PROJECTE FI DE CARRERA:

DISSENY I DESENVOLUPAMENT D'UN ESQUEMA CRIPTOGRÀFIC PER GESTIONAR DE FORMA SEGURA ELS HISTORIALS MÈDICS DELS PACIENTS A TRAVÉS D'UNA XARXA DE COMUNICACIONS

Fortià Bofill Espada
Enginyeria en Informàtica

Jordi Castellà Roca
Consultor

14 de Gener de 2011

Llicència

Resum

Amb aquest projecte es vol proporcionar una solució en el camp de la sanitat per tal de millorar l'atenció mèdica que reben els pacients donant-los la possibilitat de consultar el seu historial de forma remota i facilitar la feina als metges permetent-los consultar i afegir informació a la història mèdica dels seus pacients des de qualsevol lloc i en qualsevol instant ajudant-los a prendre una decisió correcta en la diagnosi i tractament del seus pacients.

Fer accessible la informació a través de la xarxa comporta uns riscs que s'han d'eliminar i la història mèdica d'un pacient conté dades personals que en compliment de la LOPD s'han de protegir i per tant s'ha de garantir la integritat, autenticitat i confidencialitat d'aquestes dades.

La solució que es proposa és el disseny i desenvolupament d'un esquema criptogràfic per gestionar de forma segura els historials mèdics dels pacients a través d'una xarxa de comunicacions.

Com a resultat s'obtenen dues aplicacions amb una interfície gràfica. Els pacients i metges poden utilitzar l'aplicació client per accedir als historials mèdics de forma segura i amb l'aplicació servidor un gestor pot gestionar els historials i les dades dels pacients i metges.

Paraules clau

- Seguretat
- Protecció de dades
- Historials mèdics

Nom de l'àrea de PFC

PFC-Seguretat informàtica

Índex de continguts

Índex de figures

1 Introducció

Les tecnologies de la informació i les comunicacions (TIC) en els últims anys han tingut cada cop més presència en tots els aspectes de la vida laboral i personal de les persones i han transformat aquests profundament jugant un paper molt important en el desenvolupament de la societat i en camps com l'educació, la salut i la seguretat pública. Les TIC han transformat moltes de les formes en que treballem i conduïm les nostres vides diàries i ens permet ser més productius en la feina, millorar la gestió i producció de processos, fer créixer econòmicament les empreses, ser més competitius i ens ofereixen noves oportunitats de negoci.

Utilitzant les TIC en el camp de la sanitat i aprofitant que les xarxes de comunicacions ens permeten accedir a un gran volum de dades molt ràpidament sense la necessitat de desplaçar-nos, i amb independència de l'instant de temps, es pot permetre a un metge consultar la història mèdica d'un pacient des de qualsevol lloc i en qualsevol instant. Aquestes dades poden ajudar al metge a prendre una decisió correcta en la diagnosi i tractament del pacient.

La història mèdica d'un pacient es compon de les dades del seu expedient i les seves visites. Aquestes dades mèdiques són dades de caràcter personal i com a tals han de ser tractades complint la llei orgànica de protecció de dades de caràcter personal (LOPD 15/1999 de 13 desembre) i per tant garantint la seva integritat, autenticitat, confidencialitat i control d'accés.

En aquest projecte s'implementarà un esquema criptogràfic que garantirà les necessitats de seguretat dels historials mèdics que podran ser gestionats de forma remota a través d'una xarxa de comunicacions.

1.1 Justificació del PFC

Si es permet a un metge consultar la història mèdica d'un pacient des de qualsevol lloc, en qualsevol instant, s'aconseguirà optimitzar l'atenció mèdica que rep el pacient estalviant temps i facilitant la diagnosi i tractament. També s'aconseguirà un estalvi de temps al permetre a un pacient consultar la seva història mèdica de forma remota.

El tractament de dades de caràcter personal s'ha de fer conforme la llei orgànica 15/1999 de 13 de desembre de protecció de dades de caràcter personal (LOPD). Aquesta llei estableix 3 nivells de seguretat en aquestes dades. Les que fan referència a dades mèdiques s'han de tractar en el nivell més alt de seguretat. També s'estableix que la transmissió d'aquestes dades a través de xarxes públiques o xarxes sense fils de comunicació electrònica s'hauran de fer xifrant les dades garantint la seva confidencialitat, integritat i autenticitat. També estableix que hi ha d'haver un registre d'accessos.

1.2 Objectius del PFC

L'objectiu del PFC és implementar un esquema criptogràfic que garanteixi les necessitats de seguretat de confidencialitat, autenticitat, integritat i no-repudi de les dades d'una història mèdica que pugui ser gestionada a través d'una xarxa de comunicacions per un metge, pacient o gestor.

Per poder fer aquesta gestió es realitzaran 3 aplicacions:

- Aplicació metge que utilitzarà un metge per accedir de forma segura al gestor del sistema i que el permetrà:
    - Autenticar el metge contra el gestor del sistema.
    - Consultar les dades generals d'un pacient.
    - Obtenir les dades de qualsevol de les visites d'un dels seus pacients.
    - Afegir una visita a l'historial d'un dels seus pacients.
    - Obtenir una llista dels pacients que té assignats.
    - Abandonar de forma segura el sistema.
- Aplicació pacient que utilitzarà un pacient per accedir de forma segura al gestor del sistema i que el permetrà:
    - Autenticar el pacient contra el gestor del sistema
    - Realitzar una consulta de les seves dades generals.
    - Obtenir les dades de qualsevol de les seves visites.
    -Abandonar de forma segura el sistema
- Aplicació gestor del sistema que gestiona el repositori d'historials mèdics de forma central i permetrà:
    -Registrar a nous usuaris (metges o pacients)
    - Autenticar als pacients i als metges que volen accedir al repositori.
    - Acceptar les consultes dels metges i dels pacients.
    -Guardar de forma segura els historials mèdics dels pacients.
    -Verificar que les dades que s'han inserit o modificat en un historial mèdic s'ha realitzat per un usuari autoritzat.
    - Permetre que els usuaris abandonin el sistema de forma segura.

1.3 Enfocament i mètode seguit

La gestió de les dades dels historials mèdics inclouen les accions de consultar dades generals d'un pacient, consultar les visites d'un pacient, consultar els pacients assignats a un metge i afegir una visita a l'historial mèdic.

Per cada una d'aquestes accions caldrà dissenyar un protocol criptogràfic que necessitarà que els usuaris i el gestor del sistema disposin d'una parella de claus i el seu corresponent certificat.

Per tal de gestionar aquests certificats dels usuaris s'utilitzarà una infraestructura de clau pública (PKI).

Per tant, el primer que es farà serà construir una PKI utilitzant openSSL.

Posteriorment s'implementarà l'esquema criptogràfic, format pels protocols que s'han definit per les accions, de forma lineal, és a dir, sense comunicació entre les diferents parts.

Per a representar i fer les transferències de dades que s'envien durant l'execució dels protocols criptogràfics s'utilitzarà XML.

Els usuaris per tal de sol·licitar un servei s'hauran de comunicar amb el gestor del sistema. Per fer-ho s'utilitzaran sockets.

Un cop enllestida la comunicació entre els components, mitjançant mySQL es crearà una base de dades per emmagatzemar la informació relativa als historials mèdics.

Per tal de que els usuaris puguin realitzar les seves accions, es construirà una interfície, pel metge, una pel pacient i una altra pel gestor del sistema.

A mesura que es vagi desenvolupant cadascuna d'aquestes parts, s'anirà provant el sistema paral·lelament per tal que al final s'obtingui un complet joc de proves.

1.4 Planificació del projecte

Cadascuna de les parts identificades es planifiquen de tal manera que s'obté la següent temporalització:

21 de setembre – 3 d'octubre: Instal·lació de l'eina de desenvolupament Eclipse i del IAIK i construcció de la PKI

4 d'octubre – 24 d'octubre: Construcció de l'esquema criptogràfic

25 d'octubre – 7 de novembre: Creació de l'estructura XML per representació de les dades

8 de novembre – 28 de novembre: Muntar sistema de comunicació

29 de novembre – 12 de desembre: Creació i ús de la base de dades

13 de desembre – 2 de gener: Construcció de la interfície client i servidor

3 de gener – 13 de gener: Conclusions i treball futur

1.5 Productes obtinguts

Resultat de l'elaboració del projecte s'obtindran els següents productes:

- aplicació metge: permetrà que un metge pugui consultar i afegir informació a l'historial d'un pacient de forma segura.
- aplicació pacient: permetrà que un pacient pugui consultar les dades del seu historial.
- aplicació gestor central: permetrà donar d'alta al sistema els pacients i els metges, assignar un pacient a un metge i donar accés a un metge a la història d'un pacient.

- Base de dades: on es guardaran totes les dades tals com els historials mèdics, pacients i metges.

1.6 Breu descripció dels altres capítols

Els següents seran els capítols corresponents a cada una de les parts explicades:

- PKI, on s'explicarà la implantació de la infraestructura de clau pública.
- Organització de la informació, on s'explicarà com s'organitza la informació i quines dades cal protegir.
- esquema criptogràfic, s'implementarà l'esquema criptogràfic de forma lineal. Cada tipus d'acció o servei és realitzarà d'acord a un protocol criptogràfic.
- Representació de la informació (XML), on es dissenyarà i implementarà els documents XML que es faran servir per representar les dades.
- Comunicació dels components, on es dissenyarà un protocol de comunicació.
- Gestió de la informació: BD, on es dissenyarà i s'implementarà la base de dades del sistema
- Interfície del client i servidor, on es crearan les interfícies de les aplicacions del pacient, del metge i del gestor del sistema que permetran realitzar les funcionalitats que es recullen en el projecte.

2 Infraestructura de clau pública (PKI)

Per garantir la integritat, confidencialitat, autenticitat i no-repudi de les dades es pot fer servir un sistema que utilitzi xifres de clau pública. Utilitzant xifres de clau pública només es necessiten 2 claus per cada usuari. (La seva clau pública i privada)

Cal però utilitzar algun sistema basat en clau pública que eviti un atac 'de l'home a mig camí' per tal que un intrús no es pogués fer passar pel interlocutor del usuari.

L'ús de la infraestructura de clau pública (PKI) resoldrà aquest problema.

