Autoritat de certificació PKI amb serveis en línia

Lluís Moran i Capdevila
ETIS

Carlos Ares Angulo

10 de juny de 2011

A la Txell, per haver patit el procés d'elaboració d'aquest projecte.

Als meus pares i la resta de la meva família, pel suport que m'han donat durant tota la carrera.

Resum

La finalitat d'aquest projecte és la creació d'una aplicació que faci les funcions d'una **Autoritat de Certificació** (CA) dins del marc d'una **Infraestructura de Clau Pública** (PKI). Aquesta aplicació té una interfície web que ofereix una sèrie de serveis en línia.

Les operacions que ofereix seran de dos tipus:

- **Públiques:** d'accés obert i que permet obtenir **certificats** mitjançant fitxers de petició **PKCS#10** o introduint les dades de l'usuari. També permet descarregar la **Llista de Revocació de Certificats** (CRL) i el certificat arrel de la CA.
- **Privades:** on l'usuari s'haurà d'autenticar prèviament com a administrador, ja sigui amb una connexió local i l'ús d'un nom d'usuari i una contrasenya o amb la presentació d'un certificat autoritzat. Les funcions que es podran dur a terme inclouen la posada en marxa de l'aplicació, la seva reinicialització, la gestió de drets d'administrador, la consulta de certificats, la revocació d'aquests o la publicació de la CRL.

Les operacions internes de l'aplicació estan separades de la interfície amb l'objectiu de que es puguin fer servir des d'un altre programari.

El producte ha estat desenvolupat en llenguatge **Java** i s'executa mitjançant un contenidor de servlets **Apache Tomcat** sobre **Linux** (tot i que seria fàcilment portable a altres contenidors i sistemes operatius). Es fa servir una base de dades **MySQL** per emmagatzemar la informació de l'aplicació.

L'objectiu del projecte és aprofundir en els coneixements sobre Infraestructures de Clau Pública i en les funcions i serveis que ha d'oferir una Autoritat de Certificació, posant un especial èmfasi en tot el referent a la seguretat del sistema.

Índex de continguts

Resum **3**
Índex de continguts **4**
Índex de figures **6**
1 – Introducció **7**
1.1 – Justificació del TFC 7
1.1.1 - Estat de l'art 8
1.1.2 - Objectius personals i experiència prèvia 9
1.2 – Objectius del TFC 9
1.3 – Enfocament i mètode seguit 10
1.4 – Planificació del projecte 11
1.5 – Productes obtinguts 13
1.6 – Descripció de la resta de capítols de la memòria 13
2 – Descripció general de l'aplicació **14**
2.1 – Estructura de l'aplicació 14
2.1.1 – Estructura de la base de dades 19
2.1.2 – Estructura de fitxers 21
2.2 – Funcions 23
2.3 – Requisits de programari 24
3 – Instal·lació de l'aplicació **25**
3.1 – Instal·lació 25
3.2 – Posada en marxa 25
3.3 – Reinicialització 32
4 – Funcions públiques de l'aplicació **33**
4.1 – Petició de certificats a partir d'un fitxer PKCS#10 33
4.2 – Petició de certificats a partir de les dades d'usuari 35
4.3 – Descàrrega de la llista de revocació de certificats (CRL) 37
4.4 – Descàrrega del certificat arrel de l'Autoritat de Certificació 38
5 – Funcions d'administrador de l'aplicació **39**
5.1 – Ingrés a l'aplicació com a administrador 39

5.1.1 – Ingrés local mitjançant contrasenya 41
5.1.2 – Ingrés mitjançant certificat autoritzat 42
5.2 – Posada en marxa 42
5.3 – Reinicialització 44
5.4 – Consulta de dades dels certificats 45
5.4.1 – Cerca de certificats emesos per la CA 46
5.4.2 – Cerca de certificats revocats 49
5.5 – Gestió d'administradors 52
5.6 – Revocació de certificats 53
5.7 – Publicació de la llista de revocació de certificats (CRL) 56
6 – Conclusions 58
Glossari 59
Bibliografia 60

Índex de figures

Figura 1: Parts principals de l'aplicació 15
Figura 2: Diagrama de classes 16
Figura 3: Mètodes i atributs de les classes 17
Figura 4: Interfície – Àrea de notificacions 18
Figura 5: Interfície – Emissió d'un certificat a partir d'una petició PKCS#10 . . 34
Figura 6: Interfície – Emissió d'un certificat a partir de les dades d'usuari 36
Figura 7: Importació de la CRL al navegador 38
Figura 8: Interfície – Pàgina principal 40
Figura 9: Interfície – Menú d'administrador 41
Figura 10: Interfície – Posada en marxa de l'aplicació 43
Figura 11: Interfície – Reiniciar l'aplicació 44
Figura 12: Interfície – Opcions de consulta 46
Figura 13: Interfície – Cerca de certificats emesos 47
Figura 14: Interfície – Resultat de la cerca de certificats emesos 48
Figura 15: Interfície – Cerca de certificats revocats 50
Figura 16: Interfície – Resultat de la cerca de certificats revocats 51
Figura 17: Interfície – Gestió d'administradors 52
Figura 18: Interfície – Revocació d'un certificat emès 54
Figura 19: Interfície – Confirmació de revocació 55
Figura 20: Interfície – Publicació de CRL 56

1 – Introducció

1.1 – Justificació del TFC

La seguretat i l'autenticació de la identitat cobren cada dia més importància en l'actual societat de la informació. Cada cop més operacions són dutes a terme per mitjans telemàtics en el dia a dia. Des d'enviar un simple correu electrònic fins a fer operacions amb el banc, passant per multitud de tràmits amb estaments públics o privats.

Dins d'aquest creixent ús de les TIC per dur a terme operacions tradicionalment fetes cara a cara, on la identificació era gairebé automàtica, el fet de confirmar que som qui diem ser té una importància cabdal. Tanmateix és primordial conservar la confidencialitat de les comunicacions perquè cap agent extern pugui aprofitar-se'n.

En aquest escenari es fa necessari un mètode per identificar-nos d'una manera fiable i solucionar el problema de la criptografia asimètrica: Com saber que la informació prové de qui ens pensem.

Els certificats digitals i les Infraestructures de Clau Pública (*Public Key Infrastructure* o PKI) ens donen una solució que cada cop pren més força, tot i ser una tecnologia relativament jove. Aquesta consisteix en que una entitat central de confiança signi l'estructura de dades que forma el certificat digital, donant-li així validesa.

Aquests certificats inclouen informació del seu propietari, la seva clau pública i l'esmentada signatura que el fa confiable. Un cop signat, el certificat digital podrà fer funcions d'autenticació, integritat, no-repudi i confidencialitat.

L'ús dels certificats digitals és molt divers i creix amb el temps. Actualment amb l'aparició del DNI electrònic (DNIe) s'està fomentant cada cop més la seva utilització. Aquest DNIe, que té la seva pròpia autoritat de certificació, inclou un certificat digital llest per a que els ciutadans el puguin fer servir per fer tràmits amb ajuntaments, proveïdors de serveis, hisenda, banca, etc... En molts casos es substitueix la clàssica identificació per usuari i contrasenya per un lector de targetes intel·ligents i una tarja que conté un certificat digital personal.

En el sector empresarial comença a ser imprescindible posseir un certificat per fer gestions amb diverses institucions públiques com la Seguretat Social o l'Agència Tributària, deixant de banda la impressió en paper de formularis i la presentació en persona d'aquests. Les pròpies institucions proporcionen programari per fer aquests tràmits de manera telemàtica, al mateix temps que moltes empreses l'inclouen en solucions comercials de gestió empresarial.

A part del DNIe, en l'àmbit estatal, cal destacar altres Autoritats de Certificació com la *Fábrica Nacional de Moneda y Timbre,* i en el nostre territori, la Agència Catalana de Certificació (CATCert) i el seu programa d'identitat digital (idCAT).

1.1.1 - Estat de l'art

L'elaboració del producte s'ha dut a terme amb el llenguatge de programació Java. Aquest llenguatge orientat a objectes es caracteritza per la seva capacitat per ser executat a través d'una màquina virtual, fet que el fa independent de la plataforma on s'executa. Això millora la portabilitat de les aplicacions i la seva execució, ja que es fa en un entorn controlat.

Per suplir les mancances de Java s'utilitzaran les llibreries de Bouncy Castle. Aquestes estan especialitzades en temes de criptografia, oferint molta més potència i funcions que l'API estàndard de Java.

L'aplicació que s'ha creat ofereix serveis en línia. Això vol dir que s'ha de poder accedir a les seves funcions mitjançant un navegador. Aquest accés s'ofereix amb un Contenidor de Servlets, programari que s'executa en un servidor i que proporciona un entorn accessible on fer córrer aplicacions, evitant que l'usuari hagi de descarregar-les per fer-les anar.

Per a la nostra aplicació farem servir Apache Tomcat, un dels contenidors de servlets de codi lliure més coneguts. També n'existeixen d'altres com Glassfish, JBoss o IBM Websphere.

Els Servlets, part important dels serveis en línia de la nostra aplicació, són objectes que s'executen des del servidor per oferir contingut dinàmic.

La informació que generi l'aplicació es guardarà en una base de dades. En aquest cas fem servir MySQL, un sistema de gestió de base de dades relacional (SGBD). Altres SGBD coneguts són PostgresSQL, Apache Derby o Microsoft SQL Server.

Els SGBD permeten crear taules on guardar, modificar i llegir informació de diferents tipus mitjançant comandes en llenguatge SQL (Structured Query Language), estàndard que es pot fer servir des de diverses aplicacions.

Per a la posada en marxa i les proves s'ha fet servir keytool i OpenSSL, dues eines de comandes que permeten generar i utilitzar certificats, claus i altres elements criptogràfics. OpenSSL també s'utilitza per oferir encriptació i confidencialitat en les connexions a l'aplicació mitjançant el protocol HTTPS.

Finalment, el desenvolupament es farà mitjançant un Entorn Integrat de Desenvolupament (IDE). Aquest programari ofereix ajudes interactives a l'hora d'escriure el codi, així com simulació de l'entorn d'execució i *debug* d'errors.

Existeixen diversos IDE al mercat com NetBeans, JBuilder o Eclipse. Per a aquest projecte s'ha fet servir aquest últim, una altra eina de codi lliure que gràcies al seu funcionament a base de plugins ofereix moltes possibilitats.

1.1.2 - Objectius Personals i experiència prèvia

Les assignatures de Seguretat en Xarxes de computadors i, sobretot, de Criptografia van generar-me un gran interès pel tema de la seguretat informàtica.

Per tant he intentat que aquest TFC serveixi per a aprofundir encara més en els coneixements adquirits en aquestes assignatures i en la consolidació de tot el que hi he aprés.

Igualment pretenc que em proporcioni experiència en la tasca de desenvolupar un projecte informàtic, com és la creació d'una aplicació, pràcticament des de zero.

La meva experiència prèvia en aquest camp es limita al que he aprés durant el temps que porto estudiant Enginyeria Informàtica de Sistemes a la UOC i la que he adquirit com a usuari de certificats digitals i d'aplicacions que els fan servir.

1.2 – Objectius del TFC

L'objectiu del TFC és aprofundir en els coneixements sobre Infraestructures de Clau Pública i en les funcions i serveis que ha d'oferir una Autoritat de Certificació (CA), posant un especial èmfasi en tot el referent a la seguretat del sistema.

Aquests serveis es desenvoluparan en línia, diferenciats entre serveis d'administració de la CA i serveis per a usuaris i aplicacions.

Tanmateix es pretén assolir coneixements en les diverses tecnologies relacionades amb el desenvolupament i posada en marxa de l'aplicació com són:

- Desenvolupament en llenguatge de programació Java i les diverses API que ofereix en temes de criptografia (Bouncy Castle, keytool).
- Ús del contenidor de servlets Apache Tomcat. L'aplicació funcionarà amb pàgines JSP i servlets.
- Ús de bases de dades a través de Java, mitjançant l'API JDBC, que farem servir per guardar tota la informació dels certificats i la configuració de la CA.
- Utilització de l'entorn integrat de programació Eclipse.

Tenint tot això en compte, el producte final a obtenir és una aplicació que generi una CA que tingui capacitat per emetre i gestionar certificats digitals X.509, i amb una interfície web per dur a terme les següents operacions:

- **Serveis per a administrador** (Caldrà implementar autenticació i permetre execució local si el mecanisme és feble, o remota mitjançant certificat digital):
  - Configuració de posada en marxa: creació de claus i certificat de la CA.
  - Consulta de tots els certificats emesos i cerca sobre els seus camps identificadors.
  - Revocació de certificats emesos.

- Publicació d'una Llista de Certificats Revocats (CRL).

► **Serveis públics:**

- Emissió de certificats a partir de peticions de certificació PKCS#10.

- Emissió de certificats i creació de claus en objectes PKCS#12.

A part d'aquest objectius concrets, cal afegir els referents a la planificació i desenvolupament de la metodologia de treball a seguir, la investigació i aprenentatge de les eines a fer servir, i l'elaboració de la documentació de l'aplicació i de les tasques realitzades durant el projecte en forma de la memòria final i la presentació.

1.3 – Enfocament i mètode seguit

Com es veurà en el pla de treball, l'enfocament i la temporalitat seguits en aquest projecte s'han basat en gran mesura en els lliuraments de les PAC. Aquestes m'han servit per dividir la feina i poder repartir-la en el temps d'una manera més eficient.

De tota manera, el desconeixement inicial de moltes de les tecnologies que s'han fet servir ha provocat que m'hagi encallat en alguns moments i hagi perdut més temps del desitjable. Per aquest motiu he decidit optar per intentar millorar i provar l'aplicació en comptes de fer les ampliacions de l'enunciat que tenia previstes.

La correcció de les PAC m'ha servit molt per detectar errors, sobretot en el disseny o la presa de decisions. Alguns d'aquests errors, que ha calgut modificar després d'una primera planificació, han estat:

► En principi havia de fer servir pàgines HTML creades línia a línia des dels servlets, però finalment vaig decidir fer servir pàgines JSP (Tot i que en desconeixia totalment el funcionament i la sintaxi). Això ha facilitat molt la implementació de la interfície i sobretot la posterior correcció de problemes quan ha fet falta.

► Eliminació de l'ús d'un servidor web Apache davant del contenidor de servlets. El consultor em va fer veure que no era necessari tot i que en un entorn de producció potser sí que es faria servir.

► Ús d'una connexió segura per fer servir l'aplicació que no havia previst a la planificació i que és necessària per seguretat donada la naturalesa del tipus de dades que es transmeten.

► El format de les taules de mySQL ha sofert diversos canvis durant la creació de l'aplicació a mida que anaven sorgint noves necessitats: Camps que sobraven o faltaven, mides incorrectes de camps, ...

► Inclusió d'un fitxer d'opcions de configuració modificables per l'usuari que fan servir tant l'*script* d'inicialització com l'aplicació.

► Modificació de l'estructura de carpetes original per evitar-ne l'accés, mitjançant un "security constraint" a la configuració de Tomcat.

Algunes de les coses que més problemes m'han donat han estat:

- La configuració de l'entorn de desenvolupament. Concretament el funcionament del servidor de Tomcat dins d'Eclipse.
- La lectura i càrrega de la clau privada i el certificat de la CA a l'aplicació.
- Un problema amb l'ordre del *Distinguished Name* a l'hora de crear certificats d'usuari, que feia que aquests no fossin reconeguts.

Finalment s'han pogut solucionar els problemes, malgrat que han provocat un endarreriment considerable.

Aquests problemes han fet que alguns processos es puguin millorar en el futur com per exemple:

- Els certificats creats es guarden en fitxers a l'espai de l'aplicació per poder enviar-los a l'usuari després. Seria millor que la transferència es fes directament o, en tot cas, que es guardessin a la base de dades.
- Les peticions de certificació només s'accepten en format DER. Es podria millorar fent que també s'acceptin fitxers en format PEM.

La metodologia que s'ha fet servir en aquest projecte ha estat fer una primera recerca per veure quines tecnologies s'haurien de fer servir, per després fer un esquema del disseny general i l'estructura que hauria de tenir l'aplicació.

A partir d'aquí s'ha provat d'anar implementant les funcions demanades i anar solucionant els problemes que anaven sorgint.

Finalment s'han realitzat nombroses proves de les funcions intentant cobrir diferents casos de la manera més diversa possible.

Com ja he esmentat abans, en començar aquest projecte desconeixia moltes de les tecnologies i part del programari que s'han fet servir, per tant la recerca d'informació ha estat continua durant tot el desenvolupament. Tot i que una gran part s'ha fet a la mateixa documentació dels productes utilitzats i altres publicacions, també ha estat de molta ajuda la consulta a pàgines web i fòrums de programadors a través d'internet.

1.4 – Planificació del projecte

Després d'una primera cerca d'informació les diferents funcions del producte, especificades a la secció d'objectius, s'implementaran amb les següents eines:

- Contenidor de servlets Apache Tomcat.
- Llenguatge de programació Java per a crear els servlets. Es faran servir diverses API com JDBC per les bases de dades, keytool i Bouncy Castle per les funcions criptogràfiques.
- Entorn integrat de programació Eclipse.
- Base de dades MySQL per guardar la informació dels certificats i la configuració.

