Esquema criptogràfic per historials mèdics segurs

Silvia Cernuda Ibañez
Enginyeria en Informàtica

Jordi Castellà Roca
Consultor

19 de Juny del 2006

Dedicatòria i Agraïments

Molts són els que saben que porto ja molts anys intentant acabar aquesta carrera i també l'esforç tan gran que ha suposat acabar-la. Ara només hem toca agrair a tot el que m'ha fet costat durant aquest temps.

En especial li vull agrair al Quim, que m'ha suportat i ajudat molt durant aquestes ultimes setmanes.

També li vull agrair als meus amics que alguns han compartit les mateixes sensacions que jo en aquestes dates. Als meus companys i amics de feina, la Rosi i el Manolo, que m'han encoratjat i animat fins els últims dies.

Agrair-li al meu Consultor l'ajuda donada i la paciència, doncs arribar al final ha estat el més costos.

I Finalment li dedico a la meva família, per fi, ja he acabat!!!

Resum

Quan entrem a la consulta del metge, moltes vegades ens ha passat que encara no li havien pujat al metge el nostre expedient. O quan ens hem canviat de consulta (CAP) ens han dit que trigarien uns dies en tramitar l'expedient cap a l'altra consulta. Avui dia sona molt estrany tot això, doncs tenim al nostre abast tot un conjunt d'eines tecnològiques que ens permetrien tenir-ho tot emmagatzemat i accessible des de qualsevol lloc.

Les xarxes de comunicacions ens permeten accedir a un gran volum d'informació molt ràpidament sense la necessitat de desplaçar-nos, i amb independència de l'instant de temps. Si avui dia ja en tenim això ara només ens falta afegir-hi la seguretat que hi cal en unes dades tant importants com un expedient mèdic, on la confidencialitat, la integritat, la autenticitat i el no-repudi, son aspectes claus.

Aquest PFC s'encarregarà bàsicament de desenvolupar una eina capaç d'emmagatzemar i gestionar expedients mèdics de forma segura i garantint els aspectes de seguretat anteriorment comentats.

Per això s'han utilitzat diferents eines que ens han ajudat a assolir les diferents parts d'aquest PFC. Hem utilitzat les llibreries criptogràfiques IAIK per garantir els aspectes de seguretat, l'estàndard XML per representar les dades i JDOM per gestionar-les, RMI per poder comunicar les diferents parts de l'aplicació, MySQL per emmagatzemar les dades i Java per desenvolupar el projecte.

L'aplicació final tindrà una part servidor, el Gestor, que serà l'encarregat de controlar la gestió de les dades de forma segura, i el Client, que serà també la part gràfica que veuran els usuaris i des de on s'accedirà als serveis que proporcioni el servidor remotament.

Índex

Índex de figures

Índex de taules

1 Introducció

1.1 Justificació del PFC

Les xarxes de comunicacions ens permeten accedir a un gran volum d'informació molt ràpidament sense la necessitat de desplaçar-nos, i amb independència de l'instant de temps. Aquests avantatges aporten un valor afegit més gran encara quan la persona que accedeix a la informació és un metge que consulta el historial mèdic d'un pacient. Les dades del historial poden ajudar al metge a prendre una decisió correcta en la diagnosi i tractament que ha de rebre el pacient.

El historial mèdic d'un pacient es una informació de gran valor, i per tant s'ha de protegir. Cal garantir que només serà modificada pel personal qualificat. Un altre punt important és la confidencialitat. Les dades mèdiques d'una persona són altament confidencials, i només poden ser accedides pel pacient o per personal mèdic.

Un altra valor afegit és el fet d'accedir a una informació electrònica i guardada amb un risc de pèrdua gairebé inexistent. Normalment, els historials mèdics només estan en paper i, per tant, en el CAP on va habitualment el pacient, suposant un risc de pèrdua elevat a l'hora de fer qualsevol trasllat. Amb aquest sistema es garanteix un accés a les dades independent del lloc on hi vagi el pacient (si s'escau) i sobretot segur.

1.2 Context en el qual es desenvolupa

Fa relativament pocs anys, la tecnologia a arribat a metges i hospitals en quant al tema que ens ocupa, els expedients i historials mèdics. Aquestes dades son molt confidencials i es per això que molts son els reticents a ficar aquest tipus de documentació en un sistema que estigui a l'abast de tothom.

Aquesta, és doncs, la principal raó d'aquest "retràs" al mon mèdic, però també es una raó de pes per portar-ho a la realitat garantint unes mesures de seguretat eficaces, a més d'aportar altres avantatges que seran ben segur, molt ben rebuts.

1.3 Objectius del PFC

L'objectiu d'aquest PFC és implementar un esquema criptogràfic per gestionar de forma segura els historials mèdics dels pacients a través d'una xarxa de comunicacions.

Com a resultat del PFC s'obtindrà un sistema amb els components software següents:

- Aplicació Metge: Permet que un metge pugui consultar i modificar el historial d'un pacient de forma segura
- Aplicació Pacient: El pacient utilitzarà aquesta aplicació per consultar les dades del seu historial
- Gestor central: El gestor central es qui té el repositori amb tots els historials i en controla la seva gestió.

A la implementació del PFC s'inclouran el protocols bàsics per aquesta gestió, com son:

- L'autenticació dels usuaris
- La consulta d'un expedient
- La consulta dels pacients d'un metge
- La inserció de dades a un historial mèdic

Per cada protocol es garantiran diferents aspectes de seguretat com son:

- Confidencialitat: S'ha de preservar la confidencialitat de les dades del historial mèdic dels pacients
- Autenticitat: La informació que es guarda en el sistema ha de disposar d'una prova de la seva autenticitat.
- Integritat: Un cop la informació ha estat generada s'ha de garantir en tot moment la seva integritat.
- No-repudi: Si un usuari del sistema fa una certa acció més tard no pot negar que l'hagi realitzada.

1.4 Enfocament i mètode seguit

El projecte s'ha dividit en diferents fases per tal de realitzar una implementació modular i garantir en tot moment el correcte funcionament. Així docs, per cada fase s'implementarà el codi, es faran les proves i s'integrarà amb la fase anterior.

D'aquesta manera es podran anar ampliant les funcionalitats de l'aplicació d'una manera esglaonada. Per cada fase es realitzarà la documentació oportuna i al final s'integrarà per donar pas a la memòria del PFC.

El projecte consta de les següents fases i gairebé totes representen una PAC:

- Instal·lació IAIK i PKI
- Estudi del sistema
- Definició i implementació dels protocols criptogràfics
- Definició i implementació de la representació de les dades (XML)
- Comunicació dels components (RMI)
- Gestió de la informació (DB)
- Interfície Client
- Documentació

Per la implementació del projecte s'ha utilitzat en tot moment software de lliure distribució. D'aquesta manera la implementació s'ha realitzat utilitzant codi Java sota l'entorn de desenvolupament Eclipse. La base de dades utilitzada és MySQL i per la llibreria criptogràfica, l'IAIK, que encara que no és gratuïta, es lliure per a fins educatius i d'investigació.

Pel desenvolupament del projecte s'ha utilitzat un PC Windows XP, encara que al tractar-se d'un desenvolupament en JAVA y com que totes les demes eines utilitzades ho permeten, es podria haver utilitzat un Linux/Unix.

1.5 Planificació del projecte

Per cada fase del projecte s'ha intentat realitzar una PAC amb unes dates d'entrega concretes que a es detallen a la Taula 1. Planificació Projecte:

Setmanes	Objectiu	PAC
1 (5 Març)	Instal·lació IAIK i PKI	
2 (12 Març)	Estudi del Sistema	PAC 1
3 (19 Març)	Definició protocols i planificació	PAC 2
4 (26 Març)	Definició protocols i planificació	PAC 2
5 (2 Abril)	Implementació protocols (criptografia)	PAC 2
6 (9 Abril)	Implementació protocols (criptografia)	PAC 2
7 (16 Abril)	Implementació protocols (XML)	PAC 3
8 (23 Abril)	Implementació protocols (XML)	PAC 3
9 (30 Abril)	Servidor RMI	PAC 4
10 (7 Maig)	Client RMI	PAC 4
11 (14 Maig)	DB	PAC 5
12 (21 Maig)	Vista gestor	PAC 6
13 (28 Maig)	Vista pacient	PAC 7
14 (4 Juny)	Vista metge	PAC 8
15 (11 Juny)	Vista metge	PAC 8
16 (18 Juny)	Documentació PFC	PAC 9
17 (19 Juny)	Entrega PFC	

Taula 1. Planificació Projecte

1.6 Productes obtinguts

El sistema consta dels següents productes:

- Aplicació Client: Permet que un metge pugui consultar i modificar el historial d'un pacient de forma segura i un pacient pugui consultar el seu historial.
- Aplicació Gestor: Controlarà l'autenticació, la gestió dels historials i l'accés als mateixos de forma segura.
- Repositori historials: Base de dades que contindrà tots els historials i tota la informació necessària per garantir la seguretat de la aplicació.

1.7 Breu descripció dels altres capítols de la memòria

Instal·lació IAIK i PKI

Instal·lació JDK de SUN, l'IAIK i de les politiques de seguretat de Java que permeten qualsevol longitud de clau, Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0 RC.

Estudi del sistema

Estudi de les necessitat del sistema, actors, serveis/accions, informació de les dades, etc.

Definició i implementació dels protocols criptogràfics

En aquesta fase es definiran els protocols criptogràfics corresponents a cada acció o servei, es planificarà la seva implementació i es portarà a terme.

Definició i implementació de la representació de les dades (XML)

Per realitzar les transferències de dades que utilitzaran els protocols entre Client i Gestor s'utilitzarà el protocol de dades XML. Es definiran i implementaran les estructures necessàries per el intercanvi.

Comunicació dels components (RMI)

Els clients es comunicaran amb el gestor del sistema per tal de sol·licitar un servei. Per realitzar aquesta sol·licitud es farà servir RMI, incorporada a l'API estàndard de Java. RMI consta d'un servidor que publicarà els serveis que el client necessita i al qual accedirà de forma transparent. S'implementaran les connexions entre client servidor.

Gestió de la informació (DB)

Una base de dades permetrà emmagatzemar els historial mèdics i tota la informació necessària per al seu accés. Es definirà el model de dades, la creació i accés a la base de dades.

Interfície Gràfica

Es definiran les diferents interfícies depenent del rol del usuari que accedeixi al aplicació. Es donarà una explicació des les interfícies i com realitzar les diferents accions.

2 Estudi del Sistema

2.1 Introducció

Abans de començar amb la implementació del projecte, primer s'ha realitzat un estudi del sistema en el qual es realitza un anàlisi dels diferents actors, accions/serveis que podria donar el sistema, com es gestionarà la informació i els requisits de seguretat que s'hauran de garantir.

2.2 Actors del sistema

- Gestor del Sistema: Actor que representa a l'administrador del sistema. Serà l'encarregat de controlar l'autenticació, la gestió dels historial i l'accés als mateixos.

- Metges: Actor que representa el rol dels Metges. Podran realitzar tasques de consulta i modificació dels historials mèdics.

- Pacients: Actor que representa el rol dels Pacients. En principi nomes podran consultar el seu historial, però en cap moment modificar-lo. Es podria pensar en la possible modificació de les dades personals.

2.3 Accions/serveis del sistema

- Autenticació d'usuaris: Quan s'entra al sistema s'ha de fer de forma segura, tenint en compte quin usuari es i quin rol, per determinar a que te accés i a que no.

- Consulta d'un expedient: Cada actor tindrà accés a l'expedient d'un pacient i es pot pensar que depenent del rol, a diferents parts de l'expedient.

- Consulta dels pacients assignats a un metge: Els Metges podran veure quins son els pacients que tenen assignats.

- Inserció de dades: Els Metges podran afegir dades al historial mèdic d'un pacient.

2.4 Gestió de la informació

2.4.1 Introducció

En aquest punt, s'han de definir dos conceptes: Expedient i Historial mèdic. Un expedient és tota la documentació que fa referència a un pacient, ja siguin les seves dades personals, mediques o el registre de les seves visites. Es aquí on entra el Historial mèdic, que es el que engloba totes les visites del pacient, es una "historia" de les seves visites, i que va inclòs al seu expedient.

2.4.2 Parts d'un expedient

- Dades personals: NIF, Nom, Cognoms, num. Seguretat Social
- Dades contacte: Tlf, adreça
- Dades mediques: Grup sanguini, al·lèrgies, malalties cròniques, malalties greus passades
- Historial mèdic: Data de visita o ingrés, lloc, metge, diagnòstic, tractament, evolució, privat

2.4.3 Classificació segons privadesa

La privadesa de les parts d'un expedient queden detallades al flux d'informació, però pel que fa al historial mèdic, s'ha d'afegir que si una entrada de dades es introduïda com a privada, significa que nomes el metge que les ha inserit, pot consultar-les. Això nomes passaria en el cas dels Psicòlegs, en que la seva informació es confidencial, i només ells podran posar la informació com a privada.

2.4.4 Flux d'informació

El flux d'informació ens detalla la relació de permisos entre els actors i les dades o les diferents parts d'un expedient. A la Taula 2. Flux d'informació veiem aquest flux d'una manera molt esquematitzada.

(L: Lectura, I: Inserció, M: Modificació)

Expedient	Gestor	Metge	Pacient
Dades Personals	L/I/M	L	L
Dades Contacte		L	L/I/M
Dades Mediques		L/I	L
Historial Mèdic		L/I	L

Taula 2. Flux d'informació

2.5 Requisits de seguretat

S'han de garantir els següents requisits per cada servei del sistema:

- Autenticació d'usuaris: S'ha de garantir que qui es *logina* al sistema, es realment la persona que s'identifica i se li apliquen el permisos adients.

- Consulta d'un expedient: S'ha de preservar la confidencialitat de les dades de l'expedient del pacient, garantir que les anotacions del metge han estat realitzades per ell (autenticitat) i no han estat modificades (integritat). A més el metge no podrà dir que no ha introduït alguna dada (no-repudi).

- Llistat dels pacients: S'ha de garantir la confidencialitat i autenticitat de les dades que rep el metge.

- Inserció de dades: La informació que es guarda en el sistema ha de disposar d'una prova de la seva autenticitat i un cop generada en el sistema s'ha de garantir en tot moment la seva integritat. A més s'ha de registrar qui ha fet la inserció per després poder identificar-ho. Les dades es guardaran de forma confidencial.

3 IAIK i PKI

3.1 IAIK-JCE

Es una llibreria criptogràfica, que implementa un conjunt l'APIs i implementació de múltiples algoritmes criptogràfics junt amb mètodes per la seva gestió, tot això sota la filosofia de la programació orientada a objectes. Aquesta llibreria està realitzada per ser utilitzada dintre d'aplicacions Java i ser un proveïdor de les funcions criptogràfiques per les classes de Java que treballen amb la seguretat de la informació en les versions Java 1.2 y següents.

Entre els múltiples algoritmes que IAIK implementa, s'inclouen els orientats a la generació de claus, simètrica i asimètriques. Codificació i descodificació de Dades, signatura i verificació d'informació, funcions hash, funcions resum, creació de certificats, etc. Entre aquests algoritmes i tècniques podem trobar RSH, SHA, DES, tiple DES, MD2, MD5, certificats X509 en versió 1,2 i 3, IDEA, etc.

En el nostre projecte, s'ha utilitzat aquesta llibreria per la codificació/descodificació i signatura de dades. Per la generació de claus i certificats, s'ha utilitzat OpenSSL que s'explicarà en una secció posterior.

3.2 Passos per Instal·lar IAIK

Els passos per instal·lar IAIK son els següents:

1. Descarregar la última versió del JDK de SUN i instal·lar-lo. Cal vigilar que no s'utilitzi la màquina virtual de MS al compilar o executar
2. Descarregar la última versió de IAIK. Cal registrar-se però no suposa cap cost, doncs es per ús educatiu. Es pot baixar únicament l'arxiu iaik_jec_full.jar que es la última versió completa signada.
3. Descarregar les polítiques de seguretat de Java que permeten emprar qualsevol longitud de clau, Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0 RC
4. Copia l'arxiu iaik_jce_full.jar als directoris:
    a. c:\Archivos de Programa\Java\jdk1.5.0\jre\lib\ext
    b. c:\Archivos de Programa\Java\jre1.5.0\lib\ext
5. Dins de l'arxiu jce_policy-1.5.0-beta2.zip hi ha els arxius:
    a. Local_policy.jar
    b. US_export_policy.jar
    c. Copiar-los a
        i. c:\Archivos de Programa\Java\jdk1.5.0\jre\lib\security
        ii. c:\Archivos de Programa\Java\jre1.5.0\lib\security
6. Compilar i executar

3.3 PKI i OpenSSL

3.3.1 Introducció

La criptografia de clau pública permet el intercanvi de missatges confidencials i íntegres de manera àgil, sempre que disposem de la clau pública del interlocutor amb qui ens comuniquem. El problema és com obtenir aquesta clau pública i poder estar segurs que pertany a qui ens pensem.

Les claus públiques són justament això: públiques; i un intrús pro hàbil podria tenir accés al directori on estan ubicades i substituir la clau pública d'algun usuari per la seva pròpia. D'aquesta manera, els interlocutors que utilitzessin la clau pensarien que s'estan comunicant amb aquell usuari quan, en realitat, ho farien amb el intrús. És el que es coneix com *atac de l'home a mig camí*.

Després de diferents solucions per resoldre aquest problema de seguretat, L. Kohnfelder (1978) es va basar en la idea d'una autoritat centra de confiança introduïda per Diffie i Hellman i va proposar crear uns registres de dades signades –els certificats- que permetrien que la distribució de claus es fes des de directoris públics que no requerissin confiança.

La signatura, realitzada per un usuari o entitat externa lleial, assegura la integritat contra una possible modificació no desitjada de les dades. La confiança en el signatari s'estén al subjecte del certificat.

El certificat digital és una estructura de dades que conté informació del propietari de les claus criptogràfiques, la clau pública en si i una signatura digital dels dos camps anteriors que hi dona validesa.

De tota manera, l'ús de certificats no resol totalment el problema de la distribució de claus, sinó que la trasllada a un nivell superior. Qui ha de signar el certificat? Quins mecanismes són necessaris perquè dos usuaris que no es coneixen puguin assegurar la seva identitat en una comunicació virtual? Quin és el model de confiança?

3.3.2 Propòsits d'una infraestructura de clau pública (PKI)

L'objectiu d'una PKI és la gestió eficient i confiable de les claus criptogràfiques i els certificats perquè es puguin utilitzar per a funcions d'autenticació, integritat, no-repudi i confidencialitat. La PKI crea un marc segur d'intercanvi de dades en un entorn típicament insegur com Internet.

El certificat és l'element central de la PKI al voltant del qual es crea aquesta infraestructura de suport que abraça serveis com el registre d'usuaris, l'emissió de certificats, la seva distribució des de directoris públics, la seva renovació i revocació, la recuperació de claus, etc.

3.3.3 Components d'una PKI

Encara que els components d'una PKI son l'autoritat de certificació, els subscriptors, els repositoris, l'autoritat de registre, l'autoritat de validació i l'autoritat de segellat de temps entre d'altres, només donaré una breu explicació dels més essencials.

Autoritat de Certificació

L'autoritat de certificació (CA) és la responsable d'emetre i revocar certificats. És l'entitat de confiança que dóna legitimitat a la relació d'una clau pública amb la identitat d'un usuari o servei.

