Títol: eAdministració amb signatura electrònica

Volum: 1/1
Alumne: David Gómez Guillén

Director/Ponent: Fernando Martínez Sáez
Departament: Matemàtica Aplicada II
Data: Agost 2010

DADES DEL PROJECTE

Títol del Projecte:	eAdministració amb signatura electrònica
Nom de l'estudiant:	David Gómez Guillén
Titulació:	Enginyeria en Informàtica
Crèdits:	37.5
Director/Ponent:	Fernando Martínez Sáez
Departament:	Matemàtica Aplicada II

MEMBRES DEL TRIBUNAL *(nom i signatura)*

Presidenta:	Ana Río Doval
Vocal:	Marta Fairén González
Secretari:	Fernando Martínez Sáez

QUALIFICACIÓ

Qualificació numèrica:
Qualificació descriptiva:

Data:

Índex

Capítol 1

Introducció

1.1 Motivació del projecte

És evident que els processos administratius de qualsevol organisme són necessaris pel seu bon funcionament. Aquests processos no són un fenòmen nou; durant tota l'existència humana com a civilització ha estat necessari regular els recursos per assegurar un repartiment just entre tot un grup. La seva finalitat és diversa: deixar constància d'un acord, consultar dades anteriors arxivades, elaborar resums estadístics, ...

El gran incovenient, però, és que aquest procés és tediós en molts casos. Tot i que la necessitat d'una burocràcia és acceptada per tothom, resulta igualment frustrant pel sol·licitant d'un recurs haver de presentar-se a una oficina, a una hora indicada, omplir papers, signar i finalment haver d'emportar-se una còpia de la sol·licitud. Tot això exigeix una inversió de temps, espai i paciència que, junt amb la complexitat del procés administratiu dels grans organismes, provoca una reacció desfavorable cap als sistemes burocràtics com, per exemple, el govern.

És en aquest àmbit on les tecnologies de la informació poden ajudar. Com tots sabem, ja fa molts anys que molts processos administratius es poden fer via web, estalviant temps i inconveniència al sol·licitant. Però fins i tot aquests tenen un desavantatge: no està ben estandaritzada la seva validesa legal, tema imprescindible a tenir en compte en un sistema burocràtic. És per això que tots els documents que necessiten una garantia legal necessiten la signatura del sol·licitant la qual cosa comportava, fins ara, realitzar la sol·licitud de forma presencial.

Per evitar aquest problema, en els últims anys s'han instaurat directives a nivell tant europeu (desembre de 1999) com espanyol (desembre de 2003) per definir legalment el concepte de *signatura electrònica*. Aquesta permet, de forma telemàtica, signar documents amb la mateixa vinculació legal que les signatures manuscrites i amb algunes garanties tècniques addicionals, com la integritat del document, l'autenticació unívoca del signant i el no repudi de la signatura. Aquesta novetat legal ve complementada, convenientment,

per dos fets recents que ens permetran implantar la signatura digital.

Per una banda tenim la distribució dels carnets UPC i del DNI electrònic. Amb aquestes targetes podrem realitzar signatures de forma relativament senzilla, només necessitarem un lector de targetes adequat.

Per altra banda, de forma encara més recent (2008) s'han definit també els diversos estàndards de signatura electrònica per documents PDF (*PDF Advanced Electronic Signature, PAdES*). Amb això podrem afegir les signatures als propis documents PDF de manera que sempre vagin junts, com una firma d'un document manuscrit.

Amb totes aquestes eines, dissenyarem un sistema d'eAdministració pel departament de Matemàtica Aplicada II que permetrà als usuaris realitzar sol·licituds via web amb el benefici de la signatura electrònica. Espero que aquesta eina permeti agilitzar i simplificar la burocràcia del departament, en benefici tant de la gent de l'administració com dels usuaris sol·licitants.

1.2 Metodologia

Per realitzar aquest projecte utilitzarem la filosofia de desenvolupament àgil (*Agile Software Development*) [7]. A diferència d'altres mètodes, com el desenvolupament en cascada, la metodologia àgil ens permet treballar de forma iterativa (diversos cicles de desenvolupament) i incremental (a cada iteració anem modificant el resultat de la iteració anterior). Aquesta forma de treballar té lògica en l'àmbit del desenvolupament de software, ja que els requisits no solen estar ben definits i, encara que fos així, aquests poden canviar molt sovint. Les iteracions ens permeten anar treient versions funcionals del producte mentre continuem depurant i incorporant els nous requisits que resultin del *feedback* del client al procés.

Un altre avantatge d'aquesta iterativitat és que el client veu el producte de forma més freqüent i, per tant, els comentaris de millora o rectificació són més abundants. D'aquesta manera tenim més marge de maniobra per evitar problemes en etapes inicials que repercuteixin en etapes posteriors. Cal esmentar que el "client", en aquest cas, és principalment el cap d'administració del departament (Dídac Guardia), amb el responsable informàtic (Marc Andreu) i el tutor del projecte (Fernando Martínez) en segon lloc.

En aquest projecte farem tres iteracions, cadascuna amb dos mesos previstos de duració per un total d'un projecte de sis mesos. Aquestes iteracions, de fet, estaran composades per "subiteracions" amb uns objectius comuns que definirem a l'especificació, quan estiguem preparats per prendre una decisió informada. A cada iteració incorporarem les modificacions necessàries (extretes dels comentaris del client) de la iteració anterior.

En quant a documentació del projecte, la metodologia àgil és partidaria d'evitar tots els artefactes (diagrames i esquemes del projecte d'enginyeria de software) que no siguin útils durant el procés de realització del projecte. Molts diagrames són trivials i, mentre la idea del que s'ha de fer estigui perfectament clara, dibuixar-los és una pèrdua de

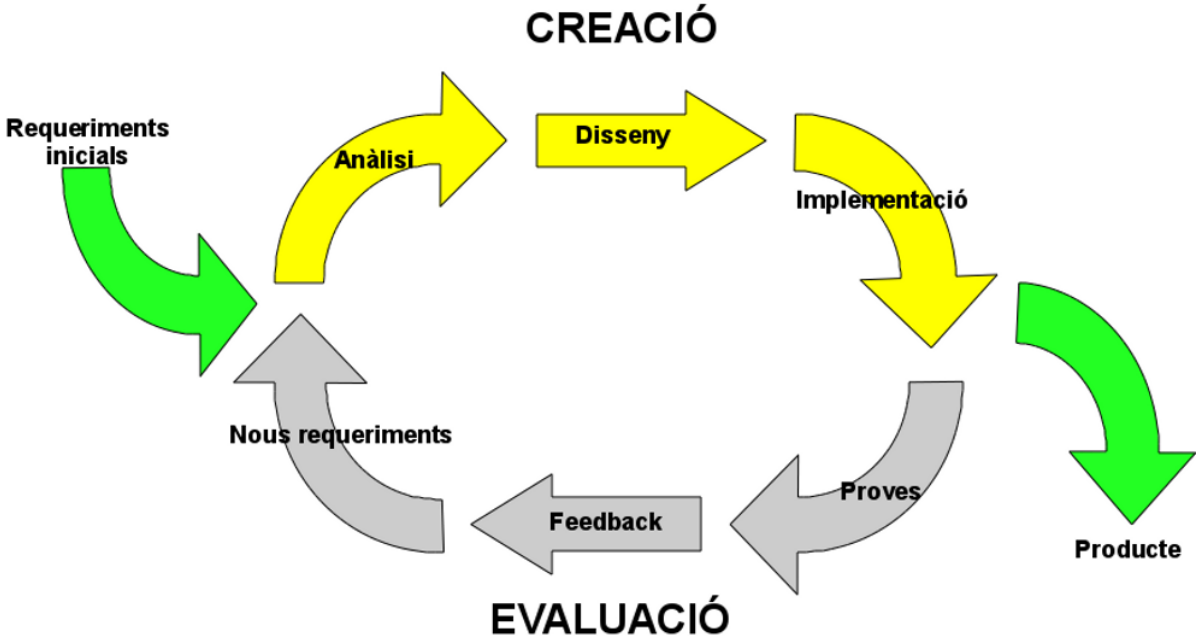

Figura 1.1: Esquema de treball de la metodologia àgil.

temps, sobretot quan aquests són molt numerosos. Així, en general, només mostrarem els diagrames que realment aportin informació al procés.

1.3 Organització de la memòria

El present document intentarà seguir aproximadament el procés de desenvolupament del software de forma cronològica, tenint en compte que les etapes de les diverses iteracions estaran agrupades dins del mateix apartat.

El document ha començat amb la motivació darrera del projecte, on s'explica la necessitat d'aquest sistema en context de la situació actual. També hem comentat la metodologia utilitzada pel desenvolupament, a més d'altra informació organitzativa relativa al projecte.

A continuació el **capítol 2** detalla l'etapa d'*especificació del sistema*. Això inclou primerament l'*anàlisi de requeriments* on elaborarem una llista de requeriments, funcionals i no funcionals, a partir de l'entrevista amb el client (el cap d'administració del departament). Aquesta llista serà analitzada per trobar possibles incongruències, ambigüetats o contradiccions, i finalment amb aquesta llista s'elaborarà una *especificació*. Aquesta informació ens dona

la possibilitat de realitzar una primera planificació i un anàlisi de riscos.

Seguidament, al **capítol 3** realitzarem un anàlisi estructural del sistema. Partirem de l'especificació anterior, de granularitat molt gruixuda, i trencarem el sistema en components més senzills fins que poguem elaborar un disseny de cadascún d'ells de forma independent. Òbviament, a més dels components també necessitarem especificar com es comunicaran entre ells, amb l'especificació de cadascun.

Abans de seguir amb el procés de disseny, ocuparem el **capítol 4** per parlar dels eixos principals del projecte: les signatures digitals i els documents PDF. Veurem en què consisteixen i quins estàndards hi ha de les primeres així com de quina manera es poden integrar en els fitxers PDF.

Una vegada tinguem especificats tots els components del nostre sistema i tinguem clar els conceptes necessaris per la implementació de la signatura digital, realitzarem el disseny de cada component al **capítol 5**. Això comportarà definir la interfície d'entrada al sistema, a més d'aplicar tecnologies concretes a cada subsistema que hem trobat a l'anàlisi per tal de deixar-ho tot llest per la següent fase.

Amb tota la documentació tècnica elaborada és hora de traslladar-ho a codi mitjançant la implementació. Al **capítol 6** comentarem quins entorns utilitzarem (de prova i de producció), així com les proves preliminars que hem realitzat i les dificultats que ens hem trobat en el procés de codificació. Aquests problemes ens faran reconsiderar algunes decisions de disseny que se'ns podien haver passat per alt, i les tindrem en compte en següents iteracions.

Quan acabem la implementació ja tindrem un producte, al menys parcialment funcional. Per comprovar la seva correctesa de forma més exhaustiva, al **capítol 7** definirem la bateria de proves que realitzarem en cada iteració per tal de detectar problemes. Dividirem les proves en proves d'ús, com les que un usuari realitzaria, i proves avançades, accions més concretes per trobar vulnerabilitats menys evidents.

Amb el projecte finalitzat, ens prendrem el **capítol 8** per intentar veure les possibilitats futures del nostre sistema: quines coses podem ampliar en cas de necessitat i amb quines hem d'anar en compte.

Per concluir la memòria, tancarem amb una conclusió sobre el projecte realitzat i els objectius complerts, així com un glossari de termes significatius al llarg del projecte i diversos annexes amb informació addicional relativa al sistema desenvolupat.

Capítol 2

Especificació

Abans de començar a parlar del nostre projecte de software, primer hem de veure com funciona el procés de sol·licituds en el departament. Una vegada tinguem clar com funciona el context que hem de tractar, elaborarem (amb l'ajuda del cap d'administració i el tutor del projecte) una llista inicial de requeriments i els canvis que s'han anat produint durant les diferents iteracions del projecte.

2.1 Situació actual

Per poder entendre bé els requeriments del nostre sistema és imperatiu conéixer bé com és el domini que estem tractant, el que en enginyeria de software es denomina la *lògica de negoci*.

Per establir una nomenclatura comuna en l'àmbit del projecte, entendrem per *formulari* la informació que s'ha d'omplir, mentre que una *sol·licitud* serà aquesta informació omplerta per un sol·licitant. Per més informació, podem veure el glossari.

El departament de Matemàtica Aplicada II és un departament relativament petit (composat per unes 70-80 persones). Existeixen quatre càrrecs (director, sotsdirector, cap d'administració i secretari acadèmic) que poden tenir la responsabilitat de signar alguns tipus de sol·licituds. Segons el cap d'administració, la majoria de sol·licituds segueixen un d'aquests camins (una vegada el sol·licitant, en cas de què faci falta, signi):

- Va directament a administració.
- Va directament a administració, on el cap d'administració signa.
- El director signa i la sol·licitud s'envia a administració.
- El sotsdirector signa i la sol·licitud s'envia a administració.

Una de les queixes del sistema actual és que en els dos últims casos de la llista el sol·licitant entrega els papers a administració, ells els entreguen al càrrec adient perquè ho signi i finalment retornen, de nou, a l'administració. Això fa que l'administració tingui una càrrega de treball innecessàriament alta amb aquest esquema de funcionament, cosa que volem evitar. A més, hem de tenir en compte que els processos anteriors no són els únics, en poden aparéixer més.

Per ajudar a il·lustrar el funcionament de l'entrega d'una sol·licitud, veiem les figures 2.1 i 2.2. De la primera podem destacar tres coses:

- Les sol·licituds poden tenir diverses pàgines.
- S'han de vigilar els camps omplerts per assegurar valors vàlids.
- De forma més subtil, en el cas de no poder signar la sol·licitud es guarda fins que se signi (no es demana al sol·licitant torna a omplir papers.).

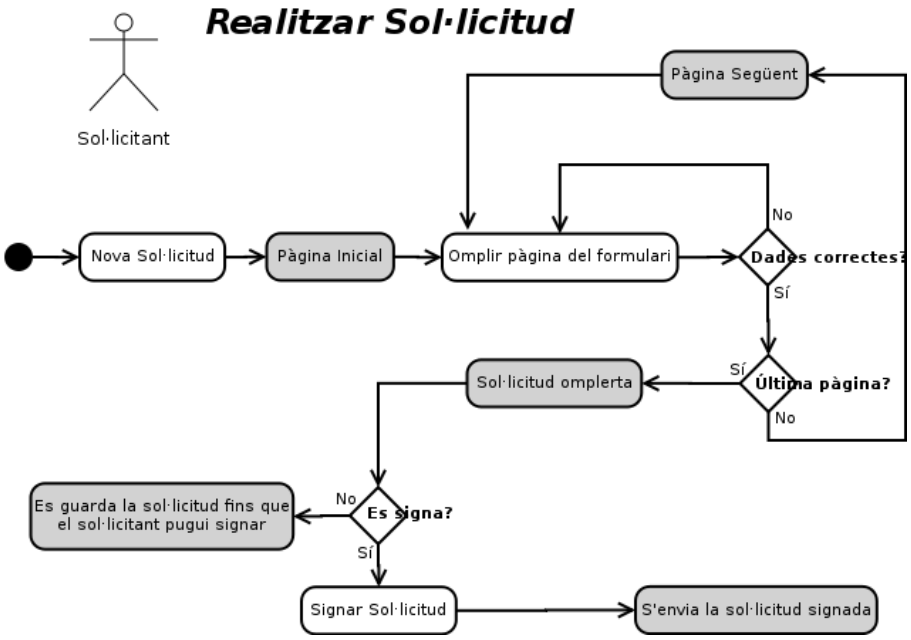

Figura 2.1: Procés del sol·licitant per realitzar una nova sol·licitud.

En la segona figura (2.2), també podem extreure conclusions:

- Una sol·licitud pot ser rebutjada per un càrrec, si no està d'acord amb el seu contingut.
- Com ja hem dit abans, les sol·licituds poden passar per un nombre indeterminat de càrrecs, tots els quals han d'acceptar perquè la sol·licitud sigui finalment acceptada.

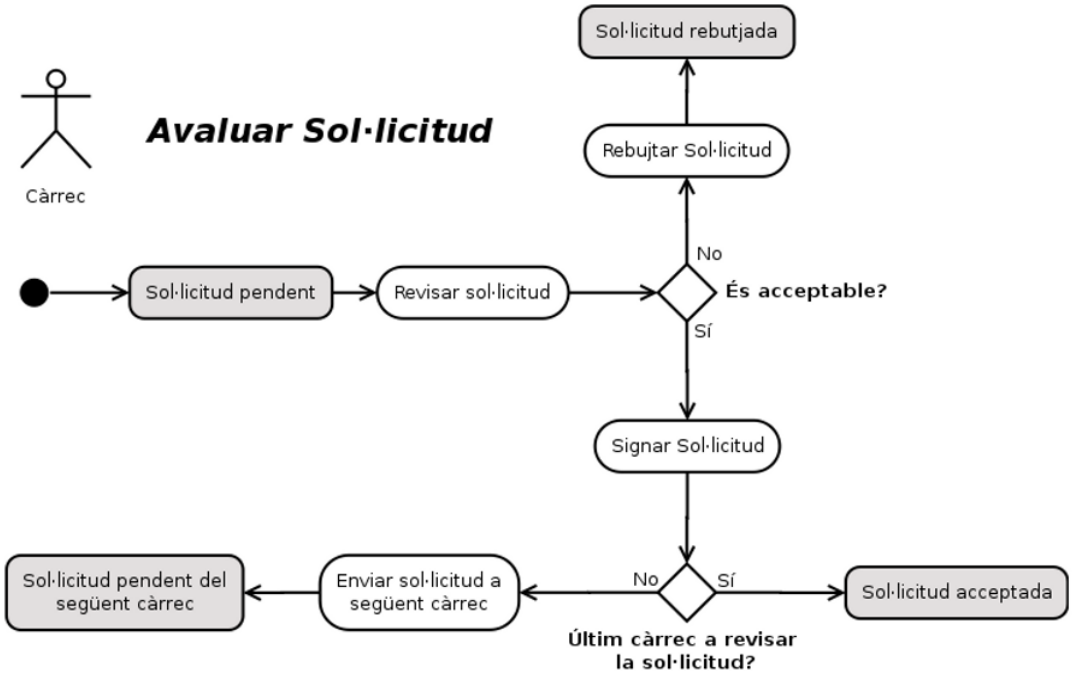

Figura 2.2: Procés d'un càrrec per avaluar una sol·licitud pendent.

Parlant de les sol·licituds, podem distingir dos casos: formularis interns i externs. Els primers són aquells que només són per la organització interna del departament. El disseny i contingut d'aquests formularis són, per tant, creats pel propi departament sense cap restricció especial. Per altra banda, els externs són els que s'utilitzen per tràmits a altres organismes, com la UPC. Aquests ja venen predefinits, així doncs ens hem d'adequar al seu format electrònic, que en principi pot ser qualsevol (freqüentment un document .doc). Aquest document l'anomenarem *document base*.

Finalment, totes les sol·licituds es guarden durant un temps determinat segons el seu tipus; moltes d'elles un any, però pot ser més temps.

En quant als càrrecs, en principi existeixen només els quatre mencionats anteriorment. Aquests càrrecs són ocupats per una única persona, excepte en casos extraordinaris (suplència, co-encarregats, ...), que poden ser compartits.

2.2 Requeriments funcionals

Sabent com funciona el procés administratiu del departament i les seves necessitats, podem elaborar una llista de requeriments inicials:

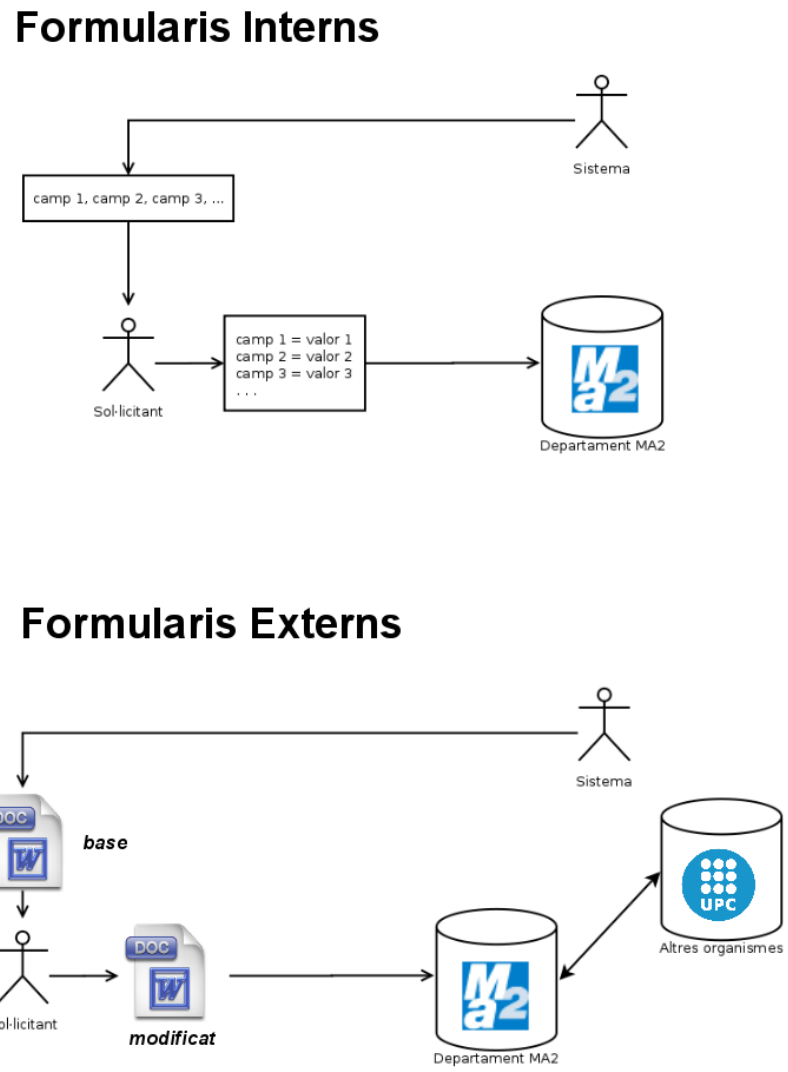

Figura 2.3: Diferències entre formularis interns i externs. Els primers els podem definir com volem i només s'utilitzen dins del mateix departament mentre que els segons ens venen donats (document base) i pot ser necessari el seu enviament a altres organismes (com la UPC).

2.2.1 Primera iteració

Formularis

1. S'han de poder crear, esborrar i editar formularis (interns i externs).

2. S'han de poder editar el contingut dels formularis creats, tant els camps dels formularis interns com el document base dels externs.

3. S'han de poder editar els càrrecs signants de cada formulari.
4. S'han de poder editar un seguit d'opcions pròpies del formulari:
    - Un formulari pot estar obert o tancat.
    - Un formulari pot requerir autenticació o no (s'ha de poder permetre l'accés anònim).
    - En un formulari es pot desactivar la generació de sol·licituds en format PDF.
5. Els camps dels formularis interns estaran classificats en diferents tipus, segons les restriccions que volguem imposar a l'usuari. Poden ser:
    - Text curt
    - Àrea de text
    - Casella
    - Desplegable
    - Multiopció exclusiva
    - Multiopció no exclusiva
6. Els camps dels formularis s'han de poder separar en seccions, per agrupar els camps de forma visual de cara a l'usuari.
7. Tant els formularis com els seus camps podran ser complementats amb una descripció que es mostrarà al sol·licitant.
8. Els camps poden ser obligatoris i/o dependents (només disponibles si el camp de tipus casella anterior està seleccionat).

Sol·licituds

1. Els sol·licitants han de poder realitzar sol·licituds a partir dels formularis (oberts) existents, en un sistema web.
2. En cas de què un usuari realitzi una sol·licitud i no pugui signar immediatament, aquesta s'ha de poder guardar i donar la opció de signar més endavant.
3. Les sol·licituds resultants s'han d'emmagatzemar de forma persistent, tant les pròpies dades com, opcionalment (segons l'anterior punt 4 de la secció de formularis), la seva representació en forma de document PDF.
4. Els sol·licitants han de poder consultar les seves sol·licituds, així com el seu estat i obtenir-ne una còpia en format PDF.
5. En el cas dels formularis externs, les sol·licituds s'han de realitzar de la següent manera:

   (a) Al crear el formulari s'establirà un document base en format .doc
   (b) El sol·licitant es descarregarà el document base, l'omplirà i l'enviarà al sistema.
   (c) El sistema s'encarregarà de tractar el document base omplert pel sol·licitant.

6. Els càrrecs han de poder veure les sol·licituds pendents de signar.

7. Els sol·licitants han de ser notificats sobre l'estat final de les seves sol·licituds, tant si són acceptades com rebutjades.

8. Els càrrecs han de ser notificats quan hi hagi noves sol·licituds pendents de la seva signatura.

Càrrecs

1. S'han de poder definir càrrecs.

2. S'ha de poder editar la gent pertanyent a cada càrrec. Per cobrir els casos extraordinaris permetrem que un càrrec el pugui ocupar més d'una persona pugui ocupar un càrrec.

Accés

1. S'ha de poder accedir al sistema mitjançant el carnet UPC.

2. S'ha de poder accedir al sistema també mitjançant l'usuari i el password de la UPC.

3. S'ha de poder limitar l'accés només a usuaris de la UPC (amb la possibilitat de restringir-ho al departament).

Signatura

1. Tant els sol·licitants com els càrrecs han de poder signar les sol·licituds PDF amb el carnet UPC.

2. La signatura ha d'estar insertada dins del mateix document per facilitar la compartició del document.

2.2.2 Segona iteració

En la segona iteració s'han introduit els següents canvis:

Formularis

1. S'ha de poder obtenir una còpia del formulari (buit) en format electrònic (i.e. PDF o el document d'un formulari extern).

2. S'ha de poder previsualitzar el formulari una vegada creat, sense haver de realitzar una nova sol·licitud.

3. Afegim la opció d'indicar una data de caducitat perquè el formulari es tanqui automàticament.

4. S'ha de poder duplicar l'estructura d'un formulari (copiar el formulari sense les sol·licituds).

5. Els formularis poden tenir diverses pàgines (on cada pàgina pot tenir diverses seccions).

6. S'afegeix el nou tipus de camp "data".

Sol·licituds

1. Els càrrecs també han de poder veure totes les sol·licituds dels formularis als que tinguin accés.

2. Per tal d'augmentar la versatilitat del procés de sol·licituds externes, permetem qualsevol tipus de fitxer com a document base, tot i que el sol·licitant haurà de fer la conversió a document PDF pel seu compte abans de transmetre el fitxer al sistema.

Càrrecs

Sense canvis.

Accés

1. S'ha de poder accedir al sistema també mitjançant el DNI electrònic.

Signatura

1. S'ha de poder signar també amb el DNI electrònic.

2.2.3 Tercera iteració

En la tercera i última iteració s'han introduit els següents canvis:

Formularis

1. S'afegeix el nou tipus de camp "número".
2. Afegim les següents opcions al formulari:
    - Els usuaris poden ser notificats sobre l'estat final de les seves sol·licituds o no.
    - Els càrrecs poden ser notificats sobre noves sol·licituds pendents de signatura o no.

Sol·licituds

1. Després d'un temps després de la data de caducitat d'un formulari, els càrrecs amb sol·licituds pendents de signatura han de ser notificats que encara hi ha sol·licituds pendents, de forma periòdica.

Càrrecs

1. S'ha de poder canviar el nom dels portadors del càrrec (per exemple, per tenir en compte el gènere: Director, Directora.

Accés

Sense canvis.

Signatura

Sense canvis.

2.3 Requeriments no funcionals

Els tres criteris fundamentals en l'elaboració del projecte seran, en ordre de preferència:

1. Seguretat

2. Usabilitat

3. Simplicitat

Òbviament, n'hi ha d'altres d'importants, com la robustesa, però els tres anteriors són especialment crítics per la intenció d'aquest projecte (facilitar els tràmits burocràtics al departament, de forma segura). De forma més concreta, podem mencionar:

2.3.1 General

1. El sistema en el seu conjunt ha de ser funcional en diverses plataformes, principalment Windows XP/Vista/7 i Linux. D'aquest últim ens centrarem en el seu ús en tres distribucions:

   OpenSuSE: Una de les distribucions de Linux més utilitzada a l'entorn UPC.

   Ubuntu: Possiblement la distribució més popular actualment en molts àmbits. Basada en Debian.

   Gentoo: (Meta)-distribució extremadament personalitzable. Si conseguim un bon funcionament en aquest cas pràcticament garantim que pot funcionar en qualsevol distribució Linux.

2.3.2 Formularis

1. S'ha de trobar un equilibri entre versatilitat i simplicitat a l'hora de crear i editar formularis. Volem donar el màxim d'opcions a l'hora de gestionar formularis, però hem d'anar en compte de fer la interfície usable.

2.3.3 Sol·licituds

1. S'ha de procurar fer el procés de realització de sol·licituds el més fàcil i ràpid possible per l'usuari.

2. S'ha de mantenir la confidencialitat de les sol·licituds.

2.3.4 Signatura

1. De nou, el procés de signatura ha de ser el més fàcil possible.

2.4 Actors del sistema

A la vista dels requeriments, podem deduir els següents actors que interactuaran amb el sistema (representats també a la figura 2.4):

Sol·licitant: L'actor principal del sistema, el què realitzarà sol·licituds. Aquest pot estar autenticat o ser anònim.

Càrrec: L'actor encarregat de revisar sol·licituds i signar-les o rebutjar-les. Com hem mencionat anteriorment, hi ha quatre càrrecs: Cap d'administració, director, sotsdirector i secretari acadèmic.

Gestor: L'actor encarregat de definir els formularis i càrrecs en el sistema.

Per altra banda hem de tenir en compte que el sistema no sabrà qui som fins que ens autentiquem, ja que tots els actors anteriors accediran al sistema de la mateixa manera (via web). Així que tots els anteriors es poden considerar caracteritzacions d'un protoactor que anomenarem *usuari*. A més, per exigències dels requeriments de la tercera iteració, veiem necessari afegir també l'actor *temporitzador* com a representació d'accions de forma periòdica en el temps.

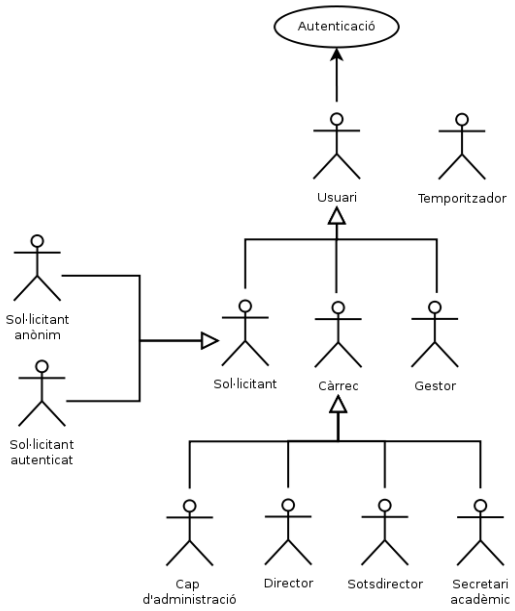

Figura 2.4: Diagrama d'actors del sistema.

2.5 Casos d'ús

Amb els actors que interactuaran amb el sistema establerts, i partint de la llista final de requeriments, obtenim els casos d'ús del sistema de la figura 2.5.

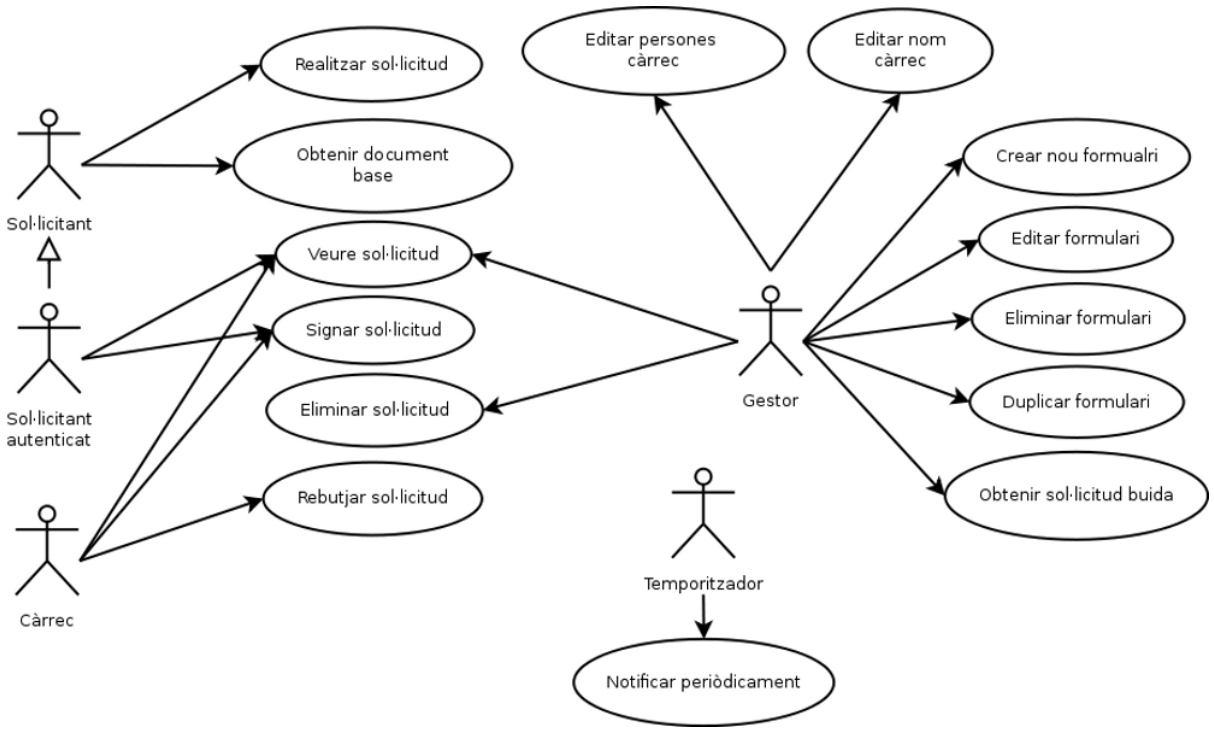

Figura 2.5: Diagrama de casos d'ús del sistema.

A continuació llistarem els contractes de cada cas d'ús, junt amb el diagrama de seqüència apropiat si és rellevant.

2.5.1 Sol·licituds

Cas d'ús:	Realitzar sol·licitud (2.6)
Actor:	Sol·licitant
Descripció:	El sol·licitant demana un formulari, l'omple i envia la informació per realitzar la sol·licitud. Només es té accés si l'usuari està autenticat o si el formulari permet sol·licituds anònimes. Si les dades enviades són correctes i el sol·licitant confirma la sol·licitud, el sistema les enregistra i les prepara per seguir el procés administratiu adequat (si existeix).
Casos alternatius:	**1.** La sol·licitud s'ha de signar i el sol·licitant signa immediatament. **2.** La sol·licitud s'ha de signar i el sol·licitant deixa la signatura per més endavant. **3.** El formulari és extern i primer s'aconsegueix el document base, s'omple i s'envia com a PDF. Acte seguit aquest és preparat per la introducció al procés administratiu.

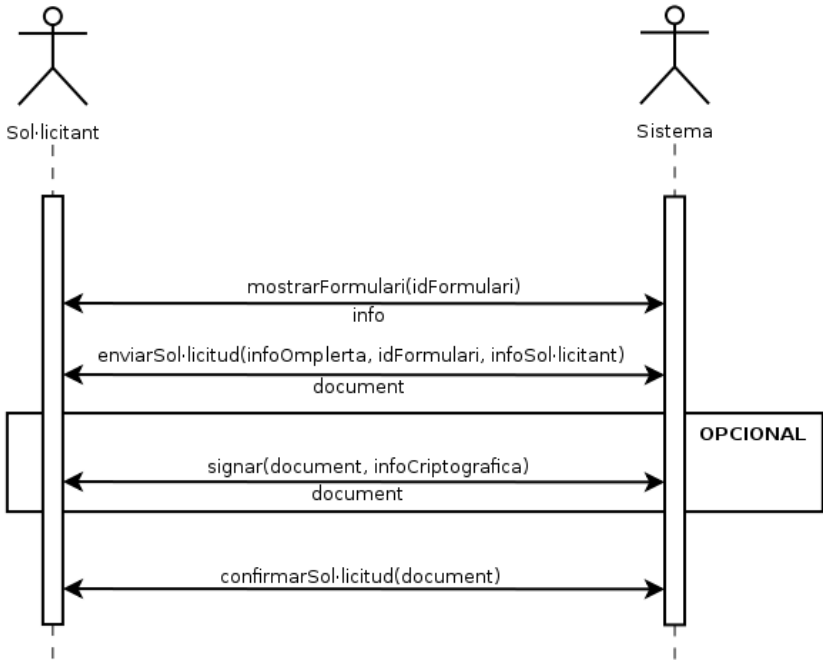

Figura 2.6: Cas d'ús principal de realitzar sol·licitud.

Cal mencionar les discussions de requeriments al tractar el tercer cas alternatiu

presentat respecte la realització de sol·licituds. Inicialment (primera iteració) es va considerar la idea d'utilitzar fitxers .doc com a documents base, que els usuaris els omplissin i els enviessin al sistema. En aquest cas el sistema s'hauria d'encarregar de fer les transformacions necessàries a PDF abans d'introduir la sol·licitud al sistema. Això, a l'etapa de disseny, comportava certs problemes:

- Els documents base estaven limitats a fitxers de Word (.doc).
- La semblança del document final amb l'original no estava garantida (segons el procés de transformació el fitxer resultant podia canviar bastant). En casos extrems, parts del document podien quedar ocultes per altres, cosa que no volem en un contracte vinculant.
- Les solucions software per la transformació a PDF d'una forma programàtica no són gaire bones ni versàtils. Podem considerar un servei extern de transformació, però això implica un temps de resposta molt variable i dependre d'un sistema extern pels processos administratius del departament amb l'exterior, opcions gens desitjables tampoc.
- Relacionat amb el punt anterior, s'ha de tenir en compte que el format .doc ha estat històricament subjecte a modificacions incompatibles amb versions anteriors. Tot i què això ha canviat des de 2008, quan Microsoft va lliberar l'especificació del format [24], n'han sortit de nous (com el .docx) que fan que no sigui possible tenir una sola eina que tracti de forma universal totes les versions del format.

Per aquestes raons finalment en la segona iteració s'ha optat (discutint-ho amb el cap d'administració i el tutor del projecte) per demanar que la transformació a PDF es faci per part del client. És un procés lleugerament menys pràctic que l'anterior, ja que obliguem a l'usuari a saber com transformar un document a PDF, però avui en dia la majoria d'editors de documents permeten l'exportació a document PDF de forma senzilla. A més, d'aquesta manera podem utilitzar qualsevol document com a base i és el sol·licitant el què decideix si el PDF resultant és representatiu o no.

En tot cas cal tenir present que el PDF obtingut del sol·licitant pot necessitar modificacions per adaptar-lo al nostre sistema (per la signatura digital, per exemple).

Cas d'ús:	Obtenir document base
Actor:	Sol·licitant
Descripció:	El sol·licitant demana el document base d'un formulari extern i el sistema li retorna.

Cas d'ús:	Veure sol·licituds
Actor:	Sol·licitant autenticat, Càrrec, Gestor
Descripció:	El sol·licitant demana veure les sol·licituds seves fetes anteriorment. El sistema li retorna una llista amb informació de les sol·licituds, el seu estat i el document PDF.
Casos alternatius:	**1.** Un càrrec demana veure les sol·licituds pendents de signar. **2.** Un càrrec demana veure totes les sol·licituds d'un formulari on tingui accés. **3.** El gestor demana veure totes les sol·licituds d'un formulari.

Cas d'ús:	Signar sol·licitud (2.7)
Actor:	Sol·licitant, Càrrec
Descripció:	El sol·licitant signa una sol·licitud abans d'enviar-la. El primer càrrec (si existeix) que necessita signar la sol·licitud és notificat.
Casos alternatius:	**1.** El sol·licitant signa un conjunt de sol·licituds prèvies sense signar. Els primers càrrecs (si existeixen) que les necessiten signar són notificats. **2.** Un càrrec signa un conjunt de sol·licituds pendents de la seva signatura. El següent càrrec del procés és notificat, o si aquest és l'últim es notifica al sol·licitant que la seva sol·licitud ha estat acceptada.

Cas d'ús:	Eliminar sol·licitud
Actor:	Gestor
Descripció:	El gestor selecciona una sol·licitud d'un formulari donat i l'elimina. Això inclou el document PDF i totes les dades relatives; el sistema no notifica a ningú ni deixa cap indici de l'esborrat.

Cas d'ús:	Rebutjar sol·licitud
Actor:	Càrrec
Descripció:	Un càrrec selecciona una sol·licitud i la rebutja, especificant la raó pel rebuig. El sol·licitant és notificat de què la seva sol·licitud ha estat rebutjada i perquè.

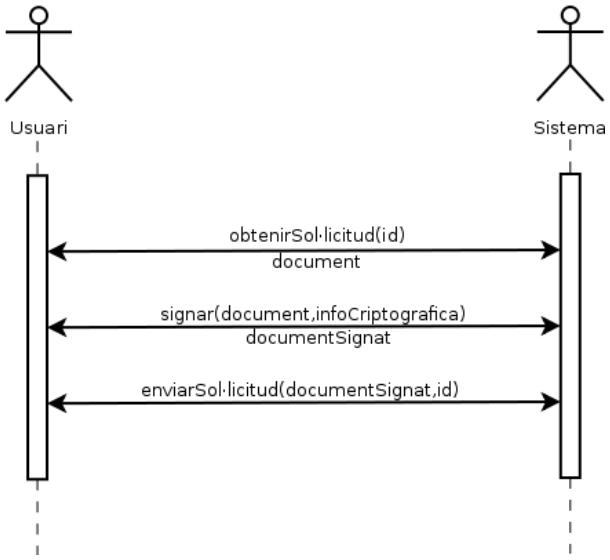

Figura 2.7: Diagrama de seqüència de la signatura de sol·licituds.

2.5.2 Càrrecs

No permetrem, com a funcionalitat del propi sistema, afegir i eliminar càrrecs. La raó d'aquesta decisió és que aquesta funció seria molt perillosa per la robustesa del sistema; eliminar un càrrec posaria el sistema en un estat incoherent si aquest participa en algun formulari. A més, una interfície d'edició de càrrecs no seria massa útil, ja que aquests varien de forma molt poc freqüent. Així doncs, a efectes del sistema, considerarem els càrrecs (però no els seus membres) com a constants.

A més, donat que l'encarregat de gestionar càrrecs és el gestor, exclourem l'administració de gestors del sistema. La raó en aquest cas és evitar que els usuaris es "tanquin" a ells mateixos, i.e. l'últim gestor fent-se fora a si mateix i que ningú pugui tornar a gestionar usuaris. Per altra banda, els usuaris amb rol de gestor normalment serà, per la naturalesa administrativa del sistema, el cap d'administració, així que tampoc caldrà ser modificat sovint. Així, també considerarem els usuaris gestors constants.

Per estar obert per modificació, però, procurarem a la etapa de disseny que els càrrecs i els gestors siguin fàcilment modificables mitjançant algun tipus de configuració no accessible pels actors del sistema. D'aquesta manera es poden modificar els càrrecs mentre simplifiquem la interficie i evitem accidents greus en la operativa del programari.

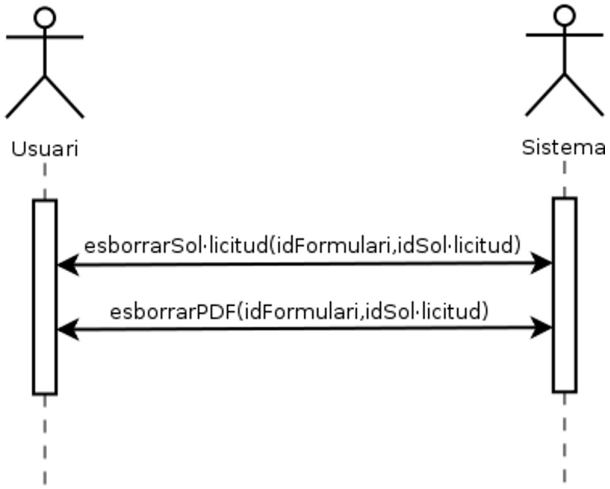

Figura 2.8: Cas d'ús d'eliminació de sol·licitud.

Cas d'ús:	Editar persona càrrec
Actor:	Gestor
Descripció:	El gestor afegeix una nova persona en un càrrec existent, especificant la informació necessària.
Casos alternatius:	
	1. El gestor canvia la informació d'alguna persona d'un càrrec.
	2. El gestor esborra una persona com a membre d'un càrrec.

Cas d'ús:	Editar nom càrrec
Actor:	Gestor
Descripció:	El gestor canvia el nom dels membres d'un càrrec (e.g. director, membre de direcció). La utilitat d'aquesta funció és representar la persona actual que ocupa el càrrec de forma adequada (per exemple, en el gènere).

2.5.3 Formularis

Cas d'ús:	Crear formulari
Actor:	Gestor
Descripció:	El gestor demana la creació d'un nou formulari intern i el sistema el crea, amb les dades necessàries (opcions, camps i procés administratiu) per defecte.
Casos alternatius:	
	1. El gestor demana la creació d'un nou formulari extern i el sistema el crea, amb les dades necessàries (opcions i procés administratiu) per defecte i sense document base definit.

Cas d'ús:	Editar formulari
Actor:	Gestor
Descripció:	El gestor selecciona un atribut del formulari (dels definits als requeriments) i li dona un nou valor.
Casos alternatius:	
	1. El gestor selecciona, en el cas d'un formulari intern, un camp i canvia els seus valors o atributs.
	2. El gestor, en el cas d'un formulari extern, envia un nou document base que substitueix l'antic.
	3. El gestor fa algun canvi en el procés d'un formulari.

Cas d'ús:	Eliminar formulari
Actor:	Gestor
Descripció:	El gestor selecciona un formulari i demana el seu esborrat. Després de què el sistema avisi de la destrucció de totes les sol·licituds i el gestor estigui d'acord, el sistema esborra el formulari i totes les sol·licituds i documents associats.
Casos alternatius:	
	1. El gestor demana l'esborrat d'un formulari però cancel·la la operació i el sistema no pateix canvis.

Cas d'ús:	Duplicar formulari
Actor:	Gestor
Descripció:	El gestor demana la duplicació d'un formulari i el sistema crea un nou formulari amb la mateixa estructura (atributs, camps i procés administratiu) que el formulari original. En el cas dels formularis externs el document base NO és duplicat.

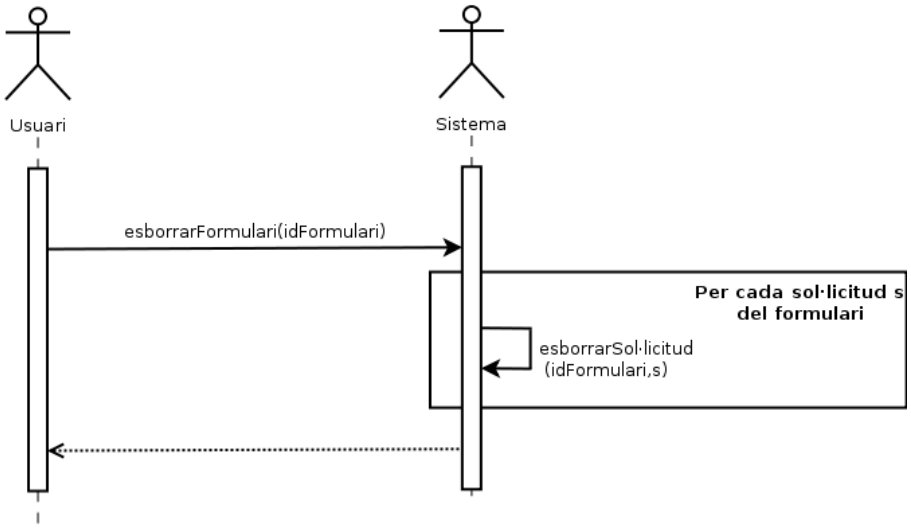

Figura 2.9: Cas d'ús d'eliminació de formulari.

Cas d'ús:	Obtenir sol·licitud buida
Actor:	Gestor
Descripció:	El gestor demana al sistema una sol·licitud en blanc d'un cert formulari intern i el sistema li retorna un document PDF que representa una sol·licitud buida.
Casos alternatius:	
	1. El gestor demana al sistema una sol·licitud en blanc d'un cert formulari extern i el sistema li retorna el document base.

2.5.4 Altres

Cas d'ús:	Autenticació
Actor:	Usuari
Descripció:	Un usuari demana accés autenticat a la web, mitjançant usuari i password UPC o per targeta, i el sistema, si les dades són correctes, li permet entrar amb els permisos associats a l'usuari.
Casos alternatius:	
	1. Un usuari autenticat decideix sortir del sistema, perdent tot accés restringit fins que es torni a autenticar.