Totes aquestes eines es faran servir sobre una distribució de Ubuntu Linux (v10.04). Durant tot el projecte es faran servir eines de codi obert i/o gratuïtes. A part de les esmentades s'ha fet servir:

- Paquet d'ofimàtica OpenOffice.org.
- Explorador de bases de dades MySQL Navigator.
- Capturador de pantalles Shutter.
- Editor d'imatges GIMP.
- Editor web KompoZer.
- Navegador web Mozilla Firefox.

La planificació temporal de les tasques a realitzar vindrà marcada pels períodes entre lliuraments de PAC:

Del 2 al 18 de març: PAC1

- Elaboració del Pla de Treball i l'Estat de l'art.

Del 19 de març a l'11 d'abril: PAC2

- Revisió del Pla de Treball.
- Cerca d'informació sobre els algorismes i generació de claus que farem servir, així com formats de fitxers PKCS#10, PKCS#12, CLR, ...
- Cerca d'informació sobre la instal·lació i posada en marxa de totes les eines que farem servir: Contenidor de servlets Apache Tomcat, Java i les API necessàries, OpenSSL, mySQL, ...
- Cerca d'informació sobre servlets: estructura i implementació.
- Especificació i disseny del producte.
- Primer lliurament parcial del producte: Inclourà les funcionalitats de posada en marxa del sistema (Generació de claus i certificat de la CA i configuració inicial del sistema) i capacitat de re-inicialitzar el sistema diversos cops, netejant les bases de dades contingudes.

Del 12 d'abril al 2 de maig: PAC3

- Revisió i millora del lliurament anterior.
- Segon lliurament parcial del producte: Funcionalitats públiques del producte. Creació de certificats dels clients.

Del 3 al 23 de maig: PAC4

- Revisió i millora del lliurament anterior.
- Tercer lliurament parcial del producte: Funcionalitats d'administració del producte. Es lliurarà el producte pràcticament enllestit.

- Formalització d'ampliacions i millores.

Del 24 de maig al 10 de juny: Memòria i Producte

- Revisió i millora del lliurament anterior.
- Elaboració de la memòria del TFC. Contindrà un resum de tota la feina feta durant el projecte.
- Elaboració de la documentació del producte.
- Lliurament de la memòria i del producte final amb la documentació específica de posada en marxa i del propi producte.

De l'11 al 16 de juny: Presentació

- Elaboració i lliurament de la presentació.

Del 20 al 27 de juny: Torn de preguntes

- Torn de preguntes del Tribunal d'Avaluació.

1.5 – Productes obtinguts

El producte obtingut en aquest TFC és una aplicació que fa les funcions d'una autoritat de certificació (CA). Aquesta CA té un seguit de serveis en línia accessibles mitjançant un navegador web.

A més del producte principal s'obtindrà un document de posada en marxa de l'aplicació, així com una memòria explicativa que contindrà informació sobre el disseny, el desenvolupament i el funcionament de l'aplicació, a més d'altres tasques que s'han dut a terme durant l'elaboració d'aquest TFC. També es crearà una presentació on es podrà veure el funcionament de l'aplicació i servirà de resum del projecte.

1.6 – Descripció de la resta de capítols de la memòria

- **Capítol 2:** Es fa una descripció general de l'aplicació, incloent la seva estructura, el disseny, els problemes afrontats durant la seva elaboració, les funcions principals que conté i els requisits de programari per a executar-la.
- **Capítol 3:** Explica el procés d'instal·lació, posada en marxa i reinicialització de l'aplicació.
- **Capítol 4:** Mostra amb detall les funcions públiques de l'aplicació.
- **Capítol 5:** Especifica el funcionament de les funcions d'administrador de l'aplicació.
- **Capítol 6:** Conclusions del TFC.

2 – Descripció general de l'aplicació

Aquest capítol descriu l'estructura interna del producte des del punt de vista del disseny i la implementació del mateix.

2.1 – Estructura de l'aplicació

El producte està dividit en diverses parts.

A l'hora d'iniciar l'aplicació es farà servir un *script* que generarà l'Autoritat de Certificació i els fitxers necessaris perquè Tomcat es connecti a través d'una connexió segura. Aquest *script* farà servir dos fitxers de configuració per a la CA i l'aplicació.

Una altra part està formada pels fitxers generats per a la CA, que es faran servir per l'aplicació a l'hora de crear i validar certificats. La part central d'aquesta CA està formada pel seu certificat, la seva clau privada i la llista de revocació de certificats.

Com ja s'ha dit, l'aplicació funcionarà des d'un contenidor de servletsTomcat. Les connexions al servidor es faran de forma segura mitjançant el protocol HTTPS. L'accés es podrà fer tant si l'usuari té un certificat expedit per la CA com si no. Però serà imprescindible tenir-lo si es vol entrar a la secció d'administradors, ja que la pròpia connexió del servidor demanarà un certificat vàlid.

Finalment la part central de l'aplicació estarà formada per les classes java i les pàgines JSP.

Aquesta part té tres seccions clarament diferenciades. A la part més externa tenim les pàgines JSP que serveixen com a interfície gràfica de l'aplicació. A la següent capa tenim una sèrie de Servlets que gestionaran les peticions fetes des de les pàgines JSP i en retornaran els resultats. Aquests Servlets es comunicaran amb el nucli de l'aplicació que realitzarà les operacions pròpiament dites.

El nucli està format per tres classes java: Caops, Dbops i LoadConfig, que realitzaran, respectivament, les operacions amb la CA, les operacions amb la base de dades i llegiran i donaran accés a la configuració de l'aplicació.

La següent figura mostra les parts principals de la part central del producte:

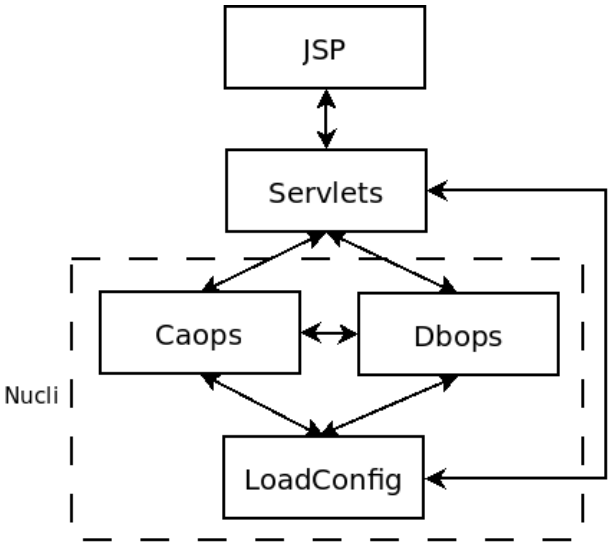

Figura 1: Parts principals de l'aplicació

En una ampliació d'aquest esquema, a la figura de la pàgina següent es pot veure un diagrama de classes amb la comunicació entre els diferents components de l'aplicació. Per fer més clar el diagrama s'ha evitat incloure els mètodes de cada classe, que es poden veure a la figura 3. També per donar claredat s'han eliminat les relacions amb la pàgina JSP sessionerror.jsp i la classe LinkForward.

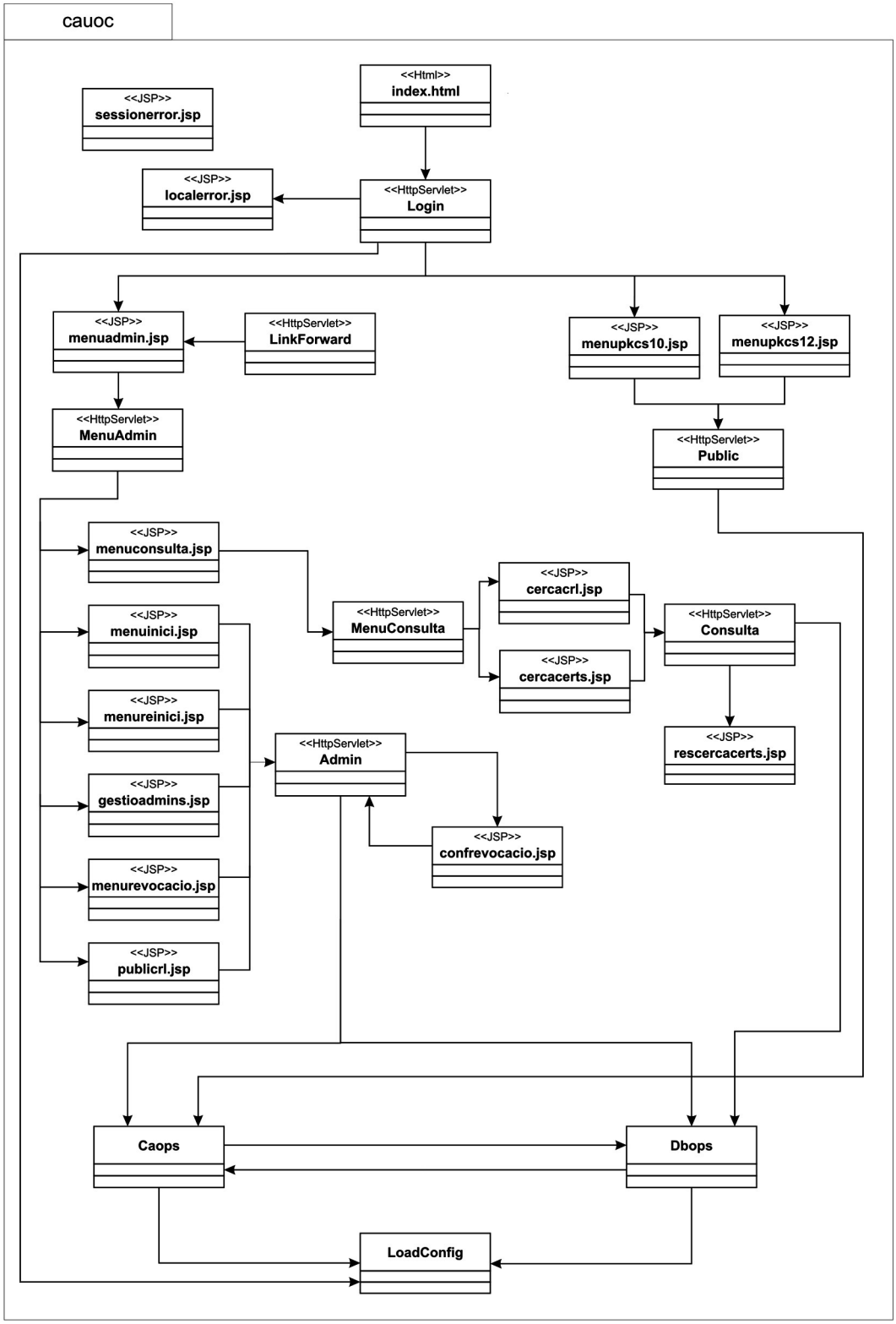

Figura 2: Diagrama de classes

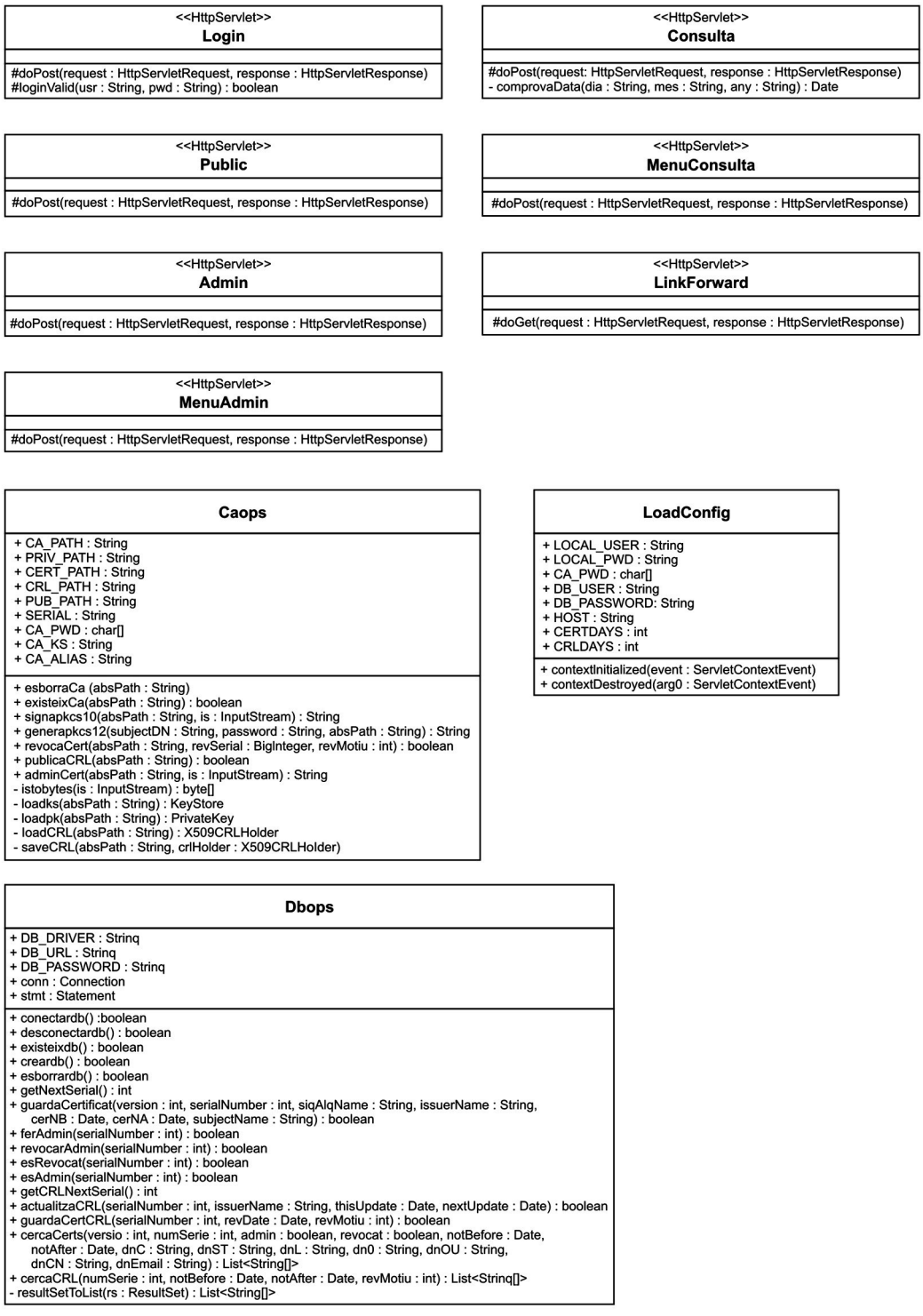

Figura 3: Mètodes i atributs de les classes

La pàgina sessionerror.jsp serà mostrada per qualsevol servlet de la secció d'administrador on es realitzarà una comprovació de si la sessió està autenticada i no ha caducat. Si falla aquesta comprovació, és mostra aquesta pàgina d'error.

La classe LinkForward s'encarrega de millorar la navegabilitat de la interfície, proporcionant un mètode per retornar al menú d'administrador des de qualsevol pàgina de la secció d'administració.

Al diagrama podem veure clarament com el flux d'execució comença amb la pàgina principal de la interfície i la classe Login, per distribuir-se entre la part pública i la part privada. Un cop es seleccionen les funcions disponibles a cada part, els servlets Public, Admin i Consulta s'encarreguen de cridar a les classes del nucli de l'aplicació (Caops, Dbops i LoadConfig) per executar les operacions i recollir els resultats per mostrar-los.

Aquesta separació permet per una banda que es pugui fer servir les classes del nucli a través d'una altre aplicació i de l'altra separar el disseny de la interfície del codi del programa, per una possible personalització d'aquesta.

Els servlets es comunicaran amb les pàgines JSP mitjançant el pas d'atributs amb els paràmetres de cada funció i controlant un atribut de sessió per comprovar que es té un accés permès a la part privada.

Les pàgines JSP implementen una àrea de notificacions (a la part superior, a sota la capçalera) que servirà per donar missatges d'error (en color vermell) o notificacions informatives (en color blau).

