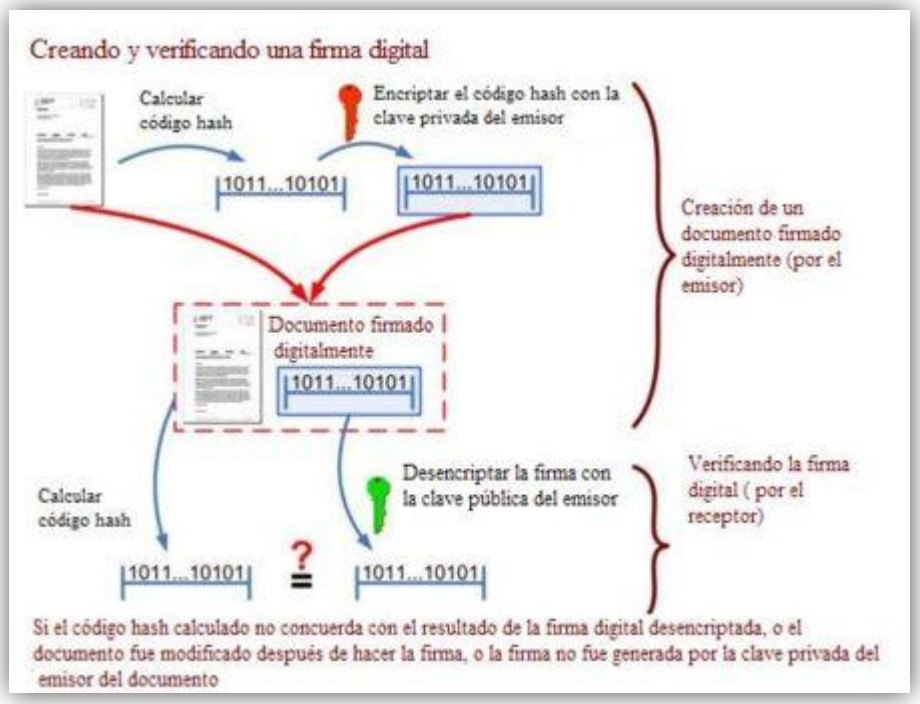
1. Es genera un enter k aleatori mòdul q.
2. Es calcula $f^1 = (g^k \bmod p) \bmod q$.

3. Es calcula $f^2 = k^{-1}(SHA(m) + f^1 r) \bmod q$.
4. Si f1 o f2 són iguals a zero, es repeteix el càlcul.

Finalment, la signatura del missatge és s = (f1, f2).

Per a la verificació de la signatura cal disposar de p, q, g i la clau pública de l'emissor u. A més s'ha rebut m (el missatge) i f_1 i f_2, la signatura. Tot seguit, es procedeix de la següent forma:

1. Es calcula $w = (f_2)^{-1} \bmod q$.
2. Es calcula $w_1 = SHA(m)*w \bmod q$.
3. Es calcula $w_2 = f_1 * w \bmod q$.
4. Finalment es calcula $v = (g^{w1} * u^{w2} \bmod p) \bmod q$
5. La signatura és validada si $v = f_1$.



7.2.1.4. XML Signature

XML Signature és una recomanació del W3C (World Wide Web Consortium) que defineix una sintaxi XML per a la firma digital. És similar al PCKS7 però més orientat a la firma de documents XML.

Aqueta signatura pot ser de dues formes, continguda en el document o apart. En el cas s'ha escollit la signatura present en el mateix document. Així, s'afegeix al document XML una part, la firma, que segueix el següent esquema:

```
<Signature>
 <SignedInfo>
  <SignatureMethod />
  <CanonicalizationMethod />
  <Reference>
    <Transforms>
    <DigestMethod>
    <DigestValue>
  </Reference>
  <Reference /> etc.
 </SignedInfo>
 <SignatureValue />
 <KeyInfo />
 <Object />
</Signature>
```

On els camps més importants són:

- SignedInfo: Conté o fa referència les dades signades i especifica quin algorisme utilitza.
- DigestMethod: Especifica l'algorisme de hash.
- DigestValue: El resultat d'aplicar l'algorisme de hash als recursos un cop transformats.
- SignatureValue: Conté el resultat de la signatura codificat en base64.
- KeyInfo: Conté la clau pública del signant, és un element opcional. En cas de no disposar de la clau pública del signant, el receptor del missatge podrà extreure-la d'aquest element.

D'aquesta forma, en visualitzar un document XML signat, un cop omplerts aquests camps, es pot veure quelcom similar al següent:

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
                20010315#WithComments" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
     <Reference URI="">
        <Transforms>
           <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>a0c+LiPlIjcIk8pM/jP5eIlMHCg=</DigestValue>
     </Reference>
 </SignedInfo>