Una PKI pot tenir una o més autoritats de certificació. La creació d'una autoritat de certificació comença amb la generació del parell de claus (publica i privada) que s'utilitzaran per a signar i validar els certificats digitals que emeti l'autoritat de certificació. Les claus han de ser prou fortes perquè la probabilitat que un atacant les trenqui sigui extremada durant el temps de vida dels certificats que s'hi signaran. Això dependrà de la combinació de la longitud de la clau i la qualitat de l'algorisme de generació de claus.

Un cop generat el parell de claus, la clau públic s'ha de distribuir d'una manera segura a totes les entitats de confiança potencials. Aquesta distribució es pot fer tant per mitjà d'un certificat digital emès per una autoritat de certificació en la qual els usuaris ja

confien, o via un certificat generat per la pròpia CA que s'acabi de crear (aquests tipus de certificats reben el nom de *certificats autosignats*). Per a garantir la seguretat de la transmissió, aquests certificats digitals de les autoritats de certificació s'hauran de distribuir des d'un canal fora de banda (caldrà passar el certificat físicament, per correu, incorporat en el programari, etc)

L'autoritat de certificació també ha de protegir la seva clau privada d'un ús no autoritzat i la seva seguritat garanteix la seguretat d la PKI. És per això que el control d'accés a les claus de la CA sigui molt estricte, per això, normalment, les claus es guarden en una targeta o maquinari criptogràfic i es fa ús de la protecció física que representa tenir la CA aïllada i sense cap connexió de xarxa.

Els subscriptors i les Entitats Finals

Els subscriptors i les entitats finals són aquells que posseeixen un parell de claus (pública i privada) i un certificat associat a la clau pública. Amb aquest parell de claus podran efectuar signatures digitals i xifrar i desxifrar documents. Una entitat final representa un organisme, mentre que un subscriptor és una persona.

Els repositoris

Els repositoris son les estructures encarregades d'emmagatzemar la informació relativa a la infraestructura de clau pública. Els dos repositoris més importants en una PKI són el repositori de certificats i el repositori de llistes de renovació de certificats (CRL) que inclou tots aquells certificats que per diversos motius són invàlids abans de la data de caducitat establerta en el mateix certificat.

El tipus de repositoris més utilitzats en la PKI són els directoris. Un directori és la base de dades especialitzada en la qual s'emmagatzema informació tipificada i organitzada sobre objectes. Està optimitzat per els accessos de lectura, les navegacions i les grans cerques, i el seu objectiu és donar respostes ràpides a un alt volum de peticions.

3.3.4 OpenSSL

OpenSSL es un projecte d'esforç col·laboratiu per desenvolupar una llibreria que implementi els protocols SSL(Secure Sockets Layer) v2/v3 i TLS(Transport Layer Security) v1, i unes funcions criptogràfiques generals, essent robusta, de nivell comercial i amb totes les característiques necessàries i de codi obert. En el projecte s'ha utilitzat (com s'ha dit en capítols anteriors) la llibreria de lliure distribució OpenSSL per tal de construir de forma ràpida i senzilla una petita PKI.

3.4 Certificats

3.4.1 Introducció

Els certificats que s'han generat tenen un format X.509, que es el que més àmpliament s'ha acceptat en la infraestructura de claus públiques i que està definit per l'ISO/IEC JTC1 SC21.

Els certificats X.509, com la resta de components de la PKI, es designen per un nom distingit. Un nom distingit (DN) és un conjunt d'atributs amb valors associats. Aquest tipus d'estructura permet que cada organització decideixi quins atributs són interessants per a identificar les entitats dins de la seva empresa.

Per a alguns usos del certificat, el sistema de noms per noms distingits pot resultar insuficient. En aquests casos es fan servir uns camps especials del certificat anomenats *noms alternatius* que permeten especificar l'adreça de correu electrònic, l'adreça IP d'una web, el DNS, etc.

Per aquests projecte s'ha utilitzat aquests atributs (veure Taula 3. Atributs utilitzats pels noms distingit dels certificats) per identificar al usuari (metge o pacient) i aconseguir algunes dades necessàries, com el seu identificador o a quin rol pertany (metges o pacients).

Atribut	Descripció	Significat al projecte
dnQualifier	Identificador	Identificador usuari (NIF)
CN	Nom comú	Nom i Cognoms
OU	Unitat Organitzativa	Rol (Gestio, Metges, Pacients)
O	Organització	UOC
C	País	ES
L	Localitat	Barcelona
S	Estat	Barcelona

Taula 3. Atributs utilitzats pels noms distingit dels certificats

Als certificats hi ha tres tipus principals de camps:

- **Bàsics**: Camps que aporten informació sobre l'autoritat de certificació que ha emès el certificat, l'entitat/subscriptor a què pertany la clau pública, la mateixa clau pública, i el període de validesa i la identificació del certificat.
- **Necessaris per a la signatura**: Camps que utilitzarà qui rebi el certificat per a comprovar que el document està signat correctament.
- **Ampliacions**: Camps que ha aparegut per a cobrir les noves necessitats d'atributs en un certificat. Alguns del possibles usos dels camps d'ampliació són la millora de la gestió de l'herència de certificació, les polítiques de seguretat o les dades sobre l'usuari i la seva clau. Les ampliacions poden estar definides en estàndards o per una organització particular que faci ús de certificats.

3.4.2 Generació de Certificats

En aquesta secció veurem com s'han generat els certificats necessaris pel projecte, seguint els passos necessaris per la construcció d'una petita PKI mitjançant OpenSSL. Per comoditat i facilitat d'execució, s'ha utilitzat el mateix password pet totes les claus i certificats "uoc0506". A més s'ha utilitzat un fitxer de configuració *openssl.conf* per determinar paràmetres per defecte a la generació dels certificats, com els noms distingits. Aquest nom distingit es pot modificar a l'hora de la creació.

Quan un usuari vol obtenir un certificat normalment realitza els passos següents. En un primer pas crea una parella de claus i realitza una petició de certificat mitjançant una Autoritat de Registre (RA). La RA valida la identitat de l'usuari que ha demanat el certificat i envia la petició a la CA. La CA rep les peticions de les RA i emet els certificats. La clau privada de la CA és un peça d'informació molt sensible, com ja s'ha comentat, i es per això que està en un entorn amb un alt nivell de seguretat.

Certificat CA

1. Generar una parella de claus per la CA
   a. `openssl genrsa -des3 -out CA.key 2048`

2. Generar i verificar un certificat autosignat amb la parella de claus de la CA, CA.key. Aquest serà el certificat de la CA
   a. `openssl req -new -sha1 -x509 -key CA.key -out CA.crt -days 360`
   b. `openssl verify -CAfile CA.crt CA.crt`

Certificat Gestor

1. Generar una parella de claus pel Gestor
   a. `openssl genrsa -des3 -out Gestor.key 1024`

2. Generar i verificar una petició de certificat pel Gestor
   a. `openssl req -new -sha1 -key Gestor.key -out Gestor.csr -config openssl.conf`
   b. `openssl req -in Gestor.csr -verify -text –noout`

3. Generar i verificar un certificat pel Gestor a partir de la petició
   a. `openssl x509 -req -in Gestor.csr -days 180 -CA CA.crt -CAkey CA.key -CAcreateserial -extfile openssl.conf -extensions usr_cert -out Gestor.crt`
   b. `openssl verify -CAfile CA.crt Gestor.crt`

4. Generar el fitxer que conté la clau privada i el certificat del Gestor en format PKCS#12
   a. `openssl pkcs12 -in Gestor.crt -inkey Gestor.key -name Gestor -chain -CAfile CA.crt -export -out Gestor.p12`

Certificats Pacients

1. Generar una parella de claus pel Pacient
   a. `penssl genrsa -des3 -out Pacient-44183031p.key 1024`

2. Generar i verificar una petició de certificat pel Pacient
   a. `openssl req -new -sha1 -key Pacient-44183031p.key -out Pacient-44183031p.csr -config openssl.conf`
   b. `openssl req -in Pacient-44183031p.csr -verify -text –noout`

3. Generar i verificar un certificat pel Pacient a partir de la petició
   a. `openssl x509 -req -in Pacient-44183031p.csr -days 180 -CA CA.crt -CAkey CA.key -CAcreateserial -extfile openssl.conf -extensions usr_cert -out Pacient-44183031p.crt`
   b. `openssl verify -CAfile CA.crt Pacient-44183031p.crt`

4. Generar el fitxer que conté la clau privada i el certificat del Pacient en format PKCS#12
   a. `openssl pkcs12 -in Pacient-44183031p.crt -inkey Pacient-44183031p.key -name Pacient -chain -CAfile CA.crt -export -out Pacient-44183031p.p12`

Certificat Metge

1. Generar una parella de claus pel Metge
   a. `openssl genrsa -des3 -out Metge-43433920f.key 1024`

2. Generar i verificar una petició de certificat pel Metge
   a. `openssl req -new -sha1 -key Metge-43433920f.key -out Metge-43433920f.csr -config openssl.conf`
   b. `openssl req -in Metge-43433920f.csr -verify -text –noout`

3. Generar i verificar un certificat pel Metge a partir de la petició
   a. `openssl x509 -req -in Metge-43433920f.csr -days 180 -CA CA.crt -CAkey CA.key -CAcreateserial -extfile openssl.conf -extensions usr_cert -out Metge-43433920f.crt`
   b. `openssl verify -CAfile CA.crt Metge-43433920f.crt`

4. Generar el fitxer que conté la clau privada i el certificat del Metge en format PKCS#12
   a. `openssl pkcs12 -in Metge-43433920f.crt -inkey Metge-43433920f.key -name Metge -chain -CAfile CA.crt -export -out Metge-43433920f.p12`

4 Criptografia

4.1 Introducció

A partir de l'estudi realitzat prèviament (veure capítol 1) s'ha dissenyat un protocol criptogràfic per cada acció que realitzen els usuaris, ja siguin els pacients o els metges.

Els protocols criptogràfics garanteixen les propietats que cal garantir per cada acció. Aquestes propietats es varen descriure a l'estudi inicial (veure secció 1.4).

Pel que fa a l'autenticació dels usuaris es realitzarà una única vegada al entrar a l'aplicació, és guardarà un identificador a la Base de Dades per poder identificar-los a cada petició de servei.

A continuació es presenten els diferents protocols implementats per aquest PFC.

4.2 Notació

En la descripció dels protocols s'empra la notació següent:

- *K*: Clau d'un criptosistema simètric
- $E_k(M)$: Xifratge simètric d'un missatge M amb la clau K.
- $D_k(C)$: Desxifratge simètric del criptograma C amb la clau K.
- $(P_{Entitat}, S_{Entitat})$: Parella de claus asimètriques propietat d'Entitat, on P correspon a la clau pública i S a la privada.
- $S_{Entitat}[M]$: Signatura digital del missatge M amb la clau privada S d'Entitat.
- $P_{Entitat}[M]$: Xifratge del missatge M amb la clau privada S d'Entitat.
- *H(M)*: Sortida d'una funció resum criptogràfica del missatge M, aquestes funcions reben el nom de funcions hash.

4.3 Protocol Autenticació

Per realitzar el protocol d'autenticació, s'ha escollit el protocol de Needham-Schroeder, però amb una petita variació, l'últim pas d'aquest protocol, la última verificació per part del gestor es farà a cada sol·licitud de servei, i així podrem, primer, acabar de verificar l'autenticació i, segon, comprovar quin usuari està demanant la petició, es manté un estat de la connexió.
Cada usuari *U* s'identifica amb $IdUsuari_u$ i disposa d'una parella de claus (P_u, S_u) amb el corresponent certificat $Cert_u$. Aquest protocol pot ser utilitzat per un metge o per un pacient.

1. *U realitza les operacions següents:*
   a. *Executar el Procedure 1 amb la clau pública P_u, i obtenir $P_g[N_i, Id\_usuari_u]$;*
   b. *Enviar $P_g[N_i, Id\_usuari_u]$ a G;*

2. *G realitza les operacions següents:*
   a. *Executar el Procedure 2 amb $P_g[N_i, Id\_usuari_u]$, i obtenir $P_u[N_i, N_g, Id\_usuari_g]$;*
   b. *Enviar $P_u[N_i, N_g, Id\_usuari_g]$ a U;*

3. *U realitza les operacions següents:*
   a. *Desxifrar $P_u[N_i, N_g, Id\_usuari_g]$ amb la clau privada S_u, i obtenir N_g, N_i' i Id_usuari_g;*
   b. *Si $N_i' =^{?} N_i$ fer:*
      i. *L'Usuari U queda parcialment autenticat*
   c. *Sino retornar error;*

4.3.1 Procedure 1 (P_g)

El Procedure 1 conté una part de l'autenticació del protocol de Needham-Schroeder. Aquest procedure serà utilitzat pels metges i pacients.

1. *Obtenir un valor de forma aleatòria, N_i;*
2. *Xifrar N_i i Id_usuari_u amb la clau pública de G, $P_g[N_i,Id\_usuari_u]$;*
3. *Enviar $P_g[N_i,Id\_usuari_u]$ a G,*

4.3.2 Procedure 2 ($P_g(N_i,Id\_usuari_u)$)

El Procedure 2 conté una altra part de l'autenticació del protocol de Needham-Schroeder. Aquesta part serà executada pel Gestor.

1. *Desxifrar $P_g[N_i,Id\_usuari_u]$ amb S_g, i obtenir; N_i i Id_usuari_u;*
2. *Obtenir el certificat de U a partir de Id_usuari_u. El sistema disposa d'una Base de Dades (BD) on per cada Id usuari trobem el seu certificat corresponent. A partir del certificat es pot obtenir la clau pública P_u;*
3. *Obtenir un valor de forma aleatòria, N_g;*
4. *Guardar a la BD els valors N_i i N_g associats amb U; Això servirà més endavant per poder identificar al Usuari que fa la petició i poder validar-lo.*
5. *Xifrar N_i, N_g , Id_usuari_u, amb la clau pública P_u de U, $P_u[N_i,N_g,Id\_usuari_g]$;*
6. *Retornar $P_u[N_i,N_g,Id\_usuari_g]$;*

4.4 Protocol Consulta d'un historial

Cada usuari *U* s'identifica amb $IdUsuari_u$ i disposa d'una parella de claus (P_u, S_u) amb el corresponent certificat $Cert_u$. Aquest protocol pot ser utilitzat per un metge o per un pacient. *G* verifica en cada cas el tipus d'usuari i només facilita el historial si l'usuari té accés.

1. *U realitza les operacions següents:*
   a. *Xifrar N_g, Consulta, Id_usuari amb la clau pública P_g de G, $P_g[N_g,Consulta,Id\_usuari]$. Consulta indica que es vol consulta l'historial de l'usuari identificat amb Id_usuari;*
   b. *Enviar $P_g[N_g,Consulta,Id\_usuari]$ a G;*

2. *G realitza les operacions següents:*
   a. *Desxifrar $P_g[N_g,Consulta,Id\_usuari]$ amb la clau privada S_g, i obtenir N_g',Consulta,Id_usuari;*
   b. *Recuperar Id_usuari_u amb N_g de la BD. En el pas 4 del Procedure 2 N_g i N_i han estat guardats a la BD;*

c. *Si $N_g' =^? N_g$ fer (S'ha trobat N_g a la BD amb N_g'):*
   i. *Si ($Id\_usuari_u =^? Id\_usuari$) o ($Id\_usuari_u$ es metge i Id_usuari és un pacient de Id_usuari_u) fer:*
      1. *Executar el Procedure 3 amb Id_usuari i P_u, i obtenir $P_u[H]$;*
      2. *Enviar $P_u[H]$ a U.*
   ii. *Sino retornar error;*
d. *Sino retornar error;*

3. *U realitza les operacions següents:*
   e. *Executar el Procedure 4 amb $P_u[H]$, i obtenir H;*
   f. *Mostrar H.*

4.4.1 Procedure 3 (*Id_usuari*,P_u)

El gestor G utilitza el Procedure 3 per trobar el historial que se li ha demanat i xifrar-lo amb la clau de l'usuari que el vol consultar.

1. *Buscar el historial H a la BD corresponent a Id_usuari;*
2. *Desxifrar la part de H que està xifrada utilitzant la clau privada S_g de G;*
3. *Comprovar que no hi ha cap salt a les visites;*
4. *Xifrar H amb la clau pública P_u, $P_u[H]$;*
5. *Retornar $P_u[H]$;*

4.4.2 Procedure 4

Un usuari utilitza el Procedure 4 per tal de desxifrar un historial enviat pel gestor G i verificar que el historial és correcte.

1. *Desxifrar $P_u[H]$ amb la clau privada S_u de U, $S_u[P_u[H]]$;*
2. *Per cada entrada de la història H que està signada fer:*
   a. *Verificar la signatura digital de M;*
   b. *Verificar la signatura digital de G;*
   c. *Verificar la seqüència;*
3. *Retornar H;*

4.5 Protocol Consulta dels pacients assignats a un metge

Amb operació típica d'un metge es buscar el historial d'un dels seus pacients. Amb aquest protocol un metge pot obtenir el llistat dels seus pacients. Veureu que només s'envien els identificadors d'usuari. Aquesta és la informació mínima per recuperar un historial. Si ho creieu convenient podeu retornar més informació de manera que es pugui mostrar més informació que el identificador d'usuari.

1. *U realitza les operacions següents:*
    a. *Xifrar N_g i Llista_pacients amb la clau pública P_g de G, $P_g[N_g$, Llista_pacients]. Llista_pacients indica que es vol un llistat dels identificadors d'usuari corresponents als pacients del metge identificat amb Id_usuari_u;*
    b. *Enviar $P_g[N_g$, Llista_pacients] a G;*
2. *G realitza les operacions següents:*
    a. *Desxifrar $P_g[N_g$, Llista_pacients] amb la clau privada S_g,i obtenir N_g' i Llista_pacients;*
    b. *Recuperar Id_usuari_u amb N_g de la BD. En el pas 4 del Procedure 2 N_g i N_i han estat guardats a la BD;*
    c. *Si $N_g' =^? N_g$ fer(S'ha trobat N_g a la BD amb N_g'):*
        i. *Si Id_usuari_u es metge fer:*
            1. *Executar el Procedure 5 amb Id_usuari_u i P_u, obtgenir $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$;*
            2. *Enviar a U $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$;*
        ii. *Si no retornar error;*
    d. *Si no retornar error;*
3. *U realitza les operacions següents:*
    a. *Executar el Procedure 6 amb $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$ i obtenir $\{Id\_usuari_1 , ... , Id\_usuari_n\}$;*
    b. *Mostrar $\{Id\_usuari_1 , ... , Id\_usuari_n\}$;*

4.5.1 Procedure 5 (*Id_usuari*,P_u)

Am el Procedure 5 el gestor G obté el llistat dels pacients assignats al metge Id_usuari.

1. *Cercar a la BD els pacients assignats al metge Id_usuari, obtenint $\{Id\_usuari_1 , ... , Id\_usuari_n\}$;*
2. *Signar $\{Id\_usuari_1 , ... , Id\_usuari_n\}$ amb la clau privada S_g de G, $S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]$;*
3. *Xifrar $\{Id\_usuari_1 , ... , Id\_usuari_n\}$ i $S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]$ amb la clau pública de Id_usuari, P_u, $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$;*
4. *Retornar $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$;*

4.5.2 Procedure 6 ($P_u(\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g(\{Id\_usuari_1 , ... , Id\_usuari_n\}))$)

El metge utilitza el Procedure 6 per obtenir la llista dels seus pacients i verificar que ha estat generada perl gestor G.