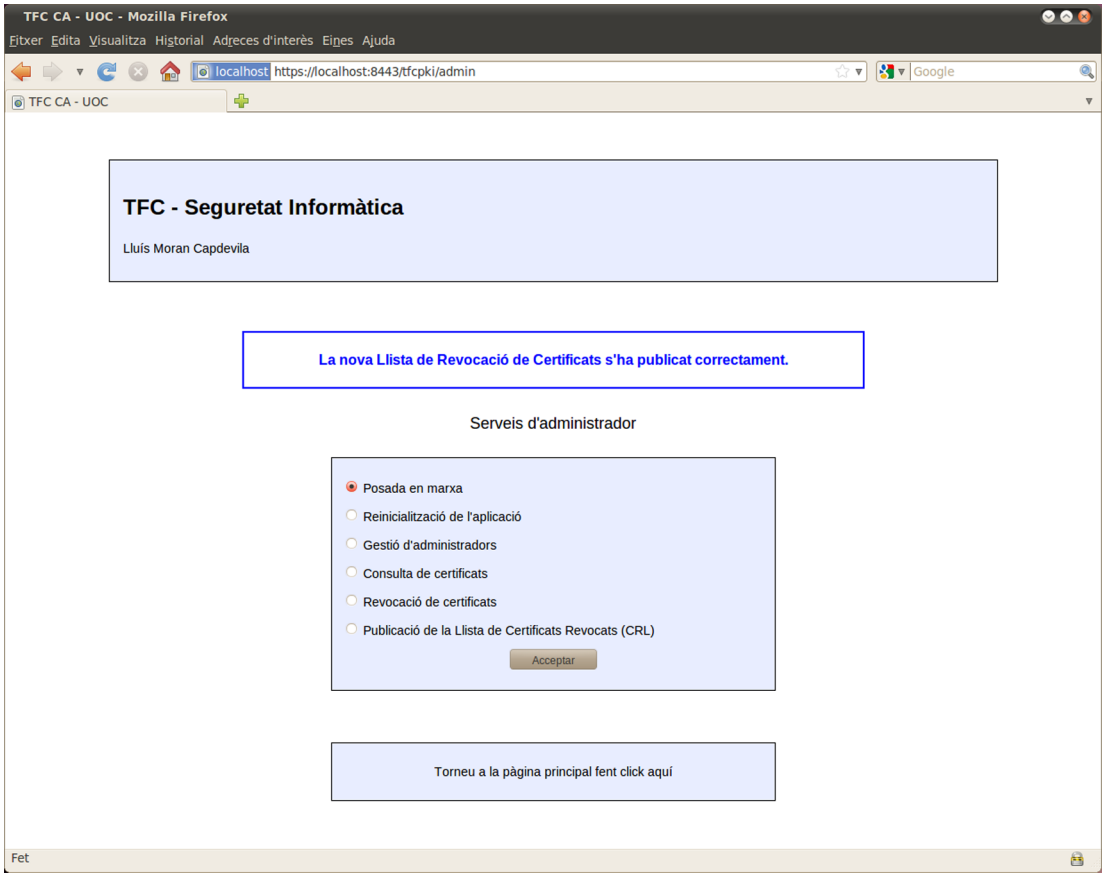

Figura 4: Interfície – Àrea de notificacions

Per altra banda l'aplicació farà servir una base de dades mySQL, l'estructura de la qual es veurà amb més detall a la secció següent.

2.1.1 – Estructura de la base de dades

Per emmagatzemar la informació de l'aplicació es farà servir una base de dades MySQL. Aquest Gestor de Bases de Dades funciona mitjançant la sintaxi del llenguatge estàndard SQL i és accessible directament des de java amb la seva API JDBC (*Java Database Connectivity*).

Durant el procés de posada en marxa es crearà un usuari per a que l'aplicació executi les ordres corresponents. També es generarà una base de dades anomenada “magatzem” que contindrà tres taules:

certificats: Taula on guardarem la informació dels certificats que crei la CA.

CRL: Taula on guardarem la informació de la llista de revocació.

CRLcerts: Taula on guardarem els certificats de la llista de revocació.

Aquestes són les comandes SQL de creació de les taules on es poden veure els tipus de dades, la mida de cada camp i una breu explicació del que s'hi guarda:

```
CREATE TABLE “certificats”
(Version integer(2),
SerialNumber integer(50) NOT NULL,
Admin bool DEFAULT False,
Rev bool DEFAULT False,
SigAlgName varchar(50),
IssuerName varchar(200),
NotBefore datetime,
NotAfter datetime,
SubjectName varchar(200),
PRIMARY KEY (SerialNumber)
);
```

certificats			
Nom del camp	***Tipus***	***Mida***	***Explicació***
Version	*integer*	*2*	*Versió del certificat*
SerialNumber	*integer*	*50*	*Número de sèrie del certificat*
Admin	*bool*		*El certificat té drets d'administrador*

Rev	*bool*		*El certificat està revocat*
SigAlgName	*varchar*	*50*	*Algorisme d'encriptació del certificat*
IssuerName	*varchar*	*200*	*Distinguished Name de l'emisor del certificat*
NotBefore	*datetime*		*Data abans de la qual el certificat no és vàlid*
NotAfter	*datetime*		*Data després de la qual el certificat no és vàlid*
SubjectName	*varchar*	*200*	*Distinguished Name del receptor del certificat*

CREATE TABLE “CRL”
(SerialNumber integer(50) NOT NULL,
IssuerName varchar(200),
ThisUpdate datetime,
NextUpdate datetime,
PRIMARY KEY (SerialNumber)
);

CRL			
Nom del camp	***Tipus***	***Mida***	***Explicació***
SerialNumber	*integer*	*50*	*Número de sèrie de la CRL*
IssuerName	*varchar*	*200*	*Distinguished Name de l'emisor de la CRL*
ThisUpdate	*datetime*		*Data de la última actualització de la CRL*
NextUpdate	*datetime*		*Data de la propera actualització de la CRL*

CREATE TABLE “CRLcerts”
(SerialNumber integer(50) NOT NULL,
RevocationDate datetime,
RevocationReason varchar(100),
PRIMARY KEY (SerialNumber)
);

CRLcerts			
Nom del camp	***Tipus***	***Mida***	***Explicació***
SerialNumber	*integer*	*50*	*Número de sèrie del certificat*
RevocationDate	*datetime*		*Data de la revocació del certificat*
RevocationReason	*varchar*	*100*	*Motiu de la revocació*

2.1.2 – Estructura de fitxers

L'estructura inicial de l'aplicació està continguda a la carpeta *tfcpki*. Aquesta és la carpeta que cal copiar a la carpeta *webapps* de Tomcat per instal·lar l'aplicació.

Dins d'aquesta s'inclouen dues carpetes, habituals en els projectes de Tomcat (META-INF i WEB-INF), una carpeta amb els fitxers de posada en marxa (bin), una amb les pàgines java server page (jsp), i una carpeta pub pels fitxers públics, a més dels següents fitxers:

- **index.html**: És la pàgina web d'inici de l'aplicació.
- **tfcpki.css**: És la pàgina d'estils que faran servir les pàgines web de l'aplicació.

La carpeta ***META-INF*** conté el fitxer de manifest i la configuració de context de l'aplicació.

La carpeta ***WEB-INF*** conté la carpeta ***classes*** on s'emmagatzema el *package* de l'aplicació java. Dins de l'estructura del *package* hi trobarem els fitxers java compilats i els fitxers de codi font. *WEB-INF* també inclou una carpeta ***lib*** on s'hi guarden les llibreries que fa servir l'aplicació. Finalment conté el fitxer ***web.xml*** per configurar l'execució de l'aplicació dins de Tomcat.

La carpeta ***bin*** guarda els fitxers de posada en marxa:

- **ca.cnf**: És el fitxer de configuració de la CA. Es fa servir des de l'*script* d'inicialització i conté paràmetres per a crear el certificat i la clau de la CA, així com per signar el certificat SSL de Tomcat.
- **CAuoc.sh**: És l'*script bash* d'inicialització que s'executa per crear la CA i tots els seus fitxers, el certificat SSL de Tomcat i per crear l'usuari de mySQL que farà servir l'aplicació. Aquest script crearà les carpetes *ca* i *tomcat* que contenen tots els fitxers de la CA i el certificat de Tomcat respectivament
- **user.conf**: És el fitxer de configuració d'usuari, on aquest pot canviar els paràmetres d'inicialització de l'aplicació.

La carpeta ***jsp*** emmagatzema els fitxers *Java Server Page* que composen la interfície web de l'aplicació.

La carpeta ***pub*** es farà servir per donar accés als fitxers públics de l'aplicació com ara la llista CRL i el certificat de la CA.

Dins la carpeta ***doc*** hi trobarem la documentació java de l'aplicació.

Un cop s'hagi realitzat el procés de posada en marxa, obtindrem dues noves carpetes: una que contindrà tot el referent a la CA de l'aplicació i una amb el magatzem de claus que conté el certificat de Tomcat per realitzar connexions segures.

L'*script* d'inicialització crearà l'estructura de carpetes i els fitxers necessaris pel funcionament de la CA. Tota l'estructura està continguda en una carpeta anomenada *ca.*

A continuació és detalla la jerarquia de carpetes. El contingut de la carpeta *ca* un cop s'ha executat l'*script* és la següent:

La carpeta *certificats* contindrà una còpia de tots els certificats que signi la CA. Com a nom es farà servir el número de sèrie del certificat. El certificat *00.pem* és el certificat que hem generat per a Tomcat.

La carpeta *crl* es farà servir per gestionar la llista de revocació de certificats. Conté la llista buida generada per l'*script* en format PEM i DER.

La carpeta *privat* conté la clau privada de la CA. Està guardada en format PEM i en format DER.

A l'arrel de la carpeta *ca* hi ha diversos fitxers. *ca.jks* conté el magatzem de claus de la CA que inclou el certificat auto-signat. Aquest certificat també el trobem com a fitxer a *cacert.pem.*

El fitxer *index.txt* guarda informació dels certificats creats. *index.txt.attr* té els seus atributs.

El fitxer *serial* conté el número de sèrie per assignar als certificats que signa la CA.

El fitxer *crlnumber* conté el número de sèrie per assignar a les noves versions de la llista de revocació de certificats.

Els fitxers *.old* són còpies de seguretat dels fitxers amb el mateix nom.

Alguns d'aquests fitxers no es faran servir dins de l'aplicació però es creen pel funcionament de l'*script* d'inicialització.

Finalment l'altra carpeta, anomenada *tomcat* contindrà el magatzem de claus de Tomcat (tomcat.jks) que inclou el certificat signat i el certificat arrel de la CA, i un altre magatzem de claus (trustedca.jks) per fer servir com a magatzem de confiança en les connexions SSL. A més hi haurà el certificat de Tomcat en format DER i en format PEM.

2.2 – Funcions

Tot i que en els capítols 4 i 5 d'aquesta memòria es mostra una descripció més a fons de les funcions de l'aplicació, aquest és un resum dels serveis que ofereix. Com ja s'ha esmentat, el producte es divideix en funcions públiques i privades:

Funcions públiques:

- Emissió d'un certificat a partir d'una petició PKCS#10: L'usuari haurà de generar un parell de claus i una petició de certificació en format DER. Aquest fitxer de petició es pujarà a l'aplicació que generarà i retornarà un certificat signat per la CA.
- Emissió d'un certificat i creació de claus en un objecte PKCS#12 a partir de les dades de l'usuari: L'aplicació presenta un formulari on s'hauran d'omplir les dades necessàries. En resposta se li retornarà un fitxer en format PKCS#12 que contindrà un certificat signat per la CA, el certificat arrel de la CA i les claus criptogràfiques generades corresponents.
- Descàrrega de la Llista de Revocació de Certificats (CRL): En aquesta opció es pot descarregar un fitxer CRL que conté l'última versió de la Llista de Revocació de Certificats per a introduir-la al navegador.
- Descàrrega del certificat arrel de l'Autoritat de Certificació: Ofereix la descàrrega del certificat arrel de la CA per introduir-la al navegador i fer que reconegui els certificats expedits per l'aplicació.

Funcions d'administrador:

- Posada en marxa de l'aplicació: Aquesta opció deixa l'aplicació llesta per a funcionar. Crea la base de dades que es farà servir i comprova que s'hagi creat els fitxers de la CA mitjançant l'ús de *l'Script* de posada en marxa.
- Reinicialització de l'aplicació: Elimina els fitxers de la CA i esborra la base de dades, deixant el sistema preparat per a tornar a crear-los mitjançant *l'Script* de posada en marxa.
- Gestió d'administradors: Permet proporcionar o revocar drets d'administrador a un certificat creat per la CA.
- Consulta de Certificats: Proporciona un sistema de cerca de certificats a partir de qualsevol dada d'aquests. Permet fer cerques sobre la llista de certificats emesos i sobre la llista de certificats revocats. Un cop feta la consulta mostra una taula amb les dades dels certificats obtinguts.
- Publicació de la Llista de Revocació de Certificats (CRL): Aquesta opció actualitza la llista de revocació i deixa el fitxer CRL preparat per a descarregar.

2.3 – Requisits de programari

És necessari que existeixi el següent programari instal·lat al servidor per poder dur a terme la posada en marxa i l'execució. Entre parèntesi es dóna la versió feta servir pel desenvolupament i una petita explicació de per què serveix el programari:

- Apache Tomcat 7 (7.0.11): És un contenidor de servlets. Proporciona l'entorn on s'executarà l'aplicació per a poder accedir als serveis en línia.
- Java SE 1.6 (1.6.0.24_b07): Llenguatge de programació orientat a objectes fet servir pel desenvolupament. Caldrà que estigui instal·lat per a fer servir la seva màquina virtual a l'hora de l'execució i per fer servir l'eina *keytool* inclosa, en el moment de la posada en marxa.
- OpenSSL (0.9.8k 25 Mar 2009): És una implementació dels protocols SSL i TLS que inclou una sèrie d'utilitats per treballar amb certificats. Es faran servir en la posada en marxa de l'aplicació.
- MySQL 5.1 (5.1.41-3ubuntu12.10): És un sistema de gestió de bases de dades que s'utilitzarà per emmagatzemar les dades de l'aplicació.

3 – Instal·lació de l'aplicació

En aquest capítol veurem tot el necessari per començar a fer servir l'aplicació.

Calen dos processos bàsics per què l'aplicació sigui plenament funcional: la instal·lació dins del contenidor de servlets i la posada en marxa.

3.1 – Instal·lació

El procés d'instal·lació de l'aplicació és molt simple.

Primer de tot caldrà tenir instal·lat i funcionant al servidor tot el programari especificat al punt 2.3. Per fer-ho caldrà consultar la documentació pròpia de cada programari.

Com ja s'ha vist al capítol 2.1.2, tota l'aplicació està continguda en una carpeta anomenada *tfcpki.* Un cop tinguem el servidor Tomcat funcionant només caldrà copiar aquesta carpeta, amb tots els fitxers que conté, dins de la carpeta *webapps* de la carpeta d'instal·lació del contenidor de servlets. Seguidament caldrà reiniciar el contenidor.

A continuació estarem preparats per passar a la següent fase i iniciar la posada en marxa.

3.2 – Posada en marxa

Una vegada tenim l'aplicació instal·lada haurem de dur a terme dos processos: la posada en marxa manual mitjançant un *script* i l'execució de la funció d'inicialització dins de la secció d'administrador de l'aplicació.

La posada en marxa es farà a través d'un *script* (CAuoc.sh) i d'un fitxer de configuració (user.conf) que l'usuari podrà modificar per personalitzar els paràmetres del programa. Tant l'*script* com el fitxer de configuració els trobarem a la carpeta *bin* de l'aplicació.

Aquesta posada en marxa preveu tres passes corresponents a les opcions de l'*script*:

sh CAuoc.sh -mysql

Crea un usuari de mySQL per a l'aplicació. No cal crear la base de dades o les taules que farem servir, ja que se n'encarrega l'aplicació amb el procediment d'inicialització.

sh CAuoc.sh -novaca

Crea l'estructura de carpetes i el certificat de la CA, la seva clau privada i un *keystore* que conté el certificat.

sh CAuoc.sh -tomcat

Crea un certificat per a Tomcat i el signa per la CA creada anteriorment (Cal haver executat l'opció -novaca abans de fer servir aquesta). Aquest certificat, guardat en un magatzem de claus, es farà servir per poder accedir a l'aplicació mitjançant una connexió segura. També crearà un magatzem de confiança per a identificar els certificats de la CA que vulguin entrar a l'aplicació com a administrador.

El fitxer de configuració **user.conf** és un fitxer de text que conté les opcions que faran servir tant l'*script* com l'aplicació. Per tant és important no modificar aquest fitxer un cop s'ha inicialitzat el sistema, ja que les dades que rebi l'aplicació seran incorrectes i caldrà reinicialitzar-la.

El format del fitxer és del tipus [clau] = [valor]. Cal respectar aquest format i no canviar els noms de les claus per evitar errors durant l'execució. Tanmateix cal que els valors no incloguin espais ni símbols especials.

Els valors per defecte del fitxer són els següents. Entre parèntesi hi ha una explicació de cada clau:

```
capassword = 8g90A3kpv4
(Contrasenya del magatzem de claus i la clau privada de la CA)
tomcatpassword = 3rt78jiK49
(Contrasenya del magatzem de claus i la clau privada del certificat de Tom-
cat)
mysqluser = dbadmin
(Nom de l'usuari de mySQL per a l'aplicació)
mysqlpassword = dik827dg3K
(Contrasenya de l'usuari de mySQL per a l'aplicació)
localuser = pkiadmin
(Nom de l'usuari per a connexions locals a l'aplicació)
localpassword = uoc
(Contrasenya de l'usuari per a connexions locals a l'aplicació)
host = localhost:8443/tfcpki
(Host de l'aplicació. El host cal que inclogui el port i la carpeta de l'a-
plicació, però no l'inici d'adreça “https://”)
certdays = 365
(Número de dies de validesa dels nous certificats)
crldays = 30
(Número de dies de validesa de la CRL)
```

En els següents apartats s'especifiquen les ordres per posar en marxa l'aplicació manualment.

Dins la resposta del *shell* està marcat en negreta on la intervenció de l'usuari és necessària.

Creació de l'usuari de mySQL per a l'aplicació

Per que l'aplicació pugui fer servir una base de dades mySQL, haurem de crear un usuari perquè s'hi connecti.

Ho farem amb la següent opció de l'*script,* executant-lo des d'una consola:

Comanda:
sh CAuoc.sh -mysql

Durant l'execució se'ns demanarà la contrasenya de *root* per tenir prou privilegis per crear l'usuari.

