Implementació d'un esquema criptogràfic per a la gestió segura d'historials mèdics a través d'una xarxa de comunicacions

Armand Salas Montalà
Enginyeria en Informàtica

Jordi Castellà Roca

Gener 2008

Pròleg

Cada vegada més es fan servir les tecnologies de la informació per a obtenir noves possibilitats en qualsevol camp, ja sigui per a realitzar tasques que senzillament abans no existien, o per a millorar d'altres de comunes oferint, com a mínim, una alternativa al traslladar-les al món virtual. Convé tenir present que, com els demés, aquest medi posseeix unes propietats inherents diferents a les d'altres mitjans, la qual cosa fa plantejar un tractament específic per a ell.

Aquest projecte es proposa dissenyar i implementar un sistema de gestió d'historials mèdics per a ser usat remotament a través d'una xarxa de comunicacions, amb un èmfasi principal en l'assoliment d'un nivell de seguretat considerat alt.

Les dades d'un historial mèdic han de ser tractades amb una gran cura, és una informació altament confidencial i delicada, i per tant s'ha de vigilar el seu constant manteniment dins del camp privat en tot el possible, junt a la total negació de que aquelles dades siguin modificades per algú altre que no sigui un metge autoritzat.

Assegurar l'acompliment d'aquestes característiques no resulta un procés simple, i encara menys si es té en compte que el sistema restarà connectat a una xarxa pública com és Internet, la qual, ja de per sí, no ofereix cap tipus de seguretat. En conseqüència, per al funcionament del sistema es creen tot un conjunt de protocols criptogràfics amb xifratge asimètric de clau pública, mitjançant la utilització dels quals és factible garantir els objectius de seguretat marcats.

El producte final permet a un usuari del sistema poder entrar al mateix des de qualsevol terminal connectat a la xarxa, concretament deixant a un pacient poder accedir a la visualització del seu historial, i a un metge a la visualització i modificació de l'historial dels seus pacients. Qualsevol altra acció realitzada per una persona diferent no és autoritzada.

Índex

Índex de taules

Índex de figures

Descripció dels capítols

A continuació s'apunta breument el contingut de cada capítol i annex, assenyalant els conceptes, dissenys i implementacions que en ells s'expliquen.

- **Introducció**

  Capítol de presentació del projecte amb la idea general dels objectius i productes a obtenir.

- **Descripció del sistema**

  Estudi sobre l'abast i necessitats concretes que s'estableixen en el sistema, marcant l'escenari en el que es mourà, actors que intervindran i accions que podran fer.

- **Infraestructura de clau pública**

  Introducció a la criptografia, el concepte principal sobre el qual es mou el projecte, i explicació de la necessitat i creació dels certificats i les claus.

- **Esquema criptogràfic**

  Definició dels protocols criptogràfics que concreten la realització de les diferents accions a implementar.

- **Representació de les dades**

  Presentació de la necessitat d'estructurar la informació per a la transmissió i emmagatzematge de la informació, i definició dels diferents documents XML utilitzats en el sistema.

- **Comunicació entre els components**

  Explicació dels conceptes sobre la comunicació RMI i la seva exigència en el sistema per al contacte entre els clients i el servidor.

- **Emmagatzematge de la informació**

  Definició i estudi del model de base de dades fet servir en el sistema per a emmagatzemar la diversa informació que es mou en ell.

- **Interfície gràfica**

  Disseny i presentació de les aplicacions realitzades que compten amb una interfície d'usuari

- **Disseny del sistema**

  Estudi sobre el sistema des d'un caire més tècnic, explicació del desplegament, paquets i classes fetes servir per a la implementació de les aplicacions.

- **Jocs de proves**

  Presentació de l'ús del sistema amb una mostra de diverses accions de realització habitual, i proves per al control d'errors i excepcions.

- **Epíleg**

  Consideracions finals sobre la feina realitzada i l'aconseguiment de les fites proposades de partida.

Annexos

- **Preparació de l'entorn de treball**

  Conjunt d'eines i programes fets servir per al desenvolupament del projecte.

- **Desplegament i posada en marxa**

  Instruccions per a la instal·lació i execució del sistema que s'implementa.

- **Bibliografia i referència**

  Diversa documentació consultada durant la realització del projecte.

- **Glossari**

  Vocabulari relatiu als temes tractats en aquesta memòria.

Convencions

Aquesta memòria segueix unes poques convencions tipogràfiques i d'estil descrites a continuació:

- El text que fa referència a directoris, arxius, codi i altres elements similars és escrit usant una mida de text fixa –per exemple, `%JAVA_JDK%\jre\lib\security`–.

- Les línies de codi i comandes de consola que excedeixen l'amplada de la pàgina són continuades en la següent línia precedides pel caràcter ➥.

- Les comandes de consola i els resultats amb sortida per pantalla són mostrats dins d'un requadre de color gris, com per exemple:

```
openssl genrsa -des3 -out gestor.key –passout pass:medic 1024
```