1. *Desxifrar $P_u[\{Id\_usuari_1 , ... , Id\_usuari_n\}, S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]$ amb la clau privada S_u de U, $S_u[P_u[S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]]]$ i obtenir $S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]$;*
2. *Verificar la signatura digital $S_g[\{Id\_usuari_1 , ... , Id\_usuari_n\}]$ amb la clau pública P_g de G;*
3. *Si la verificació anterior es correcta retornar $\{Id\_usuari_1 , ... , Id\_usuari_n\}$;*

4.6 Protocol Inserció de dades al historial mèdic

Aquest protocol només afegeix una nova visita V al historial. El gestor un cop rep una visita V d'un pacient P verifica que ha estat signada pel metge M assignat al pacient. A continuació afegeix la visita al historial H. Per garantir que l'ordre de les visites no es modifica s'afegeix a cada visita una marca temporal T i un número de sèrie X. Amb aquestes dades es pot saber l'instant de la visita i l'ordre que han seguit. La visita V, el temps T i el número de sèrie son signats pel gestor G. Si un atacant elimina un registre al mig del historial es detectarà perquè hi haurà un salt en el número a la sèrie de les visites.

La marca temporal T no protegeix el historial de la eliminació d'una visita. L'atacant no podrà refer la seqüència sense la clau privada del gestor G. Suposem que aquesta clau està ben protegida. A continuació el gestor G xifra les dades de la visita am la seva clau pública i ho guarda a la BD. Si un atacant accedeix a la BD no pot veure les dades confidencials. Finalment afegeix al historial una signatura digital de quin és l'últim número de sèrie X del historial H. Si un atacant elimina la última visita es detectarà perquè hi haurà un salt entre la última visita i el número de sèrie X signat.

1. *M realitza les operacions següents:*
    a. *Obtenir les dades de la visita V. La visita inclou Id_usuari_p;*
    b. *Signar V amb la clau privada S_m de M, $S_m[V]$;*
    c. *Xifrar N_g , V i $S_m[V]$ amb la clau pública P_g de G, $P_g[N_g$,Inserir_visita, V, $S_m[V]]$. Inserir visita indica que es vol afegir V al historial del pacient P;*
    d. *Enviar $P_g[N_g$,Inserir_visita, V, $S_m[V]]$ a G;*
2. *G realitza les operacions següents:*
    a. *Desxifrar $P_g[N_g$,Inserir_visita, V, $S_m[V]]$ amb la clau privada S_g i obtenir N_g', Inserir_visita, V, $S_m[V]$;*
    b. *Recuperar Id_usuari_m amb N_g de la BD. En el pas 4 del Procedure 2 N_g i N_i han estat guardats a la BD.*
    c. *Si $N_g' =^? N_g$ fer(S'ha trobat N_g a la BD amb N_g'):*
        i. *Obtenir Id_usuari_p a partir de V;*
        ii. *Verificar que Id_usuari_m es metge;*
        iii. *Verificar que Id_usuari_p es un pacient assignat a Id_usuari_m;*
        iv. *Si les verificacions anteriors son correctes fer:*
            1. *Verificar la signatura digital $S_m[V]$ amb la clau pública P_m;*
            2. *Obtenir l'instant de temps actual T;*
            3. *Obtenir el número de sèrie X de la última visita del historial H del pacient Id_usuari_p;*
            4. *Verificar la signatura $S_g[X, Id\_usuari_p]$;*
            5. *Incrementar en una unitat X, X+1;*
            6. *Signar V, $S_m[V]$, T, X+1, amb la clau privada S_g de G, $S_g[V, S_m[V], T, X+1]$;*
            7. *Xifrar V i $S_m[V]$ amb la clau pública P_g de G, $P_g[V, S_m[V]]$;*
            8. *Signar Id_usuari_p i X+1 amb la clau privada S_g de G, $S_g[X+1, Id\_usuari_p]$;*
            9. *Guardar a la BD $P_g[V, S_m[V]]$, X+1, T, $S_g[V, S_m[V], T, X+1]$ i $S_g[X+1, Id\_usuari_p]$;*
        v. *Si no retornar error;*
    d. Si no retornar error;

4.7 Protocol Finalitza Sessió

Degut que hem autenticat a l'usuari només un cop al principi de la sessió, es fa necessari finalitzar la sessió, es a dir, esborrar de la DB aquelles dades que permetien al Gestor identificar al usuari que feia una petició. Aquest protocol el pot utilitzar tant un metge com un Pacient.

1. *U realitza les operacions següents:*
   a. *Enviar $P_g[N_i,Id\_usuari_u]$ a G, s'havia xifrat al moment de l'autenticació;*
2. *G realitza les operacions següents:*
   a. *Desxifrar $P_g[N_i,Id\_usuari_u]$ amb S_g, i obtenir; N_i i Id_usuari_u;*
   b. *Esborrar de la BD N_i i N_g guardats en el moment de l'autenticació mitjançant N_i i Id_usuari_u;*
   c. *Si no ho troba a la BD retornar error;*

4.8 Implementació

4.8.1 Casos D'us

A continuació es presenten els diferents casos d'us del sistema. Per simplicitat, s'han obviat algunes classes com CipherManager, SignerManager, XMLDoc, DB, etc., doncs ja existeix la classe Comuns, que fa les peticions necessàries a aquestes classes.

Cas d'us general

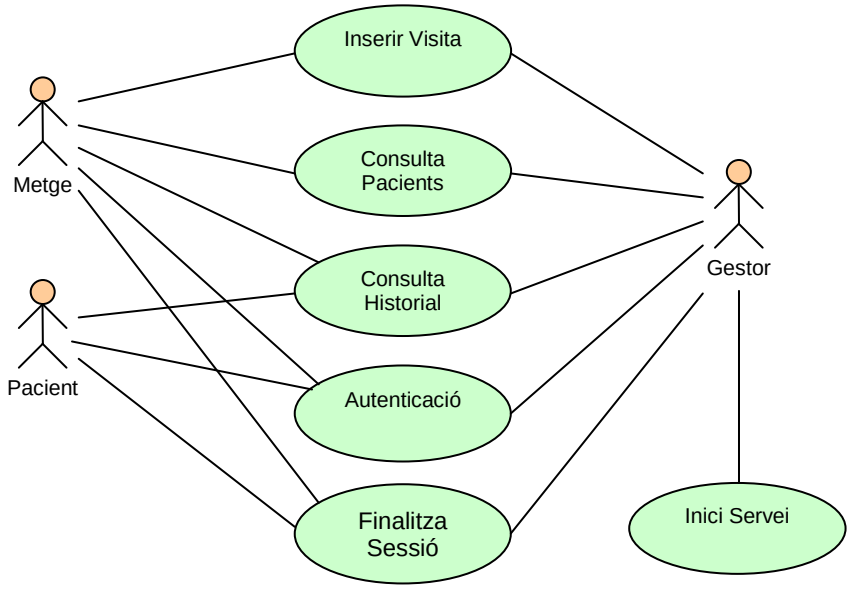


Figura 1: Diagrama Cas d'us General

Cas d'us Inici Servei

Abans de tot, el Gestor ha d'iniciar els seus serveis per poder rebre les peticions del usuaris. En aquest inici carregarà el seu fitxer P12, registrarà el seu servei i romandrà a l'espera. El diagrama de seqüència es presenta a la Figura 2. Diagrama de seqüència del cas d'us Iniciar Servei.

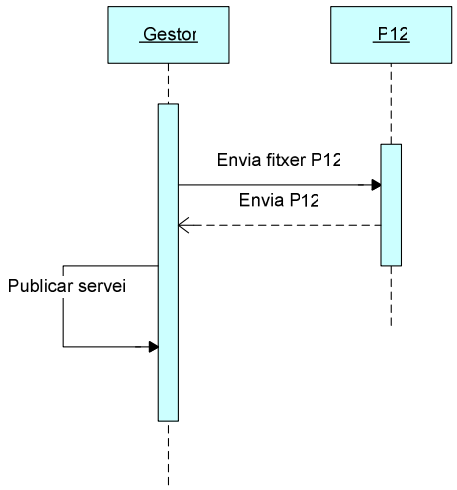


Figura 2. Diagrama de seqüència del cas d'us Iniciar Servei

Cas d'us Autenticació

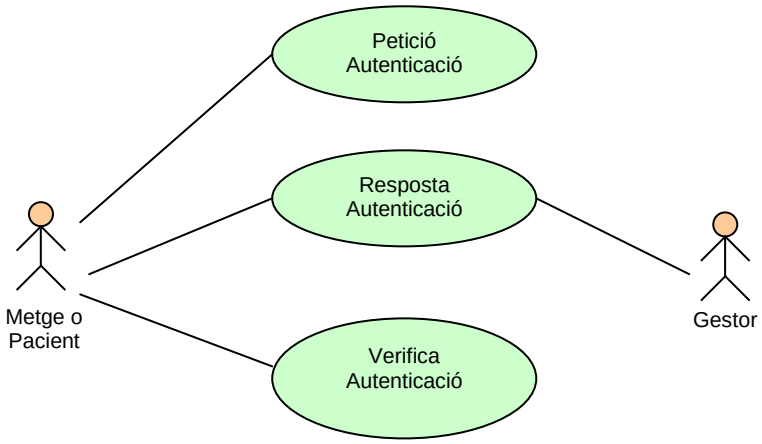


Figura 3. Diagrama Cas d'us Autenticació

Per realitzar l'autenticació el usuari (pacient o metge) com el Gestor han d'estar donats d'alta a l'aplicació i tenir el seu Certificat a la Base de Dades, per després poder desxifrar les dades que s'envien. A més, han de tenir el seu parell de claus privades per poder xifrar les dades. Pel Gestor, aquest parell de claus ja l'ha carregat. En acabar l'autenticació el usuari haurà autenticat al Gestor, però el Gestor encara no haurà fet l'autenticació complerta doncs ho acabarà a cada petició de servei i així podrà identificar als usuaris.

Petició Autenticació: L'usuari envia un valor aleatori i el seu identificador, en un XML xifrat, al Gestor. El diagrama de seqüència es presenta a la Figura 4. Diagrama de seqüència del Cas d'us Petició Autentificació.

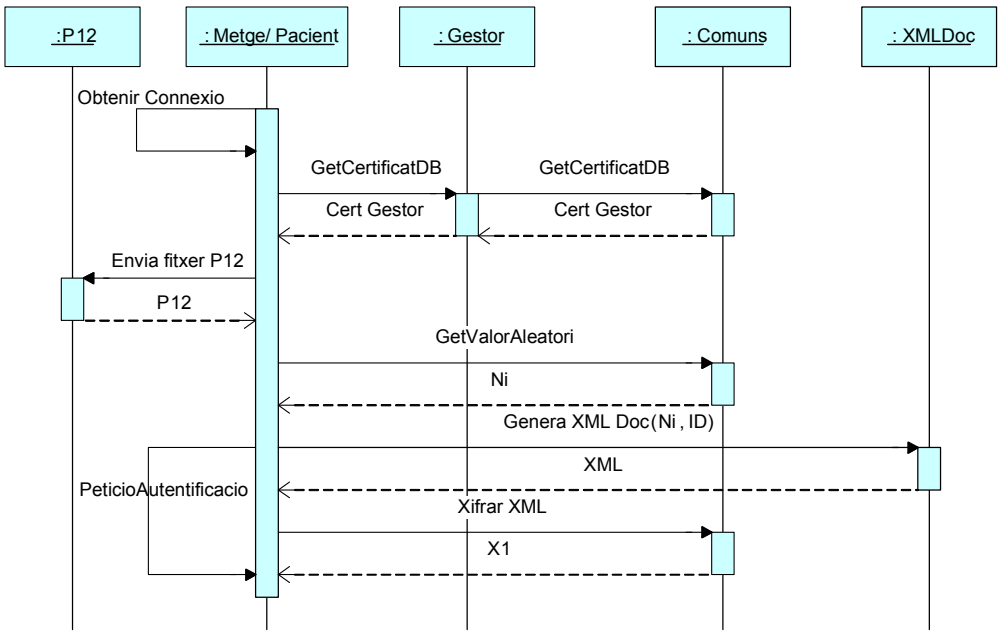


Figura 4. Diagrama de seqüència del Cas d'us Petició Autentificació

Resposta Autenticació: Un cop el gestor a rebut les dades, primer de tot les desxifra, les transforma en un XML i obté el identificador del usuari i Ni. Obte el certificat del usuari a la Base de dades i genera el seu numero aleatori Ng. Genera un XML amb Ng, Ni i el identificador del usuari, ho xifra i ho envia al usuari. El diagrama de seqüència es presenta a la Figura 5. Diagrama de seqüència cas d'us Resposta Autenticació.

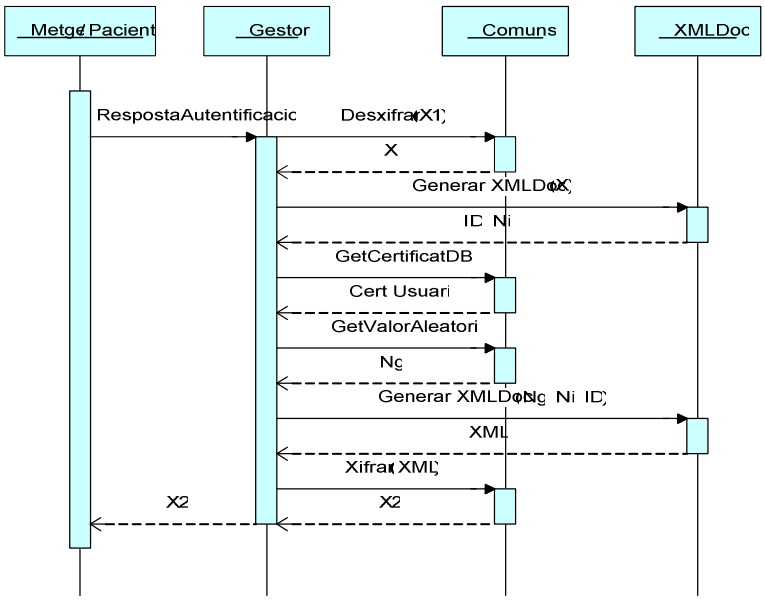


Figura 5. Diagrama de seqüència cas d'us Resposta Autenticació

Verificació Autenticació: Un cop el gestor a rebut les dades, primer de tot les desxifra, les transforma en un XML i obté el identificador del usuari i Ni. Obte el certificat del usuari a la Base de dades i genera el seu numero aleatori Ng. Genera un XML amb Ng, Ni i el identificador del usuari, ho xifra i ho envia al usuari. El diagrama de seqüència es presenta a la Figura 6. Diagrama de seqüència cas d'us Verificació Autentificació

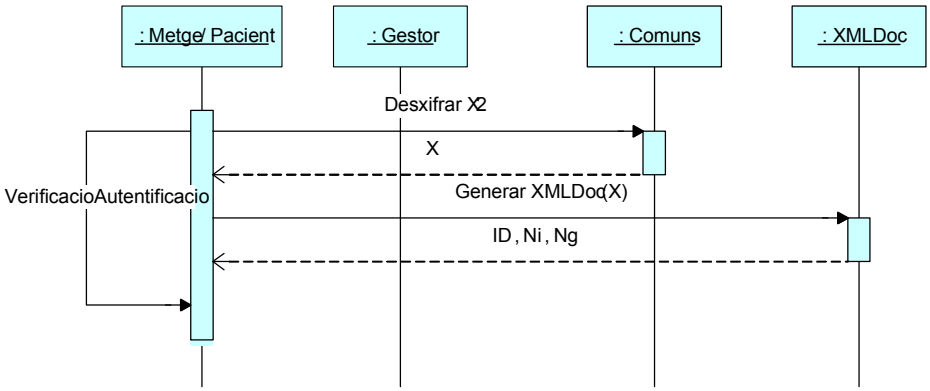


Figura 6. Diagrama de seqüència cas d'us Verificació Autentificació

Cas d'us Finalitza Sessió

Figura 7. Diagrama del Cas d'us Finalitza Sessió

Quan un usuari a acabat la sessió, avisa al Gestor i aquest s'encarrega d'eliminar la seva sessió. El diagrama de seqüència es troba a la Figura 8. Diagrama de seqüència del cas d'us Finalitza Sessió.

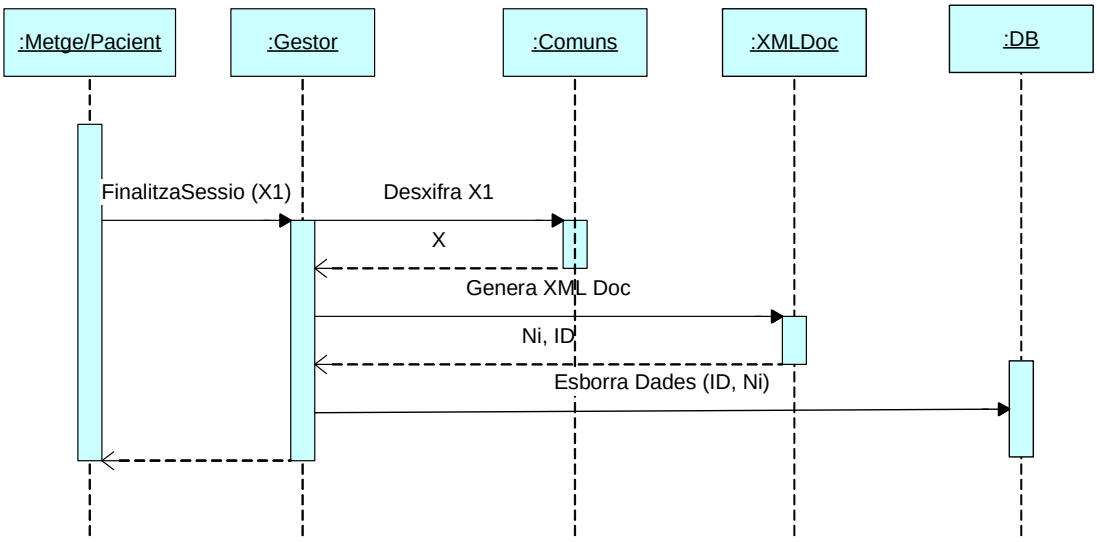


Figura 8. Diagrama de seqüència del cas d'us Finalitza Sessió

Cas d'us Consulta Historial

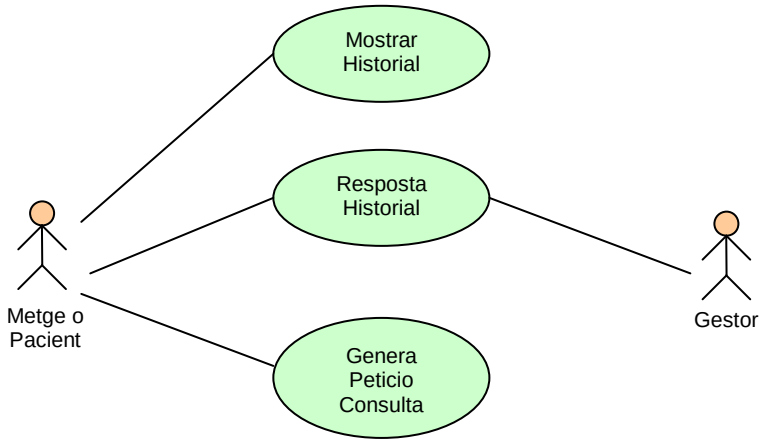


Figura 9. Cas d'us Consulta Historial

El usuari prepara les dades per demanar la petició de consulta del historial al Gestor. En aquest moment, el Gestor acaba d'autentificar al usuari i aconsegueix el seu identificador. Agafarà el historial i li enviarà a l'usuari. Un cop rebudes les dades, les mostrarà per pantalla.