Cas d'ús:	Notificació periòdica
Actor:	Temporitzador
Descripció:	Cada cert temps es comproven quins formularis ja expirats tenen sol·licituds pendents de signar per algun càrrec. Després el sistema els hi notifica de què encara es necessita la seva signatura.

2.6 Planificació inicial

Tot i no saber en massa detall el que comportarà el projecte, sabent els requeriments podem elaborar una primera planificació. Com hem dit a la introducció, ens limitarem a tres grans iteracions, compostes per subiteracions més petites, amb els objectius següents:

Primera iteració: Definir la presentació i funcionament general del sistema: interfície web, generació de formularis, enviament de sol·licituds, funcionament (bàsic) de signatura electrònica, ... La idea és obtenir un prototipus completament funcional, tot i ser de forma superficial.

Segona iteració: Refinar la signatura electrònica (tipus de signatura, informació necessària per afegir, millores d'ús, ...) i la generació de PDFs.

Tercera iteració: Afegir funcionalitats i millores menors, refinar les interfícies i acabar de depurar el sistema.

En la primera iteració ens concentrarem en la part web, la qual cosa ens fa pensar (sobretot per experiència prèvia) que la major part del temps la dedicarem a la implementació, on és possible que hi hagi problemes amb els diferents navegadors i s'hagi d'ajustar bastant el codi (relatiu a la presentació web).

Per altra banda, en quant la signatura digital, esperem reutilitzar algun software ja existent, ja que es tracta d'un procediment que, en general, ha estat implementat en molts llocs diferents i, per tant, és pràcticament segur que poguem trobar-ne algun. El que costarà més serà la adaptació d'aquest sistema a les nostres necessitats específiques, la qual cosa estudiarem a la fase de disseny. Un raonament idèntic es pot aplicar a la generació de documents PDF.

Per acabar, la tercera iteració esperem que consisteixi principalment en trobar i arreglar problemes, així que una bona part consistirà en implementació i, sobretot, proves.

2.7 Anàlisi econòmic

A partir de la planificació prèvia podem elaborar un pressupost aproximat del cost final del projecte. Per això necessitarem considerar diverses àrees.

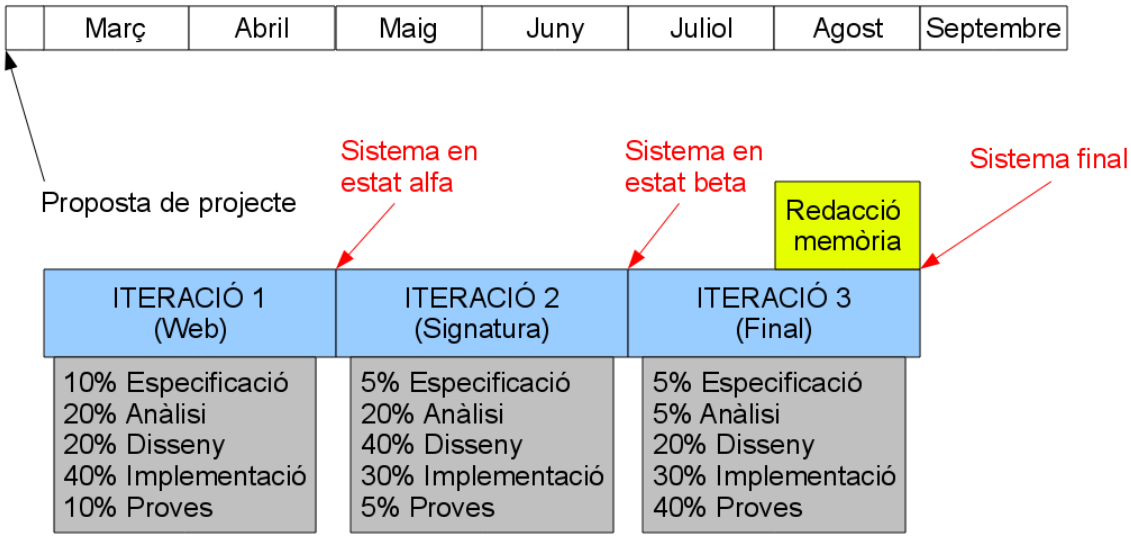

Figura 2.10: Planificació inicial del desenvolupament del projecte.

2.7.1 Recursos humans

Per calcular el cost del recursos humans utilitzats en aquest projecte desglossarem la participació humana real (una persona) en diversos rols als què assignarem un sou per hora i un cert pes. A la figura 2.10 podem veure cada iteració dividida en diverses fases del desenvolupament software. De forma aproximada podem relacionar cada fase amb un rol, als que assignarem un sou per hora:

Fase	**Rol**	**Sou per hora**
Especificació	Cap de projecte	80 euros
Anàlisi	Analista	60 euros
Disseny	Arquitecte	70 euros
Implementació	Programador	50 euros
Proves	Equip de proves	40 euros

Amb aquests sous podem ponderar un sou mitjà de tots els participants en tot el projecte de la següent manera:

$$\begin{aligned} \text{Sou ponderat} &= 80 \times \tfrac{10+5+5}{300} + 60 \times \tfrac{20+20+5}{300} + 70 \times \tfrac{20+40+20}{300} + 50 \times \tfrac{40+30+30}{300} + 40 \times \tfrac{10+5+40}{300} = \\ &= 57,02 \text{ euros/hora} \end{aligned}$$

Veiem que la duració esperada del projecte és de 6 mesos, on estimarem una dedicació de 8 hores diàries, la qual cosa ens resulta una duració total de:

$$6 \text{ mesos} \times 4 \frac{\text{setmanes}}{\text{mes}} \times 5 \frac{\text{dies}}{\text{setmana}} \times 8 \frac{\text{hores}}{\text{dia}} = 960 \text{ hores}$$

Així doncs, el cost dels recursos humans serà de $960 \times 57,02 =$ **54.739,2 euros**.

2.7.2 Hardware

Veient els requeriments funcionals, podem intuir que el sistema es dividirà en una part de client i una altra de servidor.

Per aquesta última només necessitarem un servidor on poder allotjar el software necessari. En molts casos, la organització client que utilitzi el software ja disposarà d'un servidor que subministri altres serveis. A més, mitjançant eines de virtualització es pot aconseguir que aquest servidor ja existent es comporti com diferents màquines des del punt de vista de la xarxa, en cas de què es vulguin aïllar diferents conjunts de serveis. En definitiva, el cost en hardware addicional serà **probablement nul**, tot i què en cas de no disposar d'un servidor això pot suposar un cost d'uns **1500 euros**. En aquest últim cas, però, s'ha de tenir en compte que aquesta despesa serà un inversió per l'ús d'altre software, de manera que no tot és imputable a aquest projecte.

En quant als usuaris del sistema, tota la interacció es realitzarà via web, però es necessitarà un lector de targetes criptogràfiques si es vol (o es necessita) utilitzar la funcionalitat de signatura digital i/o autenticació via targeta. Un lector d'aquestes característiques pot costar al voltant de **20 euros** per cada terminal a on es vulgui implantar.

Apart de la despesa deguda al desplegament del sistema també existeix el hardware que ha estat necessari pel seu desenvolupament. En aquest subapartat només contarem un ordinador portàtil en el que s'ha portat a terme tota la implementació del projecte: el seu cost és de 1.000 euros, però només imputarem al projecte un 20% del seu cost real, fent un subtotal de **200 euros**.

2.7.3 Software

Un dels punts importants que considerarem en aquest projecte és la utilització de software lliure. D'aquesta manera el cost de llicències serà zero, i, si podem disposar del codi font, podrem assegurar que el codi serà extensible en cas de necessitat. Així evitarem acabar "tancats" amb solucions software que resulten inevitablement obsoletes amb el pas del temps.

2.7.4 Altres despeses

De forma perifèrica a la realització d'aquest projecte, hem de considerar també el cost d'altres conceptes. En el nostre cas, contarem amb una despesa mensual de 30 euros d'electricitat i 50 euros d'accés a Internet, que amb els sis mesos de duració fan un total de **480 euros**. Aquests imports mensuals han estat calculats estimant l'ús dels recursos que aplicarem a la realització del projecte; el cost real serà més elevat.

2.7.5 Impostos i cost final

Amb els apartats anteriors hem obtingut un cost total brut de:

$$\begin{aligned} \text{Cost brut} &= \text{Cost RRHH} + \text{Cost Hardware} + \text{Altres despeses} = \\ &= 54.739,2 + 200 + 480 = \\ &= 55.419,2 \text{ euros} \end{aligned}$$

En aquest cost brut de 55.419,2 euros li afegim el 18% d'IVA per obtenir el cost final del sistema:

Cost final: 55.419,2 × 1,18 = **65.394,7 euros**

2.8 Anàlisi de riscos

Feta la planificació i l'anàlisi econòmic, pot ser útil també fer un petit *anàlisi de riscos*. Això implica identificar a priori aquelles àrees on creiem que poden sorgir dificultats i valorar la probabilitat de fracàs i l'impacte que podrien tenir sobre el projecte, així com pensar un pla alternatiu en cas de problemes.

A la figura 2.11 representem gràficament l'anàlisi de riscos possibles, per ordre d'impacte en el projecte.

1. En primer lloc es troba, previsiblement, la pròpia implementació de la signatura digital. La probabilitat moderada de fracàs no és deguda a la incapacitat d'implementar-la, sinó a la incapacitat d'implementar-la de forma amigable a l'usuari. Si no fóssim capaços l'impacte seria molt alt, ja que l'eix central del projecte és precisament la signatura digital. En aquest cas possiblement ens hauríem de plantejar algún programa d'escriptori per signar, afectant negativament a la usabilitat del sistema.

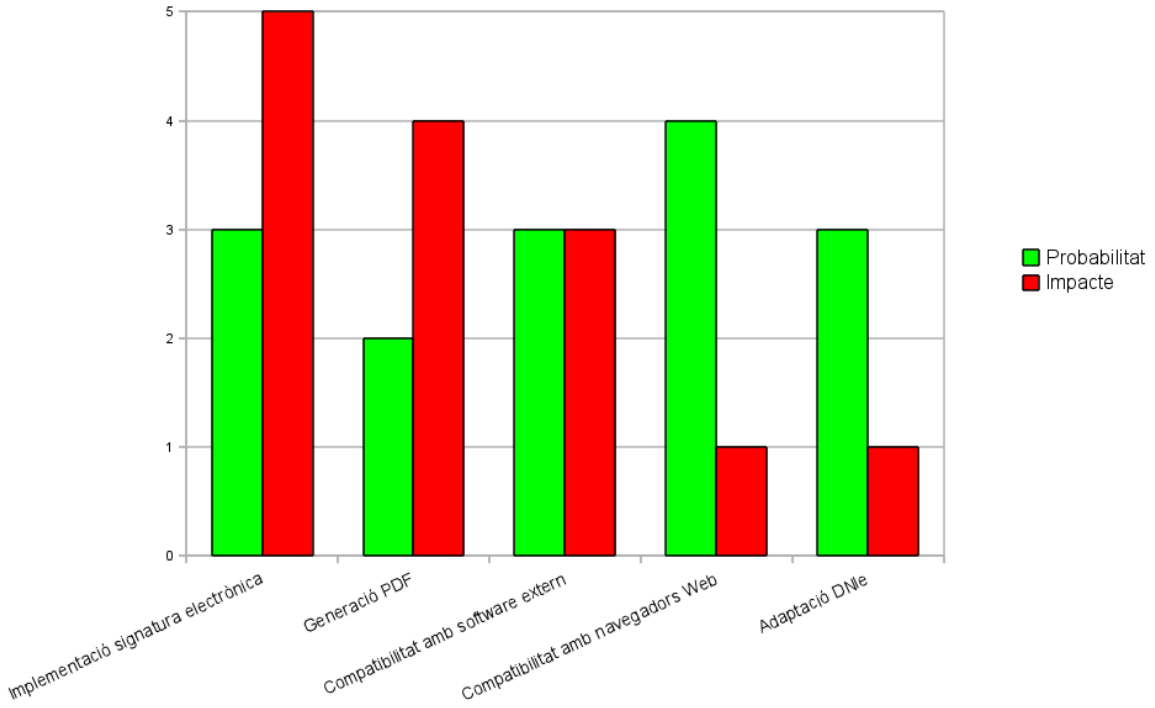

Figura 2.11: Anàlisi de riscos, amb la seva probabilitat i impacte en una escala del 1 al 5.

2. La generació de documents PDF també és un punt important. Tot i no ser massa probable el seu fracàs (hi ha multitud d'eines i sistemes que generen PDFs), el seu impacte seria bastant alt perquè com hem dit és el tipus de documents que signarem. Si ens trobéssim problemes amb aquest punt podríem desestimar utilitzar PDFs, generant i signant sol·licituds en altres formats, com per exemple XML.

3. Un altre possible risc és la compatibilitat dels artefactes que generem en el sistema (com documents PDF i signatures) amb software extern. Per la lectura de fitxers PDF confiarem amb software de tercers (com Adobe Reader), igual que en la validació de les signatures. Això té un impacte variable, ja que podem o bé buscar un altre tipus de documents (com el XML que hem dit anteriorment) o bé si no hi ha més opció crear el nostre propi software extern. Aquest és un problema sobretot amb la signatura electrònica, ja que construir un sistema de validació de certificats no seria una tasca fàcil ni ràpida.

4. Potser un subconjunt del cas anterior és la compatibilitat de la web del sistema amb els diferents navegadors. Experiència prèvia en el camp del disseny web ens diu que és molt difícil crear una web que tingui la mateixa presentació en tots els navegadors, ja que cada implementació interpreta diferent els estàndards visuals. Tot i l'alta probabilitat de tenir problemes per uniformitzar la visualització de la web, el seu impacte serà probablement baix, ja que normalment les diferències seran purament estètiques. Malauradament, un pla alternatiu és complicat; la única cosa

que podem fer és reduir la complexitat de la presentació per evitar conflictes.

5. Finalment tenim l'adaptació de l'ús del DNI electrònic en el sistema, de forma conjunta amb el carnet UPC. La probabilitat de fracàs és mitjana, ja que pot ser difícil fer conviure les dos targetes satisfactòriament. Tot i això, l'impacte és petit: si no ens en sortim sempre podem apartar el DNIe i quedar-nos amb la signatura mitjançant el carnet UPC.

Capítol 3

Anàlisi

En aquesta fase realitzarem un anàlisi global del sistema. Com qualsevol anàlisi, això implica partir d'un concepte gran i complex, com ho és el nostre projecte, i trencar-lo en parts senzilles per facilitar la seva comprensió o, en el nostre cas, la seva construcció. Una vegada tenim les parts també hem d'entendre de quina manera interaccionen entre elles, mitjançant la seva especificació. D'aquesta manera, podrem procedir a dissenyar cada subsistema de forma independent.

Començarem amb l'*arquitectura d'anàlisi*.

3.1 Arquitectura d'anàlisi global

En el capítol anterior hem especificat el sistema com una gran caixa negra. Per elaborar l'arquitectura d'anàlisi necessitem decidir en quines àrees es pot dividir (de forma independent i cohesionada) aquesta caixa i la seva funcionalitat interna. Si ho fem, podem detectar aquestes àrees:

- Web de creació i visualització de formularis
- Generació de documents PDF
- Signatura digital de documents PDF
- Sistema de notificació
- Sistema d'autenticació
- Repositori general
- Repositori de documents

Òbviament per l'ús del sistema també és necessari un navegador web, però no és necessària la seva implementació així que no l'incloïm.

Amb l'anterior llista podem començar a pensar la distribució d'aquests subsistemes i la comunicació entre ells. Tal com hem quedat amb els requeriments, la interacció de l'usuari es farà tota via web, que serà per tant el nucli del sistema. Aquesta delegarà funcions a les altres parts (que en principi no es comunicaran entre si). Seguint aquesta filosofia aconseguim minimitzar l'acoblament entre l'usuari i el sistema (un únic punt d'entrada) i entre els mateixos subsistemes. L'única excepció és la signatura digital, que per necessitat ha de ser realitzada per l'usuari i per tant ha d'existir a la part del client. Així doncs, obtenim l'arquitectura d'anàlisi de la figura 3.1.

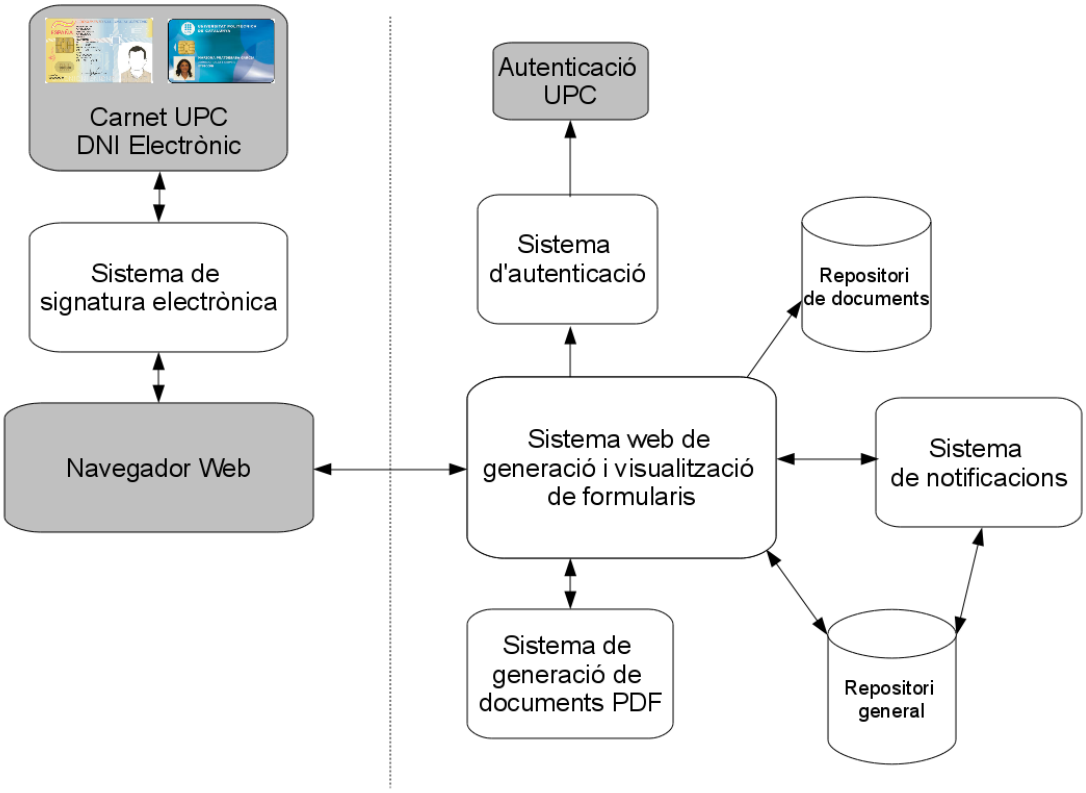

Figura 3.1: Arquitectura d'anàlisi del sistema. Els subsistemes marcats en gris són aquells externs al sistema.

Veiem que el component de notificació té una connexió directa amb el repositori general. Aquesta modificació ha estat afegida a la tercera iteració, on nous requeriments exigien que de forma periòdica es recordés als càrrecs que tenen sol·licituds pendents. En aquest cas qui inicia l'acció no és la web, sino un temporitzador, i per tant és més raonable que s'implementi amb una relació directa al repositori. Com veurem a l'especificació del sistema de notificació, aquest es pot dividir clarament en dues parts, una acoblada amb el sistema web i l'altra amb el repositori, així que tot plegat continua estant ben cohesionat.

3.2 Especificació dels components

A continuació construirem una especificació més detallada, de forma similar a com ho hem fet al capítol anterior però a nivell de component.

3.2.1 Web de formularis

Donat que ja hem establert que l'únic punt de contacte de l'usuari és la web, això vol dir que l'especificació d'aquest component es correspon pràcticament per complet a la interfície del sistema com a caixa negra (casos d'ús de la secció 26). Hi ha alguna particularitat, però:

Com hem dit, el cas de signar sol·licituds és l'únic en el que l'usuari necessita comunicació fora del component web. Així que de la transacció de tres passos al signar una sol·licitud (obtenir sol·licitud, signar sol·licitud i enviar sol·licitud), només el primer i l'últim són responsabilitat del component web (figura 3.2).

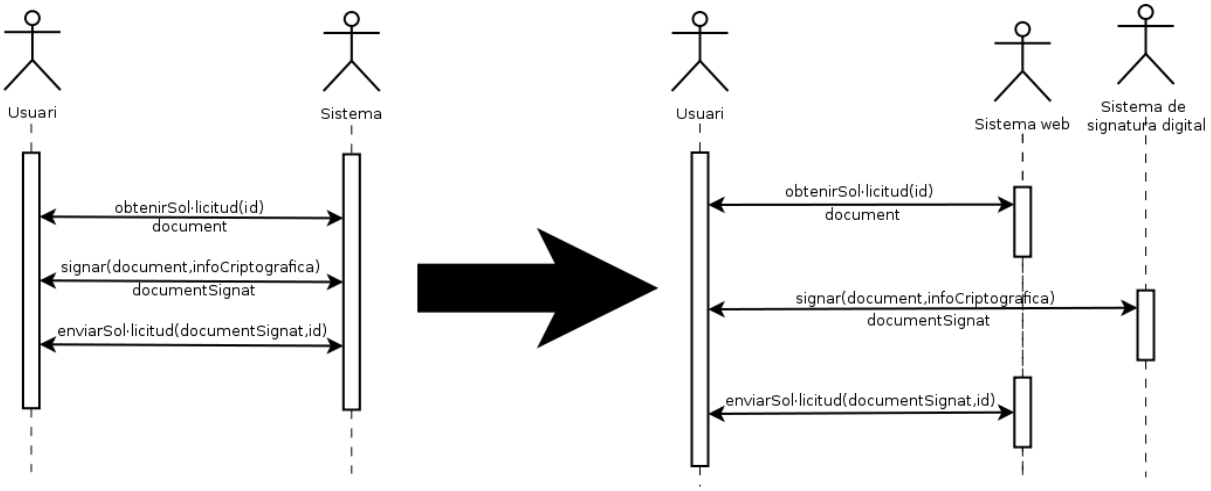

Figura 3.2: Canvis en el cas d'ús de la signatura digital de sol·licituds.

Per la resta de funcionalitats, veure els casos d'ús d'especificació del sistema (figura 2.5).

3.2.2 Generació de documents PDF

Aquest component, igual que tots els següents amb l'excepció de la signatura, són utilitzats pel component web per complir amb les seves responsabilitats. Així doncs, l'actor en aquests casos serà sempre el component web.

El cas d'ús principal de la generació de documents PDF serà la generació de sol·licituds en format PDF a partir de les dades necessàries. Com hem vist al cas d'ús per realitzar una sol·licitud d'un formulari extern (pàgina 27), també necessitarem ser capaços de modificar un document PDF enviat per l'usuari i preparar-lo per ser signat digitalment.

Cas d'ús:	Generar document PDF
Actor:	Sistema web
Paràmetres:	informació de la sol·licitud, informació del sol·licitant, informació sobre el procés administratiu del formulari
Pre:	
1.	La informació de la sol·licitud ha d'estar completa i ser vàlida.
Post:	
1.	Es retorna un document PDF amb la informació donada.

Cas d'ús:	Preparar document PDF per la signatura
Actor:	Sistema web
Paràmetres:	document PDF, informació sobre el procés administratiu del formulari
Pre:	
1.	El document PDF ha de ser vàlid.
2.	El procés administratiu ha de ser vàlid.
Post:	
1.	Es retorna el mateix document PDF però preparat per ser signat pels càrrecs especificats al procés administratiu apropiat.

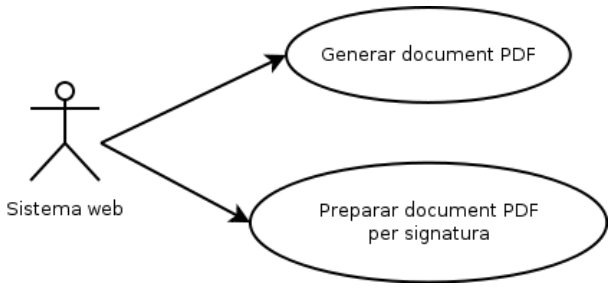

Figura 3.3: Casos d'ús del sistema de generació de documents PDF.

3.2.3 Signatura digital de documents PDF

L'especificació d'aquest subsistema és senzilla, simplement ens ha de permetre signar un document PDF amb una targeta criptogràfica (UPC i, en la segona iteració, el DNIe). Tot i això, serà una mica diferent que les signatures digitals convencionals en el sentit de què no tenim una clau secreta amb la que executar l'algoritme de xifratge, sinó que aquest s'executarà dins la mateixa targeta (ho mencionarem amb més detall al capítol 4). Per tant, haurem de considerar la pròpia targeta criptogràfica com a un subsistema més dins de l'arquitectura global.

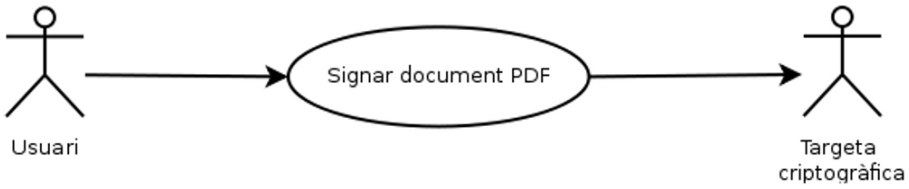

Figura 3.4: Cas d'ús del sistema de signatura digital.

Cas d'ús:	Signar document PDF
Actor:	Usuari
Paràmetres:	document PDF, rol del signant
Pre:	
1.	El document PDF ha de ser vàlid.
2.	El rol del signant ha de ser vàlid i ser necessària la seva signatura al document.
Post:	
1.	Es retorna el mateix document PDF signat per l'usuari amb el rol donat.

3.2.4 Sistema de notificació

Com hem dit anteriorment, aquest sistema es podria dividir encara més en dos subsistemes, de tal manera que l'acoblament sigui mínim de cara a l'exterior. Això queda representat a la figura 3.5.

En un principi només teniem les notificacions de noves sol·licituds (notificacions immediates a la figura), amb l'objectiu de notificar en el mateix moment en que una sol·licitud canviï d'estat (una nova sol·licitud al sistema, una sol·licitud signada a l'espera del següent signant, una sol·licitud acabada de signar i acceptada, ...). Més endavant, en la tercera iteració, es va afegir el requeriment de notificacions periòdiques, per tal d'avisar als càrrecs que encara existeixen documents per signar en formularis ja caducats. Com aquí l'agent iniciador no és un usuari, el pas per la web és innecessari, així que per aquest cas permetrem un accés directe al repositori.

Per tant, en el cas de les notificacions hi trobem dos casos d'ús: la notificació de noves sol·licituds i la notificació periòdica sobre formularis expirats.

En aquest primer escenari hi ha diversos casos:

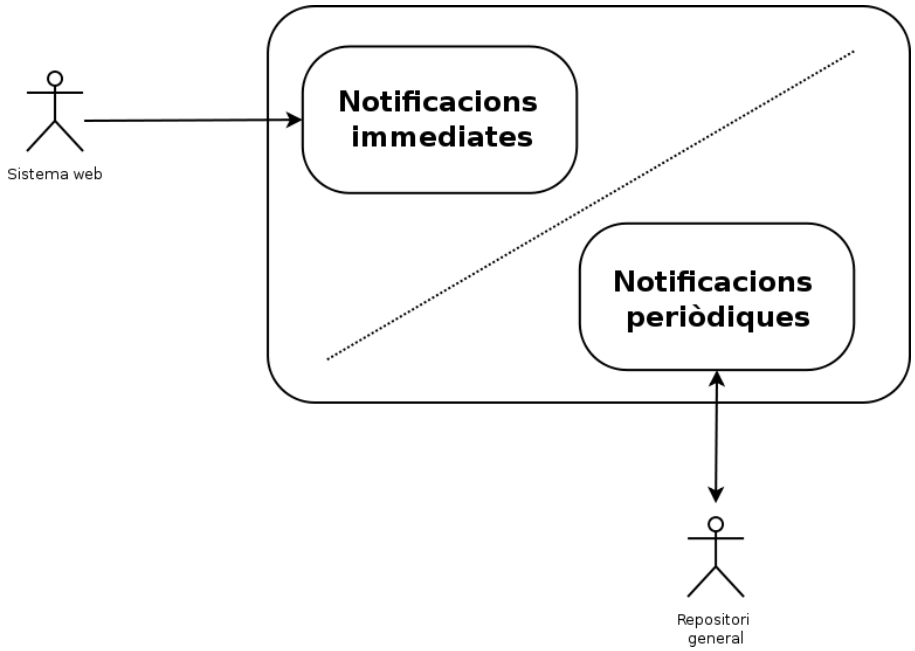

Figura 3.5: Visió interna del sistema de notificació.

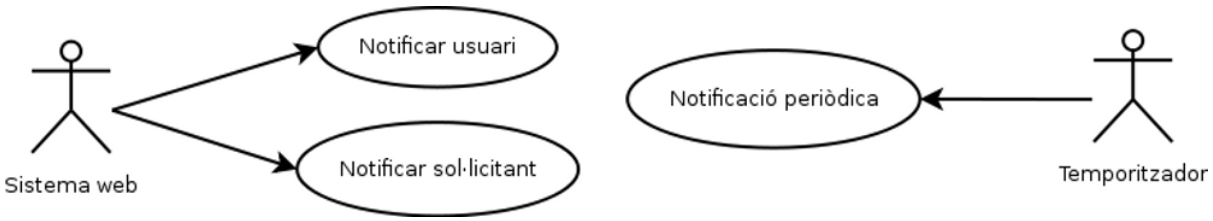

Figura 3.6: Casos d'ús del sistema de notificació.

Cas d'ús:	Notificar sol·licitant
Actor:	Sistema web
Paràmetres:	informació del sol·licitant, nom del formulari, sol·licitud acceptada?
Pre:	
1.	El correu electrònic ha de ser vàlid.
2.	El sol·licitant ha hagut de realitzar una sol·licitud del formulari especificat.
Post:	
1.	Es notifica el sol·licitant fent-li saber si la seva sol·licitud ha estat acceptada o no.

Cas d'ús:	Notificar càrrec
Actor:	Sistema web
Paràmetres:	informació dels membres del càrrec, nom del formulari
Pre:	
1.	Els correus electrònics han de ser vàlids.
2.	El càrrec ha de ser vàlid.
Post:	
1.	Es notifica els membres del càrrec avisant que hi ha noves sol·licituds (possiblement pendents de la seva signatura).

I després existeix el cas apart de la notificació periòdica:

Cas d'ús:	Notificació periòdica
Actor:	Temporitzador
Paràmetres:	-
Pre:	
Post:	
1.	Es notifiquen els càrrecs amb sol·licituds pendents de signar de formularis caducats fa un cert temps.

3.2.5 Sistema d'autenticació

Tal com està llistat als requeriments, hem de permetre l'autenticació mitjançant usuari i password UPC i mitjançant targeta electrònica. Anàlogament al cas de la signatura, hem de considerar la targeta criptogràfica com a un actor apart en l'especificació. De forma similar, hem de tenir en compte que l'autenticació final la farà la UPC (qui té les dades apropiades), així que també l'hem d'incloure com a actor extern.

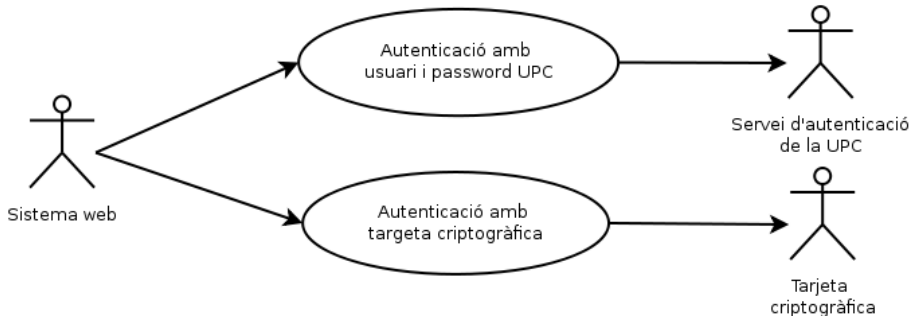

Figura 3.7: Casos d'ús del sistema d'autenticació.

Cas d'ús:	Autenticació amb usuari/password UPC
Actor:	Sistema web
Paràmetres:	usuari i password UPC
Pre:	
Post:	
1.	Es notifica si l'autenticació ha estat correcta o no, i, en cas positiu, retorna la informació de l'usuari.

Cas d'ús:	Autenticació amb targeta electrònica
Actor:	Sistema web
Paràmetres:	-
Pre:	
Post:	
1.	Es notifica si l'autenticació ha estat correcta o no, i, en cas positiu, retorna la informació de l'usuari.

3.2.6 Repositori general

La missió d'aquest subsistema serà la d'oferir totes les operacions necessàries per tractar la informació que necessita el sistema web. Com encara no sabem les necessitats concretes no explicitarem les funcionalitats concretes que ha d'oferir aquest repositori. Tot i això, sí podem estar segurs que l'única cosa que tractarem seran consultes, baixes i modificacions sobre informació relativa a la informació administrativa del sistema, com els formularis, sol·licituds i càrrecs (però excloent els documents base i PDF, que són responsabilitat del següent mòdul).

3.2.7 Repositori de documents

Donat que aquest component és un simple repositori només caldrà la possibilitat d'emmagatzemar documents i recuperar-los. Pels requeriments podem veure que hi haurà dos tipus de documents a tractar en aquest cas: PDF i els documents base dels formularis externs.

Cas d'ús:	Emmagatzemar sol·licitud PDF
Actor:	Sistema web
Paràmetres:	document PDF, identificador de la sol·licitud
Pre:	
Post:	
1.	S'elimina el document PDF anterior (si hi era).
2.	S'emmagatzema el document PDF, representant a la sol·licitud donada.

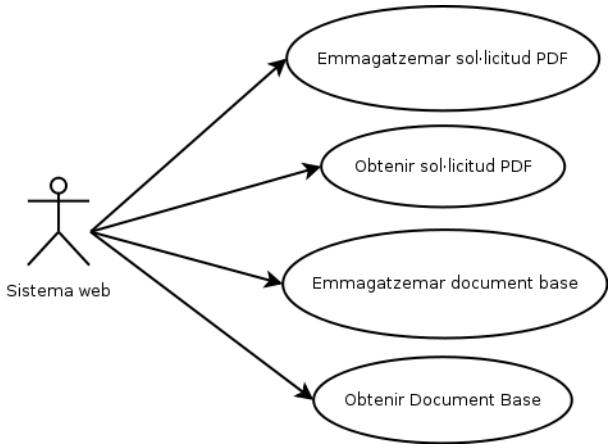

Figura 3.8: Casos d'ús del repositori de documents.

Cas d'ús:	Obtenir sol·licitud PDF
Actor:	Sistema web
Paràmetres:	identificador de la sol·licitud i formulari
Pre:	
2.	El formulari i la sol·licitud en qüestió existeixen.
Post:	
1.	Es retorna el document PDF que representa a la sol·licitud donada.

Cas d'ús:	Emmagatzemar document base
Actor:	Sistema web
Paràmetres:	document base, identificador del formulari extern
Pre:	
Post:	
1.	S'elimina el document base anterior (si n'hi havia).
2.	S'emmagatzema el document base del formulari extern donat.

Cas d'ús:	Obtenir document base
Actor:	Sistema web
Paràmetres:	identificador del formulari extern
Pre:	
1.	L'identificador correspòn a un formulari extern que té un document base definit.
Post:	
1.	Es retorna el document base del formulari extern donat.

Capítol 4

Criptografia i documents PDF

Abans de passar a l'etapa de disseny, i donat que la signatura electrònica de documents PDF és un component important del sistema, tractarem d'investigar el necessàri per poder tenir una bona idea de com dissenyar el sistema. Primerament parlarem de la base criptogràfica que necessitem per entendre el procés de la signatura electrònica, junt amb les infraestructures que utilitzarem en el projecte (l'Agència Catalana de Certificació i la Direcció General de la Policia).

A continuació estudiarem l'estructura bàsica d'un document PDF, amb el detall suficient com per poder entendre com signar i quina informació necessitem incorporar. Per això també haurem de detallar els estàndards de signatura electrònica en fitxers PDF i valorar les seves característiques.

4.1 Definició de criptografia

La *criptografia* (provinent de *kryptos*, secret, i *gráph*, escriptura) és una disciplina que engloba tant coneixements matemàtics com informàtics i que estudia sistemes de xifratge i desxifratge de missatges. De forma relacionada existeix també el *criptoanàlisi*, que s'encarrega de trobar maneres de recuperar informació a partir de missatges xifrats (*criptogrames*) sense el coneixement de les claus necessàries. És a dir, un criptògraf dissenya un criptosistema que xifra un missatge generant un criptograma, que el criptoanalista estudia per trobar vulnerabilitats al criptosistema inicial. Ambdues disciplines formen la branca matemàtica de la *criptologia*. Per més informació específica sobre criptografia, es pot consultar la referència [6].

4.2 Criptografia de clau secreta i clau pública

Podem dividir els sistemes criptogràfics en dos: sistemes de *clau secreta* (o *simètrics*) i de *clau pública* (o *assimètrics*).

La criptografia de clau secreta consisteix en sistemes on la mateixa clau és utilitzada tant per xifrar com per desxifrar un missatge. Aquests mètodes són útils i ràpids quan només intervé un usuari, però per més gent la distribució de claus és un problema, ja que perquè diferents usuaris es comuniquin els dos s'han de posar d'acord prèviament en la mateixa clau, procés que suposa un risc de seguretat.

Per solucionar aquest problema, l'any 1976 Whitfield Diffie i Martin Hellman van publicar un article amb una nova proposta, la criptografia de clau assimètrica. En aquest sistema existeixen dos claus diferents però matemàticament relacionades, una clau pública, disponible per tothom, i una clau privada, que ha d'estar sota el control únic d'un usuari o entitat. Seguint aquest esquema de funcionament, per enviar informació xifrada qualsevol pot utilitzar la clau pública per xifrar el missatge, enviar-li al receptor i aquest desxifrar el missatge amb la clau privada. La robustesa d'un algoritme d'aquestes característiques resideix en la dificultat de calcular la clau privada a partir de la pública.

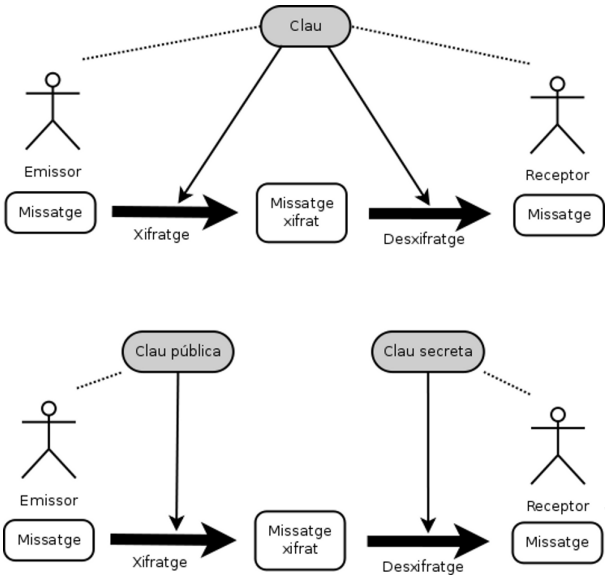

Figura 4.1: Diferència entre sistemes criptogràfics simètrics (figura superior) i assimètrics (figura inferior).

4.3 Algoritmes de xifrat de clau pública

En aquesta secció intentarem descriure els algoritmes de xifrat de clau pública, centrant-nos en el que utilitzarem en aquest projecte, RSA, per tal d'observar quins punts hem de considerar per dissenyar un sistema segur.

Per aquests algoritmes necessitem poder fer el càlcul de forma relativament eficient si disposem de la clau secreta i difícil altrament. Una manera de conseguir això és mitjançant una 'funció trampa' (*trapdoor function*), la qual és fàcil de calcular en un sentit però molt difícil de fer-ho en sentit invers, a menys que es disposi d'alguna informació addicional (com, en aquest cas, la clau secreta).

Per il·lustrar el funcionament amb un exemple, veiem el funcionament de l'algoritme de RSA [31]. Comencem amb la generació de les claus privada i pública:

1. S'escullen dos nombres primers diferents p i q.
2. Es calcula $n = pq$.
3. Es calcula $\varphi(pq) = (p-1)(q-1)$, on φ és la funció d'Euler.
4. Es tria un enter e tal que $1 < e < \varphi(pq)$, on e i $\varphi(pq)$ són coprimers.
5. Determinar una d de forma que compleixi $de \equiv 1 \pmod{\varphi(pq)}$.

Una vegada fet, es guarda d com a clau secreta i es publiquen n i e com a clau pública.

Quan tenim les claus, podem xifrar un missatge m dividint-lo en blocs d'una longitud determinada, on cadascún és transformat a un enter, afegint *padding* per evitar problemes de seguretat, i calculant el criptograma c com $c = m^e \bmod n$. Per desxifrar i recuperar el missatge original m calculem $m = c^d \bmod n$. El bon funcionament d'aquest algoritme es pot demostrar mitjançant el teorema d'Euler: resumint, utilitzar la clau privada i després la pública (o viceversa) ens retorna el missatge original ($m^{ed} \pmod{n} = m$).

Per signar el procediment és idèntic però s'intercanvia el lloc de les claus privada i pública. Això motiva el fet de què en general les claus per signar i per xifrar es generin de forma separada: un atacant podria interceptar un missatge xifrat amb la clau pública d'una persona i després convéncer a la mateixa persona per signar el missatge. Tal com podem deduir matemàticament, això li retornaria el missatge original. Aquest tipus d'atacs són coneguts com a "atacs de criptograma escollit" (*Chosen-ciphertext Attack*) [10].

Del funcionament del RSA podem observar que hi ha dos punts on es fonamenta la robustesa de l'algoritme:

Per una banda podriem obtenir la clau secreta factoritzant n, obtenint els primers p i q i replicar el procés per obtenir la clau privada. Per sort, computacionalment

la factorització és un problema difícil, amb els millors algoritmes actuals d'un cost subexponencial, així que es conjectura que el seu càlcul és, per valors de n suficientment grans, inviable.

Per altra banda podriem calcular la clau secreta (d) a partir d'un criptograma que tinguem ($m^d \pmod n$) i sabent que el missatge original (m) és el *hash* del document signat.En aquest cas ens trobem amb el problema del logaritme discret, també conegut com el problema de RSA [27]. Com en l'anterior cas, també es tracta d'un problema difícil, així que en principi per claus prou grans no hauria de ser un problema.

De tota manera, la conclusió que hem de treure és que per garantir que l'algoritme és suficientment segur hem de tenir en compte la mida de les claus a la fase de disseny per veure si són suficients per les nostres necessitats.

4.4 Infraestructura de clau pública (*PKI*)

Tot l'anterior serveix perquè es pugui construir un sistema de xifratge global, on qualsevol persona pugui enviar contingut xifrat a qualsevol altra amb una més que raonable garantia de confidencialitat. Però si realment es vol aconseguir un sistema de comunicació de confiança necessitem alguna manera d'associar de forma unívoca cada clau pública a un individu. Aquest és l'objectiu de les *infraestructures de clau pública (PKI, o* Public Key Infrastructure*)*.

Una infraestructura de clau pública és el conjunt de serveis i entitats que permeten que un conjunt d'individus es puguin identificar com a tals davant d'altres amb una certa confiança. Al centre d'aquesta infraestructura es troben els *certificats digital*, fitxers amb informació associada a una entitat o individu i que contenen dades relatives al titular del certificat, així com informació relativa a l'emissió i validació de certificat i, en particular, una clau pública. Òbviament la clau privada no està inclosa al certificat.

Aquests certificats digitals i la resta de la infraestructura poden estar implementades de diverses maneres, però la més extesa i la que tractarem és l'anomenada *PKIX*. La 'X' significa que s'utilitza el format X.509 i directives relacionades per la representació digital dels certificats, els seus atributs i altra informació que veurem més endavant com les llistes de revocació o el mètode de validació. A més, la majoria de PKIs actuals utilitzen un conjunt d'estàndards coneguts com a PKCS (*Public Key Cryptography Standards*), publicats per l'empresa RSA Security.

A continuació llistem alguns atributs típics que pot contenir un certificat X.509:

- Informació del titular
- Informació de l'emissor del certificat
- Algoritme a utilitzar

- Número de sèrie
- Versió del certificat
- Data de caducitat
- Clau pública
- Informació sobre validació
- Extensions
- . . .

Les extensions són de particular interès també, ja que defineixen aspectes necessaris per la operativa de la PKI tals com l'ús que se li donarà al certificat (signatura digital, autenticació, xifrat, ...), a on es pot validar el certificat, si té permís per emetre altres certificats, si té permís per signar informació relativa a la validació, etc. Algunes d'aquestes extensions són d'ús imprescindible (extensions crítiques), de manera que si un sistema software no és capaç de reconéixer el seu significat, s'ha de considerar un error. Per altra banda les extensions no crítiques poden ignorar-se si no s'entenen.

A l'annex A.1 podem veure un exemple complet de certificat X.509.

4.4.1 Autoritats de Certificació

Aquests certificats són expedits per *autoritats de certificació, (AC)*. Les AC recopilen dades sobre l'individu, generen les claus (o són generades per l'usuari i s'envia la clau pública a la AC), creen el certificat amb aquesta informació i seguidament signen digitalment el certificat (veurem en què consisteix la signatura a l'apartat 4.6). A vegades aquesta tasca es delega aquesta feina en altres entitats anomenades *autoritats de registre (AR)*. A més de la seva generació, les AC han de garantir la seguretat, confidencialitat, distribució i manteniment de l'estat dels certificats digitals. Per aquesta raó aquestes autoritats han de complir directives de seguretat suficientment estrictes com per garantir el bon funcionament i fiabilitat de tota la infraestructura de clau pública, definits a la normativa ISO 27002 [41].

Aquest procés de creació de certificats genera una *cadena de certificació*, on el certificat de cada AC és emès per una altra autoritat, fins que s'arriba a una autoritat el certificat de la qual és signat per ella mateixa. Aquesta autoritat se l'anomena *autoritat certificadora arrel*, mentre que la resta són *autoritats certificadores intermitges*.

4.4.2 Validació de certificats

Com hem vist anteriorment, un certificat té una data de caducitat. També, si la clau privada associada es veu compromesa, s'ha de permetre al titular la revocació (permanent) o suspensió (temporal) del certificat per impedir que s'utilitzi de forma no autoritzada per un tercer. Per aquestes raons, una bona PKI necessita un procediment per la validació de certificats. Una CA ha de disposar d'una llista de tots els certificats emesos per ella mateixa que han estat revocats, i ha de permetre a un sistema extern consultar d'alguna manera aquesta informació. Aquest servei el poden oferir les pròpies CAs o bé es poden crear *autoritats de validació (AV)* per especialitzar-se en aquest procés. Igual que amb els individus, també es validen les autoritats intermitges amb els seus emissors fins arribar a l'autoritat arrel, en la qual es confia incondicionalment.

Actualment existeixen dues maneres de validar certificats, per CRLs o per OCSP.

Llistes de revocació de certificats (*CRL*)

L'autoritat cada cert temps (típicament, dies) pot crear una llista de revocació de certificats (o *Certificate Revocation List, CRL*) [2] amb els números de sèrie dels certificats actualment revocats. Aquesta llista va signada típicament per la CA per tal de garantir la seva procedència (tot i que aquesta pot designar un substitut per signar CRLs) i finalment és publicada en un lloc conegut, normalment una adreça web. Quan algú vol verificar si un certificat és vàlid es descarrega la CRL i comprova que el certificat a validar no figura a la llista. Aquest sistema té l'avantatge de permetre la validació de forma offline; una vegada descarregada la CRL es poden validar tots els certificats necessaris sense haver de consultar res més, fins la següent actualització de la llista. Per altra banda, però, hi ha certs desavantatges:

- Fer validacions puntuals és ineficient en temps, ja que la llista de tots els certificats revocats pot arribar a ser bastant extensa.
- Tenir coneixement de tots els certificats revocats es podria considerar una manca de confidencialitat.
- Com les llistes s'actualitzen només cada cert temps la CRL pot quedar obsoleta i no reflectir realment tots els certificats revocats.

A l'annex A.3 podem veure un exemple complet de CRL.

Protocol de l'estat de certificats online (*OCSP*)

Per adreçar aquests problemes es va crear el protocol OCSP (*Online Certificate Status Protocol*) [1]. Aquest té la forma d'un servidor que respòn a peticions sobre l'estat de

certificats individuals; per aquesta raó se'ls anomena *OCSP responders*. Una petició OCSP consisteix en poc més que un número de sèrie i la resposta és un número que identifica l'estat del certificat (vàlid, revocat, suspès, error en la petició, ...). Anàlogament al cas de les CRL, les respostes OCSP venen firmades per garantir que la resposta és fiable. Segons el RFC on es detalla el protocol, aquestes respostes no cal que siguin signades necessàriament per l'emissor del certificat: aquest pot delegar la tasca en un certificat especial emès per la ocasió. En alguns casos, el servidor OCSP pot exigir també que les peticions hagin d'estar signades, per exemple en casos de servidors de pagament.