- El format de documents i el contingut d'arxius són mostrats dins d'un requadre de color groc, com per exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio>
            <aleatoriGestor>
            </aleatoriGestor>
            <identificacioUsuari>
            </identificacioUsuari>
      </sessio>
      <document />
      <signatura />
</peticio>
```

Capítol 1
Introducció

1.1 Justificació i context

La societat, cada vegada més endinsada en les tecnologies de la informació, aprofita els avantatges que aquestes proporcionen adaptant-se per tal d'aconseguir millores de diversos caires. Un dels camps específics que resulta beneficiat per aquesta situació és el de l'assistència mèdica, en el que és conegut com a telemedicina o, fent ús d'un terme amb un significat més global, eSalut[1].

De fet, i centrant-nos en les possibilitats de les comunicacions a distància, encara que per ara amb una introducció escassa o fins i tot gairebé anecdòtica, és factible, per exemple, practicar certes consultes mantenint una correspondència visual entre el doctor i el pacient a través d'una webcam, o, anant força més enllà, efectuar tota una operació quirúrgica mitjançant l'ajuda d'un robot tenint a qui la realitza i a qui la rep en hospitals diferents[2].

Aquest projecte, però, pretén enfocar-se en el tractament que se'n pot fer de les dades mèdiques dels pacients a través de la xarxa. Un historial mèdic conté informació altament delicada i vital, tant des del punt de vista de que és necessari que contingui dades fiables i accessibles en qualsevol moment –sobretot en cas d'urgència–, com que aquestes obtencions de dades siguin fetes estrictament per aquells individus amb autoritat per fer-les.

També resulta interessant, tenint en compte la notable presència d'Internet en el país, poder oferir l'opció de que una persona sigui capaç de consultar el seu propi historial còmodament des de qualsevol lloc i a qualsevol hora, cosa que alhora permet un alleugeriment de feina en el centre sanitari.

---

[1] Per tal d'aprofundir en el tema es recomana visitar l'article sobre eSalut de la Viquipèdia a on s'hi pot trobar més informació i enllaços relacionats.
eSalud, Viquipèdia, http://es.wikipedia.org/wiki/ESalud

[2] A l'edició digital de la revista Wired s'hi pot llegir la noticia de la que es considera la primera operació efectuada d'aquesta manera.
"Surgeons Here, Patient There", Wired 2001, http://www.wired.com/medtech/health/news/2001/09/46946

Al fer ús de les xarxes de comunicacions s'aconsegueix acomplir amb aquests aspectes d'accessibilitat i immediatesa arreu, ara bé, de per sí el protocol IP sobre el que funciona Internet no ofereix cap mena de seguretat, i per tant, qualsevol individu podria suplantar la identitat d'un pacient consultant les seves dades o fer el paper de metge i modificar-les al seu criteri. Resulta obligatòria la imposició de mesures que permetin evitar aquests greus inconvenients.

1.2 Objectius

L'objectiu principal del projecte és la realització del disseny i la implementació d'un sistema de gestió remot d'historials mèdics en el que es presenti un alt nivell de seguretat. Per a aconseguir fer això s'utilitzen diversos esquemes criptogràfics de clau pública, buscant satisfer els quatre grans conceptes fonamentals de la seguretat de la informació esmentats seguidament:

- **Confidencialitat:** Les dades no han de ser accedides per altres fora del personal autoritzat. Un pacient només pot mirar el seu propi historial i un metge té restriccions a l'hora de veure els historials dels pacients –només pot veure de forma completa els historials dels pacients al seu càrrec mentre que per a la resta l'historial tan sols és accessible de manera parcial–.

- **No repudi:** L'autor d'una acció qualsevol no ha de poder negar-la una vegada realitzada. Quan un metge accedeix a un historial per a la seva modificació es guarda automàticament informació relativa a aquest canvi quedant vinculat a ell.

- **Autenticació:** La identitat que es declara ha de ser certa i evitar la suplantació. Un metge o un pacient concret només pot entrar al sistema com a l'individu específic que és, assegurant alhora que les dades que pugui generar siguin autèntiques.

- **Integritat:** Les dades han de mantenir-se constants sempre que no sigui per una modificació feta de forma autoritzada. Un historial només pot variar al ser modificat pel metge que tracta al pacient titular d'aquell informe mèdic.

A més, envoltant a aquesta base principal centrada en la criptografia i per a arribar a la consecució de la disponibilitat a distància requerida, es fa ús del protocol de comunicació RMI juntament amb el format XML per a la transferència de dades. El fet que el llenguatge de programació sigui Java també afavoreix en part a aquesta propietat degut a la seva característica d'independència de la plataforma d'execució.

També és necessari dissenyar i controlar l'ús d'un magatzem de dades encarregat de contenir tota aquella informació que conforma els historials i totes aquelles dades útils per al funcionament del sistema en sí.

Finalment, per tal de que qualsevol usuari pugui treballar còmodament amb el conjunt de programari, cal implementar diferents interfícies visuals procurant oferir una utilització senzilla i intuïtiva.

1.3 Enfocament i mètode seguit

Per a la consecució dels objectius plantejats es determina una divisió del projecte en diferents fases diferenciades.

- Infraestructura de clau pública (PKI).
- Protocols criptogràfics.
- Representacions de dades en XML.
- Comunicació per RMI.

- Base de dades.
- Interfícies gràfiques.

El projecte es desenvolupa seqüencialment, no iniciant cadascuna d'aquestes parts fins que no es dóna per finalitzada l'anterior. Per a l'aconseguiment de cada mòdul es crea el disseny, es realitza la implementació i, per tal de verificar el seu correcte funcionament, es prova primerament de forma unitària per a continuació tornar a provar-lo integrat dins del sistema amb la resta de fases.

Tot el treball es documenta en aquesta memòria, en la que es descriu detalladament i en diferents capítols cada part del sistema. La creació de cada mòdul s'alterna amb la documentació del mateix, deixant els darrers dies del temps previst per a la seva revisió i últims possibles retocs.

1.4 Planificació

El projecte, iniciat al semestre de tardor de l'any 2007, es planifica d'acord amb el mostrat seguidament.

Del 19 al 23 de setembre:
- Instal·lació, proves i documentació del programari necessari per a aquesta fase.
- Creació, proves i documentació de les diferents claus i certificats que conformen la PKI.

Del 24 de setembre al 21 d'octubre:
- Disseny, implementació, proves i documentació dels protocols criptogràfics.

Del 22 d'octubre al 4 de novembre:
- Disseny, implementació, proves i documentació del format de dades en XML.

Del 5 al 18 de novembre:
- Disseny, implementació, proves i documentació de la comunicació en RMI.

Del 19 de novembre al 2 de desembre:
- Disseny, implementació, proves i documentació de la base de dades.

Del 3 al 16 de desembre:
- Disseny, implementació, proves i documentació de la GUI dels clients.

Del 17 al 30 de desembre:
- Disseny, implementació, proves i documentació de la GUI del gestor.

Del 31 de desembre al 6 de gener:
- Revisió final del projecte i la documentació.

1.5 Productes obtinguts

El sistema descrit en aquest treball presenta els següents tres components clarament diferenciats.

- **Aplicació del personal mèdic** a través de la qual un metge, després d'haver-se identificat correctament, pot consultar un historial qualsevol o modificar les dades de l'informe mèdic d'un dels seus pacients.

- **Aplicació dels pacients** mitjançant la qual una persona que s'identifiqui de forma vàlida pot tenir accés a la informació que conforma el seu historial.

- **Sistema gestor central** encarregat del manteniment i control de la base de dades amb les dades dels historials, així com de la gestió i autenticació dels accessos produïts i l'execució de les comandes indicades.

Capítol 2
Descripció del sistema

2.1 Introducció

Amb aquest projecte no es pretén obtenir un sistema exhaustiu de gestió d'informes mèdics com seria el fet per a un hospital real, sinó que es busca aconseguir un exemple simple però a la vegada prou complet permetent comptar amb totes les funcionalitats bàsiques marcades com a objectius.

En els següents apartats s'imagina i es detalla una situació concreta a on aplicar el sistema de gestió d'historials mèdics proposat de partida, establint un escenari i un abast concret per al projecte.

2.2 Actors

Es distingeixen diferents grups d'usuaris segons el tipus d'accés i el nivell general de seguretat al que tenen accés. De forma general per a un sistema com aquest, i de major a menor grau de poder i drets, les següents són les tres grans classes a tractar:

- **Administradors:** Encarregats del correcte funcionament del sistema i d'aspectes com la creació dels certificats electrònics personals[1] i la gestió de les claus d'accés de la resta d'usuaris. La clau secreta d'un usuari hauria de ser coneguda només per ell mateix, i de fet el sistema es podria dissenyar de manera que es generés automàticament evitant que ningú més es veiés obligat a tenir-hi contacte. Els administradors, de totes maneres, de forma més o menys directa han de treballar amb el gestor central del sistema; són qui més accés hi tenen i per tant els agents amb més responsabilitat.

[1] Al capítol 3, Infraestructura de clau pública, es pot trobar informació més detallada sobre els certificats digitals i altres elements relacionats.

- **Personal de l'hospital:** Treballadors sanitaris o de gestió del centre. La seva rellevància en la seguretat del sistema varia en importància depenent del càrrec concret de l'empleat, i així un metge podria modificar l'historial d'un pacient, un infermer podria consultar aquest informe mèdic però no modificar-lo, i una secretaria no podria fer res amb l'historial però sí que podria inserir noves futures cites de visita, per posar alguns casos. Tots ells tenen accés a informació dels pacients, però aquest accés i les accions que es podrien permetre queden restringides segons l'autoritat de cada individu específic i el grau de privadesa de les dades amb les que intenta tractar.

- **Usuaris externs:** Pacients del centre. El seu poder en el sistema es podria catalogar de baix, és el tipus d'usuari més restringit. Un pacient podria executar accions com consultar les seves pròpies dades o demanar hora per a una consulta, per exemple.

Tenint present que es realitza un sistema simplificat però complet, seria convenient disposar com a mínim d'un actor corresponent a cadascun dels tres grups esmentats anteriorment. Incloent al gestor del sistema, se seleccionen els tipus d'actors assenyalats a continuació:

- **Gestor:** El gestor és un actor passiu, és el sistema encarregat d'atendre les peticions rebudes per part de la resta d'actors gestionant els accessos tenint en compte les diverses restriccions de seguretat existents.

- **Metge:** Actor que representa a qualsevol metge amb accés al sistema. Un metge pot consultar i modificar les dades mèdiques dels seus pacients, i consultar les dades mèdiques de menor grau de privadesa dels pacients que no tracta.

- **Pacient:** Actor que representa a qualsevol pacient amb accés al sistema. Un pacient pot consultar les seves dades mèdiques.

- **Administrador:** Actor encarregat de donar d'alta als metges i als pacients, i de crear els seus certificats.

Per tal de simplificar lleugerament el treball es pren la determinació d'agafar aquests rols com a excloents, és a dir, un usuari disposarà d'un únic certificat que el situarà dins d'un grup i no comptarà amb cap més –un metge no serà mai un pacient, per exemple–.

Tampoc es tenen en compte les diferents especialitats existents en medicina, amb el que pot veure's la figura del metge com un paper amb capacitat per a generar qualsevol tipus d'informació mèdica. Alhora, això fa que un pacient no tingui necessitat de disposar de més d'un doctor ja que un de sol el pot atendre per a qualsevol malaltia.

Es considera que aquestes llicències tenen cabuda ja que el més interessant per ara és distingir i valorar les diferents possibilitats de cadascun dels actors, i no tant disposar d'un sistema completament preparat per a ser explotat en un entorn real.

2.3 Accions

Seguint amb l'explicació que s'apuntava al descriure els diferents actors implicats en el sistema, a continuació es detallen les accions que poden realitzar:

- **Autenticar-se:** Acció disponible per a pacients i metges. A un usuari que intenta utilitzar el sistema se li requereix passar una fase d'autenticació per tal de poder validar-lo i establir a quines altres accions hi té accés. Una vegada superat aquest procés, una sessió és creada automàticament per part del gestor aconseguint que per a altres futures accions demanades pels usuaris no calgui tornar a executar el pas de l'autenticació.

- **Tancar sessió:** Acció disponible per a pacients i metges. Aquest procés només té sentit si s'executa un cop creada una sessió durant la fase d'autenticació, i és que el que realitza és la seva eliminació per tal de finalitzar amb l'ús del sistema d'una forma segura. Per a poder fer aquesta acció, primerament el gestor verifica la validesa de la sessió comprovant que l'usuari que la demana és autèntic.

- **Consultar historial:** Acció disponible per a pacients i metges. Permet obtenir de la base de dades la informació personal i mèdica d'un pacient determinat. Aquest procés, però, varia en quant a les dades mostrades segons si l'usuari és pacient o metge, i alhora torna a diferenciar-se si, sent metge, el pacient l'historial del qual es vol consultar resta o no al seu càrrec. Per a realitzar aquesta acció, el gestor verifica la sessió creada durant l'autenticació, i només s'acaba executant si aquest darrer pas no dóna cap error.

- **Inserir dades a historial:** Acció disponible per als metges. Un metge pot introduir nova informació a l'historial d'un pacient, sempre i quan aquest pacient sigui seu. El gestor, abans d'acabar executant aquesta acció, verifica la sessió creada en el procés d'autenticació.

- **Consultar pacients assignats:** Acció disponible per als metges. Un metge pot demanar visualitzar un llistat dels pacients que tracta, i això és el s'aconsegueix amb aquesta acció. De nou, en un primer pas el gestor ha de validar la sessió existent per tal d'executar el procés.

- **Iniciar els servidors:** Acció disponible per als administradors. Per tal de poder disposar de les funcionalitats del sistema cal que inicialment sigui posat en marxa, procés que du a terme aquesta acció.

- **Crear usuari:** Acció disponible per als administradors. Aquesta acció engloba el procés de creació dels certificats que els usuaris necessiten per a poder utilitzar el sistema, i la introducció en el mateix de tota la informació convenient per a que siguin reconeguts.

- **Assignar pacient a metge:** Acció disponible per als administradors. Un pacient ha de restar vinculat a un metge per tal de poder ser atès a les visites que faci. També resulta possible que el metge que pertoca a un pacient pugui variar al llarg del temps, no sent sempre el mateix. Aquests processos són els que es duen a terme amb aquesta acció.

Totes aquestes accions poden visualitzar-se en el següent gràfic **(figura 1)** a on es mostra el diagrama de casos d'ús del sistema.

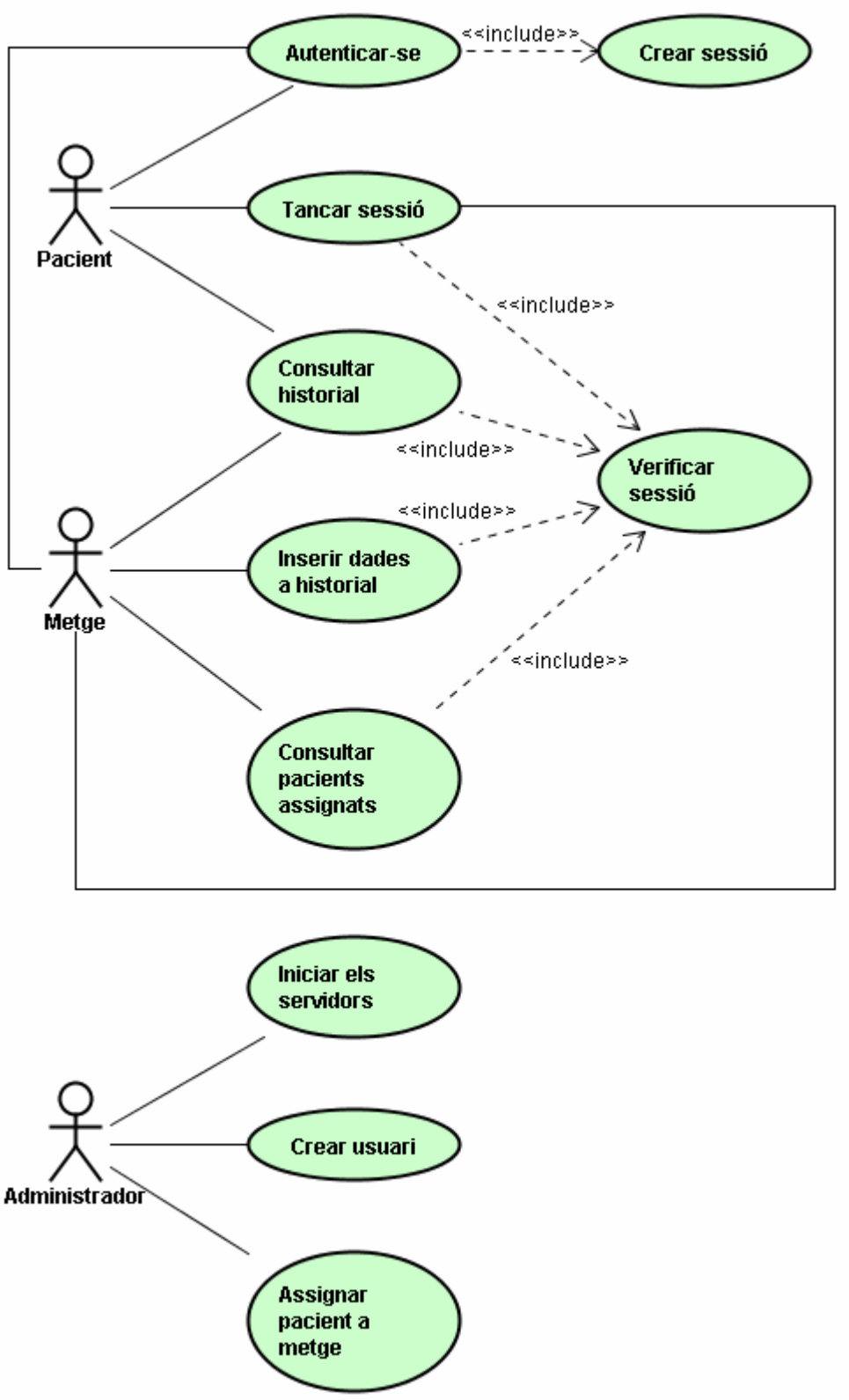

Figura 1. Model general de casos d'ús

D'igual manera que es podrien afegir altres rols als triats, també podrien establir-se altres accions o variar les existents. De fet, hi ha dades utilitzades de partida que potser no correspondria que fossin introduïdes per un administrador, com podria ser informació mèdica, sigui o no de caràcter més o menys públic –el grup sanguini, per exemple–, o tal vegada la mateixa informació personal dels usuaris –com adreça, telèfon, etc–.

D'aquestes tasques podria encarregar-se una secretaria o algun altre treballador d'administració, però com que s'estableix que no es disposa d'aquest rol i no és un aspecte a destacar per als objectius del projecte, senzillament es pot suposar que l'administrador assumeix també aquest paper i és ell qui ho fa.

2.4 Gestió de la informació

El sistema es prepara per a funcionar amb informació que conformi historials mèdics de pacients, i un primer pas es establir quines dades concretes són aquestes. Seguint la tònica exposada fins ara, per a simplificar no es considera el treball amb un gran volum d'informació diversa, sinó la suficient per a mostrar les possibilitats del sistema.

S'ha comentat també la intenció d'establir diferents graus d'accés segons l'usuari i el tipus d'informació que aquest prova d'aconseguir, i en conseqüència no totes les dades queden classificades amb el mateix nivell de privadesa. Considerant la informació mèdica, és important que qualsevol amb un mínim d'autoritat pugui accedir a dades com les al·lèrgies d'un pacient o el seu grup sanguini; donada una situació d'urgència resulta més valuós salvar la vida del pacient que no conservar el secret d'aquestes dades. Amb d'altres informacions, però, és difícil trobar un moment crític a on veure-les com a vitals, i per tant no resulta necessari deixar públiques dades com les del seguiment d'una miopia, per exemple.

S'organitza la informació en dos grups generals segons el seu grau de confidencialitat, fent bàsicament una divisió tenint per una part les dades que es generin en una visita mèdica, i una segona part per a tota la resta d'informació. Aquesta classificació és la que pot observar-se en el següent requadre.

Taula 1. Classificació de les dades segons la seva confidencialitat

Confidencialitat baixa	Confidencialitat alta
Dades personals: • DNI • Nom • Cognoms Dades mèdiques generals: • Grup sanguini	Dades de les visites mèdiques: • Observacions • Recepta

Capítol 3
Infraestructura de clau pública

3.1 Nocions generals sobre criptografia de clau pública

Aquest treball, buscant assolir l'alt nivell de seguretat proposat, se sustenta sobre un protocol criptogràfic SSL que permeti la transferència d'informació de manera segura. Abans de continuar, però, convindria introduir, i això és el que es pretén amb el present apartat, unes poques idees molt bàsiques[1] sobre alguns dels principals conceptes criptogràfics d'aparició en el projecte.

La criptografia de clau pública, també anomenada criptografia asimètrica, se centra en el fet de que cada entitat[2] disposi d'un parell de claus, una de privada i una altra de pública. Aquestes claus responen a una funció matemàtica determinada, quedant vinculades entre sí però mantenint alhora propietats diferents al resultar la clau privada de molt difícil obtenció a partir de la pública.

Es poden distingir dues vessants principals dins d'aquest tipus de criptografia: el xifratge de clau pública i la signatura digital.

- Un missatge pot ser **xifrat** per l'emissor fent servir la clau pública del destinatari, qui serà l'únic en poder-lo llegir ja que aquest xifratge només es podrà desfer utilitzant la clau privada vinculada que tan sols ell posseeix. Amb aquesta acció s'acompleix la propietat de confidencialitat.

- Un missatge pot ser **signat** per l'emissor utilitzant la seva clau privada, de manera que el destinatari pot comprovar la veracitat del que llegeix fent servir la clau pública

[1] Per tal d'endinsar-se més profundament en el tema es recomana consultar les referències bibliogràfiques citades a l'annex C.

[2] En criptografia se sol diferenciar entre entitat i subscriptor, referint-se el primer terme a un organisme o companyia, i el segon a una persona o individu concret. En aquí s'usa entitat incloent a tots ells.

relacionada. Amb aquest fet se satisfan les característiques de no repudi, integritat i autenticitat.

Fins aquí es té un sistema aparentment segur, es garanteixen les propietats de seguretat bàsiques, ara bé, hi ha un inconvenient que podria presentar-se fàcilment en un entorn no controlat i desbaratar tota aquesta fortalesa: la clau pública d'una entitat és accessible a tothom, però realment no hi ha res que sens dubte asseguri que aquesta clau és autèntica i correspon a aquella entitat[1]. La manera de solucionar aquest problema és usant una infraestructura de clau pública, també coneguda com a PKI, la qual permet vincular de forma inequívoca una clau pública amb el seu propietari mitjançant un sistema de confiança.

Dins d'una PKI s'hi engloben diversos elements entre els que destaca per sobre de tots el certificat digital, un objecte que assegura, o certifica, el lligam entre una clau pública i una entitat. La infraestructura se sustenta sobre aquests certificats, i la resta de figures que conformen la PKI hi són per tal d'oferir serveis per a la seva creació i gestió.

Un altre element rellevant d'una PKI és el que s'anomena autoritat de certificació o CA, encarregada de crear els certificats i d'atorgar-los o rebatre'ls l'autenticitat. El seu poder recau simplement en ser considerada pels usuaris de la infraestructura com a una TTP o entitat en la que s'hi pot confiar, de manera que un certificat que compta amb la signatura d'una autoritat de certificació reconeguda, és un certificat de confiança perquè aquesta l'ha validat[2].

Encara que resulta una figura més secundaria i opcional, també es pot fer esment de l'autoritat de registre o RA, la qual es dedica a validar la relació entre la clau pública i la identitat del seu propietari abans de que la CA emeti el certificat corresponent.

A l'hora d'implementar una PKI es pot triar entre diferents solucions, tant de productes estàndards com d'altres de propietaris. Entre els darrers, però adoptats com a estàndards de facto, es poden trobar algunes especificacions com els documents PKCS creats pels RSA Laboratories amb la col·laboració d'altres empreses de la indústria.

Els PKCS són una sèrie de formats usats en criptografia per tal de definir la sintaxi d'una estructura de dades concreta, i així, per exemple, en el PKCS#12 s'hi pot trobar emmagatzemada diversa informació com la parella de claus i el certificat d'una entitat, i el certificat de la CA que ha validat l'anterior.

Un altre format acceptat i utilitzat de forma àmplia és l'estàndard x.509, el qual s'usa per a especificar l'estructura d'un certificat digital indicant quina informació conté i com hi és guardada.

3.2 Ús d'una PKI en el projecte

Tal i com es comenta en el segon apartat del capítol 2, en el disseny que es planteja del projecte es distingeixen fins a tres classes d'usuari: els metges, els pacients i els gestors del sistema. D'aquest últim tipus només es preveu que hi hagi un únic usuari, però els altres dos podrien ser multitud; sigui com sigui, per al funcionament del sistema es requereix que tots i cadascun d'ells disposin de les seves claus i els seus certificats propis.

Per a tal motiu, per poder comptar amb aquests documents, s'utilitza una PKI amb la presència d'una CA que accepti les peticions dels certificats i els signi. De totes formes, i obviant altres de les possibilitats que posseeix una infraestructura, el seu ús en el projecte es limita a aquesta única acció.

---

[1] Un emissor que busca xifrar un missatge per a un destinatari podria perfectament ser enganyat i acabar obtenint de forma inconscient la clau pública d'un tercer, amb la qual cosa, aquest tercer que no havia de llegir res, es troba amb un missatge personalment xifrat per a ell. A aquesta situació se la coneix com a *atac de l'home a mig camí*.

[2] En realitat s'hauria de donar un pas més per a acabar de confiar en el certificat, i és mirar que aquest no consti en el CRL o llista de certificats revocats. Aquesta és una mena de llista negra mantinguda per la pròpia CA a on s'apunten aquells certificats prèviament aprovats als que, per un o altre motiu, convé que se'ls hi anul·li l'autenticitat.

Per a la creació de la PKI es fa servir el programari OpenSSL. La seva utilització pot configurar-se mitjançant l'ús d'un arxiu de text pla típicament anomenat `openssl.cnf`. Aquest fitxer ha estat editat per a simplificar el tractament i les operacions descrites en aquesta secció, de forma que, per exemple, s'ha indicat la cadena "medic" –sense les cometes– com a valor per a totes les contrasenyes requerides en els processos explicats. Evidentment aquest arxiu de configuració podria ser modificat per tal d'adaptar-lo a les necessitats requerides en una altra situació[1].

Convé remarcar també que els certificats originats en les següents seccions s'han creat amb la intenció de servir purament com a exemples de com haurien de ser, i al mateix temps poder comptar amb aquests elements per tal de demostrar el correcte funcionament del sistema durant el seu desenvolupament.

3.3 Creació de les claus i el certificat de la CA

El primer pas a seguir és obtenir la clau privada de la CA, cosa per a la qual es pot executar la instrucció mostrada en el següent requadre.