<SignatureValue>AI9pdGdBp01s6t6WVLrCS/mIGQRCQJ0Di5EJB1/AKwkqy87Oj/dvyA==</Signature
Value>
  <KeyInfo>
    <KeyValue>
     <DSAKeyValue>
```

<P>/X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9subVWzXg TuA
HTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL 2dfOu K2HXKu/yIgMZndFIAcc=</P>
<Q>l2BQjxUjC8yykrmCouuEC/BYHPU=</Q>

<G>9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT+ZxBxCBgLRJFn Ej6EwoFhO3
zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx+2J6ASQ7zKTxvqhRkImog9/hWuWf BpKL Zl6Ae1UlZAFMO/7PSSo=</G>

<Y>XcEifvbUM/+oK4VB9vSFMN6N9iBt1hBNQEXZoSDTeWsmXfrs4FXdeiaHc9SaPPxe5PItJhgxMZ gOz1NQUVQfQHC4XPpmqQBjxeGVZGeIgMK+lBGTr0WpSCEcUOrYmMrjDdAItUBohbyoqbCONB Doglwn T3DGYz9BR5gIG1MLp74=</Y>
</DSAKeyValue>
</KeyValue>
</KeyInfo>
</Signature>

Aquí, podem observar per exemple com s'inclouen en la signatura els paràmetres p, q i g (marcats en negreta) que com hem explicat a l'apartat 7.2.1 Signatura digital són també necessaris per a la verificació. Per altra banda podem veure que s'especifiquen els algorismes utilitzats en la signatura a: <SignatureMethod Algorithm= "http://www.w3.org/2000/09/ xmldsig#dsa-sha1" />. Es veu clarament també que aquest document segueix un format XML i que, per tant, pot ser llegit per computadores.

7.2.2. Applets

Un applet de Java és un tipus d'aplicació escrita en el llenguatge de programació Java que pot executar-se en un navegador web, de tal forma que es pot incrustar l'applet en un tros de codi HTML. Quan el navegador carrega la pàgina, aquest es descarrega a la màquina del client i comença a executar-se. Algunes de les característiques dels applets són:

- **Són multiplataforma:** Funcionen en qualsevol sistema operatiu pel qual existeixi una màquina virtual de Java.
- **És suportat per la majoria de navegadors.**
- **S'executa en el client:** Permet traslladar el treball del servidor al client. Aquest és un punt clau en el cas que en ocupa ja que permetrà que les claus en cap moment viatgin per la xarxa
- **Necessiten un navegador per ser visualitzats:** O bé un visor especial anomenat appletviewer.
- **No té un mètode principal:** Un applet pot tenir els següents mètodes principals:
  - Init: S'executa quan l'applet és carregat, només un cop en el seu cicle de vida.
  - Start: S'executa com a mínim una vegada, és executat per primer cop després del mètode init; a més, es crida cada vegada que accedim de

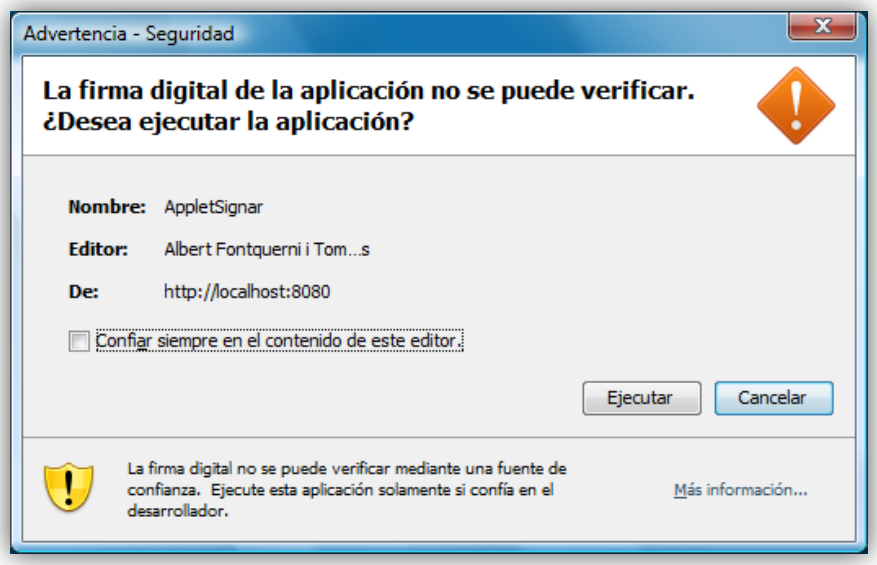
nou a la mateixa pàgina. Moltes vegades els applets l'hereten de la superclasse, ja que el seu ús més habitual és executar fils ("threads") necessaris per a l'applet.

- Stop: És cridat cada cop que el navegador deixi la pàgina, com a mínim serà executat un cop. Se sol utilitzar per aturar fils d'execució, tot i que molts cops els applets no en tenen i l'hereten de la superclasse.
- Destroy: És executat un sol cop en el cicle de vida de l'applet, quan es tanca el navegador. Sol ser usat en cas de necessitat de guardar algunes dades al servidor abans de tancar definitivament l'applet.

- **Té restriccions de seguretat:** Per motius de seguretat no es permet que els applets accedeixin a parts sensibles de l'ordinador del client. Per exemple no es permet llegir ni escriure-hi fitxers, ja que podríem fer per exemple un applet que esborrés el disc dur de tothom qui el visiti. En cas necessari, però, el client pot donar-li permisos per a fer-ho si l'applet està signat.

7.2.2.1. Applets signats

A vegades, per al funcionament d'un applet és necessari que pugui accedir al disc. La forma d'aconseguir això és firmar l'applet digitalment. Un applet signat ens permet identificar a l'autor d'aquest. D'aquesta forma a l'executar l'applet es mostrarà a l'usuari un missatge identificant a la persona que l'ha signat i preguntat a l'usuari si hi confia:



Per a signar-lo, es pot generar un fitxer jar amb els fitxers de l'applet i utilitzar la eina jarsigner, que ens permet signar-lo amb una clau que tinguem.

7.2.3. JSP

JSP (de Java Server Pages) és una tecnologia Java desenvolupada per Sun Microsystems que permet generar contingut dinàmic per a la web. En la creació d'una pàgina web amb JSP, s'introdueixen dins el codi HTML etiquetes especials per programar en codi Java. El fet de poder usar API's de Java fa de JSP una eina molt potent, ja que permet la programació orientada a objectes, la independència de plataforma que ofereix Java,…

De forma similar al què ocorre en altres llenguatges per a la creació de pàgines web dinàmiques com PHP o ASP el codi inclòs entre el codi HTML s'executa en el servidor en la primera sol·licitud de la pàgina. Uns exemples de codi JSP són:

<% int localStackBasedVariable = 1;

out.println(localStackBasedVariable); %>

On s'inicia una variable i s'imprimeix per pantalla.

<%@ include file='capcalera.jsp' %>

En aquest cas s'importa una altra pàgina capcalera.jsp al fitxer actual.

```
<%
    String nom = (String) request.getSession().getAttribute("nom");
    if(nom != null) out.println("Benvingut "+nom+"!");
    else out.println("Benvingut!</br>");
%>
```

Aquí s'agafa un paràmetre de la sessió, el nom, i mostra un missatge de benvinguda que inclou el nom en cas que existeixi.

JSP és molt similar a un servlet, amb la diferència que els servlets estan prèviament compilats i implementen la classe HttpServlet.

Un servidor web tan per a Servlets com per a JSP és Apache Tomcat, una aplicació escrita en Java que manté en execució la Màquina Virtual per tal de compilar els arxius JSP i executar els servlets. Tomcat ofereix doncs un entorn per a executar el codi Java en cooperació amb un servidor web, aquest últim afegeix eines per a la configuració i el manteniment, però també pot ser configurat editant els fitxers de configuració que normalment són en format XML.

Apache Tomcat es desenvolupa en un entorn obert i participatiu i publicat sota la llicència del programari d'Apache.

7.2.4. MPEG – 21 REL

Un dels llenguatges REL principals és el MPEG-21, és un llenguatge de propòsit general que presenta un marc d'intercanvi de contingut multimèdia legítim, respectant els drets d'autor i distribució, i adequat a les capacitats dels usuaris en cada moment. El seu propòsit principal es establir, d'una manera clara, quins són els participants en una transacció dins d'un mercat digital, en el que els béns no són més que dades.

Per a fer les proves de l'aplicació s'utilitzarà un tipus de contracte en concret. Com s'ha dit anteriorment, l'aplicació no està pensada ni dissenyada pensant en aquest contracte en concret, sinó per a un ús general; però sí que, per exemple, l'apartat de visualitzar el contracte en PDF depèn del tipus que s'utilitzi. Així doncs, en contracte utilitzat serà un contracte de drets d'autor MPEG-21 REL i tindrà en següent esquema:

