1122 DESENVOLUPAMENT D'UN GESTOR ÚNIC DE CERTIFICATS DIGITALS

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Rubén Fernández Fusté i dirigit per Helena Rifà Pous.

Bellaterra, Setembre de 2009

El firmant, Helena Rifà Pous , professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Rubén Fernández Fusté

Bellaterra, Setembre de 2009

Firmat:

A la meva família i amics.
En especial a la Cris.

Agraïments

Aquest projecte ha estat possible gràcies a l'ajuda de moltes persones. M'agradaria agrair a la meva família i amics el seu recolzament i la seva ajuda en tot moment. En especial vull des d'aquí tenir un agraïment especial a la Cris, que tant m'ha ajudant en tot moment. Sense ella aquest projecte no hagués sigut possible.
En especial també voldria agrair l'ajuda de la meva tutora Helena Rifà, que a pesar dels problemes que he tingut durant tot el projecte sempre m'ha ajudat a anar cap endavant i ha poder acabar el projecte.
A tots ells, moltes gràcies.

Índex

1 Introducció **1**

1.1 Objectiu/s del projecte 2

1.2 Breu introducció a l'estat de l'art del tema proposat 3

1.3 Estudi de viabilitat del projecte 6

1.4 Planificació temporal del treball 7

2 Anàlisis **11**

2.1 Java 11

2.2 Mozilla Firefox 12

2.3 Microsoft Internet Explorer 13

2.4 Comparativa 15

3 Tecnologies **19**

3.1 Requisits 19

3.2 General 19

3.3 Java 20

3.4 Mozilla Firefox 20

3.5 Microsoft Internet Explorer 20

3.6 Valoracions finals 21

4 Arquitectura i diseny **23**

4.1 Requisits 23

4.2 Diagrama de casos d'ús 24

5 Implementació **27**
5.1 Valoracions inicials 27
5.2 Primera metodologia d'implementació 28
5.3 Segona metodologia d'implementació 29
5.4 Diagrama de classes 30
5.5 Captures de pantalla 32
5.6 Tests unitaris 37
5.6.1 Generació de JUnit 37
5.6.2 Generació de certificats 39

6 Conclusions i treball futur **41**
6.1 Valoracions de la planificació inicial 41
6.2 Valoracions de la implementació 42
6.3 Treball Futur 43

Bibliografia **45**

Índex de figures

1.1 Llista d'etapes del projecte. 8
1.2 Primera part del diagrama de Gantt. 9
1.3 Segona part del diagrama de Gantt. 9

2.1 Certificats d'usuari a Microsoft Windows. 14
2.2 Certificats de sistema a Microsoft Windows. 14
2.3 Taula comparativa. 16

4.1 Diagrama de cas d'ús d'insertar certificats. 24
4.2 Diagrama de cas d'ús d'eliminar certificats. 25
4.3 Diagrama de cas d'ús de llistar certificats. 25
4.4 Diagrama de cas d'ús de llistar certificats. 26
4.5 Diagrama de cas d'ús de veure els detalls dels certificats. 26

5.1 Diagrama de classes de l'aplicació. 31
5.2 Pantalla principal de l'aplicació. 32
5.3 Pantalla per importar certificats. 33
5.4 Pantalla per eliminar certificats. 33
5.5 Pantalla per veure una llista dels certificats. 34
5.6 Pantalla per veure els detalls d'un certificats. 35
5.7 Pantalla de configuració de l'aplicació. 36
5.8 Pantalla d'ajuda de l'aplicació. 36
5.9 Tests unitaris executats a l'aplicació. 38

Capítol 1

Introducció

A l'actualitat i dins el món d'Internet i les comunicacions es realitzen moltes transaccions i operacions que requereixen de l'apliació de mecanismes que garanteixin la seguretat de les accions que es realitzen. Per tal de poder desenvolupar aquests mecanismes de seguretat, és necessari establir i gestionar un model de confiança i això es fa amb certificats digitals.

L'ús d'aplicacions de seguretat i per tant de certificats digitals, hauria d'estar a l'abast de qualsevol usuari d'internet. Ara bé, els usuaris finals en general no són experts en tecnologies de la segurat així que no se'ls pot exigir uns coneixements mínims sobre seguretat informàtica, certificats o criptografia i, ja que han d'usar aquestes eines, se'ls ha de facilitar la gestió amb els certificats tant com es pugui.

A l'actualitat cada aplicació que utilitza certificats digitals fa servir el seu propi gestor i la seva manera pròpia de manejar els certificats.

Per tal de solucionar aquesta mancança i unificar la gestió dels certificats digitals volem implementar una eïna que ens permeti controlar tots els gestors d'una forma ràpida i senzilla.

Per a la realització del projecte ens basarem principalment en integrar els repositoris de certificats dels clients webs (Internet Explorer [IExplorer], Mozilla Firefox [Firefox], etc) ja que últimament són les eïnes més utilitzades.

La implementació d'aquest projecte tindrà diverses aportacions i beneficis molt importants que cal esmentar i tenir en compte.

De cara a l'usuari final requerirà molta menys formació en l'aplicatiu que ha d'utilitzar perque li aportarà sencillesa, ja que no haurà de gestionar els repositoris ni tenir en compte quin repositori fa servir cada aplicació, i gestionant-ne un en tindrà prou. De cara als proveïdors de serveis serà molt útil i beneficiós, ja que disminuirà molt notanblement el nombre d'incidències com a conseqüència dels als problemes d'instal·lació i d'ús dels usuaris finals.

1.1 Objectiu/s del projecte

Aquest projecte es basa en l'implementació d'un gestor de certificats digitals.

Aquesta eina ens ha de permetre gestionar els certificats de forma única, és a dir, que només calgui usar aquest programa a l'hora de la seva gestió i les accions comuns que es realitzen, com ara la seva importació.

S'ha de solucionar la molèstia que existeix fins ara en la seva gestió i importació, ja que les principals eines del mercat utilitzen diferents mètodes per al seu ús, fet que produeix que calgui importar-los un a un i tots de forma diferent.

Una altra característica que ha de tenir el programa és que ha de ser molt portable. S'ha de poder usar desde múltiples plataformes, com per exemple via web, bluetooth o fins i tot des d'una memòria USB.

Desenvolupar un gestor de certificats que prioritzi:

- Integració de la gestió dels certificats digitals de les aplicacions més comuns.
- Facilitat d'utilització de l'aplicació. Ha de ser còmode i senzill gestionar els certificats.
- Facilitat d'aprenentatge de l'aplicació. No ha de costar a l'usuari final aprendre'n el seu funcionament.
- Portabilitat de l'aplicació. Possibilitat d'ús des de múltiples plataformes (Web, bluetooth, memòria USB,...)

1.2 Breu introducció a l'estat de l'art del tema proposat

Després de fer una petita investigació de mercat per tal de comprovar quin és l'estat de l'art del tema proposat i quins tipus d'aplicacions hi existeixen que tinguin la funcionalitat desitjada, extraiem unes conclusions que es detallaràn en aquest apartat.

La recerca es basava en intentar trobar alguna eina que ja existís al mercat que tingués com a mínim una funcionalitat similar a la que es desitja obtenir en aquest projecte.

Aquest requeriment implica que eines de software privatiu, a no ser que fossin exactament el que es busca, quedarien excloses, ja que no es podria disposar del codi font per a realizar les millores necesàries per tal d'adaptar-ho a les nostres necesitats.

La recerca s'ha realizat sobre dos pilars fonamentals, primer de tot s'ha buscat plugins per al navegador web (Firefox en aquest cas) que permetin realizar la gestió de certificats i tinguin unes funcionalitats similars a les que es busquen, i per últim s'ha mirat aplicacions stand-alone amb les mateixes característiques.

El resultat de la recerca de pluggins no ha estat massa encoratjador. Escassament s'han trobat dos o tres pluggins però que presenten poques característiques comuns al que es necessita.

Exemples d'això podrien ser

- View your certificates email adress - Permet veure l'email del propietari del certificat. <`https://addons.mozilla.org/es-ES/firefox/addon/9064`>
- Cert viewer plus - Permet visualitzar les dades dels certificats X.509 amb 2 opcions, PEM o guardant-les a un fitxer. <`https://addons.mozilla.org/esES/firefox/addon/1964`>
- Key Manager - És un addon que permet la generació de certificats via PKI, i la posterior inscripció i delegació dels certificats. <`https://addons.`

mozilla.org/es-ES/firefox/addon/4471>

El resultat de la cerca d'apliacions stand-alone ha estat més interessant. Tot i que les aplicacions no són exactament el que realment necesitem en aquest projecte.

Són aplicacions de software lliure que es poden trobar al repositori de software <http://www.sourceforge.net>

- CertForge - És una aplicació web escrita en java que permet crear, editar i gestionar els certificats digitals. <http://certforge.sourceforge.net/>
- PHP Certificate autorithy - Una altra aplicació web basada en php en aquest cas que permet crear i gestonar certificats. <http://sourceforge.net/projects/php-ca>
- Portecle - És una interfície d'usuari senzilla que permet crear, veure i gestionar certificats, claus, etc. - <http://portecle.sourceforge.net/>

Després d'un anàlisi dels resultats que hem obtingut en la cerca de software que ja existeix al mercat ens plantegem l'opció d'implementar una aplicacíó desde zero adapdata a les nostres necesitats i veure si és factible o no.

Per començar s'ha decidit que el projecte es basarà en tres aplicacions o tecnologíes fonamentals: Internet Explorer, Mozilla Firefox i Java [Java].

En el cas de Mozilla Firefox, s'ha fet una cerca de documentació tècnica per tal de trobar com gestiona aquesta aplicació els certificats digitals, com els guarda, on i en quin format.

Després de visitar recursos electrònics com ara

- <http://www.mozilla.org/projects/security/pki/nss/>
- <https://wiki.mozilla.org/NSS_Shared_DB>

s'ha obtingut resultats força positius trobant sense problema fins i tot l'estructura interna de la base de dades binària (que en el cas de Mozilla Firefox es fan

servir les NSS Shared DB [Nss]) on es guarden els certificats digitals d'aquesta aplicació. També es disposa de la documentació per part de la fundació Mozilla d'una api anomenada NSS (Network Security Services).

En el cas de de aplicacions JAVA s'ha realitzat també una cerca de com es gestionen els certificats.