```
openssl genrsa -des3 -out CA.key –passout pass:medic 2048
```

El resultat obtingut amb aquesta acció és fitxer següent:

- `CA.key` contenidor de la clau privada RSA en format PEM, xifrada amb Triple DES i amb una longitud de 2048 bits.

Es pot destacar la longitud de la clau la qual ha de ser suficientment forta –o llarga– per tal de no veure's compromesa. Si la clau és dèbil i s'aconsegueix trencar, tota la PKI pot començar a derruir-se perquè la confiança de la CA es troba malmesa.

A continuació, i gràcies a la comanda mostrada seguidament, es pot crear el certificat autosignat de la CA.

```
openssl req -new -x509 -sha1 -key CA.key -out CA.crt -config
➥openssl.cnf -days 365
```

Al fer ús d'aquesta instrucció, OpenSSL demana diversa informació identificadora de l'entitat per tal d'incorporar-la al certificat. A l'arxiu `openssl.cnf` ja s'han inclòs diferents valors proposats per defecte en aquest moment –apareguts entre claus, com `[Gestors]`–, els quals són agafats si no s'indica cap de diferent. Les entrades mostrades a continuació són aquelles en les que s'ha introduït un altre valor a l'ofert per OpenSSL.

```
Organizational Unit Name (eg, section) [Gestors]:CAs
Common Name (eg, YOUR name) []:CA
```

Amb aquesta acció s'obté finalment el següent arxiu:

- `CA.crt` contenidor del certificat x.509 en format PEM amb la clau pública i totes les dades identificadores de la CA, amb funció hash sha1 i amb una vigència de 365 dies des de la seva emissió.

[1] A l'annex A es pot obtenir el contingut del fitxer `openssl.cnf` utilitzat, i a l'annex X es pot consultar la bibliografia a on s'hi poden trobar vàries fonts per si es desitja aprofundir en l'ús d'OpenSSL i l'edició de l'arxiu de configuració per tal de modificar-lo.

A partir d'aquest moment, i gràcies a aquests fitxers esmentats, ja es pot comptar amb una CA amb capacitat per a validar altres certificats, amb la qual cosa ja es podria passar a prestar atenció al conjunt d'usuaris del que disposarà el sistema.

3.4 Creació de les claus i els certificats dels usuaris

Exactament com amb la CA, el primer pas a seguir és la generació de la clau privada, la qual pot crear-se de forma similar amb la instrucció següent preparada en aquesta ocasió per a l'usuari gestor.

```
openssl genrsa -des3 -out gestor.key –passout pass:medic 1024
```

Finalitzada l'execució de la comanda s'acaba obtenint l'arxiu següent:

- `gestor.key` contenidor de la clau privada RSA en format PEM, xifrada amb Triple DES i amb una longitud de 1024 bits.

De nou, en aquí es pot fer esment de la longitud de la clau, la qual, com a clau d'usuari i sense arribar a la consideració de ser una clau dèbil, no cal que sigui especialment tan forta com la de la CA.

Per a que una entitat qualsevol pugui ser propietària d'un certificat, cal preparar una petició del mateix. Seguidament es mostra la comanda de creació d'aquest objecte.

```
openssl req -new -sha1 -key gestor.key -out gestor.csr -config
➥openssl.cnf
```

Amb aquesta instrucció, d'igual manera que en la comanda per a la creació del certificat de la CA, OpenSSL requereix la introducció de diferents dades d'identificació. Actuant de la mateixa forma, a continuació es pot veure el camp a on s'ha entrat un valor diferent al proposat.

```
Common Name (eg, YOUR name) []:Gestor
```

Finalitzada l'execució de la comanda s'acaben obtenint els arxius següents:

- `gestor.csr` contenidor de la petició de certificat en format PEM amb la clau pública i totes les dades identificadores del gestor, i amb funció hash sha1.

El procés a realitzar seguidament és la creació del certificat de l'usuari amb la signatura de la CA, cosa que es pot assolir amb l'execució de la comanda situada en el requadre mostrat a continuació.

```
openssl x509 -req -in gestor.csr -CA CA.crt -CAkey CA.key -out
➥gestor.crt -CAcreateserial -days 365 -passin pass:medic
```

La instrucció s'executa sense requerir cap entrada, originant finalment el següent fitxer:

- `gestor.crt` contenidor del certificat x.509 en format PEM amb tota la informació del gestor existent en la petició del certificat, i amb una vigència de 365 dies des de la seva emissió.

L'últim punt correspon a l'empaquetament de tot el necessari dins d'un document PKCS#12. A continuació és mostrada la comanda que permet la seva creació.

```
openssl pkcs12 -export -in gestor.crt -inkey gestor.key -certfile
➥CA.crt -out gestor.p12 -passin pass:medic -password pass:medic
```

Novament la instrucció conclou sense necessitar cap entrada suplementària, lliurant com a resultat el següent arxiu:

- `gestor.p12` contenidor del document PKCS#12 amb les claus i el certificat del gestor, junt amb el certificat de la CA.

Amb aquest darrer pas es dóna per finalitzada la creació dels arxius necessaris per a l'usuari gestor. El que s'hauria de fer tot seguit és repetir el mateix procés per a la resta d'usuaris que utilitzaran el sistema, creant els seus certificats signats per la CA.

S'indiquen a continuació els valors introduïts per a un usuari metge en la comanda de creació de la petició del certificat.

```
Organizational Unit Name (eg, section) [Gestors]:Metges
Common Name (eg, YOUR name) []:MetgeA
D.N.I or N.S.S. []:10000000-A
```

Els següents camps són els modificats per a aconseguir la petició del certificat d'un usuari pacient.