Creació dels certificats i claus de la CA

Per dur a terme la creació del certificat auto-signat i claus de la CA farem servir l'opció -novaca de l'*script* des d'una consola.

Durant l'execució es demanarà la intervenció de l'usuari per introduir les dades de la CA *(Distinguished Name).* Es poden acceptar les opcions per defecte entre corxets. També se'ns demanarà si volem fer el certificat confiable, opció que haurem de respondré afirmativament (*yes*).

Comanda:
sh CAuoc.sh -novaca

Resposta:
```
user@host:.../tfcpki/bin$ sh CAuoc.sh -novaca
Creant el certificat de la CA ...
Generating a 2048 bit RSA private key
................+++
.....................................................................
.............................+++
unable to write 'random state'
writing new private key to './ca/privat/./cakey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ES]:
State or Province Name (full name) [Barcelona]:
Locality Name (eg, city) [Barcelona]:
Organization Name (eg, company) [UOC]:
Organizational Unit Name (eg, section) []:ETIS
Common Name (eg, YOUR name) []:CA
Email Address []:lluismoran@uoc.edu
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
```

```
    MD5: 81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
    SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0  17 30 E8 4D 05 C7 FC 22 .5.F!.9..0.M..."
0010: 21 82 0F 42                                       !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Creació dels certificat de Tomcat i signatura per part de la nostra CA

Per augmentar la seguretat, farem que la nostra aplicació només accepti connexions segures (HTTPS). Per això necessitarem instal·lar al servidor Tomcat un certificat signat per la nostra CA.

Farem servir l'opció -tomcat de l'*script* des d'una consola.

Durant l'execució és demanarà la intervenció de l'usuari per confirmar algunes opcions. Caldrà picar *Enter* en la primera qüestió i respondre afirmativament a les altres quatre (*yes*)

Comanda:

sh CAuoc.sh -tomcat

Resposta:

```
user@host:.../tfcpki/bin$ sh CAuoc.sh -tomcat
Creant el certificat de Tomcat ...
Premeu Enter quan us demani si voleu la mateixa contrassenya...
Enter key password for <tomcat>
   (RETURN if same as keystore password):
```

```
Using configuration from ./ca.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 0 (0x0)
        Validity
            Not Before: May  1 18:54:38 2011 GMT
            Not After : Apr 30 18:54:38 2012 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Barcelona
            organizationName          = UOC
            organizationalUnitName    = ETIS
            commonName                = Tomcat
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                08:AD:6C:79:45:C7:22:A7:E7:0F:15:7F:B9:1E:78:24:EB:6E:F8:2A
            X509v3 Authority Key Identifier:
                keyid:95:35:DC:46:21:F0:39:E0:17:30:E8:4D:05:C7:FC:22:21:82:0
F:42

Certificate is to be certified until Apr 30 18:54:38 2012 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
unable to write 'random state'
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
    MD5:  81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
    SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
```

```
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
     MD5:  81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
     SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
     Signature algorithm name: SHA1withRSA
     Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
Certificate reply was installed in keystore
S'ha creat el fitxer tomcatks.jks i el trustedca.jks a la carpeta tomcat.
Copieu-los a la carpeta d'instal·lació de Tomcat 7 i
seguiu les instruccions de la documentació per modificar
els fitxers de configuració.
```

Un cop finalitzada l'execució haurem de copiar els fitxers *tomcat.jks* i *trustedca.jks*, que podem trobar a la nova carpeta *tomcat*, cap a la carpeta d'instal·lació de Tomcat7.

Per últim hem de modificar el fitxer de configuració de Tomcat perquè arrenqui en SSL i només accepti connexions d'aquest tipus. Farem servir la implementació JSSE que és part del Java Runtime.

Modifiquem el fitxer *server.xml* de la carpeta *conf* de Tomcat

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
        This connector uses the JSSE configuration, when using APR,
        the connector should be using the OpenSSL style configuration
        described in the APR documentation -->

    <Connector protocol="org.apache.coyote.http11.Http11Protocol"
        port="8443" SSLEnabled="true"
        maxThreads="150" scheme="https" secure="true"
        keystoreFile="tomcatks.jks" keystorePass="3rt78jiK49"
        truststoreFile="trustedca.jks" truststorePass="3rt78jiK49"
        clientAuth="want" sslProtocol="TLS" />


   i modifiquem la línia:
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="off" />
```

Hem de tenir en compte que si modifiquem la contrasenya de la clau de tomcat abans de crear-la, també l'haurem de modificar al fitxer *server.xml* (clau keystorePass=”lanovacontrasenya” i truststorePass=”lanovacontrasenya”). Les dues claus fan servir la mateixa contrasenya

Un cop finalitzat aquest procés ja estem llestos per accedir a l'aplicació. Abans però, haurem d'iniciar (o reiniciar si ja estava corrent) el servidor Tomcat perquè carregui l'aplicació amb la nova configuració.

Abans d'accedir a l'adreça del servidor, per tal que funcioni correctament, haurem d'importar el certificat de la CA al nostre navegador o acceptar l'excepció que es produeixi, ja que la nostra CA no és reconeguda per cap autoritat de certificació de les incloses per defecte.

Ara ja podem executar l'aplicació des del navegador en qualsevol d'aquestes adreces:

https://localhost:8443/tfcpki/

http://localhost:8080/tfcpki/

Si fem servir la segona adreça, utilitzant una connexió no segura, serem re-dirigits a la primera. Quan es faci servir l'aplicació a través d'una connexió remota, haurem de canviar “localhost” per l'adreça IP o l'adreça de domini del servidor.

Un cop accedim a la interfície de l'aplicació ens haurem d'autenticar com a administrador des d'una connexió local (des del mateix servidor) i executar l'opció de Posada en marxa.

Per fer login com a administrador hem de fer servir el següent nom d'usuari i contrasenya, sempre que no els hàgim modificat en el fitxer de configuració d'usuari:

Usuari: **pkiadmin**

Contrasenya: **uoc**

Un cop tinguem accés, caldrà escollir l'opció de **Posada en marxa**, per tal de crear la base de dades i poder començar a fer servir l'aplicació. Aquest procés està explicat amb més detall a l'apartat 5.2 d'aquesta memòria.

(Nota: Les respostes de la consola poden variar en cada execució)

3.3 – Reinicialització

El procés de reinicialització consisteix en eliminar tota la informació de l'aplicació deixant-la en el mateix estat en que estava al principi de la instal·lació. Per tant s'esborraran tots els fitxers de la CA i totes les taules de la base de dades junt amb el seu contingut. Aquest procés no té volta enrere, així que no es podrà recuperar cap informació anterior a la reinicialització.

Aquesta opció també consta de dues fases, una que es realitza des de dins de l'aplicació i una altra de manual.

Per iniciar-la cal ingressar com a administrador i escollir l'opció de Reinicialització del menú d'administrador. Un cop s'acaba el procés, es tancarà la sessió i haurem de repetir l'operació de crear la CA i el certificat de Tomcat mitjançant l'*script* de posada en marxa (CAuoc.sh). L'usuari de MySQL romandrà inalterat i per tant no cal tornar a crear-lo.

Així haurem de seguir les mateixes instruccions que hem vist a la posada en marxa del capítol anterior. Un cop executat l'*script* no cal tornar a fer el procés d'inicialització de dins l'aplicació, ja que les taules de la base de dades ja s'hauran creat en el procés anterior.

Finalment cal iniciar o reiniciar el servidor Tomcat de nou perquè l'aplicació torni a ser accessible en ple funcionament.

Dins del navegador caldrà tornar a importar el certificat arrel de la CA i acceptar el certificat de Tomcat, ja que s'hauran renovat en el procés.

4 – Funcions públiques de l'aplicació

Una part de les funcions de l'aplicació són públiques. Això vol dir que són accessibles a tots els usuaris sense cap mena d'autenticació.

Malgrat això, igualment es farà servir una connexió segura HTTPS per accedir-hi.

A continuació s'especifiquen les funcions públiques.

4.1 – Petició de certificats a partir d'un fitxer PKCS#10

A través d'aquesta opció, l'usuari podrà demanar que la CA expedeixi un certificat signat mitjançant un fitxer de petició de certificat PKCS#10.

Els fitxers de tipus PKCS#10, tal com s'especifica a la RFC2986, consisteixen en fer arribar a la CA la informació de la petició (el nom, la clau pública i una sèrie d'atributs) signada amb la clau privada de qui demana el certificat.

Aquesta informació de petició, junt amb un identificador d'algoritme de signatura i la signatura del subjecte són rebuts per la CA que n'extreu la informació, la valida i genera un certificat X.509 en resposta.

L'aplicació només accepta peticions en format DER, per tant si la petició no està en aquest format, serà refusada.

Així doncs per generar una petició de certificació l'usuari haurà d'obtenir un parell de claus i, amb aquestes, crear la petició. Fent servir OpenSSL es pot obtenir un fitxer de petició amb la següent comanda:

openssl req -newkey rsa:1024 -keyout [nomfitxerclau] -out [nomfitxerpeticio] -outform DER

Aquesta ordre generarà un fitxer de petició de certificat, després de demanar tota la informació necessària. Tot seguit podrem crear un objecte PKCS#12 per importat-lo a un navegador amb la següent comanda d'OpenSSL:

openssl pkcs12 -export -clcerts -in [nomfitxercertificat] -inkey [nomfitxerclau] -out [nomfitxerpkcs12] -name [aliasdelcertificat]

Un cop accedim a la funció des de la pàgina principal de l'aplicació, se'ns mostrarà la següent pantalla:

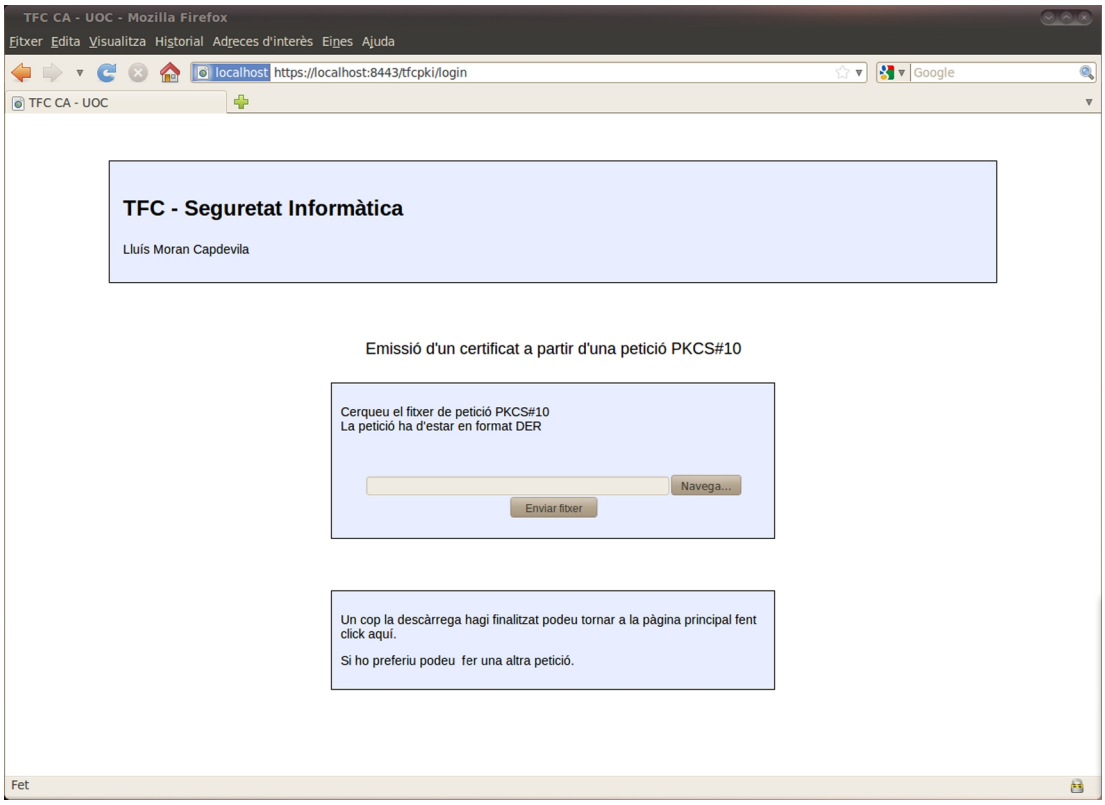

Figura 5: Interfície – Emissió d'un certificat a partir d'una petició PKCS#10

La interfície ens presenta un formulari per a cercar el fitxer de la petició de certificació PKCS#10. Abans d'introduir el fitxer se'ns remarca el fet que la petició ha d'estar en format DER pel correcte funcionament.

Un cop l'aplicació ha comprovat que el fitxer és correcte, retorna un fitxer per descarregar que conté un certificat X.509 signat per la CA.

Des del punt de vista intern, l'aplicació seguirà el següent procés (Entre parèntesi hi ha la classe que realitza la tasca):

- Càrrega del fitxer de petició de certificat (Public)
- Crida al mètode que gestiona la petició (Public)
- Verificació de la petició (Caops)
- Recol·lecció de les dades de la petició (Caops)
- Recol·lecció de les dades de la CA (Caops)
- Cerca del número de sèrie que farem servir pel certificat (Dbops)
- Lectura i format de les dates de validesa (LoadConfig i Caops)
- Creació de nou certificat i introducció de les extensions corresponents, tant de la CA com de la petició (Caops)
- Afegiment de l'adreça del punt de distribució de la CRL (LoadConfig i Caops)

- Signatura del certificat per la CA (Caops)
- Emmagatzematge de les dades del certificat a la base de dades (Dbops)
- Salvem el certificat en un fitxer (Caops)
- Retorn del fitxer a l'usuari o mostra d'errors si n'hi ha hagut algun (Public)

4.2 – Petició de certificats a partir de les dades d'usuari

Aquesta funció és semblant a la vista a l'apartat anterior, però en aquest cas, la informació de l'usuari es recull a través d'un formulari. Així l'aplicació generarà tot el necessari per a l'expedició d'un certificat a partir de la informació introduïda.

El formulari recollirà les dades corresponents al *Distinguished Name* del subjecte que fa la petició, així com una contrasenya per a protegir la clau privada que es crearà. Amb la informació obtinguda, l'aplicació generarà un parell de claus, pública i privada, per a l'usuari, crearà un certificat signat i ho encapsularà tot en un objecte PKCS#12, incloent-hi el certificat arrel de la CA.

Els fitxers PKCS#12 són un estàndard per emmagatzemar certificats X.509, junt amb la clau privada corresponent i tot protegit amb una contrasenya.

La contrasenya demanada per l'aplicació està limitada a set caràcters com a màxim. El motiu d'aquest fet és evitar que s'hagi d'instal·lar el *Java Criptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files* que limita la mida de les claus per motius de polítiques i controls d'exportació dels Estats Units.

Quan accedim a la funció des de la pàgina principal de l'aplicació se'ns presenta la següent pantalla:

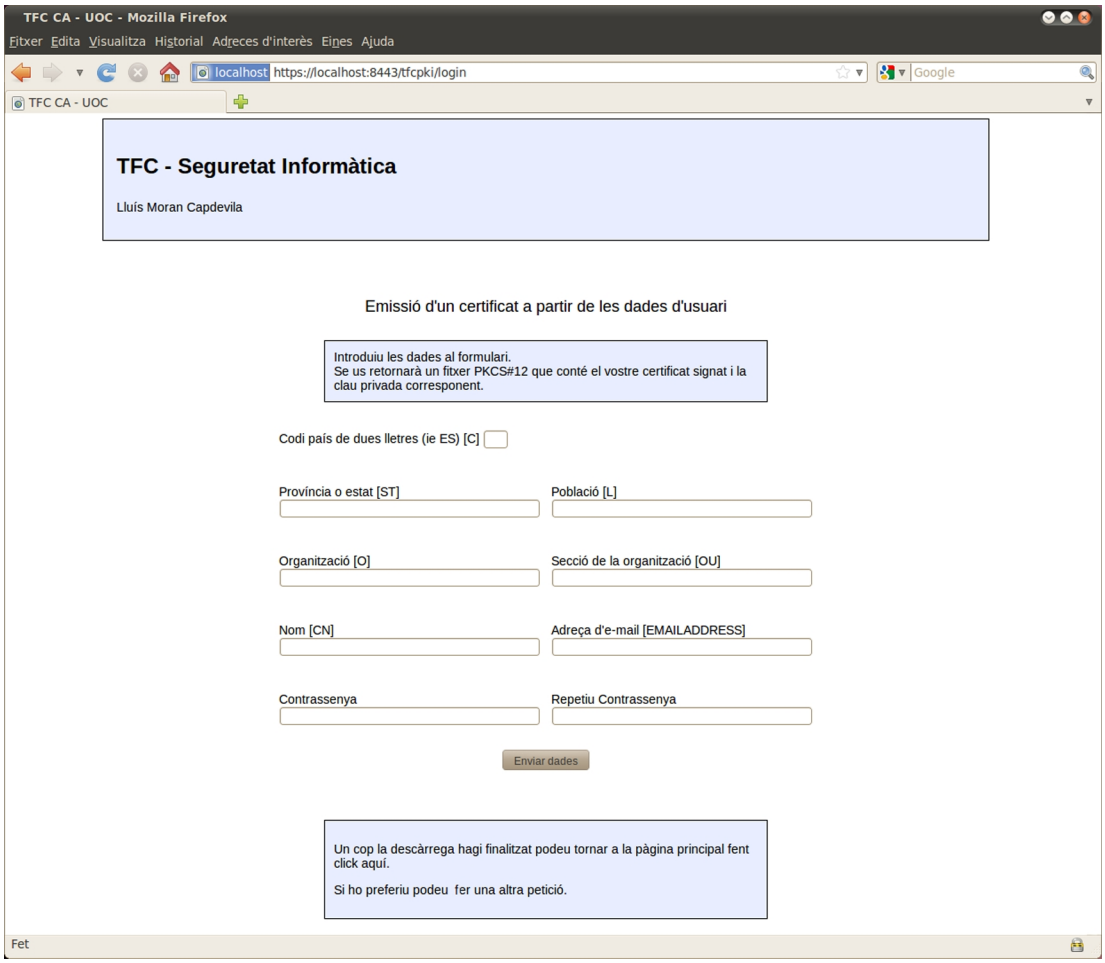

Figura 6: Interfície – Emissió d'un certificat a partir de les dades d'usuari

El formulari que obtenim conté els camps bàsics que formen el *Distinguished Name* del subjecte, a més de la contrasenya per a protegir el fitxer PKCS#12.