Genera Petició Consulta: El usuari amb el Ng obtingut de l'autenticació, genera un XML amb el identificador del pacient, ell mateix en el cas que sigui un pacient qui faci la consulta, el Ng i la petició que farà, "Consulta" en aquest cas. Aquest XML el xifrarà en el certificat del Gestor i li enviarà. El diagrama de seqüència es troba a la Figura 10. Diagrama de seqüència del cas d'us Genera Petició Consulta.

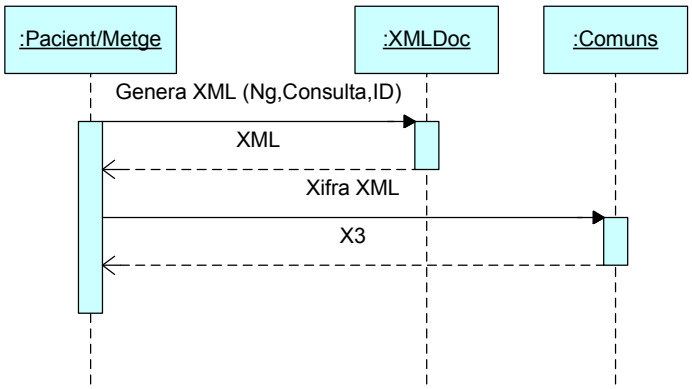


Figura 10. Diagrama de seqüència del cas d'us Genera Petició Consulta

Resposta Historial: El Gestor rep les dades amb la i les desxifra amb la seva clau privada. Obté Ng i ho busca a la DB per obtenir el id del usuari. Si coincideixen comprova que el id obtingut, bé sigui l'usuari que ho demana o un metge assignat a ell. Si es així, obté el historial del Pacient de la Base de Dades, desxifra amb la seva clau les dades xifrades (les visites) del historial, comprova que no hi hagi cap salt i xifra tot el historial amb les visites desxifrades afegides per enviar-li a l'usuari que ho ha demanat. El diagrama de seqüència es troba a la Figura 11. Diagrama de seqüència del cas d'ús Resposta Historial.

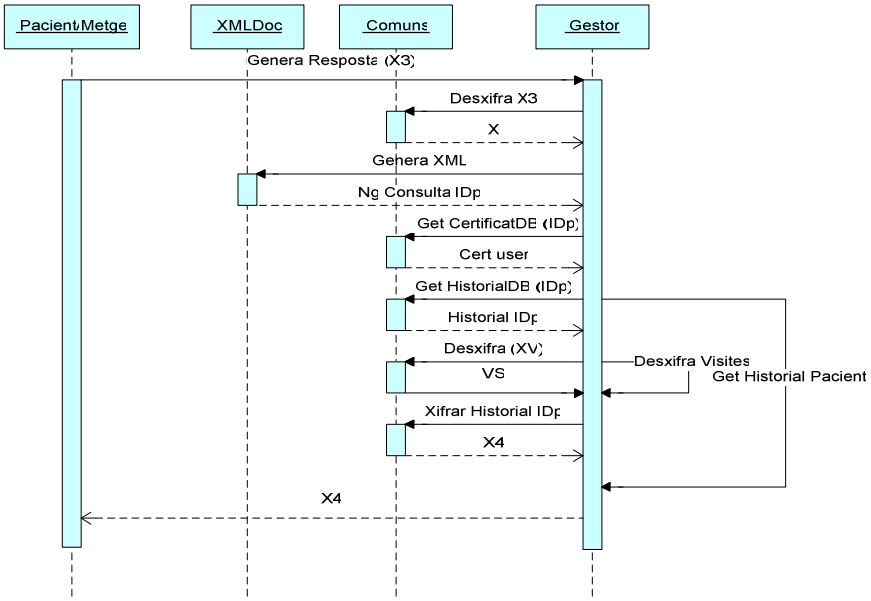


Figura 11. Diagrama de seqüència del cas d'ús Resposta Historial

Mostra Historial: L'usuari un cop rep les dades del Gestor les desxifra amb el certificat del Gestor i genera el XML corresponent. Verifica totes les signatures del metge i del gestor per cada visita comprovant que les dades no s'hagin modificat. Finalment comprova que no falta la última visita i mostra el Historial per pantalla. El diagrama de seqüència es troba a la Figura 12. Diagrama de seqüència del cas d'us Mostrar Historial.

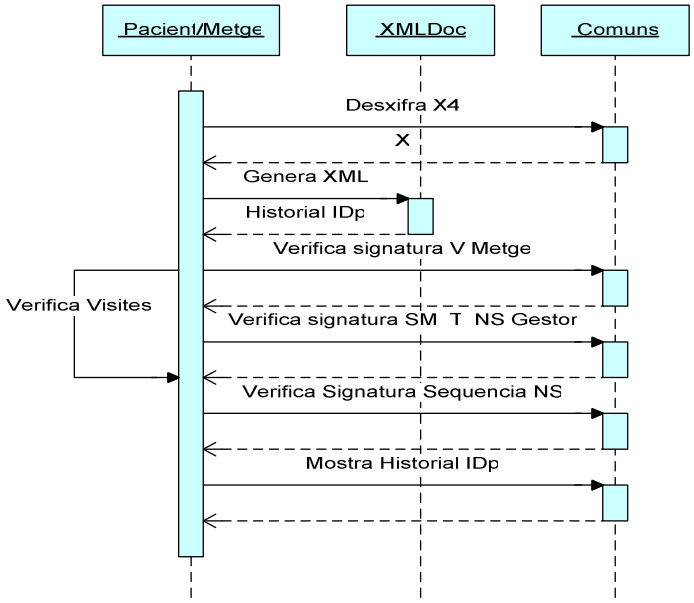


Figura 12. Diagrama de seqüència del cas d'us Mostrar Historial

Cas d'us Consulta Pacients

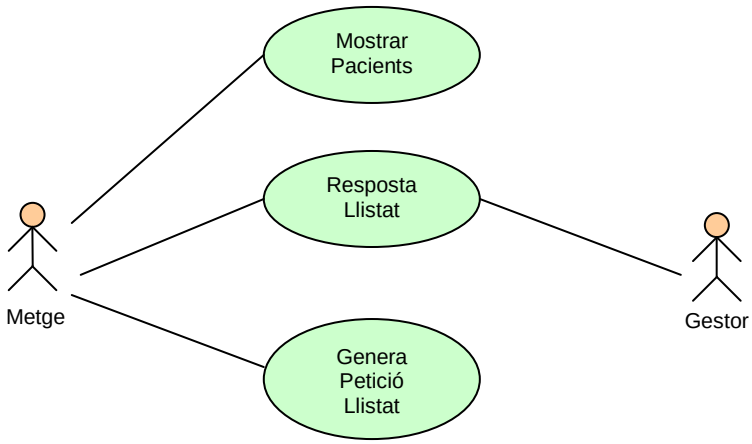


Figura 13. Diagrama Cas d'us Consulta Pacients

Aquest Cas d'us només el pot iniciar un Metge. La seqüència es gairebé la mateixa però en aquest cas el metge demana una petició de llistat dels seus pacients. El Gestor acaba d'autenticar de qui es tracta i li enviarà el llistat dels seus pacients.

Genera Petició Llistat: El Metge amb el Ng obtingut de l'autenticació, genera un XML amb el Ng i la petició que farà, "Llista_pacients" en aquest cas. Aquest XML el xifrarà amb el certificat del Gestor i li enviarà. El diagrama de seqüència es troba a la Figura 14. Diagrama de seqüència del cas d'us Genera Petició Llistat.

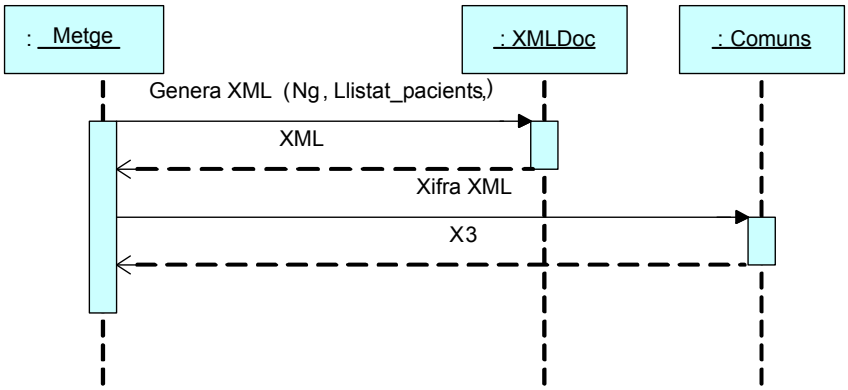


Figura 14. Diagrama de seqüència del cas d'us Genera Petició Llistat

Resposta Llistat: El Gestor rep les dades amb la petició i les desxifra amb la seva clau privada. Obté Ng i ho busca a la DB per obtenir el id del usuari. Si coincideixen comprova que el id obtingut, un metge. Si es així, obté el llistat de la Base de Dades, on traurà el identificador, el nom i els cognoms de cada pacient, el signa i xifra per enviar-li al metge que ho ha demanat. El diagrama de seqüència es troba a la Figura 15. Diagrama de seqüència del cas d'us Resposta Llistat.

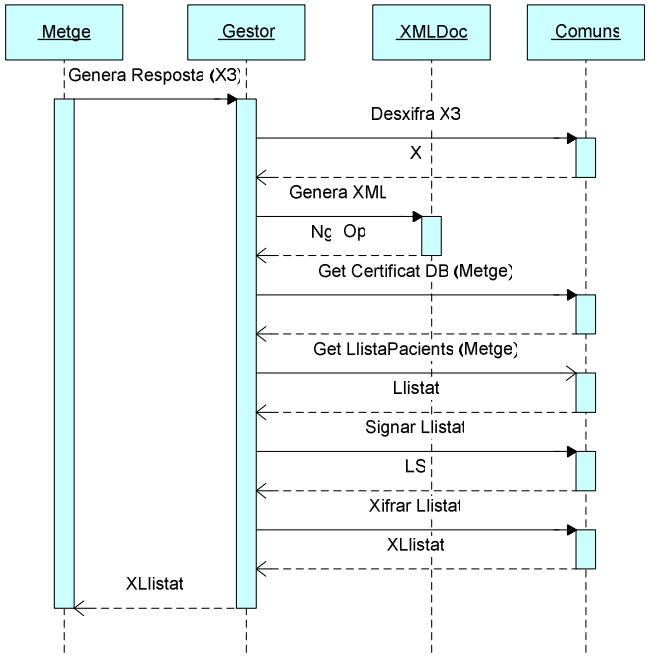


Figura 15. Diagrama de seqüència del cas d'us Resposta Llistat

Mostra Llistat: L'usuari un cop rep les dades del Gestor les desxifra amb el certificat del Gestor i genera el XML corresponent. Verifica la signatura del gestor i mostra el llistat de pacients per pantalla. El diagrama de seqüència es troba a la Figura 16. Diagrama de seqüència del cas d'us Mostra Llistat.

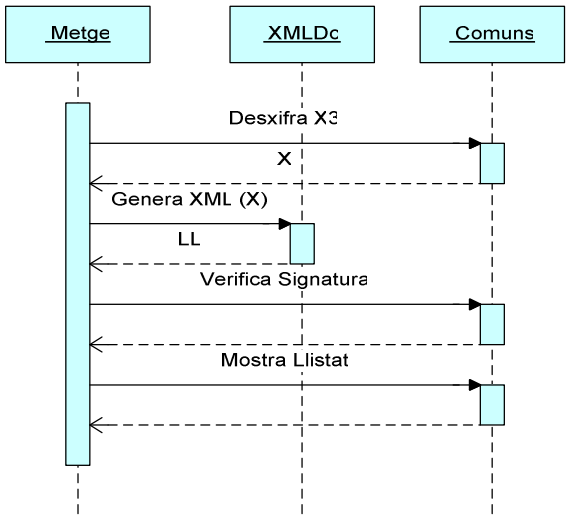


Figura 16. Diagrama de seqüència del cas d'us Mostra Llistat

Cas d'us Inserir Visita

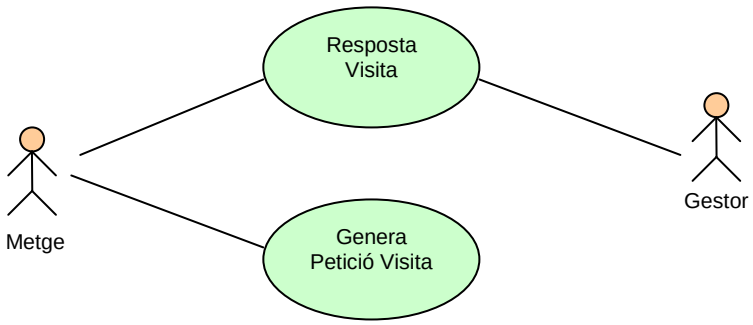


Figura 17. Diagrama del cas d'us Inserir Visita

Aquest Cas d'us només el pot iniciar un Metge. El metge demana una petició de inserció d'una visita a un dels seus pacients. El Gestor acaba d'autenticar de qui es tracta i insereix la visita al historial del pacient.

Genera Petició Visita: El Metge, una vegada ha obtingut anteriorment un Historial d'un pacient, pot inserir una visita. Genera les dades de la visita i les signa. Amb el Ng obtingut de l'autenticació, genera un XML amb el Ng, la visita, la signatura i la petició que farà, "inserir_visita" en aquest cas. Aquest XML el xifrarà amb el certificat del Gestor i li enviarà. El diagrama de seqüència es troba a la Figura 18. Diagrama de seqüència del cas d'us Genera Petició Visita.

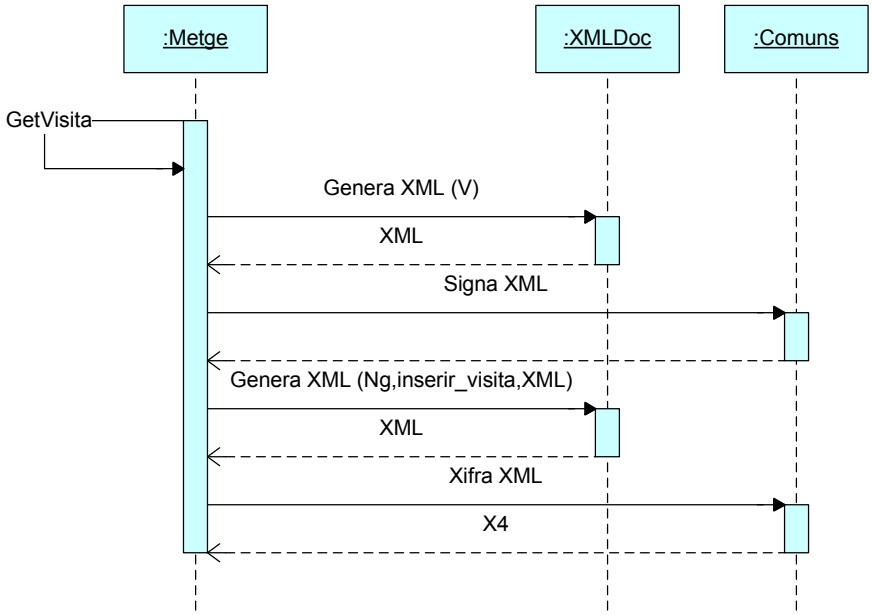


Figura 18. Diagrama de seqüència del cas d'us Genera Petició Visita

Resposta Visita: El Gestor rep les dades amb la petició i les desxifra amb la seva clau privada. Obté Ng i ho busca a la DB per obtenir el id del usuari. Si coincideixen comprova que el id obtingut, es un metge. També obté una visita i la seva signatura i comprova que el identificador de la visita es pacient del metge. Si es així, primer verificarà la signatura de la visita obtinguda. Obtindrà de la DB el historial del pacient i verificarà que el numero de visites (NS) que hi ha es el que hauria d'haver amb la signatura del historial signseq. Incrementarà en 1 el valor NS i generarà un TimeStamp T per la nova visita. Signarà la visita V rebuda, la seva signatura SM, T i NS i obtindrà la signatura SG. Xifrarà V i SM i signarà el identificador del pacient IDp amb el num. Visites NS, obtenint una nova signseq. Finalment ho guardarà a la DB com una nova visita (IDp, NS, XV, SG, T) i també la signatura signseq del pacient. El diagrama de seqüència es troba a la Figura 19. Diagrama de seqüència del cas d'us Resposta Visita.

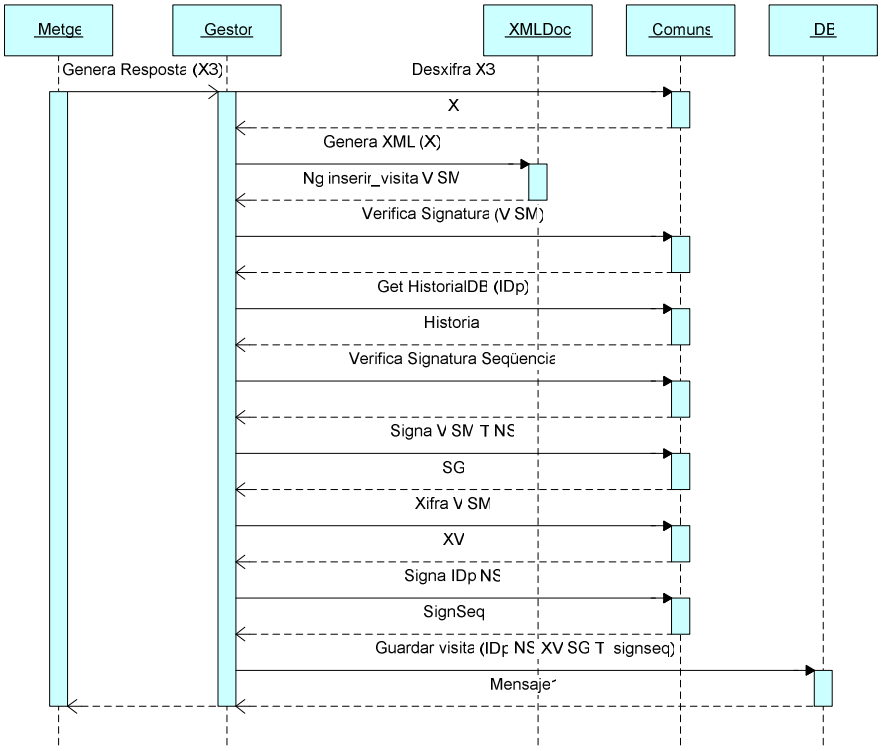


Figura 19. Diagrama de seqüència del cas d'us Resposta Visita

4.8.2 Disseny UML

A la Figura 20. Diagrama de Classes veiem el disseny UML del sistema. Primer de tot cal destacar que hi ha dues classes principals: Client i Gestor y les altres donen diferents utilitats a aquestes classes. Seguidament, es comenta cadascuna d'elles:

- Gestor: Es el servidor de l'aplicació i l'encarregat de donar resposta de forma segura als serveis de:
  - Consulta Historial
  - Consulta Llistat Pacients
  - Inserir Visita
  - Autenticació
- Client: El client es l'aplicació que executaran els usuaris del sistema, ja siguin Metges o Pacients. Aquesta classe serà l'encarregada de fer totes les peticions als serveis anteriorment comentats. Serà la que es comunicarà amb el Gestor mitjançant una connexió RMI.
- Comuns: Aquesta classe es utilitzada tant pel Client com el Gestor i es una classe que ofereix tot tipus d'utilitats a més de ser el pont entre aquestes classes i la clase SignerManager i CipherManager.
- SignerManager: Classe encarregada de realitzar signatures de dades. Ofereix tant la signatura com la seva verificació.
- CipherManager: Classe encarregada de xifrar i desxifrar dades.