```
Organizational Unit Name (eg, section) [Gestors]:Pacients
Common Name (eg, YOUR name) []:PacientA
D.N.I or N.S.S. []:00000001-A
```

Capítol 4
Esquema criptogràfic

4.1 Introducció

El projecte s'ha pensat per a oferir un conjunt d'accions i funcionalitats concretes dins del marc de comptar amb un servei per a la gestió d'historials mèdics. Tal i com s'ha presentat a l'apartat 2.3, als usuaris se'ls hi permet executar la consulta d'un historial, la consulta dels pacients assignats a un metge i la inserció de dades a l'historial mèdic. A més, per al funcionament segur del sistema, es disposen de l'acció d'autenticació i creació de la sessió, així com de la del seu tancament.

Convé plantejar i precisar protocols que defineixin totes aquestes diverses accions que es puguin produir, prestant especial atenció als aspectes criptogràfics que han de contenir per tal de satisfer els requisits de seguretat establerts al segon apartat del primer capítol.

4.2 Notació

La següent és la notació utilitzada per a la descripció dels diferents protocols criptogràfics mostrats amb posterioritat.

Taula 2. Notació dels protocols criptogràfics

E	Entitat usuari o gestor.
U	Usuari del sistema (metge o pacient).
G	Gestor del sistema.
M	Missatge.

P_E	Clau pública d'una entitat E.
S_E	Clau privada d'una entitat E.
Id_U	Identificador d'un usuari U.
N_E	Numero aleatori generat per una entitat E.
T	Instant de temps actual.
H	Historial d'un pacient P.
V	Visita d'un historial H.
Vm	Part d'una visita creada per un metge M.
X	Número de sèrie d'una visita V.
Xf	Número de sèrie de l'última visita V d'un historial H.
$P_E[M]$	Xifratge del missatge M usant la clau pública P_E de l'entitat E.
$S_E[P_E[M]]$	Desxifratge de la xifra $P_E[M]$ usant la clau privada S_E de l'entitat E.
$S_E[M]$	Signatura digital del missatge M usant la clau privada S_E de l'entitat E.
$P_E[S_E[M]]$	Verificació de la signatura digital $S_E[M]$ usant la clau pública P_E de l'entitat E.

4.3 Procediments

A continuació es concreten un conjunt de funcions ja sigui amb la idea de contenir accions d'utilització freqüent evitant d'aquesta manera la seva continua repetició, o per la simple raó de separar en parts per tal d'estructurar i simplificar la definició dels diferents esquemes criptogràfics. Per a la programació dels diferents protocols, tots aquests procediments s'implementen seguint les pautes que mostren les següents definicions.

S'assenyala també que, en cas de produir-se o controlar-se algun tipus d'error, en aquest tema es remarquen només aquells relatius als aspectes de la seguretat, el que no significa que no es generi o es tingui cura d'altres classes d'error.

peticioSimple	
Petició genèrica per part de l'usuari amb la informació mínima del testimoni de sessió requerida per a una comunicació.	
Cridat per	**Valors d'entrada**
U	N'_G, Id_U, P_G
1. Xifrar N'_G i Id_U amb la clau pública de G obtenint $P_G[N'_G, Id_U]$. 2. Retornar $P_G[N'_G, Id_U]$.	

verificarUsuariSessio
Comprovació de l'usuari i la seva sessió per part del gestor.

Retorna error en cas de que:

- La identificació de l'usuari xifrada junt amb el testimoni no coincideix amb la identificació de l'usuari amb la que aquest es troba relacionat a la base de dades, és a dir, la sessió no correspon no correspon a l'usuari.
- El valor aleatori creat originalment pel gestor i el retornat posteriorment per l'usuari no siguin els mateixos, amb la qual cosa ha fallat l'autenticació de l'usuari.

Cridat per	**Valors d'entrada**
G	Id'_U, N'_G, S_G

1. Obtenir la sessió $P_G[N_G, Id_U]$ mitjançant una consulta a la base de dades fent servir Id'_U.
2. Desxifrar $P_G[N_G, Id_U]$ amb la clau privada de G obtenint N_G i Id_U; $S_G[P_G[N_G, Id_U]]$.
3. Si $Id_U = Id'_U$ llavors
    a. Si $N_G <> N'_G$ llavors –l'usuari no supera l'autenticació–
        i. Retornar error.
4. Sinó –la sessió no pertany a l'usuari–
    b. Esborrar de la base de dades el testimoni de sessió lligat a Id_U corresponent a l'usuari U.
    c. Retornar error.

obtenirVisites

Verificació i obtenció de les dades privades de l'historial demanat per un usuari.

Cridat per	**Valors d'entrada**
U	Xf, { V_1, ..., V_n }, P_G, S_U

1. Per a cada V fer
    a. Verificar la signatura digital de M.
    b. Verificar la signatura digital de G.
    c. Verificar la seqüència.
    d. Preparar per a l'historial H les dades de la visita V.
2. Retornar H amb totes les dades tractades.

obtenirPacients

Obtenció del llistat de pacients assignats a un metge.

Cridat per	**Valors d'entrada**
G	Id_M, P_M, S_G

1. Obtenir el llistat de pacients { P_1, ..., P_n } assignats al metge M mitjançant una consulta a la base de dades fent servir Id_M.
2. Executar el procediment **prepararPacients** obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
3. Retornar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.

prepararPacients	
Preparació d'un llistat de pacients per a poder ser enviats a un metge.	
Cridat per	**Valors d'entrada**
G	$\{ P_1, ..., P_n \}$, P_M, S_G

1. Signar $\{ P_1, ..., P_n \}$ amb la clau privada de G obtenint $S_G[\{ P_1, ..., P_n \}]$.
2. Aconseguir el certificat del metge M mitjançant una consulta a la base de dades fent servir Id_M.
3. Aconseguir la clau pública del metge M a partir del seu certificat.
4. Xifrar $\{ P_1, ..., P_n \}$ i $S_G[\{ P_1, ..., P_n \}]$ amb la clau pública de M obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
5. Retornar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.

llegeixLlistaPacients	
Desxifratge i verificació del llistat de pacients demanat per un metge.	
Cridat per	**Valors d'entrada**
M	$P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$, S_M, P_G

1. Desxifrar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$ amb la clau privada del metge M obtenint $\{ P_1, ..., P_n \}$ i $S_G[\{ P_1, ..., P_n \}]$; $S_M[P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]]$.
2. Verificar la signatura digital $S_G[\{ P_1, ..., P_n \}]$ amb la clau pública de G; $P_G[S_G[\{ P_1, ..., P_n \}]]$.
3. Si la verificació no retorna error llavors
    a. Retornar $\{ P_1, ..., P_n \}$.
4. Sinó
    a. Retornar error.

guardarVisita	
Preparació i emmagatzematge de la visita d'un historial.	
Cridat per	**Valors d'entrada**
G	Id_U, Vm, $S_M[Vm]$, S_G, P_G

1. Obtenir l'instant de temps actual T.
2. Aconseguir la identificació Id'_U del pacient P a partir de Vm.
3. Obtenir el número de sèrie Xf i la signatura $S_G[Id_U, Xf]$ de l'historial H mitjançant una consulta a la base de dades fent servir Id'_U.
4. Verificar la signatura digital $S_G[Id_U, Xf]$ amb la clau pública de G; $P_G[S_G[Id_U, Xf]]$.
5. Si la verificació no retorna error llavors
    a. Incrementar en una unitat Xf obtenint Xf'; Xf' = Xf + 1.
    b. Signar Id'_U i Xf' amb la clau privada de G obtenint $S_G[Id'_U, Xf']$.
    c. Signar Xf', Id'_U, T, Vm i $S_M[Vm]$ amb la clau privada de G obtenint $S_G[Xf', Id'_U, T, Vm, S_M[Vm]]$.
    d. Xifrar Vm i $S_M[Vm]$ amb la clau pública de G obtenint $P_G[Vm, S_M[Vm]]$.
    e. Guardar a la base de dades Xf' i $S_G[Id'_U, Xf']$ per a l'historial i Xf', T, $P_G[Vm, S_M[Vm]]$ i $S_G[Xf', Id'_U, T, Vm, S_M[Vm]]$ per a les visites mitjançant una inserció a la base de dades fent servir Id'_U.

6. Sinó a. Retornar error.

obtenirTotsPacients	
Obtenció del llistat de tots els pacients del sistema.	
Cridat per	**Valors d'entrada**
G	P_M, S_G
1. Obtenir el llistat de pacients { P_1, ..., P_n } del sistema. 2. Executar el procediment **prepararPacients** obtenint P_M[{ P_1, ..., P_n }, S_G[{ P_1, ..., P_n }]]. 3. Retornar P_M[{ P_1, ..., P_n }, S_G[{ P_1, ..., P_n }]].	

4.4 Protocols criptogràfics

En els següents apartats es detallen els diferents protocols per a les accions que s'implementen en el sistema.

4.4.1 Autenticació i creació de sessió

El protocol d'autenticació intenta garantir que les dues parts que es comuniquen, el gestor i l'usuari metge o pacient, siguin vertaderes i fiables. Pretenent obtenir un sistema segur cercant alhora un funcionament remot sobre una xarxa com és Internet, caldria que aquesta autenticació es realitzés per a cadascuna de les comunicacions a fer.

De totes maneres, aquesta forma de fer, tot i que eficaç, no resulta gaire eficient, i a pesar de la limitació del protocol de xarxa es pot programar a sobre una funcionalitat per al manteniment de sessió, cosa que permet el no haver de realitzar contínuament tot l'esquema d'autenticació. El que s'ha decidit fer és, prenent com a base el protocol d'autenticació de Needham-Schroeder[1], crear sessions que estalviïn part d'aquest procés.

Es pot dividir aquest protocol en dues seccions tenint en compte l'ús que es realitza d'ell, i és que no és un protocol que es faci servir de forma completament independent sinó que en part resta inclòs dins d'altres esquemes. En el següent requadre es mostra una primera secció en la que es du a terme la verificació de l'autenticitat del gestor, l'establiment d'una sessió per a l'usuari, i la creació i el lliurament d'un testimoni (N_G) fet servir per a la posterior autenticació de l'usuari. Aquesta part s'executa a l'inici de l'aplicació quan un usuari la vol fer servir.

Autenticació del gestor i creació de la sessió i el seu testimoni
1. U realitza les operacions següents
1. Obtenir un valor aleatori N_U i conservar-lo temporalment a memòria per a usar-lo en el pas 3.

[1] Es pot consultar el funcionament del protocol de Needham-Schroeder a la corresponent entrada de la Viquipèdia Protocolo de Needham-Schroeder, Viquipèdia, http://es.wikipedia.org/wiki/Protocolo_de_Needham-Schroeder

2. Executar el procediment **procediment1_1** obtenint $P_G[N_U, Id_U]$.

procediment1_1

1. Xifrar N_U i Id_U amb la clau pública de G obtenint $P_G[N_U, Id_U]$.
2. Retornar $P_G[N_U, Id_U]$.

3. Enviar $P_G[N_U, Id_U]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment1_2** obtenint $P_U[N'_U, N_G]$.

procediment1_2

1. Desxifrar $P_G[N_U, Id_U]$ amb la clau privada de G obtenint N'_U i Id_U; $S_G[P_G[N_U, Id_U]]$.
2. Obtenir el certificat de l'usuari U mitjançant una consulta a la base de dades fent servir Id_U.
3. Obtenir un valor aleatori N_G.
4. Xifrar N_G i Id_U amb la clau pública de G obtenint $P_G[N_G, Id_U]$.
5. Guardar a la base de dades la xifra $P_G[N_G, Id_U]$ relacionant-la amb Id_U, substituint una entrada anterior en cas d'existir.
6. Aconseguir la clau pública de l'usuari U a partir del seu certificat.
7. Xifrar N'_U i N_G amb la clau pública de U obtenint $P_U[N'_U, N_G]$.
8. Retornar $P_U[N'_U, N_G]$.

2. Enviar $P_U[N'_U, N_G]$ a U.

3. U realitza les operacions següents

1. Executar el procediment **procediment1_3** obtenint N'_G.

procediment1_3

1. Desxifrar $P_U[N'_U, N_G]$ amb la clau privada de U obtenint N'_U i N'_G; $S_U[P_U[N'_U, N_G]]$.
2. Si $N_U = N'_U$ llavors
   a. Retornar N'_G.
3. Sinó
   a. Retornar error.

2. Executar el procediment **peticioSimple** obtenint $P_G[Id_U, N'_G]$.
3. Enviar $P_G[Id_U, N'_G]$ a G.

4. G realitza les operacions següents

1. Executar el procediment **procediment1_4**.

procediment1_4

1. Desxifrar $P_G[N'_G, Id_U]$ amb la clau privada de G obtenint N'_G i Id_U; $S_G[P_G[N'_G, Id_U]]$.
2. Executar el procediment **verificarUsuariSessio**.

El testimoni que l'usuari rep en aquesta primera part ha de ser utilitzat junt amb la seva identificació cada vegada que es requereix alguna de les funcionalitats descrites en altres protocols d'aquest capítol. Quan això succeeix, el gestor, a través d'aquest testimoni, comprova l'autenticitat de l'usuari i la seva correspondència amb la sessió. El procés d'autenticació es reitera en aquest darrer pas descrit fins que l'usuari decideix executar l'acció de tancar la sessió descrita en el següent apartat.

Per tal d'entendre millor el funcionament proposat, en el requadre que segueix a aquestes línies es mostra l'esquelet amb les instruccions que acabarien de definir el procés d'autenticació i gestió de la sessió. Cal assenyalar que és aquesta part bàsica la que s'anirà incloent dins d'altres esquemes adaptant-la convenientment per a cadascun d'ells.

Autenticació de l'usuari i comprovació de la sessió

4. U realitza les operacions següents:
   a. Preparar els valors propis de l'acció que s'estigui realitzant obtenint valors_acció.
   b. Xifrar Id_U i N'_G junt als valors de l'acció amb la clau pública de G obtenint $P_G[Id_U, N'_G, valors\_acció]$.
   c. Enviar $P_G[Id_U, N'_G, valors\_acció]$ a G.

5. G realitza les operacions següents:
   a. Desxifrar $P_G[Id_U, N'_G, valors\_acció]$ amb la clau privada de G obtenint Id_U, N'_G i els valors propis de l'acció que s'estigui realitzant; $S_G [P_G[Id_U, N'_G, valors\_acció]]$.
   b. Executar el procediment verificarUsuariSessio, i si no retorna error llavors
      i. Executar l'acció demanada.
      ii. Si convé, enviar els resultats a U.
   c. Sinó
      i. Retornar error.

6. U realitza les operacions següents:
   a. Si el pas 6 no retorna error llavors
      i. Realitzar els passos adients segons l'acció que s'estigui realitzant.
   b. Sinó
      i. Mostrar error.

4.4.2 Tancament de sessió

Aquesta acció permet a un usuari finalitzar l'ús de l'aplicació de manera segura, eliminant la sessió establerta amb el gestor.

Finalització de la sessió assignada a l'usuari

1. U realitza les operacions següents

1. Executar el procediment **peticioSimple** obtenint $P_G[Id_U, N'_G]$.
2. Enviar $P_G[Id_U, N'_G]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment2_2**.

procediment2_2

1. Desxifrar $P_G[N'_G, Id_U]$ amb la clau privada de G obtenint N'_G i Id'_U; $S_G[P_G[N'_G, Id_U]]$.
2. Obtenir la sessió $P_G[N_G, Id_U]$ mitjançant una consulta a la base de dades fent servir Id'_U.
3. Desxifrar $P_G[N_G, Id_U]$ amb la clau privada de G obtenint N_G i Id_U; $S_G[P_G[N_G, Id_U]]$.
4. Si $Id_U = Id'_U$ llavors
    a. Si $N_G = N'_G$ llavors
        i. Esborrar de la base de dades el testimoni de sessió lligat a Id_U corresponent a l'usuari U.
    b. Sinó –l'usuari no supera l'autenticació–
        i. Retornar error.
5. Sinó –la sessió no pertany a l'usuari–
    a. Esborrar de la base de dades el testimoni de sessió lligat a Id_U corresponent a l'usuari U.

4.4.3 Consulta d'un historial

El protocol de consultar un historial pot ser iniciat tant per un pacient com per un metge. Si l'acció és realitzada per un pacient, el resultat que se li mostra són les dades del seu propi historial. Un metge, en canvi, pot demanar veure l'historial d'un pacient, un objectiu que s'acaba acomplint amb dues variants:

- Si es verifica que el pacient seleccionat resta al seu càrrec, el metge té accés a tota la informació de l'historial, ja siguin dades públiques o privades.

- Si es comprova que el metge no està actualment assignat al pacient triat, l'accés a l'historial queda restringit a tan sols les dades menys confidencials.

Com ja s'ha comentar al capítol 2, la distinció entre dades públiques i privades es concreta amb la simplificació que s'estableix al situar la informació que es genera en una visita com a confidencial, i tota la resta com a no confidencial. És a dir, un metge que miri l'historial d'un dels seus pacients pot veure dades com el seu nom, el seu grup sanguini i tota la informació de les visites amb les que compta; un metge que miri l'historial d'un pacient al qual no tracti pot obtenir dades com el seu nom o el seu grup sanguini, però res de les visites.

En general, en quant al tractament d'errors per a aquest i a la resta de protocols, se sol actuar mostrant un missatge indicant quin és el problema i finalitzant l'acció sense poder-la acabar. Es vol destacar, però, que en el cas de detectar al pas 3 que la seqüència de les visites es errònia,

la part recuperada de l'historial és mostrada a l'usuari, indicant-li alhora quines són les visites que falten.

Consulta de l'historial d'un pacient per un metge o pel propi pacient

1. U realitza les operacions següents

1. Seleccionar un pacient Id'_U l'historial del qual es vol consultar.
2. Executar el procediment **procediment3_1** obtenint $P_G[N'_G, Id_U, Id'_U]$.

procediment3_1

1. Xifrar N'_G, Id_U i Id'_U amb la clau pública de G obtenint $P_G[N'_G, Id_U, Id'_U]$.
2. Retornar $P_G[N'_G, Id_U, Id'_U]$.

3. Enviar $P_G[N'_G, Id_U, Id'_U]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment3_2** obtenint $P_U[H]$.

procediment3_2

1. Desxifrar $P_G[N'_G, Id_U, Id'_U]$ amb la clau privada de G obtenint N'_G, Id_U i Id'_U; $S_G[P_G[N'_G, Id_U, Id'_U]]$.
2. Executar el procediment **verificarUsuariSessio**, i si no retorna error llavors
    a. Obtenir el certificat de l'usuari U mitjançant una consulta a la base de dades fent servir Id_U.
    b. Aconseguir el grup –metge o pacient– de l'usuari U a partir del seu certificat.
    c. Aconseguir la clau pública de l'usuari U a partir del seu certificat.
    d. Si (U és un pacient P) i ($Id_U = Id'_U$) –un pacient vol consultar el seu historial– o si (U és un metge M) i (Id'_U pertany a un pacient al càrrec de U) –un metge vol consultar l'historial d'un pacient seu– llavors –es té accés a tot l'historial–
        i. Obtenir les dades públiques i privades de l'historial H mitjançant una consulta a la base de dades fent servir Id'_U.
        ii. Desxifrar la part privada de l'historial H –les visites– amb la clau privada de G obtenint H'; $S_G[H]$.
        iii. Xifrar H' amb la clau pública de U obtenint $P_U[H']$.
        iv. Retornar $P_U[H']$.
    e. Si (U és un metge M) i (Id'_U pertany a un pacient que no es troba al càrrec de U) –un metge vol consultar l'historial d'un pacient que no és seu– llavors –es té accés a part de l'historial–
        v. Obtenir les dades públiques de l'historial H mitjançant una consulta a la base de dades fent servir Id'_U.
        vi. Xifrar H amb la clau pública de U obtenint $P_U[H]$.
        vii. Retornar $P_U[H]$.
    f. Sinò –no es té accés a res de l'historial–
        viii. Retornar error.

2. Enviar $P_U[H]$ a U.

3. U realitza les operacions següents

1. Executar el procediment **procediment3_3** obtenint les dades públiques de l'historial H.

procediment3_3

1. Desxifrar $P_U[H]$ amb la clau privada de U obtenint H; $S_U[P_U[H]]$.
2. Verificar la signatura digital de G per a Xf.
3. Verificar la signatura digital de G per a les dades mèdiques.
4. Retornar H.

2. Si les dades retornades per G contenen visites llavors
   a. Obtenir els certificats dels metges que han signat les visites.
   b. Executar el procediment **obtenirVisites** obtenint les dades privades de l'historial H.
3. Mostrar H.

4.4.4 Consulta dels pacients assignats a un metge

Aquest protocol defineix l'acció iniciada per un metge de poder consultar quins pacients són tractats per ell. El resultat obtingut és un llistat amb totes les identificacions i els noms corresponents a aquests pacients.

Consulta de l'historial d'un pacient per un metge o pel propi pacient

1. U realitza les operacions següents

1. Executar el procediment **peticioSimple** obtenint $P_G[Id_U, N'_G]$.
2. Enviar $P_G[Id_U, N'_G]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment4_2** obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.

procediment4_2

1. Desxifrar $P_G[N'_G, Id_U]$ amb la clau privada de U obtenint N'_G i Id_U; $S_U[P_G[N'_G, Id_U]]$.
2. Executar el procediment **verificarUsuariSessio**, i si no retorna error llavors
   a. Obtenir el certificat de l'usuari U mitjançant una consulta a la base de dades fent servir Id_U.
   b. Aconseguir el grup –metge o pacient– de l'usuari U a partir del

seu certificat.
c. Aconseguir la clau pública de l'usuari U a partir del seu certificat.
d. Si (U és un metge M) llavors
    i. Executar el procediment **obtenirPacients** obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
    ii. Retornar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
e. Sinó
    i. Retornar error.

2. Enviar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$ a M.

3. U realitza les operacions següents

1. Executar el procediment **llegeixLlistaPacients** obtenint $\{ P_1, ..., P_n \}$.
2. Mostrar $\{ P_1, ..., P_n \}$.

4.4.5 Inserció de dades a l'historial mèdic

El protocol que efectua la inserció de dades a un informe mèdic permet que un metge hi pugui emmagatzemar la informació generada en les diverses visites que es vagin produint amb el pacient propietari d'aquell historial.

Inserció de dades a l'historial mèdic d'un pacient per part d'un metge

1. U realitza les operacions següents

1. Seleccionar un pacient Id'_U per al qual es vol inserir una visita al seu historial i indicar les dades a introduir –observacio i recepta– a la visita Vm.
2. Executar el procediment **procediment5_1** obtenint $P_G[N'_G, Id_U, Vm, S_M[Vm]]$.

procediment5_1

1. Signar Vm amb la clau privada de M obtenint $S_M[Vm]$.
2. Xifrar N'_G, Id_U, Vm i $S_M[Vm]$ amb la clau pública de G obtenint $P_G[N'_G, Id_U, Vm, S_M[Vm]]$.
3. Retornar $P_G[N'_G, Id_U, Vm, S_M[Vm]]$.

3. Enviar $P_G[N'_G, Id_U, Vm, S_M[Vm]]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment5_2**.

procediment5_2

1. Desxifrar $P_G[N'_G, Id_U, Vm, S_M[Vm]]$ amb la clau privada de G obtenint N'_G, Id_U, Vm, $S_M[Vm]$; $S_G[P_G[N'_G, Id_U, Vm, S_M[Vm]]]$.

2. Executar el procediment verificarUsuariSessio, i si no retorna error llavors
    a. Obtenir el certificat de l'usuari U mitjançant una consulta a la base de dades fent servir Id_U.
    b. Aconseguir el grup –metge o pacient– de l'usuari U a partir del seu certificat.
    c. Aconseguir la identificació Id'_U del pacient P a partir de Vm.
    d. Aconseguir la clau pública de l'usuari U a partir del seu certificat.
    e. Si (U és un metge M) i (Id'_U pertany a un pacient al càrrec de U) llavors
        i. Verificar la signatura digital $S_M[Vm]$ amb la clau pública de M; $P_M[S_M[Vm]]$.
        ii. Si la verificació no retorna error llavors
            1. Executar el procediment guardarVisita.
        iii. Sinó
            1. Retornar error.
    f. Sinó
        i. Retornar error.

4.4.6 Obtenció dels pacients del sistema

Aquest protocol pretén satisfer el requeriment establert de que un metge pot consultar l'historial de qualsevol pacient –amb més o menys privilegis per a la visualització–, situació que es planteja de forma similar a la del protocol 4, però sense la restricció de només extreure aquells pacients propis.

Obtenció dels pacients del sistema per part d'un metge

1. U realitza les operacions següents

1. Executar el procediment **peticioSimple** obtenint $P_G[Id_U, N'_G]$.
2. Enviar $P_G[Id_U, N'_G]$ a G.

2. G realitza les operacions següents

1. Executar el procediment **procediment6_2** obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.

procediment6_2

1. Desxifrar $P_G[N'_G, Id_U]$ amb la clau privada de U obtenint N'_G i Id_U; $S_U[P_G[N'_G, Id_U]]$.
2. Executar el procediment verificarUsuariSessio, i si no retorna error llavors
    a. Obtenir el certificat de l'usuari U mitjançant una consulta a la base de dades fent servir Id_U.
    b. Aconseguir el grup –metge o pacient– de l'usuari U a partir del seu certificat.
    c. Aconseguir la clau pública de l'usuari U a partir del seu certificat.

d. Si (U és un metge M) llavors
    i. Executar el procediment obtenirTotsPacients obtenint $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
    ii. Retornar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$.
e. Sinó
    i. Retornar error.

2. Enviar $P_M[\{ P_1, ..., P_n \}, S_G[\{ P_1, ..., P_n \}]]$ a M.

3. U realitza les operacions següents

1. Executar el procediment **llegeixLlistaPacients** obtenint $\{ P_1, ..., P_n \}$.
2. Mostrar $\{ P_1, ..., P_n \}$.

Capítol 5
Representació de les dades

5.1 Introducció

Durant el funcionament habitual del sistema es genera diversa informació que cal guardar o moure entre els clients i el servidor. Per a emmagatzemar i manipular aquestes dades s'han d'estructurar d'alguna manera establint un determinat format, i en aquí és on resulta útil una tecnologia com XML.

L'estàndard XML és un llenguatge d'etiquetes de propòsit general que compta amb la característica de permetre definir els elements a utilitzar, fet que possibilita l'adaptació als requeriments d'una situació qualsevol. Bàsicament pot veure's com una eina amb capacitat d'ajustar-se per a crear a conveniència una estructura contenidora de dades.

Que sigui un estàndard també el fa més valuós, ja que hi ha multitud d'eines preparades per a treballar amb ell, la qual cosa facilita el desenvolupament i, en cas de necessitat futura, pot fer que la comunicació amb altres sistemes sigui més còmoda.

5.2 Disseny

Els documents XML són estructures de text pla que es troben restringits a contenir només aquest tipus de dades, amb la qual cosa és fa obligatòria la conversió a text de tota la informació que es vulgui introduir en ells.

Tal aspecte no presenta cap inconvenient de consideració, però es vol destacar que les dades binàries amb les que es treballa –com les signatures digitals, per exemple– es codifiquen per a ser convertides a caràcters imprimibles i per tant introduïbles dins d'una estructura XML.

Base64 és la codificació usada per a aquest propòsit, la qual transforma qualsevol classe d'informació a una cadena de text utilitzant un rang de 64 caràcters diferents[1].

Comentar també que en el sistema s'espera tractar informació amb caràcters fora del conjunt ASCII original, amb símbols i lletres com accents o la ç, o fins i tot es podrien requerir caràcters propis de llengües d'altres territoris –podria tenir-se un pacient de nom João, per exemple–. Per tal de manegar sense problemes aquesta situació, s'utilitza la codificació UTF-8 per als documents XML.

Aquest és un estàndard per a la representació de caràcters que abasta amb escreix la totalitat de símbols que es podrien requerir mai. De fet, UTF-8 és capaç de dedicar fins a 4 bytes per a representar un caràcter, el que fa superar tranquil·lament la quantitat de 4 mil milions de símbols representables.

Una qualitat notable de UTF-8 és que no utilitza una xifra fixa de bytes per a cada caràcter, sinó que s'adapta segons el que convingui fent que el document XML que es generi sempre ocupi el mínim possible. Per a l'ús habitual que s'estima que es farà del sistema, amb una utilització principal de l'alfabet llatí amb el català i castellà com a idiomes primers, els caràcters quedaran formats per un byte, principalment, o com a màxim dos bytes[2].

5.3 Estructura dels documents

En les següents seccions es mostren les estructures dels diferents documents XML que s'utilitzen en aquest projecte. Per a tot ells es té cura de que satisfacin les definicions bàsiques de la normativa XML i així disposar de documents considerats com a ben formats per tal de complir correctament amb l'estàndard.

Poden organitzar-se els documents segons l'ús que se'n fa d'ells en quatre grups: petició d'un servei, resposta a una petició de servei, dades per a conservar a la base de dades i arxius de configuració del sistema. A continuació es presenta una taula a on pot observar-se la classificació d'aquests grups.

Taula 3. Tipus de documents XML usats en el sistema

	Generat pel client	**Generat pel servidor**
Comunicació	Petició de servei	Resposta a la petició de servei
Emmagatzematge	Dades per a la base de dades	
		Arxius de configuració

Comentar que les etiquetes que contenen dades són aquelles que es mostren desplegades –`<etiqueta>dades</etiqueta>`– mentre que les etiquetes que es presenten tancades –`</etiqueta>`– no compten amb cap dada per al document concret.

5.3.1 Petició

S'emmarquen en aquest apartat tots aquells documents generats per un usuari i dedicats a la transmissió d'informació enfocada a demanar un determinat servei al gestor.

---

[1] Se suggereix consultar l'entrada de la Viquipèdia sobre la codificació Base64 per tal de trobar més informació sobre les seves característiques.
Base64, Viquipèdia, http://es.wikipedia.org/wiki/Base64

[2] Pot aprofundir-se en el coneixement de la codificació UTF-8 es proposa consultar la corresponent fitxa de la Viquipèdia.
UTF-8, Viquipèdia, http://en.wikipedia.org/wiki/UTF-8

L'estructura bàsica que segueix un document de petició **(figura 2)**, a partir de la qual es formen tots ells, compta amb les següents etiquetes:

- **sessio:** Camp destinat a omplir-se amb les dades del testimoni de sessió creat per a un usuari. El gestor identifica i valida a l'emissor d'una petició mitjançant la informació inclosa en aquesta etiqueta.

- **document:** Camp preparat per a contenir diversa informació necessària per a acomplir l'acció que s'estigui demanant –per exemple, si es fa una petició per a obtenir l'historial d'un pacient, aquest camp ha de contenir la identificació d'aquest pacient–.

- **signatura:** Camp pensat per a guardar la signatura de la informació continguda dins de l'etiqueta "document", amb la intenció de ser usat en aquelles situacions a on convé verificar l'autenticitat de les dades rebudes al servidor.

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio>

            ...

      </sessio>
      <document>

            ...

      </document>
      <signatura>

            ...

      <signatura>
</peticio>
```