Aquests camps són el codi de país, la província o estat, la població, la organització, la secció de l'organització, el nom i l'adreça d'e-mail.

El formulari té algunes restriccions que són controlades per l'aplicació. Els camps Nom, Organització, Estat o província, país i contrasenyes no poden estar buits. El camp e-mail ha de tenir un format del tipus xxx@xxx.xxx. A més els camps de contrasenya tenen la limitació de 7 caràcters abans esmentada i, lògicament, han de tenir el mateix valor.

Quan s'envia les dades a l'aplicació es retorna per descarregar un fitxer PKCS#12 que conté el certificat arrel de la CA, el certificat signat per la CA i la clau privada corresponent.

L'aplicació seguirà el següent procés per a gestionar l'operació:

- Comprovació de la validesa de les dades del formulari (Public)

- Crida al mètode que gestiona la petició (Public)
- Generació del parell de claus de l'usuari (Caops)
- Generació d'una petició de certificat (Caops)
- Recol·lecció de les dades de la CA (Caops)
- Cerca del número de sèrie que farem servir pel certificat (Dbops)
- Lectura i format de les dates de validesa (LoadConfig i Caops)
- Creació de nou certificat i introducció de les extensions corresponents, tant de la CA com de la petició (Caops)
- Afegiment de l'adreça del punt de distribució de la CRL (LoadConfig i Caops)
- Signatura del certificat per la CA (Caops)
- Creació d'un magatzem de claus PKCS#12 on es guardarà la cadena de certificats i el parell de claus (Caops)
- Emmagatzematge de les dades del certificat a la base de dades (Dbops)
- Salvem el magatzem de claus PKCS#12 en un fitxer (Caops)
- Retorn del fitxer a l'usuari o mostra d'errors si n'hi ha hagut algun (Public)

4.3 – Descàrrega de la llista de revocació de certificats (CRL)

Aquesta funció consisteix simplement en un vincle públic al fitxer que conté la Llista de Revocació de Certificats (CRL). Aquesta llista serveix per a mantenir un control actualitzat dels certificats de la CA que han deixat de ser vàlids per diferents motius.

La llista s'actualitza automàticament cada cop que hi ha un canvi en la revocació de certificats i a més es pot publicar manualment per un usuari amb drets d'administrador. Aquesta publicació manual s'hauria de dur a terme en un espai de temps inferior al que ve marcat des del fitxer de configuració de l'aplicació i que per defecte són 30 dies. El motiu d'aquesta mena de caducitat de les llistes és que no hi ha cap més mecanisme per saber si un certificat ha estat invalidat i per tant podria ser utilitzat entre la data de la revocació i la publicació de la següent CRL.

Quan es descarrega la llista de revocació, alguns navegadors la importen directament al seu repositori. A continuació es veu un exemple de l'avís que mostra Mozilla Firefox en el moment de la descàrrega:

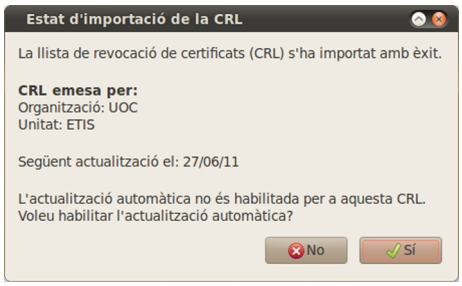

Figura 7: Importació de la CRL al navegador

4.4 – Descàrrega del certificat arrel de l'Autoritat de Certificació

L'última funció pública és la descàrrega del certificat arrel de la CA. Aquest certificat, importat convenientment al navegador com a certificat d'Autoritat, farà que es reconeguin els certificats emesos per la CA.

5 – Funcions d'administrador de l'aplicació

L'altre part de funcionalitats de l'aplicació consisteix en les operacions d'administrador. Per a fer-les servir serà imprescindible autenticar-se mitjançant un dels dos sistemes existents: a través d'una connexió local i amb un nom d'usuari i una contrasenya, o fent servir un certificat emès per la CA al qual, prèviament, se li hagin donat drets d'administrador. Un cop autenticat, l'aplicació farà un seguiment de la sessió per a comprovar que l'accés és correcte i que la sessió no expira (el temps d'expiració de la sessió és 10 minuts).

En el moment de la posada en marxa s'haurà de fer un ingrés local per poder fer el procés d'inicialització.

En les següents seccions s'explica amb detall les funcions d'administrador de l'aplicació.

5.1 – Ingrés a l'aplicació com a administrador

Com ja s'ha esmentat, per fer ús de les funcions privades s'haurà de procedir a autenticar-se. Aquest ingrés es realitza des de la pàgina principal, tal com es mostra a la següent figura:

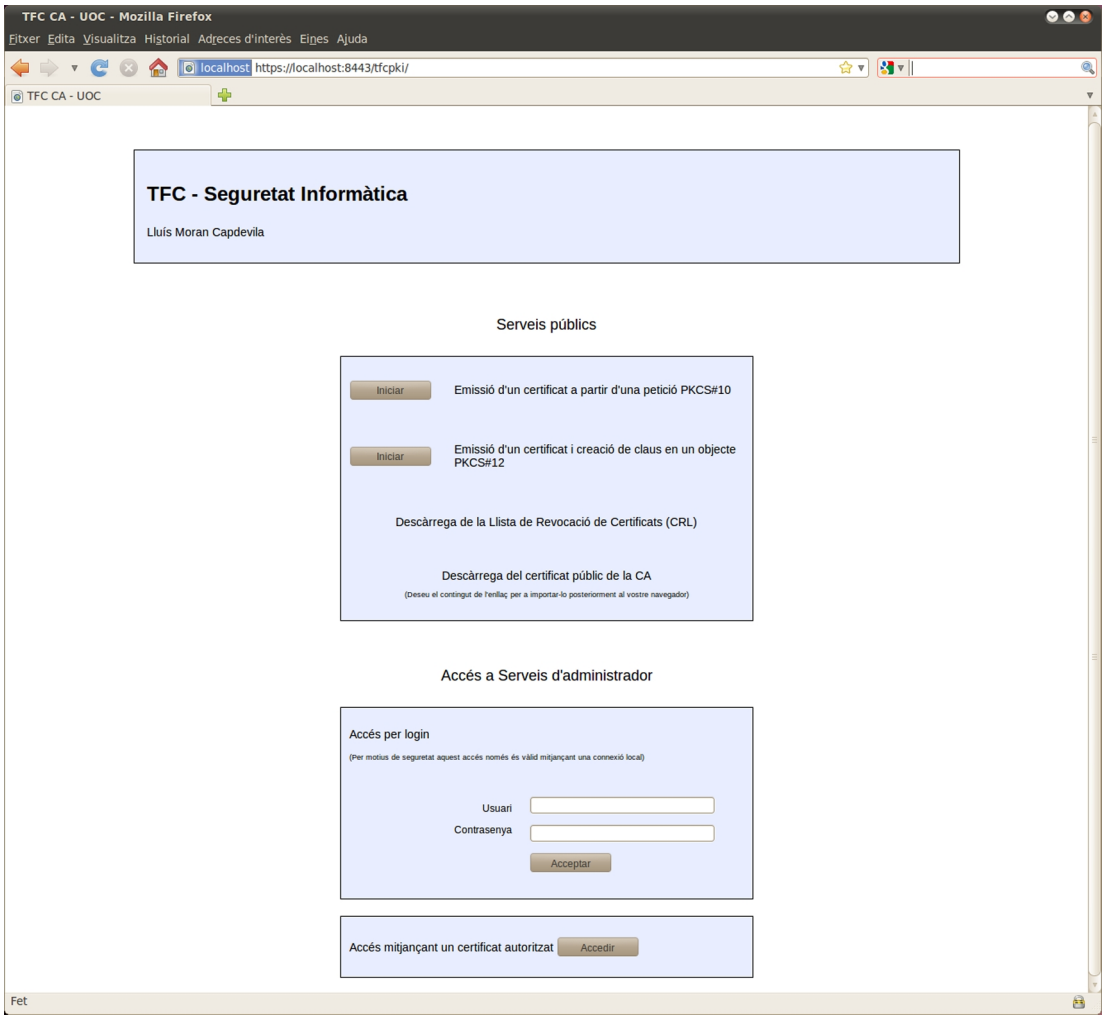

Figura 8: Interfície – Pàgina principal

Qualsevol dels dos tipus d'autenticació portarà al menú de l'administrador on es tindrà accés a totes les funcions protegides.

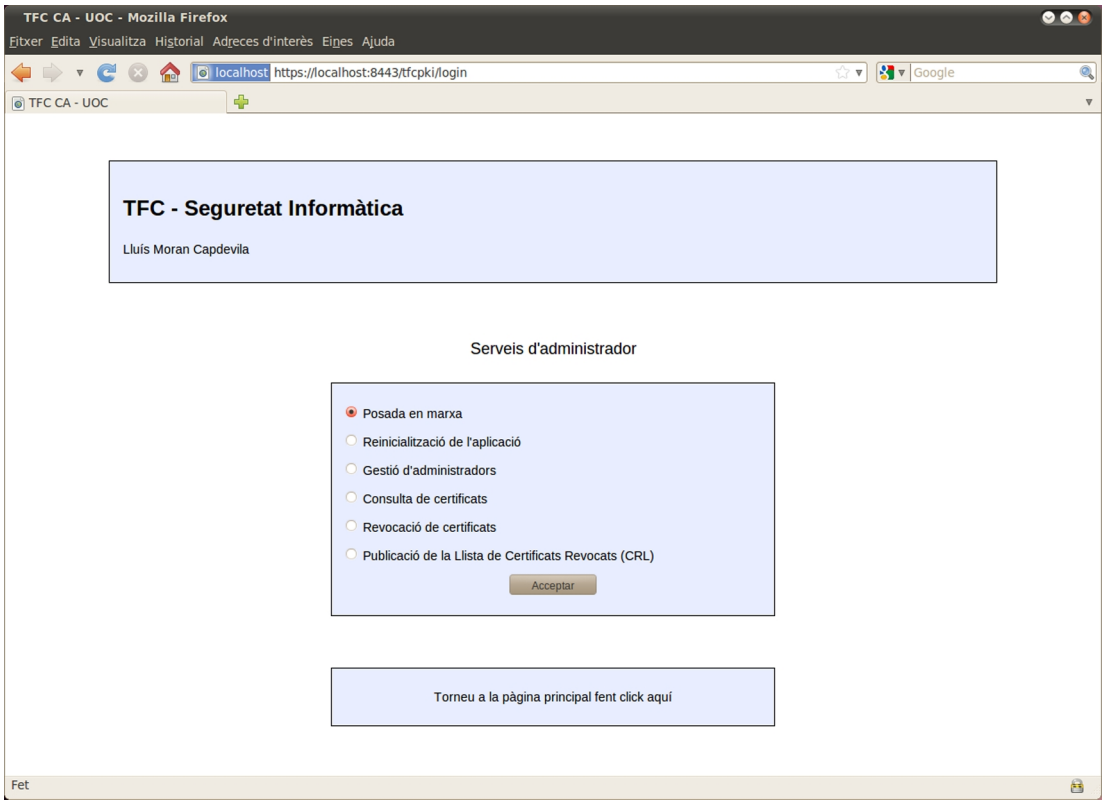

Figura 9: Interfície – Menú d'administrador

5.1.1 – Ingrés local mitjançant contrasenya

Aquest tipus d'ingrés a la zona d'administradors és d'autenticació feble. Per això només es podrà fer mitjançant una connexió local. Això s'aconsegueix comparant l'adreça IP de la connexió i comprovant que és la mateixa del host de l'aplicació.

El nom d'usuari i la contrasenya es poden canviar mitjançant l'edició del fitxer de configuració d'usuari (user.conf) que podem trobar a la carpeta *bin* de l'aplicació.

Un cop introduïm l'usuari i contrasenya, l'aplicació comprovarà si són correctes i ens donarà accés a la zona d'administradors. En cas que no siguin correctes, o que la connexió no sigui local, ens mostrarà un missatge d'error.

El funcionament intern del procés és el següent:

- Comprovació de que l'accés sigui local (Login)
- Recollida del nom d'usuari i la contrasenya del formulari (Login)
- Càrrega del nom d'usuari i la contrasenya de la configuració de l'aplicació (LoadConfig i Login)
- Comprovació de la coincidència de les dades (Login)

- En cas que les dades siguin vàlides, activació de la sessió d'administrador i crida al menú de funcions d'administrador (Login)
- En cas de no coincidència o de que l'accés no sigui local, mostra d'error (Login)

5.1.2 – Ingrés mitjançant certificat autoritzat

Per a accedir amb un certificat autoritzat haurem de tenir un certificat signat per la CA al qual, prèviament, se li hagin donat drets d'administrador. Aquests drets hauran de ser atorgats des de la secció de Gestió d'administradors, que es veurà més a fons al capítol 5.5.

En quant entrem a l'aplicació es comprovarà si tenim un certificat signat per la CA i se'ns demanarà que ens identifiquem amb ell. Tant si es té un certificat com si no (o si es cancel·la la identificació) es procedirà a la pàgina principal del programa.

A la part inferior d'aquesta pàgina tenim un botó per autenticar-nos dins de l'aplicació. L'aplicació comprovarà que el certificat sigui correcte, sigui emès per la CA, no estigui caducat o revocat i que tingui drets d'administrador. Si falla la comprovació d'algun d'aquests fets, mostrarà un error indicant quin és el motiu de refusar l'ingrés.

Si l'autenticació és correcta es donarà accés a la secció d'administradors.

Internament, el procés serà aquest:

- Càrrega del certificat de l'usuari (Login)
- Crida al mètode que comprova que el certificat sigui d'administrador (Login)
- Recol·lecció de les dades de la CA (Caops)
- Verificació de que el certificat pertany a la CA (Caops)
- Validació de la caducitat del certificat (Caops)
- Comprovació de si el certificat està revocat (Dbops)
- Comprovació de si el certificat té drets d'administrador (Dbops)
- En cas que el certificat sigui vàlid, activació de la sessió d'administrador i crida al menú de funcions d'administrador (Login)
- En cas d'invalidesa, mostra d'error per pantalla (Login)

5.2 – Posada en marxa

El procés de posada en marxa que ja hem vist al capítol 3.2, inclou una part que s'ha de dur a terme des de les funcions d'administrador.

Aquesta part del procés inclou la creació de les taules de la base de dades, tal com hem vist al capítol 2.1.1, i la comprovació de l'existència dels fitxers necessaris per a que funcioni la CA.

Un cop hem seleccionat l'opció al menú d'administrador, se'ns demanarà una confirmació per assegurar que el procés no es dugui a terme per error.

En acceptar la confirmació, l'aplicació procedirà a crear les taules de la base de dades i a comprovar que la CA ja està creada. Aquesta comprovació es fa confirmant que existeix el magatzem de claus de la CA i el fitxer de la seva clau privada.

Si no es passa alguna comprovació es mostrarà els errors corresponents a la zona de notificacions de la pàgina. Tanmateix si el procés es desenvolupa correctament, apareixerà una notificació indicant-ho a la pantalla.