2.1 Conceptes bàsics

La criptografia de clau pública permet el intercanvi de missatges de forma segura entre 2 interlocutors sempre que disposem de la clau pública del interlocutor amb qui ens comuniquem.

Un intrús podria substituir la clau pública d'un usuari per fer-se passar per ell quan es comunica amb algú altre.

Per evitar aquest problema s'utilitzen certificats digitals. Un 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 dóna validesa. (Universitat Oberta de Catalunya, 3)

Per tant hi ha una entitat, que és l'autoritat de certificació (CA) que signa la informació assegurant que la relació del propietari i de la seva clau pública és veraç.

L'objectiu d'una infraestructura de clau pública 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 infraestructura de clau pública crea un marc segur d'intercanvi de dades en un entorn típicament insegur com internet.

Una infraestructura de clau pública (PKI) és el conjunt de maquinari, programari, persones, polítiques i procediments necessaris per a crear i gestionar certificats digitals basats en criptografia de clau pública.

El certificat és l'element central de la infraestructura de clau pública 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)

2.2 Implantació de la PKI

Cada protocol criptogràfic implementat en aquest projecte necessita que els usuaris i el gestor del sistema disposin d'una parella de claus i el seu corresponent certificat.

Per tal de gestionar els certificats (emissió, revocació, etc.) d'un grup d'usuaris s'empra una infraestructura de clau pública. Típicament per fer referència a una infraestructura s'utilitzen les sigles PKI, que corresponen al terme en anglès Public Key Infrastructure.

Una PKI consta d'una autoritat de certificació notada amb CA, aquestes sigles corresponen al terme en anglès Certification Authority. Un altre component de la PKI són les autoritats de registre, notades amb les sigles RA (Registry Authority). 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 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 una peça d'informació molt sensible, i per això està en un entorn amb un nivell alt de seguretat.

En aquest cas es construeix una petita PKI amb openssl seguint els passos següents:

- Es genera la parella de claus de CA (2048 bits). Això es fa amb l'script "generarClaus". S'anomena CA.key l'arxiu amb les claus.
- Es genera un certificat autosignat amb la parella de claus de la CA, CA.key. Aquest serà el certificat de la CA. Es farà servir l'script "generaCertificatAutosignat" i la parella de claus del punt anterior. S'anomenarà CA.crt l'arxiu del certificat.
- Es genera una parella de claus per l'usuari. Per fer-ho es torna a fer servir l'script "generarClaus". La longitud de les claus són de 1024 bits.
- S'emet una petició de certificat utilitzant l'script "generaPeticioCertificat".
- S'emet el certificat utilitzant l'script "generaCertificat" i el fitxer de configuració "openssl.cnf".
- Es genera l'arxiu PKCS12 que contindrà, la parella de claus de l'usuari, el certificat del usuari i el certificat de la CA utilitzant l'script "generaPKCS12".
- Es genera una parella de claus pel gestor del sistema. Per fer-ho es torna a fer servir l'script "generarClaus". La longitud de les claus són de 1024 bits.
- S'emet una petició de certificat utilitzant l'script "generaPeticioCertificat".
- S'emet el certificat utilitzant l'script "generaCertificat" i el fitxer de configuració "openssl.cnf".
- Es genera l'arxiu PKCS12 que contindrà, la parella de claus del gestor, el certificat del gestor i el certificat de la CA utilitzant l'script "generaPKCS12".

3 Organització de la informació

La informació que tracta l'aplicació consisteix bàsicament amb els historials dels pacients, les visites dels pacients i en els metges. D'aquesta informació hi ha dades que s'han de protegir segons la sensibilitat d'aquestes. A continuació es veu una descripció d'aquesta informació indicant quines dades s'han de protegir i com.

3.1 Historials

En un historial hi ha les dades generals del pacient i també totes les visites que realitza el pacient. Les dades generals són dades que qualsevol metge pot consultar i per tant no cal protegir. En canvi les visites sí que són dades més sensibles a les quals només ha de poder accedir el metge o metges que ho siguin dels pacients als que pertany aquestes visites, i per tant és informació que cal protegir. Per mantenir aquesta informació, referent a la relació dels metges que té el pacient, al historial també s'inclou una llista dels metges que poden accedir a aquell historial i també aquesta és una informació protegida perquè ningú, excepte el Gestor, pugui modificar-la.

A continuació es descriuen breument aquestes dades:

Les dades generals del historial consisteixen en el nom i cognoms del pacient, el número de la seva targeta sanitària, el seu DNI, el grup sanguini, les al·lèrgies que pugui tenir i el seu certificat digital.

La llista de visites protegida està formada per:

- llista de descriptors de visita signada pel Gestor i xifrada per una clau de sessió
- llista d'accés, que és una llista de criptogrames fruits de xifrar la clau de sessió, que es fa servir per xifrar la llista descriptors de visita, amb les diferents claus públiques dels metges que tenen autorització per accedir al historial, amb la clau pública del pacient i la del Gestor
- signatura fruit de signar la llista de descriptors de visita xifrada.
- vector d'inicialització utilitzat en el xifratge de la llista de descriptors de visita

La llista de metges protegida és una llista amb els metges que poden accedir a l'historial del pacient i fins quan poden accedir-hi. Aquesta llista està xifrada amb la clau pública del Gestor (sobre digital) i permet a aquest verificar si el metge que demana una visita hi té realment accés.

3.2 Visites

La base de dades conté totes les visites que han realitzat els metges. Aquestes no contenen cap informació que les pugui vincular a un pacient. Cada visita està formada per un descriptor de visita, les dades de la visita i per la signatura digital del metge del descriptor de visita i de les dades de la visita. A continuació es descriuen breument aquestes dades:

- El descriptor de visita conté la informació bàsica d'una visita, que consisteix en la data de la visita, hora, el tema que tracta la visita, el metge que la realitza i la identificació de la visita, que consisteix en un valor únic i aleatori.
- Dades de la visita que consten de l'anamnesi (antecedents personals, hereditaris, familiars...etc, el diagnosi i el tractament d'aquesta.

3.3 Metges

La base de dades conté la informació de tots els metges. Aquests només els gestiona el Gestor i es caracteritzen amb el seu nom i cognoms, número de col·legiat, DNI, especialitat, certificat digital i amb la llista de pacients que té el metge assignats. Aquesta llista de pacients, és una informació que cal protegir especialment i per tant és una llista que està signada amb la clau privada del Gestor i xifrada amb la clau pública del Gestor i del metge (sobre digital).

4 Implementació dels protocols

A continuació es descriuen els protocols que es creen per cada servei que ofereix l'aplicació i que s' utilitzen en la comunicació entre l'aplicació utilitzada pels metges i pacients (Client), i el utilitzat pel Gestor (Servidor) .

4.1 Descripció dels protocols

El sistema l'utilitzen metges, pacients i el gestor.

Els serveis que ofereix són:

- Autenticació dels usuaris: cada usuari s'ha d'autenticar de forma segura així com el seu tipus d'usuari per poder distingir si és un metge, pacient o gestor.
- Registre d'usuaris: el gestor del sistema ha de poder donar d'alta als pacients i als metges al sistema.
- Consultes:
    - Consulta de les visites d'un pacient:
        - Pacients: Els pacients han de poder consultar el seu historial.
        - Metges: Els metges han de poder consultar les visites de l'historial dels seus pacients. Opcionalment es podria donar accés a l'historial d'un pacient a un metge que no és pacient seu. Per exemple, el metge del pacient pot sol·licitar l'opinió d'un altre metge.
    - Consulta de les dades generals d'un pacient: qualsevol metge pot accedir a les dades generals d'un pacient registrat.
    - Consulta de la llista de pacients: qualsevol metge pot obtenir una llista amb els pacients que té assignats.
- Afegir visita: els metges poden afegir noves dades (als historials dels seus pacients).
- Modificació de dades dels pacients:
    - El gestor ha de poder modificar les dades generals dels pacients
    - Els metges no han de poder modificar les dades sobre les visites, diagnosi o tractaments dels seus pacients.
- Eliminació de dades:
    - El metge no ha de poder eliminar dades de l'historial del pacient.
    - El gestor del sistema ha de poder afegir o eliminar usuaris del sistema, ja siguin pacients o metges.

4.2 Disseny de l'esquema criptogràfic

Per cada servei que hem comentat dissenyem un protocol criptogràfic.

La notació emprada per detallar els protocols és la 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.
- PEntitat[M]: Xifratge del missatge M amb la clau asimètrica pública PEntitat d'Entitat.
- H(M): sortida d'una funció resum criptogràfica del missatge M, aquestes funcions reben el nom de funcions hash.

4.2.1 Identificació

El sistema gestor permet que un usuari pugui realitzar una acció o una altra segons quina és la seva identitat. Per exemple si és un metge podrà consultar les dades generals de qualsevol pacient. Si és un pacient només podrà consultar les seves dades.

Per fer l'autenticació es fa servir un certificat de l'usuari. El camp Organizational Unit Name indica si l'usuari és un metge o pacient i el camp dnQualifier conté l'identificador d'usuari.

4.2.2 Autenticació

Cada cop que un usuari vol realitzar una acció s'autentica. En el mateix procés d'autenticació l'usuari sol·licita l'acció que vol realitzar.

També es podria haver optat per fer la identificació només una vegada al principi, però s'hauria de mantenir un estat de connexió de manera que el gestor puogués saber a quin usuari correspon cada connexió, tot i que seria més eficient.

4.2.3 Consulta de les dades generals d'un pacient

Al protocol 1 cada usuari U s'identifica amb Id_usuari_U i disposa d'una parella de claus (P_U,S_U) amb el corresponent certificat $Cert_U$. En el cas del gestor G el seu identificador d'usuari Id_usuari_G és el hash del certificat. El Protocol 1 pot ser utilitzat per un metge o per un pacient. G verifica en cada cas el tipus d'usuari i només facilita l'historial si l'usuari hi té accés.

Aquest protocol l'utilitza un usuari per demanar les seves dades generals o un metge per consultar les dades generals de qualsevol pacient.