Figura 2. Document XML bàsic per a les peticions

Document per a la petició d'autenticació, generat durant el procediment procediment1_1.

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio />
      <document>
            <aleatoriUsuari>
            </aleatoriUsuari>
            <identificacioUsuari>
            </identificacioUsuari>
      </document>
      <signatura />
</peticio>
```

Figura 3. Document XML per a la petició d'autenticació

Document per a una petició a on tan sols es requereixin les dades de sessió, generat durant el procediment peticioSimple.

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio>
            <aleatoriGestor>
            </aleatoriGestor>
            <identificacioUsuari>
            </identificacioUsuari>
      </sessio>
      <document />
      <signatura />
</peticio>
```

Figura 4. Document XML per a una petició simple

Document per a la petició de l'historial d'un pacient, generat durant el procediment procediment3_1.

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio>
            <aleatoriGestor>
            </aleatoriGestor>
            <identificacioUsuari>
            </identificacioUsuari>
      </sessio>
      <document>
            <identificacioHistorial>
            </identificacioHistorial>
      </document>
      <signatura />
</peticio>
```

Figura 5. Document XML per a la petició de l'historial d'un pacient

Document per a la petició d'inserció d'una visita a l'historial d'un pacient, generat durant el procediment procediment5_1.

```
<?xml version="1.0" encoding="UTF-8"?>
<peticio>
      <sessio>
            <aleatoriGestor>
            </aleatoriGestor>
            <identificacioUsuari>
            </identificacioUsuari>
      </sessio>
      <document>
            <visita>
            </visita>
      </document>
      <signatura />
</peticio>
```

Figura 6. Document XML per a la petició d'inserció d'una visita

5.3.2 Resposta

Aquest apartat engloba a tots aquells documents dedicats a transferir d'informació que són originats pel gestor com a resposta a una petició de servei prèvia per part d'un usuari.

L'estructura essencial que segueix un document de resposta **(figura 7)** i a partir de la qual tots ells es conformen, compta amb les etiquetes mostrades seguidament:

- **document:** Camp ideat per a contenir tota aquella informació que serveix com a resposta a la petició d'un usuari.

- **signatura:** Camp preparat per a guardar la signatura de la informació continguda dins de l'etiqueta "document", amb la intenció de ser utilitzada per un usuari en els moments a on cal verificar l'autenticitat de les dades rebudes des del servidor.

```
<?xml version="1.0" encoding="UTF-8"?>
<resposta>
      <document>

            ...

      </document>
      <signatura>

            ...

      <signatura>
</resposta>
```

Figura 7. Document XML bàsic per a les respostes

Document per a la resposta a la petició d'autenticació, generat durant el procediment procediment1_2.

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

Figura 8. Document XML per a la resposta a la petició d'autenticació

El document per a la resposta a la petició de consulta d'un historial, generat durant el procediment procediment3_2, varia segons el grau d'autoritat de l'usuari que l'ha demanat. Si es disposa de plena autoritat per a accedir a les dades de l'historial, el document inclou totes les dades que el conformen **(figura 9)**; si tan sols es disposa d'autoritat parcial, només s'inclou en el document aquella informació de caràcter menys confidencial **(figura 10)**.

```
<?xml version="1.0" encoding="UTF-8"?>
<resposta>
      <document>
            <autoritat>
            </autoritat>
            <cognom1>
            </cognom1>
            <cognom2>
            </cognom2>
            <nom>
            </nom>
            <dni>
            </dni>
            <grupSanguini>
            </grupSanguini>
            <signaturaDades>
            </signaturaDades>
            <numeroSerieUltim>
            </numeroSerieUltim>
            <signaturaNumeroSerieUltim>
            </signaturaNumeroSerieUltim>
            <visites>
                  <visita>
                        <numeroSerie>
                        </numeroSerie>
                        <metge>
                        </metge>
                        <data>
                        </data>
                        <dadesVisita>
                        </dadesVisita>
                        <signaturaVisita>
                        </signaturaVisita>
                  </visita>

                  ...

                  <visita>
                        <numeroSerie>
                        </numeroSerie>
                        <metge>
                        </metge>
                        <data>
                        </data>
                        <dadesVisita>
                        </dadesVisita>
                        <signaturaVisita>
                        </signaturaVisita>
                  </visita>
            </visites>
      </document>
      <signatura />
</resposta>
```

Figura 9. Document XML per a la resposta a la petició de consulta d'un historial demanada amb autoritat total

```
<?xml version="1.0" encoding="UTF-8"?>
<resposta>
      <document>
            <autoritat>
            </autoritat>
            <cognom1>
            </cognom1>
            <cognom2>
            </cognom2>
            <nom>
            </nom>
            <dni>
            </dni>
            <grupSanguini>
            </grupSanguini>
            <signaturaDades>
            </signaturaDades>
      </document>
      <signatura />
</resposta>
```

Figura 10. Document XML per a la resposta a la petició de consulta d'un historial demanada amb autoritat parcial

Document per a la resposta a la petició del llistat de pacients que un metge té assignats o per a la resposta a la petició del llistat de pacients totals del sistema, generat durant el procediment prepararPacients.

```
<?xml version="1.0" encoding="UTF-8"?>
<resposta>
      <document>
            <pacient>
                  <cognom1>
                  </cognom1>
                  <cognom2>
                  </cognom2>
                  <nom>
                  </nom>
                  <dni>
                  </dni>
            </pacient>

            ...

            <pacient>
                  <cognom1>
                  </cognom1>
                  <cognom2>
                  </cognom2>
                  <nom>
                  </nom>
                  <dni>
                  </dni>
            </pacient>
```

```
      </document>
      <signatura>
      </signatura>
</resposta>
```

Figura 11. Document XML per a la resposta a la petició d'un llistat de pacients

5.3.3 Dades

Els documents pensats per a contenir informació amb la idea de ser guardats a la base de dades, ja siguin originats per un usuari o pel gestor, apareixen en aquesta secció.

A continuació es mostren les etiquetes amb les que compta l'estructura bàsica que segueix un document de dades **(figura 12)**:

- **document:** Camp pensat per a contenir tota la informació útil que es desitja emmagatzemar.

- **signatura:** Camp destinat a guardar la signatura de la informació continguda dins de l'etiqueta "document", amb la intenció de ser utilitzada en les situacions a on és necessari verificar l'autenticitat de les dades.

```
<?xml version="1.0" encoding="UTF-8"?>
<dada>
      <document>

            ...

      </document>
      <signatura>

            ...

      <signatura>
</dada>
```

Figura 12. Document XML bàsic per a les dades

Document destinat a guardar les dades de sessió, generat durant el procediment procediment1_2.

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

Figura 13. Document XML per a les dades de sessió

Document pensat per a emmagatzemar les dades de visita, generat durant el procediment procediment5_1.

```
<?xml version="1.0" encoding="UTF-8"?>
<dada>
      <document>
            <identificacioPacient>
            </identificacioPacient>
            <observacio>
            </observacio>
            <recepta>
            </recepta>
      </document>
      <signatura>
      </signatura>
</dada>
```

Figura 14. Document XML per a les dades de visita

5.3.4 Configuració

Els documents descrits en aquesta secció pretenen conservar guardada diversa informació relativa a la configuració del sistema.

La següent és l'estructura bàsica que presenta un document de configuració **(figura 15)**, la qual queda omplerta amb les etiquetes convenients per a cada cas.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracio>

      ...

</configuracio>
```

Figura 15. Document XML bàsic per a les configuracions

Document per a la configuració del servidor, contingut en un arxiu de nom `configuracioServidor.xml` situat al servidor.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracio>
      <!-- PKCS#12 -->
      <arxiuP12>
            gestor.p12
      </arxiuP12>

      <!-- RMI -->
      <URLstub> <!-- URL de la localitzacio dels arxius stub,
➥important no deixar-se el / final -->
            file:/C:/eclipsePFC/eIAIK/
      </URLstub>
      <RMIHost> <!-- host a on s'executa RMIregistry -->
```

```
            localhost
      </RMIHost>
      <RMIPort> <!-- port pel qual s'accedeix al host a on
➥s'executa RMIregistry -->
            1099
      </RMIPort>
</configuracio>
```

Figura 16. Document XML per a la configuració del servidor

Document per a la configuració del client, contingut en un arxiu de nom `configuracioClient.xml` situat als clients.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracio>
      <!-- RMI -->
      <RMIHost> <!-- host a on s'executa RMIregistry -->
            localhost
      </RMIHost>
      <RMIPort> <!-- port pel qual s'accedeix al host a on
➥s'executa RMIregistry -->
            1099
      </RMIPort>
</configuracio>
```

Figura 17. Document XML per a la configuració dels clients

Capítol 6
Comunicació entre els components

6.1 Introducció

Segons la definició del projecte comentada en anteriors capítols, existeix la necessitat de que el sistema treballi de forma distribuïda per a poder executar l'aplicació des de qualsevol ordinador connectat a la xarxa. El tipus de funcionament que s'espera és el de Client-Servidor, sent el gestor del sistema qui assumeix aquest darrer paper a l'encarregar-se de donar servei a tots els metges i pacients, els quals actuen com a clients.

S'ha d'utilitzar un protocol de comunicació per tal de que les diferents màquines que participin en el sistema s'entenguin i l'aplicació funcioni de la forma remota prevista. Aquest protocol podria ser dissenyat i programat especialment per al projecte, o, alternativament, podria usar-se algun dels diversos mecanismes existents com són CORBA o SOAP, per dir-ne un parell de ben coneguts.

Entre les llibreries incloses dins del JDK, el programari utilitzat per a la programació Java del projecte, s'hi inclouen un conjunt d'APIs relatives a l'RMI o Remote Method Invocation. Aquestes classes ofereixen una manera senzilla i còmoda de comunicar aplicacions executades en diferents màquines virtuals, ja estiguin situades en un únic ordinador separades simplement de forma lògica, o, més interessant per al que es busca, separades de forma física entre diversos ordinadors connectats a una xarxa.

La limitació més rellevant de RMI respecte a altres tecnologies és el seu enfocament a treballar amb aplicacions Java, és a dir, no permet la comunicació entre programes que no s'hagin creat amb aquest llenguatge[1]. De totes maneres això no resulta un problema degut a que per a la implementació de totes les aplicacions només s'utilitza aquest llenguatge, i la seva facilitat d'ús

[1] S'apunta l'existència de la tecnologia RMI-IIOP, la qual intenta unificar tot allò considerat com a millor tant del concepte RMI com del CORBA. Podria considerar-se en part com un RMI amb capacitat per a comunicar aplicacions de diferents llenguatges.

junt amb el fet de que les llibreries necessàries siguin incloses dins del JDK de forma estàndard, ja fan que RMI sigui una bona solució per a la realització d'aquesta part del projecte.

6.2 Funcionament de RMI

La idea d'aquesta tecnologia és la de disposar d'un servidor que publiqui determinats objectes amb l'ajuda d'un servidor RMI per tal de que un client pugui usar-los. La declaració d'aquests objectes es realitza mitjançant interfícies, les quals permeten fer constar al client quins són els serveis disponibles.