Cal aclarir un punt important del protocol: la resposta només indica si un certificat consta com a revocat per l'autoritat o no. Si, per exemple, preguntem per un certificat no existent, el servidor ens contestarà que el certificat és vàlid. Amb això el que vol dir és que no li consta com a revocat, de la qual cosa no hem d'entendre que el certificat sigui vàlid.

Així doncs, amb aquest protocol suplim les mancances esmentades anteriorment de les CRL: el número de certificats revocats és irrelevant ja que només preguntem d'un en un, no divulguem informació d'altres certificats i les respostes estan (possiblement) més actualitzades que les llistes. Per altra banda, si hem de validar molts certificats de cop, l'ample de banda utilitzat i la càrrega de treball pel servidor pot superar a l'utilitzat amb la CRL, ja que el protocol OCSP té un *overhead* per cada petició individual.

A l'annex A.4 podem veure un exemple complet de resposta OCSP.

4.5 Segellat de temps (*timestamping*)

Un concepte que també ens convé esmentar és el de *segellat de temps*. Aquests segells són subministrats per autoritats de segellat de temps (TSA, o *TimeStamping Authorities*) i simplement ens donen la hora i data en què s'ha fet la petició, signat junt amb les dades que volem certificar. Per tant, considerant que l'autoritat sigui de confiança, podem utilitzar aquests missatges per garantir que un fet ha estat produit en una hora concreta.

En el cas de la signatura digital, es pot fer la petició a l'autoritat amb la signatura de la que volem certificar el temps i l'autoritat respòn amb el resum de la signatura que l'hem enviat, l'hora actual i tot plegat signat. D'aquesta manera podem presentar aquesta informació com a prova de que la signatura s'ha efectuat (com a màxim) a la data i hora actuals.

4.6 Signatures electròniques

Fins ara hem vist la infraestructura de clau pública i la base del seu funcionament criptogràfic, però encara ens podem preguntar quina utilitat real té tot aquest sistema.

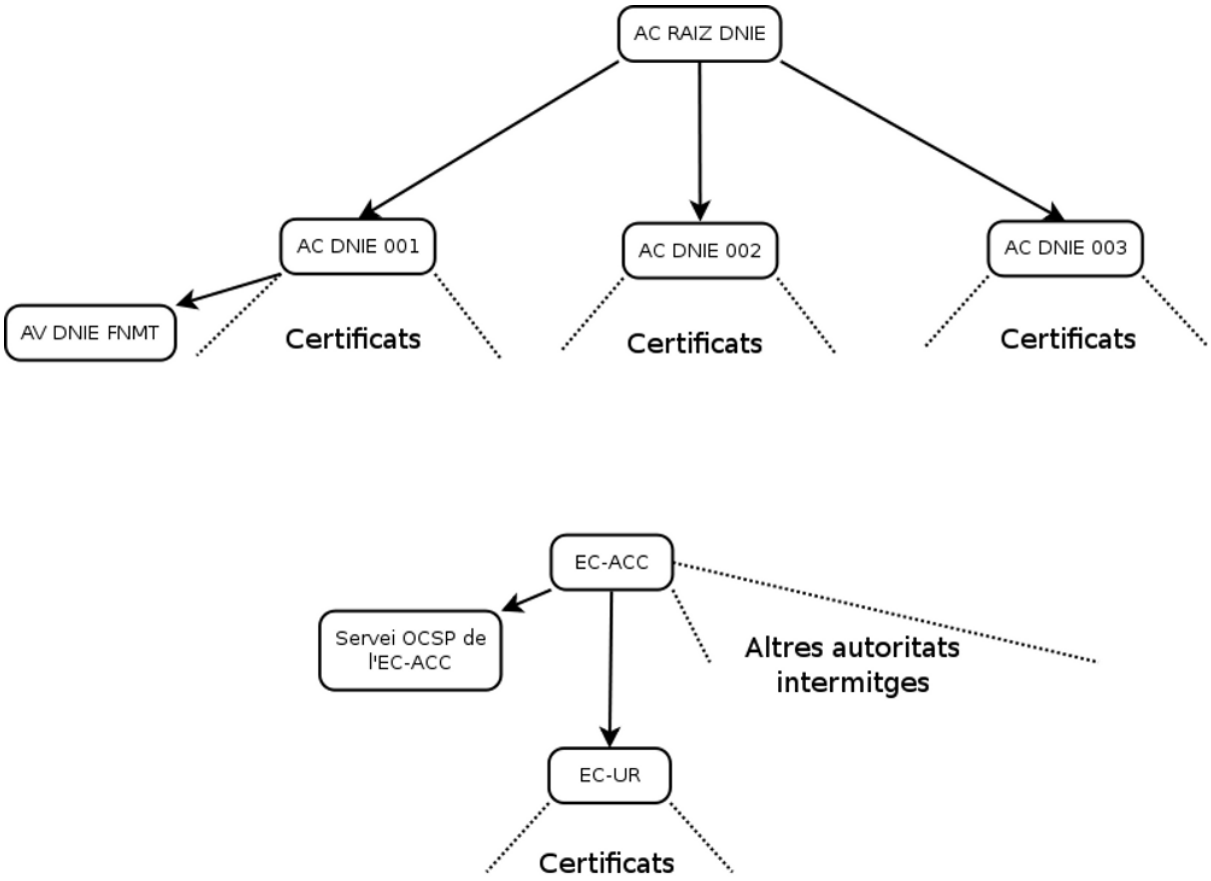

Figura 4.2: Estructura de la PKI del DNIe (a dalt) i de l'agència catalana de certificació (a sota). Es mostren també les AV (AV DNIE FNMT i Servei OCSP de l'EC-ACC).

L'ús més obvi és el xiframent de dades per una comunicació segura, però un altre ús important que utilitzarem en aquest projecte és el de les *signatures digitals*.

De forma similar a una signatura manuscrita, una signatura digital d'un document és la informació que demostra la procedència, conformitat o compromís d'un individu amb el contingut d'un document. Farem servir les signatures digitals per implementar la *signatura electrònica* en els documents de l'actual projecte. La diferència entre les dues és que aquesta última és una marca per associar un document a una persona, sense entrar en el mecanisme o implementació. A diferència de les manuscrites, la signatura digital ens garanteix tres coses:

- El document final no ha patit canvis respecte a l'original (**integritat**).
- El signant del document es pot identificar de forma unívoca (**autenticació**).
- El signant no pot negar haver signat el document a posteriori (**no-repudi**).

El primer punt ens garanteix per una banda que cap tercer ha interceptat i modificat el fitxer des de la seva signatura, així que no es pot defraudar al signant en allò signat. Però a més és una manera d'associar la signatura amb aquell document, de manera que les signatures digitals d'un mateix individu són diferents i depenen del document signat, cosa que evita la substitució de signatures.

El segon punt és potser el més important dels tres; amb una signatura electrònica podem identificar completament a l'autor de la signatura. Podem estar segurs que el certificat del signant ha estat emès per les autoritats en les que confiem i, en cas de necessitat, podem identificar a l'autor per mitjà de l'autoritat de certificació. Un problema en l'autenticació seria resultat de mancances de les AC i, per tant, se'ls hi podria demanar responsabilitats legals.

L'últim punt és important per la validesa d'una signatura, assegurant la intenció del signant. Podem garantir que si existeix la signatura aquesta ha estat creada pel propietari del certificat i per ningú més, així que no és possible al·legar el desconeixement de la signatura. Òbviament es pot donar el cas de que la clau privada es vegi compromesa i un tercer signi en nom del propietari legítim, però en aquest cas és responsabilitat de l'usuari informar a la AC perquè revoqui el certificat.

4.6.1 Funcions de *hash*

Abans de detallar el procés de la signatura digital hem de conéixer el concepte de funcions de resum o *hash*. Les funcions de *hash* són funcions matemàtiques que converteixen un conjunt de dades arbitràriament gran en un altre de mida constant (*hash*), i serveixen per obtenir una mostra representativa de qualsevol informació que vulguem tractar, de manera que podem identificar el *hash* amb les dades originals. Com és bastant obvi de la descripció anterior, aquestes funcions no són mai injectives (i.e. un mateix *hash* pot correspondre a diversos conjunts de dades, fenòmen conegut com a *col·lisió*), així que necessitem garanties addicionals per poder identificar un *hash* amb unes dades de forma segura.

Per aquestes exigències existeixen les *funcions de hash criptogràfiques*. Un bon algoritme de *hash* criptogràfic ha de complir les següents quatre propietats [26]:

Eficiència: El seu càlcul ha de ser ràpid.

Unidireccionalitat: No ha de ser viable trobar un missatge a partir d'un valor de *hash*.

Resistència dèbil a col·lisions: No ha de ser viable trobar un missatge amb el mateix *hash* que un altre missatge donat.

Resistència forta a col·lisions: No ha de ser viable trobar dos missatges amb el mateix *hash*.

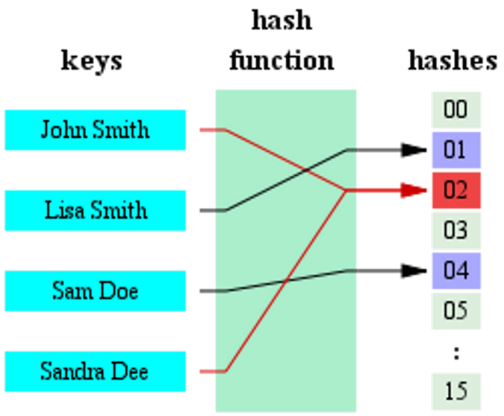

Figura 4.3: Funcionament d'una funció hash (amb una col·lisió) [25].

Exemples coneguts d'algoritmes de hash amb aquestes propietats són, actualment, la familia de SHA (com SHA-1, SHA-256, SHA-384 i SHA-512), tot i que el primer d'aquests és bastant vulnerable avui en dia.

4.6.2 Creació d'una signatura

El procés de creació d'una signatura digital es troba representat a la part superior de la figura 4.4. Consisteix en el següent:

1. Es fa el resum del document mitjançant una funció de *hash* concreta.
2. Es xifra aquest resum amb la clau privada.
3. S'envia al receptor el resum xifrat, el certificat associat a la clau privada i el document.

4.6.3 Validació d'una signatura

La validació d'una signatura digital necessita diverses fases per poder confiar-hi completament. En primer lloc s'ha de verificar que la signatura està associada realment tant al document com al certificat. Això s'aconsegueix de la següent manera:

1. Es fa el resum del document mitjançant una funció de *hash* concreta (*hash* del document).
2. Es desxifra el resum xifrat enviat per l'emissor amb la clau pública del certificat (*hash* de la signatura).

3. Si els dos *hash* anteriors són iguals, podem assegurar que la signatura ha estat creada pel propietari del certificat en aquell document.

Això no és tot; per acabar de garantir l'autenticitat de la signatura hem de comprovar que el certificat realment és vàlid, mitjançant el procés de validació de certificats que hem detallat més amunt. Si el certificat és vàlid, podem confirmar els tres punts que una signatura electrònica garanteix es mantenen.

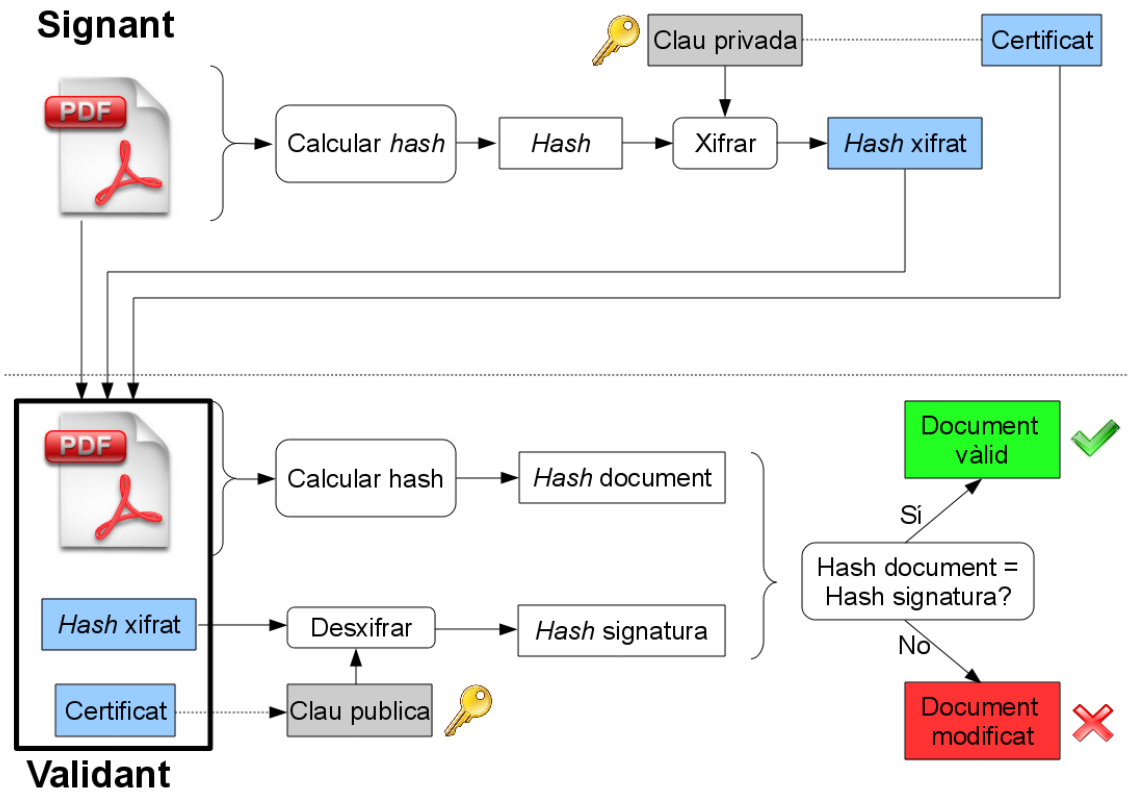

Figura 4.4: Procés de creació i validació d'una signatura digital.

4.6.4 Consideracions legals

Clarament una finalitat important de la signatura electrònica és la vinculació legal d'un signant amb el document signat. Per aconseguir això els estaments jurídics han d'establir la validesa legal d'aquestes signatures. Sense entrar en massa detall, existeixen dues fonts principals de lleis relatives a aquest tema: la directiva 1999/93/CE del Parlament Europeu [16] i la llei 59/2003 espanyola [14].

A efectes pràctics, podem veure que existeixen tres tipus de signatures electròniques a nivell legal:

Simple: Permet identificar el signant.

Avançada: Signatura electrònica simple que garanteix la integritat del document i la seguretat de la clau privada (mitjançant dispositius segurs de creació de signatura, com les targetes criptogràfiques).

Reconeguda: Signatura electrònica avançada consistent en un certificat reconegut (expedit amb verificació presencial del titular).

El tipus de signatura amb validesa legal és aquest últim. Veiem que en el nostre cas complim tots els prerrequisits: garantim la integritat del document (amb els algoritmes presentats més amunt), utilitzarem dispositius segurs de creació de signatura (carnet UPC i DNI electrònic) i els certificats són reconeguts (expedits de forma presencial). Així doncs, els documents signats mitjançant les nostres targetes seran vàlids legalment.

4.7 Targetes criptogràfiques

Tot i que parlarem de les característiques tècniques de les targetes criptogràfiques al següent capítol, no està de més saber com funcionen i com canvia el procediment de la signatura digital en aquest cas.

Una *targeta criptogràfica* (o targeta intel·ligent, *smartcard*) és una targeta amb un circuit integrat que conté un certificat digital i la seva clau secreta. On radica la seva utilitat, però, és en el fet que la clau secreta no es pot extreure de la targeta; ni tan sols el propi usuari la pot saber. En el seu lloc incorpora en el circuit integrat els algoritmes necessaris per xifrar i signar dades. D'aquesta manera l'usuari té més control sobre la seva clau, ja que només la pot utilitzar la persona que tingui la targeta i que sàpigui el PIN de la targeta.

Aquesta manera de procedir fa que s'alteri lleugerament el procés de signatura. Com veiem a la figura 4.5, el procès és idèntic a l'anterior, però per xifrar el *hash* aquest s'envia a la targeta, que respòn amb el criptograma adequat. Per altra banda, el certificat també l'hem d'extreure de la targeta per enviar-lo al validant, que realitza el procés de validació de forma idèntica al cas normal.

A continuació veurem la infraestructura dels dos tipus de certificats que utilitzarem: el carnet UPC i el DNI electrònic.

4.7.1 Carnet UPC

Els certificats dels carnets de la UPC provenen de l'entitat intermitja EC-UR (*Entitat de Certificació - Universitat i Recerca*) i aquesta de l'entitat arrel, l'Agència Catalana de Certificació (EC-ACC). Aquesta última emet cada cert temps una CRL de les entitats

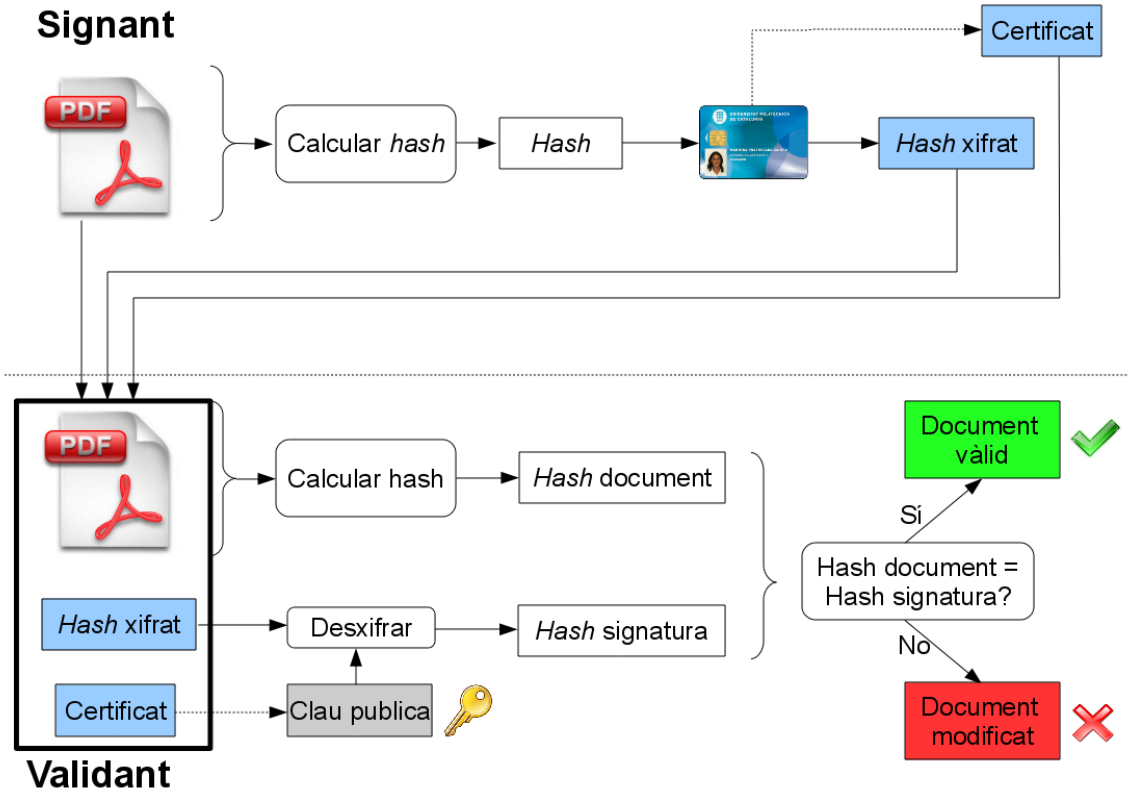

Figura 4.5: Procés de creació i validació d'una signatura digital amb una targeta criptogràfica.

intermitges, mentre que la primera permet també la validació dels seus certificats emesos amb OCSP. Les respostes del protocol venen signades per un certificat anomenat "Servei OCSP del EC-ACC", expedit per l'autoritat arrel.

Sobre els propis certificats, punts a destacar són que la clau que utilitzen és de 1024 bits i només poden signar conjunts de dades de 160 bits (resultat de l'algoritme SHA-1).

4.7.2 DNI Electrònic

La infraestructura del DNI electrònic és lleugerament més complexa. Partim de l'autoritat arrel (*AC DNIE RAIZ*), que té tres autoritats intermitges (*AC DNIE 001*, *AC DNIE 002* i *AC DNIE 003*). Finalment, els certificats del DNIe són expedits per una d'aquestes tres autoritats. En quant a la validació, cap autoritat intermitja publica CRLs (decisió lògica ja que podrien ser realment grans), així que només és possible la validació mitjançant OCSP. En aquest cas, les respostes venen signades pel certificat "AV DNIE FNMT", expedit per AC DNIE 001, fet que veurem que ens donarà problemes més endavant.

A diferència de l'anterior, la seva clau és de 2048 bits i és compatible en SHA-256 i (per compatibilitat) SHA-1.

4.8 Comunicació xifrada per TLS/SSL

Apart de la signatura electrònica, però també consistent en criptografia de clau pública, utilitzarem el protocol TLS (*Transport Layer Security*, successor de SSL, *Secure Sockets Layer* [36]) per comunicació segura via web. Aquest protocol és utilitzat extensament en entorns web per assegurar la confidencialitat de la comunicació entre clients i servidors, utilitzant un xifrat de claus provinents de certificats digitals. El seu funcionament resumit és el següent:

1. El client contacta amb el servidor, demanant una connexió segura i presentant una llista de paràmetres possibles (algoritmes de *hash* i xifrat) que pot utilitzar.

2. El servidor escull el primer conjunt de paràmetres presentats (normalment el més segur) i li fa saber al client, enviant-li també el seu certificat digital.

3. El client autentifica el certificat del servidor.

4. Si el certificat és vàlid, el client confirma que hi confia enviant-li un missatge xifrat amb la clau pública del servidor. Junt amb aquesta informació, se li envia també la clau pública del client.

5. Anàlogament al cas anterior, el servidor confirma que ha rebut la clau del client enviant-li un altre missatge xifrat amb aquesta.

6. A partir d'aquest moment els dos es comuniquen de forma xifrada.

Veiem que en el protocol s'estableixen dos garanties: autenticitat (confiança amb els certificats digitals) i confidencialitat (xifrat de la comunicació). Aquesta última es compleix sempre, però amb la primera hi ha diversos casos.

En teoria, la part d'autenticitat del protocol és opcional; la part important és que dos parts s'intercanviïn claus públiques per tal de comunicar-se de forma segura. Tot i això, no té massa sentit establir una comunicació segura amb algú desconegut, així que com a mínim es sol exigir l'autenticació del servidor. Addicionalment també es pot demanar l'autenticació del client, exigint que s'identifiqui i utilitzar la clau pública del seu certificat. En molts casos, però, no és necessari certificar que el client és conegut, així que aquest sol generar un parell de claus per utilitzar en cada nova connexió xifrada.

4.9 Documents PDF

Ja que hem de tractar documents PDF de forma extesa en el projecte, és necessàri tenir la informació suficient com per poder treballar amb ells. En el nostre cas, en tindrem suficient sabent-ne una mica d'història i de la seva estructura, sobretot de cara a l'hora d'afegir la signatura.

4.9.1 Història

El format PDF (*Portable Document Format*) va ser creat al 1993 per Adobe Systems per representar documents d'una forma independent de la plataforma *hardware* o *software*. Així com el format *postscript* és l'estàndard més extès per representar documents de forma optimitzada per la seva impresió, el format PDF s'ha tornat en el més comú pels documents optimitzats per la seva visualització directa (en monitors i pantalles). Tot i començar com a format propietari de Adobe, el juliol de 2008 es va lliberar l'especificació completa [23] i va passar a ser un format obert.

Hi ha hagut diverses versions del format, sent la 1.3 (publicada l'any 2000) [35] la primera en donar suport per signatures digitals.

4.9.2 Estructura

Un fitxer PDF està estructurat en quatre grans parts: la capçalera (*header*), el cos (*body*), la taula de referències creuades (*cross reference table*) i la cua (*trailer*).

La capçalera només indica que el fitxer és un document PDF i quina versió del format implementa.

El cos és el nucli del fitxer. Aquesta part és un repositori d'objectes interns del format. N'hi ha diversos tipus: booleans, enters, reals, cadenes de caràcters, *arrays*, diccionaris, *streams* i d'altres. Aquests estan relacionats entre si, de manera que defineixen una jerarquia en forma d'arbre: un objecte arrel format per altres objectes que en poden contenir d'altres, etc. Els objectes poden tenir identificadors i ser directes (valors) o indirectes (referències).

La taula de referències creuades és una taula de punters, a on s'indica a quina posició del fitxer es troba cada objecte amb un identificador donat.

Finalment, a la cua del fitxer es diu on comença la taula de referències creuades i, en segon lloc, es dona informació general del fitxer (com l'identificador de l'objecte arrel, l'objecte amb la metainformació del fitxer i d'altres).

Una característica important del format PDF és l'extensibilitat del seu contingut. Quan modifiquem el contingut del fitxer podem fer-ho mantenint el contingut anterior

simultàniament, i només afegint o modificant els objectes necessaris. Això és possible afegint al final d'un fitxer un nou cos, taula de referències creuades i cua per cada nova versió del fitxer. Al nou cos podem afegir nous objectes, a la nova taula podem redefinir l'adreça d'un objecte (per exemple, canviant l'adreça d'un objecte antic per un definit al nou cos) i finalment a la nova cua s'actualitza la informació del fitxer, incloent un apuntador a la cua anterior per poder accedir sempre a versions anteriors.

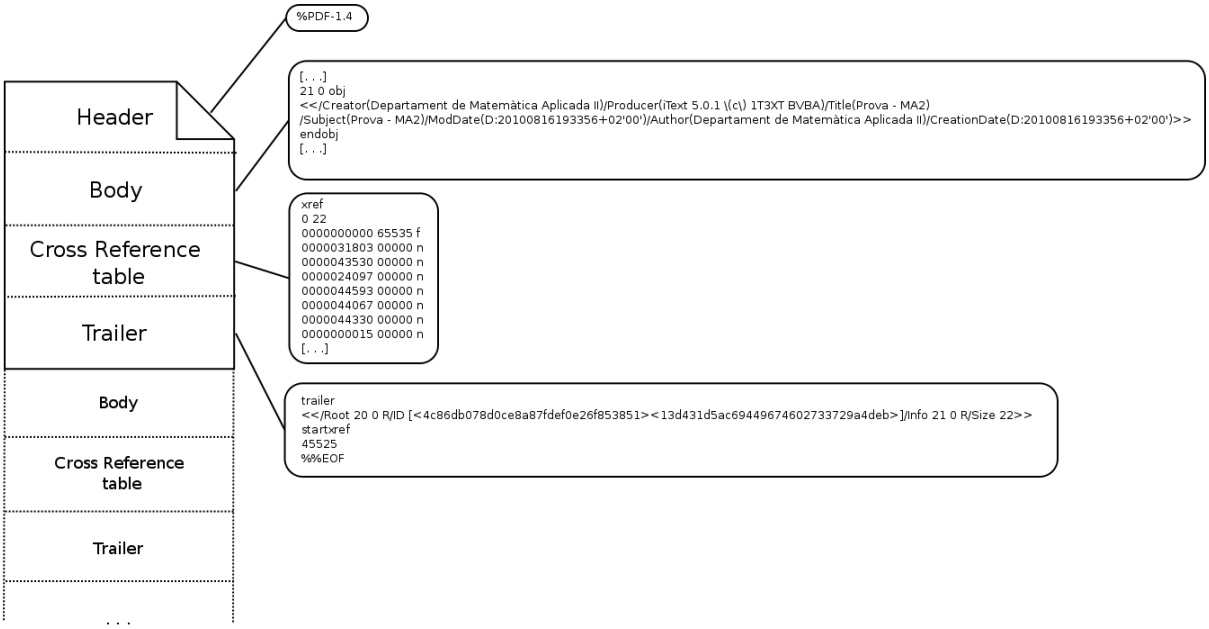

Figura 4.6: Estructura d'un PDF amb exemples del contingut.

4.9.3 Signatures digitals en un PDF

Com ja sabem com funciona el procés de signatura digital, només parlarem dels punts a considerar en el cas particular de la signatura de PDFs.

Les signatures digitals en un fitxer PDF estan implementades a partir dels camps de signatura (*signature fields*). Aquests són un dels tipus dels anomenats *Acroforms*, elements del document com ara camps de text, botons i altres components interactius. Cada camp de signatura s'identifica per un nom i poden estar associats amb un objecte de signatura. Per altra banda, cada objecte de signatura conté la informació sobre la raó (per exemple, en el nostre cas, el càrrec signant) i el lloc de firma, a més de la pròpia signatura. Per tant, signar consisteix en crear un nou objecte de signatura i associar-lo al camp adequat.

Vist això, el primer punt amb el qual hem d'anar en compte és què és el que estem signant. Donat que la signatura anirà integrada dins del mateix fitxer, no podem signar-lo tot, ja que a l'afegir informació estariem modificant el document i, per tant, invalidant

la pròpia signatura. Per solucionar aquest problema farem ús de l'extensibilitat que hem comentat:

1. En primer lloc afegirem una nova versió al document (nou cos, taula i cua). En el cos afegirem un nou objecte que representarà la signatura, reservant espai amb una estimació del que ocuparà la signatura final.
2. Fem la signatura del contingut de tot el fitxer (inclosos els components afegits en el pas anterior), amb l'excepció de l'espai reservat a la signatura.
3. Una vegada afegida tota la informació necessària (que veurem al capítol de disseny), incorporem la signatura a l'espai reservat.

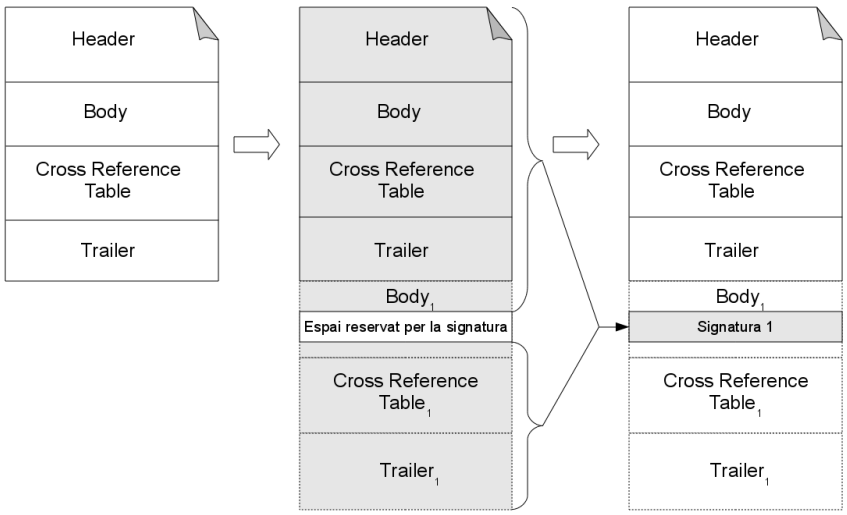

Figura 4.7: Procés (estructural) de signatura d'un PDF.

El cas de l'estimació de l'espai per la signatura pot donar problemes: si és massa petita la signatura no cap i no es pot signar, però si és massa el gran el fitxer acaba ocupant molt espai, cosa que també volem evitar. Cal notar, però, que aquesta estimació és necessària. No podem determinar amb exactitud quan ocuparà tota la signatura sencera, ja que fins i tot signatures fetes en intervals de temps molt petits poden diferir en algún detall (e.g. una nova entrada a una CRL).

Una altra conseqüència d'aquest mètode és que només permet signatures seqüències, i.e. cada individu signa el document i la signatura dels signants anteriors. Resulta impossible la signatura en paral·lel (i.e. cada individus signa el document independentment dels altres signants) sense comprometre la seguretat del sistema. Si cada modificació del

fitxer no fos signada, un atacant podria canviar la referència d'un camp de signatura a una altra diferent, i el sistema no ho podria detectar.

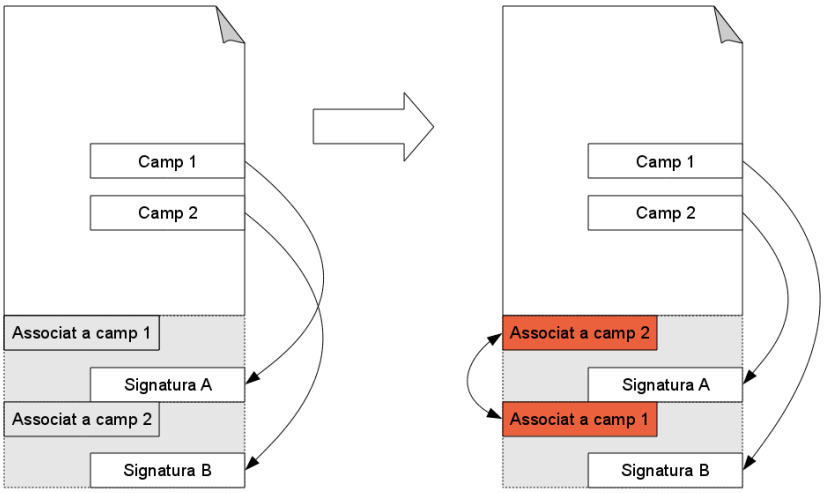

Figura 4.8: Exemple d'atac al procés de signatura de PDF en paral·lel. Si les modificacions del fitxer no estan signades, un atacant podria intercanviar signatures en un document.

Aquest punt implica, al seu torn, una petita subtilesa que haurem de tenir en compte amb els documents signats: cada firma pot estar relacionada amb un document diferent. Si, com hem dit, cada signatura exigeix afegir nou contingut al document (la pròpia firma) res ens impedeix afegir altre nou contingut (o modificar l'existent). Això no és un problema, però, ja que cadascú signa una diferent versió del fitxer i no s'invalida res. Només hem de tenir en compte que l'aparença de la versió final del document pot no ser la que han vist tots els signants, per la qual cosa si volem estar segurs de que ha signat cadascú haurem de veure la versió de cada signatura (figura 4.9).

4.10 Estàndards relacionats amb la signatura digital

Per concluir el capítol, farem una ullada ràpida als diversos estàndards criptogràfics amb relació a la signatura digital. Començarem pels més generals i anirem especificant fins a arribar als tipus de signatura propis del format PDF.

4.10.1 *Abstract Syntax Notation One* (ASN.1)

ASN.1 és una notació utilitzada per moltes branques de telecomunicacions i estudi de xarxes, que permet de forma flexible i independent de la tecnologia representar estructures

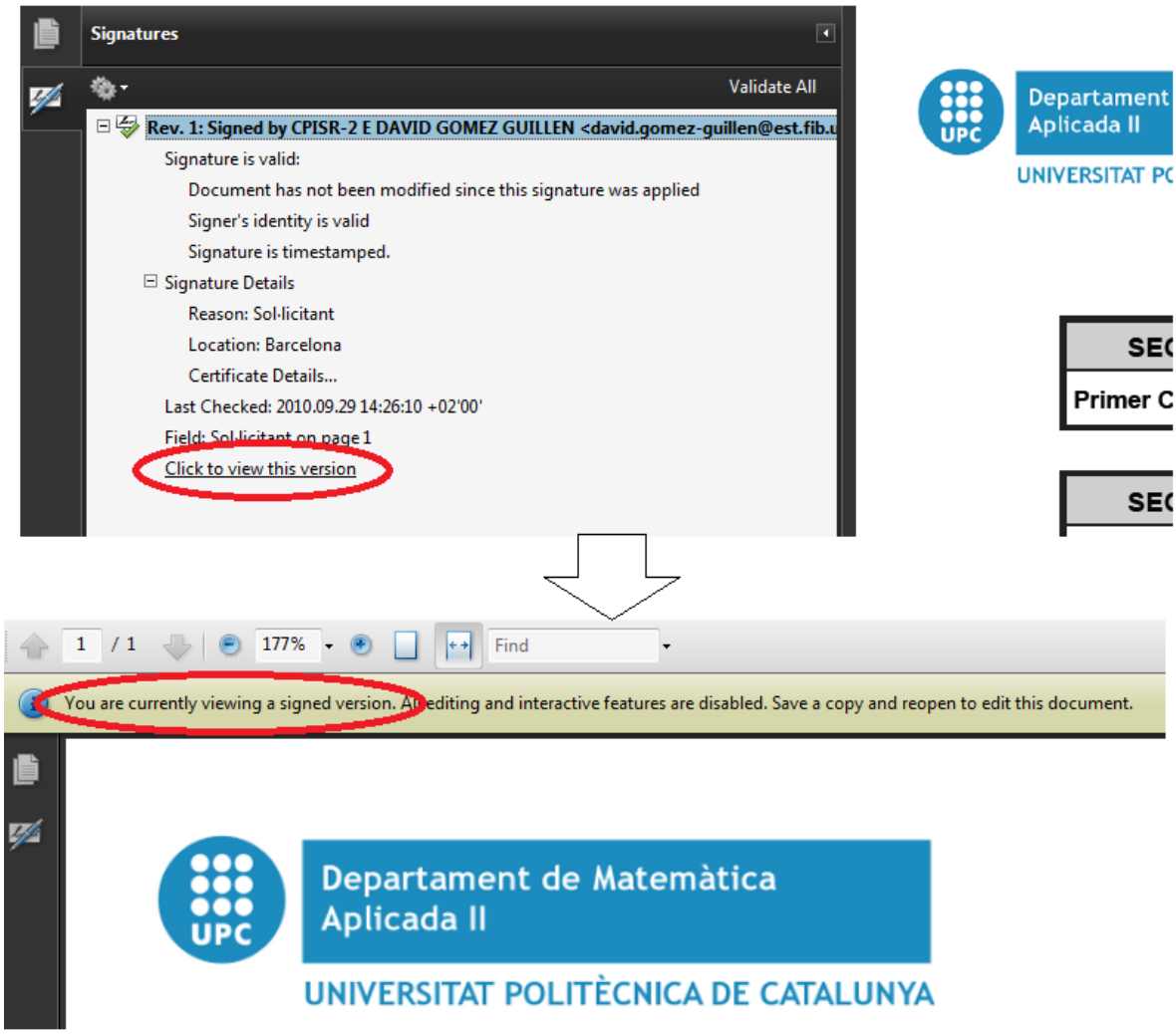

Figura 4.9: Visualització de versions signades amb Acrobat Reader.

de dades per la seva codificació, transmissió i decodificació. Va ser especificat per la Unió Internacional de Telecomunicacions (*ITU*) a l'estàndard X.683 [46], i bàsicament és un conjunt de regles que permet definir aquestes estructures de forma no ambigua. Podem veure exemples d'informació en notació ASN.1 a l'annex A.

Com el seu nom indica, es tracta d'una notació *abstracta*, que no té establerta una codificació única. Per fer-ho necessitem altres regles de codificació, com BER o DER.

4.10.2 *Basic Encoding Rules* i *Distinguished Encoding Rules* (BER i DER)

El format de codificació BER, així com les seves variacions (com DER) formen part de l'estàndard X.690 de del ITU. Aquest ens defineix les regles per la codificació d'estructures especificades amb ASN.1, permetent-nos la transferència d'estructures de dades.

Aquest es basa en trames <*tipus, longitud, valor*>, on tipus és un dels tipus predefinits. Un problema de les regles BER, però, és que hi pot haver diverses maneres de codificar una mateixa estructura, la qual cosa pot provocar problemes a l'hora de, per exemple, signar. Per evitar aquest problema existeix un subconjunt d'aquestes regles anomenat DER, que defineix una única manera de codificar unes dades. Per exemple, el booleà cert en BER es pot representar com a qualsevol byte diferent de zero, mentre que DER ho restringeix al valor 255.

4.10.3 *Cryptographic Message Syntax* (CMS)

L'estàndard CMS fa servir els dos anteriors i, com el seu nom indica, és la base per definir missatges criptogràfics com criptogrames o signatures digitals. Aquest està definit en la seva versió més recent (septembre del 2009) al RFC 5652 [4], i està basat en el protocol PKCS#7.

Aquesta sintaxi està construida al voltant dels sistemes de criptografia de clau pública i descriu camps necessaris pel reconeixement d'una signatura (certificat del signant, hora de la signatura, algoritmes utilitzats, ...).

4.10.4 *CMS Advanced Electronic Signature* (CAdES)

Aquest estàndard (definit al RFC 5126 [3]) extèn l'estàndard anterior per otorgar-li característiques compatibles amb la signatura electrònica (reconeguda) definida segons la llei. En la seva forma més bàsica això inclou tres nous atributs: el tipus del contingut signat, el *hash* del contingut i el certificat signant. Hi ha, però, diferents perfils de CAdES, cadascún complementant a l'anterior:

CAdES-BES (*basic electronic signature*): Perfil bàsic per satisfer els requeriments de signatura electrònica avançada.

CAdES-T (*timestamp*): Addició d'un segell de temps per evitar la repudiació de signatures.

CAdES-C (*complete*): Addició de referències a informació de validació per verificació futura.

CAdES-X (*extended*): Addició de segells de temps a les referències anteriors.

CAdES-X-L (*extended long-term*): Addició de la informació de validació (completa) per permetre la validació en el futur, una vegada caduquin els certificats.

CAdES-A (*archival*): Addició de la possibilitat de segellat de temps periòdic.

Aquest estàndard s'especialitza en dos vessants: XAdES i PAdES. Mentre que CAdES permet la signatura de qualsevol contingut digital, aquests estan preparats per la signatura de fitxers XML i PDF, respectivament.

Veurem més detall de la implementació concreta d'una signatura CAdES al capítol de disseny.

4.10.5 *PDF Advanced Electronic Signature* (PAdES)

Com hem dit, PAdES [38] (publicat pel ETSI (*European Telecommunications Standards Institute*) utilitza una signatura CAdES amb algunes modificacions per integrar-la en fitxers PDF. Aquest està dissenyat per ser compatible amb el procés de signatura electrònica natiu definit per Adobe a la ISO 32000-1, però afegint-hi restriccions per fer-lo compatible també amb les directives legals europees.

Una vegada definida la signatura (codificada en DER), aquesta s'ha d'integrar al PDF tal i com hem dit a la secció sobre l'estructura de PDFs. Un punt important, però, és que la pròpia signatura ha d'estar integrada com una cadena de caràcters hexadecimals (i.e. el byte 0xAB s'ha de posar com la cadena o *string* "AB"). Aquesta representació assegura que no hi haurà problemes en la transmissió de la signatura per un canal de només text (on pot alterar-se el bit de més pes de cada byte), però a canvi multipliquem per dos la mida de la signatura (la cadena "AB" ocupa dos bytes).

L'estàndard defineix cinc perfils :

PAdES-CMS (*Basic*): Perfil bàsic amb signatura CMS incrustada. No compleix els estàndards europeus.

PAdES-BES (*Enhanced*): Modificació de la signatura a CAdES-BES per complir estàndards.

PAdES-EPES (*Explicit Policy Electronic Signature*): Addició de política de certificació explícita. Només té sentit si s'utilitzen polítiques especials (e.g. eFacturació).

PAdES-LTV (*Long Term Validation*): Addició de la possibilitat de segellat de temps periòdic.

PAdES-XML: Perfil per signar contingut XML dins d'un document PDF.

Per implementar al nostre sistema volem escollir el perfil més complet possible aplicable al nostre cas (PAdES-LTV). Aquest perfil ens permetria no només actualitzar l'estat de la signatura mitjançant segells de temps periòdics, sinó també prolongar la vida útil de la signatura.

Aquest és el gran problema de les signatures digitals en definitiva: a diferència de les manuscrites, les primeres tard o d'hora expiren. En primer lloc poden expirar quan el certificat signant expiri, però això es pot solucionar afegint la informació de revocació necessària (certificats, CRLs i respostes OCSP) dins la signatura. Aquesta demostrarà que, quan es va fer la signatura, aquell certificat era vàlid, tot i que actualment pugui no ser-ho. Aquest mètode també es pot utilitzar per demostrar la validesa de les autoritats intermitges, però no podem aplicar-lo a l'autoritat arrel (hi confiem incondicionalment, no podem obtenir proves de la seva autenticitat).

Així doncs, afegint informació de revocació podem allargar la vida útil d'una signatura des de la vida del certificat signant (uns pocs anys) fins la vida de l'autoritat arrel (dècades). Però, tot i així, acabarà expirant.

4.10.6 PAdES-LTV

Per aquest motiu és especialment interessat el quart perfil presentat, PAdES-LTV [32]. Aquest ofereix dues estructures de dades noves al fitxer PDF que resulten útils pels propòsits de la signatura electrònica.

En primer lloc s'afegeix l'anomenat DSS (*Document Security Store*). Aquest és un repositori d'informació necessària per la validació de les signatures digitals del document, i està implementat com un diccionari consistent en llistes de certificats, CRLs i respostes OCSP. També té la llista de signatures del document i quina informació, de les llistes anteriors, necessita per la seva validació.

El DSS ens proporciona dos grans avantatges: per una banda tenim un repositori centralitzat d'informació, el qual vol dir que evitarem redundàncies (per exemple, els certificats de les autoritats si aquests són els mateixos). A més, el DSS també permet un accés directe a la informació necessària per la validació, sense haver de buscar i interpretar tots els recursos presents.

En segon lloc es crea el concepte de segell de temps del document (*document timestamp*). Aquest és un segell de temps estàndard, fet a partir del document sencer i que certifica que les signatures del document són correctes. Aquest segell es pot anar renovant per tal d'evitar que la seguretat criptogràfica de la clau utilitzada sigui trencada amb el pas del temps.

Malauradament, tot i els avantatges d'aquest perfil, l'hem de desestimar per raons pràctiques que ens trobarem i explicarem amb més detall en etapes posteriors (implementació i proves).

Capítol 5

Disseny

Tenint l'especificació completa de cada subsistema trobat a l'etapa d'anàlisi, en aquest capítol investigarem quines tecnologies són les adients per solucionar les nostres necessitats.

Primerament definirem com serà la interfície de cara a l'usuari mitjançant el medi que hem escollit als requeriments (web). Després, concretarem l'estructura interna i el funcionament dels diversos components tenint en compte la seva especificació.

5.1 Interfície del sistema

Com s'ha dit des de bon principi en els requeriments, l'accés al sistema s'ha de fer via web, per tots els usuaris del sistema. Això simplifica el disseny de la interfície, que serà l'únic punt d'accés al sistema des de l'exterior.

Tenint en compte les funcionalitats que hem d'implementar, hem de procurar també mantenir els requeriments no funcionals principals sota els quals treballem: seguretat, usabilitat i simplicitat. El primer l'hem de tenir en compte en diversos punts:

- Definir apartats (i permisos) per accedir a les diferents funcionalitats (restringides als actors especificats a l'especificació).
- Definir permisos per l'accés als fitxers PDF (permís només pel gestor, els càrrecs adequats i l'autor de la sol·licitud).
- Garantir confidencialitat mitjançant accés web segur (SSL) i una política de certificats vàlida (subministrada per la UPC).

En quant els dos últims requeriments, la usabilitat i simplicitat, intentarem distribuir la web de la forma més intuitiva possible, que, en general, també sol ser la més simple. Amb aquests punts elaborarem un *mapa navegacional*.

5.1.1 Mapa navegacional

Un mapa navegacional és el diagrama que descriu quines parts de la interfície (en el nostre cas quines pàgines web) es mostren a l'usuari i quines són les transicions entre elles i com es produeixen. A l'hora de decidir quantes parts (pantalles) necessitem hem d'escollir un nombre suficientment baix com per permetre un flux de navegació mínim (pocs "clicks" per accedir a les funcionalitats) però suficientment alt com per mantenir la complexitat de cada pantalla a un nivell acceptable (cada pàgina estigui ben cohesionada). La versió final (tercera iteració) del mapa navegacional és el presentat a la figura 5.1.

En aquest esquema navegacional podem veure totes les parts de la interfície del sistema (que anomenarem *pàgines* a partir d'ara). Inicialment ens trobarem amb la pàgina d'autenticació de l'usuari. Des d'aquí l'usuari només podrà autenticar-se o realitzar sol·licituds anònimes si n'hi ha. Una vegada autenticat a l'usuari se li dona accés a l'apartat de sol·licitant autenticat, així com als dels rols adequats: del seu càrrec (si en pertany a algún) o del gestor (si consta com a tal). En cada secció hi ha la pàgina principal de l'apartat (remarcat amb línies més gruixudes a la figura) i des de qualsevol pàgina es pot desconnectar del sistema i tornar a la pàgina d'autenticació.

A la secció del sol·licitant tenim la web on omplir formularis (on finalment es confirmen i possiblement signen) i on el sol·licitant pot veure les sol·licituds fetes anteriorment per ell mateix.