Protocol 1

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) Xifrar N_G, Consulta_dades_generals, Id_usuari i Id_usuari_U amb la clau pública P_G de G, $P_G[N_G$,Consulta_dades_generals,Id_usuari, $Id\_usuari_U]$.Consulta_dades_generals indica que es volen consultar les dades generals de l'usuari identificat amb Id_usuari;
      ii) Enviar $P_G[N_G$,Consulta_dades_generals,Id_usuari, $Id\_usuari_U]$ a G;
   c) Sinó retornar error;
4) G realitza les operacions següents:
   a) Desxifrar $P_G[N_G$,Consulta_dades_generals,Id_usuari, $Id\_usuari_U]$ amb la clau privada S_G, i obtenir N_G', Consulta_dades_generals, Id_usuari, Id_usuari_U;
   b) Recuperar N_G de la BD a partir del Id_usuari_U. 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:
      i) Si (Id_usuari_U = Id_usuari) o (Id_usuari_U és metge) fer:
         (1) Executar el Procedure 3 amb Id_usuari i P_U, i obtenir $P_U[H,iv]$;
         (2) Enviar $P_U[H,iv]$ a U.
      ii) Sinó retornar error;
   d) Sinó retornar error;
   e) Esborrar N_G i N_i de la BD;
5) U realitza les operacions següents:
   a) Executar el Procedure 4 amb $P_U[H,iv]$, i obtenir H;
   b) Mostrar H.

El Procedure 1 conté una part de l'autenticació del protocol de Needham-Schroeder. Aquesta part és executada pels metges i pacients. Es necessita enviar el vector d'inicialització (iv) utilitzat per xifrar la llista de descriptors de visita, perquè l'usuari el necessita per poder-ho desxifrar.

Procedure 1 (P_U)

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.

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

Procedure 2 ($P_G(N_i,Id\_usuari_U)$)

1) Desxifrar $P_G[N_i,Id\_usuari_U]$ amb S_G, i obtenir; N_i i Id_usuari_U;
2) Obtenir el certificat U a partir de Id_usuari_U. Suposem que 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;
5) Xifrar N_i, N_G,Id_usuari_G amb la clau pública P_U de U, $P_U[N_i,N_G,Id\_usuari_G]$;
6) Retornar $P_U[N_i,N_G,Id\_usuari_G]$.

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

Procedure 3 (id_usuari, P_U)

1) Buscar l'historial H corresponent a id_usuari;
2) Xifrar H amb la clau pública P_U, $P_U[H]$;
3) Retornar $P_U[H,iv]$.

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

Procedure 4

1) Desxifrar $P_U[H,iv]$ amb la clau privada S_U de U, $S_U[P_U[H]]$;
2) Desxifrar una de les entrades de la llista d'accés i obtenir la clau de sessió;
3) Desxifrar la llista de descriptors de visites xifrada;
4) Verificar la signatura digital de G sobre la llista dels descriptors de visites xifrada;
5) Retornar H

4.2.4 Consulta d'una visita d'un pacient

En el Protocol 2 l'usuari U s'identifica amb Id_usuari_U, on l'usuari pot ser un metge o un pacient. G verifica en cada cas el tipus d'usuari i només facilita l'historial si l'usuari hi té accés.

En aquest cas pot ser que l'usuari demani una de les seves visites o que l'usuari és un metge que té accés a les visites del pacient.

Protocol 2

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) Xifrar N_G, Consulta_visita, Id_usuari, descriptor_de_visita, Id_usuari_U amb la clau pública P_G de G, $P_G[N_G$,Consulta_visita,Id_usuari,descriptor_de_visita, $Id\_usuari_U]$.Consulta_visita indica que es vol consultar la visita identificada per descriptor_de_visita de l'usuari identificat amb Id_usuari;
ii) Enviar $P_G[N_G$,Consulta_visita,Id_usuari,descriptor_de_visita, $Id\_usuari_U]$ a G;

c) Sinó retornar error;

4) G realitza les operacions següents:
   a) Desxifrar $P_G[N_G$,Consulta_visita,Id_usuari,descriptor_de_visita, $Id\_usuari_U]$ amb la clau privada S_G, i obtenir N_G', Consulta_visita, Id_usuari, descriptor_de_visita, Id_usuari_U;
   b) Recuperar N_G de la BD a partir de Id_usuari_U. 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:
      i) Si Procedure 5 [Id_usuari_U,Id_usuari,descriptor_de_visita] retorna que totes les verificacions són correctes fer:
         (1) Obtenir la visita identificada per descriptor_de_visita (V) i calcular $P_U[V]$;
         (2) Enviar $P_U[V]$ a U.
      ii) Sinó retornar error;
   d) Sinó retornar error;
   e) Esborrar N_G i N_i de la BD;
5) U realitza les operacions següents:
   a) Desxifrar $P_U[V]$, i obtenir V;
   b) Mostrar V.

Procedure 5 (Id_usuari_U, Id_usuari, descriptor_de_visita)

1) Si (Id_usuari_U = Id_usuari) fer:
   a) Verificar si descriptor_de_visita està dins de la llista_de_descriptors_de_visita_xifrada de l'usuari identificat per Id_usuari_U.
   b) Retornar el resultat de la verificació.
2) Si (Id_usuari_U) és un metge fer:
   a) Verificar si Id_usuari està dins de la llista_de_pacients_protegida pel metge identificat per Id_usuari_U;
   b) Verificar si el metge identificat per Id_usuari_U està a la llista_de_metges de l'usuari identificat per Id_usuari_U;
   c) Verificar si el descriptor_de_visita pertany a l'usuari Id_usuari emprant la llista llista_de_visites_protegida de l'usuari.
   d) Retornar el resultat de la verificació.

4.2.5 Consulta dels pacients assignats a un metge

Una operació típica per un metge és buscar l'historial d'un dels seus pacients.

Amb el Protocol 3 un metge pot obtenir el llistat dels seus pacients.

Protocol 3

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) Xifrar N_G i llista_pacients, Id_usuari_U amb la clau pública P_G de G, $P_G[N_G,$ Llista_pacients, $Id\_usuari_U]$.La Llista_pacients indica que es vol un llistat dels pacients del metge identificat amb Id_usuari_U;
      ii) Enviar $P_G[N_G,$ Llista_pacients, $Id\_usuari_U]$ a G;
   c) Sinó retornar error;
4) G realitza les operacions següents:
   a) Desxifrar $P_G[N_G,$ Llista_pacients, $Id\_usuari_U]$ amb la clau privada S_G, i obtenir N_G', Llista_pacients i Id_usuari_U;
   b) Recuperar N_G de la BD a partir de , Id_usuari_U. 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:
      i) Si Id_usuari_U és metge fer:
         (1) Calcular P_U[llista_pacients_protegida];
         (2) Enviar a U P_U[llista_pacients_protegida].
      ii) Sinó retornar error;
   d) Sinó retornar error;
   e) Esborrar N_G i N_i de la BD;
5) U realitza les operacions següents:
   a) Desxifrar P_U[llista_pacients_protegida] i obtenir llista_pacients_protegida;
   b) Tractar la llista_pacients_protegida i mostrar la llista de pacients a l'usuari.

4.2.6 Afegir una visita a l'historial mèdic

En aquest protocol se suposa que prèviament a la inserció de les dades el metge U ha consultat l'historial del pacient P i per tant coneix Id_usuari_P. El protocol 4 està pensat únicament per afegir una nova visita V a l'historial. El gestor un cop rep una visita V d'un pacient P verifica que ha estat signada pel metge U assignat al pacient. A continuació afegeix el descriptor de la visita a la llista de descriptors protegida i la visita a la Base de Dades.

Protocol 4

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) Obtenir les dades de la visita V;
      ii) Signar V amb la clau privada S_U de U, $S_U[V]$;
      iii) Xifrar N_G,Afegir_visita,V,Id_usuari, $S_U[V]$ i Id_usuari_U amb la clau pública P_G de G, $P_G[N_G$,Afegir_visita,V,Id_usuari,$S_U[V]$, $Id\_usuari_U]$. Afegir_visita indica que es vol afegir V a l'historial del pacient P identificat per Id_pacient;
      iv) Enviar $P_G[N_G$,Afegir_visita,V,Id_usuari,$S_U[V]$, $Id\_usuari_U]$ a G;
   c) Sinó retornar error;
4) G realitza les operacions següents:
   a) Desxifrar $P_G[N_G$,Afegir_visita,V,Id_usuari,$S_U[V]$, $Id\_usuari_U]$ amb la clau privada S_G, i obtenir N_G' , Afegir_visita,V, Id_usuari, $S_U[V]$ i Id_usuari_U;
   b) Recuperar N_G de la BD a partir de Id_usuari_U. 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:
      i) Verificar que Id_usuari_U és metge;
      ii) Verificar que Id_usuari és un pacient assignat a Id_usuari_U;
      iii) Si les verificacions anteriors són correctes fer:
         (1) Verificar la signatura digital $S_U[V]$ amb la clau pública P_U;
         (2) Obtenir el descriptor_de_la_visita de V;
         (3) Afegir el descriptor_de_la_visita a la llista_descriptors_de_visites;
         (4) Signar amb la clau privada del Gestor P_G la llista_descriptors_de_visites;
         (5) Xifrar la llista_descriptors_de_visites amb una clau de sessió K, E_K(llista_descriptors_de_visites);
         (6) Xifrar la clau de sessió K amb les claus públiques dels metges que estan a la llista_de_metges de l'historial de l'usuari identificat amb Id_usuari, amb la clau pública del Gestor i amb la clau pública del pacient obtenint una nova llista_descriptors_visites_protegida;
         (7) Afegir V a la Base de Dades.
      iv) Sinó retornar error;
   d) Sinó retornar error;
   e) Esborrar N_G i N_i de la BD;

5 Representació de la informació

Tal i com s'ha definit en els protocols es necessita enviar dades del client (usuari) al servidor (gestor) i a l'inversa. Aquesta informació s'envia a través de la xarxa, i és una informació que té una estructura concreta, i que es necessita poder-la rebre tal i com s'envia recuperant la mateixa estructura. L'XML està dissenyat específicament per a representar i transportar informació estructurada com la que es pot guardar en una base de dades. Es tracta d'un format llegible del qual es pot extreure molt fàcilment la informació al rebre-la i també generar-la al enviar-la i és independent de la plataforma utilitzada. Totes aquestes

característiques satisfan sobradament les necessitats que es tenen i per tant la informació es representa utilitzant aquest llenguatge.

XML és l'acrònim de eXtensible Markup Language (llenguatge d'etiquetatge extensible).