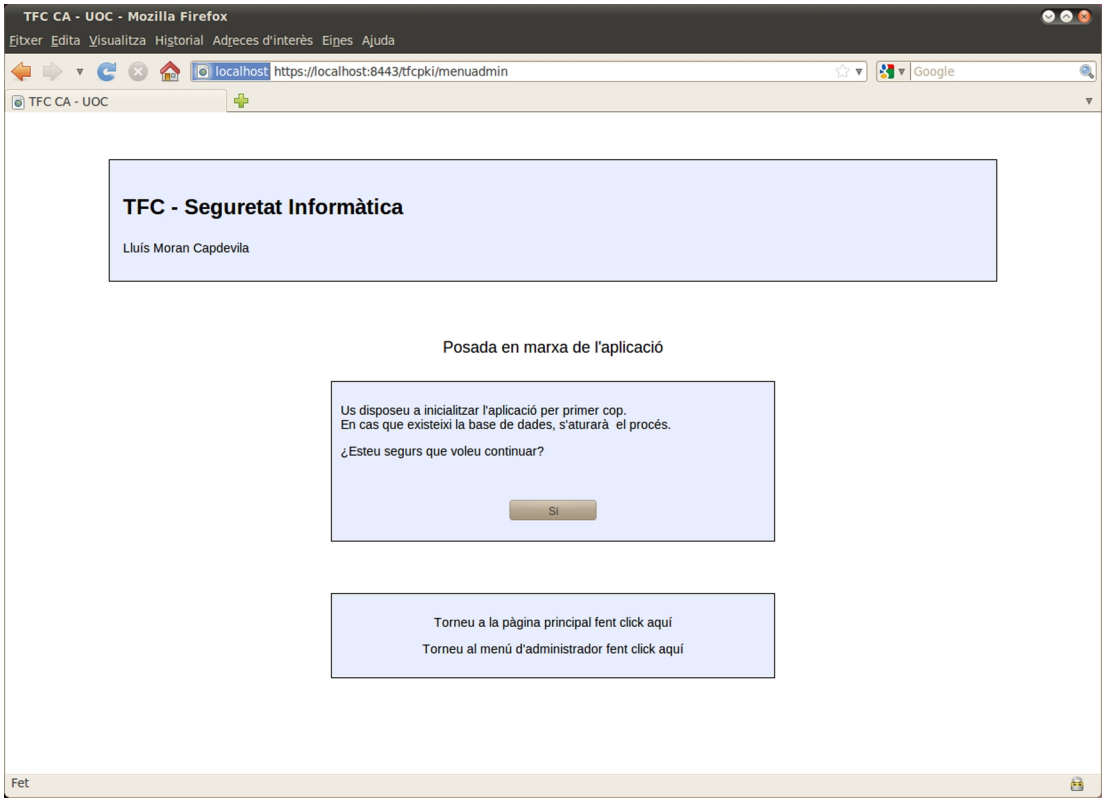

Figura 10: Interfície – Posada en marxa de l'aplicació

L'aplicació seguirà el següent procés per fer la inicialització de l'aplicació:

- Crida al mètode de creació de la base de dades (Admin)
- Connexió a la base de dades i comprovació de l'existència de les taules (Dbops)
- Creació de les taules de la base de dades (Dbops)
- Crida al mètode que comprova l'existència de la CA (Admin)
- Comprovació de si existeix el magatzem de claus i la clau privada de la CA (Caops)
- Mostra de missatges d'èxit o error en el procés (Admin)

5.3 – Reinicialització

L'opció de reinicialització és similar a la d'iniciar l'aplicació. L'objectiu d'aquest procés és deixar el producte com en el moment de la instal·lació, esborrant tota la informació emmagatzemada fins el moment.

Aquest procés complert ja s'ha vist al capítol 3.3, per tant aquí només s'especificarà la part que es porta a terme des de la zona d'administradors de l'aplicació.

Igual que en la opció anterior, un cop encetem el procés al menú d'administradors, es mostra un avís de confirmació que ens remarca que tota la informació es perdrà si seguim endavant.

Quan l'acceptem, l'aplicació procedeix a esborrar i tornar a crear les taules de la base de dades i a eliminar totes els fitxers de la CA.

Seguidament es tanca la sessió de l'aplicació i s'indica a l'usuari que ha de procedir a crear de nou la CA mitjançant l'*Script* de posada en marxa.

Un cop enllestit aquest procés no és necessari tornar a inicialitzar l'aplicació amb el procediment del capítol 5.2.

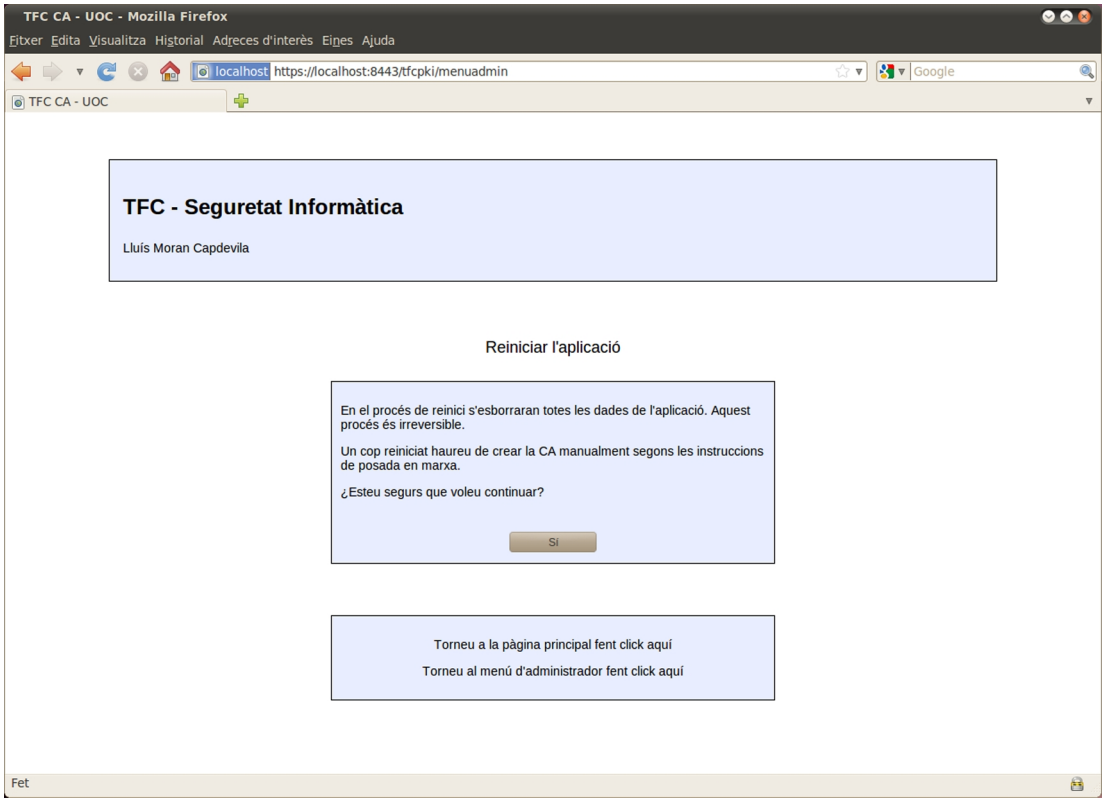

Figura 11: Interfície – Reiniciar l'aplicació

El funcionament intern de la funció és com segueix:

- Comprovació de l'existència de la base de dades i esborrat de la mateixa (Dbops)
- Crida al mètode de creació de la base de dades (Admin)
- Connexió a la base de dades i comprovació de l'existència de les taules (Dbops)
- Eliminació de les taules de la base de dades (Dbops)
- Crida al mètode que comprova l'existència de la CA (Admin)
- Comprovació de si existeix el magatzem de claus i la clau privada de la CA (Caops)
- Eliminació de tots els fitxers de la CA (Caops)
- Mostra de missatges d'èxit o error en el procés, i avís de la necessitat de tornar a crear la CA (Admin)
- Tancament de la sessió d'administrador (Admin)

5.4 – Consulta de dades dels certificats

La consulta de dades dels certificats consisteix a poder fer cerques sobre els certificats a partir de qualsevol dels paràmetres emmagatzemats a la base de dades de l'aplicació.

Es contemplen dos tipus de cerca:

- Cerca de certificats emesos per la CA: Es podrà cercar d'entre tots els certificats que han estat signats per la CA de l'aplicació.
- Cerca de certificats revocats: Permet fer una cerca sobre els certificats que han estat revocats per algun motiu.

Quan accedim a l'opció de Consulta de certificats, se'ns mostrarà un altre menú on escollirem el tipus de cerca que volem fer d'entre els dos esmentats. A la següent figura podem veure aquest menú en pantalla:

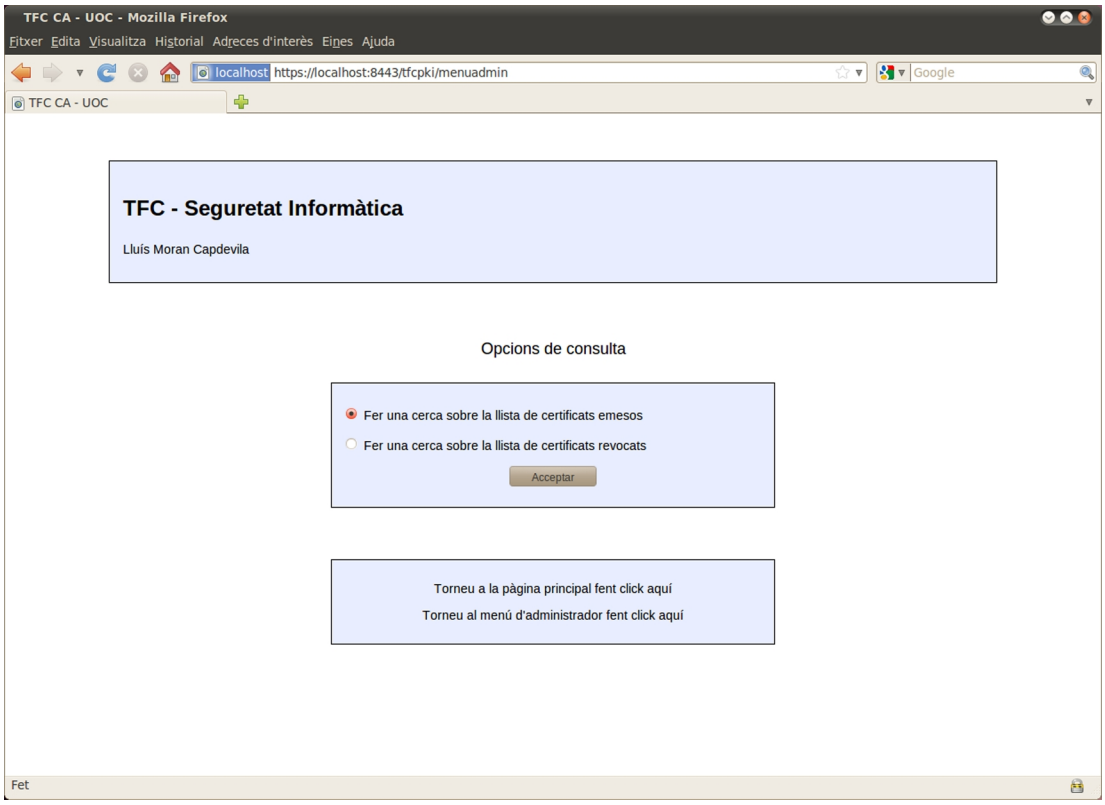

Figura 12: Interfície – Opcions de Consulta

5.4.1 – Cerca de certificats emesos per la CA

L'opció de cerca de certificats ens durà a un formulari on podem escollir els paràmetres de cerca.

Els paràmetres que es deixin en blanc proporcionaran el total de resultats d'aquest paràmetre. Així doncs si, per exemple, deixem la versió de certificat en blanc, l'aplicació ens mostrarà qualsevol certificat, sigui quina sigui la seva versió. De la mateixa manera, si no omplim cap dada del formulari i premem el botó de cerca, se'ns retornarà una llista de tots els certificats emesos per la CA fins al moment.

A la següent figura es mostra el formulari de cerca de certificats:

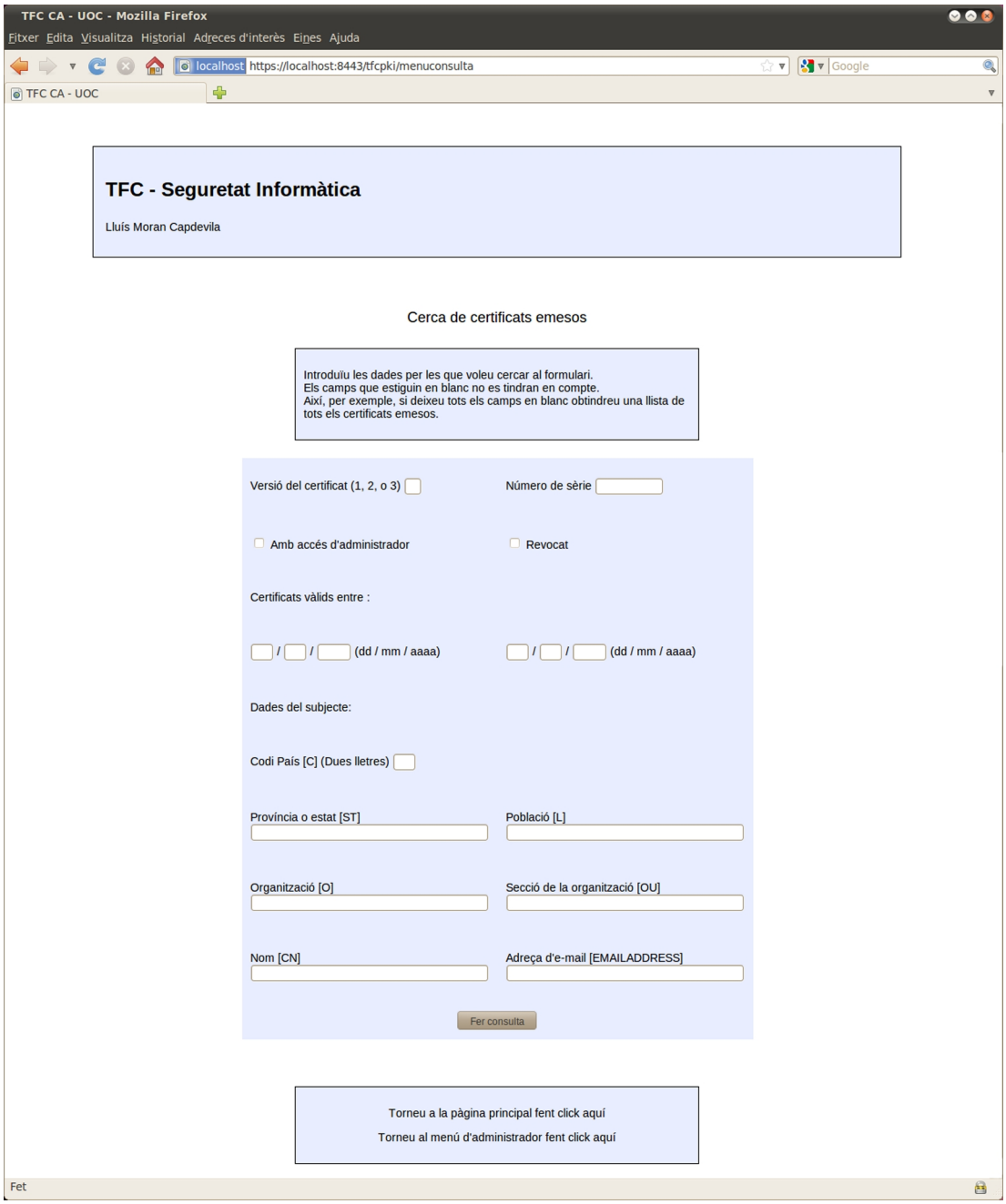

Figura 13: Interfície – Cerca de certificats emesos

Els camps pels quals podem realitzar cerques són els següents:

- Versió de certificat: Versió de certificat X.509. Pot tenir valor 1, 2 o 3.
- Número de sèrie: Número de sèrie del certificat. Aquest número és únic, per tant el resultat de la cerca només pot ser un sol certificat o cap si el certificat no existeix.
- Accés d'Administrador: Si es marca l'opció només mostrarà els certificats que tinguin atorgat dret d'administrador.