S'han visitat diverses pàgines com ara:

- <http://java.sun.com/javase/6/docs/technotes/guides/security/index.html>
- <http://java.sun.com/javase/6/docs/technotes/guides/security/certpath/CertPathProgGuide.html>
- <http://java.sun.com/javase/6/docs/technotes/guides/security/overview/jsoverview.html>

Java tracta d'una forma una mica diferent els certificats digitals respecte a Mozilla Firefox. Separa en dos bases de dades diferents les entrades de certificats digitals i claus públiques/privades dels usuaris. El format en el que es guarden aquestes bases de dades anomenades keyStores és el JKS(Java Key Store) [Java3]

I per acabar ens centrem en com gestiona Internet Explorer els certificats digitals; per fer-ho s'analitzen pàgines com ara:

- <http://msdn.microsoft.com/es-es/library/aa302378.aspx>
- <http://technet.microsoft.com/en-us/library/cc728388.aspx>
- <http://technet.microsoft.com/en-us/library/cc757138.aspx>
- <http://msdn.microsoft.com/en-us/library/aa380255(VS.85).aspx>

Internet Explorer es tracta de forma molt diferents els certificats digitals.

En aquest cas, els certificats es guarden dins el registre de Microsoft Windows i es des d'aquí on es gestionen.

Després d'estudiar tots aquests resultats que s'han obtingut, arribem a la conclusió que el software propietari que ja està implementat no ens ofereix solució a les necessitats que tenim. Per altre banda el software lliure que existeix tampoc disposa de les mínimes funcionalitats que necesitem per a la implementació del nostre projecte.

Així que després de veure que no existeix cap tipus de software parcial o totalment ja implementat, la solució més òptima per a la realització del projecte serà realitzar un programa des de l'inici que realitzi la funcionalitat que es desitja.

1.3 Estudi de viabilitat del projecte

Per tal de realitzar l'estudi de viabilitat del projecte s'han estudiat diversos aspectes sobre el projecte.

Primer de tot s'ha realitzat un estudi de les necessitats del projecte. Després de realitzar-lo s'han obtingut els següents resultats.

És necessari arribar a obtenir els objectius que s'han indicat abans. S'ha de poder realitzar una aplicació que pugui gestionar els certificats digitals de forma única.

Posteriorment a l'estudi de les necessitats del projecte s'ha realizat un estudi temporal, és a dir, quant temps trigarà el projecte a poder-se realitzar. Després d'estudiar el projecte i les activitats que cal portar a terme s'ha determinat que per a realitzar el projecte seran necessaris uns quatre mesos , coincidint amb el període en que hauria d' estar finalitzat el projecte per a procedir a la seva entrega.

Per acabar l'estudi de viabilitat el tecer i últim punt que caldria valorar és l'estudi econòmic.

En aquest aspecte al ser un projecte final de carrera, i fer-se només per una persona l'aspecte econòmic no és tan important, ja que únicament caldrà ajustar-se a complir els objectius clarament fixats dins d'un plaç determinat, tot això tenint en compte que els objectius s'hauran de satisfer mantenint uns criteris de qualitat.

Un cop s'han estudiat aquests tres aspectes del pla de viabilitat només queda fer un estudi final amb els tres resultats i valorar en conjunt si aquest projecte serà viable.

S'han trobat eines i llibreríes de gestió de certificats digitals en diversos llenguatges de programació i que a partir d'aquestes eines es pot crear una aplicació com es desitja.

Després de realitzar aquest anàlisi es constata que és viable la realització de l'aplicació.

1.4 Planificació temporal del treball

Per tal de realitzar un projecte ben organitzat i gestionar correctament totes les seves activitats i dates, s'ha realizat una planificació temporal que consta de tres punts claus: documentació, implementació i activitats posteriors a realitzar (veure figura 1.1).

Figura 1.1: Llista d'etapes del projecte.

WBS	Name	Work
1	**Projecte final de carrera**	**120d**
1.1	**Documentació**	**25d**
1.1.1	Investigació bàsica	5d
1.1.2	Recerca estat de l'art	5d
1.1.3	Investigació certificats Firefox	5d
1.1.4	Investigació certificats Explorer	5d
1.1.5	Investigació certificats Java	5d
1.2	**Implementació**	**50d**
1.2.1	Implementació mòdul Java	10d
1.2.2	Implementació mòdul Windows	10d
1.2.3	Implementació mòdul Firefox	10d
1.2.4	Implementació de l'aplicació del PFC	10d
1.2.5	Implementació del test-cases	10d
1.3	**Activitats posteriors**	**45d**
1.3.1	Solució d'errors	15d
1.3.2	Finalització de la memòria	20d
1.3.3	Preparació de la presentació	10d

Així que si sumem tots els plaços de temps que s'han definit obtindrem que necesitarem uns quatre mesos per a poder acabar el projecte.

També a partir d'aquestes activitats s'ha generat un diagrama de Gantt per tal de que es pugui observar visualment totes les activitats a realitzar i es pugui veure a quin punt del projecte s'està (veure figures 1.2 i 1.3).

Figura 1.2: Primera part del diagrama de Gantt.

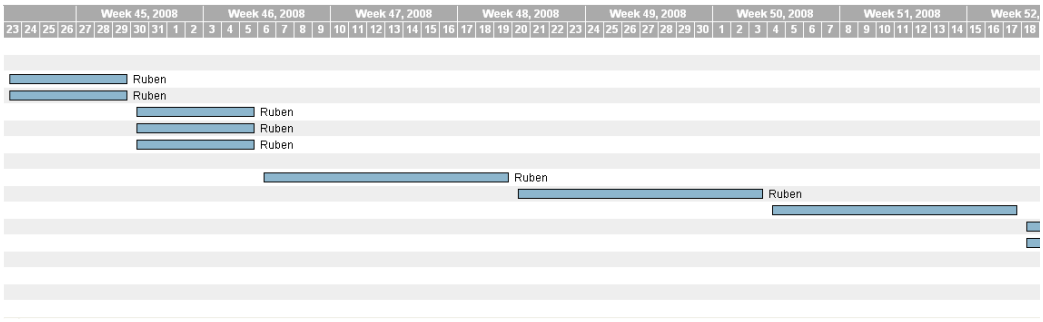
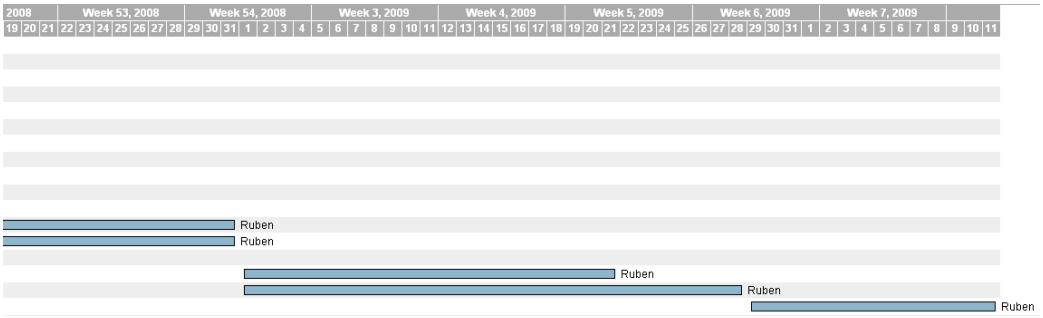


Figura 1.3: Segona part del diagrama de Gantt.



Capítol 2

Anàlisis

Aquest capítol es centrarà en realitzar un anàlisi de les tecnologíes en que es basarà el nostre projecte i s'explicarà com funcionen i quin és el seu ús.

A l'hora de realitzar aquest projecte s'ha treballat en tres tecnologies principalment. Microsoft Internet Explorer, Mozilla Firefox i Java. El projecte ha de poder treballar amb aquestes tecnologies que ara es pasa a analitzar. Cada una d'aquestes tecnologíes tracta de manera diferent els certificats digitals i els enmagatzema de forma diferent.

Per tal de començar l'anàlisis es va seguir la metodología de primer es va obtenir informació genèrica sobre seguretat computacional i certificats digitals principalment per tal de tenir molt clar que són i com funcionen.

Un cop solucionat aquest aspecte, es va proseguir a obtenir informació tècnica i específica sobre cada un dels sistemes, és a dir, com són els certificats en aquest sistema, amb quina estructura s'enmagatzemen, amb quin format i quines eïnes genèriques existeixen per gestionar-los.

2.1 Java

En el cas de la tecnologia Java aquesta proporciona l'enmagatzematge dels certificats i claus via el que s'anomenen matagtzems de certificats (*key* i *store certificates*). Un *keystore* és un magatzem de claus. Les claus privades d' un keystore

tenen una cadena de certificats associades a ell que permeten autentificar la corresponent clau pública. Un keystore també conté certificats d' entitats en les que es confia.

En canvi un *certificate store* és un magatzem de certificats on podem trobar els certificats d'un usuari qualsevol que s'han pogut instal·lar. Les clases estàndards *java.security.KeyStore* i *java.security.cert.CertStore* que proporciona Java permeten interactuar amb les claus i certificats creant-los, important-los o eliminant-los.

La plataforma Java inclou els keystore standards *PKCS11* i *PKCS12*, així com el tipus propierari anomenat *JKS* (Java Key Store). En el món de la criptografía, *PKCS* es refereix als estandàrds de criptografia pública. L'estàndard *PKCS11* [Pkcs11] és una interfície d'accés a un hardware criptogràfic mentre que l'estàndard *PKCS12* [Pkcs12-1] és simplement un contenidor on es poden guardar claus i certificats. El format JKS protegeix cada clau privada amb el seu password individual i també protegeix la integritat del keystore amb un password (pot ser diferent respecte al corresponent a les claus). Aquesta plataforma també inclou un magatzem de claus i certificats predefinits anomenada cacerts que manté el format JKS, que inclou diversos certificats coneguts i de diverses *CA* (autoritats de certificació).

En totes les plateformes Java hi ha una eina que permet gestionar claus i certificats que s'anomena *Keytool*. [Keytool]

2.2 Mozilla Firefox

L'enmagatzematge de les claus i certificat es fa mitjançant bases de dades basades en la tecnologia NSS. NSS és un conjunt de llibreries, apis, utilitats i documentació dissenyada per a la implementació de la part de seguretat d'aplicacions client-servidor.