És un llenguatge de marques o etiquetes com les que s'utilitzen en el HTML, però a diferència d'aquest no es centra en la presentació de les dades sinó en la seva representació. Mitjançant aquestes etiquetes es pot definir aquesta informació utilitzant un vocabulari concret que serveix per definir les dades, vocabulari que pot ser acordat i compartit per una comunitat o organització, utilitzant unes restriccions d'ús que permeten restringir quan i quins valors es poden utilitzar per aquestes dades (elements). Aquest llenguatge de marques o etiquetes està basat en l'alfabet universal Unicode, per tant, accepta textos en la majoria de llengües. (2)

5.1 Disseny del document XML

L'XML s'utilitza per codificar (seriar) les dades que s'intercanvien els usuaris amb el gestor del sistema durant l'execució dels protocols criptogràfics que s'han definit.

Degut a que l'XML és textual, i es fan servir dades binàries, aquestes s'han d'enviar codificades en format Base64.

Per representar totes les dades que s'han d'enviar no es necessiten gaires camps. Es dissenya un únic document amb un camp 'id' que identifica quin missatge representa.

Depenent del missatge que es tracta s'assignen valors només als elements necessaris per formar aquell missatge.

Els camps que es tenen són els següents:

- Id: identificador del missatge. De entre tots els missatges que hi ha en la especificació dels protocols criptogràfics, l'id indica de quin missatge es tracta.
- AleatoriI: representa la seqüència de bytes aleatòria generada per l'usuari metge o pacient.
- AleatoriG: representa la seqüència de bytes aleatòria generada pel Gestor.
- IdUsuari: representa l'identificador d'un usuari sobre el que es vol realitzar una acció.
- IdUsuariU: representa l'identificador d'un usuari que transmet el missatge.
- Accio: per representar l'acció que vol realitzar un usuari.
- Historial: representa l'historial d'un pacient.
- DescriptorVisita: representa el descriptor d'una visita.
- Iv:vector d'inicialització que s'utilitza per xifrar i desxifrar la llista de dexcriptors de visita.
- Visita: per representar una visita.
- LlistaPacientsProtegida: representa la llista de pacients protegida.
- SignaturaVisita: per representar la signatura d'una visita.

Aquest document XML és el que es xifra i s'envia entre els usuaris i el Gestor.

5.2 Implementació XML

Es desenvolupa la següent classe per fer la gestió XML:

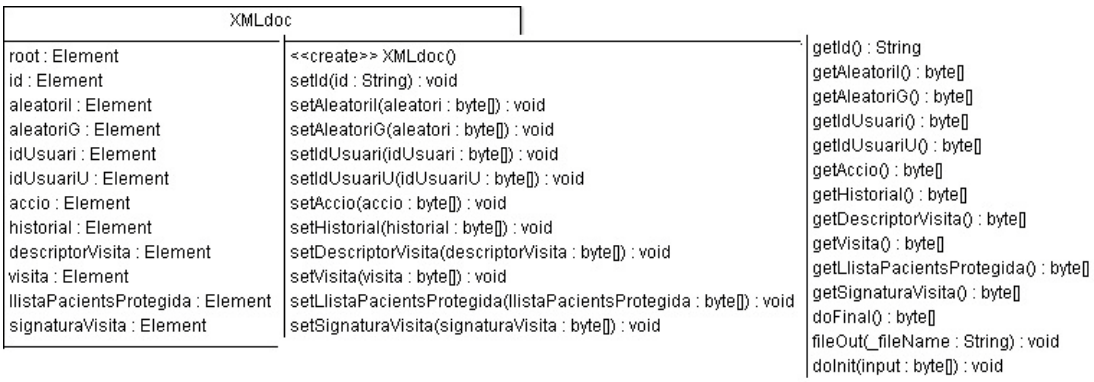

Figura 1 Classe XMLdoc per l'enviament dels missatges

Aquesta classe permet preparar una estructura amb el contingut XML per crear un missatge. Per informar del contingut de cada camp del missatge es fan servir els mètodes *set* per cada camp. Amb el mètode *doFinal* es serialitza l'estructura XML en una cadena de bytes que és el que s'envia per la xarxa. Aquesta classe també permet fer l'operació inversa, a partir d'un missatge rebut obtenir tota la informació de les dades XML que conté. Per fer-ho es fan servir els mètodes *get* per recuperar els valors de cada camp.

6 Comunicacions

Es necessita que el client (metge o pacient) es pugui connectar amb el servidor (gestor) per poder sol·licitar els serveis que vol. El servidor també ha de poder respondre a aquestes sol·licituds. La comunicació per sockets és una manera senzilla de comunicar dos aplicacions ja que fan que l'aplicació pugui veure la xarxa com si es tractés d'un fitxer del que ha de llegir per rebre dades i al que ha d'escriure per enviar-les. Si en un futur es vol programar un client que s'executi des d'un smart-phone, donades les limitacions d'aquests dispositius, utilitzar sockets per la comunicació és el més adequat. Es necessita també un sistema orientat a connexió fent servir el protocol TCP ja que el UDP, que no està orientat a connexió no garanteix l'ordre amb el que es reben els paquets. Per tant per realitzar la comunicació entre els usuaris i el gestor es fan servir sockets de flux (stream sockets).

Un socket és l'extrem d'una comunicació bidireccional entre 2 programes que s'executen en 2 equips connectats en xarxa. Per tant la comunicació es realitza entre 2 sockets. Cada socket va lligat a un port del equip on s'executa. Això permet tenir múltiples connexions utilitzant ports diferents. (1)

Per realitzar la comunicació mitjançant sockets s'utilitza el paquet java.net. Aquest paquet proporciona la classe Socket, que implementa la part del client d'una connexió (la part del usuari) i ServerSocket que implementa un socket en la part del servidor (la part del gestor) perquè aquest escolti i accepti les connexions dels clients.

6.1 Diagrama de seqüència de la comunicació

Cada cop que un usuari realitza un pas d'un protocol és realitza les accions que es representen en el següent diagrama de seqüència:

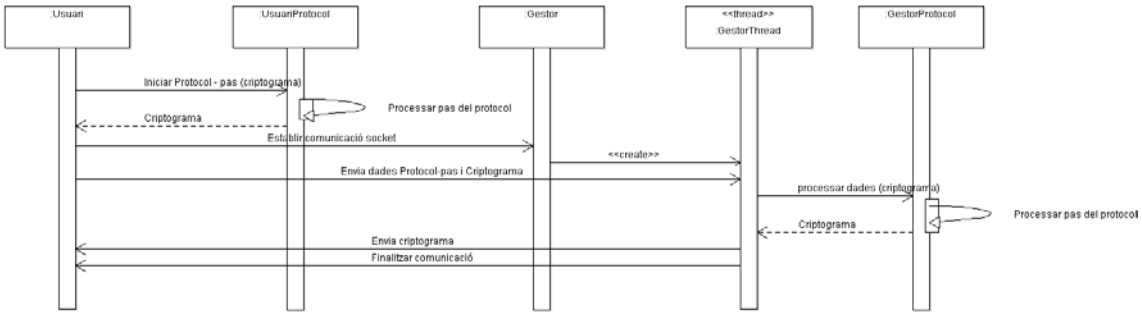

Figura 2 Diagrama de seqüència de la comunicació

Quan un usuari vol executar el pas d'un protocol, envia les dades al objecte 'UsuariProtocol' perquè aquest processi les dades d'aquell pas i protocol en concret i retorni el criptograma generat. Llavors l'usuari estableix una comunicació amb el Servidor (gestor), que alhora delega el tractament de la connexió a un fil de processament (per tal que el Servidor queda lliure per escoltar més peticions pel mateix port) , i li envia el criptograma. El gestor envia les dades al objecte 'GestorProtocol' (indicant-li el pas i protocol en que estem) i aquest processa les dades d'aquell pas i protocol i les retorna al usuari finalitzant la comunicació. Si hi ha hagut algun error les dades que retorna consisteixen amb la descripció del error.

6.2 Implementació de les comunicacions

El Gestor és la part servidora de la comunicació. La classe Gestor manté obert un socket escoltant per un port. Quan rep una petició (que correspon a processar un únic pas d'un protocol) crea el thread GestorThread que estableix un socket amb l'usuari per poder-s'hi comunicar. En aquesta comunicació primer llegeix les dades que li envia l'usuari , el criptograma que rep del usuari el passa a la classe GestorProtocol perquè realitzi totes les accions d'aquell pas del protocol, el criptograma resultant l'envia al usuari i finalment tanca la comunicació.

L'usuari és la part client de la comunicació. Crea un objecte de la classe UsuariProtocol que és qui realitza totes les accions de cada pas del protocol escollit. El criptograma resultant de l'execució d'un pas del protocol l'envia al Gestor. Per tant obre un socket contra el Gestor i li envia les dades. Per cada pas del protocol obre un socket i el tanca, per tant les dades de cada pas s'envien en un socket diferent.

7 Base de dades

Per poder fer persistents les dades que l'aplicació genera i amb les que treballa, es necessita una base de dades on poder-les emmagatzemar. Per això s'utilitza un sistema gestor de base de dades (SGBD), concretament el MySQL que es tracta d'un programari lliure sota llicència GNU GPL.