Amb els càrrecs disposem del mateix conjunt de pàgines per cada càrrec (representats a la figura per 1 .. n), només canviant la informació a la que accedeix cadascún. Tenim la visualització de sol·licituds pendents (per la seva signatura) i de totes les sol·licituds (dels formularis a on tinguin accés). Finalment també tenim una pàgina apart per especificar la raó del rebuig d'una sol·licitud abans de confirmar-la.

Finalment, separem els apartats del gestor en dos: càrrecs i formularis. En el primer només hi ha la pàgina principal, ja que només hem de permetre el canvi del nom i l'edició dels seus membres, cosa que farem des de la mateixa pàgina. En la gestió de formularis trobem per una banda la web per visualitzar qualsevol sol·licitud de qualsevol formulari i per altra l'edició del formulari.

Per implementar aquesta separació de rols (i la separació de l'edició de formularis) pensarem en implementar-ho com una barra de navegació que estigui present en tot moment, tal com es veu en la captura d'un prototip feta a la figura 5.2.

Existeixen alternatives respecte la interfície que s'havien plantejat anteriorment (en prèvies iteracions):

- En un principi s'havia pensat demanar autenticació per cada funcionalitat que la necessités, però donat que n'hi ha diverses el pas més lògic era globalitzar l'autenticació a nivell de sistema.
- En la primera iteració del desenvolupament del projecte l'edició de formularis es

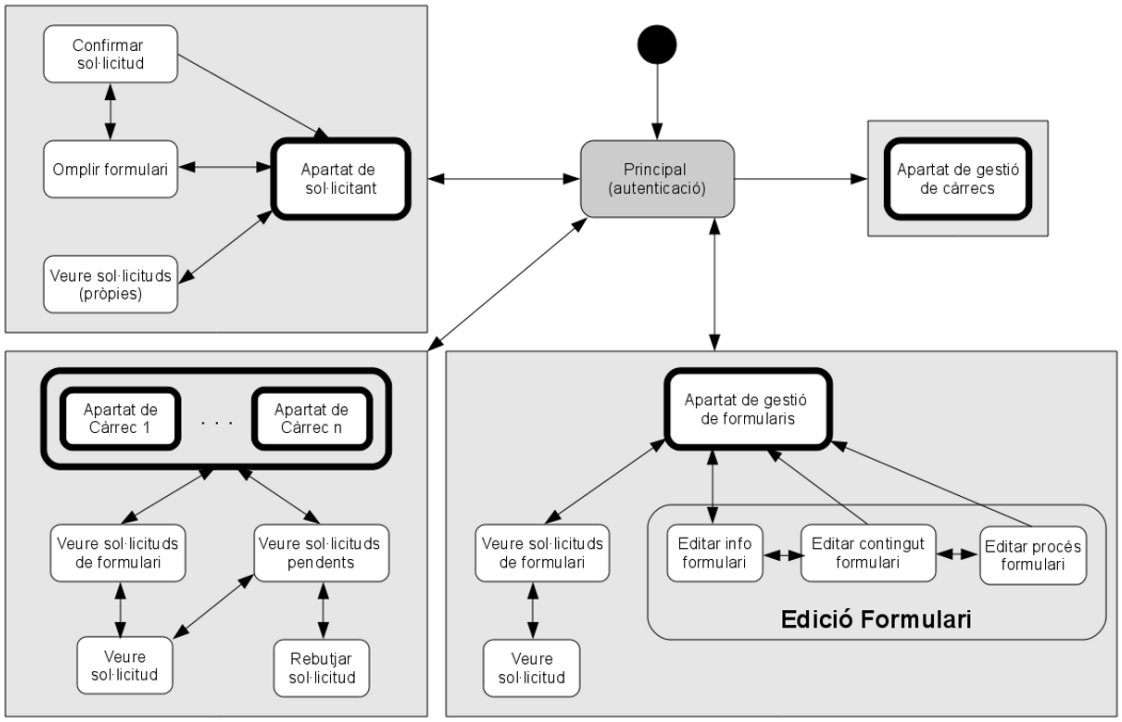

Figura 5.1: Diagrama de la navegació del sistema per part de l'usuari. Inicialment es troba la web d'autenticació, d'on es poden accedir als altres quatre grans apartats que es corresponen als actors del sistema (separant la gestió de formularis i la gestió de càrrecs).

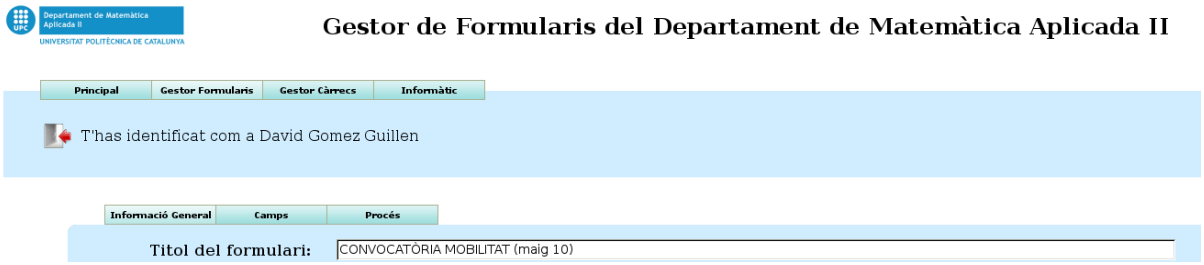

Figura 5.2: Mostra de la barra de navegació principal de la web (on Informàtic és un càrrec temporal de prova), junt amb l'edició de formulari.

trobava en una sola pàgina, però degut a la quantitat d'informació que es mostra es va optar a partir-ho en tres subpàgines: l'edició de informació general, dels continguts i del procés administratiu.

- S'havia considerat protegir l'accés com a gestor mitjançant *password*, però això exigia introduir-lo cada vegada o, si el navegador se'n recordava, acceptar el diàleg. Donat que no aquest sistema no aportava cap seguretat real addicional es va substituir pel sistema de permisos mitjançant la configuració externa (veure més avall).

- Es va mirar la possibilitat de realitzar la comunicació segura (SSL) mitjançant el certificat de la targeta criptogràfica (si s'havia utilitzat). Proves fetes durant l'etapa d'implementació van mostrar que la targeta trigava uns segons en cada petició al servidor web, així que es va rebutjar la idea per ineficiència. A més, no era realment necessari; només fa falta validar el certificat per autenticar l'usuari; no fa falta que la comunicació sigui xifrada amb la clau pública de l'usuari. Finalment, el fet de que l'encriptació amb certificat del client fos opcional feia que tot plegat tingués poc sentit; si donem la possibilitat de no seguir un protocol de seguretat aquest no serveix per res.

5.1.2 Configuració externa a la interfície

Com ja hem comentat prèviament en el document, existeixen dos tipus de dades que són massa sensibles i, alhora, poc útils com per permetre-hi l'accés directe a l'usuari: la llista de càrrecs i els usuaris amb permisos de gestor.

Ja hem dit que la llista de càrrecs és una informació que canvia molt rarament, fins al punt en que es podria considerar una constant del sistema. Permetre l'accés a un usuari podria desembocar en l'esborrat d'un càrrec i la inconsistència del sistema, com sol·licituds "penjades", cosa que no ens podem permetre en un sistema administratiu. Així doncs, els considerarem una constant, però un dels components (interns) de la web serà un fitxer de configuració amb la llista de càrrecs, per la seva fàcil modificació.

Seguirem el mateix procediment per la llista dels usuaris amb permisos de gestor, però per raons diferents. Ja que aquests són els encarregats d'administrar els usuaris, hem de fer més difícil que s'arribi a una situació de incapacitat de gestió (si l'únic gestió disponible del sistema es fes fora com a gestor a si mateix). De forma similar al cas anterior, aquesta informació tampoc canvia molt sovint; el gestor típicament serà el cap d'administració que canvia amb poca freqüència (anys). Per tant, tot i que els membres dels càrrecs normals es podran eliminar i afegir fàcilment via web, només permetrem la modificació dels gestors mitjançant un altre fitxer de configuració amb la llista d'aquests.

5.2 Arquitectura global de disseny

Partint de l'arquitectura global d'anàlisi presentada en el seu capítol, arribem finalment a l'arquitectura de disseny de la figura 5.3.

Els components separats indiquen que es poden desplegar en diferents màquines físiques, un avantatge important de definir l'arquitectura de forma modular. Tot i això, com comentarem en les properes seccions, podem veure que alguns subsistemes han estat integrats dins d'altres: el sistema de signatura electrònica ha passat ha estar dins del navegador (ja que els applets s'executen d'aquesta manera) i diversos altres han estat

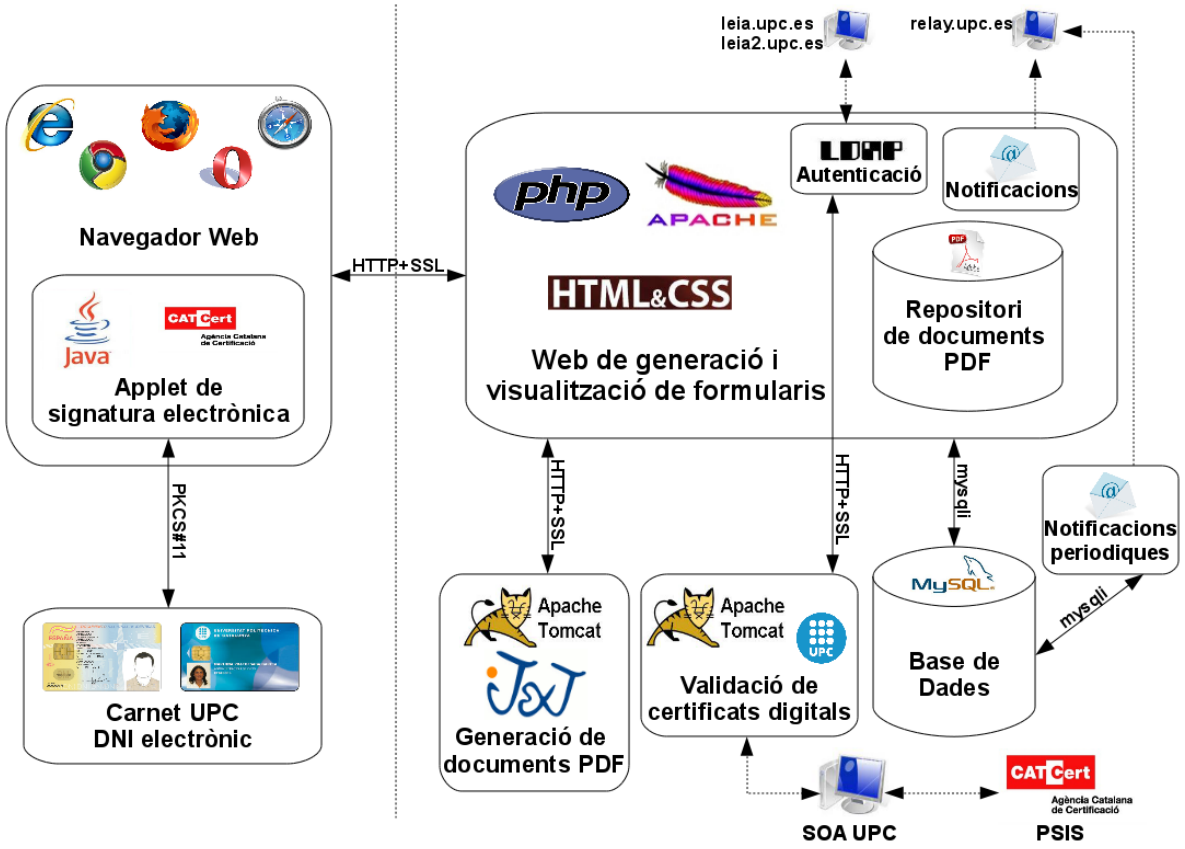

Figura 5.3: Arquitectura global de disseny.

integrats dins del sistema web (repositori de documents PDF, autenticació i notificacions). Explicarem el perquè de la decisió al seu apartat corresponent.

5.3 Web de formularis

Comencem pel nucli del sistema, la web de formularis. La tecnologia a utilitzar per aquest ja ens venia donada pels requeriments, un sistema web. Utilitzarem el servidor web Apache, software de codi obert, i PHP com a llenguatge de programació per generar contingut dinàmicament. També separarem el contingut de la presentació a nivell de codi mitjançant l'aplicació de *Cascading Style Sheets*, o CSS. Aquestes fulles d'estil ens permeten assignar una aparença als components HTML en un fitxer apart, separant i unificant la presentació.

A continuació dissenyarem l'estructura interna de la web.

5.3.1 Arquitectura interna

Per detallar el disseny hem de pensar en la forma d'actuar del protocol web (HTTP): el client fa una petició al servidor, aquest processa la petició i li respòn. Per tant cada "pas" que realitzem en el sistema estarem fent una petició. Recalquem aquest punt perquè donat això potser no resulta tant convenient una divisió típica en tres capes (presentació, lògica i dades), al menys no de la forma tradicional.

Podem dividir els tipus de petició en dos grans tipus: les peticions per una nova pàgina (e.g. enllaç a la pàgina d'un formulari) i les que envien dades per ser processades pel sistema (e.g. confirmar una sol·licitud). La primera classe de comunicació retorna el codi HTML de la pàgina apropiada, el que es pot considerar un procés de presentació, mentre que la segona processa les dades, provoca un canvi en el sistema i finalment es redirecciona a l'usuari a una altra pàgina (pertanyent per part a la primera classe). Així doncs, dividirem la capa més superficial del sistema en una part de presentació i una lògica. També dividirem la part de gestió de la resta (per poder afegir opcions de seguretat de la gestió en el futur, si es vol). Finalment, aquesta capa accedirà a les dades necessàries mitjançant un middleware d'accés.

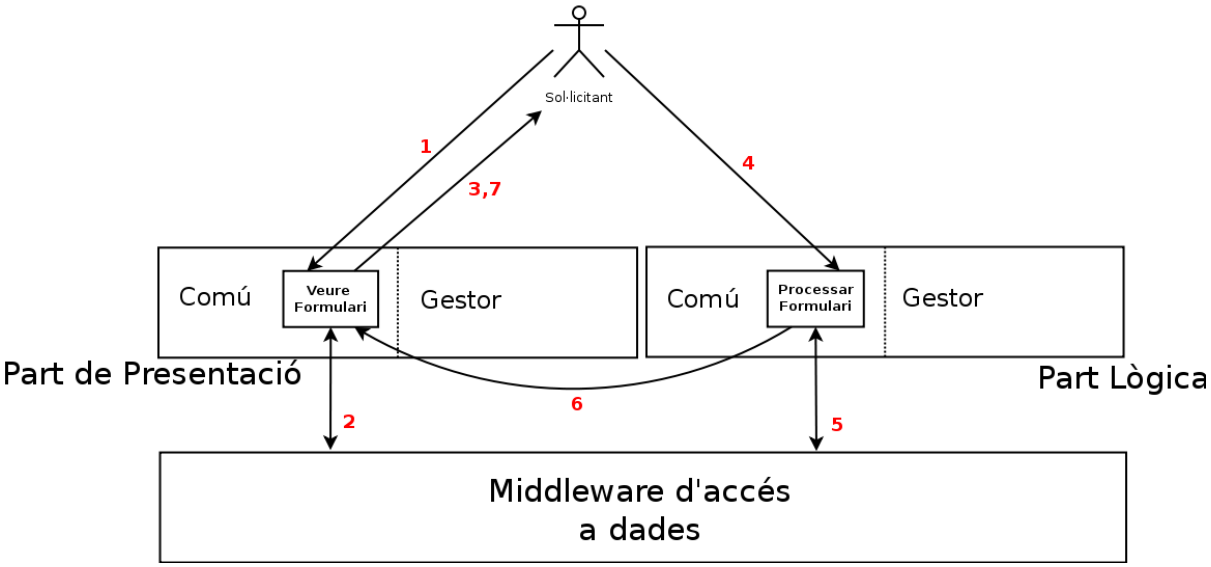

Figura 5.4: Arquitectura interna del sistema web amb l'exemple d'ús d'omplir una sol·licitud. El sol·licitant demana veure el formulari (1), el servidor el prepara amb la informació que aconsegueix de la capa d'accés a dades (2) i el retorna (3). L'usuari omple el formulari i confirma les dades, enviant aquestes al servidor (4). Aquest actualitza l'estat del sistema utilitzant l'accés a dades (5) i, finalment, es redirecciona l'usuari a una altra pàgina pertanyent a la part de presentació (6) que s'envia al sol·licitant (7), on el cicle continua.

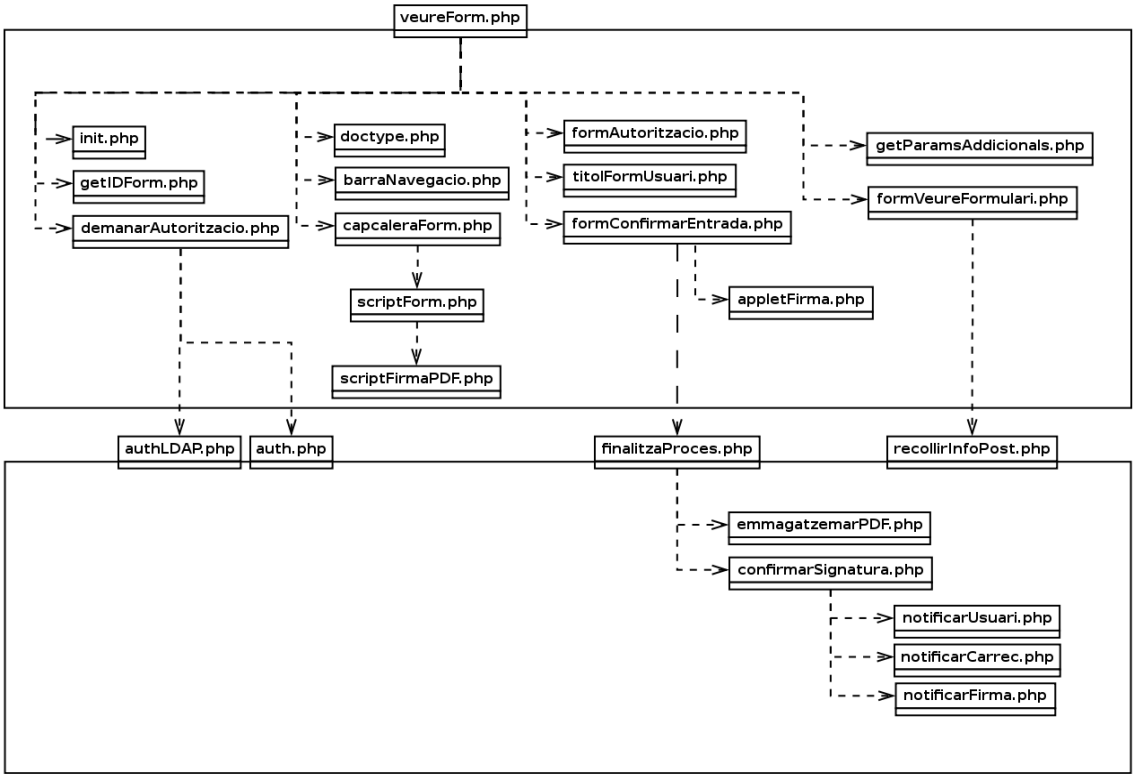

Figura 5.5: (Part de la) disposició i dependències dels components de la web, pel cas d'ús de veure formulari. La capa superior representa la part de presentació i la inferior la lògica. Veiem que la pàgina per veure un formulari pot utilitzar l'autenticació (auth.php i authLDAP.php), el confirmat de sol·licituds (finalitzaProces.php) i el recollit d'informació per següents pàgines (recollirInfoPOST.php); aquestes pàgines només processen informació i, acte seguit, redireccionen a algun altra pàgina.

5.3.2 Característiques addicionals

Per simular l'estat actual de la sol·licitud mentre aquesta s'està realitzant, i per permetre la revisió i modificació de les dades abans d'enviar-les, utilitzarem el concepte de *sessions PHP*. El mòdul PHP del servidor emmagatzema un *token* que identifica a cada connexió. Per cada sessió es poden definir variables de sessió emmagatzemades al servidor i accessibles només a la connexió associada. Així doncs, tenim les dades de la sol·licitud (de forma temporal) al servidor, donant la opció de modificar-les (e.g. anant "enrere" amb el navegador i reomplint una pàgina d'un formulari).

Com ja hem dit, utilitzarem comunicació xifrada mitjançant SSL. Per fer-ho hem necessitat un certificat apropiat pel servidor, proporcionats per UPCnet i provinents de l'autoritat de certificació arrel TERENA.

Una funcionalitat que ha patit canvis ha estat la caducitat dels formularis. Tot i

que finalment està implementat purament amb PHP (comprovant si un formulari està caducat en cada consulta), la idea inicial va ser mitjançant *events SQL*. En parlarem més a l'apartat de la base de dades.

Mencionem també que per la implementació del tipus de camp data s'ha utilitzat i adaptat un component javascript extern, de codi lliure: "Simple Calendar Widget", creat per Anthony Garret [9].

5.3.3 *Middleware* d'accés a dades

Per tal de separar la presentació i lògica de l'accés a les dades que els primers necessiten implementarem una capa de dades. Aquesta actuarà de *middleware*, o intermediari, per la informació necessària tal com sol·licituds, formularis, càrrecs, ..., a més dels documents PDF.

Un benefici important de disposar d'aquest *middleware* és que si es canvia la forma d'accedir a aquestes dades (e.g. canviant la implementació dels repositoris o restringir-hi l'accés per qualsevol raó) només hem de canviar la forma en què accedim a les dades des d'aquest subsistema, sense afectar a la resta del sistema web.

La seva composició està basada en cinc control·ladors: quatre dels àmbits d'informació que necessitem de la base de dades (formularis, sol·licituds, càrrecs i camps) i un altre pels documents PDF. Cadascún disposa de les funcions necessàries pel funcionament del sistema web.

5.3.4 Notificació i autenticació

Integrats en el sistema web trobem (parts de) dos components: les notificacions i l'autenticació.

En el cas de les notificacions, com hem apuntat a l'anàlisi, trobem les notificacions (immediates) i les notificacions periòdiques. En els dos casos les hem implementat mitjançant correu electrònic; les primeres les hem acoblat al sistema web (només s'utilitzen quan hi ha canvis en el sistema provocats via web), mentre que les segones formen un component apart que tractarem més endavant.

En els dos casos, els correus s'envien mitjançant el servidor de redirecció de correu de la UPC (relay.upc.es).

Per l'autenticació, com hem dit hem de permetre l'accés mitjançant credencials UPC i per targeta criptogràfica. El primer cas l'implementarem accedint al servidor LDAP de la UPC per confirmar que l'usuari existeixi i la paraula de pas sigui correcta. Aquest procés es pot fer des de la mateixa web, ja que existeix una llibreria PHP que ens permet una comunicació d'aquest tipus sense més problema.

Per l'accés amb targeta necessitem per una banda validar el certificat digital del

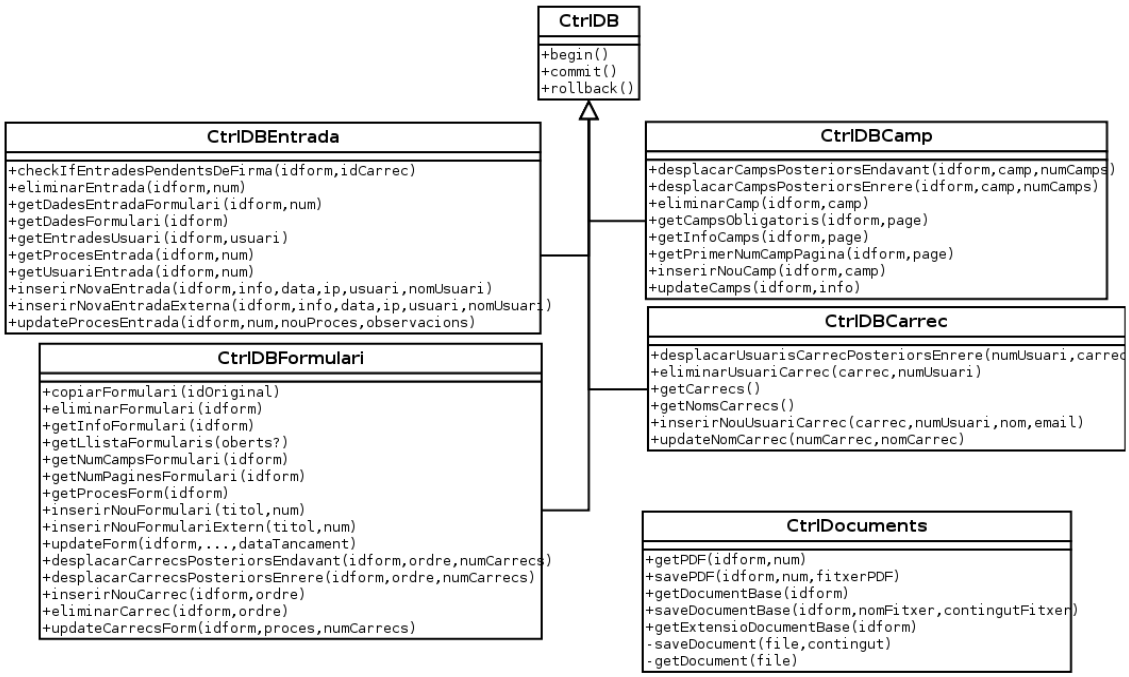

Figura 5.6: Diagrama de classes del *middleware* d'accés a dades.

client i per l'altra associar aquell certificat amb l'usuari UPC apropiat. Afortunadament la UPC disposa d'una infraestructura SOA (*Service-Oriented Architecture*, o arquitectura orientada a serveis) que, d'entre altres, té un servei que realitza aquesta tasca. Utilitzarem aquest servei mitjançant la llibreria *PKI Client*, creada també per la UPC i escrita en Java, que integrarem a un *servlet* del servidor d'aplicacions Tomcat (explicarem en què consisteixen els *servlets* a la secció del subsistema de generació de PDFs) [29]. Així doncs, per validar un certificat des de la web només ens caldrà cridar al *servlet*, el qual utilitzarà la llibreria *PKI Client* per validar el certificat digital i retornar-nos el nom d'usuari UPC del client.

Aquesta manera d'utilitzar software es coneix com a *web service*, i es basa en la idea del software com un servei que pot oferir una entitat, sense que el consumidor hagi de preocupar-se pel seu desplegament, manteniment o altres detalls irrellevants. En el nostre cas, utilitzant la llibreria *PKI client* ens assegurem que el procés de validació de certificats del nostre sistema millorarà conforme s'actualitzi el servei ofert per la UPC, cosa que allargarà la vida útil d'aquest subsistema.

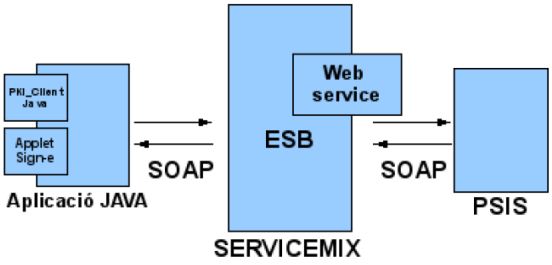

*Figura 5.7: Esquema de comunicació en la validació de certificats. L'aplicació java (en el nostre cas la llibreria PKI client) invoca el servei de validació mitjançant el bus de servei (*Enterprise Service Bus*), on la comunicació es fa amb el protocol SOAP (*Simple Object Access Protocol*) [17]. Finalment, aquest bus actua d'intermediari per fer la petició a la Plataforma de Serveis d'Identificació i Signatura (PSIS). Tot i què aquesta plataforma pertany al CATCert, aquesta és capaç de validar certificats provinents d'una gran varietat de PKIs, incloent el DNI electrònic [15].*

5.4 Base de dades

De forma bastant previsible, implementarem el repositori general d'informació com a base de dades. Utilitzarem la base de dades MySQL, base de dades lleugera i de codi obert. Aquesta serà consultada per mitjà del *middleware* d'accés a dades o pel sistema de notificació.

Descriurem l'estructura de la base de dades mitjançant el seu *esquema*, però abans hem d'elaborar el diagrama de classes del domini que tractarem.

5.4.1 Diagrama de classes

El diagrama resultant de tots els conceptes que tractarem és el presentat a la figura 5.8.

És necessari mencionar un punt important del diagrama que reflexa una decisió important de disseny: els processos estaran tant associats als formularis com a les sol·licituds. La raó d'això és la consistència del procés administratiu; si un formulari té sol·licituds en tràmit i es canvia el procés administratiu d'aquest formulari, hem de decidir que fer amb aquestes sol·licituds: anul·lar-les, alterar el seu procés per adaptar-les al nou o ignorar els canvis i seguir amb el mateix. La primera opció implica demanar als sol·licitants refer les sol·licituds i la segona és massa complexa i poc definida (e.g. eliminar un càrrec del procés que ja ha signat una sol·licitud). Així doncs, quan creem una nova sol·licitud li afegirem també la informació del procés per desacoblar-la totalment del procés del formulari.

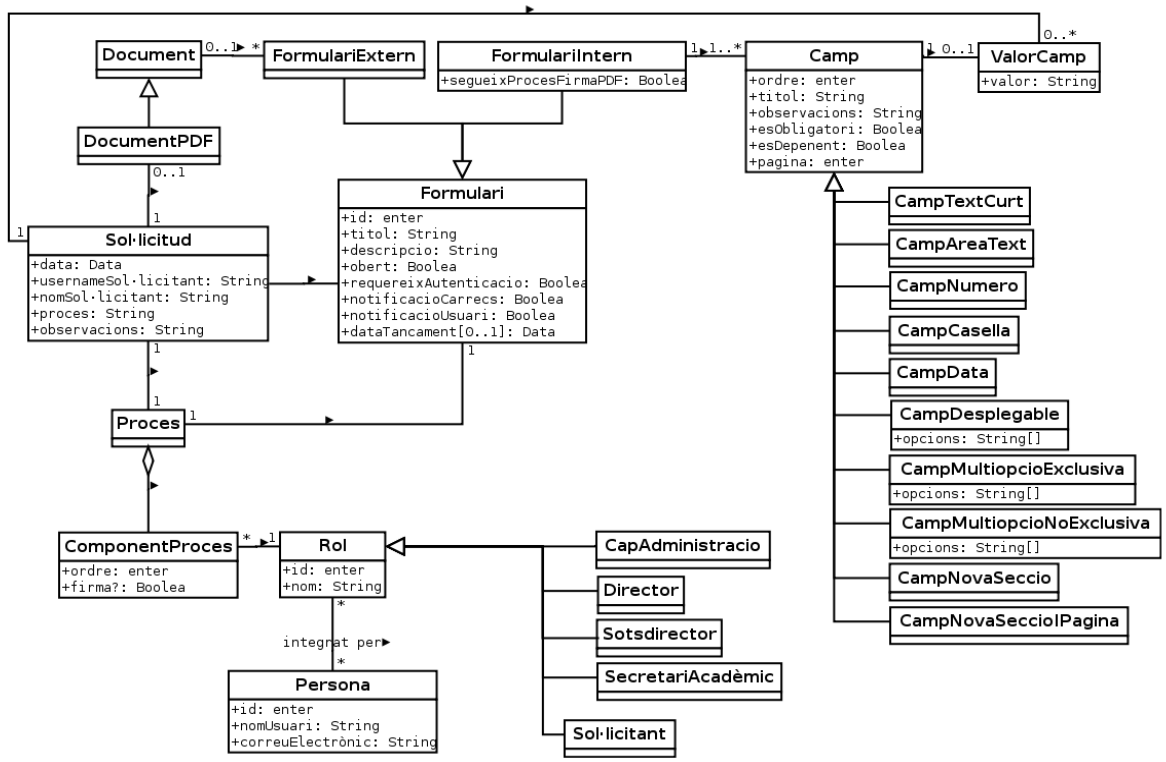

Figura 5.8: Diagrama de classes de la lògica del domini.

5.4.2 Esquema

A partir del diagrama de classes anterior realitzarem les transformacions necessàries per obtenir un esquema d'una base de dades, utilitzant patrons coneguts.

- Integrem totes les jerarquies d'herència en una mateixa taula (e.g. formulari intern i formulari extern passen a ser part de formulari). D'aquesta manera afegim tots els atributs de les especialitzacions dins de la superclase, on tindran valor nul si no és aplicable a la instància en concret. Com les subclasses no són massa diferents entre si no perdem massa espai amb redundàncies.

- Eliminem de l'esquema les classes de document i documentPDF, ja que decidim separar aquests de la base de dades (més informació a continuació).

- Integrem els valors del camp dins de les pròpies sol·licituds (com a *array*). Com cada sol·licitud pot tenir un nombre diferent de camps, crearem una nova taula per emmagatzemar la informació de les sol·licituds de cada formulari.

S'ha decidit separar els documents PDF de la base de dades per diverses raons:

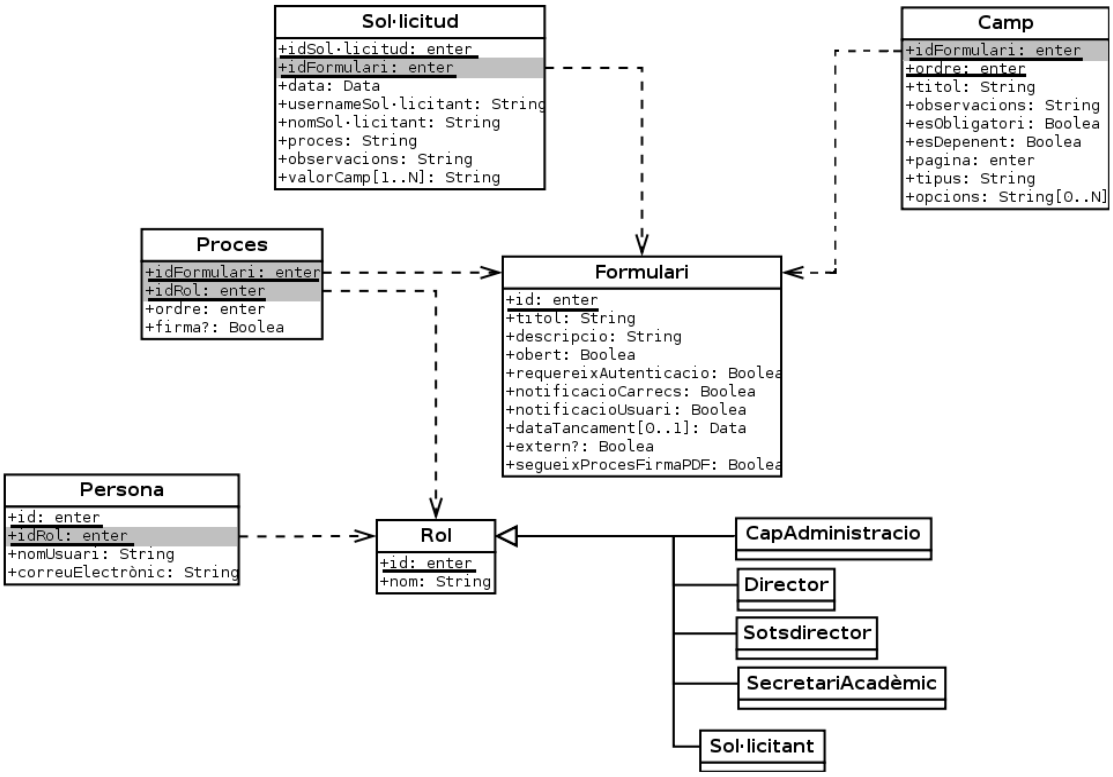

Figura 5.9: Esquema de la base de dades del nostre sistema. Els camps sombrejats són claus externes mentre que els subratllats són les claus primàries.

D'entrada els documents PDF poden ser bastant grans i proves (informals) de rendiment han fet veure que la latència és millor si són emmagatzemades directament a disc. D'aquesta manera, a més, lliurem a la base de dades de càrrega de treball, que necessita per servir altra informació a diversos clients alhora. Per altra banda també s'ha de tenir en compte que per possibilitar l'escalabilitat a un sistema més gran potser valdria la pena afegir la informació en una segona base de dades, cosa que permetria desacoblar els documents PDF del sistema web (i.e. guardar-los en un servidor físic diferent) i fer un sistema més modular.

Pel desplegament del nostre projecte, però, optarem per integrar els documents PDF al sistema web.

5.4.3 Característiques addicionals

Per garantir seguretat i consistència amb les dades del sistema, es treballa amb un motor de base de dades que suporta transaccions (InnoDB) [19]. Amb les transaccions podem assegurar que certes tasques es realitzin de forma atòmica, és a dir, o es fan totes les

subtasques satisfactoriament o no es modifica l'estat de la base de dades. Això ens resulta imprescindible en alguns casos d'ús, com en el de realitzar o signar sol·licituds.

Un tema important per l'extensió del sistema és la codificació de les dades introduïdes. Utilitzarem codificació *unicode*, que ens permet representar una gran varietat d'informació (com accents o caràcters d'altres alfabets) que en una codificació estàndard (les diverses ISO) no estarien disponibles. Això ens permetrà en el futur utilitzar tot tipus de caràcters "extranys" que podem necessitar, però ve amb l'inconvenient que les dades ocuparan més espai a memòria. Donat que no tindrem una quantitat massiva (gigabytes) d'informació emmagatzemada, podem estar tranquils.

Tal com hem mencionat a l'apartat de la web, la idea inicial amb la que implementar la caducitat de formularis va ser mitjançant l'ús d'*events SQL*. Aquests permeten inserir procediments SQL i una data en la que s'executarà, així que es podria tancar el formulari quan arribi la data de caducitat. Finalment no ha estat possible fer-ho, ja que la base de dades que utilitzem (compartida amb altres sistemes del departament) és d'una versió més antiga que no dona suport per events SQL. Donat que el departament és reaci a noves actualitzacions en l'entorn de producció, ens hem decidit per l'alternativa (menys elegant) de la comprovació de la caducitat en cada accés web.

5.5 Notificacions periòdiques

A partir de la tercera iteració hem introduit aquest component al sistema. Com el seu propòsit és relativament simple (notificar als càrrecs apropiats via correu electrònic), ho hem implementat amb un petit *script* de *bash*. Aquest contacta amb la base de dades i acte seguit utilitza el servidor de redirecció de correu de la UPC (relay.upc.es) per enviar el *email* al càrrec adequat.

Com a agent que inicia el *script* utilitzarem el planificador natiu de molts sistemes linux, *cron*. Aquest software permet declarar ordres per la seva execució de forma periòdica, en un interval definit per l'usuari.

5.6 Applet de signatura digital

Possiblement la part més important del sistema, implementarem el component de la signatura digital com a un applet de Java. És necessàri utilitzar una tecnologia que s'executi per part del client, ja que és ell qui firma amb la seva targeta; no hi ha manera de traslladar la feina al servidor. De les dues tecnologies d'aquest tipus més utilitzades, *Java* i *Javascript*, només la primera permet accés al disc (mitjançant la signatura del codi del applet). Així doncs, per necessitat utilitzarem Java.

L'agència Catalana de Certificació (*CATCert*) ha desenvolupat un applet (*Eina web*

de signatura-e [18]) amb moltes de les característiques que necessitem pel nostre sistema, així que partirem del seu codi per acabar d'adaptar una solució. En concret utilitzarem la versió 1.9.3, la més actual a l'inici d'aquest projecte.

La eina web de signatura-e accepta un gran ventall de paràmetres d'entrada; hi ha diverses maneres d'introduir el contingut a signar (fitxer local, URL, contingut d'una variable, múltiples fitxers, ...), on deixar el contingut signat (localment, en una variable javascript, enviar-lo com a contingut d'un formulari HTML, ...), quin tipus de signatura realitzar (CMS, diversos perfils de CAdES, XAdES, PAdES, ...), com accedir al magatzem de claus (certificat en fitxer, certificat de Windows, targeta criptogràfica, ...), algoritme de *hashing* (SHA-1, SHA-256, SHA-512, ...), etc.

Quasi tots els paràmetres els deixarem fixats, excepte la manera d'introduir el contingut a signar: en el cas de la signatura d'una sol·licitud immediatament després d'omplir-la, li passarem el fitxer PDF directament al client i aquest al applet per mitjà d'una variable javascript. D'aquesta manera estem assegurant al client que el que signa és realment el fitxer que està visualitzant en aquell moment i no ha estat canviat des de que ell l'ha visualitzat. Per altra banda, quan donem la possibilitat de signar múltiples fitxers (tant a un sol·licitant com als càrrecs) no és convenient seguir aquest procediment (podria haver molts fitxers). Així que en aquesta cas li passarem a l'applet una llista de URLs on aconseguir els fitxers. No és tant segur com el primer cas, però tenim els fitxers PDF amb els permisos necessaris per garantir la seva modificació legítima només a mans del nostre sistema.

La sortida del contingut signat es fa sempre mitjançant el mateix procediment: funcions javascript (onSignOK i onMultiSignOK). Aquesta és cridada quan el procés de signatura ha acabat satisfactòriament, i l'hem dissenyat de tal manera que s'envia un formulari amb el document al servidor, on aquest es processa.

5.6.1 Diagrama de classes

El diagrama de classes (de la part que utilitzarem) del sistema dissenyat pel CATCert és el mostrat a la figura 5.10. Veiem que existeix un paquet per la interfície gràfica (*gui*) i un altre per la part criptogràfica (*crypto*), que inclou l'accés al magatzem de claus de la targeta (*KeyStoreImpl*) i la creació de la signatura (*signImpl*). També existeixen altres paquets auxiliars per coses com fer peticions HTTP per CRLs o respostes OCSP.

Aquest diagrama representa el seu disseny original; no ha fet falta modificar l'arquitectura de l'applet per les nostres necessitats. La funcionalitat d'alguna d'aquestes classes sí que ha estat modificada, però, com comentem a continuació.

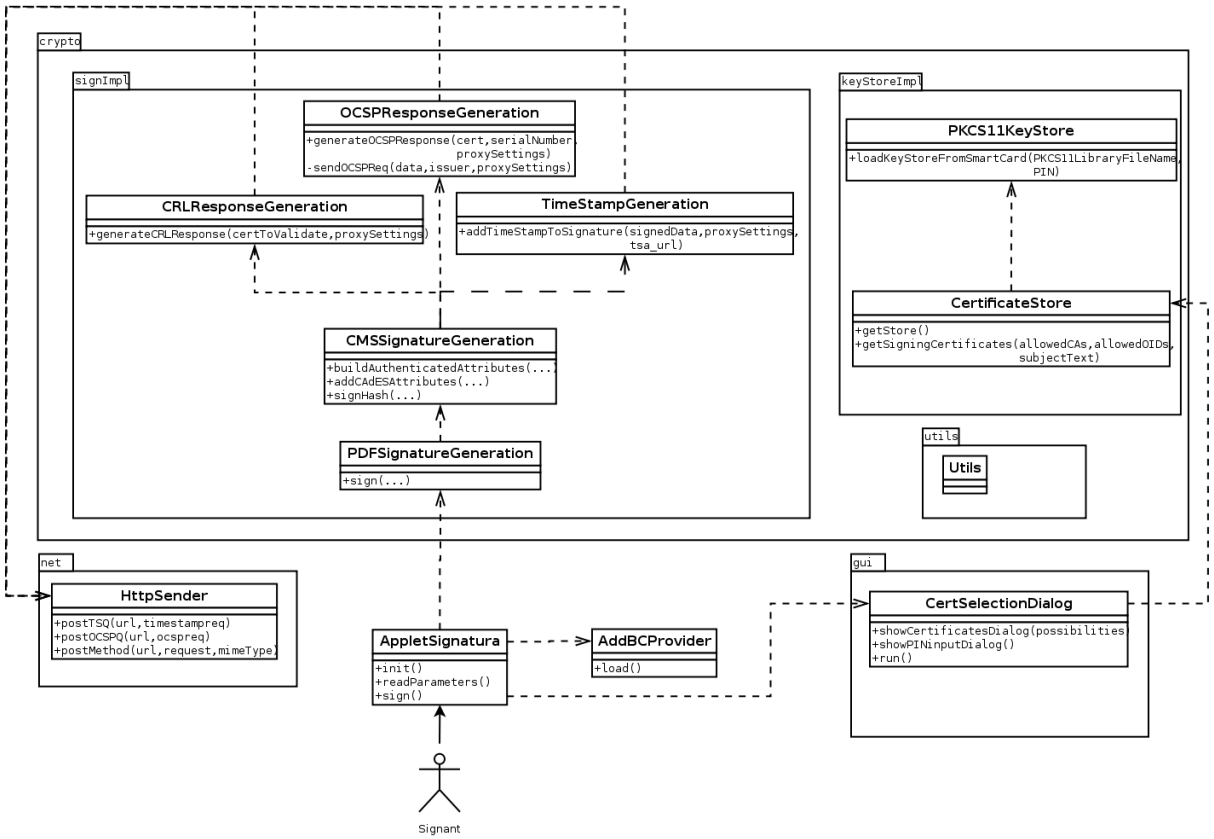

Figura 5.10: Diagrama de classes del applet de signatura electrònica. Només mostrem la part que utilitzarem.

5.6.2 Estructura de la signatura implementada

Per tal de tenir present com serà la nostra signatura (quina informació contindrà) la il·lustrem a la figura 5.11.

Explicarem l'estructura d'aquesta signatura començant per l'exterior i anant concretant les parts internes. Veiem que la signatura PAdES es compon de dues parts, la informació de la signatura i la pròpia firma en format CAdES. La primera part forma part del document PDF, i és la que enllaça la signatura amb el document i afegeix informació de les circumstàncies (rol, lloc i hora). La segona és on es troba la seguretat criptogràfica, i ha de seguir un dels perfils de CAdES. En el nostre cas afegirem el segell de temps (per garantir la hora i data de la firma) i la informació de validació (per garantir la validesa del certificat en el moment de signar), tot i que aquests no són estrictament necessaris pel compliment dels perfils CAdES més bàsics.

Un punt important és quina informació es xifra en una signatura CAdES. Per assegurar un conjunt d'informació sensible, com el certificat signant (per evitar atacs de substitució de certificats), el que fa CAdES és crear un conjunt de camps que anomena *atributs*

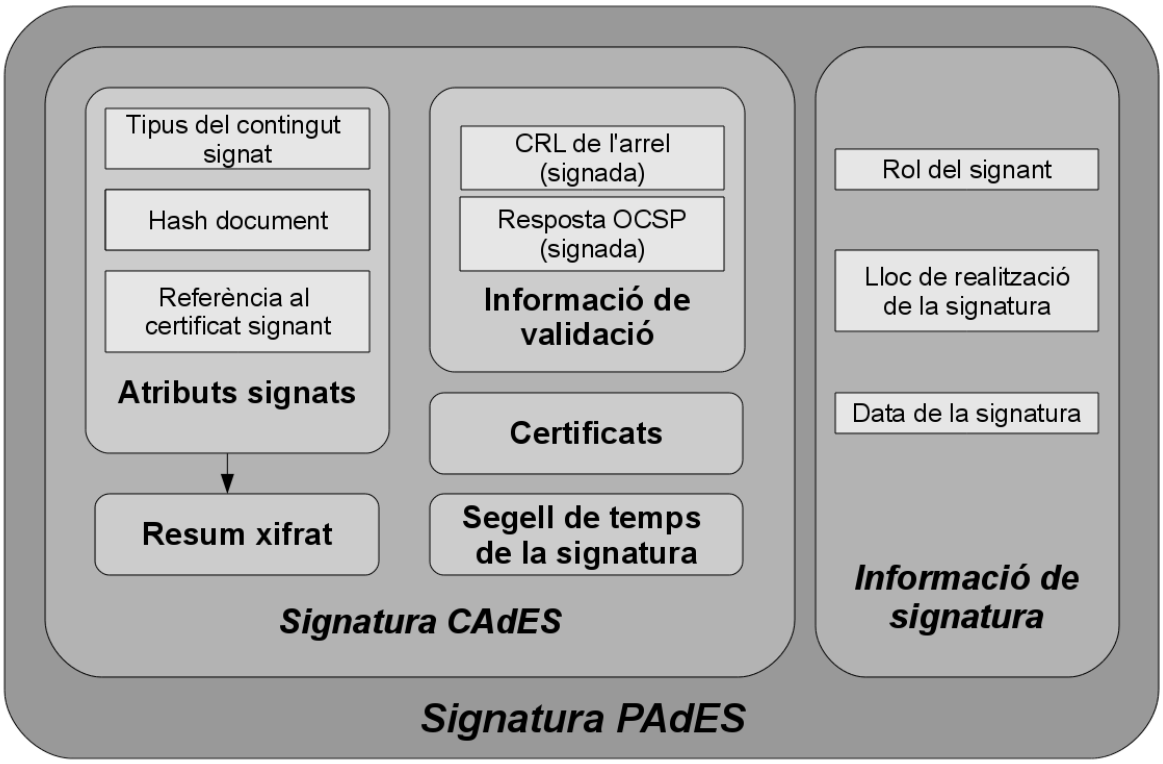

Figura 5.11: Estructura interna de la signatura que utilitzarem en el nostre sistema.

signats, (signed attributes) i és d'aquest conjunt del què es calcula (de nou) el *hash* i es xifra.