```
<?xml version="1.0" encoding="UTF-8" ?>
<license xmlns="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-MX-NS rel-mx.xsd">

 <!-- The license is issued and signed by Alice, the author of the digital item.  -->
   <grant>
      <forAll varName="principal0">
         <propertyPossessor>
            <sx:commonName>IEEE</sx:commonName>
         </propertyPossessor>
      </forAll>
      <principal varRef="principal0" />
       <right xsi:type="mx:Print" />
      <mx:diReference>
         <mx:identifier>urn:paper:a1-abcde-1234567890-f</mx:identifier>
      </mx:diReference>
   </grant>
   <!-- The license is issued and signed by Alice, the author of the digital item.  -->
   <issuer>
      <details xsi:type="IssuerDetails">
         <timeOfIssue>2003-12-31T12:59:59</timeOfIssue>
      </details>
   </issuer>
   <otherInfo xsi:type="">
      <paperinfo xmlns="urn:paperdesc:2009">
         <papertitle>The Title of the Paper</papertitle>
         <journalName>20th International Workshop on Database and Expert Systems Application </journalName>
         <signingauthor>Author A</signingauthor>
         <authors>
            <author>Author A</author>
            <author>Author B</author>
         </authors>
```

```
    </paperinfo>
  </otherInfo>
</license>
```

En aquest exemple es pot identificar clarament l'element "grant" on es concedeix el dret. A dins d'aquest hi ha l'element "Right" que especifica quin dret es dóna (en aquest cas imprimir, "print"). Per altra banda hi ha la informació sobre l'emissor del certificat, dins l'element "issuer". Finalment, trobem l'element "otherInfo"; on s'ha creat una estructura que permetrà aconseguir la informació necessària per tal de generar el document PDF, com poden ser els autors, el títol de l'article…

7.2.5. XSL:FO i FOP

En l'objectiu de poder transformar els contractes amb els que treballem en format XML a un format PDF hi intervenen diverses tecnologies, totes elles importants per a fer la transformació; la qual consta de dues parts:

- Transformar el document XML a XSL:FO.
- Transformar el document XSL:FO a PDF mitjançant el processador FOP.

7.2.5.1. XML a XML:FO

Per a la transformació del document XML a XSL:FO s'ha estudiat la possibilitat d'utilitzar la tecnologia XSL, una tecnologia fortament relacionada amb el format de documents XML que defineix un conjunt de regles que s'apliquen als elements dels documents XML, per cadascun d'ells descriu una sortida. Finalment, però, s'ha optat per fer-ho a través de Java, ja que tenia una millor coneixença d'aquest i degut a que es tractava de mostrar una de les possibilitats que oferia la tecnologia XML. Per a fer-ho, Java té la llibreria javax.xml que en concret té la llibreria xpath que conté un conjunt d'eines per a extreure elements del document XML. D'aquesta manera es pot extreure del document XML la informació desitjada i a continuació generar un document XSL:FO incloent-hi aquesta informació.