Són tres els elements necessaris per a establir una comunicació mitjançant RMI:

- **Interfície:** Interfície amb la definició de les funcions i mètodes per a ser executats des del servidor de manera remota.

- **Objecte:** Classe amb la implementació definida per la interfície comentada en el punt anterior.

- **Stub:** Classe amb la implementació de la interfície però, en comptes d'implementar la funcionalitat que s'espera dels mètodes definits, tal i com fa l'objecte, el que realitza és preparar una crida per mitjà de la xarxa a aquest darrer per tal d'usar els seus mètodes.

La interfície i l'stub han de ser accessibles tant per a la banda del client com per a la del servidor per a que totes dues parts hi puguin accedir. L'objecte, en canvi, només hauria d'estar al costat del servidor convertint-se així en un objecte remot.

Quan un usuari requereix executar una funció, l'stub obté els resultats de l'execució utilitzant el mètode de l'objecte i els retorna com si els hagués generat ell mateix. La comunicació amb el servidor es realitza de forma transparent, amb la qual cosa un client no acaba distingint si aquell objecte que ha fet servir és seu o si l'ha usat des del servidor a través d'un stub.

6.3 Implantació de RMI al sistema

S'ha dit que cal que els clients tinguin accés a la classe stub, però relacionat amb aquest aspecte és possible definir dues maneres de funcionament per a un sistema implementat amb RMI: fer que els usuaris disposin en tot moment de tal arxiu, o fer que es carregui de forma dinàmica des del servidor.

En els següents apartats es mostren les accions dutes a terme per a la implantació de la tecnologia RMI en el sistema pensant en la primera manera de fer, és a dir, incloent el fitxer stub directament a l'aplicació dels clients. Igualment, en cas de convenir utilitzar l'altra forma, els canvis a fer serien mínims.

Les següents línies de codi pertanyen a l'aplicació del gestor i tenen com a objectiu, primer indicar la ruta de l'stub per tal de que el servidor RMI el pugui trobar, i després instanciar un objecte remot SFacana per a la seva publicació donant-se a conèixer amb un nom determinat. Per a aquest nom s'ha indicat l'adreça i port del servidor a on s'ofereix el servei RMI, que per defecte s'estableix al 1099.

```
import java.rmi.Naming;

...

System.setProperty("java.rmi.server.codebase",
      "file:/C:/PFC/bin/servidor/");
SFacana facana = new SFacana();
```

```
Naming.rebind("rmi://localhost:1099/SFacana", facana);
```

El programa de l'usuari ha d'executar el codi mostrat a continuació per tal d'obtenir l'objecte remot –de fet el que es rep és l'stub de l'objecte remot– i així executar els seus mètodes com amb qualsevol altra classe.

```
import java.rmi.Naming;

...

SFacanaInterficie facana =
      (SFacanaInterficie)
      Naming.lookup("rmi://localhost:1099/SFacana");
```

6.3.1 Interfície

Una interfície amb la intenció de ser remota ha d'heretar l'interfície Remote i a més, per a cadascun dels mètodes inclosos, s'ha d'indicar un llançament de l'excepció RemoteException. El descrit pot veure's en el següent fragment de codi corresponent a la interfície feta servir en aquest projecte.

```
import java.rmi.Remote;
import java.rmi.RemoteException;

...

public interface SFacanaInterficie extends Remote {

      public byte[] getCertificatGestor() throws RemoteException;

...

}
```

Una condició que s'ha de seguir per a l'ús dels mètodes de la interfície és que el tipus dels valors retornats i els tipus dels paràmetres que rebin han d'implementar la classe Serializable. En el codi que es realitza, però, es dóna la situació de que pràcticament tota la comunicació entre el client i el servidor es fa amb el tipus bàsic byte[] –els missatges es transmeten xifrats i s'envien directament en format binari–, amb la qual cosa ja no cal prendre cap mesura respecte a aquest tema. L'excepció són els mètodes que retornen els certificats del gestor i els metges, amb els que s'ha optat per passar-los també a byte[] a l'hora de transferir-los.

6.3.2 Objecte remot

Per tal de simplificar la codificació, l'objecte remot que implementa la interfície ha d'heretar la classe UnicastRemoteObject. En les línies de codi mostrades a continuació s'apunta quin camí segueix la implementació de SFacana.

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
```

```
...

public class SFacana extends UnicastRemoteObject implements
SFacanaInterficie {

      public SFacana() throws RemoteException {
      }

      public byte[] getCertificatGestor()
                                    throws RemoteException {

            ...

      }

      ...

}
```

6.3.3 Stub

Pot generar-se un stub de forma automàtica compilant l'objecte remot mitjançant l'eina rmic inclosa en el JDK, cosa que s'aconsegueix amb la instrucció que es mostra en el següent requadre.