Per poder-s'hi connectar s'utilitza el driver JDBC Connector/J (versió 5.1.14) que es tracta d'un driver de tipus 4 (driver java pur que implementa protocols de xarxa específics per al SGBD de MySQL i permet connectar-se a la base de dades directament)

7.1 Disseny de la Base de Dades

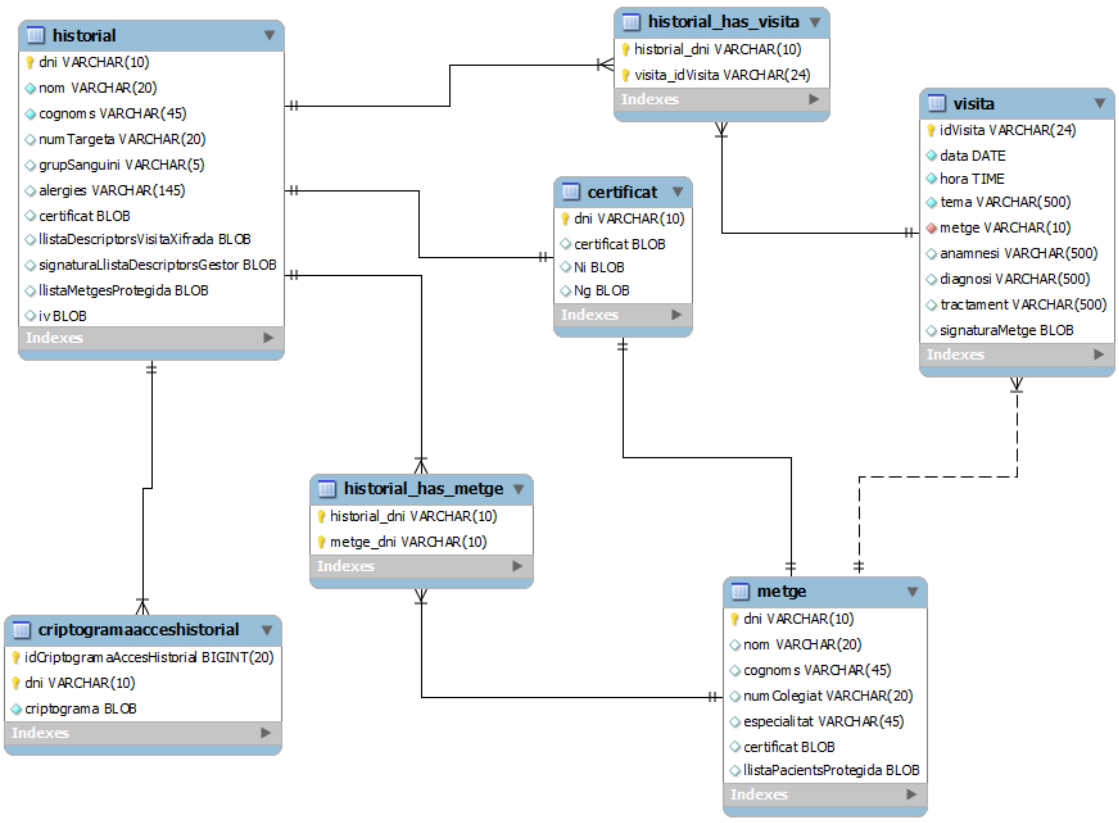

Figura 3 Disseny de la base de dades

Les taules següents de la base de dades permeten emmagatzemar les dades que ja hem descrit a la organització de la informació:

- Historial: conté les dades relatives als historials dels pacients. La llista d'accés s'implementa relacionant la taula historial amb la 'criptogramaacceshistorial'.
- criptogramaacceshistorial: conté cada criptograma fruit de xifrar la clau de sessió que s'utilitza per xifrar la llista de descriptors de visita d'un historial amb les diferents claus

públiques dels metges que tenen autorització per accedir a aquell historial, amb la clau del pacient i la del Gestor.

- certificat: conté els certificats dels pacients i metges i disposa dels camps Ni i Ng per guardar els valors aleatoris que es generen ens els protocols 1 i 2 de la comunicació i que posteriorment es consulten i esborren durant el transcurs de la comunicació.
- visita: conté les dades relatives a les visites. Inclouen aquelles referents al descriptor de la visita, a les dades de la visita i a la signatura digital del metge de les anteriors.
  Les visites es guarden a la base de dades sense cap informació que les permeti vincular a un pacient. Per tant si algú accedís a la base de dades no podria saber quines visites corresponen a quins pacients.
- metge: conté les dades referents al metge que hem descrit anteriorment a la organització de la informació.

7.2 Implementació de la Base de Dades

Per accedir a la base de dades per tal d'inserir, actualitzar o esborrar dades es desenvolupa una classe 'BD' que ofereix mètodes per realitzar aquestes operacions i per gestionar la connexió i desconnexió a aquesta.

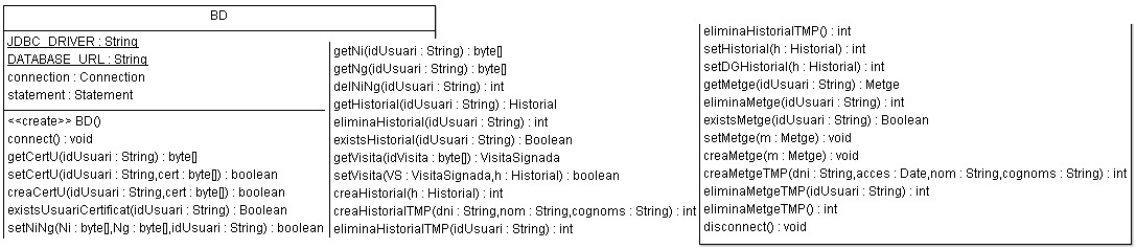

Figura 4 Classe BD per gestionar la base de dades

- connect: mètode per realitzar la connexió a la Base de Dades
- getCertU: mètode per recuperar el certificat d'un usuari de la taula Certificat
- setCertU: mètode per assignar un certificat a un usuari en la taula Certificat
- creaCert: insertar a un usuari en la taula Certificat
- existsUsuariCertificat: comprova si existeix ja un usuari en la taula Certificat
- setNiNg: actualitzar la taula Certificat assignant els valors Ni i Ng
- getNi: recuperar el valor Ni de la taula Certificat d'un usuari
- getNg: recuperar el valor Ng de la taula Certificat d'un usuari
- delNiNg: esborrar valors Ni i Ng de la taula Certificat d'un usuari
- getHistorial: recuperar l'objecte Historial d'un usuari de la taula historial
- eliminaHistorial: Elimina l'historial del usuari indicat de la taula Historial i les dades relacionades de la taula CriptogramaAccesHistorial i de la taula Certificat
- existsHistorial: comprova si existeix un pacient en la taula Historial
- getVisita: recuperar una visita signada
- setVisita: afegir una visita a un historial

- creaHistorial: insertar un pacient en la taula Historial
- setHistorial: actualitzar les dades d'un pacient de la taula Historial
- setDGHistorial: actualitzar només les dades generals d'un pacient de la taula Historial
- getMetge: recuperar el metge indicat
- eliminaMetge: elimina un metge de la taula Metge
- existsMetge: comprova si existeix un metge en la taula Metge
- setMetge: Actualitzar les dades d'un metge
- creaMetge: inserta un metge en la taula Metge
- disconnect: Tancar la connexió amb la Base de Dades

8 Interfície gràfica

Una interfície gràfica ofereix un mecanisme amigable per l'usuari per interactuar amb una aplicació i permet als usuaris familiaritzar-se ràpidament amb aquesta, permetent-los un ràpid aprenentatge i un ús més productiu.

8.1 Decisions de disseny

En Java es disposa de dos conjunts d'eines per dissenyar interfícies gràfiques d'usuari. El AWT (Abstract Window Toolkit) i Swing, aquest introduït posteriorment en la versió 1.2 del J2SE.

El problema del AWT és que les aplicacions amb interfícies gràfiques fetes utilitzant aquestes eines, al executar-se en diferents plataformes, presenten una interfície gràfica amb una aparença pròpia de la plataforma. Per exemple, si l'aplicació s'executa en la plataforma Windows, la interfície gràfica agafarà el mateix aspecte que el que tenen les altres aplicacions típiques de Windows. Algunes vegades la manera en què l'usuari pot interaccionar amb un component AWT en concret també és diferent segons la plataforma en què s'executi l'aplicació.

En canvi, amb els components de la interfície gràfica de Swing es pot especificar una aparença i una manera en que l'usuari interactua amb l'aplicació, uniforme per a totes les plataformes en què l'aplicació s'executa, però també dóna la possibilitat d'escollir utilitzar l'aparença i manera en que l'usuari interactua amb l'aplicació, pròpia de la plataforma tal i com es comporta AWT. Fins i tot Swing permet canviar aquesta aparença i manera d'interactuar durant l'execució de l'aplicació deixant que l'usuari escolli la seva preferència.

Els components Swing estan implementats en Java i per tant són més portables (independents de la plataforma) i flexibles que els primitius components AWT que estaven basats en els components de la plataforma subjacent en la que s'executaven. Per aquesta raó els components Swing són generalment preferentment escollits, i són els que es faran servir.

Així i tot per a la gestió dels esdeveniments dels components i per la gestió de la presentació dels components sí que s'usarà la llibreria java.awt que permet utilitzar-la tant amb components AWT com Swing.

8.2 Disseny de les classes

S'han desenvolupat 2 classes. Una pel servidor (gestor) i l'altre pel client(usuari).

La classe del servidor mostra les pantalles per gestionar els metges, per gestionar els pacients i per configurar el servidor.

Per la gestió de metges i pacients únicament es crida a les funcions de la classe BD per crear-los, modificar-los, eliminar-los, i crear-los el certificat.

Per modificar la llista dels pacients del metge i la llista dels metges del pacient també s'utilitza una funció de la classe BD per modificar el metge i el pacient amb la nova llista desxifrant-la i xifrant-la un cop modificada.

Pel que fa a la configuració és fa ús de la classe XMLfile per guardar en un fitxer XML la configuració del servidor i per poder-la restaurar al tornar a engegar el programa.

La classe del client presenta les pantalles per consultar les dades generals d'un pacient, per consultar una visita, per llistar els pacients del metge, per afegir una visita i per configurar el servidor.

Pel que fa a la configuració, igual que en el servidor, és fa s'usa la classe XMLfile per guardar en un fitxer XML la configuració del servidor i per poder-la restaurar al tornar a engegar el programa.

Per la resta d'accions, cadascuna correspon a un protocol de comunicació. Tal i com es comentava a l'apartat de Comunicació, s'utilitza la classe UsuariProtocol per gestionar cada pas del protocol.