7.2.5.2. XSL:FO

XSL:FO és una extensió de XSL, que inclou característiques per a definir la representació gràfica dels elements, FO de "Formatting objects". Així, va néixer amb la intenció de permetre aplicar un conjunt de regles a un document XML i obtenir un document XSL:FO on es defineix com ha de ser la representació d'aquest, de tal forma que aquest document obtingut, pugui ser processat obtenint un format concret (com per exemple PDF en el cas d'aquest projecte). La Figura 1mostra aquest procés.

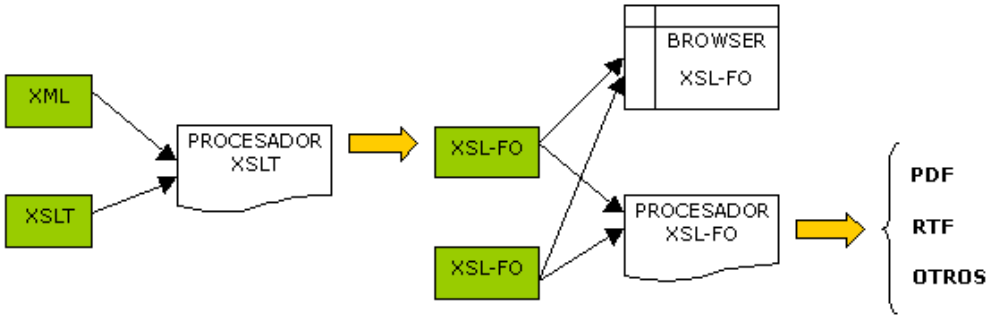


Figura 1. Esquema del procés

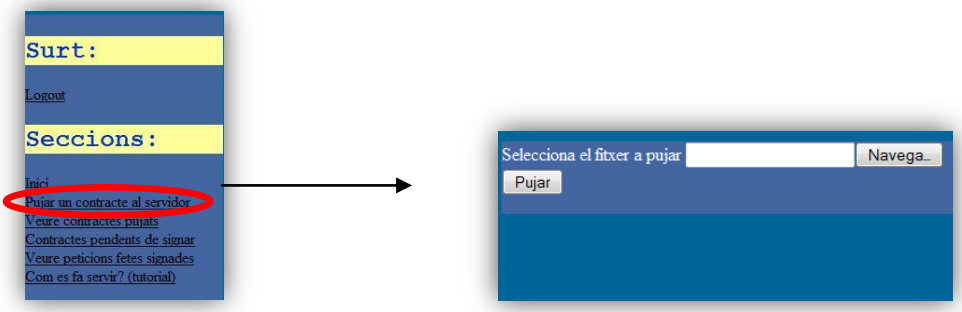
7.2.5.3. Apache FOP

Fop és una aplicació Java que llegeix un document en el format estàndard XSL:FO i el transforma en un format de sortida, en especial en format PDF. FOP és una aplicació que forma part del projecte Apache XML Graphics, és de codi obert i distribuït sota la llicència Apache Software. Així, des d'un codi Java, amb la llibreria org.apache.fop, podem generar un fitxer PDF fàcilment.

7.3. Cas d'ús exemple

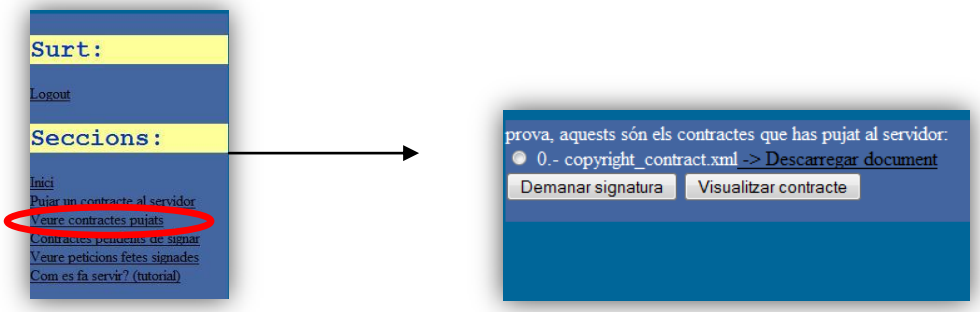
En aquest apartat s'explicarà detalladament un exemple d'ús de l'aplicació desenvolupada, explicant detalladament què és el què ocorre en cada moment.

Primerament, seguirem el procés amb el què l'usuari A demana una signatura d'un contracte a l'usuari B. Per a fer-ho un cop l'usuari A s'ha loguejat al sistema, ha d'anar a la secció "Pujar contracte al servidor". Aleshores el sistema li mostra la pàgina on pot seleccionar el fitxer:

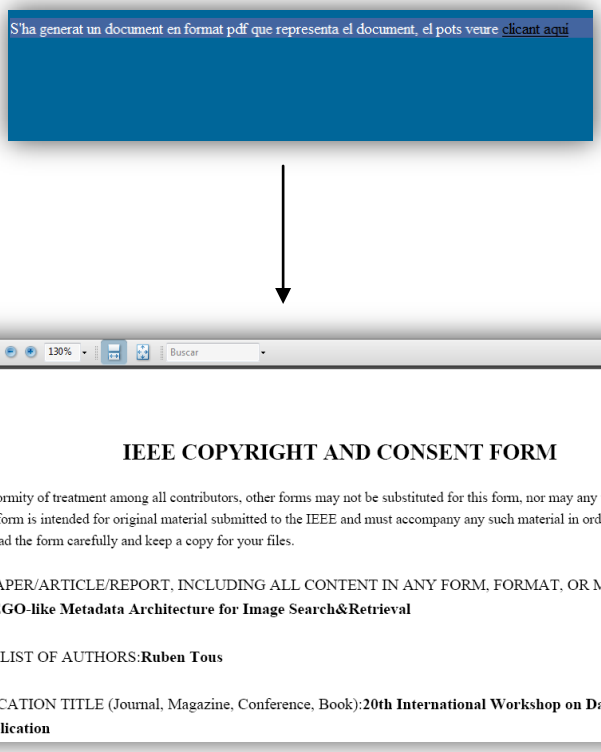


Aleshores l'usuari selecciona el contracte XML a pujar i clica el botó "Pujar". Tot seguit el sistema llegeix el fitxer de l'ordinador de l'usuari, comprova que la extensió sigui XML i el guarda en una carpeta al servidor anomenada contractes, introduint a la base de dades aquesta ruta, l'identificador de l'usuari que l'ha pujat…

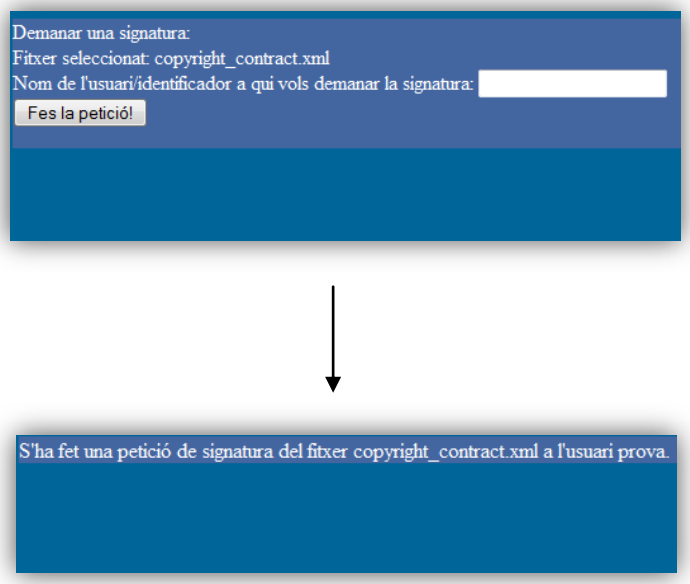
A continuació, l'usuari pot demanar la signatura d'aquest contracte. Clicant a la secció "Veure contractes pujats" pot veure una llista de tots els contractes que ha pujat al servidor (el sistema fa una consulta a la base de dades de tots els contractes que tenen de propietari l'usuari A: "select * from contracte WHERE idPropietari = 'idUsuari';", on idUsuari és l'identificador de l'usuari A), on pot seleccionar-ne un per a demanar-ne una signatura o per a visualitzar-lo.



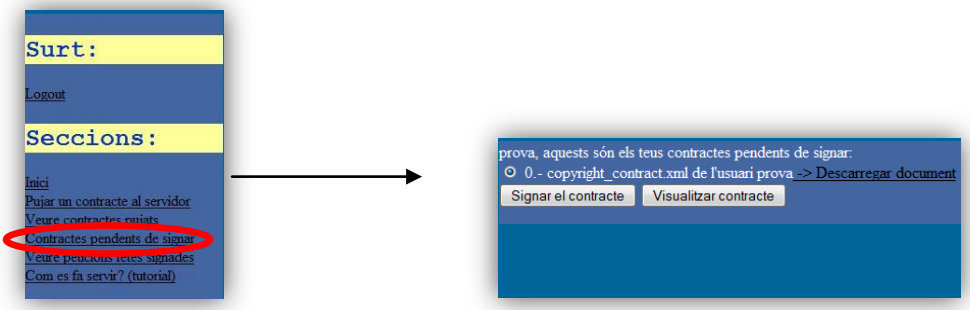