En aquest cas, Firefox presenta tres bases de dades on s'enmagatzemen els certificats i claus. Aquestes bases de dades o magatzems són arxius d' usuari que estan situats al directori personal de l'usuari.

En el cas d'utilitzar Microsoft Windows podríem trobar-los a una ruta sem-

blant a aquesta: *C:\Documents and Settings\Ruben \Datos de programa\Mozilla\Firefox \Profiles\sylvpyl8.default*. Aquí trobariem el directori de l'usuari Ruben on són els magatzems. En el cas d' usar GNU/Linux podriem trobar-los a una ruta com aquesta: *~/.mozilla/firefox/hm3yeh6a.default* I en el cas d' usar Mac Os X ho podriem trobar a */Users/ruben/Library/ Application Support/Firefox/Profiles/9t7h1512.default*

Cal fer notar en aquest punt que en el cas de Microsoft Windows i GNU/Linux aquests magatzems són a directoris ocults, així que cal que l'usuari permeti veurel's de forma espressa.

Un altre aspecte a tenir en compte són els directoris *9t7h1512*, *sylvpyl8* o *hm3yeh6a*. Són perfils de Mozilla Firefox, on es guarda tota la configuració de l'usuari. El nom del directori sempre és generat arbitràriament.

- **cert8.db** Aquesta base de dades enmagatzema els certificats.
- **key3.db** Aquesta base de dades enmagatzema les claus privades.
- **secmod.db** Aquí es guarda el mòdul de configuració del PKCS11.

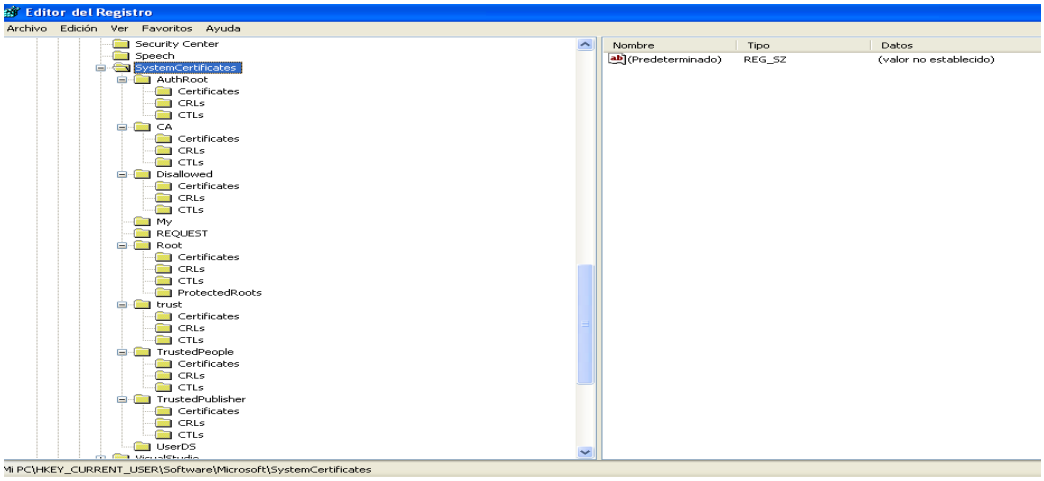
En aquest cas també existeix una aplicació que forma part de la fundació Mozilla anomenat *certutil* que permet gestionar els certificats amb aquesta aplicació.

2.3 Microsoft Internet Explorer

Internet Explorer basa l'enmagatzematge dels certificats en el registre de Microsoft Windows. Existeixen diferents magatzems on es guarden els diferents tipus de certificats. Hi ha dos rutes principals al registre de windows que mantenen els magatzems. Una per els certificats personals i altre per els de sistema.

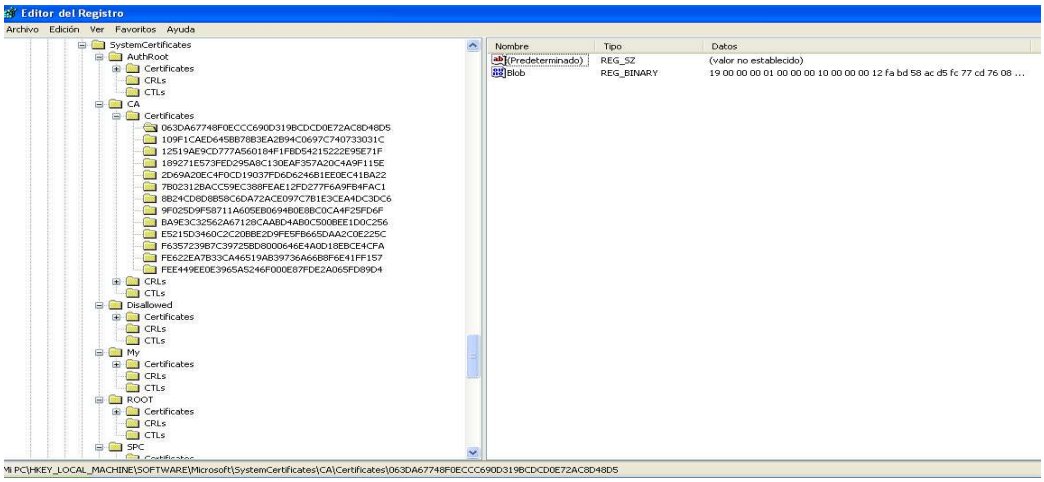
El magatzem de certificats d'usuari es pot trobar sota la ruta *\HKEY_ CURRENT_USER\Software\Microsoft \SystemCertificates* (veure figura 2.1)

Figura 2.1: Certificats d'usuari a Microsoft Windows.



El magatzem de certificats de sistema en canvi es pot trobar a *\HKEY_ LOCAL_MACHINE\Software \Microsoft\SystemCertificates* (veure figura 2.2)

Figura 2.2: Certificats de sistema a Microsoft Windows.



Dins de cada magatzem hi ha altres localitzacions pròpies per cada funció que manté el certificat.

A cada una de les rutes (usuari i sistema) hi ha diverses subrutes que permeten enmagatzemar certificats per al seu ús. Els principals són el magatzem *My*, on es guarden els certificats personals; el magatzem *ROOT* i *CA*, on es guarden els certificats de les autoritats emisores i els certificats de les autoritats de certificació respectivament. També podem trobar els magatzem *Trust* (*Trust*, *TrustedPeople* i *TrustedPublisher*). Aquests magatzems enmagatzemen els certificats en els quals es confia, ja siguin genèrics, de persones o d' empreses. Els certificats en si s'enmagatzemen com una entrada en format binari sota un nom que es realitza mitjançant un hash de la clau del certificat.

Per tal de poder gestionar de forma senzilla existeix també una utilitar proporcionada per Microsoft anomenada *certmgr*.

2.4 Comparativa

Els certificats digitals de les tres aplicacions que es volen gestionar utilitzen diferents tecnologies a l'hora d'enmagatzemar els certificats. Cada una a la seva manera realitzen una funcionalitat semblant. S'utilitzen diferents formats per a aquest proposit de l'enmagatzematge, com pot serusar el registre de Microsoft Windows, o usar un format de base de dades específic o un altre format.

Si es realitza una comparativa entre els formats que s'utilitzen no hi ha gaires semblanças. Java i Firefox en ambdos casos usen una base de dades, mentre que en el cas de Windows es fa servir un mètode molt diferent, el registre de Windows, on s'enmagatzemen els certificats digitals. Cada un dels certificats a Windows s'enmagatzema de forma binaria en base a una estructura que podem observar en [X.509 (1)]

En el clas del present projecte s'ha de poder usar els tres formats alhora, això implica que es necesita poder gestionar-los i usar-los de forma còmode i senzilla. Es necesita usar algun tipus d'eina que permeti l'ús dels diferents formats.

Per tal de realitzar aquestes tasques hi ha dues opcions: O implementar totes

les funcionalitats que es necesiten per al correcte ús de l' aplicació o bé utilitzar usem alguna eina o eines que ja estigui implementada que permeti realitzar la mateixa funcionalitat.

Aquí s'observa (figura 2.3) una taula comparativa de les tecnologies que fem servir en aquest projecte.

Figura 2.3: Taula comparativa.

Certificats Digitals	Java	Firefox	Windows
On es guarden	Directori d'usuari de Java	Directori d'usuari del perfil de Firefox	Registre de Windows
Com es guarden	Es guarden en un magatzem amb format JKS o PKCS12	Es guarden en tres arxius de bases de dades diferents amb el format Berkeley DB	Es guarden en un Blob (Secuència de dades en format binari)

A nivell d'estructura jeràrquica de com organitza cada tecnologia els seus certificats digitals, s'obté la següent estructura:

Microsoft Internet Explorer

- **Usuari**
  - AuthRoot
  - CA
  - Dissallowed
  - My
  - Request
  - Root
  - Trust
  - Trusted People
  - Trusted Publisher

  - User DS

- **Sistema**

  - AuthRoot
  - CA
  - Dissallowed
  - My
  - Root
  - SPC
  - Trust
  - Trusted People
  - Trusted Publisher

Java

- **Usuari**

  - Trusted
  - Secure Site
  - Signer CA
  - Secure Site CA
  - Client Authentication

- **Sistema**

  - Trusted
  - Secure Site
  - Signer CA
  - Secure Site CA
  - Client Authentication

Mozilla Firefox

- **Usuari**
  - Your Certificates
  - People
  - Servers
  - Authorities
  - Others

I, a nivell de compatibilitat entre fabricants, la diferència més gran és que en el cas de Mozilla Firefox, al ser una aplicació d'usuari. no disposa de certificats de sistema, al contrari que Microsoft Internet Explorer (que obté els certificats del registre de Microsoft Windows) o Java (que també disposa d' aplicacions de sistema).

Tal i com hem vist en l'esquema anterior, aquestes teconologies tenen semblances en la seva estructura de certificats, i es poden crear unes correspondències entre les estructures de Java, Microsoft Internet Explorer i Mozilla Firefox simultàniament.

Correspondències

- Signer CA - Root - Authorities
- Secure Site CA - Ca - Servers
- Secure Site - Trust - Others
- Client Authentication - My - People
- Trusted - TrustedPublisher - Other

Capítol 3

Tecnologies

3.1 Requisits

Sobre l'aspecte de les tecnologies van analitzar les diferentes opcions de les que existeixen per tal de poder portar a terme aquest projecte.