La informació que necessita la classe UsuariProtocol per gestionar cada protocol es proporciona creant un fitxer XML per cada acció. Així que únicament ha de llegir del fitxer XML corresponent al protocol. Per tant hi ha les següents classes:

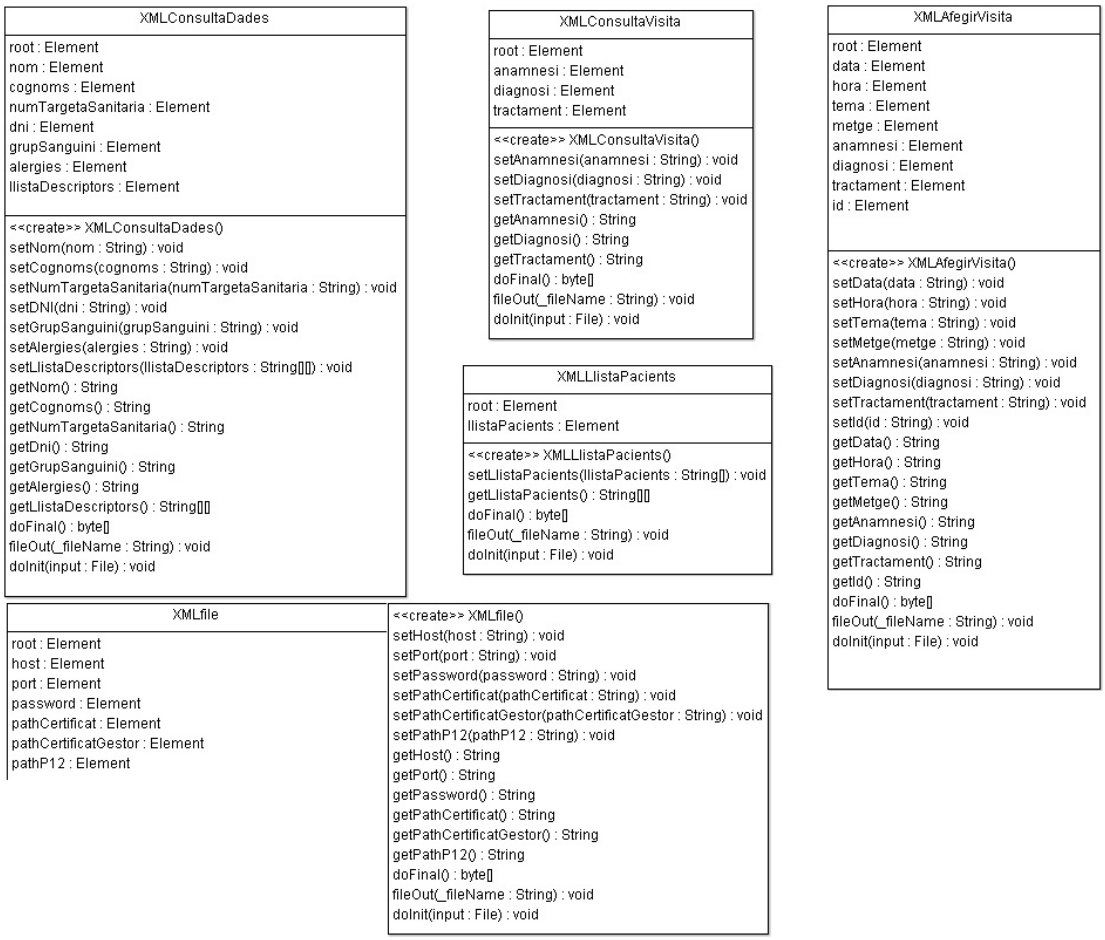

Figura 5 Classes XML de la interfície gràfica

Totes elles tenen mètodes *set* i *get* per poder definir i recuperar els valors dels camps respectivament. El mètode *doFinal* serialitza l'estructura del fitxer XML en una cadena de bytes, el mètode *fileOut* escriu l'estructura XML en un fitxer i el mètode *doInit* crea l'estructura XML del fitxer a partir del fitxer XML ja existent.

Els camps de XMLConsultaDades corresponen a les dades generals dels pacients o metges que es volen consultar. Els camps de XMLConsultaVisita són l'anamnesi, diagnosi i tractament de la visita que es vol consultar. Els de XMLAfegirVisita són els propis de la visita que es vol afegir. El camp de XMLLlistaPacients consisteix únicament en una llista de pacients que consisteixen en elements dni que es creen al fer el *set*.

Per últim els camps de XMLfile es componen de tots aquells camps que es fan servir en la configuració del servidor i del client. Pel servidor són el port per on escolta, el pathCertificat que és la ruta del certificat del Gestor, el pathP12 que és la ruta del fitxer PKCS12 del gestor i password que és la paraula de pas del fitxer P12. Pel client són el port del servidor al que es connecta, el host que és la direcció IP del servidor, el pathCertificat que és la ruta del seu certificat, el pathCertificatGestor que és la ruta del certificat del Gestor, el pathP12 que és la ruta del fitxer PKCS12 del client i el password que és la paraula de pas del fitxer P12, entenent client com el pacient o el metge.

8.3 Funcionalitats

De les funcionalitats del client hi ha algunes que són comunes pels pacients i metges i altres que són específiques pel metge. De les comunes hi ha la configuració del client, la consulta de les dades generals d'un pacient i la consulta de les visites d'un pacient. Cal dir però, que els pacients poden consultar només les seves dades generals i les seves visites. Els metges poden consultar les dades generals de qualsevol pacient i les visites de només pacients seus.

De les funcionalitats específiques pel metge hi ha la consulta de la llista de pacients que té assignats i la d'afegir una visita a l'historial mèdic d'un pacient seu.

Les funcionalitats del gestor són la configuració del servidor, la gestió dels metges (alta, modificació, baixa i assignació de pacients) i la gestió dels pacients (alta, modificació, baixa i assignació de metges).

A continuació es veu cadascuna de les funcionalitats de l'aplicació client:

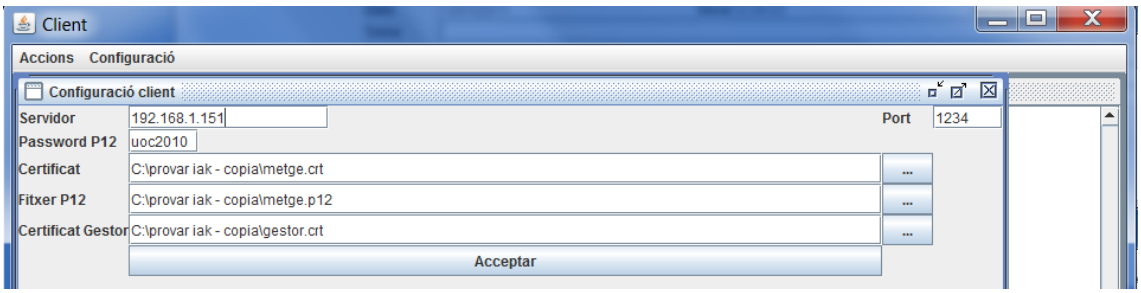

Figura 6 Configuració client

Per configurar el client (menú Configuració -> Configuració client) s'ha d'indicar la direcció IP i el port del servidor, el password del fitxer P12, el certificat del usuari de l'aplicació (del metge o pacient), el fitxer P12 del usuari de l'aplicació (del metge o pacient) i el certificat del gestor.

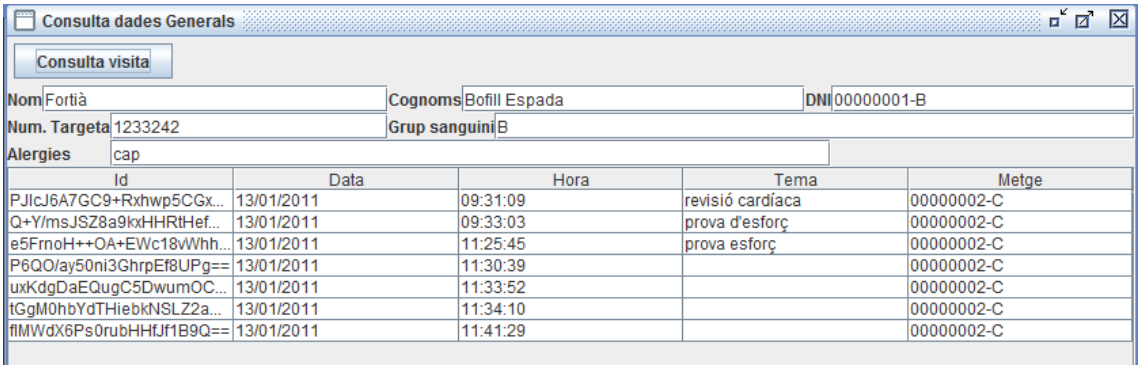

Figura 7 Consulta de les dades generals d'un pacient (demanat pel pacient)

Quan el pacient consulta les seves dades (menú Accions -> Consulta dades generals) veu les seves dades generals (nom i cognoms, dni, número de targeta sanitària, grup sanguini i alergies) i els descriptors de les seves visites.

Per consultar una de les seves visites només l'ha de seleccionar i prémer el botó de 'consulta visita':

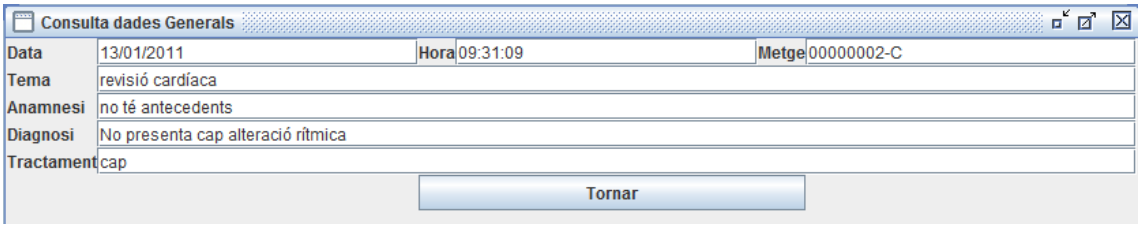

Figura 8 Consulta d'una visita

En el formulari de la consulta d'una visita es mostren les dades de la visita(data i hora en que es fa la visita, metge que la fa i el tema, anamnesi, diagnosi i tractament de la visita).