Primerament, vegem què passa si l'usuari selecciona "Visualitzar el contracte". En clicar el botó a l'usuari se li obre una nova pàgina on el sistema genera el document PDF. En el sistema hi ha una classe Java que té un mètode que s'encarrega de transformar un document XML a XSL:FO, i un altre que utilitzant FOP transforma el document XSL:FO a PDF. Així tenim una carpeta al servidor que s'anomena PDF on guardem aquests dos documents. Un cop fet això, s'informa a l'usuari i se li dóna una enllaç al fitxer PDF.



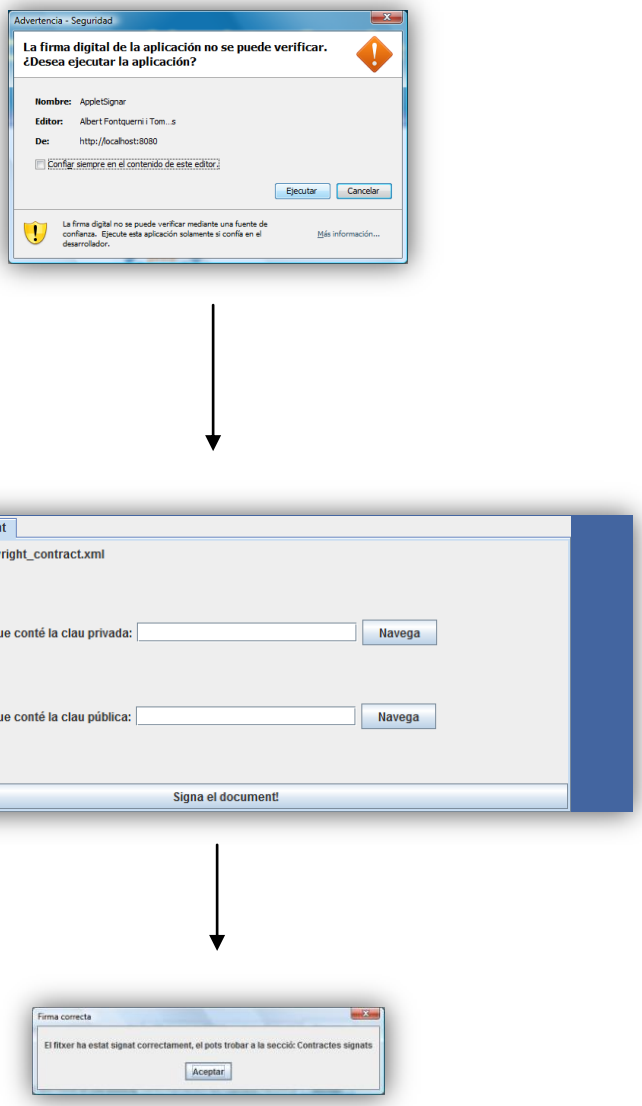
Per altra banda, vegem que ocorre si l'usuari selecciona un dels contractes que ha pujat i escull el botó "Demanar signatura". Un cop l'usuari ha seleccionar el fitxer i ha clicat el botó corresponent, se li mostra una pàgina on ha d'omplir un camp amb l'identificador de l'usuari a qui vol demanar la signatura. Aleshores l'usuari introduiria l'identificador de l'usuari B, si l'identificador introduït és correcte se li mostrarà un missatge de confirmació. Abans de fer-ho, però, el sistema introdueix a la base de dades una nova fila a la taula signatura, amb els camps data i documentSignat amb valor nul, la qual cosa vol dir que la signatura no ha estat realitzada: "insert into signatura VALUES (idPropietari, idContracte, idSignant, "null", "null");", on idPropietari és l'identificador de l'usuari A, idContracte és l'identificador del contracte seleccionat per l'usuari i idSignant és l'identificador que ha introduït l'usuari A.



Tot seguit, l'usuari B ja pot realitzar la signatura, clicant a la secció "Contractes pendents de signar"; aleshores el sistema fa una consulta a la base de dades de totes les signatures que tenen a l'usuari B com a signant i que no han estat realitzades (les columnes data i documentSignat tenen valor nul); i mostra a l'usuari una llista dels documents que compleixen la condició.



A continuació l'usuari pot seleccionar el contracte que vulgui signar i clicar el botó "Signar el contracte". Aleshores el sistema li mostrarà la pàgina que conté l'applet; el qual, com que ha de tenir accés al disc i per tant està signat, sol·licitarà permís per executar-se a l'usuari.

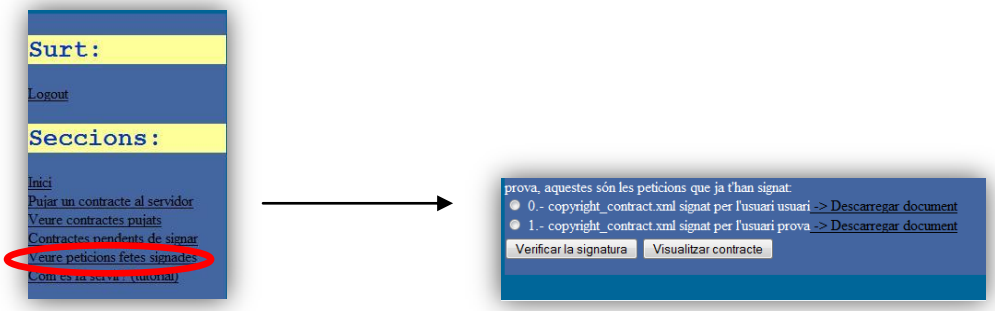


A l'applet, l'usuari selecciona la seva clau privada i la clau pública i clica el botó "Signa el document". Aleshores el sistema genera el document XML signat en el mateix applet (les claus en cap moment surten de l'ordinador de l'usuari), i el guarda al servidor, en una carpeta amb tots els documents firmats. A més, introdueix a la base de dades la data i la ruta al

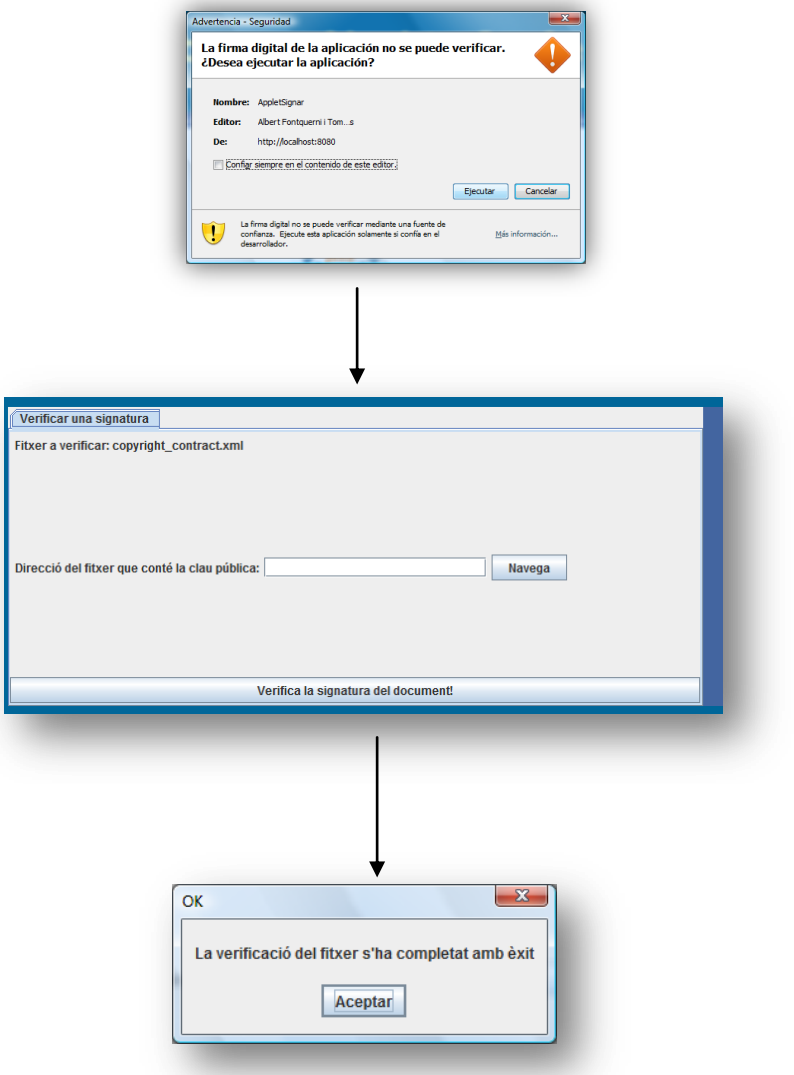
document signat, finalment mostra a l'usuari un missatge de confirmació obrint, a través de l'applet una nova pàgina del navegador:

```
AppletContext a = getAppletContext();
URL url = new URL(urlString);
a.showDocument(url);
```

Finalment l'usuari A, podrà verificar la firma un cop aquesta hagi estat realitzada. Per a fer-ho, haurà d'anar a la secció "Veure peticions fetes signades", on el sistema li mostrarà una llista amb tots els contractes que compleixen aquesta condició. Per a fer-ho, el sistema fa una consulta a la base de dades de totes les signatures de les quals és el propietari del contracte i que hagin estat firmades, és a dir, que els camps data i documentSignat no siguin nuls.



Des d'aquí, l'usuari pot descarregar-se el contracte, visualitzar-lo (segueix el mateix procés explicat anteriorment) i verificar la signatura. Si escull aquesta última opció, el sistema li mostrarà l'applet per a fer-ho. De nou, l'applet ha de tenir accés al disc de l'usuari, per la qual cosa està signat i se li demanarà permís per a executar-lo.



Aleshores l'usuari A selecciona la clau privada de l'usuari que ha realitzat la signatura (de l'usuari B en aquest cas) i la introdueix. A continuació clica al botó "Verifica la signatura del document" i el sistema li mostrarà un missatge de confirmació, si la verificació ha estat correcta. En aquest pas l'applet llegeix la clau publica introduïda i ell mateix s'encarrega de fer la verificació, en cap cas la clau viatja per la xarxa cap al servidor.

8. Planificació i costos

Aquí s'explicarà la planificació fet a l'inici del treball i explicarem com s'ha seguit i si hi ha hagut retards. Tot seguit es farà un anàlisi dels costos reals que hauria tingut el desenvolupament del projecte.

8.1. Planificació

La planificació es va dur a terme en tres fases clarament diferenciades. La primera era una fase d'exploració de les eines que es podien utilitzar i una especificació del sistema. A continuació, es procediria a la implementació del sistema i finalment a la escriptura de la memòria. Cadascuna d'aquestes fases es va dividir en tasques més concretes, de tal forma que es podien avaluar millor.

	Tasques	Hores previstes	Hores reals
Exploració i especificació	Especificació	60	60
	Desenvolupament aplicació per signar	75	75
	Aplicació amb XML Signature	50	60
Implementació	Applet amb XML Signature	50	60
	Configurar Base de Dades	40	40
	Lloc web	80	120
	Transformació XML a PDF	50	50
Memòria		75	100
Total		480	565

Com es veu en el quadre anterior, hi ha hagut algunes tasques que s'han retardat. En especial el desenvolupament del lloc web i l'escriptura de la memòria han costat més temps de l'esperat inicialment. El motiu del primer és en part el desconeixement de la tecnologia JSP, la qual cosa ha comportat la necessitat d'unes quantes hores d'exploració.

8.2. Cost

El cost d'un projecte es divideix en costos de hardware, software i recursos humans.

8.2.1. Hardware

Donat que el projecte consistia en el desenvolupament d'un software, els costos de hardware que hi hauria en cas de voler-lo utilitzar-lo no es tenen en compte.

- Cost en el desenvolupament del projecte: 0€

8.2.2. Programari / software

Pel què fa al software, totes les eines utilitzades (Apache Tomcat, XML Signature, …) són programari lliure, per la qual cosa el cost en software és de 0 euros. En tot cas, l'únic que podria reportar un cost, seria el gestor de bases de dades MySql, que té versions de pagament segons les necessitats que tinguem, però de nou com que l'objectiu era el desenvolupament de l'aplicació i no la seva aplicació en un servidor real, hem utilitzat la versió gratuïta.

- Cost en el desenvolupament del projecte: 0€

8.2.3. Recursos humans

Pel què fa al cost de recursos humans, suposarem que tenim un treballador desenvolupant el projecte i un cap de projectes (que serien similars al paper de l'alumne i del tutor o director respectivament). A continuació s'expliquen les seves funcions:

- **Desenvolupador:** És l'encarregat de fer primer l'especificació i disseny de l'aplicació i la seva posterior implementació. També s'ha d'encarregar de fer les suficients proves per a comprovar el seu correcte funcionament.

- **Cap de projectes:** La seva funció consisteix en guiar el projecte. Marca els objectius i fa un seguiment del desenvolupament de l'aplicació, gestionant així el temps i els recursos. Les seves hores dedicades es corresponen bàsicament a reunions periòdiques amb el desenvolupador i en el temps dedicat a fixar els nous objectius.

Categoria laboral	Hores dedicades	Sou (€/hora)	Cost
Desenvolupador	565	€ 40,00	22.600,00 €
Cap de projectes	17	€ 75,00	1.275,00 €
Total	582		23.875,00 €

- Cost en el desenvolupament del projecte: 23.875,00 €.

8.2.4. Altres

En cas que el desenvolupament del projecte s'hagués dut a terme amb una empresa, també caldria anotar als costs algunes despeses en material, com ara fulls, algun bolígraf,… i altres coses com aigua.

- Cost en el desenvolupament del projecte: 30,00 €.

8.2.5. Cost total

Així doncs, el cost total del projecte seria de: 23.875,00 + 30,00 = 23.903,00€