En un primer pas es van analitzar les alternatives existents al mercat per tal de poder realitzar l'implementació de tota la funcionalitat propia del projecte.

3.2 General

Hi ha més d'una llibreries de proposit general alhora de tractar amb contingut criptogràfic i certificats digitals.

Cal destacar una llibreria de funcions que d' una forma bastant còmode i senzilla permet fer ús d' eines criptogràfiques com ara claus privades, públiques, certificats, etc.

Aquesta llibreria agrupa la funcionalitat que moltes altres tenen per separat en una de sola. Aquesta llibreria es diu *Bouncy Castle* [Bouncy], es pot obtenir desde la seva web `http://www.bouncycastle.org/` i es *open source*. Aquesta llibreria permet desenvolupar codi amb llenguantge Java.

El terme open source és l'equilavent anglès de codi obert. És a dir, software creat i distribuit lliurement.

3.3 Java

En l'apartat de Javas'ha trobat que existeixen bastantes opcions a l' hora de treballar amb certificats, claus privades, públiques i criprografia en general.

Per una banda es poden fer servir les llibreries de proposit general com ara Bouncy Castle que disposa de la implementació d'una versió en Java, i per altra banda d'alguna classe pròpia de la seva api que realitza funcions amb certificats com ara la *KeyStore* que forma part del paquet *java.security*.

3.4 Mozilla Firefox

Per tal de realitzar la part de Mozilla Firefox després d' analitzar les solucions existents es va descobrir l' existencia d'una llibreria proporcionada per la fundació Mozilla (`http://www.mozilla.org/`) que permet la implementació d'aplicacions amb funcionalitats criptogràfiques i tecnologies *NSS* que són amb les que es basa l'enmagatzematge dels certificats en Firefox.

Aquesta llibreria es diu *JSS* i com indica el seu nom és una adaptació de les NSS per a Java. Es pot trobar més informació sobre aquesta lllibreria a la següent pàgina web `https://developer.mozilla.org/En/JSS`.

3.5 Microsoft Internet Explorer

En l'apartat de Microsoft Internet Explorer s'ha pogut comprovar que si havia diverses opcions a l' hora de realitzar la implementació.

La primera d'aquestes opcions és usar una sèrie de llibreries que proporciona Microsoft per tal de desenvolupar aplicacions amb contingut criptogràfic.

Aquestes són la *CryptoApi, CAPICOM*, etc. Es pot trobar més informació sobre elles en aquesta pàgina web `http://msdn.microsoft.com/en-us/library/aa380255(VS.85).aspx`. L'ús d'aquestes llibreries comportra l' obligació d' usar un llenguatge i un entorn de programació basats en C++ i Microsoft Windows.

En la segona opció a l'hora d' implementar aquesta part, es va observar que que el sistema de certificats es basa en l'ús del registre de Microsoft Windows i es sabia a quina ruta del registre es guardaba cada tipus de certificat només cal accedir al registre per tal de insertar o eliminar certificats.

En aquest cas l'únic que cal és una llibreria que permeti la lectura i escriptura al registre. D'aquestes llibreries n' existeixen de diversos tipus i per a la realització d'aquest projecte s'ha tingut en compte una que es diu *JNIregistry* que ha realitzat el grup *ICE Engineering* (`http://www.trustice.com/java/jnireg/index.shtml`) i permet realitzar tota aquesta funcionalitat mitjançant el llenguatge Java.

3.6 Valoracions finals

Després d'haver analitzat les diferents alternatives existens per a realitzar el projecte, si es tenen en compte els requeriments inicials que s'han de tenir (comptant que l'aspecte de la portabilitat ha de ser un factor clau en el diseny del projecte) es pot observar que el millor llenguatge de programació per a portar a terme aquest projecte serà Java, ja que de tots els llenguatges existents és segurament dels més portables gràcies a la seva màquina virtual i permetra l'ús de la aplicació en una gran quantitat de sistemes operatius.

Al mateix temps Java presenta en la codificació, comparat amb altres alternatives existentens com poden ser C/C++.

Un cop decidit el llenguatge de programació que s'utilitzarà s'ha de decidir quina eina es farà servir per a cada part del projecte.

En el cas de Java la millor opció ha estat usar la llibreria *Bouncy Castle*. Per a la part de Microsoft Internet Explorer, com que l' alternativa que proporciona Microsoft és només per al llenguatge C++, s'utilitzarà la via d' inserció i eliminació de certificats de forma directa mitjançant l'accés al registre de Windows amb la llibreria *JNIregistry*, amb el qual obtimdrem també més senzillessa en la codificacióni només caldrà accedir al registre de Windows i editar-lo.

I per a la part de Mozilla Firefox la llibreria més útíl és la *JSS* ja que permet

l'accés a les bases de dades de certificats de Firefox de forma senzilla i còmode tal i com recomana de fer la fundació Mozilla en l'ús de les seves aplicacions.

Com a eina de desenvolupament es farà servir l'entorn de programació *Eclipse*,(`http://www.eclipse.org/`),un entorn multillenguatgemolt complet, gratuit i senzill d'utilitzar amb el que ja es tenia experiència anteriorment.

D'aquesta manera que la curva d'aprenentatge és gairebé nula i no es perdrà temps per apendre a fer funcionar l'entorn de programació i es podrà passar directament a la part de codificació.

Capítol 4

Arquitectura i diseny

En aquest capítol parlarem de l'arquitectura i el diseny de l'aplicació, és a dir, tractarem l'aspecte dels requisits funcionals que són necesàris a l' hora de fer l'aplicació, què ha de fer i com ha de fer-ho.

També es poden trobar els diagrames de casos d'ús de l'aplicació que expliquen quina serà la funcionalitat i com estarà implementada.

4.1 Requisits

A l' hora de disenyar l'aplicació el primer pas que es va realitzar va ser plantejar-se des d'un principi quins havien de ser els requisits de l'aplicació.

El principal requisit que havia de tenir era que cumplís la funcionalitat que es demanava. És a dir, que fós un gestor únic de certificats digitals i permetés la inserció, eliminació, llistar, veure els detalls dels certificats i poder exportar-los a un fitxer. Això implica disenyar una aplicació que permeti alhora la interacció de les diferents aplicacions amb les que es vol interactuar i permeti la seva gestió.

Altre dels requisits importants en aquesta aplicació és que aquesta està dirigida a un usuari sense masses coneixements de seguretat informàtica, així que havia de ser senzilla en el seu ús, que no li resultés complexa d'aprendre a fer servir. Així que es va decidir realitzar també una interfície gràfica per tal que qualsevol usuari ha de poder-la utilitzar sense requerir un gran entrenament per la seva part.

A l' hora del diseny de l'intefície es va intentar que no fós complexa ni tingués masses elements que poguessin fer distreure a l'usuari de la seva funcionalitat. Es van afegir els mínims botons i opcions possibles per tal que l'usuari pogués realitzar tota la funcionalitat amb el mínim esforç posible.

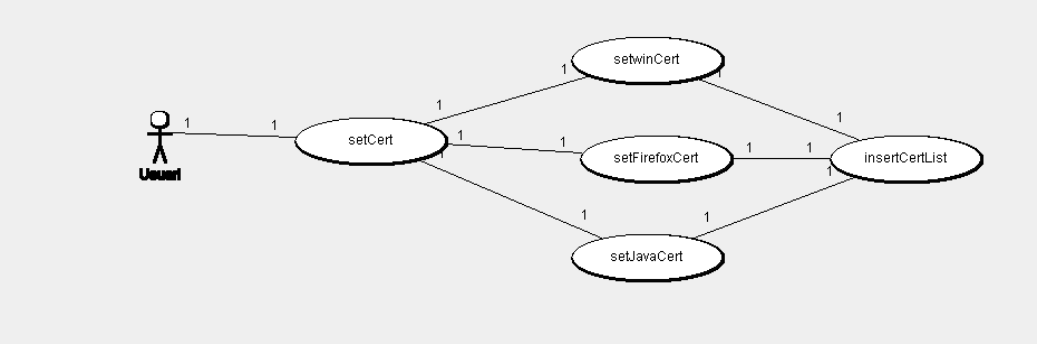
En total existeixen set botóns: per insertar un certificat, per eliminar-los del sistema, per llistar els certificats que tenim instal·lats, per exportar-los a un fitxer físic, per configurar l'aplicació, per mostrar l' ajuda del programa i un botó final per sortir de l'aplicació.

4.2 Diagrama de casos d'ús

En aquesta secció es mostren diversos diagrames de casos d'ús que s'han implementat durant el disseny del projecte.

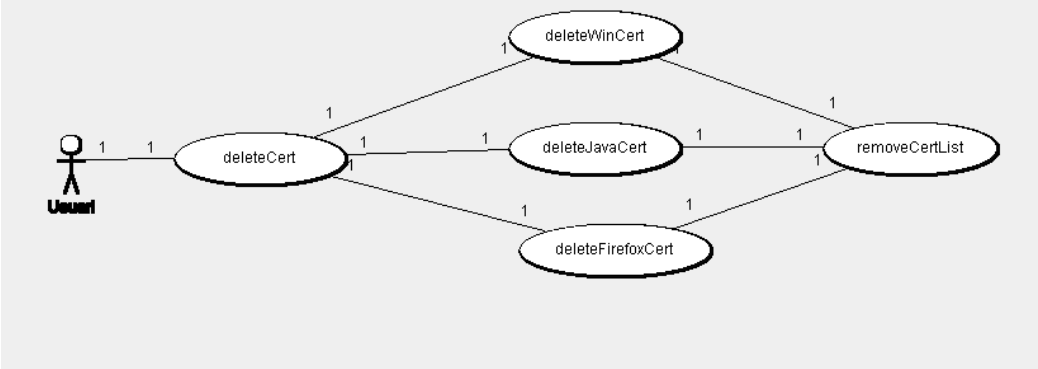
En la figura 4.1 podem observar el diagrama de casos per a la inserció d'un certificat, on es veu la funcionalitat que tindrà i com funcionarà.

Figura 4.1: Diagrama de cas d'ús d'insertar certificats.