També veiem, tot i que de nou no és estrictament necessari incloure'ls, la llista de certificats implicats amb la signatura (certificat signant més la seva cadena de certificació).

5.6.3 Canvis respecte el disseny original

En el punt següent veurem el cas d'ús d'aquest subsistema (signatura electrònica), però abans mencionarem les parts que hem canviat del disseny per adaptar-ho al nostre sistema:

- En l'applet original s'ha d'especificar la ruta del controlador (*driver*) de la targeta criptogràfica a utilitzar. Això és un greu problema d'usabilitat per l'ús en el nostre projecte: no només no ens permet un sistema multiplataforma, sinó que tampoc podriem canviar entre tipus de targeta (UPC i DNIe). En el nostre cas implementarem un sistema de detecció de *drivers* en tres etapes:

1. Cerca automàtica del controlador, on buscarem els *drivers* a partir d'una llista de rutes per defecte. En aquesta etapa es detectaran la majoria de casos.
2. Cerca de controladors previs, on buscarem el sistema de fitxers per la ruta del controlador utilitzat amb èxit de forma prèvia.
3. Selecció manual de controlador, on deixem a l'usuari escollir la ruta si els anteriors passos fallen.

Una vegada aconseguim signar, guardem la ruta del controlador correcte al sistema de fitxers local de l'usuari per consultar-lo en properes signatures (pas 2).

- Per defecte l'addició de la informació de revocació (a fitxers PDF) està desactivada. En el nostre cas l'activarem.

- No existeix suport per PAdES-LTV, així que comencem a implementar el concepte de DSS i segell de temps del document (secció 4.10.6). Malauradament, per temes de compatibilitat amb altre software (secció 6.5.4), desestimarem el perfil LTV i ens decidirem pel BES, soportat per l'applet.

- El càlcul de valors *hash* per les autoritats certificadores estava precalculat. Nosaltres realitzarem el càlcul, disminuint l'eficiència però permetent una vida més llarga al sistema (funcionarà quan canviïn els certificats).

- La validació dels certificats es feien sempre mitjançant CRL. En aquest cas les farem mitjançant OCSP.

En el cas del DNI electrònic no tenim més remei que validar mitjançant OCSP, però en el cas del carnet UPC podem escollir. L'elecció resulta prou fàcil; una resposta OCSP és poc més que un byte amb l'estat del certificat (a més d'altra informació com certificats signants), mentre que una CRL pot ser bastant més voluminosa. A efectes de comparació, la CRL de l'entitat EC-UR durant la redacció d'aquest document ocupa 87 kB. Si multipliquem aquesta llista per dos (per la representació de cadena hexadecimal), cada signatura significaria uns 200 kB. També, donat que hem de reservar l'espai per la signatura amb antel·lació, podriem córrer el risc de no tenir-ne suficient. Amb uns fitxers que s'espera que no ocupin més que uns pocs kB això resulta absurd, així que farem la validació sempre mitjançant OCSP.

Afegirem més modificacions al sistema, però la necessitat d'aquestes serà vista a l'hora de la implementació, que veurem al següent capítol.

Amb tots aquests canvis ja tenim clar quin tipus de signatura implementarem al nostre sistema: signarem els PDFs de forma integrada (com hem quedat als requeriments) mitjançant l'estàndard PAdES, afegint la informació de validació i segellat de temps segur. Amb totes aquestes dades podrem disposar d'una signatura de llarga durada amb garanties sòlides de no repudi.

5.6.4 Cas d'ús: Signar

Amb aquestes modificacions de disseny elaborarem dos diagrames de seqüència que mostrin el funcionament intern de la signatura: primer la obtenció del magatzem de claus (5.12) i després la signatura pròpiament dita (5.13).

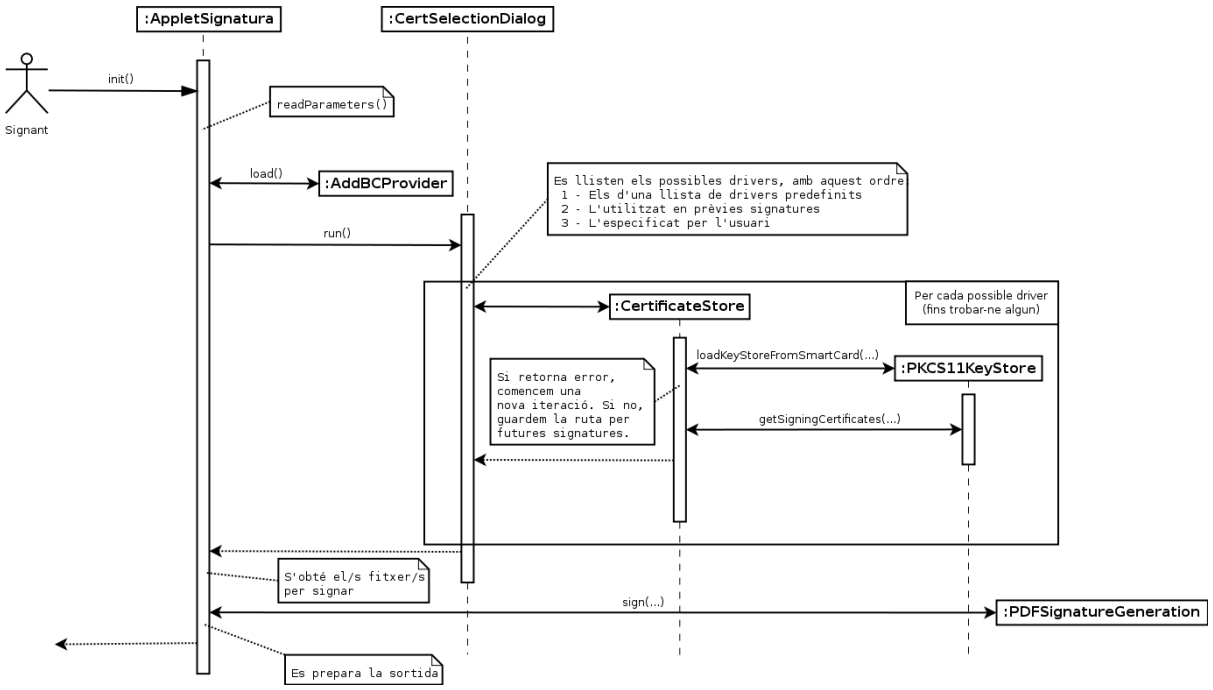

*Figura 5.12: Diagrama de seqüència (simplificat) de la obtenció del magatzem de claus (*keystore*). Podem veure el funcionament el mecanisme de cerca de controladors.*

5.7 Generació de PDFs

Aquest sistema serà l'encarregat de traduir les dades referents a una sol·licitud (informació del formulari, valor dels camps, nom del sol·licitant, signants, ...) a un document PDF. El seu funcionament serà de petició-resposta, és a dir, aquest component no guardarà de forma persistent cap informació; només rebrà informació i retornarà el document creat al remitent. Així evitem acoblaments innecessaris i donem la possibilitat d'utilitzar el sistema de forma externa (e.g. servei web).

La seva missió secundària serà afegir els camps de signatura a un document donat. Com ja hem vist en etapes anteriors, això serà necessari per les sol·licituds de formularis externs, on el sol·licitant ens envia el document PDF final i nosaltres l'hem de preparar pel procés administratiu. Per la preparació de les signatures necessitem, com hem mencionat a la seccio 2.5.1, modificar el fitxer, insertant els camps de signatura definits al procés del

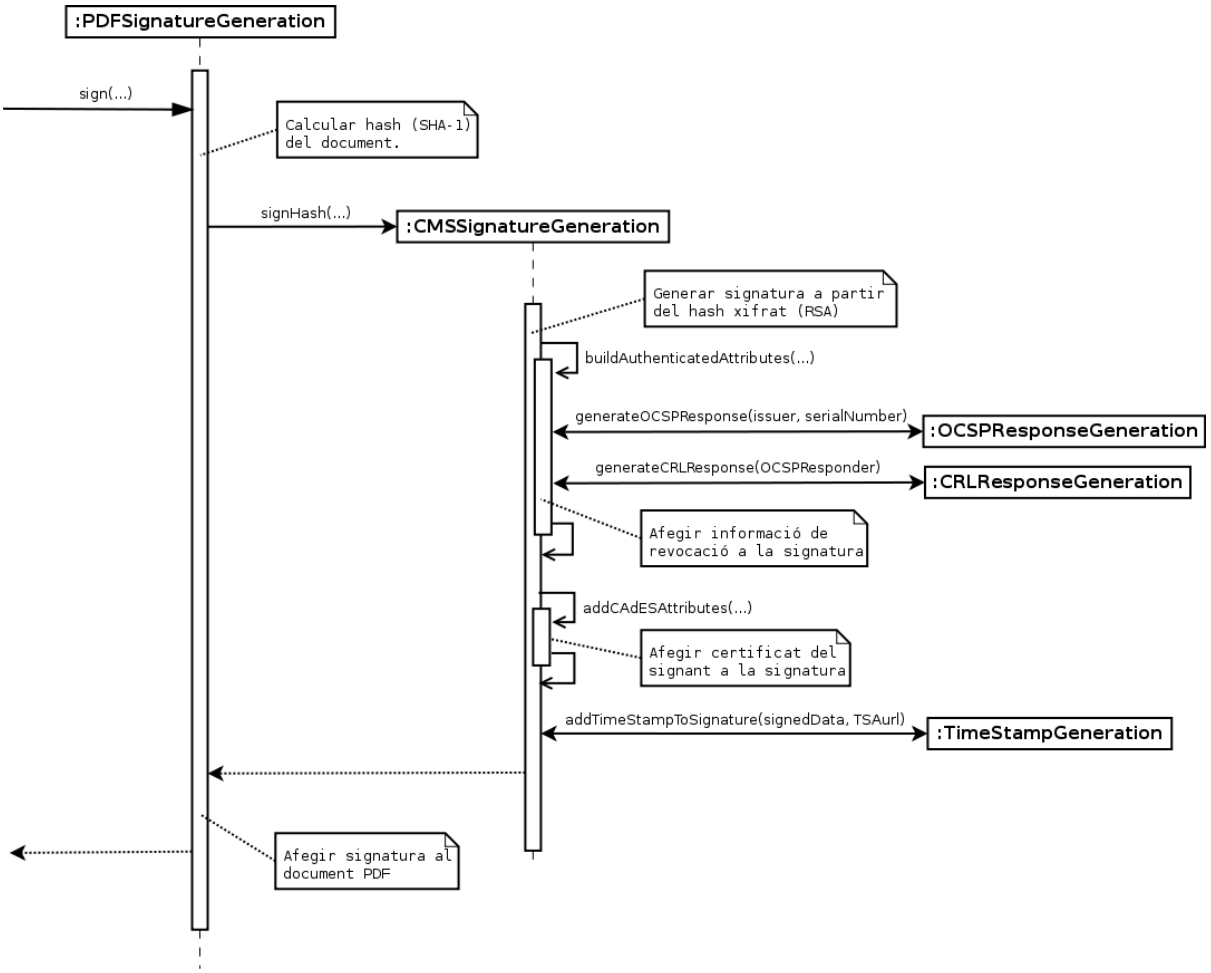

Figura 5.13: Diagrama de seqüència (simplificat) de la creació de la signatura. Inclou l'addició de la resposta OCSP i de les CRLs (d'autoritats revocades).

formulari. Això afegirà una nova versió del document amb els camps apropiats i llestos per signar.

En quant a la tecnologia amb la que ho implementarem, escollim la tecnologia Java dels *servlets*, igual que amb l'autenticació per targeta. Un servlet és una aplicació en Java que accepta com a entrada una connexió al servidor, normalment amb el protocol HTTP. El *servlet* processa la petició amb els paràmetres passats pel mètode HTTP i finalment retorna al client una resposta HTTP amb la informació processada. Utilitzarem *Tomcat* com a contenidor de servlets a l'entorn de producció final. Aquest forma part de la mateixa familia de productes que el servidor web que utilitzem, Apache, i és un dels productes més utilitzats del seu tipus.

Com a llibreria auxiliar optem per la utilització de *iText*. Aquesta és una llibreria de codi obert que ens permet crear i editar fitxers PDF de forma ràpida i senzilla.

Inicialment es va plantejar també la possibilitat d'utilitzar el què es coneix com a

plantilles XSLT. Aquestes formen part de la família de llenguatges XSL (*eXtensible Stylesheet Language*), els quals poden ser utilitzats per definir un flux de transformacions per convertir un fitxer XML a un fitxer PDF. Mostrem el procés esquematitzat a la figura 5.14.

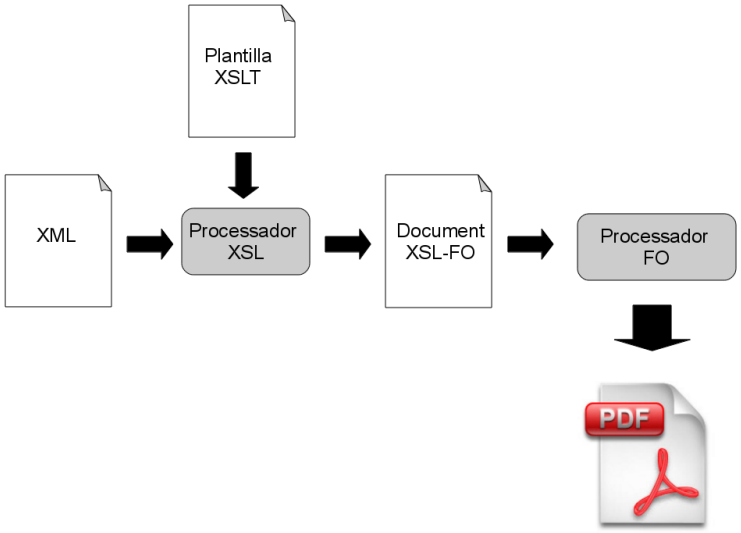

Figura 5.14: Procés d'obtenció d'un document PDF a partir d'un fitxer XML mitjançant XSLT. El primer pas és transformar el XML, amb ajuda d'una plantilla i un processador XSL, a un document XSLT-FO. Aquest conté la informació de tant el contingut com la presentació del document final. Finalment mitjançant un processador FO es transforma aquest document XSLT-FO en un altre document arbitrari (com PDF, RTF, DOC, ...).

De tota manera, un problema d'aquest mètode és que, tot i ser més ràpid que l'anterior, és més complicat de dissenyar i menys obvi el seu funcionament. Apart, les possibilitats del PDF generat no són massa personalitzables. Per aquestes dues raons hem optat per la llibreria iText.

5.7.1 Especificació de l'entrada i la sortida

Com hem dit, l'entrada d'un servlet és una connexió HTTP, però hem de definir quina informació portarà aquesta i de quina manera.

Recordem que, en el cas de generació de documents, el tipus d'informació que rebrem és una estructura amb la informació general d'una sol·licitud (autor, data, títol del formulari, ...), del contingut (nom i valor de cada camp) i del procés (càrrecs que necessiten signar). Així doncs necessitem representar aquesta informació estructurada en format text, per la qual cosa el XML és la elecció idònia. Definirem la següent sintaxi:

```
<entrada>
    <formulari> Títol del formulari </formulari>
    <idForm> 0 </idForm>
    <nomSolicitant> David </nomSol·licitant>
    <firma><carrec>Director</carrec></firma>
    . . . (altres firmes)
    <seccio>
        <nomSeccio>Títol de la secció</nomSeccio>
        <camp>
            <nomCamp>Camp 1</nomCamp>
            <valorCamp>Valor del camp 1</valorCamp>
        </camp>
        . . . (altres camps)
    </seccio>
    . . . (altres seccions)
</entrada>
```

En el cas de l'addició dels camps de signatura, necessitarem per una banda el document PDF i per l'altra informació sobre els signants. Per aquesta última utilitzarem la part XML anterior adequada (elements *firma*) i pel primer passarem el document codificat en base 64, que permetrà fer arribar el fitxer sense problemes de codificació.

En els dos casos es retornarà la sol·licitud PDF codificada en base 64.

5.7.2 Diagrama de classes

Veiem el diagrama 5.15 i observem que és bastant simple: dues classes heredant de la classe HttpServlet de Java, cadascuna oferint un dels serveis que necessitem (representats a la frontera del sistema). Finalment també hi ha classes auxiliars per realitzar estampats en totes les pàgines d'un document (en el nostre cas el logotip del departament) i per parsejar el XML (SAX, *Simple API for XML*).

5.7.3 Cas d'ús: Generar document PDF

Mostrem el procés simplificat de la generació de documents PDF en el diagrama 5.16.

El diagrama de seqüència de l'addició de camps de signatura seria un subconjunt del de generació, només afegint els camps de signatura a un document passat com a paràmetre (en lloc de generar-lo nosaltres).

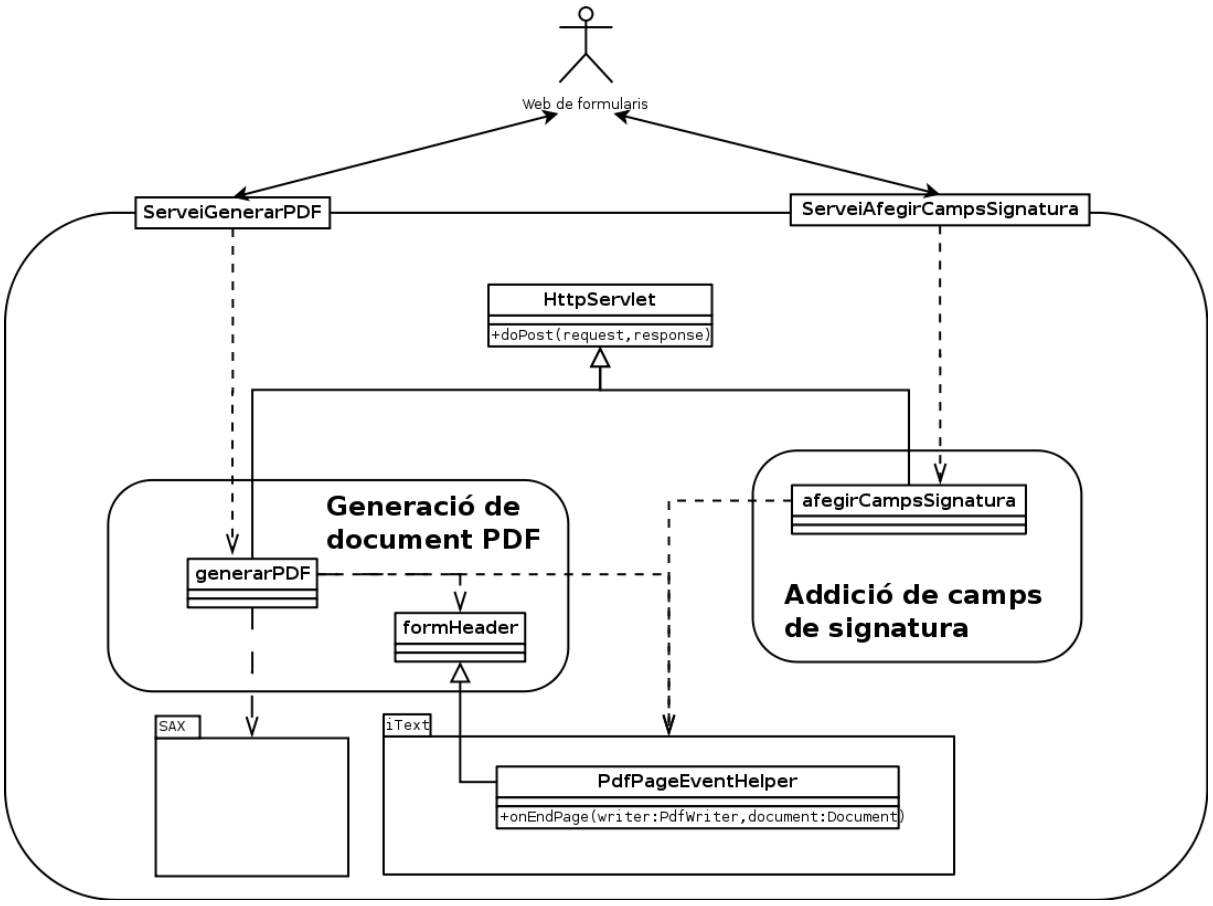

Figura 5.15: Diagrama de classes (simplificat) del subsistema de generació (i preparació) de documents PDF.

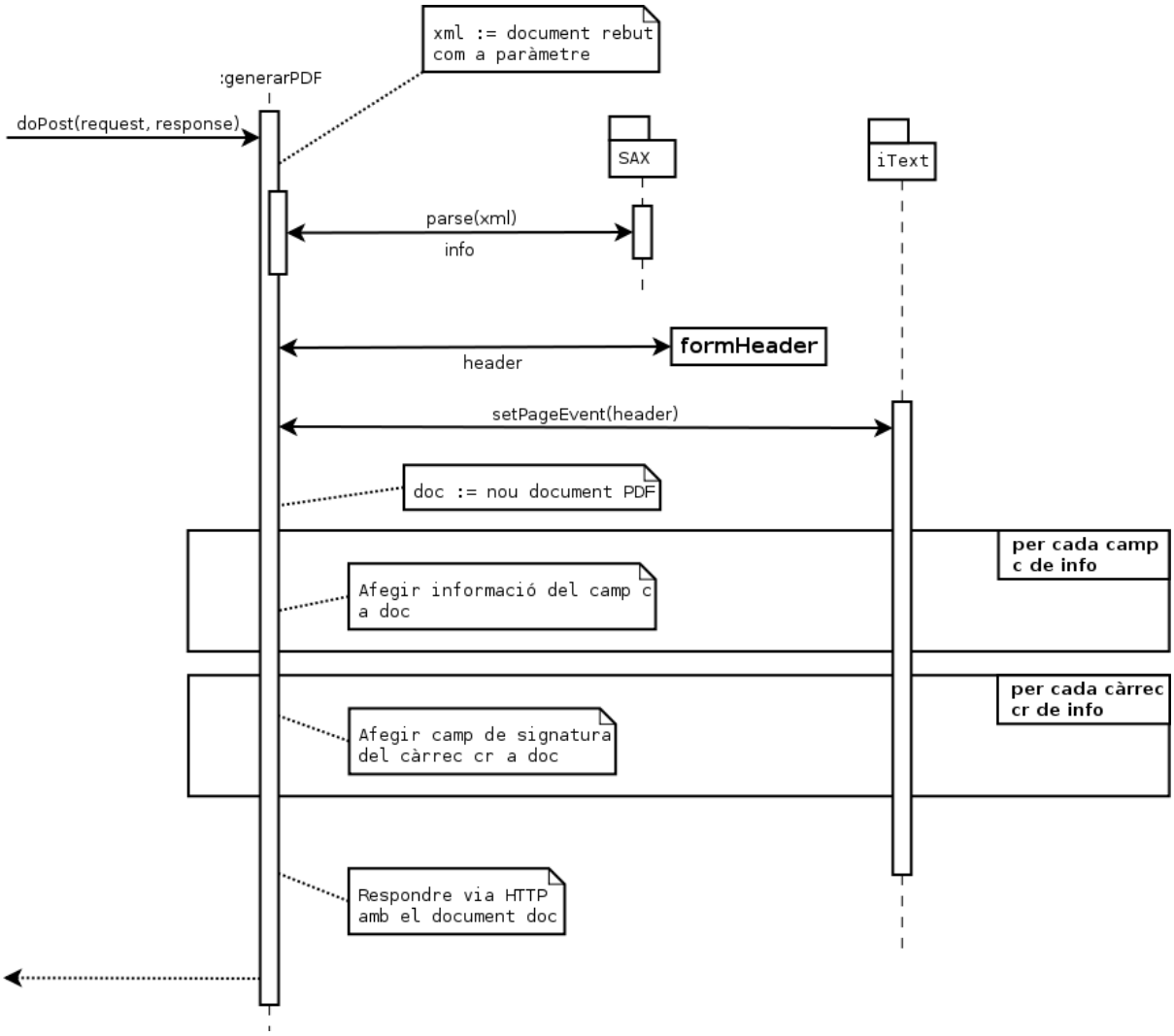

Figura 5.16: Diagrama de seqüència (simplificat) de la generació de PDFs.

Capítol 6

Implementació

En aquest capítol detallarem el procés d'implementació del projecte. En general, l'objectiu d'aquesta fase és escriure el codi a partir del disseny anterior. Amb un bon disseny, la implementació no ha de ser massa problemàtica, tret de problemes molt concrets, sovint trobats amb la comunicació amb sistemes software externs.

Començarem detallant per una banda en quin entorn hem desenvolupat la implementació (les eines, màquines i software utilitzat) (*entorn de proves*), així com l'entorn final on es desplegarà el sistema (*entorn de producció*).

A continuació, tot i que tècnicament formen part del disseny, inclourem aquí les proves de concepte (*proof of concept*) per determinar si alguna decisió de disseny dubtosa és realment factible a la pràctica. Si no és així, no ens quedarà més remei que tornar a la fase de disseny a buscar una altra alternativa.

Finalment, cal saber que l'existència de l'etapa de proves no implica que en aquesta fase no es faci cap comprovació; en cada tros de codi suficientment gran i independent farem proves elementals per comprovar que funciona. Passar aquestes proves no implica el bon funcionament del sistema, i és només per evitar petits problemes (com errors tipogràfics o el bon funcionament local, a nivell de funció). Sobre això, comentarem els problemes que ens hem trobat.

6.1 Eines de desenvolupament

El software de desenvolupament de codi ha estat Eclipse, tant per Java (servlet de PDFs i applet de signatura electrònica) com per PHP (web).

També hem rebut l'ajuda de UPCnet per l'obtenció del certificat del servidor web (per l'ús segur SSL), tot i que no ens ha estat possible que ens generin un certificat pel signat del codi. En aquest últim cas no ens ha quedat més remei que crear-nos un certificat

autosignat, com comentem més avall.

6.2 Entorn de proves

L'entorn de proves on s'ha desenvolupat el projecte ha estat de naturalesa domèstica, amb una potència limitada. Com a efecte secundari útil inesperat, hem pogut ser molt més sensibles als efectes de l'eficiència del sistema en l'infraestructura informàtica.

6.2.1 Hardware

Els detalls (rellevants) de la màquina de l'entorn de proves són:

- Processador AMD Athlon(tm) 64 X2 Dual Core Processor 5200+ (dos nuclis)
- Freqüència de rellotge de 1 GHz
- Memòria cache de 1 MB
- Memòria RAM de 2 GB
- Connexió mitjançant xarxa local de 100 Mb/s

6.2.2 Software

El sistema operatiu utilitzat ha estat una distribució Debian de Linux virtualitzada dins d'un altre Debian, ambdós de la versió *etch*. La màquina virtual utilitzava 1 GB de RAM i utilitzava un dels dos nuclis disponibles.

La *virtualització* consisteix en simular el funcionament de diverses màquines independents (amb el seu processador, memòria, espai de disc, ...) en una sola màquina física. Aquest mètode és una bona idea ja que el món del hardware, en general, avança més ràpidament que el del software i un sol servidor físic moltes vegades treballa a un ritme de treball molt per sota del seu potencial (inferior al 50 %). D'aquesta manera s'aprofiten millor els recursos i es centralitza totes les funcions d'un negoci en una sola màquina física. A més, les màquines virtuals, com a fitxers que són, poden ser copiades, mogudes o esborrades sense gaire esforç.

La resta de software usat en els subsistemes del projecte està especificat al capítol anterior, amb l'excepció del contenidor de *servlets*. En l'entorn de proves hem utilitzat *jetty* [28], un software més lleuger que Tomcat però possiblement menys versàtil.

6.3 Entorn de producció

6.3.1 Hardware

El hardware físic on s'executarà el sistema es trobarà en el servidor *gregary*, pertanyent al departament de MA2. Aquest servidor conté diverses màquines virtuals oferint diversos serveis al personal del departament, com accés a disc i impresores o espai web, així que el software desenvolupat en aquest projecte haurà de compartir recursos amb altre software.

- Processador Intel(R) Xeon(R) CPU X5460 (quatre nuclis)
- Freqüència de rellotge de 3.1 GHz
- Memòria cache de 6 MB
- Memòria RAM de 8 GB

6.3.2 Software

En l'entorn de producció també s'ha utilitzat un sistema virtualitzat, amb la mateixa disposició de sistemes operatius (Debian tant a la màquina física com a la virtual). En aquest cas s'utilitza la versió de Debian més recent, *lenny*.

6.4 Proves d'implementació

Per provar un (sub)sistema és bo aïllar aquest de la resta, i per això el provem utilitzant *stubs* en lloc dels altres components que interaccionen amb ell. Un *stub* és codi que actua com a representant d'una part del sistema sense contenir cap lògica real, només una simulació, i els utilitzarem per provar el funcionament bàsic del subsistema web. Només s'han utilitzat en aquest cas perquè, com ja hem dit, aquest component és l'únic que es comunica amb altres.

A continuació llistem els *stubs* utilitzats:

Servlet de PDFs: Retorna un PDF buit.

Applet de signatura: Retorna el mateix PDF introduit.

Autenticació: Retorna una autenticació correcta, amb un identificador inventat.

Notificacions immediates: Escriu un fitxer al disc dur com a prova de què s'ha realitzat la notificació.

En els altres possibles casos no s'han utilitzat perquè la seva implementació no era massa complexa.

6.5 Proves de concepte i problemes trobats

Per demostrar si una tecnologia o tècnica és aplicable en el nostre projecte, hem desenvolupat petites proves de concepte que ens permeten comprovar si són factibles. A més, durant altres proves d'implementació sovint ens podem trobar amb problemes que ens hagués estat impossible o al menys molt difícil de localitzar a l'etapa de disseny.

A continuació llistarem els problemes i proves de concepte que ens han obligat a reconsiderar algunes decisions prèvies:

6.5.1 Sistema web

- Les plantilles CSS, encarregades de definir l'aparença de la web, han donat molts problemes per establir la mateixa visualització en tots els navegadors. Tot i ser un estàndard publicat pel W3C (*World Wide Web Consortium* [13]), l'adherència de les implementacions a aquest estàndard és bastant pobre, causant que la web es vegi en cada navegador de forma diversa (en petits detalls). Moltes hores d'implementació han estat dedicades per tal de que això no afecti la usabilitat, però en el producte final continuen havent-hi discrepàncies. De tota manera, les que hi ha no afecten la usabilitat i, com no considerem l'estètica un requeriment no funcional important, no ho considerarem important.

- Un tema recorrent en tot el sistema ha estat la codificació dels càracters en la comunicació. Per una banda hem utilitzat UTF-8, cosa que ens hem hagut d'assegurar que funciona a tot arreu, però a més hi ha el fet de què segons l'entorn que tractem hi ha alguns caràcters "prohibits" que hem de tractar prèviament. Per exemple, la cometa simple (') en javascript o el símbol de percentatge (%) en una URL han de ser escapats.

6.5.2 Servlet de generació PDF

- Durant la segona iteració es va pensar en utilitzar l'estàndard PDF-A per la generació de PDFs [11]. El format *PDF-A* és una restricció del format PDF pensat per l'arxivament de documents. Aquest consisteix en la inclusió de tota la informació necessària per visualitzar el fitxer, de manera que no faci falta res de forma externa (com per exemple fonts). El problema radica en què les signatures electròniques no estan incloses a l'estàndard PDF-A, ja que no garanteixen que siguin vàlides en el

futur. Així doncs, com complir un estàndard a mitges no serveix de res, desestimem la utilització de PDF-A.

- Igual que a la web, s'ha d'anar en compte amb alguns caràcters especials. Si enviem alguna sol·licitud amb un símbol de % l'hem d'escapar o ens pot donar un error de codificació al generar el document.

6.5.3 Base de dades / Notificacions periòdiques

- Des del principi de la segona iteració es va decidir utilitzar events SQL a la base de dades per la funcionalitat de les notificacions periòdiques. Tot i que a l'entorn de proves estava així implementat, la versió de la base de dades no dona suport per events i per tant hem hagut de buscar l'alternativa d'implementar (a la tercera iteració) la consulta de la caducitat a cada accés a un formulari.

- Per tercera vegada, hem hagut de sincronitzar la codificació de la base de dades amb la de la resta del sistema, UTF-8).

6.5.4 Applet de signatura electrònica

- Un problema greu de seguretat, una vegada introduït el PIN a l'applet aquest l'emmagatzemava i no era necessari la seva introducció vàlida en signatures posteriors, si no es treia la targeta. Això s'arregla utilitzant el mètode *logout* de la clase *AuthProvider* [5] de Java, en altre cas guarda el PIN en memòria fins que la màquina virtual finalitza la seva execució.

- El codi original de l'applet dona per suposat l'us exclusiu de les targetes criptogràfiques provinents de EC-ACC. Quan veiem que no funciona amb el DNI electrònic acabem descobrint una funció de càlcul de *hash* implementada com una llista precalculada. Com el cas del DNI electrònic no apareix, no es calculava bé. S'ha hagut de modificar la funció perquè realment faci el càlcul.

- El signat de documents PDF donava també per suposat l'aplicació de l'algoritme de *hash* SHA-1, ignorant l'escollit com a paràmetre, per tal de maximitzar l'eficiència amb una implementació de SHA-1 optimitzada (a jutjar pels comentaris del codi). Tot i la possible pèrdua d'eficiència, considerem més important l'extensibilitat del sistema per l'aplicació de les versions més modernes de SHA en el futur, així que farem els canvis per oferir la versatilitat d'algoritmes inicial.

- Per la mateixa causa que l'anterior, l'applet no permetia l'accés a un tipus de targeta (UPC o DNIe) després d'usar prèviament l'altra. Això era degut a que el proveidor de serveis criptogràfics (clase java *AuthProvider* [5]) no era eliminat després de la signatura i es conservava entre signatures (la màquina virtual de Java

continua executant-se una vegada ha finalitzat l'applet). Ho hem arreglat eliminant el *provider* després de cada signatura.

- A l'hora de mostrar els certificats disponibles per la signatura, l'applet original mostrava els que tenien els permisos d'ús (*usages*) de signatura digital o els de no repudiació. Com algunes autoritats consideren que el permís de signatura digital inclou l'autenticació (aquesta és implementada mitjançant la signatura digital d'un missatge enviat pel autenticador), ho modifiquem perquè apareguin només els que ens garanteixi no repudiació (el qual inclou, implícitament, l'ús com a signatura digital). D'aquesta manera surten els certificats apropiats també en el cas del DNIe.

- El sistema de *cache* implementat per les llistes de revocació estava mal implementat; no es comprovava el temps de creació de la CRL per la caducitat. Com la freqüència de buidat dels directoris temporals no està definit, podríem tenir una CRL molt antiga i afegir-la a la signatura. S'ha modificat perquè es comprovi i caduqui de forma setmanal.

- Des d'un punt de vista d'usabilitat l'applet permet signar amb un certificat invàlid (per revocació o caducitat). Tot i que això no és un problema de seguretat (òbviament la signatura serà invàlida), és un problema deixar creure a l'usuari que ha signat correctament. Ho hem modificat perquè retorni error en cas de certificat invàlid. (Es podria donar la situació en que no es considerés vàlid perquè no hi ha connexió al servidor OCSP per validar el certificat, cas en el que tampoc permetrem la signatura.)

- Per algun problema de les classes de Java (en concret la *X509Certificate* [44]), no ha estat possible extreure del certificat la URL del servidor OCSP per la validació (aconseguim tota la informació necessària però aquest no és mostrat en una llista completa de la informació que conté el certificat). Per remediar-ho afegirem les URLs pels dos tipus de targeta externament, com a fitxers de text, i hi accedirem mitjançant la classe *ResourceBundle* [43].

- Com ja hem comentat en el capítol de disseny, ens haviem plantejat utilitzar l'estàndard de signatura de llarga validació (PAdES-LTV). Per desgràcia, donada la recent aparició d'aquest tipus de signatures, cap software de validació actual hi dona suport encara, i en molts casos no és compatible amb signatures anteriors. Com signar documents d'una manera que encara no està reconeguda (i pot ser que tardi en ser-ho) és un comportament massa optimista per les nostres necessitats, hem decidit utilitzar el tipus PAdES-BES, que dona un suport més extès.

- De forma no massa crítica, la idea era que el servei de suport informàtic de la UPC, UPCnet, ens proporcionés un certificat per signar l'applet, expedit a nom del departament. Finalment això no ha pogut ser possible ja que redIRIS (la xarxa nacional de recerca espanyola) no proporciona aquest servei actualment. Així, no

tenim més remei que signar l'applet amb un certificat autosignat, tot esperant alguna alternativa més robusta en el futur.

6.5.5 DNI electrònic

- En la versió de Windows del driver del DNI electrònic, aquest demana una confirmació addicional (una nova finestra) per cada signatura que es realitzi. Això és una molèstia en el cas de la signatura en lot, on ens demana confirmació per cada document que haguem seleccionat. Tot i que és molest, no hi podem fer res.

- En el cas de la validació OCSP (la que utilitzem nosaltres) existeix un problema entre la infraestructura del DNI electrònic i el software lector de PDFs més extès actualment, Adobe Reader [37]. El problema fa que les signatures amb DNIe surtin com a no vàlides en aquest software tot i que ho són. La causa és la següent: segons l'estàndard OCSP [1], les respostes OCSP han de ser signades per un dels següents:

  1. L'emissor del certificat.
  2. Un certificat especial emès per l'emissor del certificat a validar, especialitzat en signar respostes OCSP.
  3. Un certificat en el què es confiï de forma incondicional (autoritat arrel).

  Com ho hem pogut veure a la figura 4.2, el DNIe no compleix cap dels tres requisits, així que la resposta OCSP obtinguda no és considerada vàlida. L'única alternativa és buscar altre software de validació, com el mostrat a l'annex B.

Capítol 7

Proves

En aquest apartat descriurem una sèrie de casos de test (*test cases*), que serveixen per comprovar el bon funcionament del sistema. Aquests estaran dividits en actors i altres àrees, i finalment realitzarem uns casos de test més avançats on provarem detalls més subtils, com possibles problemes de seguretat.

7.1 General

7.1.1 Accedir mitjançant nom d'usuari UPC (correcte)

Descripció: Accés a la web mitjançant l'autenticació per mitjà del nom d'usuari i password UPC, amb unes credencials correctes.

Resultat esperat: Ha de donar accés a tot l'apartat principal i sortir el nom del subjecte autenticat.

Resultat: Correcte

7.1.2 Accedir mitjançant nom d'usuari UPC (incorrecte)

Descripció: Accés a la web mitjançant l'autenticació per mitjà del nom d'usuari i password UPC, amb unes credencials incorrectes.

Resultat esperat: La web no ha de donar accés i ha de sortir un missatge d'error de credencials invàlides.

Resultat: Correcte

7.1.3 Accedir mitjançant carnet UPC

Descripció: Accés a la web mitjançant l'autenticació per mitjà del carnet UPC, amb el PIN correcte.

Resultat esperat: El navegador ens demana el PIN per accedir a la targeta, ens demana quin certificat volem utilitzar i finalment ens dona accés al sistema de la mateixa manera que amb les credencials UPC.

Resultat: Correcte, al menys en els navegadors de Internet Explorer, Firefox, Google Chrome i Safari (versions tant de Windows com Linux en tots ells). L'accés per Opera no està disponible, ja que actualment no hi ha suport per targetes [22], i l'accés en sistemes Apple no ha pogut ser comprovat per manca de recursos.

7.1.4 Accedir mitjançant carnet UPC (pin incorrecte)

Descripció: Accés a la web mitjançant l'autenticació per mitjà del carnet UPC, amb el PIN incorrecte.

Resultat esperat: El navegador no troba certificats disponibles i es dona un error d'autenticació.

Resultat: Correcte, tot i que segons la implementació del navegador aquest ens pot donar a escollir altres certificats instal·lats (com a fitxers). En tot cas no es dona accés, ja que només acceptem certificats provinents de AC RAIZ DNIE o EC-ACC.

7.1.5 Accedir mitjançant DNI electrònic

Descripció: Accés a la web mitjançant l'autenticació per mitjà del DNIe, amb el PIN correcte.

Resultat esperat: Ídem al cas del carnet UPC

Resultat: Correcte

7.1.6 Accedir mitjançant DNI electrònic (pin incorrecte)

Descripció: Accés a la web mitjançant l'autenticació per mitjà del DNIe, amb el PIN incorrecte.

Resultat esperat: Ídem al cas del carnet UPC

Resultat: Correcte, amb les mateixes aclaracions que al carnet UPC

7.2 Sol·licitant

7.2.1 Realitzar sol·licitud (formulari intern)

Descripció: El sol·licitant entra al formulari, omple les dades, veu el document PDF generat i finalment confirma la sol·licitud (possiblement firmant abans).

Resultat esperat: El sistema respòn que la sol·licitud ha estat enviada correctament.

Resultat: Correcte

7.2.2 Realitzar sol·licitud (error al sistema)

Descripció: El sol·licitant entra al formulari, omple les dades, veu el document PDF generat i finalment confirma la sol·licitud (possiblement firmant abans). El sistema no pot enregistrar les dades (perquè, per exemple, la base de dades no està disponible).

Resultat esperat: El sistema respòn que la sol·licitud no ha pogut ser registrada.

Resultat: Correcte

7.2.3 Realitzar sol·licitud (formulari extern)

Descripció: El sol·licitant entra al formulari, es descarrega el document base, l'omple (de forma externa al sistema), el converteix a PDF i l'envia al sistema, on finalment confirma la sol·licitud.

Resultat esperat: El sistema respòn que la sol·licitud ha estat enviada correctament.

Resultat: Correcte

7.2.4 Realitzar sol·licitud (formulari extern, fitxer no PDF)

Descripció: Ídem al cas anterior, però l'usuari envia un document de format diferent a PDF al sistema (per equivocació o desconeixement).

Resultat esperat: El sistema respòn que la sol·licitud no ha estat processada correctament degut a què el document enviat no era un PDF vàlid.

Resultat: Correcte

7.2.5 Realitzar sol·licitud (camp obligatori)

Descripció: El sol·licitant entra un formulari, l'omple però deixa en blanc un camp obligatori abans de passar a la pàgina següent.

Resultat esperat: El sistema avisa de què el camp afectat no pot estar en blanc.

Resultat: Correcte

7.2.6 Realitzar sol·licitud (camp dependent)

Descripció: El sol·licitant entra a un formulari, l'omple i selecciona una casella anterior a un camp dependent.

Resultat esperat: El camp dependent, prèviament no disponible, passa a estar disponible.

Resultat: Correcte

7.2.7 Realitzar sol·licitud (seccions)

Descripció: El sol·licitant entra a un formulari amb més d'una secció definida.

Resultat esperat: La pàgina del formulari és mostrada en diverses seccions degudament agrupades.

Resultat: Correcte

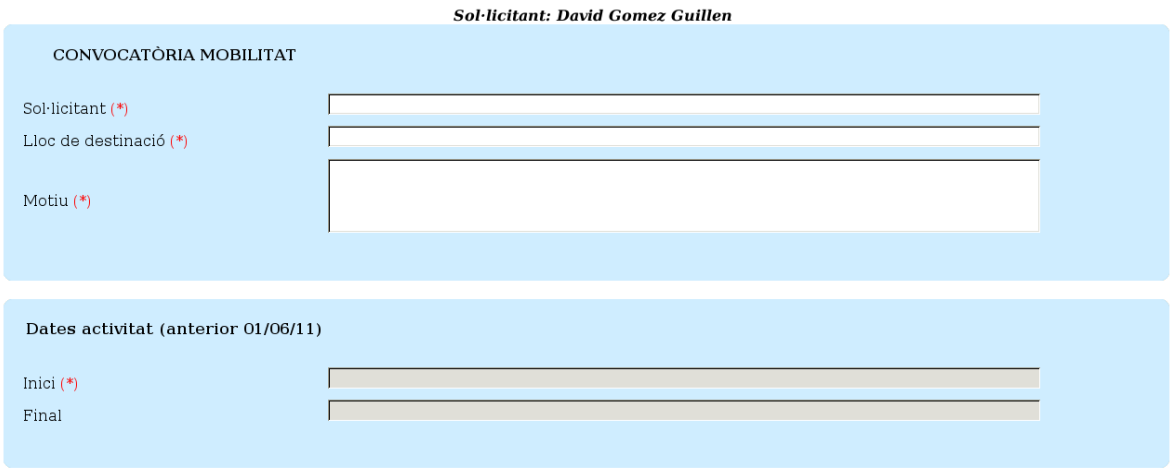

Figura 7.1: Seccions d'un formulari.

7.2.8 Realitzar sol·licitud (pàgines)

Descripció: El sol·licitant entra a un formulari amb més d'una pàgina definida, omple la pàgina inicial i confirma aquestes dades.

Resultat esperat: S'ha de mostrar la següent pàgina del formulari.

Resultat: Correcte

7.2.9 Realitzar sol·licitud (tipus de camps)

Descripció: El sol·licitant entra un formulari amb diversos tipus de camps definits.

Resultat esperat: Cada camp és mostrat, i posteriorment tractat, de la forma adequada per ser posteriorment traslladat a text.

Resultat: Correcte

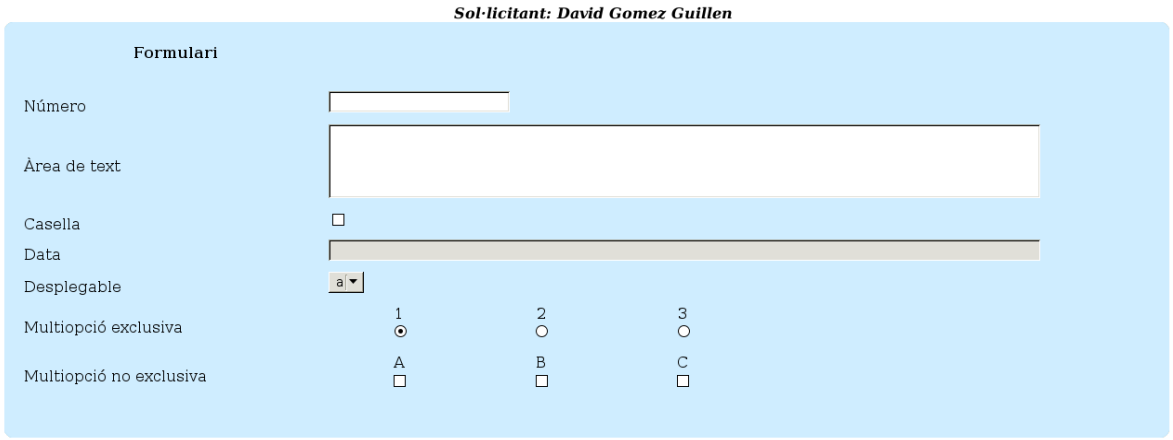

Figura 7.2: Representació dels tipus de camps.

7.2.10 Realitzar sol·licitud (signar sol·licitud)

Descripció: El sol·licitant omple un formulari i signa la sol·licitud abans d'enviar-la.

Resultat esperat: El sol·licitant rep la confirmació de que s'ha processat la sol·licitud correctament, i el sistema actualitza l'estat de la base de dades i el sistema de fitxers amb el document PDF.

Resultat: Correcte en els navegadors de Internet Explorer, Firefox i Opera. Els navegadors Google Chrome i Safari, ambdos basats en el motor de renderitzat web Webkit, tenen *bugs* amb la comunicació Java - Javascript que impossibiliten el pas del document a l'applet, fent impossible la signatura [42].

7.2.11 Realitzar sol·licitud (enviar sol·licitud sense signar)

Descripció: El sol·licitant omple un formulari i envia la sol·licitud sense signar-la.

Resultat esperat: El sol·licitant rep la confirmació de que s'ha processat la sol·licitud correctament, però avisa de que no serà vàlida fins que la signi. El sistema actualitza l'estat de la base de dades i el sistema de fitxers amb el document PDF.

Resultat: Correcte

7.2.12 Veure sol·licitud feta (HTML)

Descripció: El sol·licitant entra a l'apartat de sol·licituds realitzades i en selecciona una (de interna) per veure-la en una finestra del navegador.

Resultat esperat: El navegador obre una nova finestra amb la informació de la seva sol·licitud, així com el seu estat actual de signatura.

Resultat: Correcte

7.2.13 Veure sol·licitud feta (PDF)

Descripció: El sol·licitant entra a l'apartat de sol·licituds realitzades i en selecciona una per descarregar-se-la en format PDF.

Resultat esperat: El navegador segueix el seu procediment estàndard per la descàrrega de fitxers, on el fitxer és el document PDF.

Resultat: Correcte

7.2.14 Signar sol·licituds fetes (una)

Descripció: El sol·licitant entra a l'apartat de sol·licituds realitzades i selecciona una sol·licitud per signar-la.

Resultat esperat: Una vegada signada i enviada el navegador refresca la pàgina de sol·licituds realitzades i mostra el nou estat de la sol·licitud signada. El primer càrrec en signar és avisat d'una nova sol·licitud.