```
rmic servidor.SFacana
```

El resultat d'aquesta sentencia és una classe anomenada `SFacana_Stub.class` la qual ha de trobar-se tant a la banda del client com a la del servidor per a que totes dues parts hi puguin accedir.

Capítol 7
Emmagatzematge de la informació

7.1 Introducció

La funcionalitat bàsica que un usuari espera del sistema és poder tractar la informació referent als expedients mèdics, amb les dades dels pacients, metges, visites i demés elements. Totes aquestes dades han de ser permanents, s'han de conservar en algun lloc per a poder ser consultades en qualsevol moment.

El sistema en sí, per al seu funcionament, també requereix l'ús de diversa informació que s'ha d'emmagatzemar, ja sigui d'una manera constant o simplement provisional, però sempre llesta per a poder obtenir-la quan sia necessari.

La manera més senzilla i eficient de realitzar tot això és comptar amb una base de dades, un sistema preparat per a l'emmagatzematge i gestió de grans quantitats d'informació. A més, utilitzant un SGBD d'un cert valor, com el que es fa servir en aquest projecte, es pot disposar d'una gestió dels usuaris que possibiliti atorgar diferents graus de poder i accés; considerant la importància a la seguretat que se li dóna al treball, aquest seria un punt a tenir en compte.

7.2 Disseny

En la imatge mostrada a la pàgina següent **(figura 18)** pot observar-se el disseny que presenta la base de dades del sistema amb les taules utilitzades i les relacions existents.

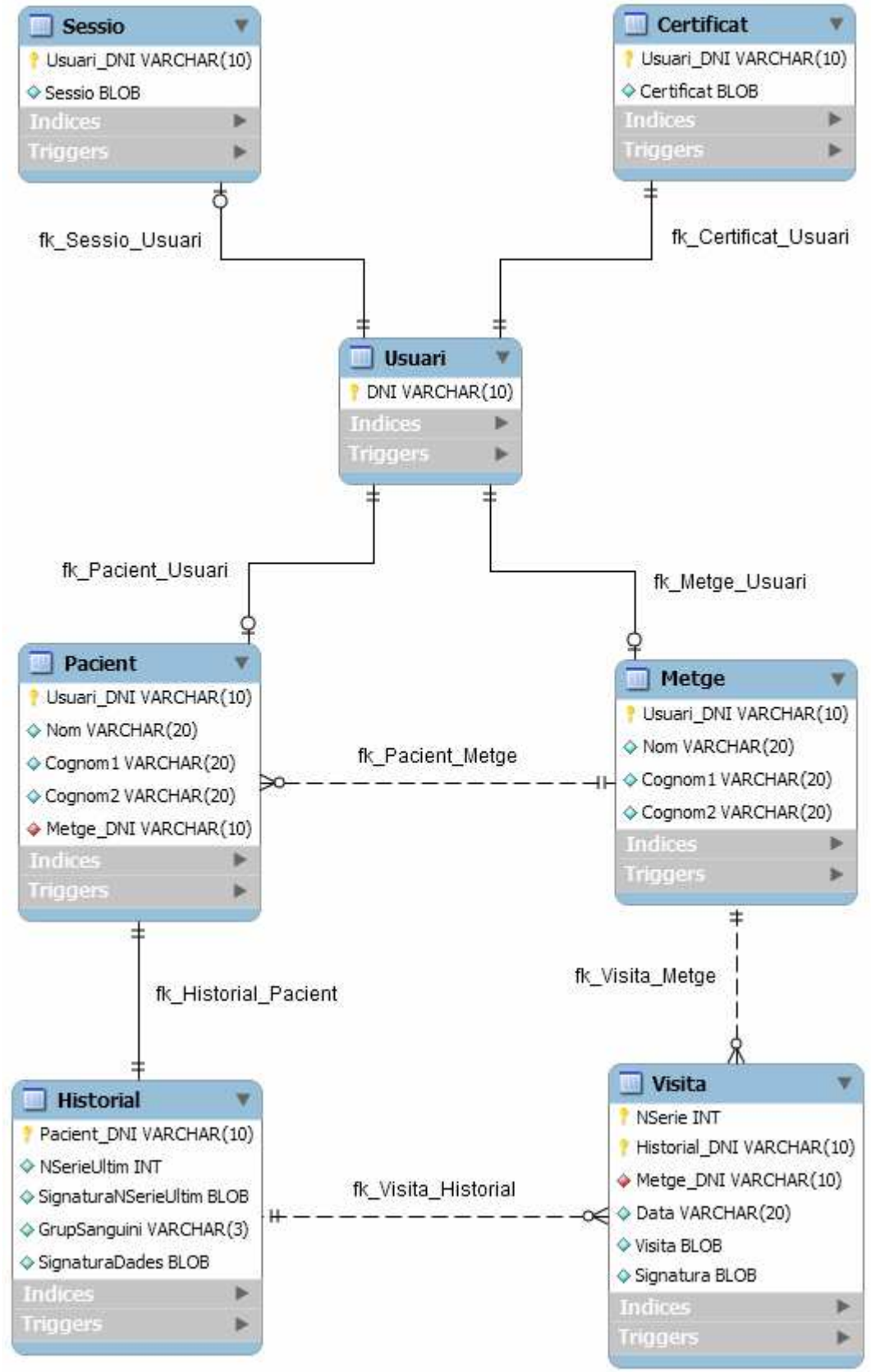

Figura 18. Model entitat-relació per al disseny físic de la base de dades

El disseny de la base de dades es realitza tenint presents les idees apuntades en la definició del sistema, la qual cosa porta a tenir un magatzem de dades preparat per a mostrar l'ús dels diferents protocols abans de per a ser un model per a un escenari real. Això es reflecteix principalment en el fet de que cada entrada de la taula Usuari ha d'estar relacionada obligatòriament amb la taula Metge o la taula Pacient de forma excloent; és a dir, tot usuari serà pacient o metge i només serà un d'ells, no es dóna l'oportunitat de que un metge pugui ser pacient dins del mateix centre a on exerceix.

Una altra conseqüència derivada d'aquest fet és que la relació entre les taules Usuari i Certificat és d'un a un; si un metge o un pacient només exerceixen el seu paper, mai tindran més d'un certificat.

Per una altra banda, s'ha definit que un pacient ha d'estar assignat a un metge i que també l'assignació pot canviar al llarg del temps. Tot i així, la relació entre la taula Client i la taula Metge és de molts a un, amb una clau forana per al client que identifica al metge. Tal circumstància fa que no es guardi cap històric pròpiament dit de per a quins metges ha anat passant un pacient, sinó que simplement es té en compte el metge assignat en el moment present –podrien obtenir-se, com a molt, els metges que haguessin creat una visita per al pacient, ja que a la taula Visita es conserva la informació del doctor que l'ha originada–.

7.3 Taules

De manera global, les taules s'han pensat intentant trobar un equilibri entre l'eficiència per al treball i el manteniment d'un sistema obert que permeti créixer d'una forma relativament senzilla.

Comentar que per a la totalitat de les taules s'ha triat com a clau primària el DNI d'un usuari, i aquesta situació permet una reducció en el consum de recursos del sistema ja que evita associar les taules en el moment de fer una consulta –les demandes d'informació principalment es realitzen en base a un usuari, i amb el seu DNI poden obtenir-se les dades vinculades a ell situades en qualsevol taula–.

7.3.1 Usuari

Taula que guarda dades relatives als usuaris que poden utilitzar el sistema.

Amb la informació que es necessita per a implementar aquest projecte, la taula Usuaris resulta una mica pobre ja que tan sols emmagatzema el DNI, quedant-se amb la principal funció d'assegurar i mantenir la integritat referencial entre les taules. Es comenta més sobre aquesta situació en la següent secció dedicada a la taula Certificat, ja que una alternativa vàlida seria haver unificat aquestes dues taules.

Un altre possible disseny per a Usuari hagués estat contenir els camps comuns de les taules Metge i Pacient, afegint tal vegada, per facilitar la feina, un camp que indiqués quin rol té l'usuari. També es pensa, però, que quan s'intentin obtenir dades d'un pacient o d'un metge sembla força probable que es requereixin camps tant d'aquestes taules com de la d'usuari. Es troba més eficient mantenir les dades a Metge i Pacient per tal d'estalviar recursos a l'hora d'extreure la informació.

Taula 4. Taula Usuari de la base de dades

Camp	Descripció
DNI	Clau primària. DNI identificador d'un usuari del sistema.

7.3.2 Certificat

Aquesta taula conté el conjunt de certificats dels usuaris del sistema.

La taula Certificat manté una relació d'un a un amb la taula Usuari, motiu pel que no hi hauria problema en integrar totes dues taules en una de sola, però, pensant en facilitar una possible ampliació futura del sistema, es creu convenient conservar-les per separat.

Es considera bastant cert que en cas d'incrementar la quantitat de camps de la taula Usuari, aquesta nova informació intenti ser obtinguda de forma independent als certificats, amb la qual cosa, aquests darrers, que no són dades especialment lleugeres, podrien fer innecessàriament feixuga l'acció de manipular la informació.

Taula 5. Taula Certificat de la base de dades

Camp	Descripció
Usuari_DNI	Clau primària i forana. DNI de l'usuari al que correspon el certificat.
Certificat	Certificat de l'usuari.

7.3.3 Sessió

Taula que emmagatzema les dades corresponents a les sessions obertes pels usuaris connectats al sistema.

Idòniament tan sols existiran entrades per als usuaris que estan utilitzant el sistema en un moment determinat, però podrien arribar a conservar-se sessions sense utilitat si els usuaris als quals es troben vinculats no han realitzat l'acció de tancar sessió. No s'ha establert cap sistema que detecti de forma automàtica si una sessió guardada s'està o no utilitzant –i si per tant pot ser eliminada– però de totes maneres s'ha definit que un usuari només pugui tenir una sessió com a màxim, amb el que no resulta un inconvenient greu.

Taula 6. Taula Sessio de la base de dades

Camp	Descripció
Usuari_DNI	Clau primària i forana. Identificador de l'usuari al que correspon la sessió.
Sessio	Document XML xifrat, corresponent al descrit a l'apartat 5.3.3, amb les dades de la sessió.

7.3.4 Pacient

Taula encarregada de contenir les dades personals dels pacients, així com el metges als quals es troben assignats.

Taula 7. Taula Pacient de la base de dades

Camp	Descripció
Usuari_DNI	Clau primària i forana. Identificador d'un usuari pacient.
Nom	Nom del pacient.
Cognom1	Primer cognom del pacient.

Cognom2	Segon cognom del pacient.
Metge_DNI	Clau forana. Identificador del metge al que es troba assignat el pacient.

7.3.5 Metge

Taula que conserva les dades personals corresponents als metges

Taula 8. Taula Metge de la base de dades

Camp	Descripció
Usuari_DNI	Clau primària i forana. DNI identificador del metge.
Nom	Nom del metge.
Cognom1	Primer cognom del metge.
Cognom2	Segon cognom del metge.

7.3.6 Historial

Aquesta taula emmagatzema les dades mèdiques d'un pacient considerades com a menys confidencials. També guarda signat el número de sèrie de la darrera visita introduïda a l'historial, cosa que permet verificar si a l'historial li falta o no cap visita.

La taula historial està relacionada mitjançant un lligam un a un amb la taula pacient, cosa que faria possible unificar sense inconvenients totes dues, però de nou, considerant també la idea d'una ampliació futura, es creu més adient mantenir per separat la informació mèdica i les dades personals d'un pacient.

Taula 9. Taula Historial de la base de dades

Camp	Descripció
Pacient_DNI	Clau primària i forana. Identificador del pacient al que correspon l'historial.
NSerieUltim	Número de visites de l'historial o, el que és equivalent, valor del camp NSerie corresponent a la darrera visita inserida per a l'historial.
SignaturaNSerieUltim	Signatura digital realitzada pel gestor per al conjunt de camps Pacient_DNI i NSerieUltim.
GrupSanguini	Grup sanguini del pacient.
SignaturaDades	Signatura digital realitzada pel gestor per al conjunt de camps Pacient_DNI i els que conformen les dades mèdiques –tan sols GrupSanguini en aquest cas–.

7.3.7 Visita

Aquesta taula emmagatzema les dades que conformen una visita.

Taula 10. Taula Visita de la base de dades

Camp	Descripció
NSerie	Clau primària. Número de sèrie identificador de la visita.
Historial_DNI	Clau primària i forana. Identificador de l'historial al que correspon la visita, que alhora, pel que significa el camp Pacient_DNI de la taula Historial, pot servir com a identificador del pacient al que correspon la visita.
Metge_DNI	Clau forana. Identificador del metge creador de la visita.
Data	Data de creació de la visita.
Visita	Document XML xifrat, corresponent al descrit a l'apartat 5.3.3, amb les dades de la visita.
Signatura	Signatura digital realitzada pel gestor per al conjunt de camps NSerie, Historial_DNI, Data i Visita.

7.4 Usuaris de la base de dades

Segons els actors definits en el capítol 2 i les accions que duen a terme, només l'administrador i el gestor treballaran directament amb la base de dades. Els altres dos tipus d'usuari, pacient i metge, interactuaran amb el gestor quan desitgin obtenir qualsevol dada que emmagatzemi; és a dir, el gestor actua com a un intermediari servint alhora de filtre que incrementa la seguretat de les dades.

Convé crear comptes d'usuari només per a aquells que han de tenir accés a la base de dades, i aquests comptes han de tenir definits quins són els privilegis que disposen. A la figura de l'administrador, el qual se suposa que ha de gestionar el sistema, se li concedeixen tots els drets possibles de la base de dades. Al gestor, en canvi, se li limiten més els seus moviments fent que tan sols pugui realitzar les operacions justes per a realitzar les accions establertes en els protocols.
A continuació poden observar-se els privilegis amb els que compta el gestor per a cadascuna de les taules.

Taula 11. Privilegis del gestor per a la base de dades

	Consultar	Inserir	Actualitzar	Eliminar
Certificat	✓	✘	✘	✘
Historial	✓	✘	✓	✘
Metge	✓	✘	✘	✘
Pacient	✓	✘	✘	✘
Sessio	✓	✓	✓	✓
Usuari	✓	✘	✘	✘
Visita	✓	✓	✘	✘

Capítol 8
Interfície gràfica

8.1 Introducció

Per a utilitzar el sistema cal disposar d'una interfície d'usuari i així poder executar de forma còmoda les diverses accions que s'ofereixen. Idealment les interfícies haurien de ser el més amigables, intuïtives i simples possibles, i aquestes són característiques que s'han de vigilar més si cal en una situació com la que es presenta en aquest projecte amb un ventall d'usuaris totalment enorme. De forma potencial qualsevol podria ser pacient i, per tant, qualsevol, amb més o menys habilitat per a utilitzar un ordinador, podria ser un usuari del sistema.

De totes maneres el principal objectiu d'aquest treball gira en torn a la seguretat i, tot i que no es descuida el tema de la interfície, el que principalment es busca al crear-la és mostrar les possibilitats del sistema i què és el que proporciona.

8.2 Disseny

Tal i com s'ha comentat anteriorment, existeixen dues aplicacions diferenciades per als usuaris, una dedicada a aquells que són metges i una altra pensada per als que són pacients.

8.2.1 Aplicació per als metges

El programa per als metges només pot ser utilitzat per aquest tipus d'usuari, és a dir, el certificat de la persona que el vol utilitzar ha de pertànyer al grup dels metges. Si no és així, encara que l'usuari intenti entrar amb un certificat i una contrasenya vàlides, no es permet accedir a les opcions del programa. Aquesta acció es du a terme a la pantalla d'autenticació.

Una vegada superat aquest procés, incloent la validació i creació de sessió per part del gestor, s'accedeix a la vista principal de l'aplicació, des d'on el metge pot dur a terme totes les accions que s'han definit per a ell.

Es pot sortir del programa, si és que es desitja finalitzar amb el seu ús, un cop s'hagi eliminat la sessió creada. Això pot aconseguir-se cridant directament l'opció de tancar sessió o simplement tancant la finestra principal, ja que du de forma implícita executar el tancament de sessió. Faci el que es faci, l'usuari torna a la pantalla d'autenticació, i des d'aquí sí que finalment es pot tancar l'aplicació.