- P12: Classe encarregada de generar un PKCS#12 i oferir mètodes per accedir i manipular els magatzems de dades PCKCS#12, que contenen el parell de claus i el certificat del gestor i usuaris.
- DB: Es la classe que dona accés a la Base de dades, oferint mètodes per connexió i desconnexió, inserció, consulta i updates.
- XMLDoc: Aquesta classe ofereix mètodes per crear i generar documents XML, així com buscar elements en aquests documents.
- Windows: Es la classe encarregada de la interfície gràfica. Pot generar i obrir finestres amb les dades que se li envien.

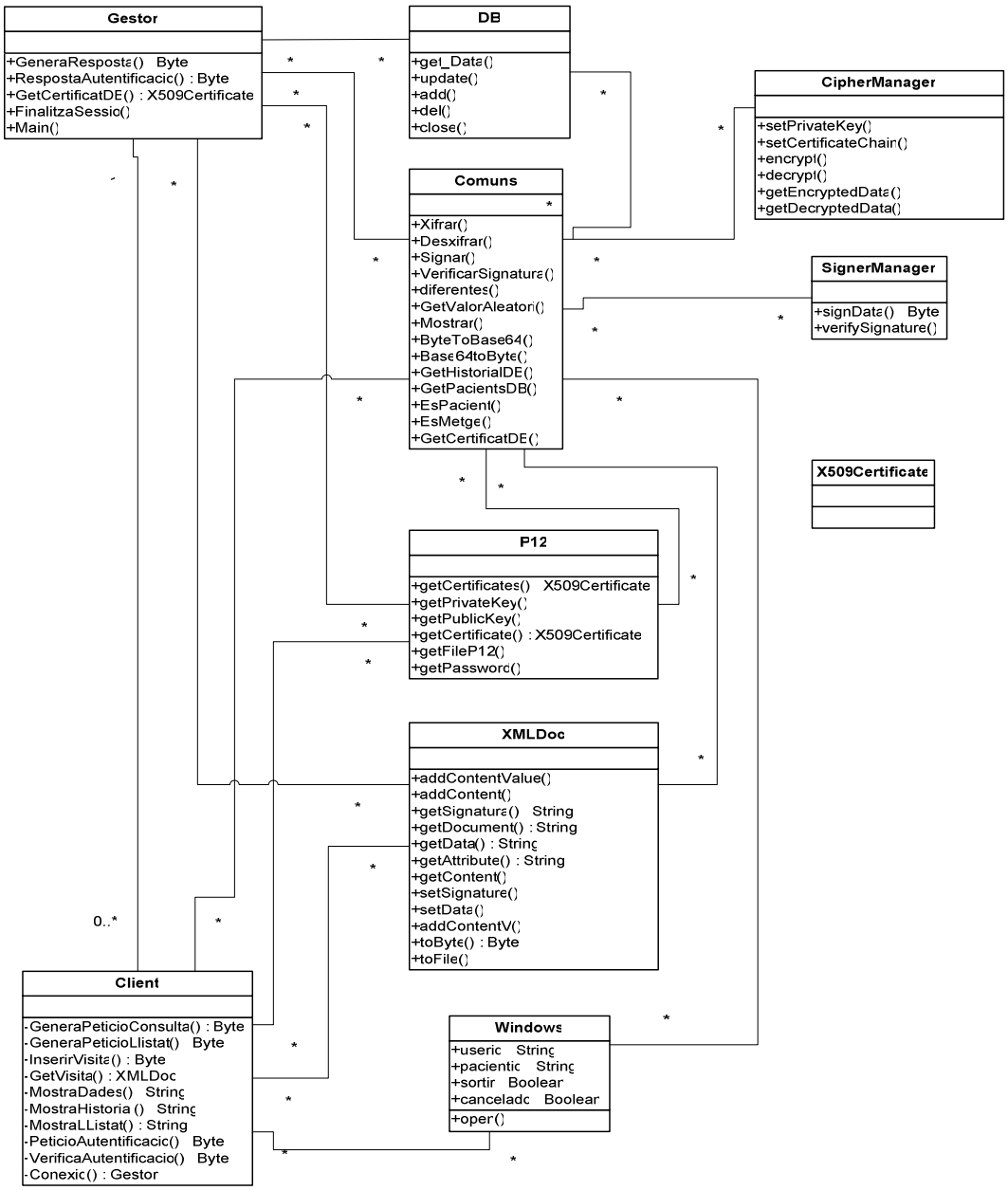


Figura 20. Diagrama de Classes

5 XML

5.1 Definició de XML

XML és l'acrònim de eXtensible Markup Language. Des de que va aparèixer aquesta forma de representar les dades s'ha imposat com una de les formes més eficients per intercanviar i emmagatzemar dades entre aplicacions i/o protocols.

Els documents XML es basen en documents de text pla en els que s'utilitzen etiquetes per delimitar els *elements* d'un document. XML defineix aquestes etiquetes en funció del tipus de dades que està escrivint i no de la aparença final que tindran, com ho fan els documents HTML. A més permet definir noves etiquetes i ampliar les existents.

Un document XML està *well-formed* (ben-format), si acompleix amb totes les definicions bàsiques de format i poden, per tant, ser analitzats correctament per qualsevol "*parser*" que acompleixi amb la norma. Es separa això del concepte de validesa, que implica que no només el documents està ben format sinó que també la seva estructura es correspon amb la definida a un document extern (expressat com DTD o XSchema).

5.2 Tipus de XML usats

En el projecte s'ha utilitzat XML per a fer les transferències de dades que s'envien durant l'execució dels protocols criptogràfics, quan es fan les peticions al Gestor i quan aquest envia la resposta.

Cada cop que s'envien les dades entre client i gestor només s'enviarà la part necessària segons la fase del protocol en la que s'estigui duent a terme la comunicació. Es per això que la classe creada per manegar aquests documents, XMLDoc, està molt oberta i ofereix els mecanismes per crear, primer, un Figura 21. Document XML Basic, i després, mètodes per poder afegir noves dades al document.

Per realitzar aquests mètodes a fet falta utilitzar una llibreria pública JDOM (*Java Document Object Model*) que proveeix d'una solució completa basada en Java per accedir, manipular i exportar dades XML.

A continuació, es detallarà la estructura de cada XML emprat als protocols.

5.2.1 Document XML Basic

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

Figura 21. Document XML Basic

A partir d'aquesta estructura bàsica, s'aniran formant tots els altres documents XML.

5.2.2 Document XML Petició Autenticació

Segons la Figura 4. Diagrama de seqüència del Cas d'us Petició Autentificació, hem de generar un document XML (Figura 22. Document XML Petició Autenticació) amb el numero aleatori Ni i el identificador ID del usuari que vol autenticar-se. El número aleatori Ni, haurà de transformar-se, primer, de byte a Base 64, i després, a String, per poder inserir-lo a un document de text, com es el XML. Això passarà amb totes les dades de tipus byte.

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ni></Ni>
          <ID></ID>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

Figura 22. Document XML Petició Autenticació

Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ni>+mMas/b9LukwiQyPSjbBTzkpQDo</Ni>
          <ID>44183031-P</ID>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

5.2.3 Document XML Resposta Autenticació

Segons la Figura 5. Diagrama de seqüència cas d'us Resposta Autenticació, hem de generar un document XML (
Figura 23. Document XML Resposta Autenticació) amb el numero aleatori Ng, Ni i el identificador ID del usuari que vol autenticar-se.

Figura 23. Document XML Resposta Autenticació

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ni></Ni>
          <Ng></Ng>
          <ID></ID>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

5.2.4 Document XML Genera Petició Consulta

Segons la Figura 10. Diagrama de seqüència del cas d'us Genera Petició Consulta, hem de generar un document XML (

Figura 24. Document XML Genera Petició Consulta) amb el numero aleatori Ng, "Consulta" i el identificador ID del pacient.

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ng></Ng>
          <Op></Op>
          <ID></ID>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

Figura 24. Document XML Genera Petició Consulta

Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ng>+mMas/b9LukwiQyPSjbBTzkpQDo</Ng>
          <Op>Consulta</Op>
          <ID>44183031-P</ID>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

5.2.5 Document XML Resposta Historial

Segons la Figura 11. Diagrama de seqüència del cas d'ús Resposta Historial, hem de generar un document XML (Figura 25. Document XML Resposta Historial) amb el historial del pacient que recull el Gestor de la Base de Dades. En aquesta part del protocol s'han de formar petits XML per anar desxifrant i recol·locant les dades al historial. Tots es basen en el mateix, en el document XML Basic (*Figura 21. Document XML Basic*) i se

li afegeixen les dades necessàries. Per les dades xifrades i signades, al igual que amb els números aleatoris, com ja s'ha dit, primer es necessari o bé passar-lo a Base64 si estem escrivint, o be passar-los de Base64 a byte si estem llegint les dades. A l'exemple de sota no s'han posat totes les dades xifrades, nomes un petit exemple de 3 línies, doncs ocuparien massa. Aquest es l'XML més important, així doncs mostrarem el seu DTD corresponent a una secció posterior.

Exemple:

```
<SignedDocument>
        <document>
                <expedient>
                        <DadesPersonals>
                                <NIF>44183031-P</NIF>
                                <NSS>234566444</NSS>
                                <Nom>Silvia</Nom>
                                <Cognoms>Cernuda Ibanez</Cognoms>
                        </DadesPersonals>
                        <DadesContacte>
                                <tlf>23456566</tlf>
                                <adreca>c/raureda</adreca>
                        </DadesContacte>
                        <DadesMediques>
                                <GrupSang>0+</GrupSang>
                                <alergies>
                                        <alergia>pi</alergia>
                                        <alergia>alzinapi</alergia>
                                </alergies>
                                <MalCroniques>
                                        <MalCronica>colesterol</MalCronica>
                                </MalCroniques>
                                <MalGreus>
                                        < MalGreu>turberculosi</MalGreus>
                                </MalGreus>
                        </DadesMediques>
                        <SignDades> …
AQICCQDY/SROug1kQTANBgkqhkiG9w0BAQUFADCBmTELMAkGA1UEBhMCRVMxEjAQ&#xD;
EwlCYXJjZWxvbmExEjAQBgNVBAcTCUJhcmNlbG9uYTEMMAoGA1UEChMDVU9DMQ8w&#xD;
Fs+In1VxFDAJBgNVHREEAjAAMAkGA1UdEgQCMAAwDQYJKoZIhvcNAQEFBQADggEB …
                        </SignDades>
                        <Historial>
                                <visites>
                                        <visita>
                                                <T>1150044041203</T>
                                                <NS>1</NS>
                                                <XV> …… </XV>
                                                        <V>
                                                                <data>12/05/2006 12:34</data>
                                                                <lloc>hostpi</lloc>
                                                                <IDm>43433920-F</IDm>
                                                                <IDp>44183031-P</IDp>
                                                                <diagnostic>braç trencat </diagnostic>
                                                                <tractament>escaiola</tractament>
                                                                <evolucio>normal</evolucio>
                                                                <privat>si</privat>
                                                        </V>
                                                        <SM></SM>
                                                <SG> ……</SG>
                                        </visita>
                                </visites>
                                <signseq> ……</signseq>
                        </Historial>
                </expedient>
        </document>
        <signatura>
        </signatura>
</SignedDocument>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
	<document>
		<expedient>
			<DadesPersonals>
				<NIF></NIF>
				<NSS></NSS>
				<Nom> </Nom>
				<Cognoms></Cognoms>
			</DadesPersonals>
			<DadesContacte>
				<tlf></tlf>
				<adreca></adreca>
			</DadesContacte>
			<DadesMediques>
				<GrupSang></GrupSang>
				<alergies>
					<alergia></alergia>
				</alergies>
				<MalCroniques>
					<MalCronica></MalCronica>
				</MalCroniques>
				<MalGreus>
					< MalGreu></MalGreus>
				</MalGreus>
			</DadesMediques>
			<SignDades>
			</SignDades>
			<Historial>
				<visites>
					<visita>
						<T></T>
						<NS></NS>
						<XV>
							<V>
								<data></data>
								<lloc></lloc>
								<IDm></IDm>
								<IDp></IDp>
								<diagnostic></diagnostic>
								<tractament></tractament>
								<evolucio></evolucio>
								<privat></privat>
							</V>
							<SM></SM>
						</XV>
						<SG></SG>
					</visita>
				</visites>
				<signseq></signseq>
			</Historial>
		</expedient>
	</document>
	<signatura>
	</signatura>
</SignedDocument>
```

Figura 25. Document XML Resposta Historial

5.2.6 Document XML Mostra Historial

Segons la Figura 12. Diagrama de seqüència del cas d'us Mostrar Historial, hem de generar un document XML amb parts del historial del pacient per verificar algunes dades. Aquestes parts, seran fragments del Document XML Resposta Historial (Figura 25. Document XML Resposta Historial).

5.2.7 Document XML Genera Petició Llistat

Segons la Figura 14. Diagrama de seqüència del cas d'us Genera Petició Llistat, hem de generar un document XML (Figura 26. Document XML Genera Petició Llistat) amb el numero aleatori Ng i "Llista_pacients".

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <Ng></Ng>
          <Op></Op>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

Figura 26. Document XML Genera Petició Llistat

5.2.8 Document XML Resposta Llistat

Segons la Figura 15. Diagrama de seqüència del cas d'us Resposta Llistat, hem de generar un document XML (Figura 27. Document XML Resposta Llistat) amb el numero aleatori Ng i "Llista_pacients".

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
     <document>
          <pacient>
               <NIF></NIF>
               <Nom></Nom>
               <Cognoms></Cognoms>
          </pacient>
     </document>
     <signatura>
     </signatura>
</SignedDocument>
```

Figura 27. Document XML Resposta Llistat

Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
      <document>
            <pacient>
                  <NIF>44183031-P</NIF>
                  <Nom>Silvia</Nom>
                  <Cognoms>Cernuda Ibanez</Cognoms>
            </pacient>
            <pacient>
                  <NIF>44183032-P</NIF>
                  <Nom>Ana</Nom>
                  <Cognoms>Rius Perez</Cognoms>
            </pacient>
      </document>
      <signatura>
      </signatura>
</SignedDocument>
```

5.2.9 Document XML Genera Petició Visita

Segons la Figura 18. Diagrama de seqüència del cas d'us Genera Petició Visita, hem de generar un document XML (Figura 28. Document XML Genera Petició Visita) amb el numero aleatori Ng, "inserir_visita", la visita V i la signatura de la visita SM.

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedDocument>
      <document>
            <Ng></Ng>
            <Op></Op>
            <V>
                  <data></data>
                  <lloc></lloc>
                  <IDm></IDm>
                  <IDp></IDp>
                  <diagnostic></diagnostic>
                  <tractament></tractament>
                  <evolucio></evolucio>
                  <privat></privat>
            </V>
            <SM></SM>
      </document>
      <signatura>
      </signatura>
</SignedDocument>
```

Figura 28. Document XML Genera Petició Visita

5.3 DTD Document XML Resposta Historial

Com que el XML més important es el del Historial, a continuació es mostra el DTD corresponent a aquest document (Figura 29. DTD Document XML Resposta Historial).

```
<!ELEMENT SignedDocument (document,signatura)>
<!ELEMENT document (expedient)>
<!ELEMENT expedient
(DadesPersonals,DadesContacte,DadesMediques,SignDades,Historial)>
<!ELEMENT DadesPersonals (NIF,NSS,Nom,Cognoms)>
<!ELEMENT DadesContacte (tlf,adreca)>
<!ELEMENT DadesMediques (GrupSang,alergies,MalCroniques,MalGreus)>
<!ELEMENT Historial (visites,signseq)>
<!ELEMENT visites (visita*)>

<!ELEMENT alergies (alergia*)>
<!ELEMENT MalCroniques (MalCronica*)>
<!ELEMENT MalGreus (MalGreu*)>
<!ELEMENT visita (T,NS,XV,VS,SG)>

<!ELEMENT VS (V,SM)>
<!ELEMENT V (data,lloc,IDp,IDm,diagnostic,tractament,evolucio,privat)>


<!ELEMENT signatura (#PCDATA)>
<!ELEMENT signseq (#PCDATA)>
<!ELEMENT NIF (#PCDATA)>
<!ELEMENT NSS (#PCDATA)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Cognoms (#PCDATA)>
<!ELEMENT tlf (#PCDATA)>
<!ELEMENT adreca (#PCDATA)>
<!ELEMENT GrupSang (#PCDATA)>
<!ELEMENT alergia (#PCDATA)>
<!ELEMENT MalCronica (#PCDATA)>
<!ELEMENT MalGreu (#PCDATA)>
<!ELEMENT SignDades (#PCDATA)>
<!ELEMENT T (#PCDATA)>
<!ELEMENT NS (#PCDATA)>
<!ELEMENT XV (#PCDATA)>
<!ELEMENT SG (#PCDATA)>
<!ELEMENT SM (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT lloc (#PCDATA)>
<!ELEMENT IDm (#PCDATA)>
<!ELEMENT IDp (#PCDATA)>
<!ELEMENT especialitat (#PCDATA)>
<!ELEMENT diagnostic (#PCDATA)>
<!ELEMENT tractament (#PCDATA)>
<!ELEMENT evolucio (#PCDATA)>
<!ELEMENT privat (#PCDATA)>
```

Figura 29. DTD Document XML Resposta Historial

6 RMI

6.1 Definició de RMI

El RMI (*Java Remote Method Invocation*) permet al programador crear aplicacions distribuïdes basades en Java, en les quals els mètodes dels objectes remots de Java poden ser invocats des de maquines virtual de Java, possiblement en diferents maquines.

Les aplicacions RMI comprenen dos programes separats, un servidor i un client. La aplicació servidora típica crea un conjunt d'objectes remots, fent accessibles unes referències a aquests objectes i espera a la crida d'aquests mètodes o objectes remots per part dels clients. Una aplicació client típica obté una referència d'un o més objectes remots en el servidor i en crida els seus mètodes.

RMI proporciona el mecanisme per el que es comuniquen i passen informació el client i el servidor. Se les anomenen aplicacions d'objectes distribuïts.

Les aplicacions d'objectes distribuïts necessiten:

- Localitzar objectes remots:
    - Les aplicacions poden utilitzar un de dos mecanismes per obtenir referències a objectes remots. Una aplicació pot registrar els seus objectes remots amb una eina molt simple de RMI, el *rmiregistry*, o la aplicació pot passar i retornar referències a un objecte remot com a part del la seva operació normal.

- Comunicar amb objectes remots
  - Els detalls de la comunicació entre objectes remots son donats per RMI al programador. La comunicació remota sembla un invocació d'un mètode estàndard.
- Càrrega de la classe bytecodes per objectes que son passats com paràmetres o retorn de valors
  - Ja que RMI permet crides per passar objectes a objectes remots, RMI proveeix dels mecanismes necessaris per carregar un codi d'objecte tant com per transmetre les seves dades.

La figura de sota (Figura 30. Esquema comunicació RMI) mostra un esquema d'una aplicació distribuïda que utilitza el registre per obtenir referències a objectes remots. El servidor crida al registre per associar un nom amb l'objecte remot. El client busca al objecte remot pel seu nom en el registre del servidor i llavors invoca al mètode. La figura també mostra que el sistema RMI utilitza un servidor web existent per carregar bytecodes de les classes escrites en Java, des de el servidor al client i des de el client al servidor, per objectes quan son necessitats. RMI pot carrega els bytecodes de les classes utilitzant algun protocol URL (pe. HTTP, FTP, file, etc) que es suportat per la plataforma Java.

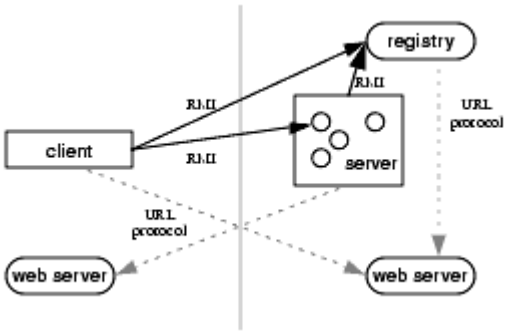


Figura 30. Esquema comunicació RMI

6.1.1 La interfície `java.rmi.Remote`

En RMI, una interfície remota es una interfície que declara un conjunt de mètodes que poden ser invocats des de una maquina virtual de Java. Una interfície remota ha de satisfer els següents requeriments.

- Una interfície remota ha de extendre com a mínim, directa o indirectament la interfície `java.rmi.Remote`.
- Cada declaració de mètode en una interfície remota o les seves superinterfícies han de satisfer els requeriments d'una declaració de mètode remot con segueix:
  - Una declaració de mètode remot ha d'incloure la excepció `java.rmi.RemoteException` (o una de les seves superclasses com `java.io.IOException` o `java.lang.Exception`) a la seva clàusula *`throws`*, afegint-se a alguna excepció específica de l'aplicació.
  - En una declaració de mètode remot, un objecte remot declarat com a paràmetre o com a valor de retorn ha de ser declarat com a interfície remota.

La interfície `java.rmi.Remote` es una interfície feta que no defineix mètodes:

```
Public interface Remote { }
```

6.1.2 Localització d'objectes Remots

La simple activació del servidor de noms esta proveïda per guardar referències de noms a objectes remots. Una referència a un objecte remot pot ser guardat utilitzant mètodes basats en URL de la classe `java.rmi.Naming`.

Per un client invocar un mètode a un objecte remot, primer ha d'obtenir una referència a l'objecte. Una referència a un objecte remot es normalment obtinguda com a paràmetre o valor de retorn en una crida a un mètode. El sistema RMI proveeix d'una activació del servidor de noms des de el qual obté objectes remots en la maquina donada. La classe `java.rmi.Naming` proveeix mètodes basats en URL (Uniform Resource Locator) per buscar, enllaçar (*bind*), reenllaçar (*rebind*), desenllaçar (*unbind*), i llistar el nom de l'objecte que es manté en una particular maquina i port.

6.2 Diagrama de classes RMI

En el cas del nostre sistema apareix una altra classe (veure Figura 31. Diagrama de classes RemoteServer) abans no mencionada, però, que encara que transparent pel client, si te un paper important a l'hora de realitzar la comunicació entre Gestor i Client tal i com s'ha explicat en la secció anterior. Es tracta de la classe *RemoteServer* que exten la classe java.rmi.Remote.

6.3 Excepcions Remotes

La classe `java.rmi.RemoteException` es una superclasse de excepcions a traves de la runtime de RMI durant una invocació a un mètode remot. Per assegurar la robustesa de les aplicacions que utilitzen el sistema RMI, cada declaració d'un mètode remot en una interfície remota ha d'especificar `java.rmi.RemoteException` (o una de les seves superclasses) a la seva clàusula *`throws`*.

La excepció `java.rmi.RemoteException` es transportada quan una invocació d'un mètode remot falla per alguna raó. Algunes raons de fallada inclouen:

- Error de comunicació (el servidor remot no es troba o refusa connexions; la connexió s'ha tancat pel servidor, etc)
- Error duran enviament d'un paràmetre o valor de retorn.
- Errors de protocol

Al projecte s'han capturat totes les excepcions possibles de la banda del servidor i s'han transformat com a noves excepcions de la classe `java.rmi.RemoteException`. Això s'ha fet per a que el client sigues informat de cada error. A continuació un exemple d'això:

```
try {
	VS4 = Comuns.Desxifrar(XV4,this.P12.getPrivateKey());
} catch (Exception e){
	System.out.println(e);
throw new RemoteException("Historial: Dades no valides,
				han sigut modificades");
```

En aquest exemple, si desxifrem una visita i ens dona error, vol dir que aquestes dades no son les correctes i algú sense permís les ha modificat. Això se li ha de informar al client i es fa mitjançant una excepció remota.

També s'han generat excepcions noves en cas d'error del funcionament del protocol. En el següent exemple es fa una verificació de que cada visita existeix al seu ordre i no en falta cap. Si faltes alguna, es generaria una nova `RemoteException` per passar-la cap el client.