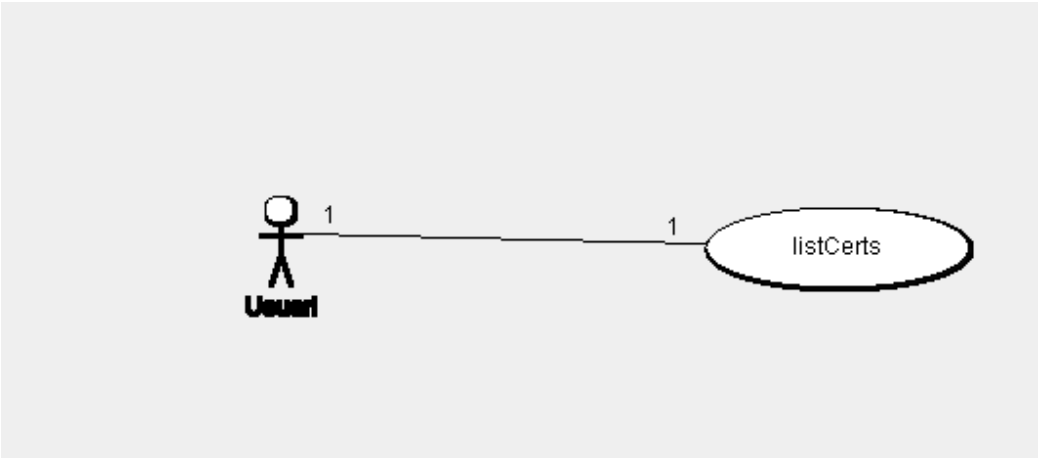
A la figura 4.2 veiem el diagrama per a esborrar un certificat.

Figura 4.2: Diagrama de cas d'ús d'eliminar certificats.



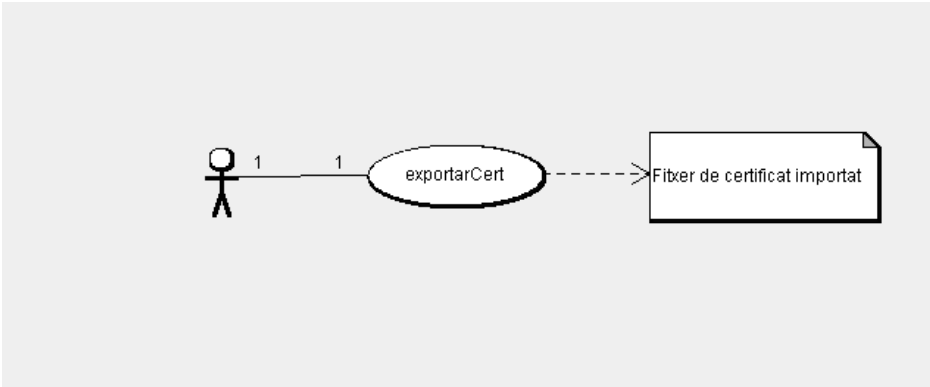
A la figura 4.3 veiem el diagrama per veure una llista dels certificats que tenim instal·lats.

Figura 4.3: Diagrama de cas d'ús de llistar certificats.



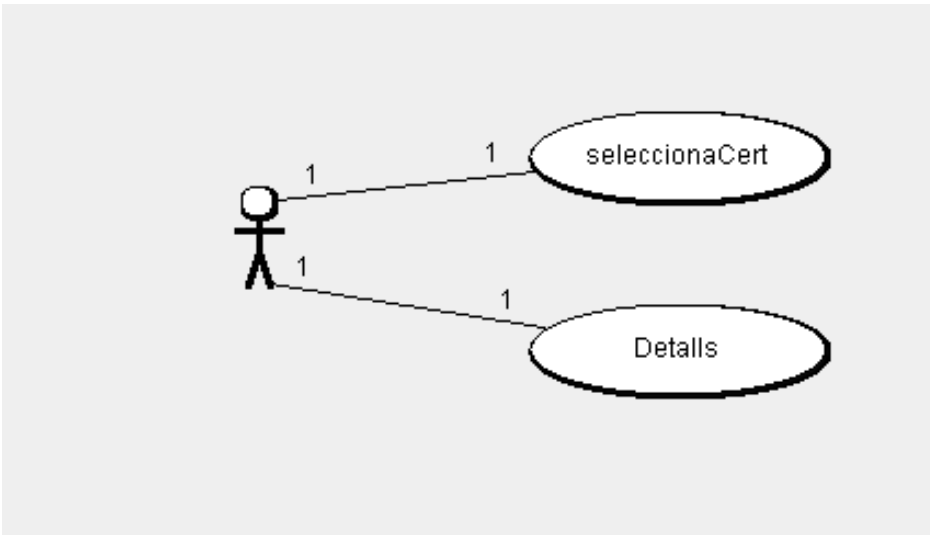
A la figura 4.4 veiem el diagrama per a exportar un certificat que tinguem instal·lat a un fitxer físic a part.

Figura 4.4: Diagrama de cas d'ús de llistar certificats.



L' última figura, 4.5 diagrama UML mostra com es realitza la petició de mostrar els detalls d'un certificat.

Figura 4.5: Diagrama de cas d'ús de veure els detalls dels certificats.



Capítol 5

Implementació

En aquest capítol parlarem de com ha estat la implementació del projecte, quines metodologies existeixen a l'hora de realitzar-lo, quina s'ha acabat utilitzant, amb quin llenguatge s'ha programat, quins són els diagrames de clases, com són les pantalles de l'aplicació i quins són els tests unitaris que s'han realitzat.

5.1 Valoracions inicials

Com ja s'ha indicat anteriorment tenint en compte els requisits del projecte i les llibreries de les que disposàvem es va acabar de decidir que el projecte s'implementaria en Java. [Api]

El Java és un llenguatge la característica principal del qual es la portabilitat que permetrà a la nostra aplicació ser molt portable i que tant es pugui usar en un sistema amb Microsoft Windows, Apple Macintosh o Gnu/Linux, sense tampoc oblidar que es podria arribar a utilitzar vía bluetooth des de qualsevol mòbil que disposi d'aquesta tecnología.

A l'hora de escollir la metodologia amb la qual s'implementa l'aplicació s'han tingut en compte dues opcions. Per una banda es podia implementar tota la funcionalitat que es necessitava o bé es podien utilitzar eines que ja existeixen i ajuden a fer més senzilla la implementació.

En una primer moment es va optar per escollir la metodologia d'implentar tota

la funcionalitat sense usar eines externes.

5.2 Primera metodologia d'implementació

Aquesta primera via d'implementació es va iniciar fent que per a cada una de les parts que havia de gestionar l' aplicació es disenyés un mòdul indepent de la resta i a l' hora d' implementar-lo es va fer servir un patró Adapter.

La necessitat d'usar aquest patró Adapter va venir donada perquè voliem poder disposar d'una interfície comuna per a tots els mòduls que s'haviem d'implementar i al mateix temps disposar d'una manera senzilla d'insertar nous mòduls,amagant la interfície no desitjada que té cada un d'ells.

En un primer moment abans de començar a realitzar la implentació de l'aplicació es va iniciar una petita investigació per aprendre el funcionament de les llibreries e interfícies que es disposaven així com tota la seva documentació.

Un cop es va conèixer el funcionament de totes les eines de les que disposàvem es va començar a implementar. En una fase de programació es va iniciar un procés de diseny de totes la funcionalitat del projecte i la seva posterior codificació en Java a partir de les lliberies que es tenien.

Per exemple, en al cas del mòdul de Java, a partir de la llibreria Bouncy Castle [Bouncy] es va crear codi per a poder llegir un arxiu que contingués un certificat digital per a posteriorment poder-lo insertar a un keystore (magatzem de certificats) determinat (d' usuari o un passat per paràmetres). També es van realitzar funcions per eliminar aquest certificats, llistar-los o mostrar els seus detalls.

Posteriorment a aquest mòdul es va crear codi per els mòduls de Windows i Firefox.

Un cop es va començar a codificat es va trobar un problema a l'hora de usar les llibreries JSS que ens inpedia usar qualsevol tipus de codi basat en aquesta tecnologia.

Després de buscar documentació sobre l' error a Internet tot semblava indicar que era un error de configuració de les llibreries de JSS, tot i que es seguien tot els passos que s'indicaven. No es resolia el problema.

La solució al problema va venir donada per una acció ben diferent. Mitjançant l'aplicació Dependency Walker [Depens] es va descubrir que l'error estava produit no per una configuració incorrecte, sinó per un error de diferents arxius de llibreries dll. Aquest programa va permetre observar quines llibreries depenien d' altres i veure on hi havia els errors.

Un cop comprovat que totes les llibreries dll eren correctes es va poder prosseguir amb la implementació gaires problemes.

Mentre s' implementaven aquests codis, a mesura que es feien revisions de tot el procés d' implementació, es va proposar un petit canvi en el diseny del projecte que podria simplificar-lo, augmentar la seva velocitat d'ús i el temps de codificació. Un altre aspecte que va provocar aquest canvi en el diseny van ser les dificultats que es van produir durant el període d'implementació.

5.3 Segona metodologia d'implementació

Aquesta segona metodologia es basava en la idea de que, en comptes de crear i implementar totes les funcionalitats que necessitàvem al projecte es podrien usar algunes utilitats ja implementades i proporcionades per les mateixes organitzacions que han creat Java, Firefox o Windows. (Sun, Fundació Mozilla o Microsoft).

Totes aquestes empreses proporcionen productes per a la gestió dels certificats. El problema que presenten és que són programes basats en línea de comandes, són en mode texte i s'utilitzen de forma diferent.

Així que l' aplicació s'encarregaria de gestionar amb la resta d' eines els certificats i l'usuari podria utilitzar-los de forma molt sencilla.

Després d'un petit anàlisi de rendiment d' aquesta alternativa s'ha considerat una modificació positiva, tot i que la velocitat d'execució s'incrementarà. És donarà molta més sencillessa al codi, ja que farà el mateix amb menys linies de codi. El temps de codificació del projecte s'escurçarà notablement, així com el temps previst per a la comprovació d'errors, ja que estarem treballant amb eines molt provades i usades ja.

Així que un cop decidit el cambi d'enfocament en la implementació del pro-

jecte es va proseguir per a aquesta via.

Es va crear una funció que permet executar comandes del sistema per tal de poder executar les utilitats que s'anàven a usar.

Aquesta funció per tal de mantenir intactes els requisits del projecte havia de ser molt portable, així que al disenyar-la es va tenir en compte que es pogués usar en entorns Windows, Gnu/Linux i Macintosh.