Resultat: Correcte (veure 7.2.10)

7.2.15 Signar sol·licituds fetes (més d'una)

Descripció: Ídem a l'anterior, però amb més d'una sol·licitud.

Resultat esperat: Ídem a l'anterior.

Resultat: Correcte, tot i que amb el DNI electrònic es mostra un diàleg de confirmació per cada document signat. Veure també 7.2.10.

7.2.16 Signar sol·licituds fetes (cap)

Descripció: Ídem a l'anterior, però sense seleccionar cap sol·licitud.

Resultat esperat: El navegador avisa de què no hi ha cap sol·licitud seleccionada per signar.

Resultat: Correcte (veure 7.2.10)

7.3 Càrrec

7.3.1 Veure totes les sol·licituds d'un formulari

Descripció: El càrrec entra a l'apartat per veure totes les sol·licituds d'un formulari.

Resultat esperat: Es mostren totes les sol·licituds del formulari junt amb el seu estat actual (acceptada, rebutjada o a l'espera d'algú).

Resultat: Correcte

7.3.2 Veure sol·licituds pendents

Descripció: El càrrec entra a l'apartat per veure les sol·licituds pendents d'un formulari.

Resultat esperat: Es mostren totes les sol·licituds pendents del formulari i la interfície per signar-les en lot.

Resultat: Correcte

7.3.3 Veure sol·licituds pendents (HTML)

Descripció: Ídem al mateix cas del sol·licitant.

Resultat esperat: Ídem al mateix cas del sol·licitant.

Resultat: Correcte

7.3.4 Veure sol·licituds pendents (PDF)

Descripció: Ídem al mateix cas del sol·licitant.

Resultat esperat: Ídem al mateix cas del sol·licitant.

Resultat: Correcte

7.3.5 Signar sol·licituds pendents (una)

Descripció: Ídem al mateix cas del sol·licitant.

Resultat esperat: Ídem al mateix cas del sol·licitant, però si és l'últim càrrec en signar es notifica al càrrec.

Resultat: Correcte (veure 7.2.10)

7.3.6 Signar sol·licituds pendents (més d'una)

Descripció: Ídem al mateix cas del sol·licitant.

Resultat esperat: Ídem al cas anterior.

Resultat: Correcte (veure 7.2.10)

7.3.7 Signar sol·licituds pendents (cap)

Descripció: Ídem al mateix cas del sol·licitant.

Resultat esperat: Ídem al mateix cas del sol·licitant.

Resultat: Correcte (veure 7.2.10)

7.3.8 Rebutjar sol·licitud

Descripció: El càrrec entra a l'apartat de sol·licituds pendents i selecciona una sol·licitud per rebutjar-la, afegint la raó.

Resultat esperat: Una vegada confirmat el rebuig, el navegador torna a la pàgina de sol·licituds pendents sense la sol·licitud rebutjada. Per altra banda el sistema actualitza l'estat de la sol·licitud i el sol·licitant és notificat del rebuig.

Resultat: Correcte

7.4 Gestor

7.4.1 Crear formulari (intern)

Descripció: El gestor demana la creació d'un nou formulari intern.

Resultat esperat: Es crea un nou formulari amb nom "Nou formulari" i amb un camp de tipus text. Seguidament es redirecciona a la pàgina d'edició d'aquest formulari.

Resultat: Correcte

7.4.2 Crear formulari (extern)

Descripció: El gestor demana la creació d'un nou formulari extern.

Resultat esperat: Es crea un nou formulari extern amb nom "Nou formulari", sense document base. Seguidament es redirecciona a la pàgina d'edició d'aquest formulari.

Resultat: Correcte

7.4.3 Veure sol·licituds d'un formulari

Descripció: El gestor demana la llista completa de sol·licituds d'un formulari concret.

Resultat esperat: Es mostra la llista de sol·licituds del formulari junt amb la interfície per esborrar-les i veure-les.

Resultat: Correcte

7.4.4 Esborrar formulari (confirmant)

Descripció: El gestor demana l'esborrat d'un formulari en concret i confirma l'acció.

Resultat esperat: S'esborra el formulari del sistema i es redirecciona a la pàgina principal del gestor amb els formularis actualitzats.

Resultat: Correcte

7.4.5 Esborrar formulari (avortant)

Descripció: El gestor demana l'esborrat d'un formulari en concret però avorta en el moment de confirmar.

Resultat esperat: El sistema no pateix canvis i es redirecciona a la pàgina principal del gestor.

Resultat: Correcte

7.4.6 Obtenir sol·licitud buida

Descripció: El gestor demana una sol·licitud buida d'un formulari.

Resultat esperat: El sistema ofereix la descàrrega d'un fitxer PDF buit en el cas dels formularis interns i el document base en el cas dels externs.

Resultat: Correcte

7.4.7 Editar informació general d'un formulari

Descripció: El gestor accedeix a l'apartat d'edició del formulari i realitza canvis en la informació general.

Resultat esperat: El sistema emmagatzema aquests canvis que es veuran reflexats la pròxima vegada que algú accedeixi en aquell formulari.

Resultat: Correcte

7.4.8 Editar camps d'un formulari intern

Descripció: El gestor accedeix a l'apartat d'edició del formulari i realitza canvis (vàlids) en els seus camps.

Resultat esperat: Ídem al cas anterior.

Resultat: Correcte

7.4.9 Editar camps d'un formulari intern (camp dependent)

Descripció: El gestor marca un camp com a dependent, sense que existeixi cap camp de tipus casella anteriorment (en l'ordre).

Resultat esperat: El sistema avisa que no és possible marcar aquell camp com a dependent i no guarda els canvis.

Resultat: Correcte

7.4.10 Editar camps d'un formulari intern (camp sense nom)

Descripció: El gestor edita un camp i li posa un nom en blanc.

Resultat esperat: El sistema avisa que tot camp ha de tenir un nom i no guarda els canvis.

Resultat: Correcte

7.4.11 Editar camps d'un formulari intern (valors buits)

Descripció: El gestor edita un camp de tipus desplegable o multiopció i deixa els valors possibles en blanc.

Resultat esperat: El sistema avisa que els camps d'aquests tipus han de tenir una llista de valors possibles no nul·la.

Resultat: Correcte

7.4.12 Editar document base d'un formulari extern

Descripció: El gestor accedeix a l'apartat d'edició d'un formulari extern i realitza canvis en el document base.

Resultat esperat: El sistema emmagatzema aquests canvis que es veuran reflexats la pròxima vegada que algú accedeixi en aquell formulari.

Resultat: Correcte

7.4.13 Editar procés d'un formulari

Descripció: El gestor accedeix a l'apartat d'edició del formulari i realitza canvis en el procés administratiu.

Resultat esperat: El sistema emmagatzema aquests canvis que es veuran reflexats la pròxima vegada que algú accedeixi en aquell formulari.

Resultat: Correcte

7.4.14 Duplicar formulari

Descripció: El gestor demana la creació d'un nou formulari amb la mateixa estructura que un d'existent.

Resultat esperat: El sistema crea un nou formulari còpia de l'existent i refresca la pàgina actual amb els canvis fets.

Resultat: Correcte

7.4.15 Realitzar sol·licitud en mode de prova

Descripció: El gestor demana visualitzar un formulari.

Resultat esperat: Es mostra la mateixa visualització i procés de sol·licituds que veuria un sol·licitant, però encara que enviï sol·licituds no es produeixen canvis en el sistema.

Resultat: Correcte

7.4.16 Editar nom de càrrec

Descripció: El gestor posa un nou nom al portador d'un càrrec.

Resultat esperat: El sistema guarda els canvis i refresca la pàgina actual amb la nova informació.

Resultat: Correcte

7.4.17 Afegir persona a un càrrec

Descripció: El gestor afegeix una nova persona i correu electrònic a un càrrec.

Resultat esperat: Ídem a l'anterior.

Resultat: Correcte

7.4.18 Esborrar persona d'un càrrec

Descripció: El gestor elimina una persona existent d'un càrrec.

Resultat esperat: Ídem a l'anterior.

Resultat: Correcte

7.5 Signatures digitals

7.5.1 Validar signatura (Adobe Reader i UPC)

Descripció: Una persona vol validar una signatura realitzada amb el carnet UPC en una sol·licitud en format PDF.

Resultat esperat: Si la signatura és correcta, el software d'Adobe l'ha de marcar com a vàlida.

Resultat: Correcte

7.5.2 Validar signatura (Adobe Reader i DNIe)

Descripció: Una persona vol validar una signatura realitzada amb el DNI electrònic en una sol·licitud en format PDF.

Resultat esperat: Si la signatura és correcta, el software d'Adobe l'ha de marcar com a vàlida.

Resultat: Incorrecte, pels problemes que hem comentat a l'etapa d'implementació amb la validació OCSP del DNI electrònic. Per solucionar-ho haurem d'utilitzar un altre software de validació (annex B).

7.5.3 Signar (amb certificat revocat)

Descripció: Una persona, en qualsevol dels escenaris possibles més amunt, signa un document utilitzant un certificat revocat.

Resultat esperat: El sistema ha de retornar error i avisar a l'usuari que el certificat ha estat revocat.

Resultat: Correcte. Cal dir que, com no hem disposat d'un certificat revocat, el que hem fet és aconseguir un número de sèrie d'un certificat present a la CRL i modificar el codi perquè realitzés la petició OCSP amb aquell número.

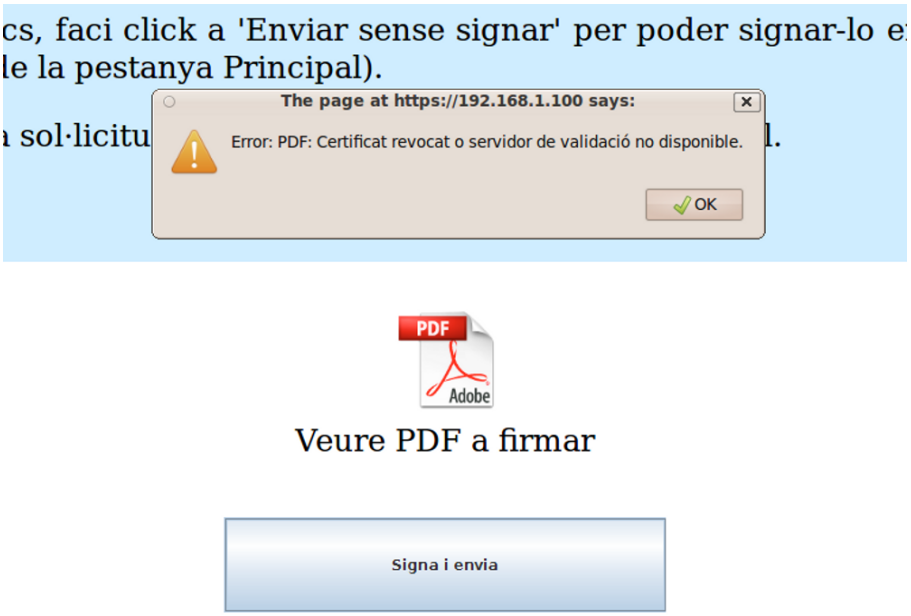

Figura 7.3: Error mostrat a l'usuari quan el seu certificat ha estat revocat (o el servidor OCSP té dificultats).

7.5.4 Validar signatura (document modificat)

Descripció: Una persona vol validar una signatura realitzada amb el DNI electrònic en una sol·licitud en format PDF, però aquesta ha estat modificada (intencionadament o no).

Resultat esperat: La signatura s'ha de detectar com a no vàlida perquè el document ha estat modificat.

Resultat: Correcte

7.5.5 Signar (driver en ruta típica)

Descripció: El sol·licitant es disposa a signar, amb el controlador a una ruta típica, i l'applet de signatura el cerca de forma automàtica.

Resultat esperat: L'applet troba el driver i continua el procés de signatura.

Resultat: Correcte, tot i que s'han donat errors molt puntuals que no han estat possibles de reproduir, possiblement deguts a algun problema amb la implementació de l'accés a targetes de Java.

7.5.6 Signar (driver en ruta no típica)

Descripció: El sol·licitant es disposa a signar, amb el controlador a una ruta no típica, i l'applet de signatura el cerca de forma automàtica.

Resultat esperat: L'applet no troba el driver i, pregunta a l'usuari per la ruta correcta (figura 7.4). Una vegada especificada, es continua el procés.

Resultat: Correcte

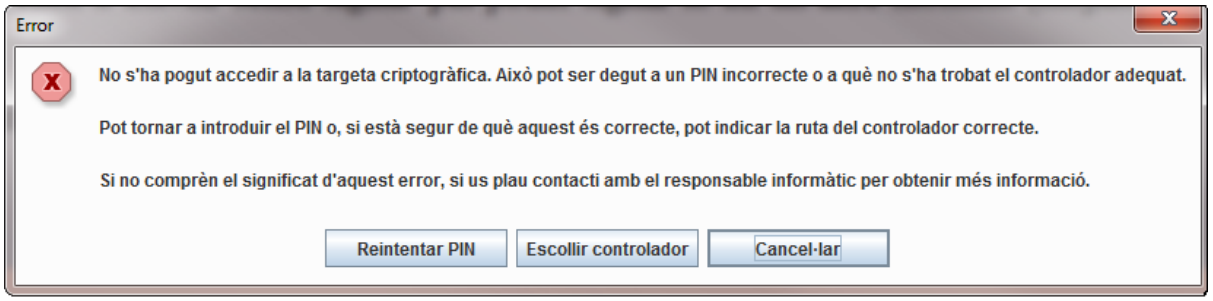

Figura 7.4: Controlador de la targeta no trobat. Es demana la ruta del driver, o bé reintroduir el PIN.

7.5.7 Signar (driver en ruta no típica, firmes posteriors)

Descripció: El sol·licitant realitza l'escenari anterior i es torna a disposar a signar.

Resultat esperat: L'applet no troba el driver de forma automàtica, però recorda la signatura anterior i agafa el mateix que ha funcionat en ocasions anteriors, continuant el procés de signatura.

Resultat: Correcte

7.6 Proves avançades

7.6.1 Accedir a un formulari sense haver-se identificat

Descripció: Obtenint la URL d'accés directe a un formulari, intentem accedir-hi sense estar autenticats.

Resultat esperat: Se'ns redirecciona a la pàgina d'autenticació.

Resultat: Correcte

7.6.2 Accedir a un apartat al que no es té permís

Descripció: Obtenint la URL d'accés a un apartat al que no tenim accés (e.g. un usuari estàndard a l'apartat de gestor), intentem accedir-hi.

Resultat esperat: Se'ns redirecciona a la pàgina principal (sol·licitant).

Resultat: Correcte

7.6.3 Intercepció de comunicació SSL

Descripció: Des del mateix ordinador amb el que ens comuniquem amb el sistema simulem una intercepció de dades amb un capturador de paquets HTTP.

Resultat esperat: El protocol SSL ens ha de fer impossible extreure informació de la comunicació.

Resultat: Correcte

7.6.4 Accés no autoritzat a PDF

Descripció: Obtenint la URL d'accés a una sol·licitud, intentem accedir-hi sense permís.

Resultat esperat: El sistema ens dona un error confirmant que no tenim accés al fitxer.

Resultat: Correcte

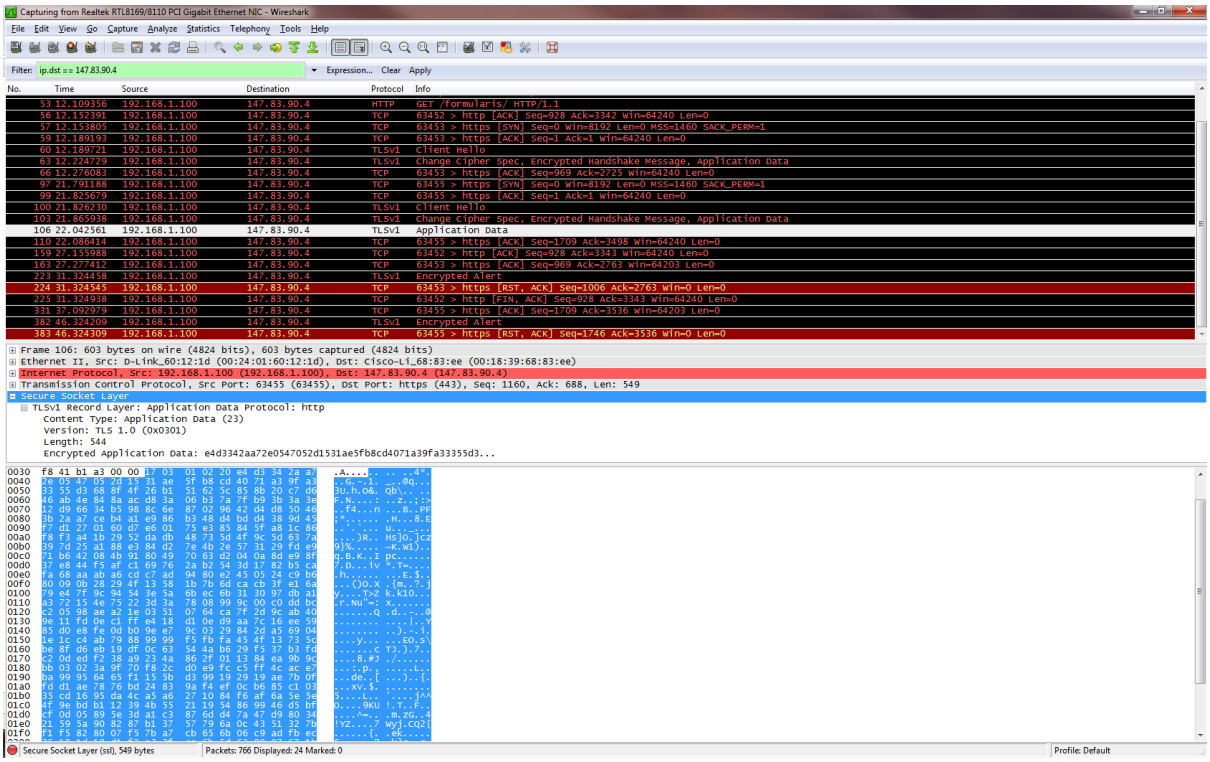

Figura 7.5: Captura de la comunicació amb el servidor. Veiem que la comunicació (xifrada) comença en la línia blanca; abans d'això podem veure la negociació dels paràmetres de la connexió.

7.6.5 Modificació de la sol·licitud entre pàgines

Descripció: El sol·licitant omple una pàgina d'un formulari, la confirma, però després tira enrere amb el navegador.

Resultat esperat: El sistema ofereix la pàgina anterior amb la informació que havia posat, i quan s'acabi d'omplir la informació final inclourà els canvis que hagi pogut realitzar l'usuari.

Resultat: Correcte

Capítol 8

Consideracions finals

El camp de la informàtica, inclosa la criptografia, avança molt ràpidament. Per tal de tenir un sistema útil no n'hi ha prou en que funcioni avui, sinó que s'ha de pensar de quina manera es podrà ampliar en el futur, quan la tecnologia inevitablement evolucioni. Amb el projecte ja implementat, en aquest capítol repassarem quins possibilitats d'extensió oferim.

8.1 Implementació de nous estàndards

Com hem comentat diverses vegades al llarg del procés de desenvolupament, els estàndards criptogràfics que compleixen amb les directives europees adequades són uns quants. Com a requeriment inicial ens hem decidit pel format PDF per una fàcil distribució tant dels documents com de les seves signatures.

És per això que ha estat una llàstima no poder integrar l'estàndard de signatura més complet de la familia PAdES (PAdES-LTV), la qual cosa hagués pogut prolongar la vida útil de les signatures. La manca de suport en les implementacions dels productes actuals fa que hagués estat tremendament arriscat deixar el sistema amb aquest tipus de signatures, sense manera de poder validar-les. De tota manera, la majoria de companyies que ofereixen serveis de validació de signatura planegen començar a oferir-hi suport durant el 2011.

Per això és important deixar el sistema suficientment obert per l'ampliació del tipus de signatura de bàsica a llarga validació. En el nostre cas ho hem fet, i de dues maneres diferents:

Per una banda hem afegit el codi necessari a l'applet per signar amb LTV, és a dir, la generació del *document timestamp* i el *DSS*. Òbviament no hem pogut provar el codi, així que no podem assegurar que sigui correcte, però en tot cas és un començament.

En segon lloc, la naturalesa de les signatures PAdES-LTV permet "l'actualització" d'una signatura simple a la seva versió més avançada. Així, en el moment en què sigui factible utilitzar-les no només podrem modificar el sistema de generació de signatures, sinó els mateixos documents signats podran ser actualitzats (sempre que continuin sent vàlids).

8.2 Algoritmes criptogràfics i seguretat de les claus

És important també tenir en compte que les signatures electròniques no poden ser mai perpètues: encara que els certificats no caduquéssin els algoritmes i les claus d'aquests algoritmes poden ser i seran "trencats". La única incògnita és quan: la clau més gran trencada fins ara ha estat de 768 bits, utilitzant un sistema distribuit de factorització de nombres [33], però es van fent avenços contínuament.

Com la dificultat creix exponencialment respecte el tamany de la clau, sembla que 1024 bits (mida de la clau del carnet UPC) són segurs de moment. Alguns experts, però, creuen que també es podrà trencar en un futur relativament proper. Els 2048 bits del DNI electrònic semblen encara estar bastant lluny per comprometre la seguretat de l'algoritme, raó per la que la UPC està planejant renovar en el futur els carnets amb claus d'aquesta mida.

Similarment, també hem de tenir en compte la fortalesa dels algoritmes de *hashing*. Per compatibilitat amb sistemes antics actualment encara s'utilitza bastant el SHA-1, tot i que s'han fet bastants avenços per comprometre'l. En aquest projecte l'utilitzem perquè el carnet UPC actualment només dona suport per aquest algoritme, però quant s'uniformitzi l'ús d'altres més avançats s'haurà d'actualitzar el sistema (de forma fàcil, com hem comentat: només cal modificar un simple paràmetre per disposar de qualsevol algoritme de la familia de SHA-2, fins a SHA-512). Per acabar amb aquest punt, també cal comentar que el 27 de novembre del 2007 va començar el concurs per determinar la futura familia d'algoritmes SHA-3, el successor de SHA-2 [21].

Una altra dificultat per la seguretat d'aquests algoritmes en un futur més llunyà és la computació quàntica. Sense entrar en detalls dels seus fonaments, existeix un algoritme que si és possible implementar permetria la factorització de nombres en temps polinòmic: l'algoritme de Shor [12] [45]. Afortunadament per moltes aplicacions criptogràfiques, inclosa la nostra, la computació quàntica es troba encara en un estat fundamentalment teòric a causa de l'extrema dificultat de la implementació d'un sistema d'aquestes característiques. Si algun dia s'arriba a crear, però, s'haurien de rebutjar tots els algoritmes que es basen en la dificultat de la factorització de nombres i buscar paradigmes alternatius de criptografia (que, curiosament, la pròpia computació quàntica també ofereix [8]).

8.3 Seguretat en l'ús de targetes criptogràfiques

Relacionat amb la criptografia tenim l'ús de les targetes criptogràfiques. La introducció d'aquestes ha estat un pas endavant en la creació de signatures digitals segures, evitant problemes de claus secretes compromeses. Tot i això, existeixen alguns problemes d'usabilitat tal i com estan implementades avui en dia:

Per una banda hi ha el problema que tota la seguretat de la targeta criptogràfica no serveix per res si el teu PIN es veu compromès. Per fer això és bastant fàcil per un atacant inserir un *key logger* a l'ordinador del client per tal de capturar les tecles premudes, corresponents al PIN. Alguns fabricants han aconseguit prevenir això mitjançant teclats amb una part numèrica segura, especialment dissenyat per aquests propòsits. En un teclat normal, però, pot ser relativament fàcil que l'accés a la targeta es vegi compromès.

Finalment hi ha un tema més subtil: el coneixement de l'usuari del que realment està signant. Cada vegada que l'applet demana confirmació per una signatura, realment no sabem a què ens estem comprometent; l'usuari ha de confiar implícitament en què el sistema no l'enganya. I com la signatura realment només necessita el *hash*, i per definició no podem saber d'on ha sortit, no tenim manera d'estar segurs del contingut signat. per combatre això hi ha gent que ha començat a proposar un sistema WYSIWYS (*What You See Is What You Sign*), però seria difícil d'implementar per raons tècniques (com tractem els diferents formats, com garantim la seva visualització, què és el que realment estem signant (representació física, lògica, semàntica, ...), etc...) [30].

8.4 Estandardització del format visual dels documents PDF

Deixant de banda la criptografia, també hem de pensar en el futur del format PDF. Com hem dit, des de 2008 és un estàndard obert, però això no garanteix que els visualitzadors futurs siguin compatibles amb les primeres versions o, si ho són, que no es produeixin canvis en la manera en què es presentin. Per garantir una vista consistent al llarg de les versions, es va publicar l'estàndard PDF-A, un subconjunt del format PDF que diu quines condicions ha de complir un PDF perquè la seva presentació estigui completament definida.

Com també hem dit, PDF-A no dona suport per la signatura electrònica, ja que òbviament no podem garantir que una signatura tingui les mateixes qualitats ara que en el futur. Per això no l'hem implementat, però hem minimitzat el contingut del PDF per tal d'evitar problemes a l'hora de visualitar-lo en el futur.

8.5 Optimització de la base de dades

Amb l'esquema de la base de dades que hem dissenyat hem procurat minimitzar la mida de la informació emmagatzemada, augmentant així també l'eficiència de la transmissió de dades. Tot i això, en cas d'utilitzar aquesta base de dades en un entorn més massiu s'hauria de comprovar l'eficiència amb més exactitud. MySQL és un gestor de base de dades prou bo, però per aplicacions d'alt rendiment és possible que altres solucions siguin més apropiades.

Fins i tot mantenint el mateix gestor possiblement es podria aconseguir una lleugera millora d'eficiència. En contrapartida, possiblement perjudicaria la versatilitat i extensibilitat del sistema, així que no hem aprofundit massa en la optimització de la base de dades. El que s'ha de tenir present és que és possible fer-ho si és necessari.

8.6 Repositori de documents extern

Seguint amb la idea d'expandir el sistema a un àmbit més gran, un punt que podria necessitar modificacions és el repositori de documents PDF. Tal com ho hem dissenyat, els fitxers es troben dins del mateix servidor web, el que pot ser un inconvenient en un sistema més modular. Hi havia diverses alternatives:

- Com hem comentat en el disseny, sempre existeix la possibilitat d'integrar els documents dins la base de dades. El problema d'aquesta opció, com hem notat, és que es dona més càrrega de treball a la base de dades, la qual cosa pot donar problemes en un entorn amb moltes sol·licituds. A més, hem comprovat que la latència és menor si accedim als fitxers de forma local al disc, probablement degut a problemes de *cache* a la BD (al carregar PDFs es buida la *cache* d'altra informació més utilitzada i important).
- L'altra alternativa seria utilitzar un repositori extern, de manera que desacoblem els documents de la resta de la web. Tot i això seguim tenint un dels problemes anteriors: la latència elevada. També hi ha un problema més important; si permetem l'accés extern hem de restringir els permisos dels fitxers per impedir que un atacant hi accedeixi, i aquests permisos han de coincidir amb els del nostre sistema. Com els permisos estan inherentment acoblats amb la web, sembla bastant convenient integrar el repositori a la web.

En tot cas, per qualsevol canvi que es pugui fer a l'hora d'expandir el sistema, els canvis d'accés es veuran limitats només al *middleware*. Així, la transició resulta fàcil i indolora.

Capítol 9

Conclusió

Arribem al final del projecte, i en aquesta etapa hem de mirar enrere per veure quins objectius hem complert i quins podriem millorar.

Comparant la cronologia real del projecte (figura 9.1) amb la planificació inicial (2.10, a l'especificació), veiem que la primera iteració s'ha allargat una mica més, deixant menys temps per la tercera iteració. Afortunadament la reducció de temps en aquesta última no ha estat un problema per acabar el projecte de forma satisfactòria. També la redacció de la memòria ha suposat més temps de l'inicialment previst.

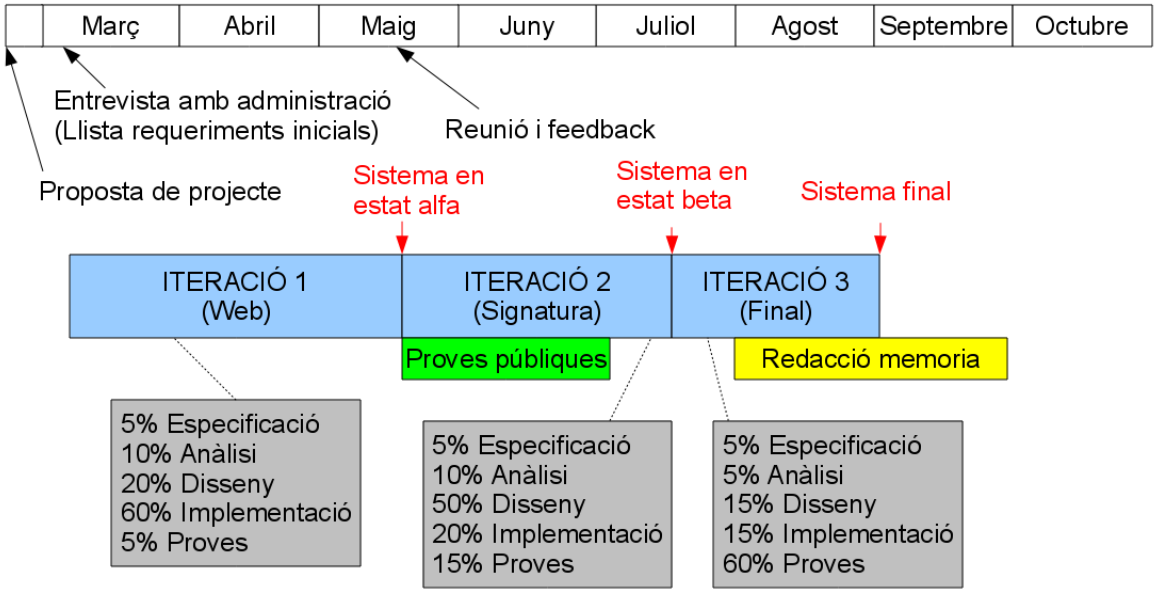

Figura 9.1: Cronologia del desenvolupament del projecte.

A més, com era d'esperar, el percentatge de cada etapa per iteració ha estat bastant

diferent de la planificada. En la primera iteració la fase d'implementació ha suposat un 60% del temps total (sobretot per qüestions referents a la programació de PHP / CSS). A la segona, l'etapa infraestimada ha estat el disseny, ja que hem hagut de documentar-nos profundament sobre els diferents tipus de signatura. També les hem hagut de provar, la qual cosa ha incrementat el pes de l'etapa de proves. Finalment, a la tercera iteració esperàvem molts canvis a implementar que al final han resultat no ser tants, així que la major part del temps la hem dedicat a proves.

Si tornem a calcular el cost de recursos humans a partir d'aquesta cronologia podem comprovar que ens hem ajustat bastant bé al pressupost inicial: fent els càlculs ens dóna 53.678,4 euros; molt semblant als 54.739,2 pressupostats inicialment.

Si mirem els problemes que hem patit al llarg del projecte, un dels contratemps més notables ha estat la manca de suport pel tipus de signatura més avançat que voliem implementar, PAdES-LTV. Tot i que pels usos actuals del departament no aportaria massa utilitat, la possibilitat de refrescar les signatures d'un document hagués pogut resultar útil a llarg termini.

En segon lloc també hi ha hagut el problema de que l'ús de la criptografia de clau pública no és encara suficientment homogeni; ha estat relativament difícil abstraure el procés de signar i validar respecte el tipus de targeta utilitzada. Amb el que hem vist durant el desenvolupament del projecte, queda clar que tant les infraestructures de clau pública com el software apropiat encara han de pulir alguns trets del seu funcionament perquè aquests sistemes funcionin de forma universal.

Tot i això, i segons la llista de requeriments (la declaració d'intencions del projecte), el projecte ha resultat més que satisfactori. A més de complir les exigències del departament de Matemàtica Aplicada II, hem creat un sistema fàcil d'utilitzar, ràpid i, sobretot, segur. Conseqüentment, totes aquestes característiques també passaran a formar part de l'administració del departament, que, com hem dit al principi d'aquest document, aportarà un benefici tant al personal d'administració com als propis membres del departament.

Appendix A

Exemples

En aquest annex mostrarem exemples de la informació criptogràfica que s'utilitza durant el procés de signatura. Tots ells estan representats en notació ASN.1.

A.1 Certificat X.509

```
  0 2097: SEQUENCE {
  4 1817:   SEQUENCE {
  8    3:     [0] {
 10    1:       INTEGER 2
       :       }
 13   16:     INTEGER 26 2A 2F 19 C3 5C 8B 32 4B 72 AD C7 18 28 9F 56
 31   13:     SEQUENCE {
 33    9:       OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 113549 1 1 5)
 44    0:       NULL
       :       }
 46  280:     SEQUENCE {
 50   11:       SET {
 52    9:         SEQUENCE {
 54    3:           OBJECT IDENTIFIER countryName (2 5 4 6)
 59    2:           PrintableString 'ES'
       :           }
       :         }
 63   59:       SET {
 65   57:         SEQUENCE {
 67    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
 72   50:           PrintableString
       :             'Agencia Catalana de Certificacio (NIF Q-0801176-'
       :             'I)'
```

```
        :           }
        :         }
124   52:       SET {
126   50:         SEQUENCE {
128    3:           OBJECT IDENTIFIER localityName (2 5 4 7)
133   43:           PrintableString
        :             'Passatge de la Concepcio 11 08008 Barcelona'
        :           }
        :         }
178   46:       SET {
180   44:         SEQUENCE {
182    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
187   37:           PrintableString 'Serveis Publics de Certificacio ECV-2'
        :           }
        :         }
226   53:       SET {
228   51:         SEQUENCE {
230    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
235   44:           PrintableString
        :             'Vegeu https://www.catcert.net/verCIC-2 (c)03'
        :           }
        :         }
281   31:       SET {
283   29:         SEQUENCE {
285    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
290   22:           PrintableString 'Universitats i Recerca'
        :           }
        :         }
314   14:       SET {
316   12:         SEQUENCE {
318    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
323    5:           PrintableString 'EC-UR'
        :           }
        :         }
        :       }
330   30:     SEQUENCE {
332   13:       UTCTime 10/02/2010 12:59:51 GMT
347   13:       UTCTime 10/02/2014 12:59:48 GMT
        :       }
362  318:     SEQUENCE {
366   11:       SET {
368    9:         SEQUENCE {
370    3:           OBJECT IDENTIFIER countryName (2 5 4 6)
```

```
375    2:           PrintableString 'ES'
       :           }
       :         }
379   45:       SET {
381   43:         SEQUENCE {
383    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
388   36:           TeletexString 'Universitat Politècnica de Catalunya'
       :           }
       :         }
426   12:       SET {
428   10:         SEQUENCE {
430    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
435    3:           PrintableString 'FIB'
       :           }
       :         }
440   60:       SET {
442   58:         SEQUENCE {
444    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
449   51:           TeletexString
       :             'Serveis Públics de Certificació CPISR-2 d'Estudi'
       :             'ant'
       :           }
       :         }
502   66:       SET {
504   64:         SEQUENCE {
506    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
511   57:           PrintableString
       :             'Vegeu https://www.catcert.net/verCPISR-2Estudian'
       :             'tUR (c)05'
       :           }
       :         }
570   22:       SET {
572   20:         SEQUENCE {
574    3:           OBJECT IDENTIFIER surname (2 5 4 4)
579   13:           PrintableString 'GOMEZ GUILLEN'
       :           }
       :         }
594   14:       SET {
596   12:         SEQUENCE {
598    3:           OBJECT IDENTIFIER givenName (2 5 4 42)
603    5:           PrintableString 'DAVID'
       :           }
       :         }
```

```
610   12:       SET {
612   10:         SEQUENCE {
614    3:           OBJECT IDENTIFIER title (2 5 4 12)
619    3:           PrintableString 'EST'
       :           }
       :         }
624   18:       SET {
626   16:         SEQUENCE {
628    3:           OBJECT IDENTIFIER serialNumber (2 5 4 5)
633    9:           PrintableString 'XXXXXXXX'
       :           }
       :         }
644   38:       SET {
646   36:         SEQUENCE {
648    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
653   29:           PrintableString 'CPISR-2 E DAVID GOMEZ GUILLEN'
       :           }
       :         }
       :       }
684  159:     SEQUENCE {
687   13:       SEQUENCE {
689    9:         OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
700    0:         NULL
       :         }
702  141:       BIT STRING, encapsulates {
706  137:         SEQUENCE {
709  129:           INTEGER

   . . .

       :                     [ Another 1 bytes skipped ]
841    3:           INTEGER 65537
       :           }
       :         }
       :       }
846  975:     [3] {
850  971:       SEQUENCE {
854   28:         SEQUENCE {
856    3:           OBJECT IDENTIFIER issuerAltName (2 5 29 18)
861   21:           OCTET STRING, encapsulates {
863   19:             SEQUENCE {
865   17:               [1] 'ec_ur@catcert.net'
       :               }
```

```
        :             }
        :           }
 884   70:         SEQUENCE {
 886    3:           OBJECT IDENTIFIER subjectAltName (2 5 29 17)
 891   63:           OCTET STRING, encapsulates {
 893   61:             SEQUENCE {
 895   35:               [1] 'david.gomez-guillen@est.fib.upc.edu'
 932   22:               [4] {
 934   20:                 SEQUENCE {
 936   18:                   SET {
 938   16:                     SEQUENCE {
 940    3:                       OBJECT IDENTIFIER serialNumber (2 5 4 5)
 945    9:                       PrintableString 'XXXXXXXXX'
        :                       }
        :                     }
        :                   }
        :                 }
        :               }
        :             }
        :           }
 956   14:         SEQUENCE {
 958    3:           OBJECT IDENTIFIER keyUsage (2 5 29 15)
 963    1:           BOOLEAN TRUE
 966    4:           OCTET STRING, encapsulates {
 968    2:             BIT STRING 6 unused bits
        :               '11'B
        :             }
        :           }
 972   41:         SEQUENCE {
 974    3:           OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
 979   34:           OCTET STRING, encapsulates {
 981   32:             SEQUENCE {
 983    8:               OBJECT IDENTIFIER clientAuth (1 3 6 1 5 5 7 3 2)
 993    8:               OBJECT IDENTIFIER emailProtection (1 3 6 1 5 5 7 3 4)
1003   10:               OBJECT IDENTIFIER
        :                 smartcardLogon (1 3 6 1 4 1 311 20 2 2)
        :               }
        :             }
        :           }
1015   17:         SEQUENCE {
1017    9:           OBJECT IDENTIFIER
        :             netscape-cert-type (2 16 840 1 113730 1 1)
1028    4:           OCTET STRING, encapsulates {
```

```
1030    2:             BIT STRING 5 unused bits
        :               '101'B
        :             }
        :           }
1034   29:         SEQUENCE {
1036    3:           OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
1041   22:           OCTET STRING, encapsulates {
1043   20:             OCTET STRING
        :               35 C8 3F 3A 95 FF A3 1B AF A0 B6 A5 65 7A 09 3C
        :               A5 94 0E 94
        :             }
        :           }
1065  305:         SEQUENCE {
1069    3:           OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
1074  296:           OCTET STRING, encapsulates {
1078  292:             SEQUENCE {
1082   20:               [0]
        :                 48 9F 8D 86 D1 CB F1 D6 8A 52 7B 7F 15 A2 0A 51
        :                 F8 97 FE 0B
1104  249:               [1] {
1107  246:                 [4] {
1110  243:                   SEQUENCE {
1113   11:                     SET {
1115    9:                       SEQUENCE {
1117    3:                         OBJECT IDENTIFIER countryName (2 5 4 6)
1122    2:                         PrintableString 'ES'
        :                         }
        :                       }
1126   59:                     SET {
1128   57:                       SEQUENCE {
1130    3:                         OBJECT IDENTIFIER organizationName (2 5 4 10)
1135   50:                         PrintableString
        :                   'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :                   'I)'
        :                         }
        :                       }
1187   40:                     SET {
1189   38:                       SEQUENCE {
1191    3:                         OBJECT IDENTIFIER
        :                           organizationalUnitName (2 5 4 11)
1196   31:                         PrintableString 'Serveis Publics de Certificacio'
        :                         }
        :                       }
```

```
1229   53:                     SET {
1231   51:                       SEQUENCE {
1233    3:                         OBJECT IDENTIFIER
        :                           organizationalUnitName (2 5 4 11)
1238   44:                         PrintableString
        :                       'Vegeu https://www.catcert.net/verarrel (c)03'
        :                         }
        :                       }
1284   53:                     SET {
1286   51:                       SEQUENCE {
1288    3:                         OBJECT IDENTIFIER
        :                           organizationalUnitName (2 5 4 11)
1293   44:                         PrintableString
        :                       'Jerarquia Entitats de Certificacio Catalanes'
        :                         }
        :                       }
1339   15:                     SET {
1341   13:                       SEQUENCE {
1343    3:                         OBJECT IDENTIFIER commonName (2 5 4 3)
1348    6:                         PrintableString 'EC-ACC'
        :                         }
        :                       }
        :                     }
        :                   }
        :                 }
1356   16:               [2]
        :                 06 A5 5F 3C B2 81 95 28 3F E0 49 E7 F9 31 9D 6C
        :               }
        :             }
        :           }
1374  270:         SEQUENCE {
1378    3:           OBJECT IDENTIFIER certificatePolicies (2 5 29 32)
1383  261:           OCTET STRING, encapsulates {
1387  257:             SEQUENCE {
1391  254:               SEQUENCE {
1394   13:                 OBJECT IDENTIFIER '1 3 6 1 4 1 15096 1 3 1 82 2 1'
1409  236:                 SEQUENCE {
1412   57:                   SEQUENCE {
1414    8:                     OBJECT IDENTIFIER cps (1 3 6 1 5 5 7 2 1)
1424   45:                     IA5String
        :                   'https://www.catcert.net/verCPISR-2EstudiantUR'
        :                     }
1471  174:                   SEQUENCE {
```

```
1474    8:                    OBJECT IDENTIFIER unotice (1 3 6 1 5 5 7 2 2)
1484  161:                    SEQUENCE {
1487  158:                      VisibleString
        :                   'Aquest .s un certificat personal reconegut d.ide'
        :                   'ntificaci. i signatura reconeguda de classe 2 d.'
        :                   'estudiant. Vegeu https://www.catcert.net/verCPIS'
        :                   'R-2EstudiantUR'
        :                       }
        :                     }
        :                   }
        :                 }
        :               }
        :             }
        :           }
1648   51:         SEQUENCE {
1650    8:           OBJECT IDENTIFIER authorityInfoAccess (1 3 6 1 5 5 7 1 1)
1660   39:           OCTET STRING, encapsulates {
1662   37:             SEQUENCE {
1664   35:               SEQUENCE {
1666    8:                 OBJECT IDENTIFIER ocsp (1 3 6 1 5 5 7 48 1)
1676   23:                 [6] 'http://ocsp.catcert.net'
        :                 }
        :               }
        :             }
        :           }
1701   24:         SEQUENCE {
1703    8:           OBJECT IDENTIFIER qcStatements (1 3 6 1 5 5 7 1 3)
1713   12:           OCTET STRING, encapsulates {
1715   10:             SEQUENCE {
1717    8:               SEQUENCE {
1719    6:                 OBJECT IDENTIFIER etsiQcsCompliance (0 4 0 1862 1 1)
        :                 }
        :               }
        :             }
        :           }
1727   96:         SEQUENCE {
1729    3:           OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
1734   89:           OCTET STRING, encapsulates {
1736   87:             SEQUENCE {
1738   85:               SEQUENCE {
1740   83:                 [0] {
1742   81:                   [0] {
1744   38:                     [6] 'http://epscd.catcert.net/crl/ec-ur.crl'
```

```
1784   39:                     [6] 'http://epscd2.catcert.net/crl/ec-ur.crl'
         :                     }
         :                   }
         :                 }
         :               }
         :             }
         :           }
         :         }
         :       }
         :     }
1825   13:   SEQUENCE {
1827    9:     OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 113549 1 1 5)
1838    0:     NULL
         :     }
1840  257:   BIT STRING
         :     68 20 CF A8 C1 70 5D 2D 6B C5 6D B2 5F CA 3F F2
         :     54 63 4D 3D 09 1D 54 5D 39 21 48 07 7C 88 41 57
         :     E5 14 A3 1A 55 9C E0 B5 26 17 CE 87 96 5D 02 9A
         :     03 37 89 D4 67 6B CC BF 3C FF 8B F0 F5 69 47 0F
         :     3B 7D 5A 21 D7 BF 89 9D 12 41 E0 FB 46 10 22 AD
         :     E7 97 23 7A 87 49 3E 68 1C 74 3F 75 D0 F4 D4 38
         :     E1 BE 06 DF 2D 51 00 57 58 A4 D8 01 F1 A8 71 52
         :     C9 B0 5A 2F 15 33 93 94 A1 EC D5 B7 76 BB 93 B6
         :             [ Another 128 bytes skipped ]
         :   }
```