Quan un metge consulta les dades generals d'un pacient la pantalla que apareix ofereix a més l'opció d'afegir visita tal i com es pot apreciar en la següent figura:

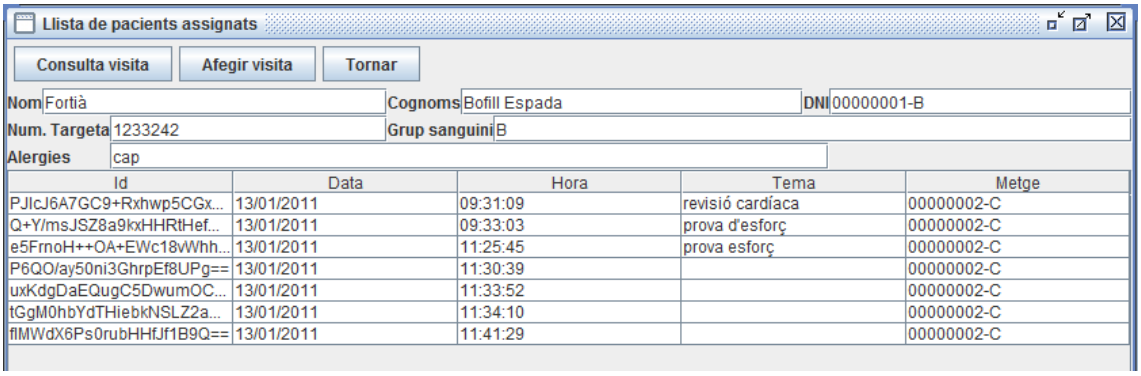

Figura 9 Consulta de les dades generals d'un pacient (demanat pel metge)

Un metge per arribar a consultar les dades d'un pacient ha de consultar primer la llista dels seus pacients (menú Accions -> Llista pacients) , seleccionar el pacient i prémer el botó de la consulta de les dades generals del següent formulari:

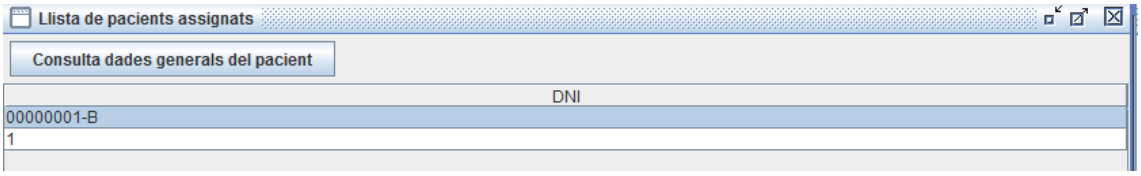

Figura 10 Llista dels pacients

La finestra que se li visualitza al metge quan consulta una visita no té cap diferència amb la que se li presenta al pacient.

El formulari d'afegir una visita és el mateix que el de consultar una visita amb la diferència que es poden acceptar els canvis tal i com es pot veure en la següent pantalla:

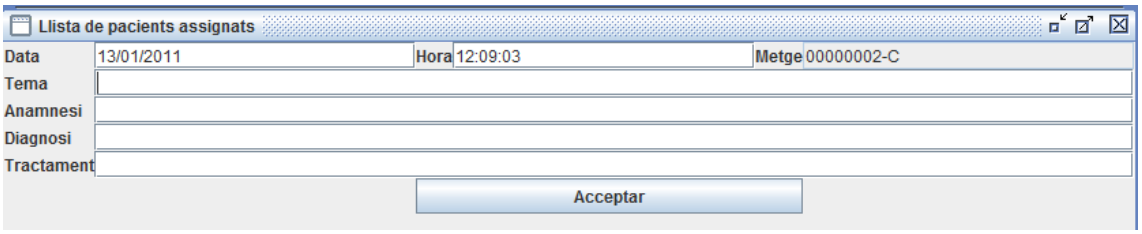

Figura 11 Afegir visita

A continuació es descriuen les funcionalitats de l'aplicació servidor (el gestor):

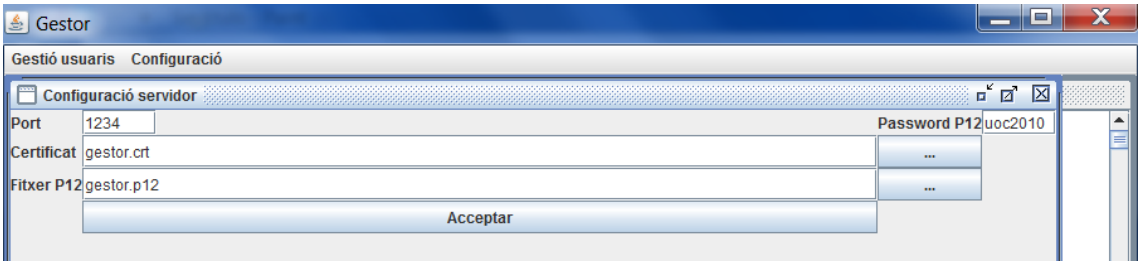

Figura 12 Configuració del servidor

Per configurar el servidor (menú Configuració -> Configuració servidor) el gestor ha d'especificar el port per on escolta el servidor, el password del fitxer P12, el certificat del gestor i el fitxer P12 del gestor.

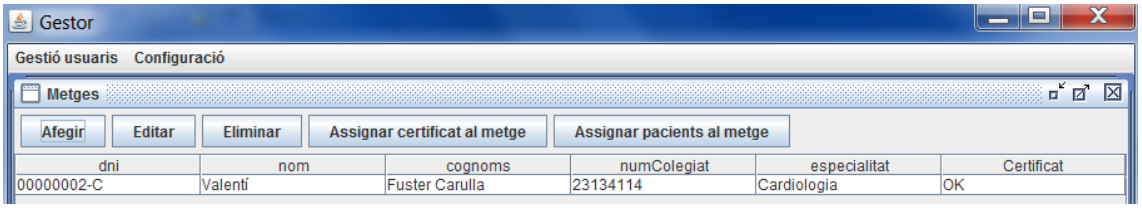

Figura 13 Gestió dels metges

Quan el gestor escull gestionar els metges (menú Gestió usuaris -> Metges) se li mostra una pantalla amb la llista dels metges des d'on pot afegir un de nou, editar un metge de la llista, eliminar un metge de la llista, assignar un certificat al metge i assignar un pacient al metge.

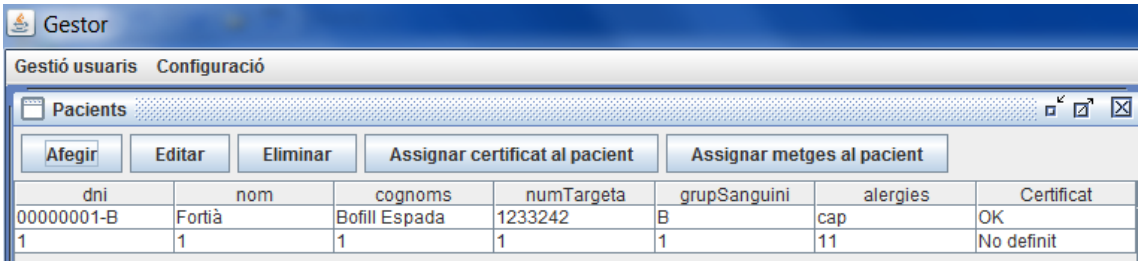

Figura 14 Gestió dels pacients

Quan el gestor escull gestionar el pacients (menú Gestió usuaris -> Pacients) se li visualitza una pantalla amb la llista dels pacients des d'on pot afegir un de nou, editar un pacient de la llista, eliminar un pacient de la llista, assignar un certificat al pacient i assignar un metge al pacient.

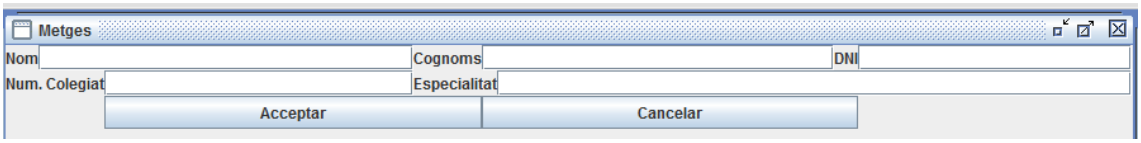

Figura 15 Afegir metge

Tal i com es pot veure en aquesta figura al afegir un metge es presenta una pantalla a on es pot introduir totes les dades del metge (nom, cognoms, dni, número de col·legiat i especialitat del metge).

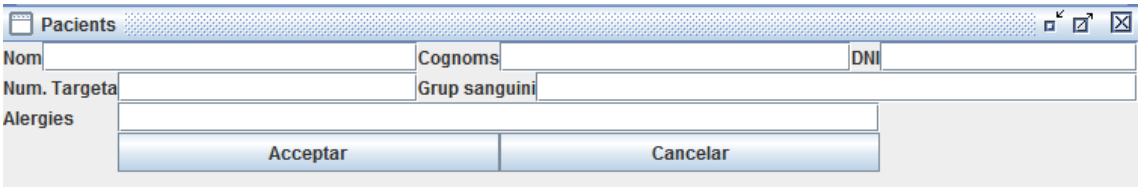

Figura 16 Afegir pacient

De la mateixa manera tal i com es mostra en aquesta altra figura al afegir un pacient es visualitza una pantalla on es pot introduir totes les dades del pacient (nom i cognoms, dni, número de targeta sanitària del pacient, grup sanguini i al·lèrgies).

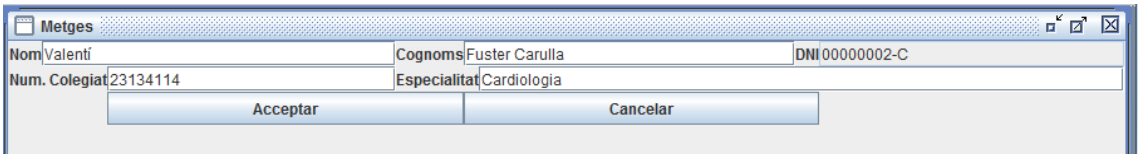

Figura 17 Editar metge

Al editar un metge es presenten les seves dades i permet editar-les totes excepte el DNI. Es pot apreciar en la figura com el camp DNI està deshabilitat.

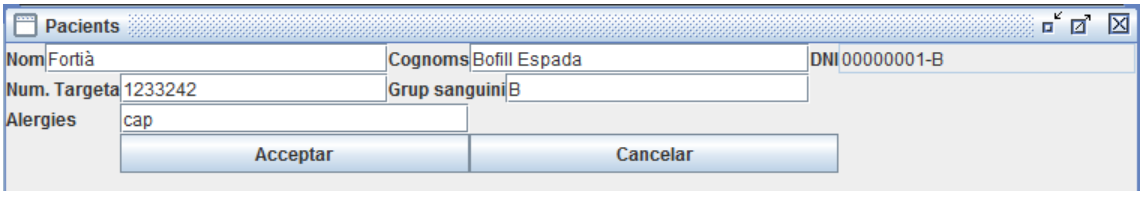

Figura 18 Editar pacient

De la mateixa manera al editar un pacient es mostren les seves dades i permet editar-les totes excepte el DNI, que com es veu és un camp amb l'edició deshabilitada.