9. Conclusions i línies de futur

En aquesta part, en primer lloc revisarem els objectius inicials del projecte, per veure fins a quin punt s'han complert. En segon lloc, veurem possibles extensions del projecte, com es podria millorar i/o continuar. Finalment, hi haurà una part amb la experiència personal i la opinió de l'autor.

9.1. Conclusions

L'objectiu principal del projecte era desenvolupar un aplicació web per a signar i verificar documents, aquest objectiu es dividia però, en els següents:

- **Signatura:** Complert. L'aplicació desenvolupada permet la signatura digital de documents, amb un applet tal i com era l'objectiu inicial.
- **Validació de la signatura:** Complert. L'aplicació conté un altre applet que permet verificar les signatures realitzades.
- **Entorn web:** Complert. Els dos applets s'han integrat perfectament a un lloc web, on un usuari pot interactuar amb la resta fent peticions de signatures i veien les signatures realitzades.
- **Mostrar un document:** Complert. En aquest cas s'ha canviat una mica el procés, ja que la idea iniciat era utilitzar XSLT, finalment s'ha fet a través de codi Java, però l'objectiu de visualitzar un document XML en PDF s'ha complert.
- **Gestió d'esquemes XML (*objectiu opcional):** Incomplert. Aquest objectiu opcional no s'ha pogut desenvolupar degut a la manca de temps.

Així doncs, podem afirmar que els objectius del projecte s'han complert, el funcionament de tots els objectius obligatoris és el correcte. L'únic que ha quedat per fer ha sigut un objectiu opcional, que havíem posat per si ens donava temps, però teníem molt clar que no era el prioritari.

9.2. Línies de futur

Com s'ha dit en el punt anterior, els objectius principals ha estat complerts, però lògicament això no vol dir que l'aplicació estigui perfectament acabada. Al principi es van posar uns límits i s'han respectat, però això no vol dir que no es pugui continuar, alguns dels elements principals que es podrien afegir serien:

- **Gestió d'esquemes XML:** Era ja una opció que s'havia tingut en compte, consistiria en tractar amb una sèrie catàlegs de XML schemas per tal que l'usuari pugui verificar un document abans de signar-lo.

- **Generar els documents XML:** Amb el mateix catàleg de XML Schemas, es podria afegir al lloc web una pàgina que sol·licités a l'usuari els camps necessaris per a omplir el document XML i generar-lo.
- **Comprovació dels certificats:** El lloc web desenvolupat, només sol·licita a l'usuari el certificat, però no en comprova la validesa ni la data ni el fet que no estigui revocat. Si es volgués utilitzar l'aplicació seria un punt imprescindible.
- **Ús de fitxers de clau oficials:** Les claus utilitzades per a fer i verificar les signatures són unes claus generades amb Java, per la qual cosa no utilitza per exemple un fitxer amb extensió .p12. Es podria millorar l'applet per tal que demanés la contrasenya necessària per accedir a aquest tipus de fitxer i en llegís la clau desitjada en cada cas.

9.3. Experiència personal

En aquest apartat s'exposa una reflexió personal de l'autor en referència al desenvolupament d'aquest projecte.

Un cop acabat aquest projecte, puc afirmar que ha estat una experiència bastant positiva. Per primera vegada en la carrera d'enginyeria t'enfrontes a un projecte que has de desenvolupar com tu creguis millor. Si bé hi ha força conceptes que s'han tractat durant la carrera que et són útils per a prendre decisions, per exemple he intentat aplicar, tant el l'especificació com en el disseny, els conceptes que he après en les assignatures de enginyeria del software; en aquestes, però has de prendre decisions com ara quines plataformes i/o llibreries usar.

Per altra banda, un dels factors que he trobat particularment interessant ha sigut la quantitat d'eines noves per a mi amb les que he treballat. Aquest és un fet que no queda reflectit en el projecte, però que penso que és important, ja que potser en un futur puc necessitar alguna eina similar i ja sabré de què tracta. De fet, crec que el projecte final de carrera, ha de ser una oportunitat més per aprendre coses noves i actuals.

Finalment, també he pogut veure com fer un seguiment del projecte, he anat veient com algunes de les dates fixades en la planificació inicial s'endarrerien i el què en la planificació inicial havien de ser uns dies de marge han acabat sent dies de treball intens. Cal destacar també, la importància de posar uns límits al projecte, ja que, com passa en moltes aplicacions desenvolupades, sempre se t'acudeixen possibles millores.

Apart de l'experiència personal en referència a la realització del projecte, crec que l'aplicació desenvolupada és una eina interessant. Crec que una eina com ara aquesta serà utilitzada en un futur per a signar documents, segurament tindrà elements diferents, ja que probablement hi haurà eines noves o dissenys millors, però l'objectiu final serà el mateix. Així doncs, crec que aquest projecte ha sigut una experiència molt interessant a més a més de que m'ha permès aprendre molt.