A.2 Signatura CAdES

Exemple de signatura CAdES integrada a un document PDF.

```
  0 NDEF: SEQUENCE {
  2    9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
 13 NDEF:   [0] {
 15 NDEF:     SEQUENCE {
 17    1:       INTEGER 1
 20   11:       SET {
 22    9:         SEQUENCE {
 24    5:           OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
 31    0:           NULL
         :           }
         :         }
```

```
  33 NDEF:       SEQUENCE {
  35    9:         OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
        :         }
  48 4044:       [0] {
  52 1939:         SEQUENCE {

[ ... Certificat EC-ACC ... ]

        :           }
1995 2097:         SEQUENCE {

[ ... Certificat EC-UR ... ]

        :           }
        :         }
4096 6751:       SET {
4100 6747:         SEQUENCE {
4104    1:           INTEGER 1
4107  302:           SEQUENCE {
4111  280:             SEQUENCE {
4115   11:               SET {
4117    9:                 SEQUENCE {
4119    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
4124    2:                   PrintableString 'ES'
        :                   }
        :                 }
4128   59:               SET {
4130   57:                 SEQUENCE {
4132    3:                   OBJECT IDENTIFIER organizationName (2 5 4 10)
4137   50:                   PrintableString
        :                   'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :                   'I)'
        :                   }
        :                 }
4189   52:               SET {
4191   50:                 SEQUENCE {
4193    3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
4198   43:                   PrintableString
        :                   'Passatge de la Concepcio 11 08008 Barcelona'
        :                   }
        :                 }
4243   46:               SET {
4245   44:                 SEQUENCE {
```

```
4247    3:                  OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
4252   37:                  PrintableString 'Serveis Publics de Certificacio EC
         :                                  V-2'
         :                  }
         :                }
4291   53:              SET {
4293   51:                SEQUENCE {
4295    3:                  OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
4300   44:                  PrintableString
         :                  'Vegeu https://www.catcert.net/verCIC-2 (c)03'
         :                  }
         :                }
4346   31:              SET {
4348   29:                SEQUENCE {
4350    3:                  OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
4355   22:                  PrintableString 'Universitats i Recerca'
         :                  }
         :                }
4379   14:              SET {
4381   12:                SEQUENCE {
4383    3:                  OBJECT IDENTIFIER commonName (2 5 4 3)
4388    5:                  PrintableString 'EC-UR'
         :                  }
         :                }
         :              }
4395   16:            INTEGER
         :              26 2A 2F 19 C3 5C 8B 32 4B 72 AD C7 18 28 9F 56
         :            }
4413    9:          SEQUENCE {
4415    5:            OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
4422    0:            NULL
         :            }
4424 3415:          [0] {
4428   24:            SEQUENCE {
4430    9:              OBJECT IDENTIFIER contentType (1 2 840 113549 1 9 3)
4441   11:              SET {
4443    9:                OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
         :                }
         :              }
4454   28:            SEQUENCE {
4456    9:              OBJECT IDENTIFIER signingTime (1 2 840 113549 1 9 5)
4467   15:              SET {
4469   13:                UTCTime 14/09/2010 19:34:24 GMT
```

```
         :                }
         :              }
4484   35:            SEQUENCE {
4486    9:              OBJECT IDENTIFIER messageDigest (1 2 840 113549 1 9 4)
4497   22:              SET {
4499   20:                OCTET STRING
         :                  7B 70 4C B1 55 02 62 69 14 0D 56 24 EF 0C 71 E6
         :                  E0 7E 03 FD
         :                }
         :              }
4521   43:            SEQUENCE {
4523   11:              OBJECT IDENTIFIER
         :                signingCertificate (1 2 840 113549 1 9 16 2 12)
4536   28:              SET {
4538   26:                SEQUENCE {
4540   24:                  SEQUENCE {
4542   22:                    SEQUENCE {
4544   20:                      OCTET STRING
         :                  10 D3 1A F8 D5 80 DF A9 55 6C 60 90 9C 60 32 34
         :                  97 F3 04 91
         :                      }
         :                    }
         :                  }
         :                }
         :              }
4566 3273:            SEQUENCE {
4570    9:              OBJECT IDENTIFIER '1 2 840 113583 1 1 8'
4581 3258:              SET {
4585 3254:                SEQUENCE {
4589 1009:                  [0] {
                  [ ... CRL EC-ACC ... ]
         :                  }
5602 2237:                  [1] {
                  [ ... resposta OCSP sobre el certificat ... ]
         :                  }
         :                }
         :              }
         :            }
7843   13:          SEQUENCE {
7845    9:            OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
7856    0:            NULL
         :            }
```

```
7858  128:           OCTET STRING

   . . .

7989 2858:           [1] {
7993 2854:             SEQUENCE {
7997   11:               OBJECT IDENTIFIER
         :                 timeStampToken (1 2 840 113549 1 9 16 2 14)
8010 2837:               SET {
8014 2833:                 SEQUENCE {
8018    9:                   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
8029 2818:                   [0] {
8033 2814:                     SEQUENCE {
8037    1:                       INTEGER 3
8040   11:                       SET {
8042    9:                         SEQUENCE {
8044    5:                           OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
8051    0:                           NULL
         :                           }
         :                         }
8053  140:                       SEQUENCE {
8056   11:                         OBJECT IDENTIFIER
         :                           tSTInfo (1 2 840 113549 1 9 16 1 4)
8069  125:                         [0] {
8071  123:                           OCTET STRING, encapsulates {
8073  121:                             SEQUENCE {
8075    1:                               INTEGER 1
8078    6:                               OBJECT IDENTIFIER '0 4 0 2023 1 1'
8086   33:                               SEQUENCE {
8088    9:                                 SEQUENCE {
8090    5:                                   OBJECT IDENTIFIER
         :                                     sha1 (1 3 14 3 2 26)
8097    0:                                   NULL
         :                                   }
8099   20:                                 OCTET STRING
         :                   CF 67 5B 32 BE 13 DA 82 12 46 D3 E4 AD BA 05 78
         :                   9A CA 0A 30
         :                                 }
8121   20:                               INTEGER
         :                   27 D6 33 69 76 7D 83 C5 9E C3 04 D8 AA 9C 49 9E
         :                   35 A3 DE A8
8143   15:                               GeneralizedTime 14/09/2010 19:34:25 GMT
8160    9:                               SEQUENCE {
```

```
8162    1:                                  INTEGER 1
8165    1:                                  [0] 01
8168    1:                                  [1] 01
        :                                  }
8171    1:                                BOOLEAN TRUE
8174   20:                                INTEGER
        :                   2D 37 37 35 35 33 34 38 37 35 33 35 32 37 31 31
        :                   35 30 31 37
        :                              }
        :                            }
        :                          }
        :                        }
8196 1817:                      [0] {
8200 1813:                        SEQUENCE {
               [ ... Certificat Servei de segellat de temps ... ]
        :                          }
        :                        }
10017  830:                      SET {
10021  826:                        SEQUENCE {
10025    1:                          INTEGER 1
10028  264:                          SEQUENCE {
10032  243:                            SEQUENCE {
10035   11:                              SET {
10037    9:                                SEQUENCE {
10039    3:                                  OBJECT IDENTIFIER
        :                                    countryName (2 5 4 6)
10044    2:                                  PrintableString 'ES'
        :                                  }
        :                                }
10048   59:                              SET {
10050   57:                                SEQUENCE {
10052    3:                                  OBJECT IDENTIFIER
        :                                    organizationName (2 5 4 10)
10057   50:                                  PrintableString
        :                   'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :                   'I)'
        :                                  }
        :                                }
10109   40:                              SET {
10111   38:                                SEQUENCE {
10113    3:                                  OBJECT IDENTIFIER
        :                                    organizationalUnitName (2 5 4 11)
10118   31:                                  PrintableString 'Serveis Publics
```

```
 :   de Certificacio'
         :                                   }
         :                                 }
10151   53:                               SET {
10153   51:                                 SEQUENCE {
10155    3:                                   OBJECT IDENTIFIER
         :                                    organizationalUnitName (2 5 4 11)
10160   44:                                   PrintableString
         :                     'Vegeu https://www.catcert.net/verarrel (c)03'
         :                                   }
         :                                 }
10206   53:                               SET {
10208   51:                                 SEQUENCE {
10210    3:                                   OBJECT IDENTIFIER
         :                                    organizationalUnitName (2 5 4 11)
10215   44:                                   PrintableString
         :                     'Jerarquia Entitats de Certificacio Catalanes'
         :                                   }
         :                                 }
10261   15:                               SET {
10263   13:                                 SEQUENCE {
10265    3:                                   OBJECT IDENTIFIER
         :                                    commonName (2 5 4 3)
10270    6:                                   PrintableString 'EC-ACC'
         :                                   }
         :                                 }
         :                               }
10278   16:                             INTEGER
         :                     63 9D 4A 00 59 A5 00 AC 4A 76 BD 92 11 97 E5 99
         :                              }
10296    9:                            SEQUENCE {
10298    5:                              OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
10305    0:                              NULL
         :                              }
10307  394:                            [0] {
10311   26:                              SEQUENCE {
10313    9:                                OBJECT IDENTIFIER
         :                                 contentType (1 2 840 113549 1 9 3)
10324   13:                                SET {
10326   11:                                  OBJECT IDENTIFIER
         :                                   tSTInfo (1 2 840 113549 1 9 16 1 4)
         :                                  }
         :                                }
```

```
10339   35:                             SEQUENCE {
10341    9:                               OBJECT IDENTIFIER
         :                                messageDigest (1 2 840 113549 1 9 4)
10352   22:                               SET {
10354   20:                                 OCTET STRING
         :                     71 3F 58 02 44 6F 3B EE 95 0E 42 33 C2 DC B9 6B
         :                     E5 FC 8D 38
         :                                 }
         :                               }
10376  325:                             SEQUENCE {
10380   11:                               OBJECT IDENTIFIER
         :                                signingCertificate (1 2 840 113549 1
             9 16 2 12)
10393  308:                               SET {
10397  304:                                 SEQUENCE {
10401  300:                                   SEQUENCE {
10405  296:                                     SEQUENCE {
10409   20:                                       OCTET STRING
         :                     B7 52 71 0F 8C C5 FF BE 94 A1 5F 94 F1 78 11 25
         :                     C5 0A DB 5F
10431  270:                                     SEQUENCE {
10435  249:                                       SEQUENCE {
10438  246:                                         [4] {
                 [ . . . ]
         :                                         }
         :                                       }
10687   16:                                     INTEGER
         :                     63 9D 4A 00 59 A5 00 AC 4A 76 BD 92 11 97 E5 99
         :                                     }
         :                                   }
         :                                 }
         :                               }
         :                             }
10705   13:                           SEQUENCE {
10707    9:                             OBJECT IDENTIFIER
         :                              rsaEncryption (1 2 840 113549 1 1 1)
10718    0:                             NULL
         :                             }
10720  128:                           OCTET STRING
         :                     08 50 6E 55 47 8B 96 C0 88 7D 9F D3 D4 7E 86 19
         :                     DC 64 1A EB 07 98 73 40 D2 66 DD FB F7 A2 20 08
```

```
    :                   B8 33 80 EF 7D 03 C1 47 81 32 4F 90 0A 74 5D 3B
    :                   AA 57 E8 B1 AD B2 15 60 AF DA 1C 76 01 81 C8 D3
    :                   2A FE BE CA 4D 44 D3 AE 08 D8 9D A8 F8 EE BA B5
    :                   CD 35 10 8A 87 CB 9E 10 17 95 A6 6B 4B 6B 79 C1
    :                   61 60 CF A7 67 BD 3C 88 42 DC C7 93 58 39 A2 FC
    :                   ED 7D D2 84 6B 8F 5C 68 E1 26 A2 3B EC A9 BF B8
    :                           }
    :                         }
    :                       }
    :                     }
    :                   }
    :                 }
    :               }
    :             }
    :           }
    :         }
    :       }
    :     }
    :   }
```

A.3 CRL

Mostra de CRL provinent de EC-UR. Hem reduit el número d'entrades de la llista per reduir espai a partir del byte 525.

```
  0 86623: SEQUENCE {
  5 86342:   SEQUENCE {
 10     1:     INTEGER 1
 13    13:     SEQUENCE {
 15     9:       OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 113549 1 1 5)
 26     0:       NULL
        :       }
 28   280:     SEQUENCE {
 32    11:       SET {
 34     9:         SEQUENCE {
 36     3:           OBJECT IDENTIFIER countryName (2 5 4 6)
 41     2:           PrintableString 'ES'
        :           }
        :         }
 45    59:       SET {
 47    57:         SEQUENCE {
 49     3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
```

```
 54   50:           PrintableString
        :             'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :             'I)'
        :           }
        :         }
106   52:       SET {
108   50:         SEQUENCE {
110    3:           OBJECT IDENTIFIER localityName (2 5 4 7)
115   43:           PrintableString
        :             'Passatge de la Concepcio 11 08008 Barcelona'
        :           }
        :         }
160   46:       SET {
162   44:         SEQUENCE {
164    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
169   37:           PrintableString 'Serveis Publics de Certificacio ECV-2'
        :           }
        :         }
208   53:       SET {
210   51:         SEQUENCE {
212    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
217   44:           PrintableString
        :             'Vegeu https://www.catcert.net/verCIC-2 (c)03'
        :           }
        :         }
263   31:       SET {
265   29:         SEQUENCE {
267    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
272   22:           PrintableString 'Universitats i Recerca'
        :           }
        :         }
296   14:       SET {
298   12:         SEQUENCE {
300    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
305    5:           PrintableString 'EC-UR'
        :           }
        :         }
        :       }
312   13:     UTCTime 16/07/2010 11:00:10 GMT
327   13:     UTCTime 23/07/2010 11:00:10 GMT
342 85579:     SEQUENCE {
347   47:       SEQUENCE {
349   16:         INTEGER 1F 8E 8D 29 EE BB BD 22 42 70 B7 F9 96 3A 13 B3
```

```
367   13:         UTCTime 28/04/2005 10:20:13 GMT
382   12:         SEQUENCE {
384   10:           SEQUENCE {
386    3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
391    3:             OCTET STRING, encapsulates {
393    1:               ENUMERATED 0
        :               }
        :             }
        :           }
        :         }
396   32:       SEQUENCE {
398    1:         INTEGER 15
401   13:         UTCTime 28/04/2005 10:19:53 GMT
416   12:         SEQUENCE {
418   10:           SEQUENCE {
420    3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
425    3:             OCTET STRING, encapsulates {
427    1:               ENUMERATED 0
        :               }
        :             }
        :           }
        :         }
430   47:       SEQUENCE {
432   16:         INTEGER 14 B1 F6 25 7D CF 69 A8 44 72 DB 46 BB B6 68 19
450   13:         UTCTime 03/10/2006 06:41:16 GMT
465   12:         SEQUENCE {
467   10:           SEQUENCE {
469    3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
474    3:             OCTET STRING, encapsulates {
476    1:               ENUMERATED 4
        :               }
        :             }
        :           }
        :         }
479   47:       SEQUENCE {
481   16:         INTEGER 12 AE C2 39 44 A1 4E 92 44 72 DC 76 CF DD CE 69
499   13:         UTCTime 07/11/2007 14:38:50 GMT
514   12:         SEQUENCE {
516   10:           SEQUENCE {
518    3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
523    3:             OCTET STRING, encapsulates {
525    1:               ENUMERATED 0
        :               }
```

```
         :             }
         :           }
         :         }


          [ ... moltes més entrades de la CRL ... ]


85877   47:       SEQUENCE {
85879   16:         INTEGER 5E 41 74 BA D7 33 2A AE 4C 3C 0E 5A 78 86 A1 EC
85897   13:         UTCTime 13/07/2010 07:14:39 GMT
85912   12:         SEQUENCE {
85914   10:           SEQUENCE {
85916    3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
85921    3:             OCTET STRING, encapsulates {
85923    1:               ENUMERATED 4
         :              }
         :            }
         :          }
         :        }
         :       }
85926  422:     [0] {
85930  418:       SEQUENCE {
85934   11:         SEQUENCE {
85936    3:           OBJECT IDENTIFIER cRLNumber (2 5 29 20)
85941    4:           OCTET STRING, encapsulates {
85943    2:             INTEGER 2004
         :            }
         :           }
85947  305:         SEQUENCE {
85951    3:           OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
85956  296:           OCTET STRING, encapsulates {
85960  292:             SEQUENCE {
85964   20:               [0]
         :                 48 9F 8D 86 D1 CB F1 D6 8A 52 7B 7F 15 A2 0A 51
         :                 F8 97 FE 0B
85986  249:               [1] {
85989  246:                 [4] {
85992  243:                   SEQUENCE {
85995   11:                     SET {
85997    9:                       SEQUENCE {
```

```
85999    3:                        OBJECT IDENTIFIER countryName (2 5 4 6)
86004    2:                        PrintableString 'ES'
         :                       }
         :                     }
86008   59:                     SET {
86010   57:                       SEQUENCE {
86012    3:                         OBJECT IDENTIFIER organizationName (2 5 4 10)
86017   50:                         PrintableString
         :                   'Agencia Catalana de Certificacio (NIF Q-0801176-'
         :                   'I)'
         :                       }
         :                     }
86069   40:                     SET {
86071   38:                       SEQUENCE {
86073    3:                         OBJECT IDENTIFIER
         :                          organizationalUnitName (2 5 4 11)
86078   31:                         PrintableString 'Serveis Publics de Certifi
 cacio'
         :                       }
         :                     }
86111   53:                     SET {
86113   51:                       SEQUENCE {
86115    3:                         OBJECT IDENTIFIER
         :                          organizationalUnitName (2 5 4 11)
86120   44:                         PrintableString
         :                   'Vegeu https://www.catcert.net/verarrel (c)03'
         :                       }
         :                     }
86166   53:                     SET {
86168   51:                       SEQUENCE {
86170    3:                         OBJECT IDENTIFIER
         :                          organizationalUnitName (2 5 4 11)
86175   44:                         PrintableString
         :                   'Jerarquia Entitats de Certificacio Catalanes'
         :                       }
         :                     }
86221   15:                     SET {
86223   13:                       SEQUENCE {
86225    3:                         OBJECT IDENTIFIER commonName (2 5 4 3)
86230    6:                         PrintableString 'EC-ACC'
         :                       }
         :                     }
         :                   }
```

```
         :                   }
         :                  }
86238   16:                [2]
         :                  06 A5 5F 3C B2 81 95 28 3F E0 49 E7 F9 31 9D 6C
         :                }
         :              }
         :            }
86256   94:          SEQUENCE {
86258    3:            OBJECT IDENTIFIER issuingDistributionPoint (2 5 29 28)
86263   87:            OCTET STRING, encapsulates {
86265   85:              SEQUENCE {
86267   83:                [0] {
86269   81:                  [0] {
86271   38:                    [6] 'http://epscd.catcert.net/crl/ec-ur.crl'
86311   39:                    [6] 'http://epscd2.catcert.net/crl/ec-ur.crl'
         :                  }
         :                }
         :              }
         :            }
         :          }
         :        }
         :      }
86352   13:    SEQUENCE {
86354    9:      OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 113549 1 1 5)
86365    0:      NULL
         :      }
86367  257:    BIT STRING
         :      25 DB CE DA 30 AC EF 29 9E 23 A8 A0 14 4A 60 1F
         :      3D 6B 80 04 72 37 0F 55 F6 C4 0E CB 22 65 BE 62
         :      81 36 33 74 8C D7 4A A1 63 35 05 E4 CF A2 5D 49
         :      85 5D BF 5A F0 1E 24 01 B2 06 E0 17 D0 93 10 9B
         :      9B 6C 0C AC 97 8B 37 6D F8 97 95 A9 C3 95 78 15
         :      9D FC E9 A9 5F 89 5D 1D 33 DC 30 BC 0C 56 48 E0
         :      E0 16 23 73 A4 2E 83 FA CE 33 77 95 52 AA 65 35
         :      7B 66 69 B5 CF B9 88 42 12 E3 7B B2 74 F3 07 8E
         :              [ Another 128 bytes skipped ]
         :    }
```

A.4 Resposta OCSP

A continuació veiem la resposta OCSP inclosa a la signatura que hem mostrat abans. Podem comprovar que el certificat signant és vàlid amb el tag [0] del byte 5767. L'error que surt a la decodificació és un problema de semàntica: l'estàndard OCSP permet que el camp de l'estat del certificat, en cas de ser vàlid, pugui omitir el valor i deixar el camp buit. Això provoca un error a la decodificació per mitjà del programa utilitzat (*dumpasn1*) perquè espera sempre un valor, però això no implica que la resposta estigui mal formada.

```
5625    9:              OBJECT IDENTIFIER
        :                ocspBasic (1 3 6 1 5 5 7 48 1 1)
5636 2203:              OCTET STRING, encapsulates {
5640 2199:                SEQUENCE {
5644  183:                  SEQUENCE {
5647   22:                    [2] {
5649   20:                      OCTET STRING
        :    EE 28 74 94 F7 0A 5F 5D 0B 50 62 92 AD 55 2A 2B
        :    B5 C2 19 5F
        :                      }
5671   15:                    GeneralizedTime 14/09/2010 19:34:23 GMT
5688  115:                    SEQUENCE {
5690  113:                      SEQUENCE {
5692   73:                        SEQUENCE {
5694    9:                          SEQUENCE {
5696    5:                            OBJECT IDENTIFIER
        :                              sha1 (1 3 14 3 2 26)
5703    0:                            NULL
        :                            }
5705   20:                          OCTET STRING
        :    52 1F 8D 94 72 77 17 F1 E6 83 B4 41 0D A6 AB CC
        :    44 12 D3 AE
5727   20:                          OCTET STRING
        :    48 9F 8D 86 D1 CB F1 D6 8A 52 7B 7F 15 A2 0A 51
        :    F8 97 FE 0B
5749   16:                          INTEGER
        :    26 2A 2F 19 C3 5C 8B 32 4B 72 AD C7 18 28 9F 56
        :                          }
5767    0:                        [0]
        :                          Error: Object has zero length.
5769   15:                        GeneralizedTime 14/09/2010 10:43:29 GMT
5786   17:                        [0] {
5788   15:                          GeneralizedTime 21/09/2010 10:43:29 GMT
```

```
         :                         }
         :                       }
         :                     }
5805   23:                     [1] {
5807   21:                       SEQUENCE {
5809   19:                         SEQUENCE {
5811    9:                           OBJECT IDENTIFIER
         :                             ocspNonce (1 3 6 1 5 5 7 48 1 2)
5822    6:                           OCTET STRING
         :     01 2B 11 BD A7 06
         :                           }
         :                         }
         :                       }
         :                     }
5830   13:                   SEQUENCE {
5832    9:                     OBJECT IDENTIFIER
         :                       sha1withRSAEncryption (1 2 840 113549 1 1 5)
5843    0:                     NULL
         :                     }
5845  129:                   BIT STRING
         :     91 6B 73 32 57 E1 36 65 CA F4 DA 68 E8 AC A9 BC
         :     D9 8E 25 89 A3 5A 0E 0B 89 9B DA F0 1D 7A 77 70
         :     89 81 81 FA 9A A1 F1 EC 81 CF 76 76 98 ED 33 76
         :     AE 27 80 F7 B6 28 C1 69 4A D3 EA 2D 3D 6A 3E 8D
         :     77 E7 24 CE A3 D0 40 AD 09 FA 00 92 1C C8 35 A5
         :     25 89 62 06 06 53 6F 15 00 03 86 FC 9E 43 72 B4
         :     62 7B 4E 06 10 C6 0D 15 95 49 B2 40 5D DB E3 AB
         :     FE 0F 9D 93 CA B9 90 A1 57 5B 94 19 97 F4 8E 27
5977 1862:                   [0] {
5981 1858:                     SEQUENCE {
5985 1854:                       SEQUENCE {
5989 1574:                         SEQUENCE {
5993    3:                           [0] {
5995    1:                             INTEGER 2
         :                             }
5998   16:                           INTEGER
         :     21 3B 74 33 F7 CB 24 83 4B 9A 59 59 6E 09 97 AC
6016   13:                           SEQUENCE {
6018    9:                             OBJECT IDENTIFIER
         :                               sha1withRSAEncryption (1 2 840 113549
             1 1 5)
6029    0:                             NULL
         :                             }
```

```
6031  243:                           SEQUENCE {
6034   11:                             SET {
6036    9:                               SEQUENCE {
6038    3:                                 OBJECT IDENTIFIER
        :                                   countryName (2 5 4 6)
6043    2:                                 PrintableString 'ES'
        :                                 }
        :                               }
6047   59:                             SET {
6049   57:                               SEQUENCE {
6051    3:                                 OBJECT IDENTIFIER
        :                                   organizationName (2 5 4 10)
6056   50:                                 PrintableString
        :     'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :     'I)'
        :                                 }
        :                               }
6108   40:                             SET {
6110   38:                               SEQUENCE {
6112    3:                                 OBJECT IDENTIFIER
        :                                   organizationalUnitName (2 5 4 11)
6117   31:                                 PrintableString 'Serveis Publics de
   Certificacio'
        :                                 }
        :                               }
6150   53:                             SET {
6152   51:                               SEQUENCE {
6154    3:                                 OBJECT IDENTIFIER
        :                                   organizationalUnitName (2 5 4 11)
6159   44:                                 PrintableString
        :     'Vegeu https://www.catcert.net/verarrel (c)03'
        :                                 }
        :                               }
6205   53:                             SET {
6207   51:                               SEQUENCE {
6209    3:                                 OBJECT IDENTIFIER
        :                                   organizationalUnitName (2 5 4 11)
6214   44:                                 PrintableString
        :     'Jerarquia Entitats de Certificacio Catalanes'
        :                                 }
        :                               }
6260   15:                             SET {
6262   13:                               SEQUENCE {
```

```
6264    3:                                  OBJECT IDENTIFIER
        :                                    commonName (2 5 4 3)
6269    6:                                  PrintableString 'EC-ACC'
        :                                  }
        :                                }
        :                              }
6277   30:                            SEQUENCE {
6279   13:                              UTCTime 12/03/2010 15:10:17 GMT
6294   13:                              UTCTime 12/03/2014 15:10:17 GMT
        :                              }
6309  264:                            SEQUENCE {
6313   11:                              SET {
6315    9:                                SEQUENCE {
6317    3:                                  OBJECT IDENTIFIER
        :                                    countryName (2 5 4 6)
6322    2:                                  PrintableString 'ES'
        :                                  }
        :                                }
6326   59:                              SET {
6328   57:                                SEQUENCE {
6330    3:                                  OBJECT IDENTIFIER
        :                                    organizationName (2 5 4 10)
6335   50:                                  UTF8String
        :    'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :    'I)'
        :                                  }
        :                                }
6387   46:                              SET {
6389   44:                                SEQUENCE {
6391    3:                                  OBJECT IDENTIFIER
        :                                    organizationalUnitName (2 5 4 11)
6396   37:                                  UTF8String 'Serveis Publics de Cer
  tificacio CIO-1'
        :                                  }
        :                                }
6435   53:                              SET {
6437   51:                                SEQUENCE {
6439    3:                                  OBJECT IDENTIFIER
        :                                    organizationalUnitName (2 5 4 11)
6444   44:                                  UTF8String
        :    'Vegeu https://www.catcert.cat/verCIO-1 (c)05'
        :                                  }
        :                                }
```

```
6490   53:                            SET {
6492   51:                              SEQUENCE {
6494    3:                                OBJECT IDENTIFIER
       :                                  organizationalUnitName (2 5 4 11)
6499   44:                                UTF8String
       :     'Jerarquia Entitats de Certificacio Catalanes'
       :                                }
       :                              }
6545   30:                            SET {
6547   28:                              SEQUENCE {
6549    3:                                OBJECT IDENTIFIER
       :                                  commonName (2 5 4 3)
6554   21:                                UTF8String 'Servei OCSP de EC-ACC'
       :                                }
       :                              }
       :                            }
6577  159:                          SEQUENCE {
6580   13:                            SEQUENCE {
6582    9:                              OBJECT IDENTIFIER
       :                                rsaEncryption (1 2 840 113549 1 1 1)
6593    0:                              NULL
       :                              }
6595  141:                            BIT STRING, encapsulates {
6599  137:                              SEQUENCE {
6602  129:                                INTEGER
       :     00 B8 97 D2 3E 64 B2 D2 6E 32 55 36 9D CE C6 94
       :     FB BC BA 65 B9 FE 58 98 B0 11 03 37 58 33 BA A2
       :     7E E7 04 74 68 EF 04 53 D3 81 4D 0C 32 49 86 70
       :     BA 8C 85 30 66 BA C4 0D D7 64 97 F7 F3 23 1E 9B
       :     68 54 48 8A B5 00 7C CD B7 D1 50 F6 59 71 7C FE
       :     B7 DA 3A 3E D1 AB ED 2A F2 11 7F C2 27 6F 1A 22
       :     10 B5 CB 66 E8 4A 3D 32 27 15 65 BD 79 A6 1A C8
       :     A3 F8 3B 07 41 57 C6 35 3D 73 87 FE EF A0 1C 82
       :             [ Another 1 bytes skipped ]
6734    3:                                INTEGER 65537
       :                                }
       :                              }
       :                            }
6739  824:                          [3] {
6743  820:                            SEQUENCE {
6747   29:                              SEQUENCE {
6749    3:                                OBJECT IDENTIFIER
       :                                  issuerAltName (2 5 29 18)
```

```
6754   22:                        OCTET STRING, encapsulates {
6756   20:                          SEQUENCE {
6758   18:                            [1] 'ec_acc@catcert.net'
         :                            }
         :                          }
         :                        }
6778   54:                      SEQUENCE {
6780    3:                        OBJECT IDENTIFIER
         :                          subjectAltName (2 5 29 17)
6785   47:                        OCTET STRING, encapsulates {
6787   45:                          SEQUENCE {
6789   20:                            [1] 'ocsp_acc@catcert.cat'
6811   21:                            [4] {
6813   19:                              SEQUENCE {
6815   17:                                SET {
6817   15:                                  SEQUENCE {
6819    3:                                    OBJECT IDENTIFIER
         :                                      serialNumber (2 5 4 5)
6824    8:                                    PrintableString 'Q081176I'
         :                                    }
         :                                  }
         :                                }
         :                              }
         :                            }
         :                          }
         :                        }
6834   14:                      SEQUENCE {
6836    3:                        OBJECT IDENTIFIER
         :                          keyUsage (2 5 29 15)
6841    1:                        BOOLEAN TRUE
6844    4:                        OCTET STRING, encapsulates {
6846    2:                          BIT STRING 6 unused bits
         :                            '11'B
         :                          }
         :                        }
6850   19:                      SEQUENCE {
6852    3:                        OBJECT IDENTIFIER
         :                          extKeyUsage (2 5 29 37)
6857   12:                        OCTET STRING, encapsulates {
6859   10:                          SEQUENCE {
6861    8:                            OBJECT IDENTIFIER
         :                              ocspSigning (1 3 6 1 5 5 7 3 9)
         :                            }
```

```
        :                                   }
        :                                 }
6871   29:                               SEQUENCE {
6873    3:                                 OBJECT IDENTIFIER
        :                                   subjectKeyIdentifier (2 5 29 14)
6878   22:                                 OCTET STRING, encapsulates {
6880   20:                                   OCTET STRING
        :     EE 28 74 94 F7 0A 5F 5D 0B 50 62 92 AD 55 2A 2B
        :     B5 C2 19 5F
        :                                   }
        :                                 }
6902  305:                               SEQUENCE {
6906    3:                                 OBJECT IDENTIFIER
        :                                   authorityKeyIdentifier (2 5 29 35)
6911  296:                                 OCTET STRING, encapsulates {
6915  292:                                   SEQUENCE {
6919   20:                                     [0]
        :     A0 C3 8B 44 AA 37 A5 45 BF 97 80 5A D1 F1 78 A2
        :     9B E9 5D 8D
6941  249:                                     [1] {
6944  246:                                       [4] {
6947  243:                                         SEQUENCE {
6950   11:                                           SET {
6952    9:                                             SEQUENCE {
6954    3:                                               OBJECT IDENTIFIER
        :                                                 countryName (2 5 4 6)
6959    2:                                               PrintableString 'ES'
        :                                               }
        :                                             }
6963   59:                                           SET {
6965   57:                                             SEQUENCE {
6967    3:                                               OBJECT IDENTIFIER
        :                                                 organizationName (2 5
         4 10)
6972   50:                                               PrintableString
        :     'Agencia Catalana de Certificacio (NIF Q-0801176-'
        :     'I)'
        :                                               }
        :                                             }
7024   40:                                           SET {
7026   38:                                             SEQUENCE {
7028    3:                                               OBJECT IDENTIFIER
        :                                                 organizationalUnitName
```

```
          (2 5 4 11)
7033   31:                                          PrintableString 'Serveis
 Publics
 de Cert
 ificacio'
         :                                          }
         :                                        }
7066   53:                                      SET {
7068   51:                                        SEQUENCE {
7070    3:                                          OBJECT IDENTIFIER
         :                                            organizationalUnitName
          (2 5 4 11)
7075   44:                                          PrintableString
         :    'Vegeu https://www.catcert.net/verarrel (c)03'
         :                                          }
         :                                        }
7121   53:                                      SET {
7123   51:                                        SEQUENCE {
7125    3:                                          OBJECT IDENTIFIER
         :                                            organizationalUnitName
          (2 5 4 11)
7130   44:                                          PrintableString
         :    'Jerarquia Entitats de Certificacio Catalanes'
         :                                          }
         :                                        }
7176   15:                                      SET {
7178   13:                                        SEQUENCE {
7180    3:                                          OBJECT IDENTIFIER
         :                                            commonName (2 5 4 3)
7185    6:                                          PrintableString 'EC-ACC'
         :                                          }
         :                                        }
         :                                      }
         :                                    }
         :                                  }
         :                                [2]
7193   16:    EE 2B 3D EB D4 21 DE 14 A8 62 AC 04 F3 DD C4 01
         :                                  }
         :                                }
         :                              }
7211  182:                            SEQUENCE {
7214    3:                              OBJECT IDENTIFIER
         :                                certificatePolicies (2 5 29 32)
```

```
7219  174:                                   OCTET STRING, encapsulates {
7222  171:                                     SEQUENCE {
7225  168:                                       SEQUENCE {
7228   11:                                         OBJECT IDENTIFIER '1 3 6 1 4 1
   15096 1 3 1 19'
7241  152:                                         SEQUENCE {
7244   44:                                           SEQUENCE {
7246    8:                                             OBJECT IDENTIFIER
        :                                               cps (1 3 6 1 5 5 7 2 1)
7256   32:                                             IA5String 'https://www.cat
   cert.cat/verCIO-1'
        :                                             }
7290  104:                                           SEQUENCE {
7292    8:                                             OBJECT IDENTIFIER
        :                                               unotice (1 3 6 1 5 5 7 2 2)
7302   92:                                             SEQUENCE {
7304   90:                                               VisibleString
        :     'Aquest .s un certificat de servei OCSP de classe'
        :     ' 1. Vegeu https://www.catcert.cat/verCIO-1'
        :                                               }
        :                                             }
        :                                           }
        :                                         }
        :                                       }
        :                                     }
        :                                   }
7396   69:                                   SEQUENCE {
7398    8:                                     OBJECT IDENTIFIER
        :                                       authorityInfoAccess (1 3 6 1 5 5 7 1 1)
7408   57:                                     OCTET STRING, encapsulates {
7410   55:                                       SEQUENCE {
7412   53:                                         SEQUENCE {
7414    8:                                           OBJECT IDENTIFIER
        :                                             caIssuers (1 3 6 1 5 5 7 48 2)
7424   41:                                           [6]
        :     'http://www.catcert.cat/descarrega/acc.crt'
        :                                         }
        :                                       }
        :                                     }
        :                                   }
7467   98:                                   SEQUENCE {
7469    3:                                     OBJECT IDENTIFIER
        :                                       cRLDistributionPoints (2 5 29 31)
```

```
7474   91:                                  OCTET STRING, encapsulates {
7476   89:                                    SEQUENCE {
7478   87:                                      SEQUENCE {
7480   85:                                        [0] {
7482   83:                                          [0] {
7484   39:                                            [6] 'http://epscd.catcert.net
 /crl/ec-acc.crl'
7525   40:                                            [6] 'http://epscd2.catcert.net
 /crl/ec-acc.crl'
         :                                          }
         :                                        }
         :                                      }
         :                                    }
         :                                  }
         :                                }
         :                              }
         :                            }
         :                          }
7567   13:                          SEQUENCE {
7569    9:                            OBJECT IDENTIFIER
         :                              sha1withRSAEncryption (1 2 840 113549 1 1 5)
7580    0:                            NULL
         :                            }
7582  257:                          BIT STRING
         :    A3 70 87 0B 2E 97 A0 4D D1 60 A9 A1 AD 35 7E 5B
         :    8E 68 19 81 4D 0B 7D F5 D3 C4 6A C7 FC 61 D1 73
         :    CF 8D 61 13 C4 67 18 E2 0A 55 4D 6B E0 9A 9E 59
         :    EC D7 FA A1 9F FE FC 91 F9 67 E6 62 9B 52 08 6A
         :    25 E4 1E 53 09 58 B0 40 6B 90 2E A4 A8 8D 77 8E
         :    A2 35 28 76 D3 76 8F 96 08 B8 6A 8F EA CF ED 5B
         :    04 1D A1 29 D1 54 77 43 CE 1C 8C 35 CC 61 27 27
         :    99 41 8F 0D 50 DA 05 8D 6D 0A 58 87 9A 67 93 E2
         :            [ Another 128 bytes skipped ]
         :                        }
         :                      }
         :                    }
         :                  }
         :                }
         :              }
         :            }
```

Appendix B

Software de validació de signatures electròniques

Per validar les signatures creades pel nostre sistema necessitarem un software extern de validació. Per aquesta finalitat veurem l'ús de dos eines diferents: Adobe Reader i Sinadura 2.0.

B.1 Adobe Reader

Des de la lliberació del format al 2008, s'han creat molts lectors de documents PDF, tant en Windows com en Linux. Cap d'ells, però, (que haguem trobat) donen suport per signatures electròniques excepte el lector original, Adobe Reader [37]. Així, si volem validar les signatures ens veurem obligats a utilitzar aquest software.

Prèviament a la validació necessitarem establir les autoritats de confiança del programa. A la versió 9.3 això es pot fer mitjançant el menú "Document", apartat "Administrar autoritats de confiança" (o similar). En el diàleg obert s'ha d'afegir un nou contacte i introduir el fitxer amb els certificats d'autoritat apropiats (EC-ACC pel carnet UPC i AC RAIZ DNIE pel DNI electrònic), marcant-los com a autoritats arrel.

Fet això, la validació de signatures amb Adobe Reader és molt senzilla: a l'obrir un document signat automàticament es validen totes les signatures incloses i ens mostra l'estat a la barra superior (figura B.1).

Si volem més detalls sobre una signatura en particular podem accedir a la barra lateral de signatures, on podem accedir a informació més detallada de cada signatura (e.g. certificat signant, rol, data, ...) (figura B.2).

Finalment podem demanar informació sobre el certificat (i la cadena de certificació) en aquesta barra (figura B.3).

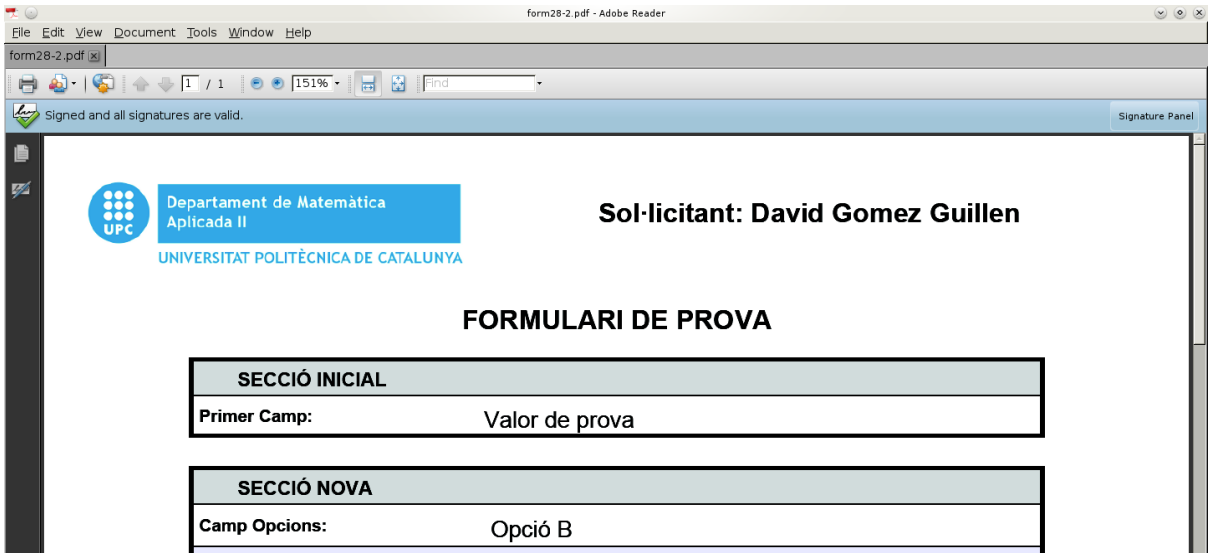

Figura B.1: Document PDF signat, visualitzat amb Adobe Reader.

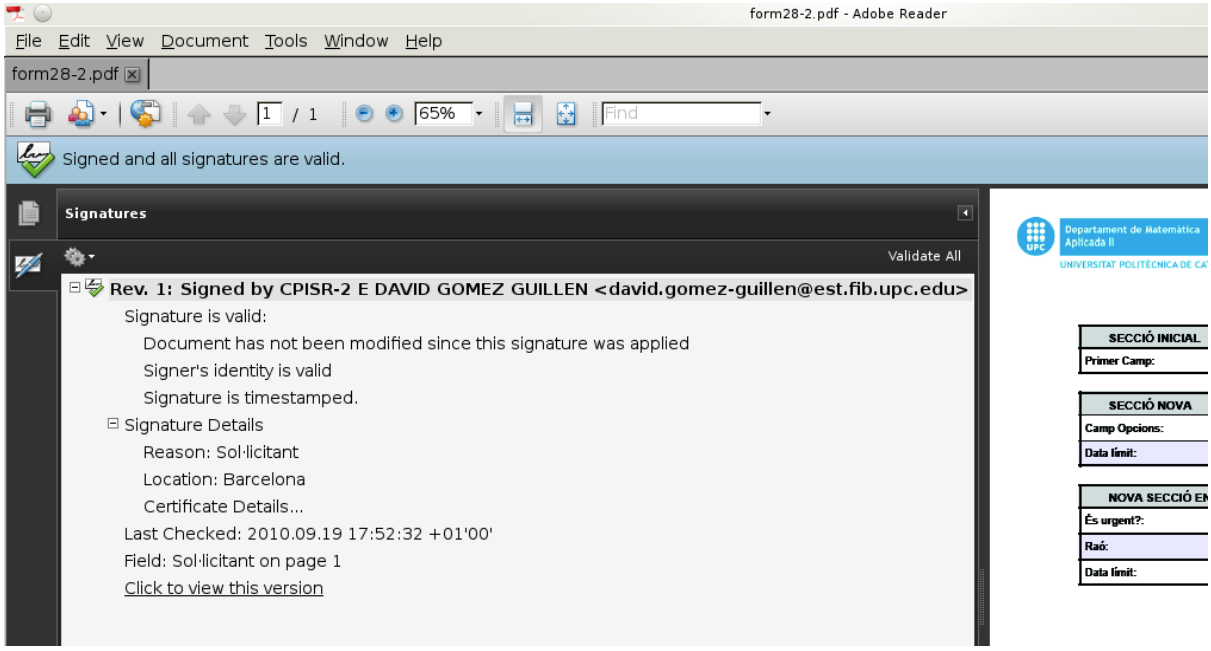

Figura B.2: Detalls d'una signatura a Adobe Reader.

Malaraudament, pel problema que hem comentat en capítols anteriors, Adobe Reader no valida correctament alguns certificats del DNI electrònic. Com a solució podem utilitzar un software alternatiu: la versió 2.0 de *sinadura*.

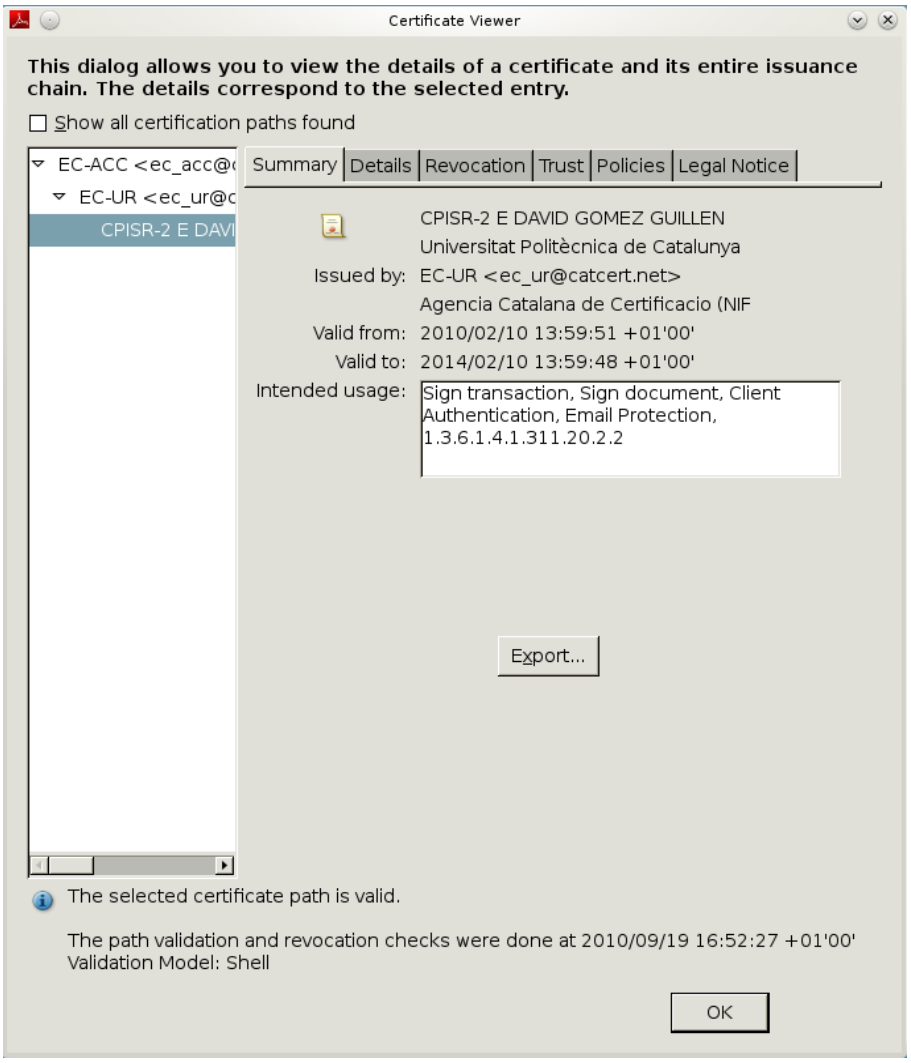

Figura B.3: Detalls del certificat (i cadena de certificació) del signant.

B.2 Sinadura 2.0

Sinadura és un software de signat i validació de documents PDF [20]. La seva interfície està orientada pel signat i validació en lot (múltiples fitxers), deixant la visualització en segon pla.

L'avantatge d'aquest és que no té les restriccions del Adobe Reader a l'hora de validar certificats del DNI electrònic, tot i que també necessitarem afegir diversos certificats perquè es pugui validar correctament (Menú "Sinadura", "Preferències", "Validació"). Una vegada preparat, validar documents és tan fàcil com afegir-los a la llista principal, seleccionar-los i prémer el botó de validar (figures B.5 i B.6).

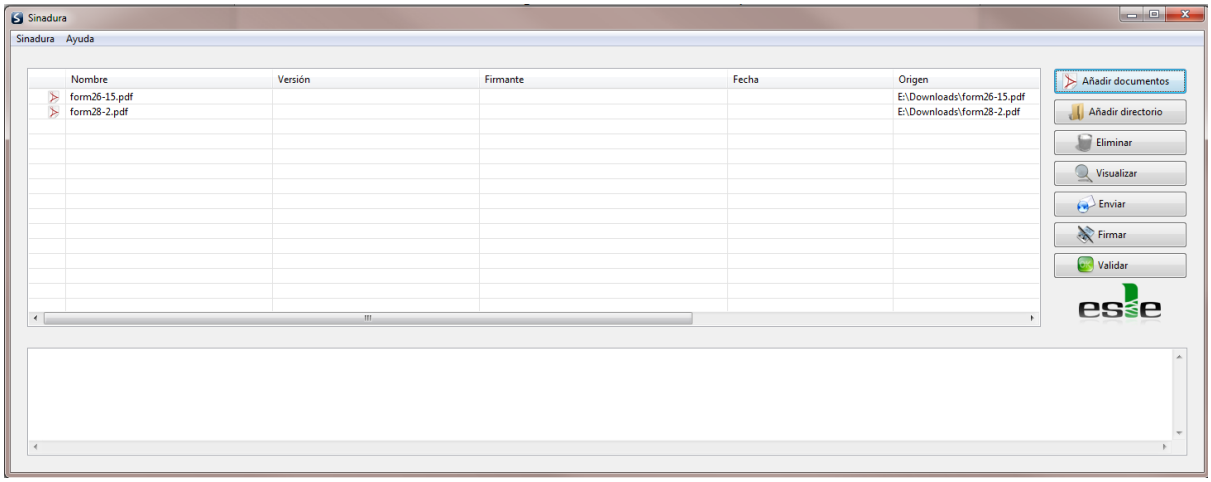

Figura B.4: Interfície principal de sinadura.

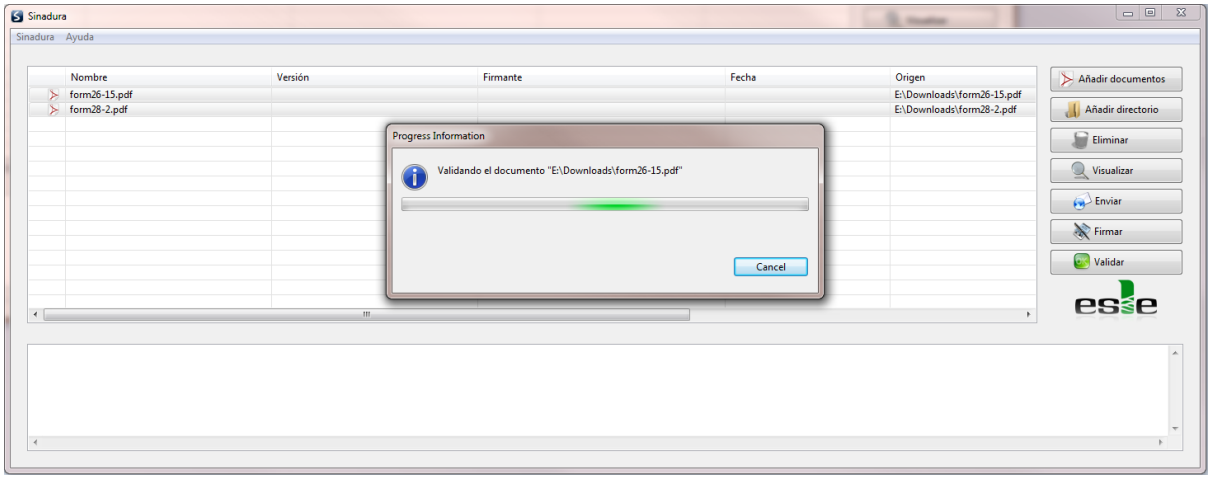

Figura B.5: Validació de múltiples documents PDF de sinadura.

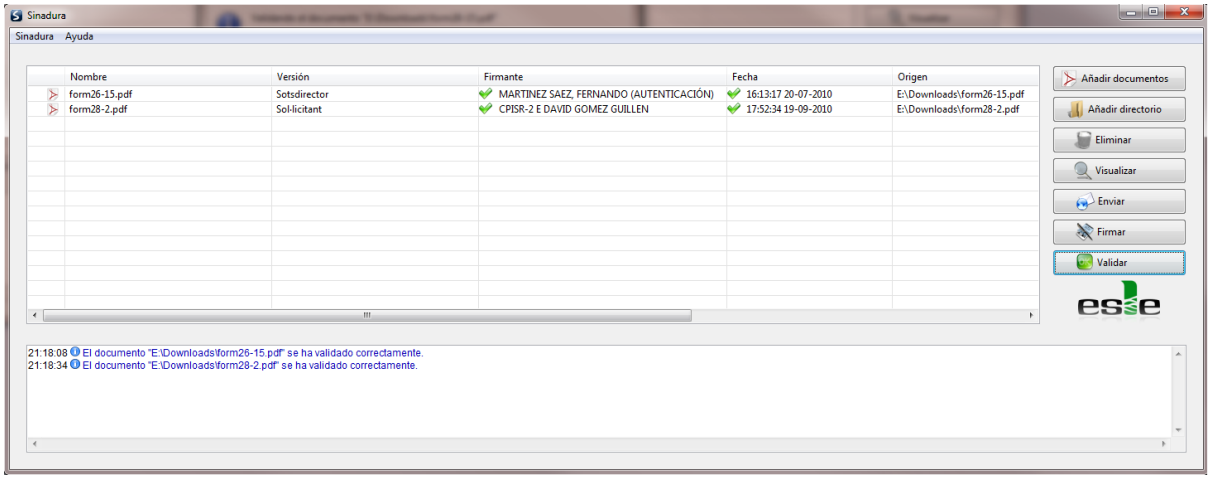

Figura B.6: Resultat de la validació de sinadura.