Un cop realitzada aquesta funció es va prosseguir a analitzar i provar totes i cada una de les comandes que calia executar per tal de gestionar els certificats com es volia.

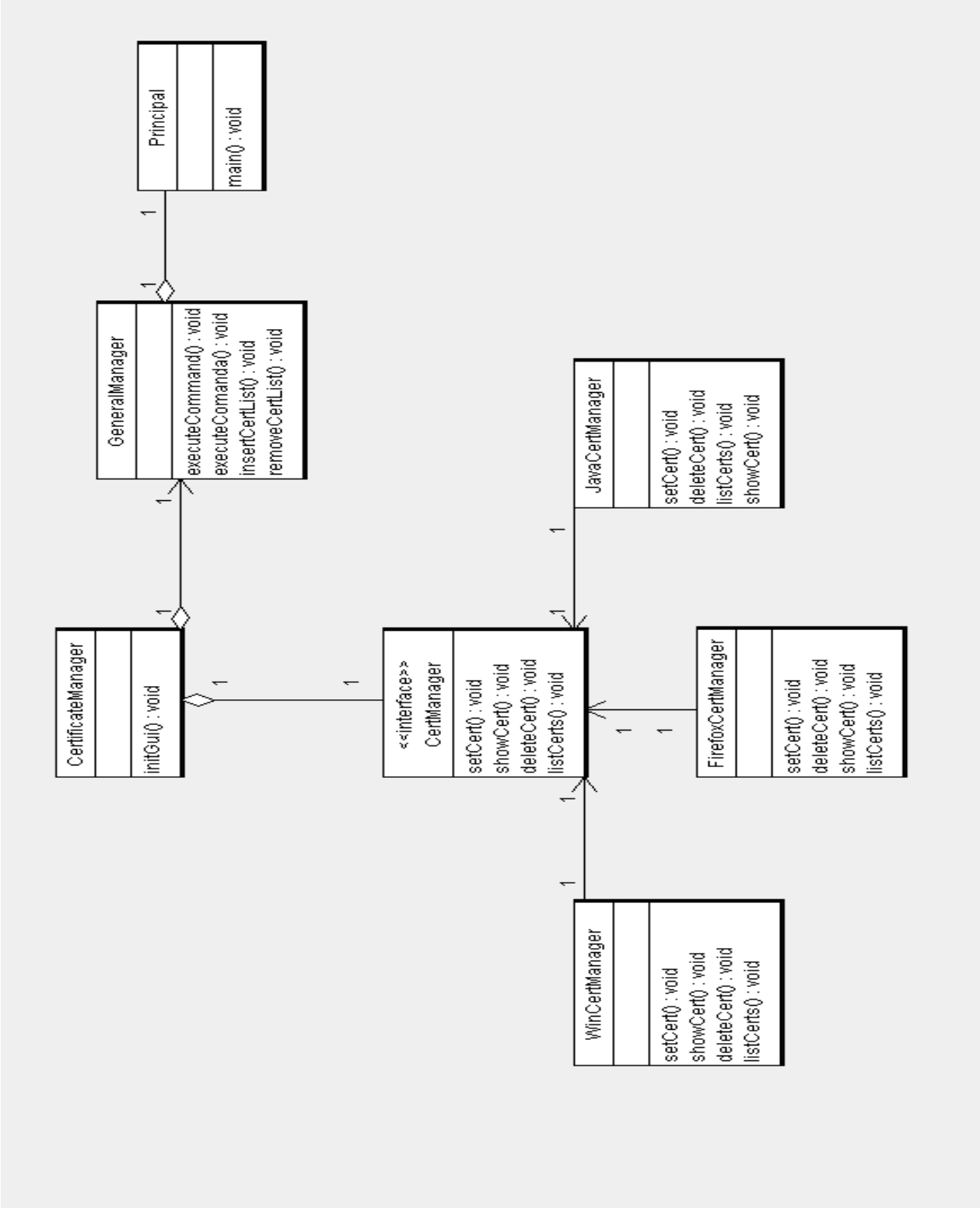
Un cop l'aplicació va estar acabada, seguint el requisit de tenir facilitat d'ús, es vàren realitzar les accions necessàries per a que es pogués instal·lar i executar amb facilitat.

Gràcies a l'eina JavaExe [Installer2] es va poder crear un arxiu executable per a l'aplicació a partir de l'arxiu jar que contenia tot el codi. Un cop solucionat aquest aspecte amb l'eina [Installer1] es va realitzar un procés d'instal·lacio per a l'aplicació. D'aquesta manera qualsevol usuari per tal de poder executar l'aplicació simplement hauria d'instal·lar l'aplicació, executar-la i, després de configurar-la a les seves necesitats, ja es podria utilitzar sense dificultats.

5.4 Diagrama de classes

En la figura 5.1 es pot observar el diagrama de classes que es va disenyar abans de començar la implementació del projecte.

Figura 5.1: Diagrama de classes de l'aplicació.



5.5 Captures de pantalla

En aquesta secció es mostren totes les finestres que disposa l'aplicació i s'indica quina és la seva funcionalitat.

En la figura 5.2 veiem com és la pantalla principal de l'aplicació. Es poden observar tots els botons dels que disposa.

Figura 5.2: Pantalla principal de l'aplicació.

En la figura 5.3 es veu la pantalla des d'on es poden importar els certificats. Aquesta pantalla disposa de diversos camps que s'han d'omplir. La ruta on està el certificat (que es pot seleccionar mitjançant una finestra de selecció), el nom o àlies que se li vol donar al certificat i un password del magatzem on són els certificats.

Figura 5.3: Pantalla per importar certificats.

En la figura 5.4 hi ha la pantalla per a poder eliminar un certificat del nostre sistema. Aquí podem observar que els camps necessàris són el àlies (que anteriorment es van proporcionar durant la instal·lació del certificat) i el password que pugui tenir el magatzem de certificats

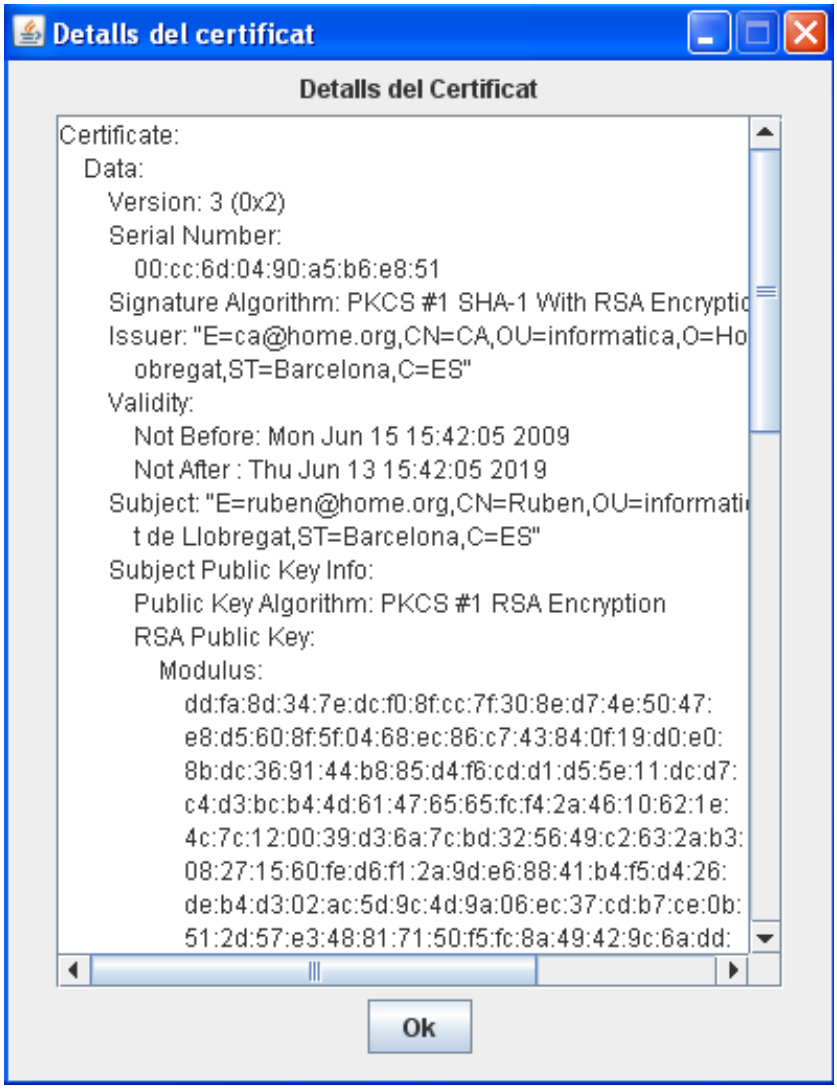
Figura 5.4: Pantalla per eliminar certificats.

En la figura 5.5 s'observa la pantalla desde on es pot veure una llista dels certificats instal·lats en el nostre sistema. Desde aquesta finestra si seleccionem un dels certificats que es mostrin podrem veure els detalls d'aquest certificat.

Figura 5.5: Pantalla per veure una llista dels certificats.

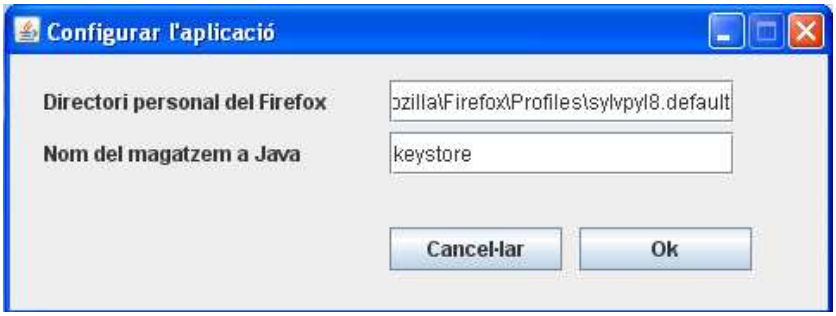
En aquesta figura es pot veure la finestra que es mostrarà quan es veuen els detalls d'un certificat.

Figura 5.6: Pantalla per veure els detalls d'un certificats.