9.3.1. Agraïments

Voldria agrair a totes aquelles persones que m'han ajudat en el desenvolupament del projecte; però en especial al Rubén Tous, el tutor del projecte que sempre s'ha mostrat implicat en el projecte i disposat fer les reunions necessàries per al seu correcte desenvolupament.

10.Bibliografia

Bibliografia en paper

- Dolors Costa, M. Ribera Sancho, Ernest Teniente , *"Enginyeria del Software: Especificació",* Edicions UPC, 2000.
- Carles Farré, Antoni Olivé, Carme Quer "Enginyeria *del Software: Disseny II*", Edicions UPC, 2003.

Bibliografia digital

General

- http://es.wikipedia.org/wiki/Wiki
  (en els diferents idiomes)

Applets

- http://laurel.datsi.fi.upm.es/~ssoo/DAW/web03-04/presentaciones/07JavaApplets.pdf
- http://java.sun.com/docs/books/tutorial/deployment/applet/
- http://java.sun.com/docs/books/tutorial/ui/features/components.html
- http://www.proactiva-calidad.com/java/jdbc/applet_swing_jdbc.htm
- http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=app_keyt_jars

Claus

- http://blogs.warwick.ac.uk/kieranshaw/entry/creating_a_java/
- http://portecle.sourceforge.net/create-keystore.html

Signatura Digital

- http://java.sun.com/j2se/1.5.0/docs/guide/security/p11guide.html#Keys
- http://www2.krellinst.org/UCES/archive/modules/charlie/pke/
- http://www-ma2.upc.es/~cripto/

XML Signature

- http://java.sun.com/javase/6/docs/technotes/guides/security/xmldsig/XMLDigitalSignature.html
- http://java.sun.com/developer/technicalArticles/xml/dig_signature_api/
- http://www.w3.org/TR/xmldsig-core/#sec-CoreValidation
- http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=xmlSignature

XSL:FO i FOP

- http://xmlgraphics.apache.org/fop/
- http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7-7911-470b-8b6b-71ba72783fdd&ID=50

JSP

- http://www.programacionfacil.com/java_jsp/start
- http://geneura.ugr.es/~jmerelo/JSP/
- http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=287

Tomcat

- http://casidiablo.net/instalacion-de-un-entorno-web-tomcat-jsp-mysql/

REL

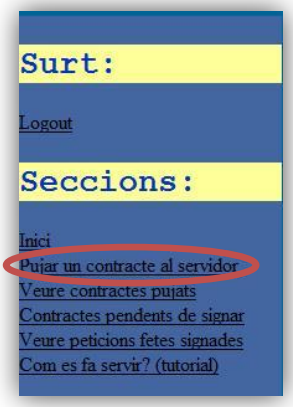
- http://rhizomik.net/~roberto/thesis/html/RightsExpressionLanguages.html
- http://odrl.net/workshop2004/prez/odrl-polo-prez.ppt

11.Annex

11.1. Manual d'usuari

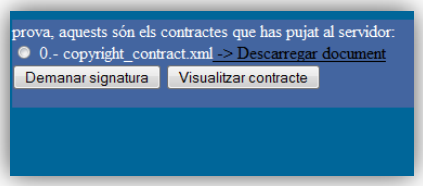
- **Pujar un contracte al servidor**

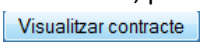
Per pujar un contracte al servidor, cal clicar a la secció "Pujar contracte al servidor":



Aleshores ens demana que seleccionem un fitxer. Ho fem, cliquem al botó "Pujar fitxer" i ja tenim el fitxer pujat al servidor.

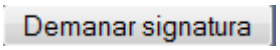
Ho podem comprovar clicant a la secció "Veure contractes pujats"

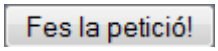


Des d'aquí podem descarregar el fitxer de nou, clicant a l'enllaç "Descarregar document", podem demanar-ne una signatura i podem visualitzar-lo en format PDF clicant a 

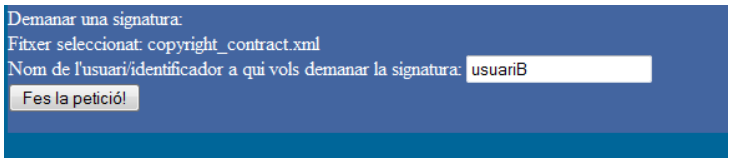
- **Demanar una signatura**

Per demanar una signatura, primer cal que haguem pujat el contracte corresponent al servidor (veure punt 1). Aleshores cliquem a la secció veure contractes pujats:

Un cop allà seleccionem el que vulguem que ens firmin i cliquem al botó 

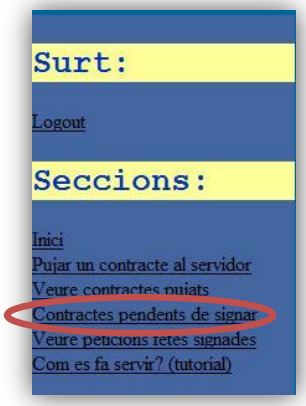
. A continuació ens demanarà l'identificador (o nom d'usuari) de l'usuari a qui volem demanar la signatura; l'introduïm i cliquem al botó 

.

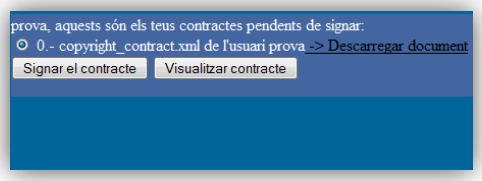


- **Signar un document**

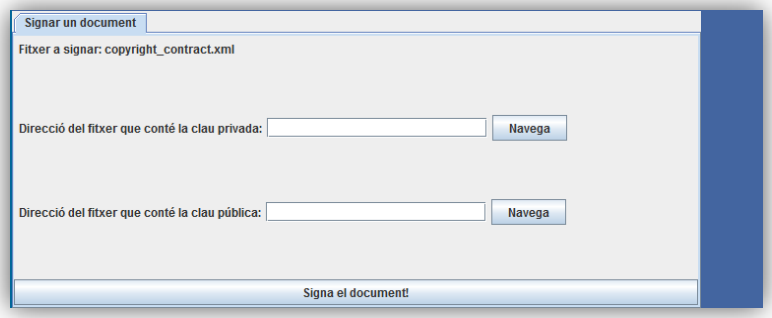
Per a signar un document, cal que primerament s'hagi rebut una petició de signatura d'aquest. Si ja tenim una petició, cliquem a la secció "Contractes pendents de signar".



Aquí veurem tots els contractes pels quals se'ns ha fet una petició de signatura i que encara no hem signat. A més, podem veure l'usuari que ens ha fet la petició de signatura de cada document.



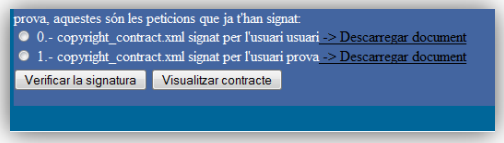
Tot seguit escollim el contracte que vulguem signar i cliquem al botó Signar el contracte. Aleshores se'ns mostrarà l'applet des del qual podrem signar el document introduint les nostres claus privada i pública. Veurem que abans d'executar-se ens demanarà permís, cal que li'n donem ja que necessita tenir accés a disc.

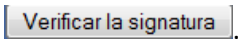


- **Verificar una signatura**

Per a verificar una signatura, és necessari que haguem demanat una signatura del fitxer corresponent a un usuari. Un cop fet això, si l'usuari ha signat el document, podrem veure'l clicant a la secció "Veure peticions fetes signades".

Aquí, se'ns mostrarà una llista amb tots els documents que compleixen aquestes condicions.



Tot seguit, escollim el document del qual vulguem verificar la signatura i cliquem el botó 

. Allà, amb la clau pública de l'usuari corresponent podrem verificar si l'ha signat ell amb la seva clau privada (corresponent a la clau pública introduïda).