Les opcions eliminar de les pantalles del metge i pacients simplement eliminen el metge o pacient escollit.

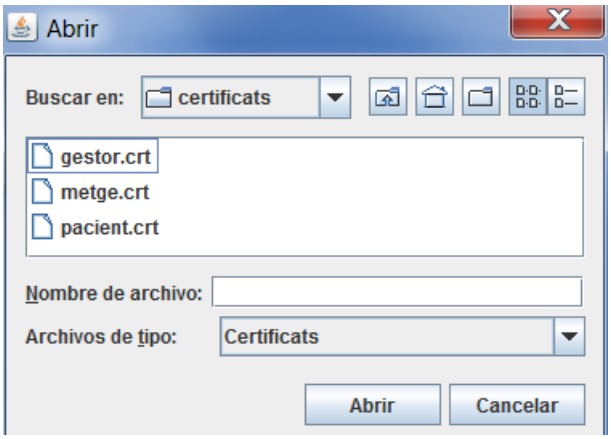

Figura 19 Assignar certificat al metge i al pacient

Les opcions d'assignar un certificat al metge i d'assignar un certificat al pacient ensenyen una finestra des d'on podem escollir el certificat corresponent pel metge o pacient.

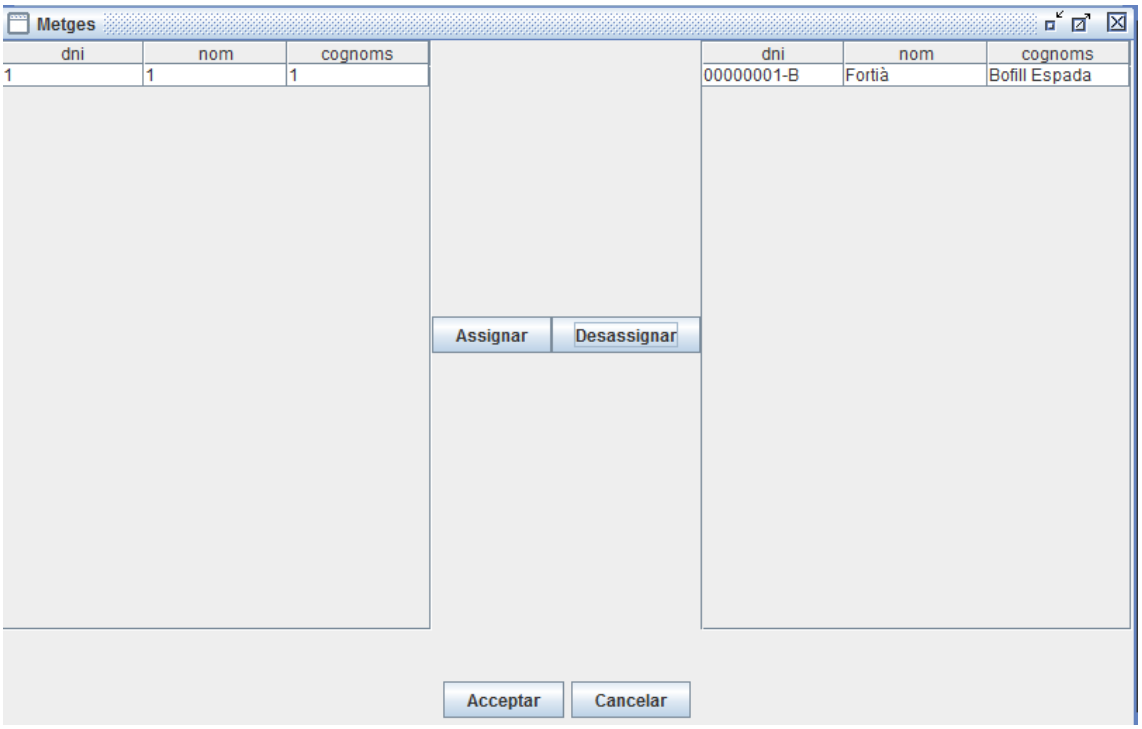

Figura 20 Assignar pacients al metge

Amb l'opció d'assignar un pacient al metge, es veu com es presenta una pantalla dividida en dos. A l'esquerra hi ha tots els pacients que no són del metge. A la dreta hi ha tots els pacients del metge. Amb el botó 'assignar' es pot assignar un pacient al metge i amb el botó desassignar es pot treure un pacient de la llista dels pacients del metge.

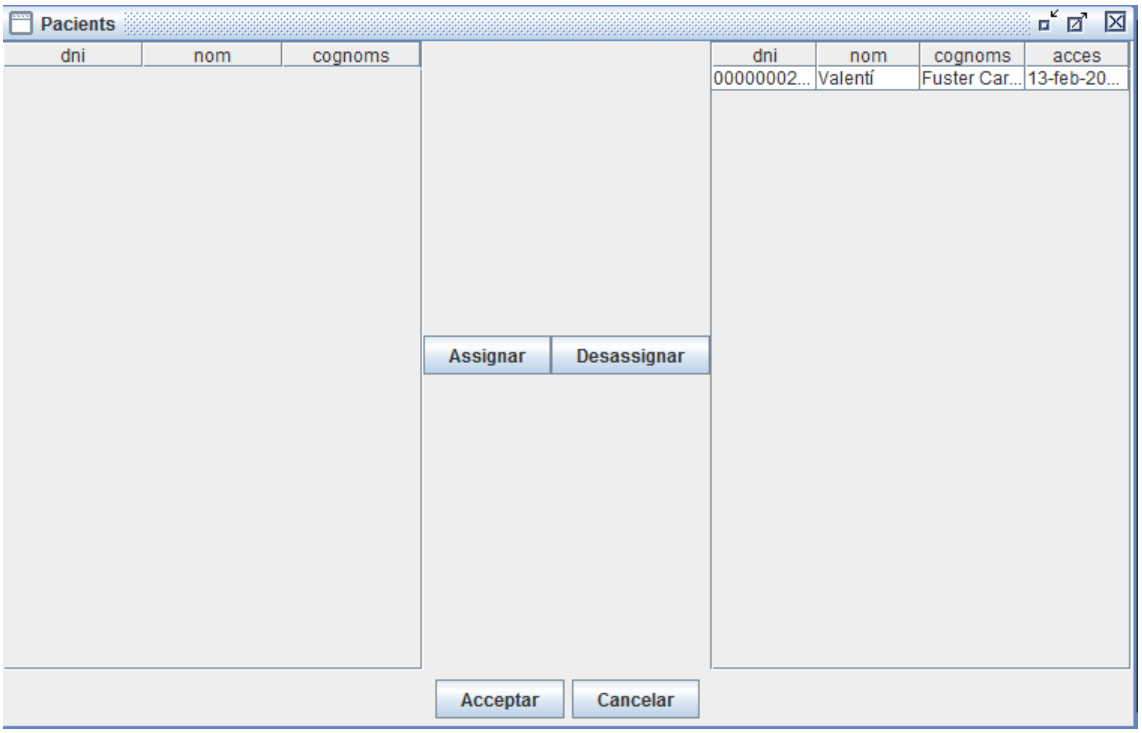

Figura 21 Assignar metges al pacient

De forma similar, amb l'opció d'assignar metges al pacient, es mostra una finestra amb el mateix funcionament, només que ara a l'esquerra tenim els metges que no són del pacient i a la dreta els metges que actualment són del pacient. De la mateixa manera amb els botons 'assignar' i 'dessasignar' es poden assignar els metges que es volen al pacient.

9 Conclusions

En aquest projecte s'ha dissenyat i desenvolupat un esquema criptogràfic per gestionar de forma segura els historials mèdics dels pacients a través d'una xarxa de comunicacions per tal de millorar l'atenció mèdica rebuda pels pacients i per tal d'ajudar a un metge a prendre una decisió correcta en la diagnosi i tractament dels seus pacients al permetre'l accedir a la història mèdica dels seus pacients remotament.

S'han obtingut dues aplicacions client-servidor que permeten satisfer les necessitats dels metges, dels pacients i del gestor del sistema complint amb els objectius marcats. Per tant la funcionalitat de metges i pacients s'ha integrat en una única aplicació comuna.

Concretament els metges poden accedir de forma segura al gestor del sistema per consultar les dades generals d'un pacient seu, per obtenir les dades de les visites dels seus pacients i per afegir-los les visites que se'ls realitza.

Als pacients se'ls permet accedir de forma segura al gestor del sistema per realitzar una consulta de les seves dades generals i per recuperar les dades de qualsevol de les seves visites.

L'aplicació del gestor del sistema el permet gestionar el repositori d'historials mèdics de forma central permetent-li registrar a nous usuaris (metges o pacients), autenticar als pacients i als metges que volen accedir al repositori acceptant-los les seves consultes, guardar de forma segura els historials mèdics dels pacients i verificant que les dades que s'insereixen o modifiquen en un historial mèdic es realitzen per un usuari autoritzat.

10 Treball futur

Una interessant millora seria implementar el client en un smart-phone per tal que pacients o metges no haguessin de dependre necessàriament de tenir a l'abast un ordinador en el moment de voler accedir al sistema. Per la limitació d'aquests dispositius la tecnologia idònia a utilitzar per la comunicació serien els sockets que és el que s'ha utilitzat en el projecte. Pel cas de smart-phones amb plataforma android únicament s'hauria d'implementar una interfície gràfica específica per aquest entorn podent fer servir el codi ja creat degut a que és compatible amb java.

La generació de certificats és una tasca que es podria integrar en l'aplicació del gestor i es podria modificar aquest també per validar els certificats dels usuaris que es volen autenticar.

Un altre punt a considerar i que no es realitza en aquest projecte seria la possibilitat de fer un control d'accés registrant els accessos que es fan indicant qui i què es consulta o modifica.

Bibliografia

1. Oracle. *http://download.oracle.com/javase/tutorial/networking/sockets/definition.html.*

2. Universitat Oberta de Catalunya. (2004). *Arquitectura de sistemes distribuïts.*

3. Universitat Oberta de Catalunya. (2006). *Criptografia.*