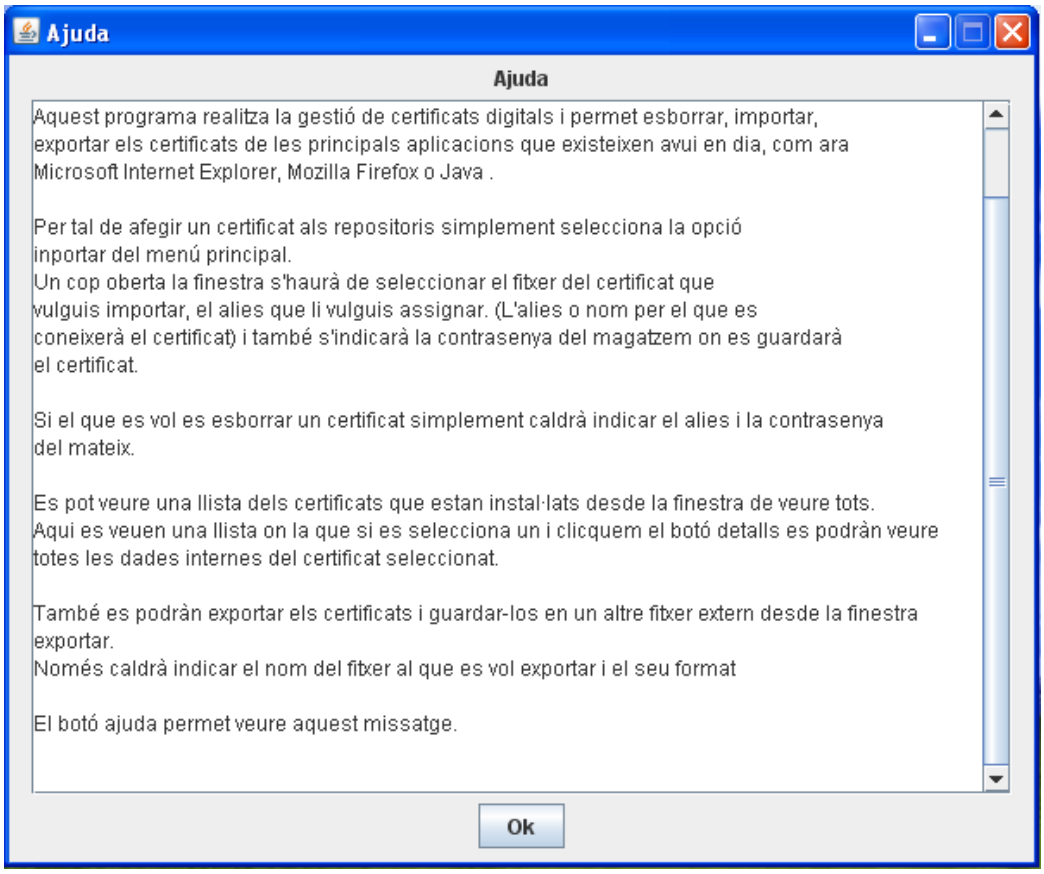
Des d' aquí es pot veure la configuració de l' aplicació i al mateix temps es podrà editar.

Figura 5.7: Pantalla de configuració de l'aplicació.



En aquesta pantalla es pot observar l' ajuda de la que disposa l'aplicació.

Figura 5.8: Pantalla d'ajuda de l'aplicació.



5.6 Tests unitaris

A la vegada que s'anava implementant el projecte s'ha anat generant alguns tests unitaris per tal d' anar comprovant a mesura que es generava codi que aquest era correcte, no generava errors i realitzava la funcionalitat esperada.

Els tests unitaris són una eina eficaç a l' hora de provar el correcte funcionament d'un mòdul de codi. Amb això s'aconsegueix assegurar que cada un dels mòduls funcioni correctament per separat.

5.6.1 Generació de JUnit

Per fer aquestes verificacions s'han generat cinc test unitaris que comproven aquests punts amb algunes característiques determinades.

S'han realitzat amb l'eina JUnit [Junit1], que és un conjunt de biblioteques que s'utilitzen per a fer tests unitaris en aplicacions implementades en java.

JUnit és un conjunt de clases que permet l'execució de clases Java de forma controlada, per a poder avaluar si el funcionament de cada un dels mètodes de les clases és l'esperat. És a dir, en funció d'algun valor d'entrada s'avalua el valor de retorn esperat; si la clase compleix amb l'especificació JUnit retornarà que el mètode de la clase ha passat amb èxit la prova; en el cas que el valor esperat sigui diferent al que s'obté JUnit retornarà un error en el mètode corresponent.

Aquests tests són automatitzables ja que no requereixen intervenció manual; cobreixen la major part de codi de l'aplicació, és a dir són complets; es poden usar tots els cops que es necessiti; Són reutilitzables i a la vegada són independents; No cal l'ús d'un per usar l'altre.

A pesar que els tests ajuden molt durant el període d' implementació cal afegir que no descubriràn tots els errors per si sols.

Per a realitzar les comprovacions es fa servir la llista de certificats que es manté actualitzada amb els certificats que hi ha instal·lats i no. S'ha implementat una funció que diu quants certificats hi ha instal·lats en el moment que se la crida i que ajudarà en aquestes tasques.

Els dos primers tests, realitzen una comprovació de la correcta inserció de

certificats als magatzems corresponents amb una comprovació del nombre de certificats que existeixen posteriorment a la nostre de certificats.

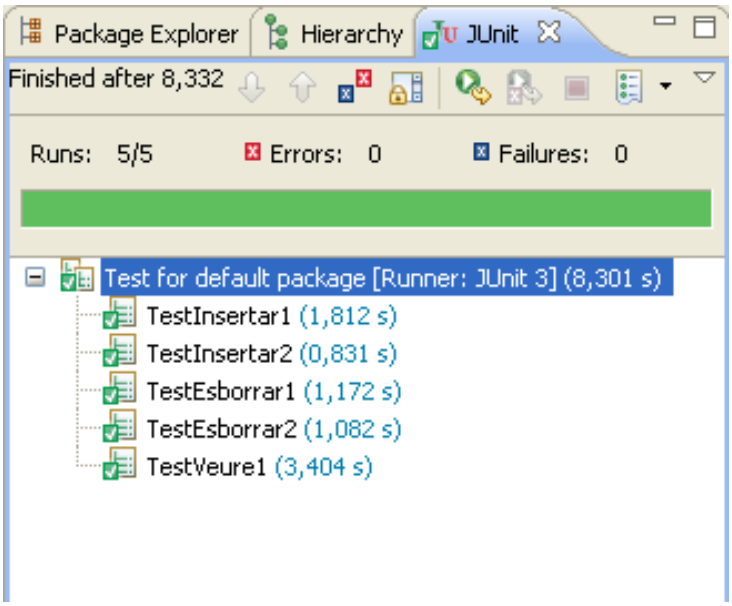
Posteriorment a aquests dos tests sobre la inserció s'han creat dos tests més que comproven la correcta eliminació dels certificats dels seus magatzems. Es realitzen les comprovacions esborrant els dos certificats que es creen al primer test i d'aquesta manera es quedarà amb un balanç net de zero.

I per acabar es farà també una comprovació de les llistes de certificats per veure si a l'hora de la inserció i eliminació dels mateixos existeixen un nombre correcte de certificats a la nostra llista.

Com a petita mostra del funcionament dels tests unitaris aquí es mostra un exemple d'un resultat que obtenim a l'executar-los.

Captura de pantalla on es mostra el resultat d'una de les execucions dels tests unitaris que s'han implementat (veure imatge 5.9).

Figura 5.9: Tests unitaris executats a l'aplicació.



5.6.2 Generació de certificats

Per tal de fer les corresponents proves amb JUnit s'han creat alguns certificats digitals.

L'eina que s'ha utilitzat ha estat Openssl [Openssl], que és un robust paquet d'eines d'administració i llibreries relacionades amb la criptografia, que subministren funcions criptogràfiques a altres paquets com Openssh [Openssh] i navegadors web.

Per a crear els certificats s'han hagut de realitzar les següents comandes en un sistema amb el paquet openssl instal·lat:

Primer creem una autoritat de certificació pròpia per tal de que ens pugui signar els nostres certificats.

openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -days 3650 -out cacert.pem

D'aquesta manera abans de poder fer una simple petició de certificat, es necessita disposar d'una clau pública, així que per a crear-la es fa:

openssl genrsa -des3 -out rubenkey.pem -passout pass:ruben 2048

Un cop es disposa de la clau, es procedeix a fer la petició de certificat:

openssl req -new -key rubenkey.pem -passin pass:ruben -out rubenpeticio.pem

I finalment per a crear el certificat és necessari fer:

openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in rubenpeticio.pem -days 3650 -extfile config.txt -sha1 -CAcreateserial -out ruben.pem

Capítol 6

Conclusions i treball futur

Al llarg d'aquesta memòria s'ha desgranant l'objectiu del projecte i com aquest s'ha anat realitzant.

6.1 Valoracions de la planifiació inicial

Sobre l'aspecte de la planificació inicial del projecte s'ha de dir que s'ha arribat a la realització dels objectius bàsics que s'havien proposat a l' hora d' iniciar el desenvolupament del projecte.

Durant el primer semestre es van complir sense masses problemes els plaços per a les primeres etapes del projecte on es va anar realitzant la documentació del projecte i la investigació de cara a la posterior implementació del mateix. Durant aquestes primeres fases es van investigar les aplicacions ja existients al mercat, possibles eines a usar per la implementació, el seu ús i funcionament, etc.

En una segona etapa es va iniciar la implementació del projecte. En aquest punt van sorgir diversos problemes que han acabat fent que la planificació inicial no es pogués complir tal i com s'havia previst i s'hagi hagut d'acabar presentant aquest projecte al setembre.

Si s'ha de destacar algun dels problemes que s'ha tingut en la implementació es podria donar l'exemple d'un problema amb les llibreries que necesitava el JSS [JSS] per funcionar que estaven corruptes i per més que es configuraben com

s'havia de fer no funcionaba i el segon aspecte que va fer retardar més va ser un canvi en el diseny de l' aplicació en el que es va haver d' eliminar bastant codi fet i refer altre codi.

Encara que com s'ha indicat anteriorment va ser bastant més ràpida aquesta via d' implementació.

Com a conclusions finals podriem indicar que possiblement es va errar en les previsions inicials de la planificació inicial no deixant suficient marge d'error per a posibles problemes durant l' etapa d' implementació del projecte, tot i que aquesta etapa ja comptava amb un marge, encara que a posteriori s'ha vist que era massa petit.