Appendix C

Manual d'ús

En aquesta secció explicarem l'ús de la web per els diversos rols: sol·licitant, càrrec i gestor. Actualment la web es troba a l'adreça `http://www-ma2.upc.edu/formularis`.

Abans de res és necessaria l'autenticació al sistema, excepte en el cas dels formularis anònims. Un usuari es pot identificar al sistema mitjançant usuari i password UPC o amb l'ús de la targeta electrònica (UPC o DNIe). Una vegada autenticat, i segons els permisos que aquest tingui, es poden veure un o més elements a la barra superior de navegació (figura C.1).

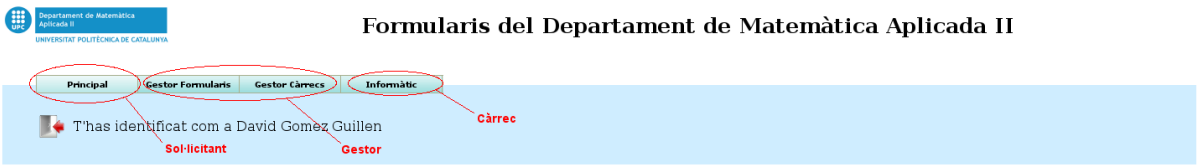

Figura C.1: Barra de navegació de la web.

C.1 Sol·licitant

A la secció principal hi trobem la llista de formularis actualment oberts. D'aquests alguns poden estar marcats com a expirats, de forma que no serà possible la introducció de noves dades. Per cada formulari hi ha dues opcions: veure i realitzar sol·licituds.

En el primer apartat es poden veure totes les sol·licituds fetes prèviament pel propi usuari, junt amb el seu estat (a l'espera d'un càrrec, acceptada o rebutjada), la data d'enviament i l'enllaç a la sol·licitud PDF si aquestes estan activades. Per una ràpida visualització també es pot veure la sol·licitud com a taula HTML, en una finestra apart. A més, si hi ha alguna petició pendent de signar, es pot fer des d'aquest apartat, seleccionant-les amb les caselles de la part esquerra i prement el botó de signar.

Figura C.2: Interfície principal del sol·licitant.

Figura C.3: Apartat de sol·licituds pendents del sol·licitant, amb la interfície per signar-les en lot.

El segon apartat s'utilitza per enviar sol·licituds. Una vegada dins es mostren els camps definits pel gestor amb la intenció de què l'usuari els ompli. Existeixen diferents tipus de camps que limiten la manera d'introduir dades, la majoria d'ells prou intuitius: text, números, caselles, botons d'opció, dates, ... La majoria necessiten que javascript estigui habilitat pel seu funcionament correcte. Aquests camps poden estar dividits en seccions visualment distinguibles i també en diferents pàgines.

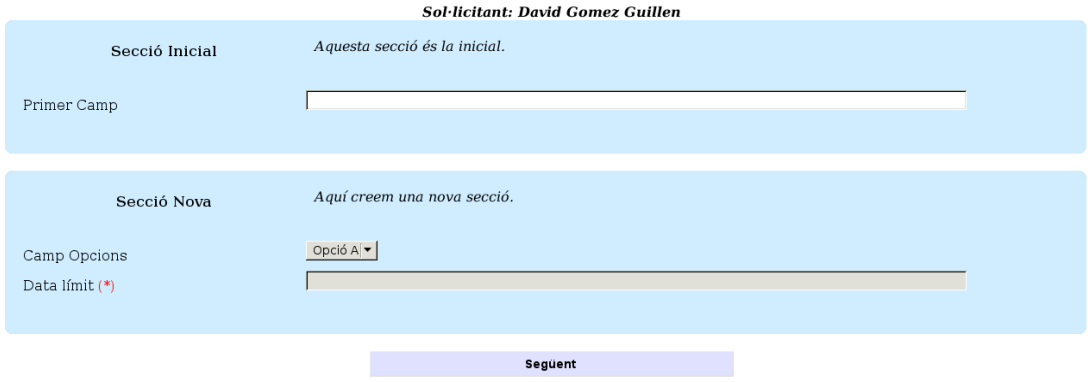

Figura C.4: Exemple de formulari amb dues seccions.

Una vegada s'han introduit totes les dades s'arriba a la pàgina de confirmació. Aquesta

mostra un enllaç al document PDF (si estan habilitats) i, si és necessari, es demana la signatura electrònica del sol·licitant. Si hi ha problemes tècnics amb la signatura (e.g. no es disposa de la targeta, no es té un lector, aquest no està configurat, ...) es permet també l'enviament de la sol·licitud sense signar. Quan es confirma, i si tot ha anat bé, es mostra un missatge de confirmació i, en el cas de no haver signat quan era necessari, un recordatori perquè l'usuari se'n recordi de fer-ho com més aviat millor.

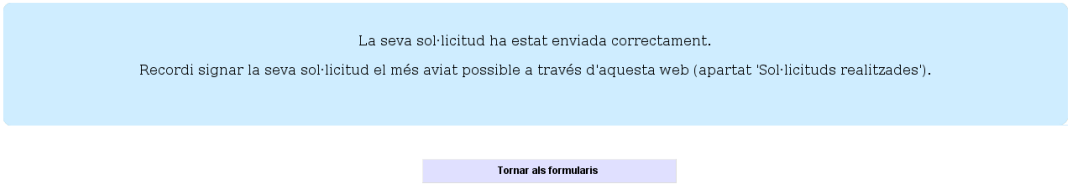

Figura C.5: Confirmació de l'enviament de la sol·licitud, amb el recordatori de signar-la aviat.

Independentment de com se signi, la sol·licitud entra al sistema i es notifica el següent càrrec definit al procés administratiu.

C.2 Càrrec

Si un usuari està registrat al sistema (pel gestor) com a pertanyent a un càrrec, li sortirà la pestanya a la barra de navegació associada al rol del càrrec. Aquest apartat té la llista de formularis, oberts i tancats, en els que el càrrec participa. Per cada formulari existeixen dos subapartats: la visualització de totes les entrades i la de només aquelles pendents de la seva signatura.

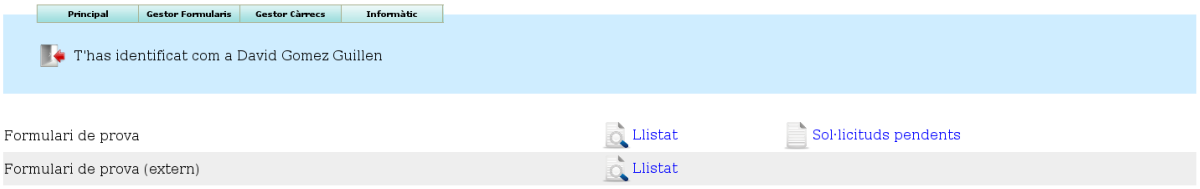

Figura C.6: Interfície principal d'un càrrec (Informàtic, càrrec temporal de prova).

El primer apartat és semblant al de l'apartat del sol·licitant, amb la informació d'estat, temps, taula HTML i enllaç al PDF de cada sol·licitud d'aquell formulari. A més, incorporem la possibilitat d'exportar la taula a un format de fulla de càlcul per la seva descàrrega. El segon (sol·licituds pendents) és un subconjunt del primer, però amb la possibilitat de seleccionar les entrades que es volen signar de cop. Si no s'està d'acord amb alguna també es poden rebutjar, especificant la raó i confirmant el rebuig. En qualsevol dels dos casos les sol·licituds deixen d'estar a l'espera de la signatura del càrrec actual i per tant desapareixen de la vista de sol·licituds pendents.

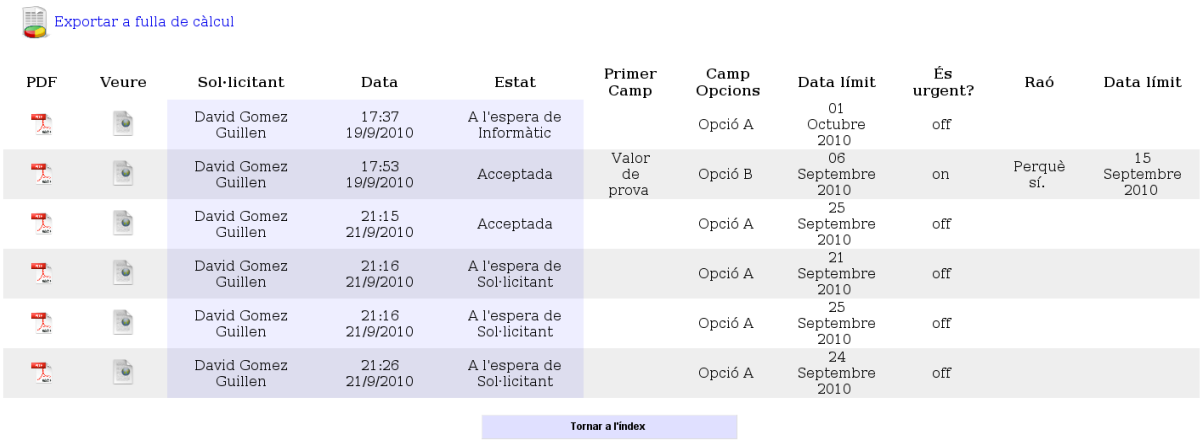

Figura C.7: Llistat de totes les sol·licituds d'un formulari.

Figura C.8: Sol·licituds pendents de signatura, amb la interfície per signar.

Similarment al cas del sol·licitant, una vegada signades les sol·licituds es notifica el següent càrrec de la llista. Si l'actual era l'últim càrrec, o si aquest ha rebutjat l'entrada, es notifica el sol·licitant de la decisió presa (acceptat o rebutjat).

C.3 Gestor

Finalment trobem els apartats més complexos i importants del sistema, els de gestió. Aquests estan dividits en dos: la gestió de càrrecs i la gestió de formularis.

La gestió de càrrecs no és massa complicada; trobem una taula per cada càrrec amb la gent que té associada. En aquesta pantalla podem canviar el nom de la persona que ocupa el càrrec (e.g. Director, Directora, ...), així com afegir o treure persones a un càrrec, especificant el seu nom d'usuari i correu electrònic. Per afegir o eliminar càrrecs ho haurem de fer fora de la interfície web, editant fitxers de configuració del sistema, per temes de seguretat (veure annex D.4).

La gestió de formularis és on es concentra gran part de la funcionalitat i versatilitat del sistema. A baix de tot trobem els botons per crear un formulari intern o extern, acció que ens porta a la pantalla d'edició del nou formulari creat. Ocupant la majoria de la

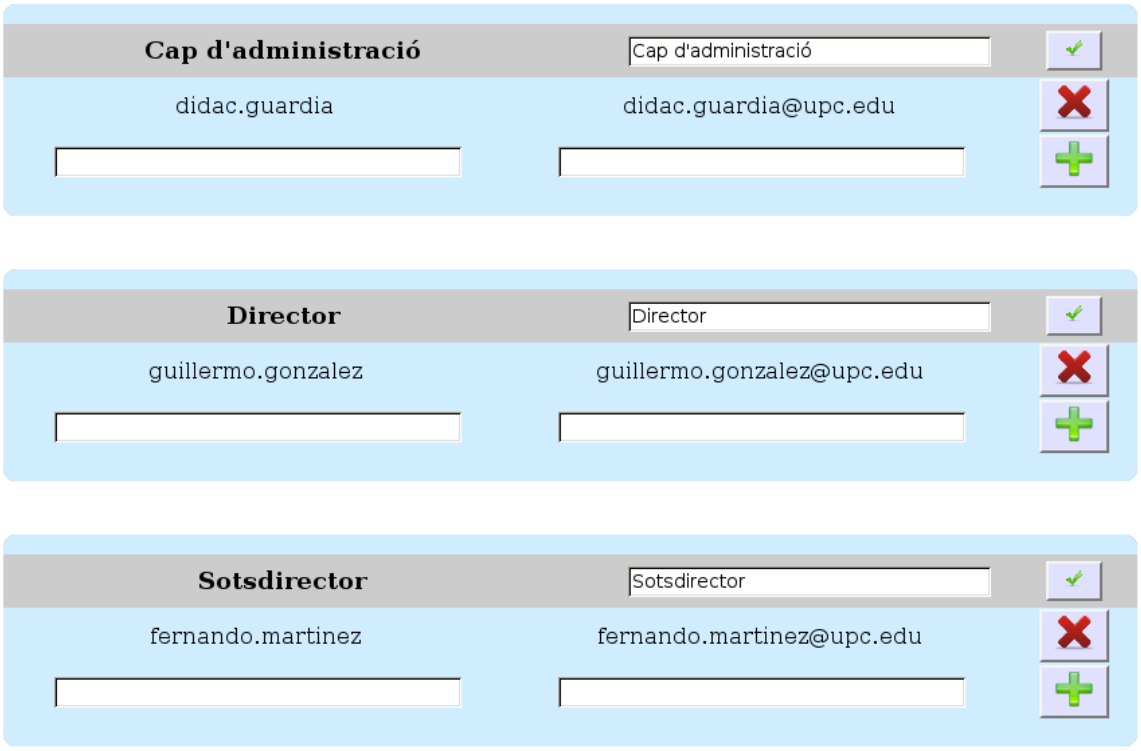

Figura C.9: Interfície de gestió de càrrecs.

pàgina, però, es mostra una llista de tots els formularis junt amb cinc botons. Per ordre d'aparició, aquests són:

1. Llistat d'entrades del formulari, on es mostra la mateixa vista que la del càrrec, amb l'afegit de què és permet esborrar entrades individuals.

2. Edició del formulari, on podrem editar les característiques, continguts i procés administratiu de cada formulari. En parlarem més a continuació.

3. Duplicar formulari, on es crea un nou formulari idèntic estructuralment a un altre d'existent però sense les dades introduïdes.

4. Esborrar formulari, on s'esborra tant el formulari com les sol·licituds d'aquest. Per evitar errors es demana confirmació prèviament.

5. Obtenció de sol·licitud PDF buida, on es descarrega un document PDF que representa un formulari buit per omplir en paper. Aquesta opció es dóna per si es necessita omplir alguna sol·licitud en paper.

La interfície més complexa és la d'edició del formulari. En aquesta veiem una segona

barra de navegació amb els tres tipus d'informació editable: informació general, camps (o document base en els formularis externs) i procés administratiu.

La informació general d'un formulari inclou coses com el títol del formulari i la seva descripció, així com diverses caselles. Aquestes indiquen, per ordre d'aparició: si el formulari està obert al públic, si necessita autenticació del sol·licitant, si segueix el procés administratiu definit (el qual inclou la generació de PDFs i signatures electròniques), l'habilitació de notificacions de càrrecs i sol·licitants i, finalment, si el formulari té caducitat i, si és així, quan expira. En el cas dels formularis externs l'autenticació i el seguiment del procés és obligatoria, ja que per força necessitem tractar amb documents PDF.

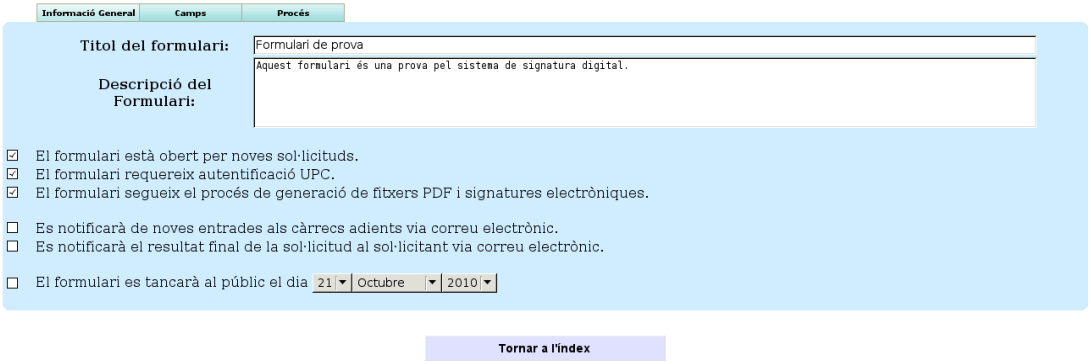

Figura C.10: Edició de la informació general d'un formulari.

La segona pestanya detalla els camps del formulari, o el document base en el cas dels externs. Els camps del formulari intern es visualitza com una llista (figura C.11), on es pot editar la informació necessària per cada camp: nom del camp, tipus, valors possibles (aplicable en el cas dels tipus de multiopció o desplegables), descripció, si és obligatori i si és depenent. Finalment al costat de cada camp hi ha botons per eliminar-lo i per crear-ne un de nou a continuació d'aquest.

La majoria d'opcions dels camps només afecten al sol·licitant restringint quin tipus d'informació pot introduir i de quina manera. Ja hem vist a la taula C.12 el què fan els diferents tipus. En quant als seus atributs, un camp obligatori sense omplir per l'usuari resulta en un avís sense deixar continuar a la següent pàgina fins que s'ompli, mentre que un camp depenent no estarà disponible a no ser que el camp anterior més proper de tipus casella no estigui seleccionat (exemple C.13).

El cas dels formularis externs és més simple: senzillament es tracta d'un botó per inserir el document base desitjat del sistema de fitxers local.

Finalment trobem el subapartat del procés administratiu. Aquest consta d'una llista de càrrecs, amb caselles per indicar si a cadascú li fa falta signar o no. També tenim, de forma similar als camps, botons per eliminar i afegir càrrecs al procés actual.

El procés administratiu funciona de la següent manera: comença amb el sol·licitant, del que es pot demanar la signatura o no. Si és així el procés no continua fins que aquest

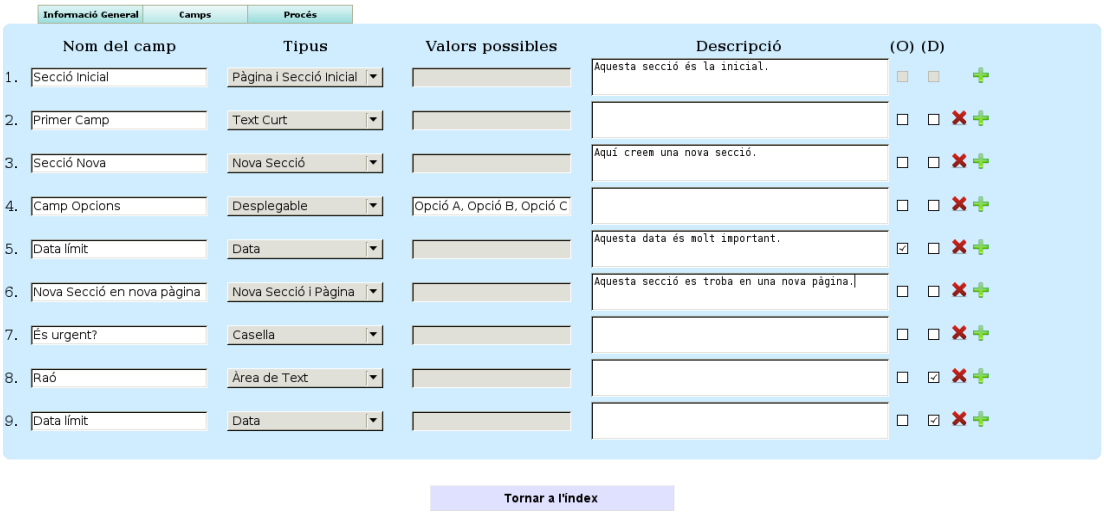

Figura C.11: Edició dels camps d'un formulari.

hagi signat. Una vegada ha acabat es notifica als càrrecs de forma seqüencial i en l'ordre establert, fins arribar a un del què es necessiti la firma. En aquest cas el procés queda parat i no continua fins que aquest hagi signat. El procés segueix fins arribar al final de la llista (on la sol·licitud queda acceptada) o algun càrrec decideix rebutjar la sol·licitud. Les notificacions esmentades en aquest paràgraf només es donaran si són habilitades a la informació general del formulari.

Text curt:	Un text curt (una línia) per frases curtes.
Número:	Introducció exclusiva de nombres, amb un format automàtic (com punts separadors dels milers).
Àrea de text:	Text més llarg que el text curt, per redactar una petita explicació.
Casella:	Resposta sí/no. Pot activar o desactivar camps posteriors si aquests són marcats com a depenents.
Data:	Petita finestra amb un calendari per la senzilla introducció de dates.
Desplegable:	Llista desplegable per escollir entre diferents opcions.
Multiopció exclusiva:	Conjunt de valors dels què només es pot escollir un. Funcionalment equivalent a l'anterior, però amb una visualització diferent.
Multiopció no exclusiva:	Conjunt de valors dels què es poden escollir diversos.
Nova secció:	Marcador perquè els camps posteriors es trobin en una nova secció, dins la mateixa pàgina.
Nova secció i pàgina:	Marcador perquè els camps posteriors es trobin en una nova secció d'una nova pàgina.

Figura C.12: Tipus de camps definits i el seu ús.

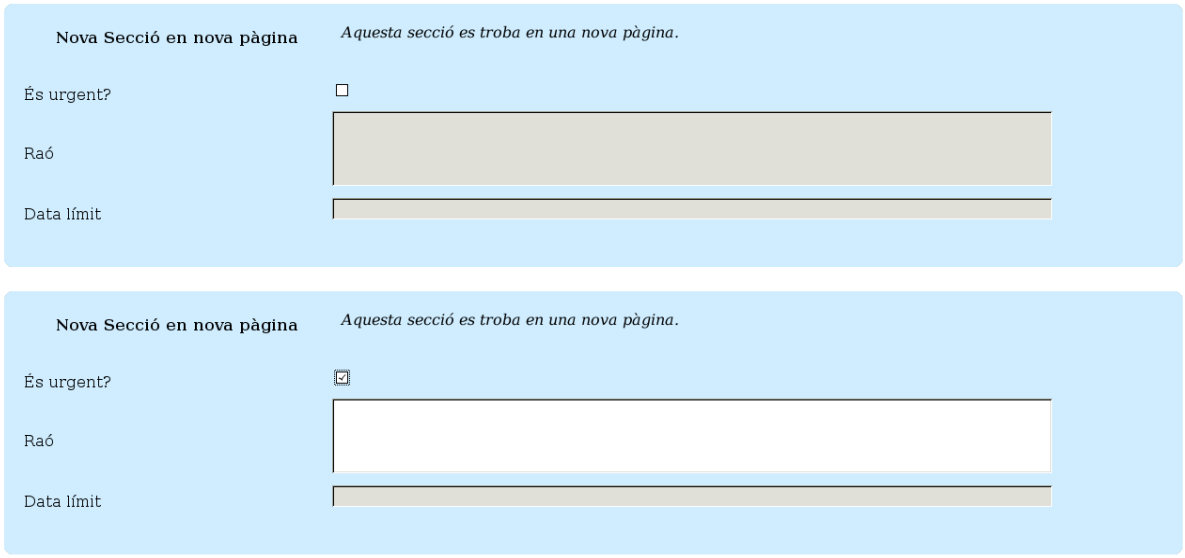

Figura C.13: Funcionament d'un camp depenent (Raó) respecte el camp de tipus casella anterior (Urgent?).

Figura C.14: Edició del procés administratiu d'un formulari.

Appendix D

Configuració

D.1 Paràmetres web

Al fitxer **init.php** de l'arrel de la web es troben definides un conjunt de funcions útils, l'inicialització de valors necessaris, així com variables que defineixen l'entorn utilitzat. En concret hi ha les adreces de la web, tant la URL com la ruta local del servidor on es troba, ports, adreces del contenidor de servlets (per la generació de PDFs) i la ruta al logotip de Matemàtica Aplicada II (per passar-li a l'applet).

D.2 Paràmetres de la signatura digital

Com hem comentat a l'apartat de disseny, l'applet de signatura digital accepta una gran varietat de paràmetres pel seu funcionament. Pel nostre sistema no serà necessari modificar-los ja que han estat adaptats per funcionar de la manera que hem establert.

Tot i així esmentarem algún paràmetre que pugui ser interessant conéixer per possibles consideracions futures. El codi de crida a l'applet, junt amb els seus paràmetres es troba a l'arrel de la web, en el fitxer **appletFirma.php**.

signature_mode: Identificador del tipus de signatura a fer (CAdES, XAdES, PAdES, ... amb els seus diferents perfils). Estan definits a la documentació adjunta a la "Eina web de signatura-e" del CATCert [18].

pdf_location: Camp on es posa el lloc on s'ha signat. Actualment està posat a Barcelona, per la localització física del servidor que rep les peticions.

pdf_reserved_space: Espai a reservar dins del fitxer per la signatura (tal com hem explicat al capítol de criptografia). De moment hi ha espai suficient per qualsevol signatura, però si en el futur hi ha canvis importants (com certificats digitals amb

més informació) pot ser necessari augmentar aquest valor, augmentant també la mida total del fitxer signat.

D.3 Paràmetres de la base de dades

Al directori "dades" de la web trobem diferents fitxers de configuració relatius a l'accés a la base de dades. Dels interessants per la seva edició trobem **defCredencialsBD.php**, amb el nom i ubicació de la base de dades, junt amb l'usuari i password per entrar, i també el fitxer **defDB.php**, amb informació de com utilitzar la base de dades de forma relativa al nostre sistema (actualment només es troba el número de camps de metainformació en les taules de formularis: data, nom del sol·licitant, estat, ...).

```
<?php  // defCredencialsDB.php
        $servidor='localhost';
        $usuari='xxx';
        $password='yyy';
        $nomBaseDades='formularis';
?>
```

```
<?php    // defDB.php
// Definició d'informació rellevant per la base de dades
        include 'defCredencialsDB.php';

        // Número de columnes de les taules fm_* de la BD que no representen
        // informació dels camps del formulari (id, data, ip, usuari, numUsuari,
        // proces)
        $numMetaColumnes = 7;
?>
```

D.4 Càrrecs i gestors

Com hem mencionat durant el desenvolupament del projecte, hi ha dos paràmetres que hem decidit excloure de la interfície web per la seva sensibilitat: els càrrecs i els usuaris amb permisos de gestor.

Els càrrecs es troben al fitxer **defCarrecs.php**, dins del directori *dades* de l'arrel de la web. Aquest té la següent estructura:

```
<?php

        // Com a pseudocarrec, el 0 es el Sol·licitant
        $carrecsCanon[0] = 'Sol·licitant';
        $carrecsCanon[1] = 'Cap d\'administració';
        $carrecsCanon[2] = 'Director';
        $carrecsCanon[3] = 'Sotsdirector';
        $carrecsCanon[4] = 'Secretari Acadèmic';
?>
```

Veiem que és una simple declaració de variables indexades, amb el nom canònic del càrrec. El nom es refereix al nom del propi càrrec, el nom de la persona que ocupa el càrrec és editable via web. Per seguretat hem optat per indexar explícitament els càrrecs perquè el número del càrrec és important; si s'ha de restaurar en el futur serà important tenir-los en compte.

De forma semblant tenim el fitxer de configuració **defUsuarisAdmin.php** a l'arrel de la web, amb la següent estructura:

```
<?php   // Llista d'usuaris amb permisos administratius
// (creació/edició de formularis i càrrecs)

        $usuarisAdmin = array('david.gomez-guillen', 'marc.andreu',
           'didac.guardia', 'fernando.martinez');
?>
```

En aquest cas l'ordre no és important així que només cal afegir (o treure) l'usuari (concretament el nom d'usuari UPC) a la llista existent. Perquè es vegin reflexats els canvis és necessari que l'usuari es desconnecti i torni a entrar al sistema.

Appendix E

Configuració per l'ús de targetes criptogràfiques

E.1 Instal·lació

Per poder utilitzar les targetes criptogràfiques, tant per autenticar-se com per signar, necessitarem fer dues coses.

En primer lloc caldrà instal·lar el controlador del lector de targetes. Segons el model i tipus del lector el controlador variarà, així que la millor opció és consultar el manual del lector. En el cas de Linux és bastant comú el *driver* CCID, disponible en la majoria de repositoris de les diferents distribucions.

En segon lloc cal instal·lar els controladors de les targetes criptogràfiques (carnet UPC [40] i/o DNI electrònic [39]). En Windows el procediment és igual que la instal·lació de qualsevol altre software, però el cas de Linux és una mica més complex.

Els drivers de linux, tant del carnet UPC com del DNIe, estan basats en la llibreria openSC [34]. Aquesta ofereix una plataforma amb la que desenvolupar drivers per targetes específiques per mitjà de *plugins*, de manera que es parteix d'una base comuna a tota targeta per estalviar feina. Per utilitzar un *driver* basat en openSC primer haurem d'instal·lar el paquet ***opensc***, disponible en la majoria de repositoris linux. Disponible també en els repositoris es troba el paquet ***pcscd***, incorporant l'especificació PC/SC a l'entorn linux i permetent la comunicació a baix nivell amb el lector de targetes.

Cal tenir en compte que, en el cas de Linux, s'han donat casos d'algunes versions del *driver* del DNI electrònic que no són compatibles amb algunes versions de openSC, així que si no funciona el DNIe pot ser que s'hagi de tenir en compte quina versió es té instal·lada i documentar-se sobre problemes d'incompatibilitat coneguts.

Com a referència, la ruta típica (per defecte) dels drivers són:

```
Carnet UPC (Windows): C:\Archivos de programa\Gemalto\Classic Client\BIN\gclib.dll
Carnet UPC (Linux): /usr/lib/pkcs11/libgclib.so
DNIe (Windows): C:\Windows\System32\UsrPkcs11.dll
DNIe (Linux): /usr/lib/libopensc-dnie.so
```

(Pot ser que en alguns casos s'instal·lin en rutes semblants, e.g. Program Files, Archivos de programa (x86), /usr/lib64/...)

Una vegada tot està instal·lat, podem passar a la configuració.

E.2 Configuració

En aquest cas distingirem els dos usos de la targeta (autenticació i signat). Per aquest últim no fa falta cap tipus de configuració especial: només assegurar-se que es disposa d'alguna versió de la **màquina virtual de Java** per tal de poder executar *l'applet* de signat de la web.

En el cas de l'autenticació hi ha molts casos en què tampoc fa falta cap configuració addicional: tots els navegadors web en Windows (excepte Firefox) poden utilitzar-la de forma automàtica ja que accedeixen al magatzem de claus del sistema operatiu. En Firefox es necessita una configuració manual: al menú de "Preferències" → "Avançat" → "Xifratge" → "Dispositius de seguretat" es defineixen les targetes criptogràfiques amb el botó "Carregar" i afegint un nom identificatiu i la ruta del controlador (llistats a la secció anterior els valors típics).

A l'hora de redactar aquest document, el navegador Opera no té suport per autenticació via targeta, així que no és compatible per aquest ús en el nostre sistema. Cal advertir també que els navegadors Google Chrome i Safari semblen tenir algun *bug* compartit (tenen el mateix motor de renderitzat, WebKit) que provoca que els applets Java no es puguin comunicar bé amb javascript, de manera que, a menys que s'actualitzi properament, no són compatibles amb la signatura per mitjà de la web de formularis.

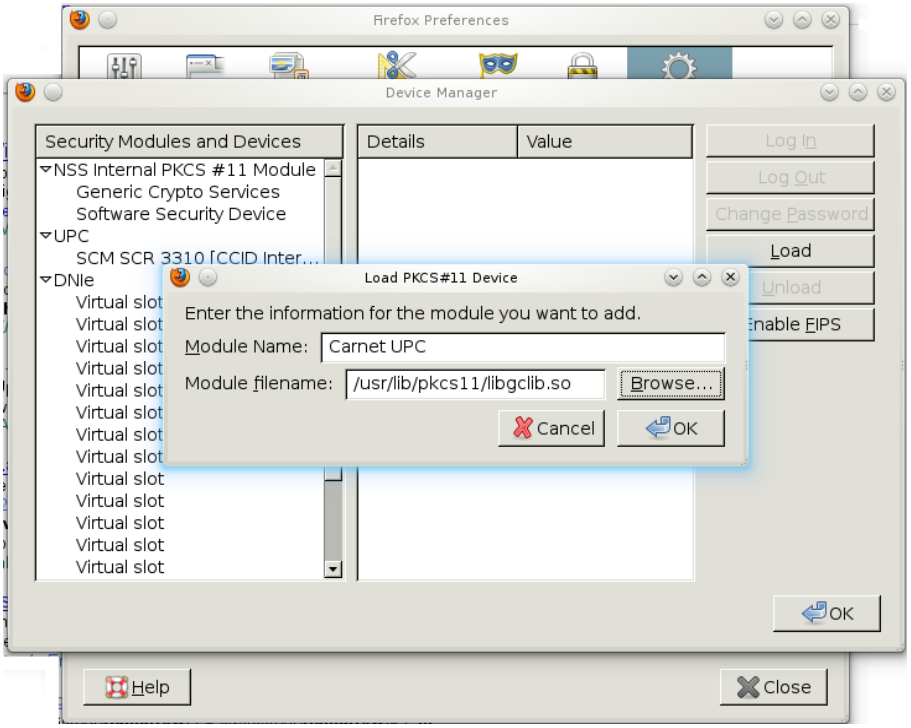

Figura E.1: Configuració de targetes criptogràfiques en Firefox.

Appendix F

Migració del sistema

Aquesta secció la dedicarem a resumir quins fitxers i altra informació es necessita preservar per la migració del sistema a un nou entorn. Ho farem a mode de llista:

1. **Base de dades:** Base de dades "formularis", tot preservant el tipus de les taules (innoDB vs MyISAM). No conservar els tipus provocarà el malfuncionament del sistema.
2. **Apache + SSL:** Configuració d'Apache i certificats SSL del servidor, on es troba el mapeig de directoris locals a adreces web. Hi trobarem definit també l'accés segur i la petició del certificat client per l'autenticació (/etc/apache2).
3. **LDAP:** Configuració i certificats necessaris per l'accés LDAP a servidors de la UPC (/etc/ldap).
4. **Tomcat:** Fitxers de configuració així com les *webapps* (/usr/share/tomcat5.5).
5. **Web:** Tots els fitxers PHP i recursos relacionat (com imatges, scripts i, sobretot, l'applet de signatura). Localitzat a /home/formularis, junt amb el directori dels documents PDF i el script de notificacions.
6. **Sol·licituds PDF:** Veure Web.
7. **Script de notificacions:** Veure Web.

Glossari

AC: Veure *Autoritat de certificació*.

AC RAIZ DNIE: Autoritat arrel de la PKI del DNI electrònic.

AC DNIE 001, 002, 003: Autoritats intermitges de la PKI del DNI electrònic.

AR: Veure *Autoritats de registre*.

ASN.1: *Abstract Syntax Notation One*. Notació utilitzada per representar estructures de dades de forma independent a la codificació.

Autoritat arrel: Autoritat de certificació amb un certificat signat per ella mateixa. En aquestes s'originen les cadenes de certificació i es necessita confiança incondicional per part del sistema de validació.

Autoritat de certificació: Entitats que mantenen el funcionament d'una PKI. Els seus serveis inclouen el registre, creació, emissió, validació i revocació de certificats digitals.

Autoritat de registre: Entitats en les que es delega el registre d'informació per l'emissió de nous certificats.

Autoritat de segellat de temps: Entitats que emeten segells de temps segurs.

Autoritat de validació: Entitats en les que es delega la validació de certificats.

AV DNIE FNMT: Autoritat de validació de la PKI del DNI electrònic.

AV: Veure *Autoritat de validació*.

BER: *Basic Encoding Rules*. Un dels conjunts de regles per la codificació de dades en ASN.1.

Cadena de certificació: Conjunt de certificats amb una relació jeràrquica de certificats emissors/emesos. Comença sempre per una autoritat arrel fins acabar amb un certificat no pertanyent a una autoritat (e.g. del carnet UPC o DNI electrònic.

CAdES: *CMS Advanced Electronic Signature*. Ampliació de l'estàndard CMS per la creació de signatures complint amb les directives europees apropiades.

Camp: Parts de les que està format un formulari. Aquests inclouen tant tipus de dades per introduir (text, número, data, ...) com marcadors per la seva presentació (nova secció i nova pàgina).

Camp depenent: Camp en el que només és possible la introducció de dades si el camp de tipus casella anterior (en ordre) a aquest està seleccionat.

Camp obligatori: Camp que necessita un valor per la realització d'una sol·licitud.

Càrrec: Càrrecs oficials del departament (Director, sotsdirector, secretari acadèmic i cap d'administració). Alternativament també podem referir-nos amb aquest terme a les persones que consten com a tals càrrecs dins del sistema.

Certificat digital: Document digital que conté informació d'un titular junt amb una clau pública pel seu ús en una PKI. Són emesos per autoritats certificadores.

CMS: *Cryptographic Message Syntax*. Sintaxis per la representació de signatures digitals, derivada de PKCS#7.

CRL: *Certificate Revocation List*. Llistes publicades per les autoritats de validació amb informació dels certificats actualment revocats.

CSS: *Cascading Style Sheets*. Fulles d'estil que descriuen la visualització d'una pàgina web (HTML). El seu ús és motivat per una separació entre contingut i presentació.

Component: Part integrant del sistema amb una funcionalitat pròpia i (relativament) independent. El sistema desenvolupat consta de diversos components interaccionant entre ells.

DER: *Distinguished Encoding Rules*. Conjunt de regles de codificació derivat de BER per notació ASN.1. S'imposa la restricció addicional de que cada estructura té una codificació única.

Document base: Document de qualsevol tipus que serveix de base perquè un sol·licitant pugui descarregar-se'l, omplir-lo i enviar-lo al sistema com a fitxer PDF.

EC-ACC: Autoritat arrel de la PKI de l'agència catalana de certificació.

EC-UR: Autoritat intermitja de la PKI de l'agència catalana de certificació; l'emissora de tots els certificats de carnets UPC.

Formulari: Informació relativa a la manera de realitzar un tipus de sol·licitud concreta. Això inclou (en l'àmbit del projecte) informació general com títol i descripció, contingut dels camps a omplir i procés administratiu a seguir.

Gestor: Rol d'usuari del sistema amb permís per administrar formularis i càrrecs: crear, editar, esborrar, ...

Informació de validació: Tota aquella informació que demostra que un certificat és vàlid en un instant de temps concret. Això pot incloure tant CRLs com respostes OCSP amb un segell de temps.

LDAP: *Lightweight Directory Access Protocol.* Protocol per consultar directoris d'informació remotament (mitjançant IP). És utilitzat en el projecte per la validació del nom d'usuari i password UPC.

No repudi: En l'àmbit de la signatura electrònica, garantia de que l'usuari ha signat un document per voluntat pròpia. Amb aquesta característica una persona no pot negar una signatura seva a posteriori.

MIME: *Multipurpose Internet Mail Extensions.* Extensió del correu electrònic pel suport d'altres tipus de dades. Un dels primers estàndards criptogràfics de clau pública es va definir a partir d'aquesta extensió, creant S/MIME.

OCSP: *Online Certificate Status Protocol.* Protocol de tipus pregunta / resposta per obtenir l'estat de validesa d'un certificat en un instant de temps.

PAdES: *PDF Advanced Electronic Signature.* Familia d'estàndards de signatura electrònica per documents PDF. És el tipus de signatura que implementarem en aquest projecte per la seva integració amb el propi fitxer.

PDF: *Portable Document Format.* Format electrònic per la representació de documents de forma digital, optimitzat per la seva visualització en pantalla.

PEM: Un dels primers protocols per signar correu electrònic mitjançant criptografia de clau pública. No va arribar a una gran difusió, però va derivar amb el protocol CMS.

PKCS: *Public Key Cryptographic System.* Llista d'estàndards criptogràfics publicats per l'empresa RSA Security. D'especial interés són el PKCS#1 (algoritme RSA), el PKCS#7 (CMS) i el PKCS#11 (API d'accés a targetes criptogràfiques).

PKI: *Public Key Infraestructure.* Conjunt de serveis, polítiques i entitats (com les autoritats de certificació) que fan possible l'ús de certificats digitals i, per tant, la criptografia de clau pública.

Procés: En l'àmbit d'aquest projecte, la llista ordenada de càrrecs que rebran notificació i, possiblement, necessitin signar les sol·licituds entrants d'un formulari concret.

Procés administratiu: Veure *procés*.

RSA: Algoritme criptogràfic de clau pública, anomenat així pels seus creadors (Rivest, Shamir i Adleman).

S/MIME: *Secure MIME*. Estàndard pel xifrat i signat de dades MIME.

Segell de temps: Missatge signat on s'especifica el temps i hora en què es va fer una petició. Es sol utilitzar per demostrar que unes dades han estat emeses en un instant de temps determinat.

Servei web: Veure *Web service*.

Sol·licitud: En aquest projecte, són les dades creades per un sol·licitant amb la informació que demana un formulari.

SHA: *Secure Hashing Algorithm*. Algoritme de *hashing* utilitzat en el projecte. Hi ha versions de 160, 256, 384 i 512 bits.

Signatura digital: Dades que relacionen de forma unívoca una persona amb unes dades.

Signatura electrònica: Signatura digital amb informació addicional per garantir l'autenticitat i el no repudi del signant.

SOAP: *Simple Object Access Protocol*, protocol basat en XML per comunicar informació estructurada en un entorn de serveis web, per exemple peticions i respostes.

Sol·licitant: Rol de l'usuari que envia sol·licituds.

SSL: *Secure Sockets Layer*. Es tracta d'un sistema de xifratge de comunicacions utilitzant criptografia de clau pública. L'utilitzarem per la comunicació web segura.

Subsistema: Veure *component*.

Targeta criptogràfica: Targeta amb un xip integrat pel seu ús criptogràfic. Aquest conté el certificat digital associat al titular de la targeta, així com el circuit pel signat d'informació amb la clau secreta.

Timestamp: Veure *segell de temps*.

Virtualització: Filosofia consistent en la separació d'un recurs físic en diversos recursos lògics (com diverses màquines virtuals en una sola física).

Web Service: Interfície d'ús d'un software, on es realitzen peticions i es reben respostes de forma remota. En aquest paradigma s'entén l'accés al software com un servei ofert per una entitat a una altra.

XAdES: *XML Advanced Electronic Signature*. Familia d'estàndards de signatura electrònica per documents XML.

XSLT: *Extensible Stylesheet Language Transformations*. Llenguatge declaratiu que especifica com transformar un document XML en un altre.

XSL-FO: *Extensible Stylesheet Language - Formatting Objects*. Llenguatge declaratiu utilitzat per generar altres tipus de documents, com per exemple PDF.

Bibliografía

[1] OCSP Profile (RFC 2560). http://www.ietf.org/rfc/rfc2560.txt.

[2] CRL Profile (RFC 3280). http://www.ietf.org/rfc/rfc3280.txt.

[3] CAdES (RFC 5126). http://tools.ietf.org/html/rfc5126.html.

[4] CMS (RFC 5652). http://tools.ietf.org/html/rfc5652.

[5] AuthProvider (Java Platform SE 6). 2010. http://download.oracle.com/javase/6/docs/api/java/security/AuthProvider.html.

[6] Paul C. van Oorschot Alfred J. Menezes and Scott A. Vanstone. Handbook of applied cryptography. 2010. http://www.cacr.math.uwaterloo.ca/hac/.

[7] Principles behind the Agile Manifesto. 2010. http://agilemanifesto.org/principles.html.

[8] Brassard Bennett, Charles H.; Gilles. Quantum cryptography: Public-key distribution and coin tossing. 1984. Proceedings of IEEE International Conference on Computers, Systems and Signal Processing 1984. IEEE Computer Society. pp. 175-179.

[9] Simple Calendar Widget by Anthony Garrett. 2010. http://www.tarrget.info/calendar/scw.htm.

[10] Chosen ciphertext attack. 2010. http://en.wikipedia.org/wiki/Chosen-ciphertext_attack.

[11] PDF/Archive Committee. 2010. http://aiim.org/Resources/Standards/Committees/PDFA.

[12] Quantum Computing and Shor's Algorithm. 2010. http://alumni.imsa.edu/~matth/quant/299/paper/index.html.

[13] World Wide Web Consortium. http://www.w3.org/.

[14] BOE 304 de 20/12/2003 Sec 1 Pag 45329 a 45343. www.boe.es/boe/dias/2003/12/20/pdfs/A45329-45343.pdf.

[15] CATCert Plataforma de serveis d'identificació i signatura (PSIS). 2010. http://www.catcert.cat/web/cat/1_4_3_plataforma.jsp.

[16] Directiva 1999/93/CE del Parlamento Europeo y del Consejo. https://www.sede.fnmt.gob.es/sede/normas/Directiva-199-93-CE.pdf.

[17] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). 2010. http://www.w3.org/TR/soap12-part1/.

[18] CATCert Eines. 2010. http://www.catcert.cat/web/cat/6_6_eines.jsp.

[19] InnoDB Website InnoDB Features. 2010. http://www.innodb.com/products/innodb/features/.

[20] Sinadura La firma digital libre. 2010. http://www.sinadura.net/.

[21] Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA–3) Family. 2010. http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf.

[22] Lobbying for smartcard/ID card support in Opera. 2010. http://www.jaanuskase.com/en/2007/03/lobbying_for_smartcardid_card.html.

[23] PDF format becomes ISO standard. http://www.iso.org/iso/pressrelease.htm?refid=Ref1141.

[24] Microsoft Office Binary File Formats. http://www.microsoft.com/interop/docs/OfficeBinaryFormats.mspx.

[25] Hash function. 2010. http://www.itl.nist.gov/div897/sqg/dads/HTML/hash.html.

[26] Trapdoor One-Way Function. 2010. http://mathworld.wolfram.com/TrapdoorOne-WayFunction.html.

[27] RSA Laboratories What is the discrete logarithm problem? 2010. http://www.rsa.com/rsalabs/node.asp?id=2193.

[28] Jetty. 2010. http://www.eclipse.org/jetty/.

[29] Alfonso González Jordi Sala. Avaluació de solucions de gestió de la identitat. 2008. https://www.upc.edu/identitatdigital/certificatdigital/sistemes-dinformacio/infraestructura-soa/DesplegamentServiceMix.pdf.

[30] A. Jøsang and B. AlFayyadh. Robust wysiwys: A method for ensuring that what you see is what you sign. 2008. Proceedings of the Australasian Information Security Conference (AISC'08), Wollongong, Australia.

[31] RSA Laboratories. 2010. http://www.rsa.com/rsalabs/node.asp?id=2125.

[32] PAdES Long Term Validation (LTV). http://www.etsi.org/deliver/etsi_ts/102700_102799/10277804/01.01.01_60/ts_10277804v010101p.pdf.

[33] Factorization of a 768-bit RSA modulus. 2010. http://eprint.iacr.org/2010/006.

[34] OpenSC. http://www.opensc-project.org/opensc.

[35] Versions Portable Document Format. 2010. http://en.wikipedia.org/wiki/Portable_Document_Format\#Versions.

[36] The TLS Protocol. http://www.ietf.org/rfc/rfc2246.txt.

[37] Adobe Adobe Reader. http://get.adobe.com/reader/.

[38] ETSI PDF Advanced Electronic Signature. http://stf364ms.e.ac.upc.edu/phpmyfaq/index.php?action=artikel\&cat=1\&id=43\&artlang=en.

[39] Portal Oficial sobre el DNI electrónico Área de descargas. http://www.dnie.es/descargas/index.html.

[40] Ja tinc el certificat. I ara què ... ? 2010. http://www.upc.edu/identitatdigital/certificatdigital/que\%20he\%20de\%20fer\%20per\%20utilitzar-el-certificat.

[41] Introduction to ISO 27002. http://www.27000.org/iso-27002.htm.

[42] Java Applet to JavaScript communication (liveconnect) is not working. 2010. http://code.google.com/p/chromium/issues/detail?id=580.

[43] ResourceBundle (Java 2 Platform SE v1.4.2). 2010. http://download.oracle.com/javase/1.4.2/docs/api/java/util/ResourceBundle.html.

[44] X509Certificate (Java 2 Platform SE v1.4.2). 2010. http://download.oracle.com/javase/1.4.2/docs/api/java/security/cert/X509Certificate.html.

[45] Breaking RSA Encryption with a Quantum Computer: Shor's Factoring Algorithm. 2010. http://people.ccmr.cornell.edu/~mermin/qcomp/chap3.pdf.

[46] ASN.1 Specification (X.683). http://www.itu.int/rec/T-REC-X.683/en.