- Revocat: Si es marca l'opció només mostrarà els certificats que hagin estat revocats.
- Certificats vàlids entre dues dates: Es mostraran els certificats que siguin vàlids entre aquestes dues dates. Si es deixa qualsevol data en blanc la cerca tindrà en compte que no hi ha límit en el sentit d'aquella data. Les dates han de ser complertes o estar en blanc, o sigui no es pot posar només l'any o el mes. La cerca tindrà en compte les dates des de les 00:00 hores de la primera data fins a les 23:59 de la segona data, per tant les dues dates estan incloses.
- Dades del subjecte: Dades del receptor del certificat corresponents al *Distinguished Name.*
  - Codi País: Haurà de ser de dues lletres (i.e.: IR per a Irlanda).
  - Província o estat
  - Població
  - Organització
  - Secció de la organització
  - Nom
  - Adreça d'e-mail

Un cop la cerca es dugui a terme es mostrarà una pantalla amb els resultats de la consulta. Aquesta pantalla contindrà una taula amb tota la informació emmagatzemada dels certificats obtinguts. A continuació podem veure un exemple de resultat de cerca:

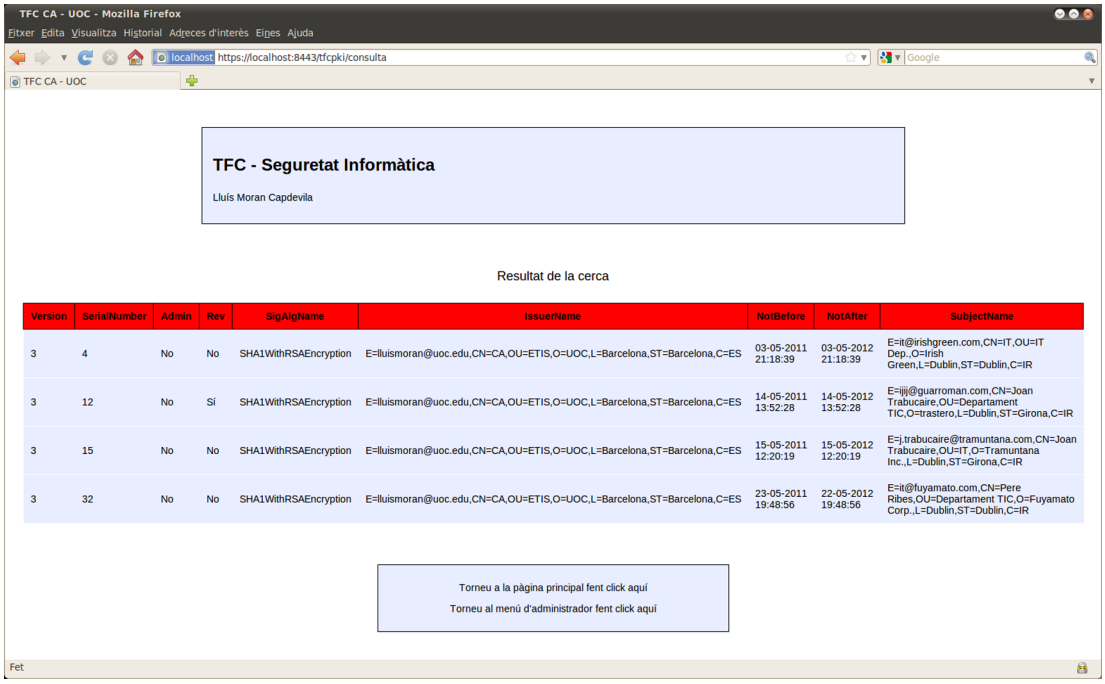

Figura 14: Interfície – Resultat de la cerca de certificats emesos

A nivell intern el procés de cerca serà com segueix:

- Recollida i validació de format dels paràmetres de la cerca (Consulta)
- Crida al mètode que fa les cerques de certificats (Consulta)
- Generació de la comanda SQL que farà la cerca a la base de dades a partir dels paràmetres introduïts (Dbops)
- Retorn del resultat de la cerca (Dbops)
- En cas que hi hagi algun error es mostrarà a la zona de notificacions (Consulta)
- Crida a la pàgina JSP que mostrarà els resultats de la cerca si no hi ha hagut errors (Consulta)

5.4.2 – Cerca de certificats revocats

El funcionament de la cerca de certificats revocats és idèntic al de la consulta de certificats. Les úniques diferències són el canvi de paràmetres de cerca i que es durà a terme sobre els certificats revocats. Igualment les dades en blanc que deixem al formulari serviran per a mostrar tots els certificats sense importar aquella dada.

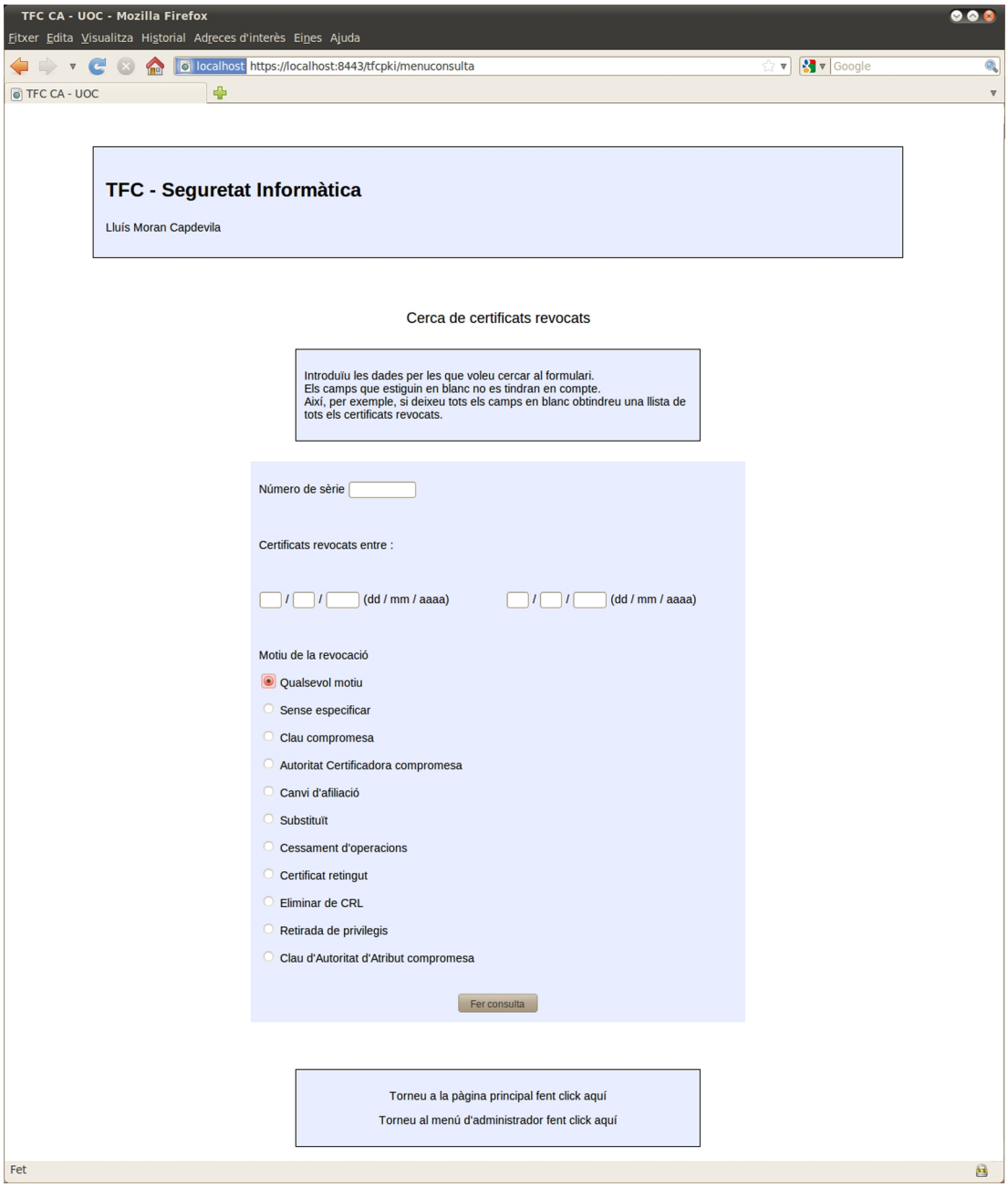

Figura 15: Interfície – Cerca de certificats revocats

Els camps de cerca seran els següents:

- Número de sèrie: Número de sèrie del certificat revocat.
- Certificats revocats entre dues dates: Mostrarà tots els certificats que tinguin la data de revocació entre les dues dates donades. Si es deixa qualsevol data en blanc la cerca tindrà en compte que no hi ha límit en el sentit d'aquella data. Les dates han de ser complertes o estar en blanc, o sigui no es pot posar només l'any o el mes. La cerca tindrà en compte les dates des de les 00:00 hores de la

primera data fins a les 23:59 de la segona data, per tant les dues dates estan incloses.

- Motiu de revocació: El motiu pel qual ha estat revocat el certificat. En cas que vulguem veure tots els motius de revocació cal seleccionar “Qualsevol motiu”.

El resultat de la cerca, anàlogament al cas anterior, mostrarà una taula amb tota la informació dels certificats revocats. Ho podem veure a la següent figura:

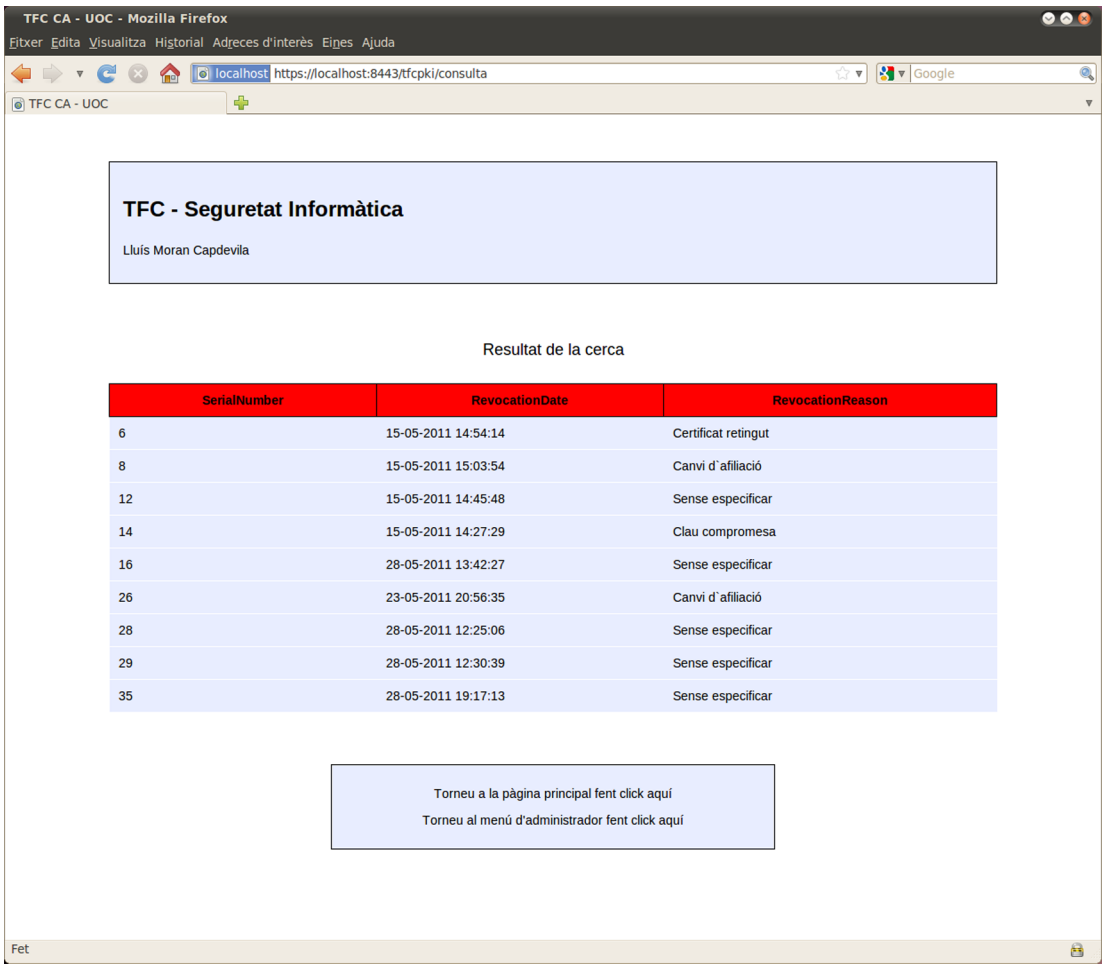

Figura 16: Interfície – Resultat de la cerca de certificats revocats

La funció tindrà les següents fases:

- Recollida i validació de format dels paràmetres de la cerca (Consulta)
- Crida al mètode que fa les cerques de la llista CRL (Consulta)
- Generació de la comanda SQL que farà la cerca a la base de dades a partir dels paràmetres introduïts (Dbops)
- Retorn del resultat de la cerca (Dbops)
- En cas que hi hagi algun error es mostrarà a la zona de notificacions (Consulta)

► Crida a la pàgina JSP que mostrarà els resultats de la cerca si no hi ha hagut errors (Consulta)

5.5 – Gestió d'administradors

L'apartat de gestió d'administradors serveix per atorgar o revocar accés d'administrador a un certificat en concret. Aquest canvi en el nivell d'accés es fa a partir del número de sèrie.

L'aplicació guarda a la base de dades el tipus d'accés de cada certificat (sense accés d'administrador per defecte) de manera que quan s'intenta fer un ingrés a l'aplicació es pugui comprovar i decidir si es deixa entrar a l'usuari o no.

Com es pot veure a la figura següent, la interfície d'aquesta operació és molt simple. Només es demana un número de sèrie i si volem donar o treure l'accés d'administrador a aquest certificat.

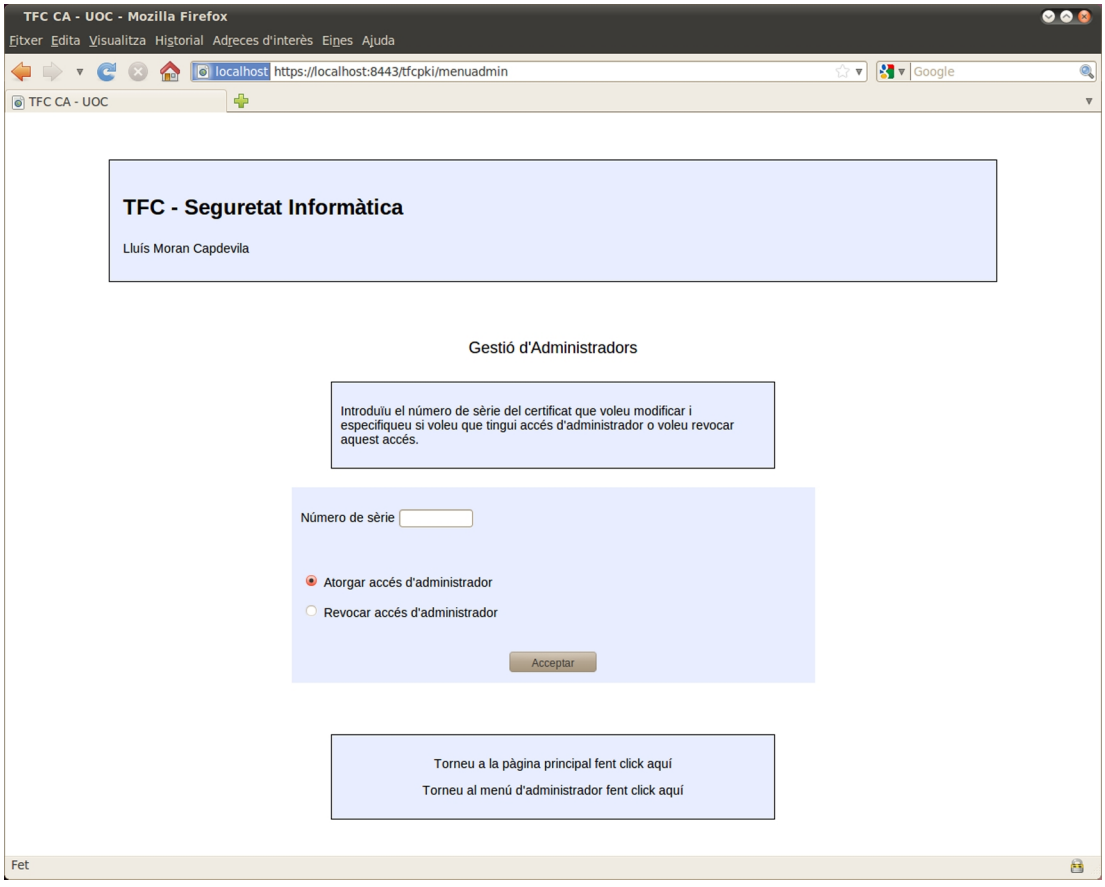

Figura 17: Interfície – Gestió d'Administradors

Un cop acceptat es mostra un missatge a l'àrea de notificació informant del resultat de la operació.

L'aplicació procedirà internament seguint el següent procés:

- Càrrega dels paràmetres de la funció (Admin)
- Comprovació de l'existència del certificat fent una cerca a la base de dades (Admin i Dbops)
- Comprovació de si el certificat està revocat (Dbops)
- Donar drets d'administrador o revocar-los segons sigui el cas (Dbops)
- Mostra de missatges d'èxit o error en el procés (Admin)

5.6 – Revocació de certificats

Un certificat pot deixar de ser vàlid per diverses raons, a més de per haver caducat, com per exemple que la seva clau privada s'hagi vist compromesa o hagi estat substituït per un certificat nou.

L'eina per notificar a tots els usuaris que un certificat ha deixat de ser vàlid és la Llista de Revocació de Certificats (CRL) que és publicada per la CA de manera periòdica.

El procés de revocació es fa a partir del número de sèrie del certificat. L'administrador haurà d'introduir el número al formulari i marcar un motiu de revocació. També té l'opció de deixar el motiu sense especificar, però no es una bona pràctica segons la norma RFC5280.

Els possibles motius de revocació són:

- Clau compromesa
- Autoritat Certificadora compromesa
- Canvi d'afiliació
- Substituït
- Cessament d'operacions
- Certificat retingut
- Eliminar de la CRL
- Retirada de privilegis
- Clau d'Autoritat d'Atribut compromesa

L'aplicació té en compte el cas especial de "Certificat Retingut" i "Eliminar de la CRL". La retenció d'un certificat provoca una revocació temporal, durant la qual el certificat esdevé invàlid a tots els efectes. En qualsevol moment un administrador pot enviar l'ordre de "Eliminar de la CRL" fent que s'esborri de la llista de revocació de certificats de manera que sigui vàlid novament.