Per a un futur projecte d'aquestes característiques es tindrien molt en compte les experiències obtingudes en aquest projecte i es podria realitzar una planificació molt més acurada i realista amb la qual es podria afinar molt més quan podria finalitzar el projecte i no haver d' anar al final amb retard.

6.2 Valoracions de la implementació

Es partia de l'objectiu d' implementar un projecte que permetés realitzar la gestió dels certificats digitals de diferents aplicacions que usen diferents tecnologies per enmagatzemar els seus certificats. S'havia de realitzar una aplicació que des d' ella mateixa pogués importar o eliminar un certificat i, a la vegada, s'importés o eliminés als magatzems de tots els programes que es fessin servir, independentment de com estessin enmagatzemats o codificats.

Tal i com ja s'ha explicat amb anterioritat, el projecte es va definir amb uns requisits molt clars que s'han intentat mantenir dins les posibilitats.

Tal i com estava planejat, finalment s'ha obtingut una aplicació que, mantenint les bases dels requisits inicials, permet la gestió de certificats desde Microsoft Internet Explorer, Mozilla Firefox o Java. Disposa d'una senzilla però còmode interfície de finestres desde on es pot controlar tota la funcionalitat de l'aplicació.

Gràcies a la senzillessa de diseny és una interfície fàcil d'utilitzar pels usuaris i còmode d' apendre a utilitzar.

S'ha intentat evitar a l'usuari la complicació tècnica que poden arribar a tenir els certificats digitals, la criptografia, la seguretat informàtica i els complexos noms que poden tenir alguns conceptes sobre aquests temes.

6.3 Treball Futur

Després de la realització d'aquest projecte hi ha alguns aspectes que de cara a un treball futur i millores es podrien realitzar.

La primera d' elles és bastant clara tenint en compte com s'ha realitzat el projecte. Es tractaria d' augmentar la velocitat d'execució del mateix al mateix temps que augmentar el control sobre les accions que es realitzen sobre els magatzems de certificats en de la seva gestió.

Així que per a realitzar aquesta millora caldria continuar per el punt on es va fer el canvi de direcció en de la implementació i acabar les funcions que restaven.

Es calcula que això donaria un augment important de velocitat a l'aplicació i es disposaria de més control sobre ella mateixa, ja que hi hauria elements externs que realitzarien la feina però que al no disposar del codi d' aquestes utilitats no es pot assegurar a la perfecció tot el seu funcionament.

Una segona millora que es podria realitzar en aquest projecte seria que es pogués fer la gestió de certificats des de qualsevol ordinador i no només des de el que tingui l'aplicaciò instal·lada.

Aquesta funcionalitat es podria portar a terme tenint un petit servidor dedicat a aquest ús. Caldría també implementar una aplicació web des d' on poder descarregar tots els magatzems de certificats.

Aquesta nova funcionalitat permetria una gran millora en l'aspecte de la portabilitat i la mobilitat. Dos aspectes claus en els que es basaven els requisits de l'aplicació des l' inici del projecte.

I un exemple d' ús amb el que seria de molta utilitat podria ser un usuari amb una gran mobilitat que està tot el dia fora de l'oficina i molts cops no pot utilitzar el seu ordinador propi.

Una altra possibilitat de millora vindria donada per la implementació que per-

metés el funcionament de l'aplicació via Bluetooth. És a dir, des d' un mòbil amb aquesta tecnologia es disposaria d' un aplicatiu que conectaria amb l'ordinador mitjançant Bluetooth i permetria la gestió i ús dels certificats digitals i claus privades dels usuaris.

Aquesta seria una altra opció de cara a realitzar una funcionalitat similar a l'anterior però en la qual l'usuari podria gestionar-ho tot directament des del seu propi mòbil i amb molta comoditat.

Una altra millora que es podria fer seria implementar una millora en la configuració de l'aplicació. D'aquesta manera no caldria afegir les dades de configuració manualment sinó que l'aplicació automàticament ho realitzaria detectant on hi ha els directoris adecuats.

Bibliografia

[X.509 (1)] Anatomy of an X.509 v3 Certificate.
<http://www.jensign.com/JavaScience/GetTBSCert/>

[Explorer1] Almacenes de certificados.
<http://msdn.microsoft.com/es-es/library/aa302378.aspx>

[ArgoUML] ArgoUML Modeling tool.
<http://argouml.tigris.org/>

[Explorer6] CAPICOM Reference.
<http://msdn.microsoft.com/en-us/library/aa375732(VS.85).aspx>

[Latex] The Comprehensive LATEX Symbol List.
<http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>

[Explorer4] Creating, Viewing, and Managing Certificates.
<http://msdn.microsoft.com/en-us/library/aa379872(VS.85).aspx>

[Lucena] Criptografía y seguridad en Computadores.
<http://wwwdi.ujaen.es/~mlucena/wiki/pmwiki.php?n=Main.HomePage>

[Swing1] Creating a GUI with JFC/Swing.
<http://java.sun.com/docs/books/tutorial/uiswing/>

[Depens] Dependency Walker.
<http://www.dependencywalker.com/>

[Eclipse] Eclipse.org.
<http://www.eclipse.org/>

[PKI1] Everything you never wanted to know about PKI but were forced to find out.
<http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitutorial.pdf>

[Firefox] Mozilla Firefox.
<http://www.mozilla-europe.org/es/firefox/>

[Installer1] Free Windows Installer.
<http://www.advancedinstaller.com/>

[Latex] KOPKA, H.; W.DALY, P. (2004). Guide to latex. Tools and techniques for computer typesetting. 4th ed. Ed.Pearson Education: Massachusetts

[Certmgr.exe] Herramienta de administración de certificados (Certmgr.exe).
<http://msdn.microsoft.com/es-es/library/e78byta0(VS.80).aspx>

[Installer2] JavaExe.
<http://devwizard.free.fr/html/en/JavaExe.html>

[Api] Java Platform, Standard Edition 6 API Specification.
<http://java.sun.com/javase/6/docs/api/>

[Java] Pàgina oficial de Java.
<http://www.java.com/es>

[Java2] Java security architecture.
<http://www.j2ee.me/j2se/1.4.2/docs/guide/security/spec/security-specTOC.fm.html>

[Java1] Java Security Overview.
<http://java.sun.com/javase/6/docs/technotes/guides/security/overview/jsoverview.html>

[Keytool] keytool - Key and Certificate Management Tool.
<http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html>

[Java3] Key Management.
<http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html>

[Explorer2] Manage Certificates and Certificate Stores.
<http://www.aspencrypt.com/task_certs.html>

[IExplorer] Microsoft Windows Internet Explorer.
<http://www.microsoft.com/spain/windows/products/winfamily/ie/default.mspx>

[Microsoft] Microsoft.public.security.crypto.
<http://groups.google.com/group/microsoft.public.security.crypto/topics>

[Mozilla1] Mozilla.dev.security.
<http://groups.google.com/group/mozilla.dev.security/topics>

[Mozilla2] Mozilla.dev.tech.crypto.
<http://groups.google.com/group/mozilla.dev.tech.crypto/topics>

[Nss] NSS Shared DB.
<https://wiki.mozilla.org/NSS_Shared_DB>

[JSS] Network Security Services for Java (JSS).
<http://www.mozilla.org/projects/security/pki/jss/>

[Openssh] OpenSSH: Keeping your communiqués secret.
<http://www.openssh.com/>

[Openssl] OpenSSL: The open source toolkit for SSL/TLS.
<http://www.openssl.org/>

[Pkcs11] PKCS 11 v2.20: Cryptographic Token Interface Standard.
<ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf>

[Pkcs12-1] PKCS 12 v1.0: Personal Information Exchange Syntax.
<ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf>

[Pkcs12-2] Pkcs 12 v1.0 TC 1.
<ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12-tc1.pdf>

[PKI2] PKI:It’s not dead, just resting.
<http://www.cs.auckland.ac.nz/~pgut001/pubs/notdead.pdf>

[Explorer5] System Store Locations.
<http://msdn.microsoft.com/en-us/library/aa388136(VS.85).aspx>

[SWT] SCARPINO,M;HOLDER,S;NG S.;MIHALKOVIC,L(2005). SWT/JFace in action. 1st ed. Ed. Manning Publications. United States

[Bouncy] The legion of the bouncy castle.
<http://www.bouncycastle.org/>

[Swing2] Tutorial de Java - Swing.
<http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/parte14/cap14-1.html>

[Certutil] Using the Certificate Database Tool.
<http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html>

[uml05] UML Unified Modeling Language, juny 2005.
<http://www.uml.org/>

[Junit] LINK,J (2002). Unit Testing in java. How tests drives the code. 1st ed. Ed. Morgan Kaumann Publishers: Germany

[Junit1] JUnit.org Resources for test driven developement.
<http://www.junit.org/>

[Explorer3] Viewing certificate information.
<http://technet.microsoft.com/en-us/library/cc738650(WS.10).aspx>

[X.509 (2)] X.509 style guide.
<http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt>

[Registry] Windows Registry API Native Interface.
<http://www.trustice.com/java/jnireg/index.shtml>

Firmat:

Bellaterra, Setembre de 2009

Resum

Aquest projecte s'ha desenvolupat en l'àmbit de la seguretat informàtica i té com a objectiu la creació d'una aplicació que permeti la gestió dels certificats digitals de diferents aplicacions i tecnologies a la vegada i de forma conjunta, estalviant a l'usuari gestionar-los de forma individual. Al mateix temps aquest projecte pretén disminuir la complexitat d'alguns aspectes de la seguretat als que no tots els usuaris dels certificats digitals hi estan familiaritzats.

Resumen

Este proyecto se ha desarrollado en el ámbito de la seguridad informática y tiene como objetivo la creación de una aplicación que permita la gestión de los certificados digitales de diferentes aplicaciones y tecnologías a la vez y de forma conjunta, ahorrando al usuario su gestión de forma individual. Al mismo tiempo este proyecto pretende disminuir la complejidad de algunos aspectos de la seguridad a los que no todos los usuarios de los certificados están familiarizados.

Abstract

This project developed in the area of the computer security aims to achieve the creation of an application for the management of the digital certificates of different applications and technologies simultaneously and of joint form, saving the user it's management of an individual form. At the same time this project tries to diminish the complexity of some aspects of the security to which not all the users of the certificates are acquainted.