```
if (NSant+1!=NS)
    throw new RemoteException("Historial Error (servidor): Falta la visita
                                numero "+(NSant+1));
```

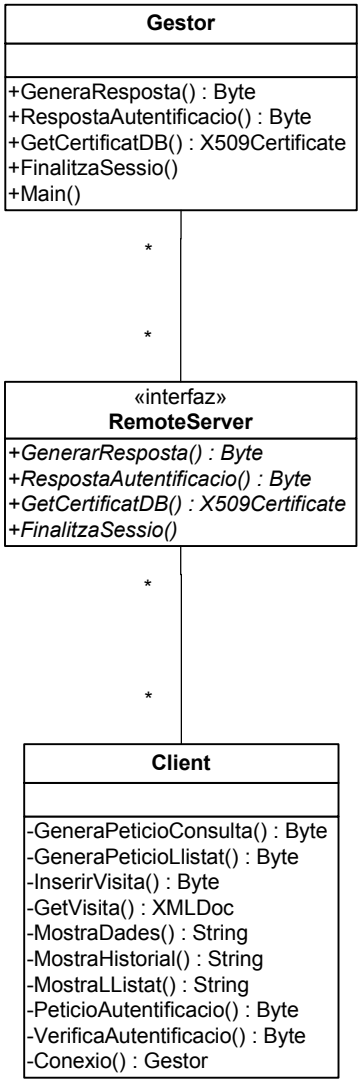


Figura 31. Diagrama de classes RemoteServer

7 Base de Dades

7.1 Introducció

Fins ara, a les proves i execucions de test, les dades es guardaven directament en fitxers XML.

La utilització d'una base de dades permetrà emmagatzemar els historials mèdics d'una forma més segura. En aquest PFC aquesta part es essencial. Per a fer-ho utilitzarem el gestor de Bases de Dades MySQL. Es tracta d'un sistema de dades relacionals SQL de codi obert, desenvolupat, distribuït i suportat per l'empresa MySQL AB.

Podem disposar d'una versió complerta i gratuïta de MySQL, si és per a fins acadèmics.

7.2 Dades

Totes les dades es guardaran a la base de dades excepte els fitxers PKCS#12 (que inclouen les claus privades). Es guardarà total la informació dels pacients, dels metges, les sessions i els seus certificats.

Aquestes dades, són gairebé les mateixes que els camps que s'han utilitzat per generar els XML per fer més fàcil la consulta i generació dels documents.

A la Base de Dades només hi accedirà el gestor, i es per això, que totes les dades que demani el client que hi siguin guardades a la Base de Dades (com serà el més habitual), seran demanades al Gestor i aquest s'encarregarà de accedir-hi.

7.3 Diagrama relacional de la Base de Dades

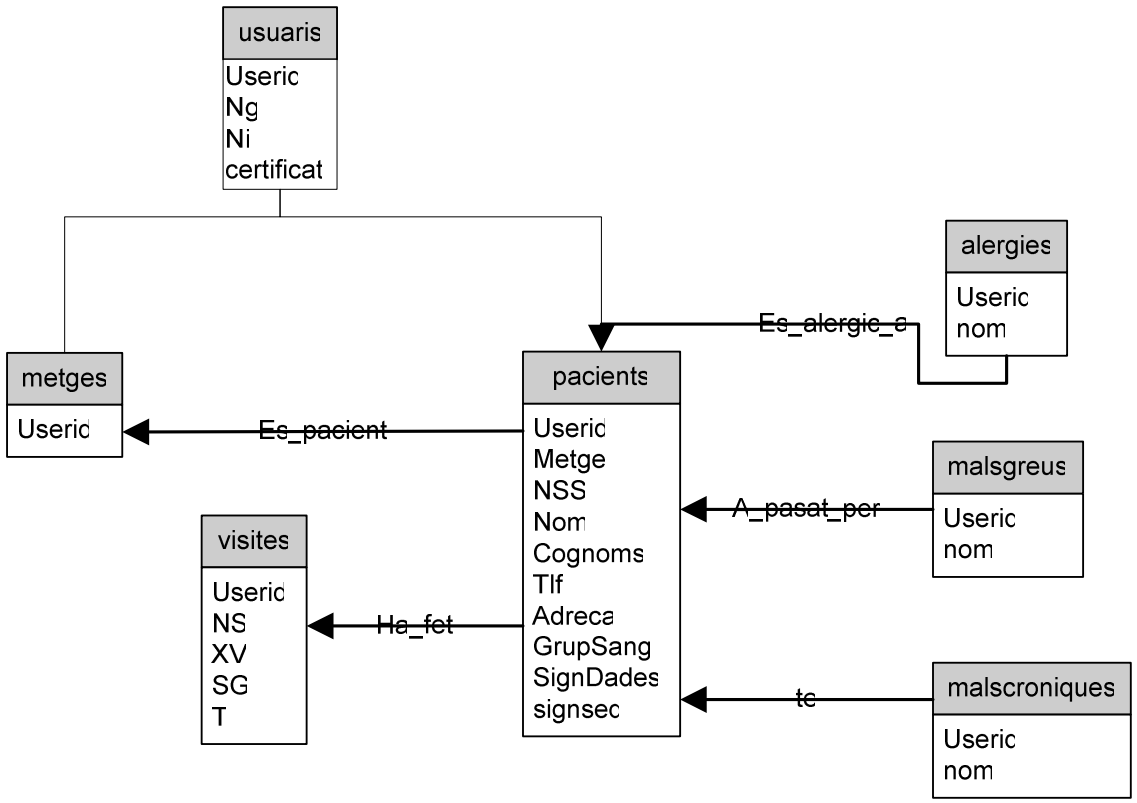


Figura 32. Diagrama relacional de la Base de Dades

Segons en diagrama de la Figura 32. Diagrama relacional de la Base de Dades veiem totes les taules i relacions implicades al model. A continuació es detalla cada relació:

- (metges → usuaris): Relació de 1 a 1. Aquesta no es una relació pròpiament dita, es més una herència, però les dues taules es relaciones doncs pertanyen a la mateixa persona.

- (pacients → usuaris): Relació de 1 a 1. Aquesta no es una relació pròpiament dita, es més una herència, però les dues taules es relaciones doncs pertanyen a la mateixa persona.

- Es_pacient (pacients → metges): Relació * a 1. Relaciona els pacients de la taula Pacients amb els metges de la taula Metges. Un pacient només pot tenir assignat un metge i un metge pot tenir molts pacients assignats.

- Es_alergic_a (alergies → pacients): Relació * a 1. Relaciona les al·lèrgies de la taula alergies amb els pacients de la taula pacients. Un pacient pot tenir moltes alergies.

- A_passat_per (malsgreus → pacients): Relació * a 1. Relaciona les malalties greus de la taula malsgreus amb els pacients de la taula pacients. Un pacient pot haver passat per moltes malalties greus.

- Te (malscroniques → pacients): Relació * a 1. Relaciona les malalties cròniques de la taula malscroniques amb els pacients de la taula pacients. Un pacient pot tenir moltes malalties cròniques.

- Ha_fet (pacients → visites): Relació 1 a *. Relaciona les visites de la taula visites amb els pacients de la taula pacients. Un pacient pot tenir moltes visites al seu historial.

7.4 Descripció de les taules

7.4.1 Taula usuaris

Aquesta taula guarda les sessions dels usuaris que es connecten i els seus certificats. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària.
- ni: Numero aleatori generat per l'usuari
- ng: Numero aleatori generat pel Gestor, i que servirà per autenticar i identificar al usuari a cada petició.
- Certificat: Certificat del usuari.

7.4.2 Taula pacients

Aquesta taula guarda les dades corresponents al pacient, dades personals, de contacte i mediques. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.
- metge: identificador del metge assignat. Referència a userid de la taula usuaris.
- NSS: Numero de la Seguretat Social.
- Nom: Nom del pacient.
- Cognoms: Cognoms del pacient.
- Tlf: Telèfon del pacient.
- Adreca: Adreça del pacient.
- GrupSang: Grup Sanguini.
- SingDades: Signatures de les dades del expedient, excepte les visites. Degut a que les dades de l'expedient no estan xifrades, es fa necessari, posar una mica de seguretat i garantir que no han estat modificades.
- Signseq: Signatura corresponent al userid i NS (numero de visita).

7.4.3 Taula metges

Aquesta taula guarda les dades corresponents als metges. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.

7.4.4 Taula alergies

Aquesta taula guarda el nom de les al·lèrgies que pateix un pacient. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.
- Nom: Nom de l'al·lèrgia.

7.4.5 Taula malscroniques

Aquesta taula guarda el nom de les malalties cròniques que pateix un pacient. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.
- Nom: Nom de la malaltia.

7.4.6 Taula malsgreus

Aquesta taula guarda el nom de les malalties greus per les que ha passat un pacient. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.
- Nom: Nom de la malaltia.

7.4.7 Taula visites

Aquesta taula guarda les visites per les que ha passat un pacient. Esta composta dels següents camps:

- userid: Camp que identifica al usuari. Serà el seu NIF. Clau primària. Referència a userid de la taula usuaris.
- NS: Camp que identifica a la visita. Clau primària.
- XV: Dades de la visita xifrades.
- SG: Signatura de la visita, NS i T.
- T: Timestamp.

8 Interfície Gràfica

8.1 Introducció.

La interfície gràfica es la part que més es miraran els usuaris que la utilitzin i per tant la més crítica. Hauria de ser simple, fàcil d'utilitzar i que causi bona impressió, doncs una interfície mal dissenyada pot causar el fracàs de l'aplicació.

En canvi, en aquest projecte, s'ha donat més importància a la part de seguretat, doncs, es d'això el que es tractava, i no pas a la interfície, que s'ha fet el més senzill possible i donant totes les opcions per realitzar totes les accions del sistema.

En aquest capítol veurem cada finestra però en l'annexa Joc de proves veurem en detall com executar l'aplicació i com preparar-lo per la seva execució.

8.2 Implementació

A l'hora de fer la implementació d'aquesta interfície s'ha utilitzat la llibreria gràfica de Java SWT (*Standard Widget Toolkit)*, que permet dissenyar interfícies d'una manera senzilla. Aquesta llibreria es de lliure distribució i es pot aconseguir amb l'Eclipse, que es l'entorn de treball que s'ha utilitzat.

Tal com es veu al Figura 20. Diagrama de Classes, hi ha la classe Windows. Aquesta classe serà la encarregada de generar les finestres gràfiques.

8.3 Finestra Login

Degut a que el primer que es fa al entrar a l'aplicació es posar el usuari i contrasenya, aquesta finestra (Figura 33. Finestra Login) serà la mateixa tant per un pacient com per un metge. Haurem de posar el identificador, el password i a on esta el nostre fitxer PKCS#12 amb les claus privades. Hi ha un botó per poder buscar aquest fitxer.

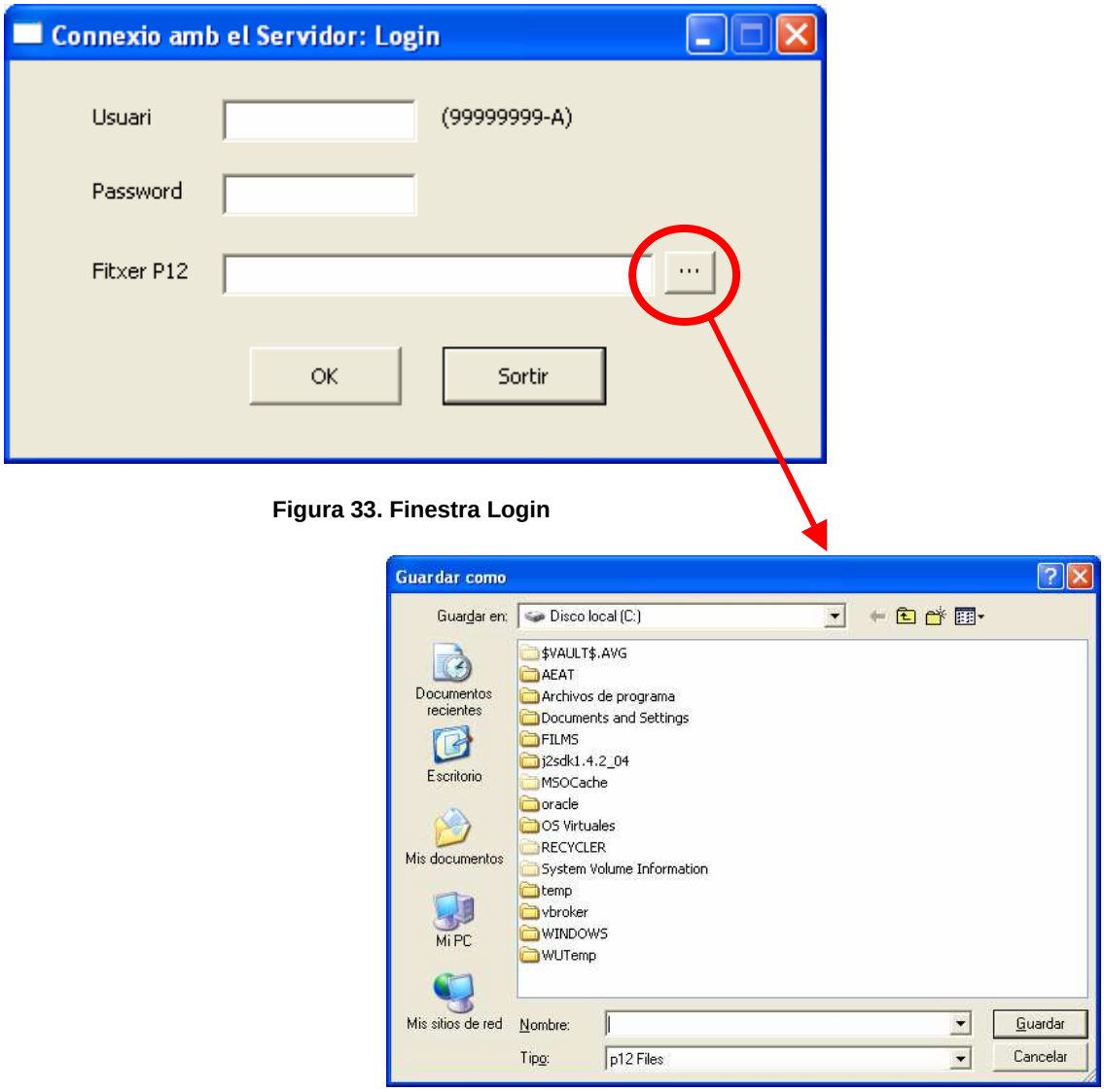


Figura 33. Finestra Login

8.4 Finestra Principal

Un cop hem estat autenticats (parcialment tal com s'explica a la secció Protocol Autenticació) entrarem a la finestra principal on tindrem dues opcions a escollir. A la part de dalt de la finestra, apareixerà el nostre nom (esquerra) i el nostre rol (dreta), es a dir, si som metges o pacients. Depenent del rol, tindrem uns permisos o uns altres. Si som metges tindrem totes les opcions activades (Figura 34. Finestra Principal Metge) mentre que si som pacients, només podrem consulta el nostre historial (
Figura 35. Finestra Principal Pacient).

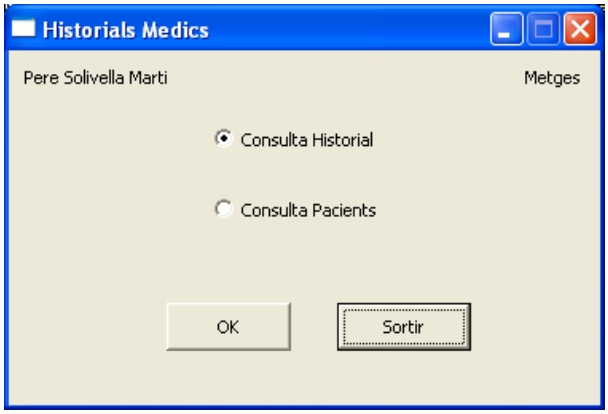


Figura 34. Finestra Principal Metge

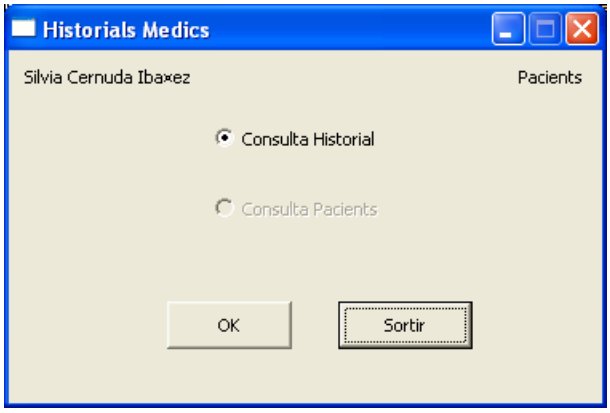


Figura 35. Finestra Principal Pacient

En aquesta finestra apareixen dos botons, el botó "Sortir" que finalitzarà la sessió i sortirà de l'aplicació; i el botó "OK" que donarà pas a la opció escollida.

Opció Consulta Historial: Si som un pacient anirà directament a la Finestra Historial però si es tracta d'un metge apareixerà una finestra (Figura 36. Finestra Inserir Pacient) a on haurem de posar el identificador del pacient del qual volem consultar el seu Historial, i una vegada validat es mostrarà la finestra Historial.

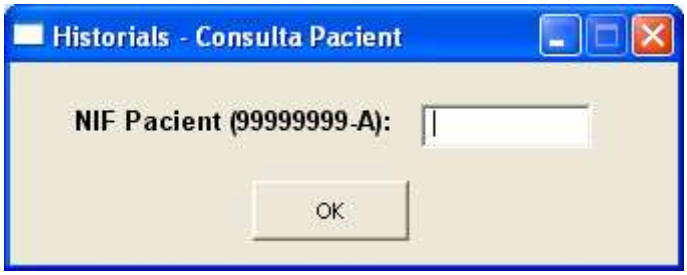


Figura 36. Finestra Inserir Pacient

Opció Consulta Pacients: Aquesta opció només està activada per a pacients i quan s'escull ens envia a la Finestra Llistat Pacients.

8.5 Finestra Historial

A la primera finestra del historial (Figura 37. Finestra Dades Personals) apareixen les dades personals del pacient. Aquesta finestra esta dividida en dues parts. La part de dalt, a on es poden veure diferents pestanyes que dones pas a diferents parts del expedient, i la part de sota, que son les opcions possibles a realitzar després. Començarem per les pestanyes.

Pestanya Dades Personals: En aquesta finestra, com ja hem dit, podem veure, tant, les dades personals del pacient, com, les dades seves dades de contacte.

Pestanya Dades Mèdiques: En aquesta finestra (Figura 38. Finestra Dades Mèdiques) apareixen les dades mèdiques del pacient, el grup sanguini, al·lèrgies i malalties.

Pestanya Historial: En aquesta finestra (Figura 39. Finestra Historial) podem veure totes les visites registrades del pacient.

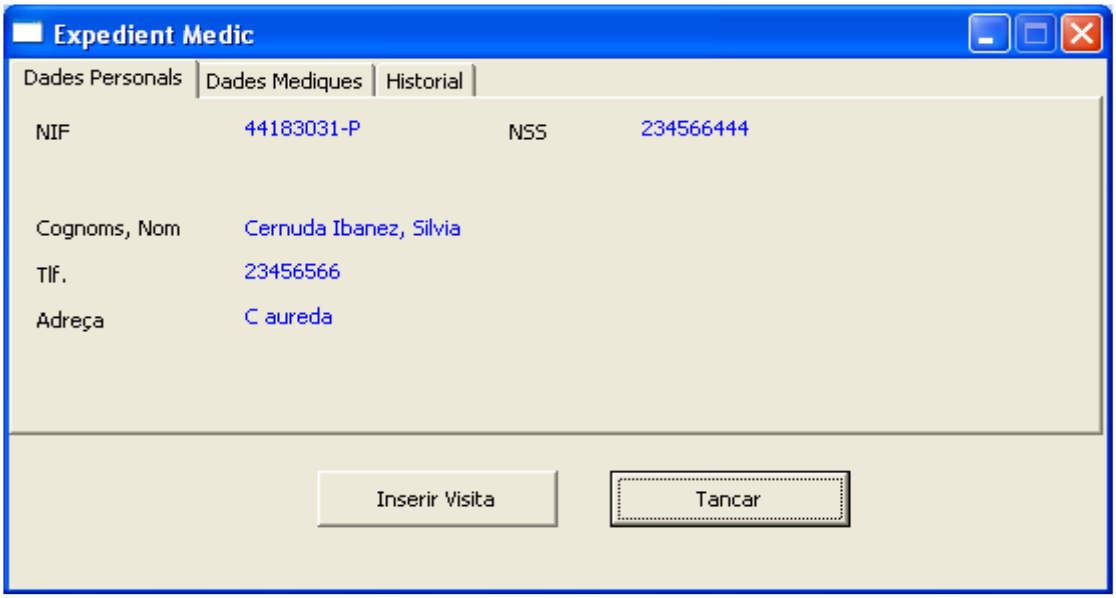


Figura 37. Finestra Dades Personals

Totes les pestanyes son de lectura, és a dir, en cap moment es poden modificar, cosa que garanteix els requeriments o necessitats del sistema. En canvi, un metge si que ha de poder inserir les visites. Això es pot fer a la part de sota del Expedient.

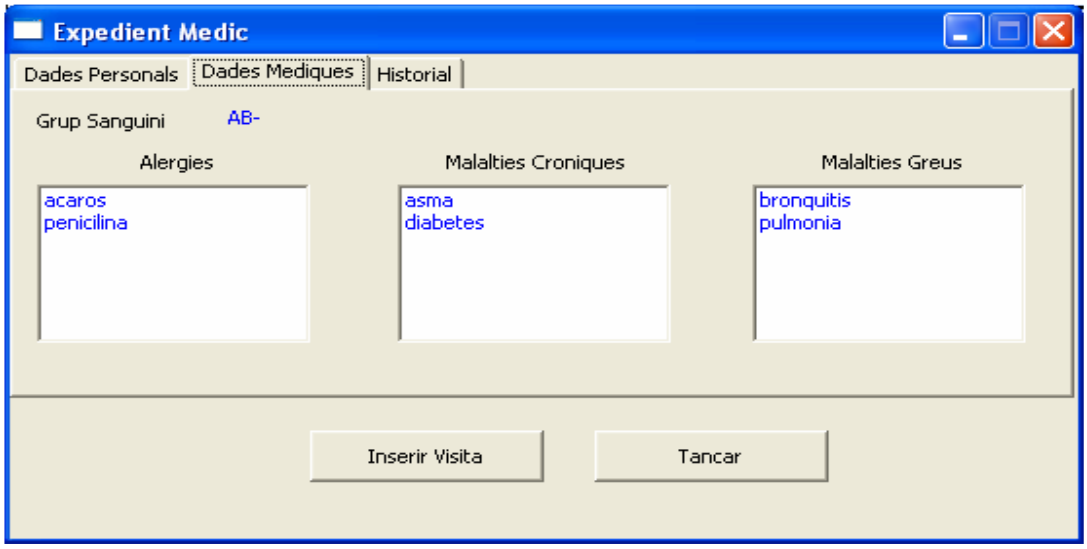


Figura 38. Finestra Dades Mèdiques

Si ens fixem tenim dues opcions a la part de sota de totes les pestanyes. Es una part comú. Tenim dos botons diferents

Botó Tancar: Aquest botó, com el seu nom indica, serveix per tancar la finestra de l'expedient mèdic, i apareixerà la finestra principal del metge o pacient, segons sigui l'usuari que esta loginat (Figura 34. Finestra Principal Metge).

Botó Inserir Visita: Aquest botó té la particularitat que, si el usuari que ha entrar al sistema es un metge, estarà activat, però en canvi, si es un pacient, estarà desactivat. Aquesta opció ens portarà a la Finestra Visita.

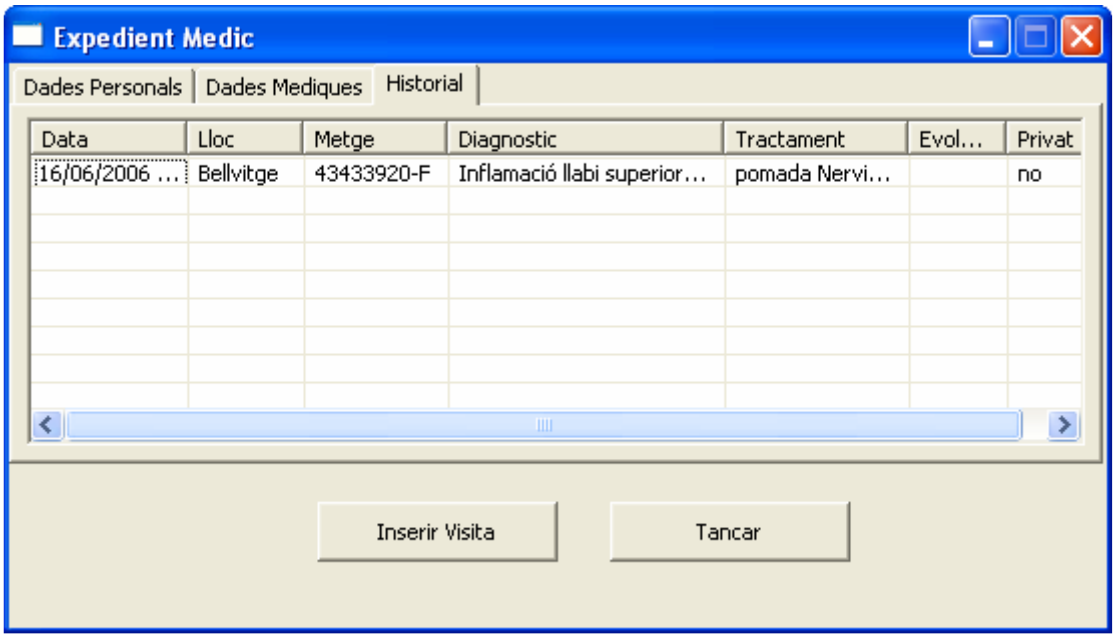


Figura 39. Finestra Historial

8.6 Finestra Visita

Aquesta finestra (Figura 40. Finestra Inserir Visita) permetrà a un metge inserir les dades corresponents a una visita. Aquí també tindrem dos botons corresponents a dues opcions:

Botó Cancel·lar: Amb aquest botó podrem cancel·lar la operació d'inserció retornant a la finestra de l'expedient del pacient que havíem consultat (Figura 37. Finestra Dades Personals).

Botó OK: Accepta les dades posades a la visita i s'insereixen a la Base de Dades retornant a la finestra de l'expedient del pacient (Figura 37. Finestra Dades Personals) que havíem consultat i amb les dades actualitzades.

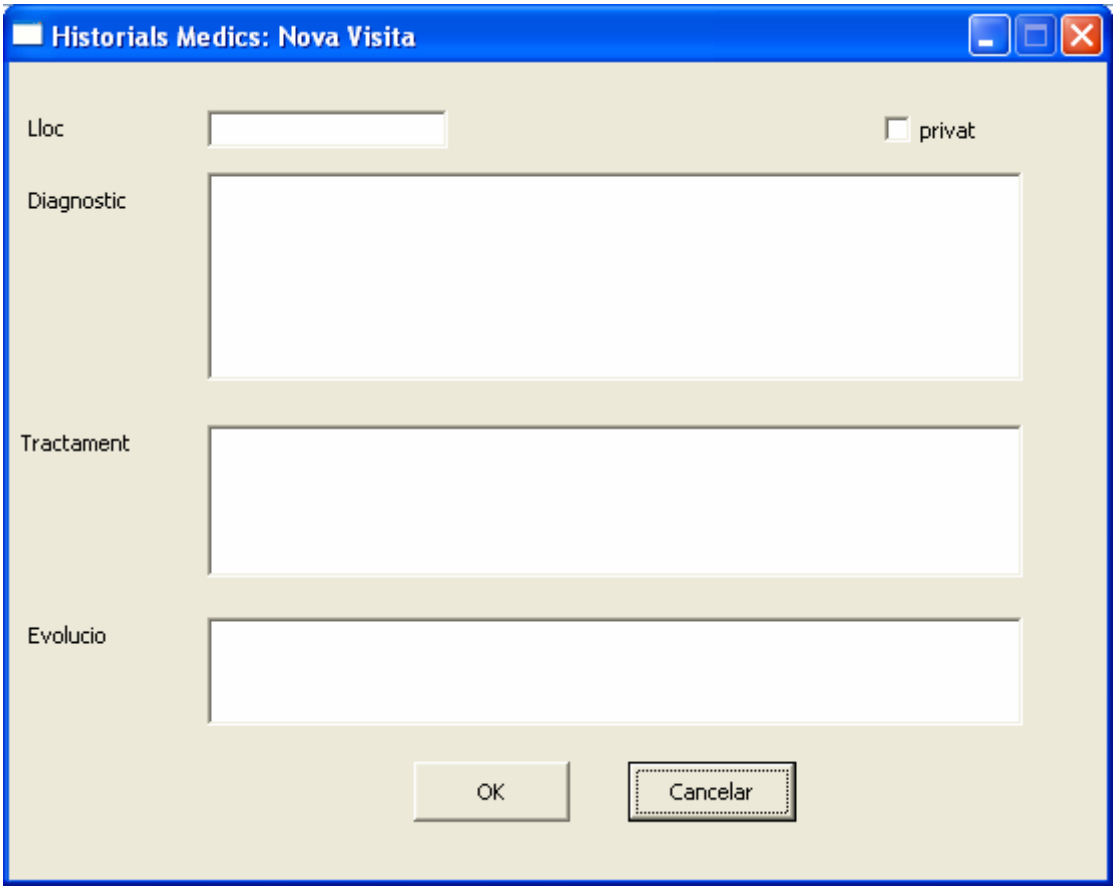


Figura 40. Finestra Inserir Visita

8.7 Finestra Llistat Pacients

Aquesta finestra (Figura 41. Finestra Llistat Pacients) només hi podrà accedir un metge i donarà com a resultat un llistat dels pacients que té assignats el metge que realitza la consulta. En aquesta finestra tenim dos parts, una amb el llistat dels pacients i una altra amb dos botons corresponents a dues opcions.

A la part del llistat dels pacients, apareix per cada pacient el seu identificador (NIF) nom i cognom, i, a més, a la primera columna, la possibilitat d'escollir només un dels pacients de la llista.

A la part dels botons tenim dues opcions:

Botó Tancar: Aquest botó ens tancarà la finestra i retornarem a la finestra Principal (Figura 34. Finestra Principal Metge).

Botó Veure Expedient: Aquest botó només estarà activat si s'ha escollit algun pacient. Si està activat i l'escollim, accedirem a la finestra Historial (Figura 37. Finestra Dades Personals) amb les dades del pacient escollit.

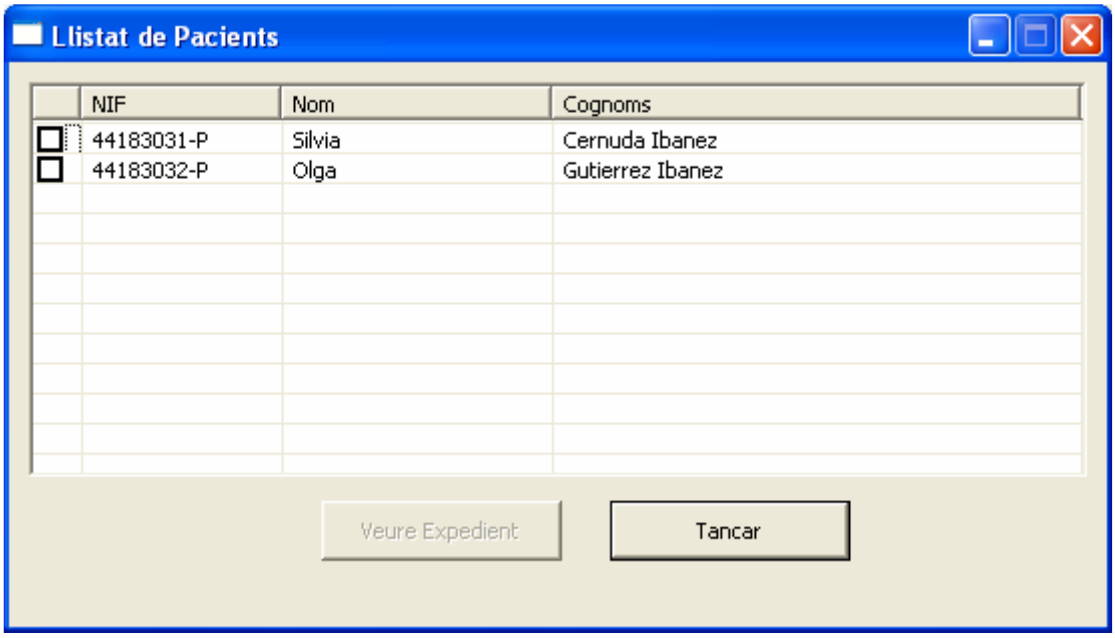


Figura 41. Finestra Llistat Pacients

8.8 Finestra Missatges

Aquesta finestra (Figura 42. Finestra Missatges) s'ha implementat per mostrar els missatges d'error. Algun d'ells podria ser el següent:

Figura 42. Finestra Missatges

9 Treball Futur

9.1 Introducció

Aquesta aplicació, encara que ha assolit totes les fites proposades al PFC, no es pot dir que estigui finalitzada. Son rares les vegades que es pot donar per finalitzat un projecte doncs sempre hi ha alguna cosa que no s'ha tingut en compte o algun aspecte a millorar.

A continuació es detalles algunes coses que es podrien millorar i algunes possibles ampliacions.

9.2 Millores Base de Dades

Les millores a la base de dades poden ser moltes. Des de afegir-ne més informació inclús millorar el gestor.

Millora de la informació

Com a dades, es podria millorar i ampliar el model de dades, afegint-ne molta més informació relativa i reestructurant les taules. Com a exemples, es podria ficar informació relativa a metges, doncs ara, el que consta, només es el seu identificador, si bé es cert que el seu nom i cognoms es pot obtenir de la seva clau P12, però només hi seria una informació de la seva sessió, no pas inclosa a la base de dades. A més, aquesta informació comú a metges i pacients (nom, cognoms, tlf, adreça, etc), posar-la a la taula usuaris.

També es podria ampliar la informació relativa a cada visita, es a dir, posar els símptomes, si esta prenent algun medicament, s'hauria de fer un estudi exhaustiu de

quines dades necessita un metge a l'hora de posar la informació d'una visita. El mateix passaria amb les dades mèdiques, segurament, hi ha molta més informació que en seria d'utilitat.

La reestructuració del model de dades vindria una mica en funció de les dades afegides.

Millora del gestor

Pel que fa a les millores del gestor, Mysql es un base de dades poc extesa i que caldria assegurar-se de quin es el volum de dades màxim. Una base de dades per guardar historials i dades mèdiques podria arribar a tenir un gran volum de dades i caldria tenir ben clar si Mysql garanteix aquests requeriments o seria millor buscar un altre Gestor com Oracle o SQL Server, entre d'altres.

9.3 Millores Actors

Ara mateix només tenim 3 actors definits, com son el Gestor, Metge i Usuari. En canvi, en la realitat, intervenen molts més actors, com podrien ser infermers, administratius sanitaris, etc. A continuació es detallen aquests dos:

- Infermers: Actor que representa el rol dels Infermers. Faran anotacions del dia a dia del pacient, com per exemple, si ha tingut febre, reaccions a medicaments, etc. Podran consultar la part del historial on el Metge hagi deixat els medicaments a donar al pacient. S'hauria d'estudiar si es necessari que tinguessin accés a tot el historial mèdic.
- Administratius Sanitaris: Actor que representa el rol del Administratiu. Faran anotacions del tipus entrada/sortida a l'hospital, comprovacions/canvis de dades personals, trasllats, etc.

9.4 Millores Serveis/Accions

Els serveis que publica el Gestor ara mateix només son 3, una consulta del historial, un llistat dels pacients assignats a un metge i una inserció de visites. Ens faltarien moltes coses encara, com per exemple com es farà el registre del usuaris que entraran al sistema o com es farà la introducció de les dades mèdiques que no pertanyen a una visita, etc.. A continuació, s'expliquen algunes d'elles:

- Administració usuaris: Els administratius o el rol escollit (potser seria necessari un rol administrador), s'encarregarà de donar d'alta o baixa als usuaris del sistema i assignar-los un rol o un altre.

- Inserció de dades: Metges, Infermers i Administratius Sanitaris podran afegir dades a l'expedient d'un pacient. Els metges i infermers podran afegir dades al historial mèdic però no els Administratius. Algú hauria d'afegir les dades mèdiques.

- Modificació dades: Nomes els administratius i pacients podran modificar les dades personals, mai el historial mèdic, que ningú podrà modificar. Els pacients, a més, només podran modificar les dades de contacte.

- Recerca dades: Els metges haurien de poder buscar dades especifiques als historials dels pacients, com per exemple, les dades que nomes ha introduït ell, o pacients amb els mateixos símptomes o diagnòstic, o que pateixen les mateixes malalties, etc.

9.5 Millores Interfície Gràfica

Com ja havia comentat la interfície gràfica no era pas el més important en aquest projecte i per tant, aquí hi ha moltes coses a millorar, entre elles l'aspecte, com es mostren les dades, ara mateix les visites no es veuen gaire clares, encara que si hi es tota la informació.

Es podria afegir també una interfície per al rol que modifiqui les dades personals o mediques dels pacients i una altra per qui doni d'alta un usuari al sistema, doncs ara mateix es fa manualment al SGBD.

9.6 Millores Criptogràfiques

Es podria permetre la comprovació dels certificats amb el certificat de la CA. Això implicaria incloure un repositori de CA's de confiança.

També es podria millorar el tema del login al sistema, doncs ara, l'usuari que entra ha de tenir el seu fitxer PKCS12 i portar-lo a tot arreu on es connecti. Es podria tenir una tarja amb la seva identificació i utilitzar-la pel login.

10 Conclusions

10.1 Conclusions Generals

S'ha dissenyat i implementat un esquema criptogràfic per gestionar de forma segura els historials mèdics dels pacients a través d'una xarxa de comunicacions.

S'han complert tots els objectius en quant a seguretat, desenvolupant el sistema amb els protocols i eines especificades en el PFC. S'ha implementat l'autentificació dels usuaris, ja no tan sols a cada execució dels altres protocols, sinó per separat i en una única vegada. S'ha implementat el protocol de consulta d'un expedient. S'ha implementat la consulta dels pacients d'un metge i, a més, s'ha donat la possibilitat de veure l'expedient del pacient des de aquesta consulta. S'ha implementat la inserció de visites a un historial mèdic. Tots els protocols s'han implementat amb control d'accessos.

Per cada protocol s'han garantit els diferents aspectes de seguretat com son la confidencialitat, l'autenticitat, la integritat i el no-repudi de les dades.

Finalment, com a resultat del PFC s'ha optingut una aplicació que ha permes que un metge pugui consultat i modificar el historial d'un pacient de forma segura; que un pacient pugui utilitzar aquesta aplicació per consultar les dades del seu historial; i que un Gestor Central tingui un repositori (Base de Dades MySQL) amb tots els historial i controli la seva gestió de forma segura.

10.2 Conclusions Personals

En un principi, el tema d'aquest PFC ja hem resultava bastant engrescador, doncs veure com es podia donar seguretat a una cosa tant important com un expedient mèdic ja era molt important i semblava de molta utilitat.

Per a desenvolupar aquesta aplicació a estat necessari aprendre a utilitzar i/o consolidar coneixements de certes eines informàtiques i de certs temes que fins ara havia vist per separat i mai amb una certa utilitat.

El fet de poder englobar tots aquest coneixements en una aplicació amb uns objectius clars a fet que certs coneixement s'anessin consolidant.

He consolidat els coneixements adquirits a assignatures com Criptografia, Eng. del programari de components i sistemes distribuïts, Arquitectura de sistemes distribuïts, entre d'altres.

A més he aprés moltes més coses sobre la programació Java amb Eclipse, especialment amb la llibreria gràfica SWT, doncs mai havia desenvolupat cap aplicació Java amb gràfics.

Respecte a la Criptografia hi havia molts aspectes que no en tenia molt clars i després d'haver finalitzat aquest PFC tinc unes bases bastant assentades sobre aquest tema.

La valoració global ha estat molt satisfactòria i la sensació que tenia en un principi s'ha verificat.

11 Glossari

- **API**: *Application Programming Interface*. Interfície a través de la qual un programa accedeix als serveis del sistema operatiu i d'altres. Una API proveeix un nivell d'abstracció entre l'aplicació i el kernel per tal d'assegurar la portabilitat del codi.

- **CA**: Autoritat certificadora que emet certificats digitals.

- **Eclipse**: Eina que permet el desenvolupament de programari escrit en llenguatge Java de manera integrada, des de la redacció del codi, fins la posterior compilació i l'execució a passos (debugging) per verificar els errors.

- **IAIK**: Llibreria criptogràfica per Java

- **Java**: llenguatge de programació multi-plataforma, robust, interpretat, distribuït, orientat a objectes, portable, desenvolupat per SUN Microsystems.

- **JDOM**: *Java Document Object* Model. Biblioteca de classes per manipular documents XML amb Java.

- **MySQL**: Gestor de bases de dades relacionals SQL de llibre distribució.

- **OpenSSL**: Programari per gestionar certificats criptogràfics.

- **PKCS**: *Public-Key Cryptography Standards*. Són un conjunt d'estàndards definits pels laboratoris RSA que especifiquen els estàndards de clau pública.

- **PKI**: *Public Key Infrastructure*. Infraestructura de clau pública.

- **RMI**: *Remote Method Invocation*. API propietària de Java que permet a les aplicacions locals executar codi que es troba allotjat en una altra màquina remota. Aquesta última posa a disposició uns mètodes públics que seran accessibles a través d'una interfície.

- **SQL**: Llenguatge de bases de dades.

- **UML**: *Unified Model Language*. Metodologia de disseny d'aplicacions informàtiques. És un mètode utilitzat per a especificar, visualitzar, construir i documentar un sistema orientat a objectes en fase de desenvolupament.

- **XML**: *eXtensible Markup Language.* Llenguatge de representació de dades.

12 Bibliografia

- Apunts de Criptografia de la UOC (2004)
- JavaTM 2 Platform Standard Edition 5.0 API Specification
  http://java.sun.com/j2se/1.5.0/docs/api/index.html
- IAIK
  http://www.rediris.es/cert/iris-pca/gti-pca/tras-JT01/Librerias_Criptografica.ppt
  http://jce.iaik.tugraz.at/sic/media/product_pdfs/iaik_jce
- OpenSSL
  http://www.openssl.org/
- PKCS#12
  http://www.faqs.org/rfcs/rfc3280.html
- XML
  http://www.w3.org/TR/2004/REC-xml-20040204/
- RMI
  http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmi-objmodel2.html
- JDOM Documentation
  http://www.jdom.org/downloads/docs.html

- MySQL Reference Manual 5.0.0
  http://helpguide.inmotionhosting.com/mysql_faq/
  http://sunsite.mff.cuni.cz/MIRRORS/ftp.mysql.com/doc/en/index.html

- Wikipedia: Glossari Basic
  http://www.wikipedia.org

- Eclipse universal tool platform
  www.eclipse.org/downloads/index.php

- The Win32 OpenSSL Installation Project
  www.slproweb.com/products/Win32OpenSSL.html

13 Annexos

13.1 Relació dels arxius adjunts a la memòria

Juntament amb la memòria s'adjunten els arxius que forment part tant del codi, com binaris, documentació, etc. Seguidament es fa un detall de tots ells:

Carpeta	Contingut	Descripció
/scr	*.java /db	Codi Font del Sistema més arxius necessaris per la creació de la base de dades
/project	Tots els arxius Java, Class i necessaris per un projecte Eclipse	Exportació del projecte desenvolupat amb Eclipse
/pki	*.p12 *.crt Openssl.conf	Tots els arxius PKCS#12 i els certificats dels usuaris utilitzats a les proves. A més s'inclou el fitxer de configuració utilitzat per generar-los.
/doc	PFC_Memoria.doc PFC_Memoria.pdf	Documentació del projecte
/bin	*.class *.dll /lib/*.jar README.txt	Tots els binaris i llibreries necessàries per l'execució. S'adjunta un fitxer README amb les ordres d'execució.

Taula 4. Relació d'arxius del PFC

13.2 Passes per posar en marxa el joc de proves

Com que inicialment la Base de dades està buida, hem de ficar les dades inicials a mà, com es la taula d'usuaris, pacients, metges i si es vol alergies, malsgreus i malscroniques; A l'aplicació no hi cap manera de fer-ho.

Creació Signatures de les dades

Inicialment, al omplir la Base de Dades, hi ha uns camps pels pacients que son les signatures de les dades personals, de contacte i mediques signades, el camp SingDades, i el camp singseq que es la signatura del identificador ID + el numero de visites. Inicialment no hi són. Abans d'omplir la Base de Dades necessitarem aquestes signatures. S'ha desenvolupat un petit codi per generar aquestes signatures. Es tracta del fitxer `Datos.java` que s'ha afegir al directori "`src`" del projecte i el seu corresponent `.class` al `bin`.

Per executar aquest codi cal pasar-li el fitxer XMLDades per cada pacient, primer amb les dades per la signatura SingDades. S'han inclòs al projecte al directori `src/db` per cada pacient (`Pacient-44183031p-expedient.xml` i `Pacient-44183032p-expedient.xml`):