La següent figura mostra el formulari de revocació de certificats:

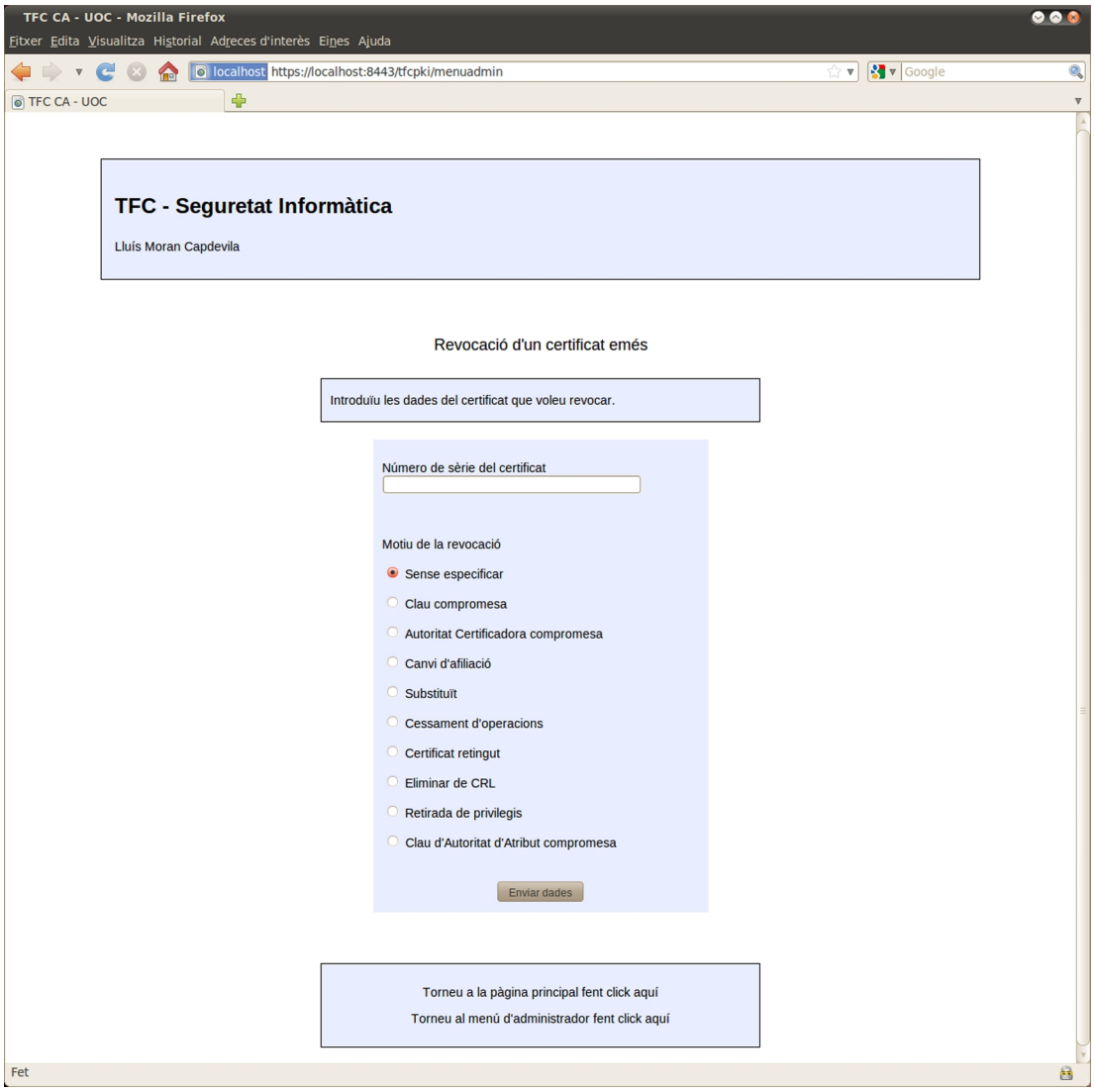

Figura 18: Interfície – Revocació d'un certificat emés

Un cop s'envia el formulari, l'aplicació mostrarà una pantalla de confirmació on podrem veure les dades del certificat que volem revocar. Si acceptem, la revocació es durà a terme.

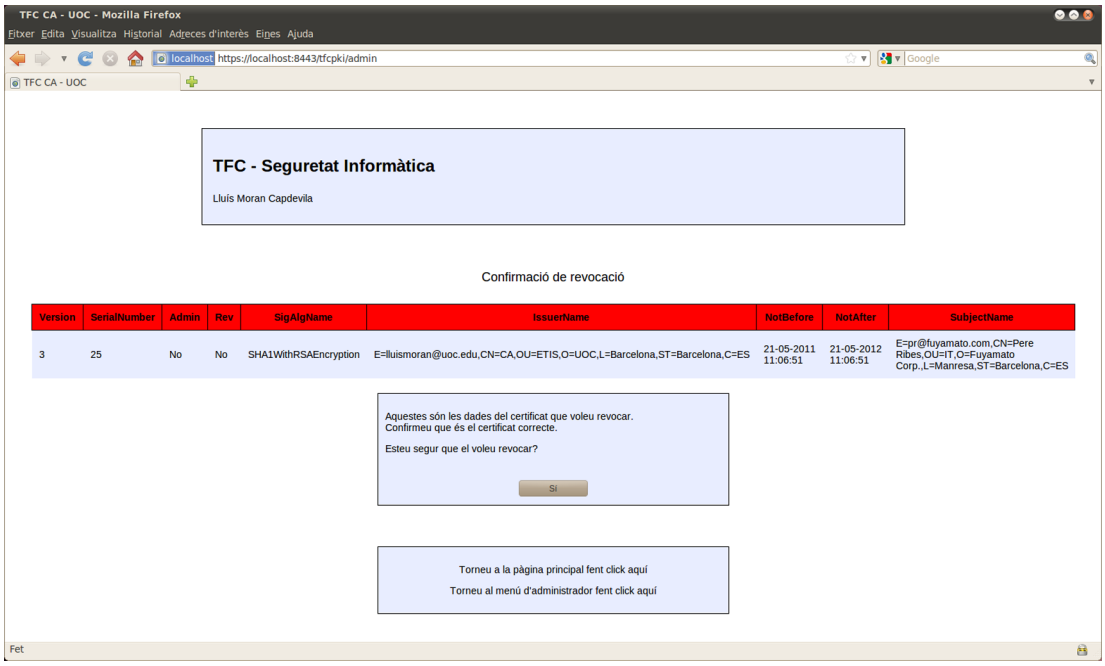

Figura 19: Interfície – Confirmació de revocació

En cas que el certificat no existeixi o ja estigui revocat es mostrarà un avís d'error a l'àrea de notificacions.

Aquest és el procés intern que seguirà l'aplicació:

- Comprovació de la validesa de les dades del formulari (Admin)
- Comprovació que el certificat existeix (Admin i Dbops)
- Comprovació i tractament del cas especial de “Eliminar de la CRL”, mirant si el certificat està a la CRL amb motiu “Retingut” (Admin i Dbops)
- Mostra de les dades del certificat i confirmació de la revocació o mostra d'errors si n'hi ha hagut (Admin)
- Crida al mètode que fa la revocació del certificat (Admin)
- Recol·lecció de les dades de la CA i la CRL (Caops)
- Càrrega de la CRL anterior (Caops)
- Lectura i format de la data de la següent actualització (LoadConfig i Caops)
- Lectura del proper número de sèrie de la CRL (Caops)
- Afegeix el certificat a la llista (Caops)
- Signatura de la CRL (Caops)
- Emmagatzematge de les dades de la nova CRL i del certificat a la llista de revocació (Dbops)
- Guardar la nova CRL en un fitxer (Caops)

► Mostra del missatge de confirmació o error del procés (Admin)

5.7 – Publicació de la llista de revocació de certificats (CRL)

Com ja s'ha esmentat al punt anterior, la llista de revocació de certificats és l'eina que té la CA per fer saber als usuaris quins certificats han deixat de ser vàlids. La llista s'actualitza i es publica cada cop que hi ha un canvi, com per exemple una nova revocació de certificat.

D'altra banda s'ofereix la possibilitat de publicar la llista actualitzada manualment. El motiu de la publicació periòdica de la llista és que els navegadors poden consultar-la automàticament a partir de la data que s'hi inclou per saber la propera actualització, tal com s'indica a l'atribut *NextUpdate*.

Aquest període entre actualitzacions es pot variar mitjançant el fitxer de configuració d'usuari i per defecte té un valor de 30 dies. També en aquest fitxer es pot modificar l'adreça de publicació de la llista que s'inclourà a tots els certificats que emeti la CA.

El procés de publicació és automàtic i només cal confirmar-lo, tal com es veu a la següent figura:

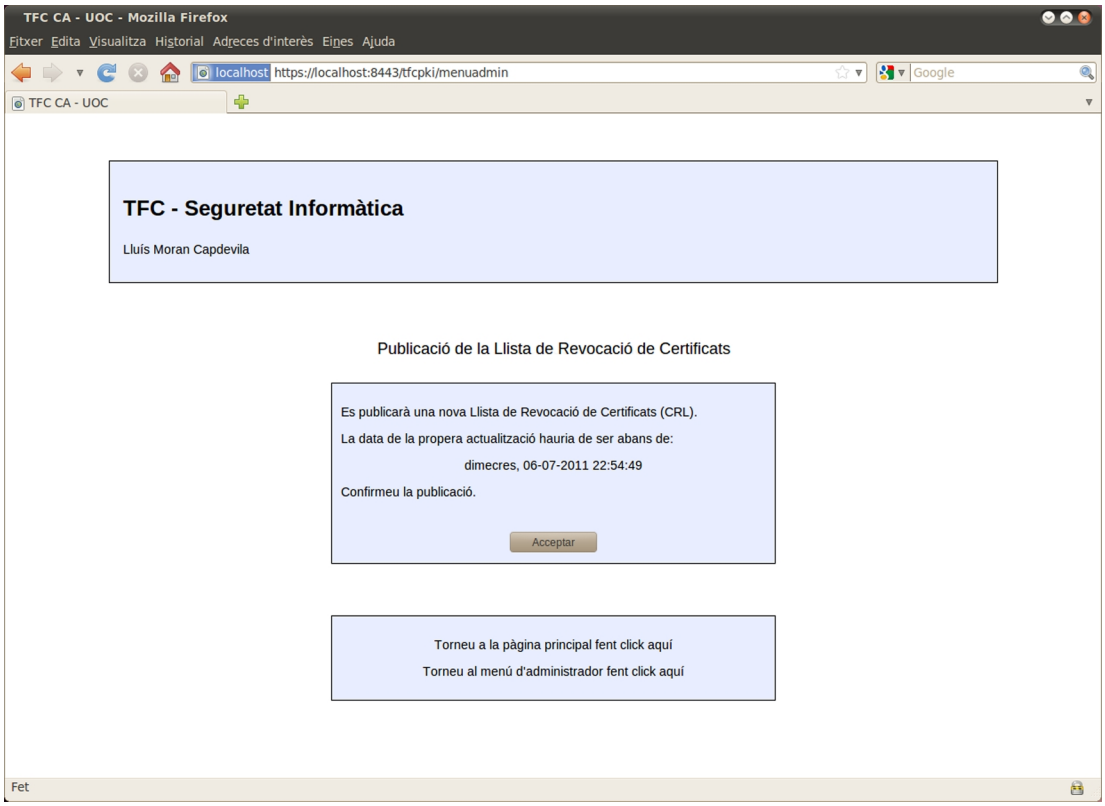

Figura 20: Interfície – Publicació de CRL

El funcionament intern de la funció és el següent:

► Crida del mètode que realitza la publicació de la llista (Admin)

- Recol·lecció de les dades de la CA i la CRL (Caops)
- Càrrega de la CRL anterior (Caops)
- Lectura i format de la data de la següent actualització (LoadConfig i Caops)
- Lectura del proper número de sèrie de la CRL (Caops)
- Signatura de la CRL (Caops)
- Emmagatzematge de les dades de la nova CRL (Dbops)
- Guardar la nova CRL en un fitxer (Caops)
- Mostra del missatge de confirmació o error del procés (Admin)

6 – Conclusions

Crec que els objectius d'aquest projecte s'han assolit amb escreix. La finalitat bàsica era aprofundir en el coneixement del funcionament d'una Autoritat de Certificació en el marc d'una Infraestructura de Clau Pública.

En aquest sentit he aprés molts detalls que, per falta d'espai i temps, no vaig veure a l'assignatura de Criptografia.

A més, he descobert el funcionament de moltes tecnologies que tant sols coneixia pel nom com ara servlets, Tomcat, JSP, ... i que al principi del projecte em semblaven difícilment abastables. En d'altres que ja coneixia com Java, connexions segures o la gestió de bases de dades he reforçat coneixements.

D'altra banda he quedat una mica decepcionat de que les limitacions de temps i els problemes que he trobat no m'hagin permès incloure algunes de les ampliacions que s'havien proposat a l'enunciat.

Tot i això, he descobert que amb els coneixements obtinguts durant els estudis d'Enginyeria Tècnica Informàtica de Sistemes a la UOC, he adquirit una capacitat d'aprenentatge ràpid d'elements i entorns relacionats amb la programació, malgrat que no els conegui gaire d'entrada.

En resum, penso que ha estat una experiència força enriquidora en la meva formació.

Glossari

Autoritat de Certificació (CA o Certification Authority): Entitat de confiança encarregada d'emetre i revocar certificats digitals. És la part central d'una infraestructura de clau pública.

CA: Veure *Autoritat de Certificació.*

Certificat digital: Estructura de dades que serveix per identificar un subjecte o entitat mitjançant la garantia d'un tercer agent de confiança.

CRL: Veure *Llista de Revocació de Certificats.*

DER (Distinguished Encoding Rules): Format de codificació del llenguatge ASN.1 (Abstract Syntax Notation One). És un subconjunt de BER (Basic Encoding Rules) i proporciona una única manera de codificar un valor ASN.1.

Distinguished Name (DN): Seqüència d'atributs en format atribut=valor, separat per comes, que serveix per identificar el subjecte d'un certificat i obtenir-ne informació.

DN: Veure *Distinguished Name.*

Infraestructura de Clau Pública (PKI o Public Key Infraestructure): Conjunt de programari, maquinari, entitats, polítiques i procediments necessaris per a crear, gestionar, distribuir, fer servir, emmagatzemar, i revocar certificats digitals.

JSP (Java Server Pages): Format de pàgines que permet generar contingut web dinàmicament, en format HTML, XML o d'altre tipus.

Llenguatge de Consulta Estructurat (SQL o Structured Query Language): Llenguatge estàndard de comunicació amb bases de dades relacionals.

Llista de Revocació de Certificats (CRL o Certificate Revocation List): Llista de certificats que han estat revocats per l'Autoritat de Certificació corresponent, i per tant ja no són confiables.

PEM: Format de codificació que fa servir Base64 i guarda la informació en línies curtes de caràcters de 7 bits, sobre un alfabet de 64 caràcters.

Petició de certificat: Veure *PKCS#10.*

PKCS#10: Estàndard de petició de certificat especificat a la norma RFC2986. Una petició de certificat (CSR o Certification Signing Request) és un missatge enviat a una CA per a optar a un certificat digital.

PKCS#12: Estàndard que defineix un format de fitxer, habitualment utilitzat per guardar claus privades i certificats X.509, protegits per una contrasenya.

PKI: Veure *Infraestructura de Clau Pública.*

Script Bash: Arxiu de text que conté línies amb comandes executables, per a bash de linux. L'arxiu s'executa des d'un interpret de comandes.

Servlet: És un objecte que s'executa dins del context d'un Contenidor de Servlets i que està especialment dissenyat per oferir contingut dinàmic des d'un servidor web.

SQL: Veure *Llenguatge de Consulta Estructurat.*

x.509: Format estàndard per a certificats de clau pública, llistes de revocació de certificats, certificats d'atribut i algorismes de validació de ruta de certificació.

Bibliografia

HOOK, David. *Beginning Cryptography with Java*. Wrox Press. 2005.

Diversos autors. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 5280)*. The Internet Engineering Task Force (IETF). Maig 2008. <http://tools.ietf.org/html/rfc5280>.

NYSTROM, M., KALISKI, B. *PKCS #10: Certification Request Syntax Specification. Version 1.7 (RFC2986).* The Internet Engineering Task Force (IETF). Novembre 2000. <http://tools.ietf.org/html/rfc2986>.

RSA Laboratories. *PKCS #10 v1.7: Certification Request Syntax Standard.* RSA Laboratories. Maig 2000. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-10/pkcs-10v1_7.pdf>

RSA Laboratories. *PKCS 12 v1.0: Personal Information Exchange Syntax.* RSA Laboratories. Juny 1999. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf>

The Legion of the Bouncy Castle. *Bouncy Castle Cryptography Library 1.46 API.* <http://www.bouncycastle.org/docs/mdocs1.6/index.html>. <http://www.bouncycastle.org/docs/docs1.6/index.html>.

Oracle. *Java Platform Standard Ed. 6 API*. <http://download.oracle.com/javase/6/docs/api/>

The Apache Software Foundation. *Servlet 3.0 API - Apache Tomcat 7.0.14.* <http://tomcat.apache.org/tomcat-7.0-doc/servletapi/index.html>

The Apache Software Foundation. *Documentation Apache Tomcat 7.0.14.* <http://tomcat.apache.org/tomcat-7.0-doc/index.html>. Maig 2011.

Oracle. *MySQL 5.0 Reference Manual.* <http://dev.mysql.com/doc/refman/5.0/en/index.html>.

Altres fonts de referència:

Wikipedia: <http://en.wikipedia.org/>

JavaRanch: <http://www.javaranch.com/>

ServletWorld: <http://www.servletworld.com/>

StackOverflow: <http://stackoverflow.com/>