```
java Datos ../pki/Gestor.p12 uoc0506 ../src/db/Pacient-44183031p-expedient-sign.txt ../src/db/Pacient-44183031p-expedient.xml

java Datos ../pki/Gestor.p12 uoc0506 ../src/db/Pacient-44183032p-expedient-sign.txt ../src/db/Pacient-44183032p-expedient.xml
```

Aconseguirem un fitxer per cada pacient amb les dues signatures, que li passarem a les comandes de la base de dades (també s'inclouen els fitxers ja generats).

Ara executarem el mateix codi per generar les signatures singseq:

```
java Datos ../pki/Gestor.p12 uoc0506 ../src/db/Pacient-44183031p-signseq-sign.txt ../src/db/Pacient-44183031p-signseq.xml

java Datos ../pki/Gestor.p12 uoc0506 ../src/db/Pacient-44183032p-signseq-sign.txt ../src/db/Pacient-44183032p-signseq.xml
```

Aconseguirem un fitxer per cada pacient amb les dues signatures, que li passarem a les comandes de la base de dades (també s'inclouen els fitxers ja generats).

Creació Base de dades

Crear la Base de dades i omplir-la amb l'script: ../src/db/create-db.sql.

```
mysql -u root

GRANT ALL PRIVILEGES ON *.* TO 'scernuda'@'localhost'
IDENTIFIED BY 'password' WITH GRANT OPTION;

mysql -u scernuda -p
create database Historials;

use Historials;
```

Creació Taules

```
create table usuaris (
       userid varchar(10) NOT NULL PRIMARY KEY,
       Ng char(27),
       Ni varchar(27),
       certificat blob NOT NULL);

create table pacients (
       userid varchar(10) NOT NULL PRIMARY KEY references usuaris(userid),
       metge varchar(10) NOT NULL references metges(userid),
       NSS varchar(10) NOT NULL,
       Nom varchar(20),
       Cognoms varchar(50),
       tlf varchar(10),
       adreca varchar(50),
       GrupSang varchar(3),
       SignDades Text NOT NULL,
       signseq varchar(3794) NOT NULL);

create table metges (
       userid varchar(10) NOT NULL PRIMARY KEY references usuaris(userid),
       especialitat varchar(30) not null);

create table alergies (
       userid varchar(10) NOT NULL references pacients(userid),
       nom varchar(20) NOT NULL,
       PRIMARY KEY(userid,nom));

create table malscroniques (
       userid varchar(10) NOT NULL references pacients(userid),
       nom varchar(20) NOT NULL,
       PRIMARY KEY(userid,nom));

create table malsgreus (
       userid varchar(10) NOT NULL references pacients(userid),
       nom varchar(20) NOT NULL,
       PRIMARY KEY(userid,nom));

create table visites (
       userid varchar(10) NOT NULL references pacients(userid),
       NS int NOT NULL,
       XV text NOT NULL,
       SG text NOT NULL,
       T mediumtext NOT NULL,
       PRIMARY KEY (userid,NS));
```

Introducció dades

```
insert into usuaris values (
       '00000000-A',
       null,
       null,
       LOAD_FILE('../pki/Gestor.crt'));
```

```
insert into usuaris values (
       '44183031-P',
       null,
       null,
       LOAD_FILE('../pki/Pacient-44183031p.crt'));

insert into usuaris values (
       '44183032-P',
       null,
       null,
       LOAD_FILE('../pki/Pacient-44183032p.crt'));

insert into usuaris values (
       '43433920-F',
       null,
       null,
       LOAD_FILE('../pki/Metge-43433920f.crt'));

insert into pacients values (
       '44183031-P',
       '43433920-F',
       '234566444',
       'Silvia',
       'Cernuda Ibanez',
       '23456566',
       'C/raureda',
       '0+',
       LOAD_FILE('../src/db/Pacient-44183031p-expedient-sign.txt'),
       LOAD_FILE('../src/db/Pacient-44183031p-signseq-sign.txt'));

insert into pacients values (
       '44183032-P',
       '43433920-F',
       '233467884',
       'Olga',
       'Gutierrez Ibanez',
       '32456784',
       'C/vinaroz, 20, 3-2',
       'AB-',
       LOAD_FILE('../src/db/Pacient-44183032p-expedient-sign.txt'),
       LOAD_FILE('../src/db/Pacient-44183032p-signseq-sign.txt'));

insert into alergies values ('44183031-P','acaros');
insert into alergies values ('44183031-P','penicilina');
insert into alergies values ('44183032-P','acaros');
insert into alergies values ('44183032-P','penicilina');

insert into malscroniques values ('44183031-P','asma');
insert into malscroniques values ('44183031-P','diabetes');
insert into malscroniques values ('44183032-P','asma');
insert into malscroniques values ('44183032-P','diabetes');

insert into malsgreus values ('44183032-P','pulmonia');
insert into malsgreus values ('44183032-P','bronquitis');

insert into metges values ('43433920-F','capçalera');
```

13.3 Joc de proves

Execució

- Crear la Base de Dades amb l'script anteriorment comentat: `../src/db/create-db.sql` modificant els directoris on son els fitxers
- Executar el Gestor:
  - `rmic Gestor`
  - `rmiregistry 2001`
  - `java Gestor "..\pki\Gestor.p12" "uoc0506"`
- Executar el client:
  - `set LIBS=..\bin\lib`
  - `set CLASSPATH=%CLASSPATH%;%LIBS%/org.eclipse.swt.win32.win32.x86_3.1.2.jar;%LIBS%/org.eclipse.core.commands_3.1.0.jar;%LIBS%/org.eclipse.core.runtime_3.1.2.jar;%LIBS%/org.eclipse.jface_3.1.1.jar;%LIBS%/org.eclipse.jface.text_3.1.2.jar;%LIBS%/org.eclipse.osgi_3.1.2.jar;%LIBS%/org.eclipse.swt.win32.win32.x86_3.1.2.jar;%LIBS%/org.eclipse.text_3.1.1.jar;%LIBS%/org.eclipse.ui.forms_3.1.0.jar;%LIBS%/org.eclipse.ui.workbench_3.1.2.jar;.`
  - `java Client`

Proves amb un Pacient

- Posar les dades d'un pacient (pe. `44183031-P`, `uoc0506`, `../pki/Pacient-44183031p.p12`) i prémer OK
- A la finestra Historials Mèdics comprovar que nomes es pot seleccionar Consulta Historial. Seleccionar-lo i prémer OK
- Apareixerà el seu historial, comprovar que no pot inserir cap visita.
- Prémer boto Tancar i Sortir.

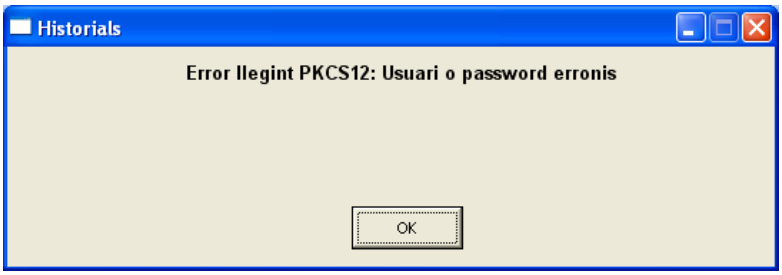
Proves amb un Metge

- Posar les dades d'un metge (pe. `43433920-F`, `uoc0506`, `../pki/Metge-43433920f.p12`) i prémer OK
- A la finestra Historials Mèdics seleccionar Consulta Historial i prémer OK
- Ficar el identificador d'un pacient (pe. `44183032-P`) i prémer OK
- Apareixerà el seu historial, comprovar que no hi ha cap visita.
- Prémer botó "Inserir Visita"
- Apareixerà una finestra per introduir dades. Introduir-les i prémer OK
- Comprovar que s'ha inserit la visita a la pestanya Historial
- Prémer boto Tancar
- Seleccionar la opció Consulta Pacients
- Apareixerà una finestra amb el llistat dels pacients
- Escollir un y prémer botó "Veure Expedient"
- Tancar i Sortir

Proves d'error

- Posar dades les dades d'un usuari, però sense fitxer p12 (pe. `43433920-F, uoc0506,`) i prémer OK

- Posar les dades d'un usuari, però sense password (pe. `43433920-F, , ../pki/Metge-43433920f.p12`) i prémer OK



- Posar les dades d'un usuari, però amb el NIF equivocat (pe. `4343-F, uoc0506, ../pki/Metge-43433920f.p12`) i prémer OK

- Posar les dades d'un metge (pe. `43433920-F, uoc0506, ../pki/Metge-43433920f.p12`) i prémer OK
- A la finestra Historials Mèdics seleccionar Consulta Historial i prémer OK
- Ficar el identificador d'un pacient però equivocat (pe. `4418-P`) i prémer OK