La pantalla principal és usada per a crear altres vistes del programa, concretament la de llistar els pacients, consultar un historial i introduir una visita. Amb això s'ha establert que les accions que es puguin realitzar des de la finestra principal també siguin comunes per a les altres tres, i així, per exemple, tant a la vista principal com a la de llistar pacients es pot cridar a la pantalla d'informació del programa.

Per a visualitzar de forma gràfica l'estructura que segueix la interfície, es mostra un diagrama d'estats **(figura 19)** amb les finestres i vistes principals. Apuntar que, per a simplificar la imatge, s'han indicat només els fluxos únics obviant allò explicat a l'anterior paràgraf, i així, seguint amb l'exemple, des de la pantalla de llistar pacients no apareix cap fletxa a la finestra d'informació del programa, però no deixa de ser una acció que sí que es permet.

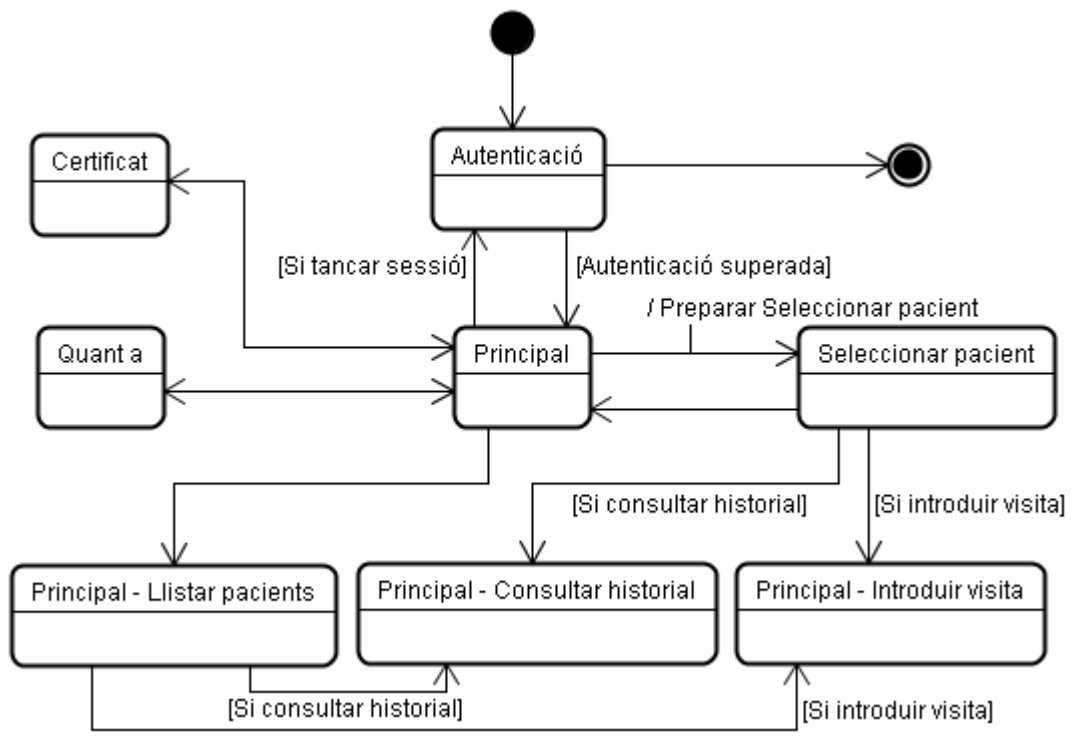

Figura 19. Diagrama d'estats de l'aplicació per als metges

8.2.2 Aplicació per als pacients

De forma similar al programa per als metges, l'aplicació per als pacients només permet ser utilitzada pels usuaris que puguin autenticar-se amb un certificat pertanyent al grup de pacients excloent a qualsevol altra part.

Aquesta aplicació pot veure's com una versió simplificada de la dels metges, amb menys opcions que les que permetia aquell i pràcticament amb la mateixa funcionalitat per a la resta

de les accions mantingudes. El seu ús pràcticament és el mateix i no val la pena fer cap comentari addicional.

En el següent diagrama d'estats **(figura 20)** es mostra l'esquema de la interfície d'aquest programa amb les vistes i fluxos existents.

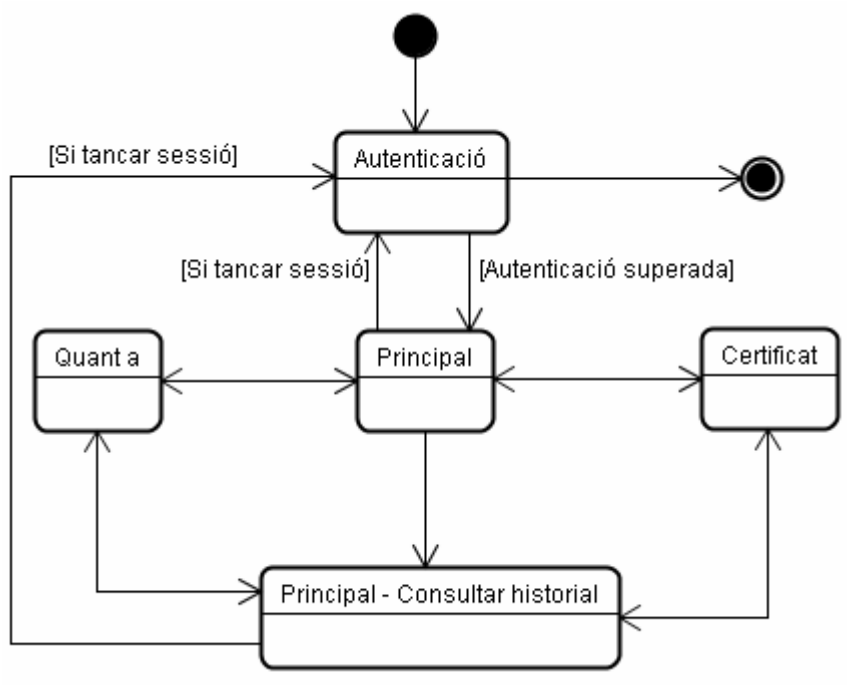

Figura 20. Diagrama d'estats de l'aplicació per als pacients

8.3 Pantalles

En els següents apartats es detallen les pantalles aparegudes en els diagrames d'estats tant de l'aplicació per als metges com del programa per als pacients.

Comentar que per a totes les finestres es té cura de les situacions d'error que es poguessin produir amb el seu ús, com pot ser intentar veure la visita d'un pacient sense haver seleccionat prèviament un, o provar de gravar una nova visita sense cap dada en ella. A aquesta classe d'errors se'ls hi suma tots aquells generats per capes inferiors del programa, ja sigui un error com no poder carregar l'arxiu PKCS#12, o un error com que el gestor denega la realització d'una acció per no disposar de suficients permisos.

Durant la utilització de l'aplicació també poden generar-se diferents missatges d'informació, indicant que una sessió ha finalitzat de forma correcta o que una visita s'ha gravat sense inconvenients, per exemple.

Tots aquests missatges, tant els d'error com els d'informació, es visualitzen a l'usuari per mitja de senzills quadres de diàleg.

8.3.1 Autenticació

La primera finestra que se li presenta a l'usuari que inicia el programa és la pantalla per a l'autenticació **(figura 21)**, pas que haurà de realitzar en primera instància per a poder accedir a la resta de l'aplicació.

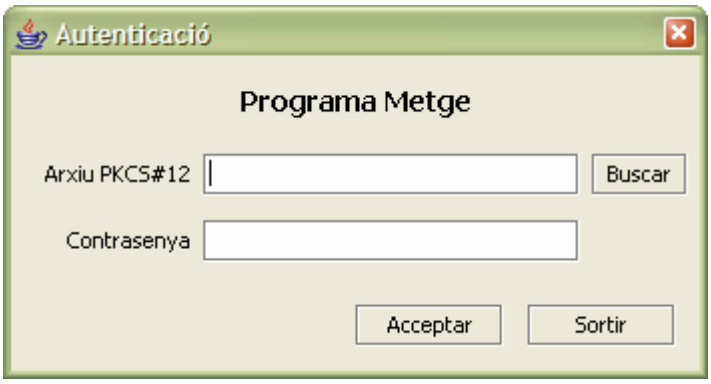

Figura 21. Pantalla d'autenticació

Per a autenticar-se l'usuari ha d'indicar la ruta de directoris a on es troba l'arxiu del seu certificat, ja sigui escribint-la directament al quadre de text corresponent o ajudant-se prement a sobre del botó "Buscar", acció que fa aparèixer un quadre de diàleg amb l'arbre d'arxius del sistema per a poder triar-ne un **(figura 22)**. També cal que s'indiqui la contrasenya corresponent al certificat seleccionat anteriorment. Una vegada omplerta aquesta informació, es pot polsar el botó "Acceptar" per a executar al procés en sí d'autenticació.

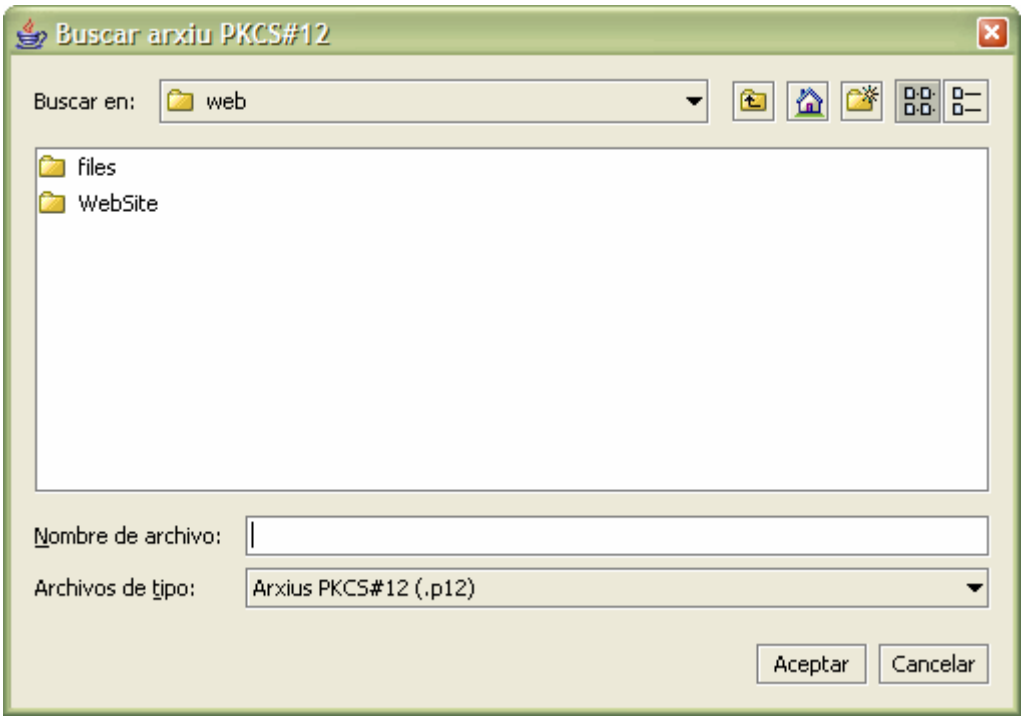

Figura 22. Pantalla de buscar arxius

8.3.2 Principal

Una vegada superada l'autenticació es mostra la finestra principal del programa **(figura 23)** des de la que es té accés a totes les accions que es permeten realitzar.

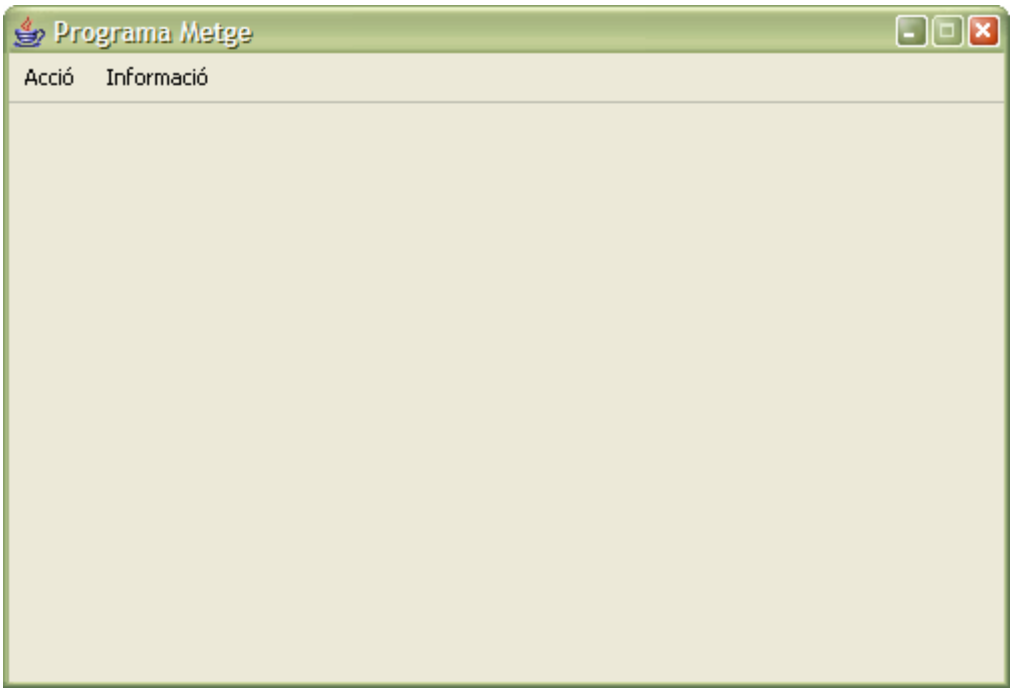

Figura 23. Pantalla principal

Aquesta pantalla disposa d'un menú que varia segons si es tracta del programa per als metges **(figura 24)** o el programa per als pacients **(figura 25)**.

Figura 24. Menú de la pantalla principal del programa per als metges

Seguidament s'expliquen les opcions que apareixen a l'aplicació enfocada als metges:

- **Acció**
  - **Llistar pacients:** Mostra la pantalla **(figura 26)** amb el llistat dels pacients que estan assignats al metge que està utilitzant el programa.
  - **Consultar historial:** Mostra un quadre de diàleg **(figura 27)** amb un llistat de tots els pacients existents al sistema, amb la idea de triar un per a poder consultar el seu historial.
  - **Introduir visita:** Mostra un quadre de diàleg **(figura 28)** amb el llistat dels pacients que estan assignats al metge que està utilitzant el programa, amb

la intenció d'escollir un per a poder introduir una nova visita en el seu historial.

- **Tancar sessió:** Executa el procés de finalitzar la sessió amb el servidor, tornant seguidament a la primera pantalla d'autenticació **(figura 21)**. Es procedeix de la mateixa manera si l'usuari tanca la pantalla principal amb qualsevol dels mètodes habituals –fent clic al botó X, per exemple–.

- **Informació**
  - **Certificat:** Mostra un quadre de diàleg **(figura 31)** amb diversa informació sobre el certificat amb el que s'ha entrat a l'aplicació.

  - **Quant a:** Mostra un quadre de diàleg **(figura 32)** amb informació sobre el programa.

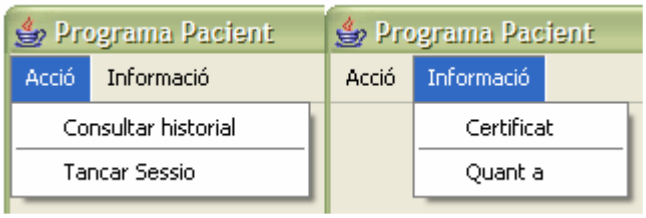

Figura 25. Menú de la pantalla principal del programa per als pacients

El programa destinat als pacients no compta amb cap opció afegida a les ja descrites per a l'aplicació dels metges, ans al contrari, les que són pròpies pels metges ja no hi apareixen: "Llistar pacients" i "Introduir visita".

El funcionament de la resta d'opcions és idèntic a l'explicat per al programa dels metges, a excepció de l'acció "Consultar historial" que varia lleugerament fent aparèixer directament la pantalla de consulta de l'historial **(figura 27)** en comptes del quadre de diàleg per a seleccionar un pacient **(figura 29)** –i és que un pacient no pot fer més que consultar el seu propi historial–.

8.3.3 Llistar pacients

Aquesta pantalla **(figura 26)**, només accessible als metges a l'aparèixer al clicar sobre l'opció del menú principal "Llistar pacients", presenta un llistat amb tots els pacients –DNI i nom sencer– que resten al càrrec del metge que s'ha autenticat i utilitza l'aplicació.

Des d'aquesta finestra és possible executar l'acció de consultar l'historial o afegir-hi una visita per a qualsevol dels pacients llistats, per al que simplement cal triar un dels pacients i prémer el botó "Consultar historial" o "Introduir visita" i així realitzar la respectiva acció. El botó "Consultar historial" condueix a la pantalla de consulta de l'historial **(figura 27)** mentre que el botó "Introduir visita" du a la finestra d'introducció de visites **(figura 28)**.

Remarcar que la base d'aquesta pantalla és la finestra principal **(figura 23)**, i que les opcions presents en aquella segueixen mantenint-se per a aquesta altra.

Figura 26. Pantalla de llistar pacients

8.3.4 Consultar historial

Un pacient pot arribar a la pantalla de consultar un historial marcant l'opció adient al menú principal. Un metge hi pot accedir tant des de la pantalla de llistar pacients prement el botó "Consultar historial", o per mitjà de la finestra de seleccionar un pacient per a mostrar el seu historial.

Es poden distingir dues parts per a aquesta vista: una superior que mostra dades de caire menys confidencial amb informació tant de tipus personal com mèdic –en concret s'imprimeix el nom, DNI i grup sanguini del pacient–, i una part inferior amb les dades mèdiques més privades –o, tal i com s'ha definit al quart apartat del segon capítol, tota la informació relativa a les visites–. Seguint l'esquema marcat pels protocols, no tothom té accés a aquestes darreres dades, i per tant les visites només es visualitzen si es dóna el cas.

Com a l'anterior pantalla, en aquí també es pren de base la finestra principal **(figura 23)** conservant les opcions que s'ofereixen en aquella.

Figura 27. Pantalla de consultar historial

8.3.5 Introduir visita

A la pantalla d'introduir visites hi poden arribar els metges a través del botó "Introduir visita" de la pantalla de llistar pacients **(figura 26)** o una vegada triat un pacient des del quadre de diàleg de seleccionar-ne un per a crear una nova visita **(figura 30)**.

En aquesta finestra hi apareixen un parell de quadres de text corresponents als camps d'observacions i recepta de la visita, els quals esperen ser omplerts pel metge amb les dades corresponents. També es visualitza una etiqueta informant de per a quin pacient es fa la nova visita. Finalment es mostra el botó "Gravar visita" que al ser premut desencadena l'acció per a emmagatzemar-la.

De nou, aquesta pantalla també pren com a base la finestra principal **(figura 23)**, i com a les últimes vistes presentades, també segueixen estant disponibles les opcions presentades en aquella.

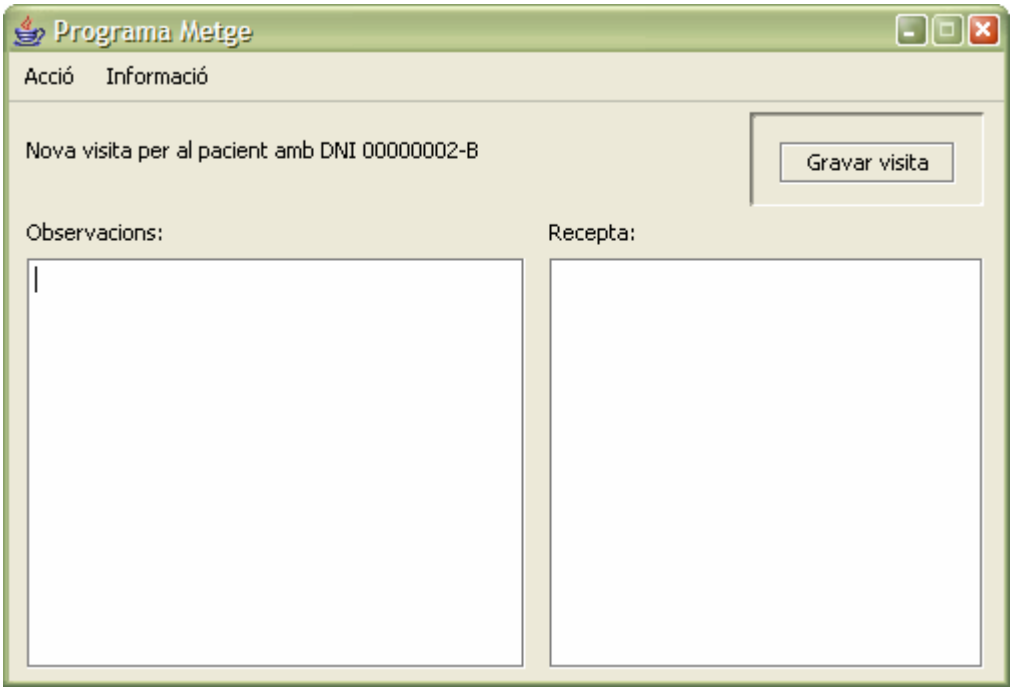

Figura 28. Pantalla d'introduir visita

8.3.6 Seleccionar pacient

La finestra de seleccionar pacient sorgeix per al mestre al marcar les opcions del menú "Consultar historial" o "Introduir visita". Aquest quadre de diàleg és un pas previ necessari per a poder arribar a les pantalles de consulta d'historial **(figura 27)** i d'introducció de visita **(figura 28)**, ja que és a on es permet triar un pacient per aplicar-li l'acció a executar.

La possibilitat d'escollir un pacient és diferent per a cadascuna d'aquestes dues accions esmentades, i mentre que un metge pot visualitzar l'historial de qualsevol pacient –amb més o menys restriccions per a obtenir la totalitat de les dades–, només pot crear una nova visita per als pacients que són seus. Això implica que la finestra varia lleugerament segons l'opció triada, adaptant-se segons el cas oferint tot el rang de pacients **(figura 29)** o només els propis del metge **(figura 30)**, i acabant conduint a una o altra pantalla una vegada polsat el botó "Acceptar".

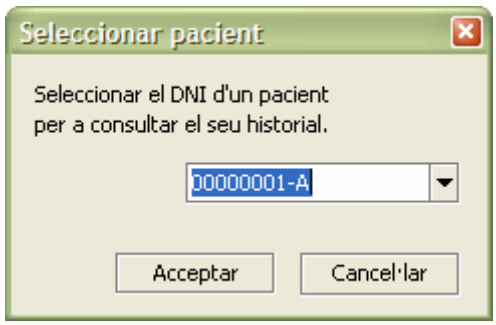

Figura 29. Pantalla de selecció de pacient per a la consulta de l'historial

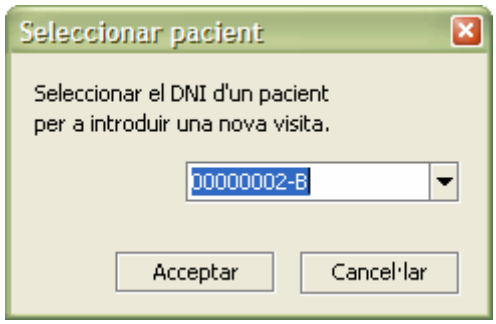

Figura 30. Pantalla de selecció de pacient per a la introducció d'una visita

8.3.7 Certificat

Aquest quadre de diàleg **(figura 31)** apareix al prémer l'opció "Certificat" del menú principal i senzillament mostra informació diversa sobre el propietari del certificat amb el que s'hagi autenticat per a entrar.

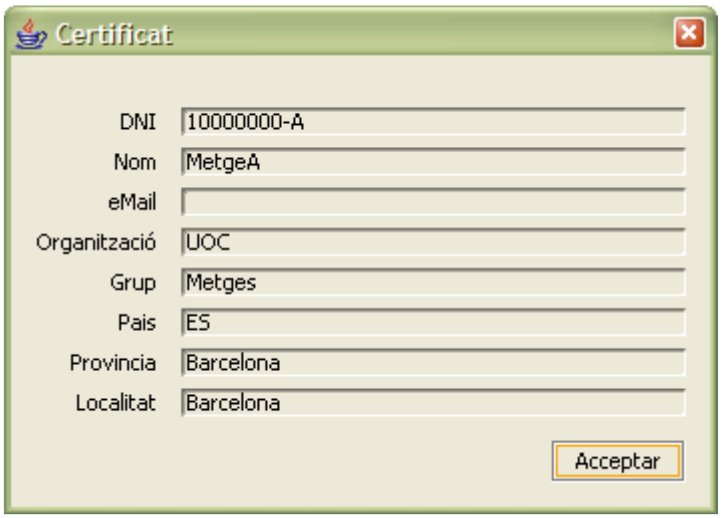

Figura 31. Pantalla d'informació del certificat

8.3.8 Quant a

A l'escollir l'opció "Quant a" del menú principal sorgeix aquest quadre de diàleg **(figura 32)** amb informació sobre el programa.

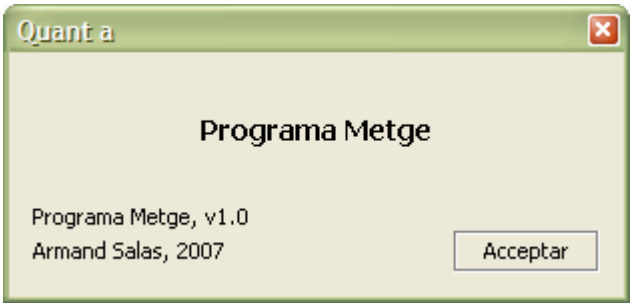

Figura 32. Pantalla d'informació del programa

Capítol 9
Disseny del sistema

9.1 Introducció

Aquesta secció intenta ser una continuació de l'estudi del sistema presentat en el segon capítol, però des d'un enfocament més tècnic i recopilant les valoracions vistes al llarg dels altres temes fins arribar a aquest punt.

De fet, com en aquell capítol, la informació aquí mostrada correspon a la part de disseny i planificació del sistema, un aspecte vital en qualsevol projecte al ser necessari per a poder abordar la seva execució, i per tant, un dels primers passos a realitzar. S'ha volgut, però, esperar a introduir els temes vists a la resta de capítols per tal de poder-los relacionar amb els següents apartats i així comprendre millor la informació que contenen.

9.2 Disseny

El sistema es dissenya seguint una solució típica d'arquitectura en tres capes:

- **Presentació:** És la capa dels usuaris metge i pacient, el programari que s'executa a les seves màquines. Aquesta capa té l'objectiu de capturar les peticions i paràmetres que indiquen els usuaris, així com de presentar la informació que el sistema crea –un historial que s'ha demanat veure, un missatge d'error, etc–. La lògica de negoci que es realitza en aquesta fase és totalment mínima, deixant el gruix de la feina per a la següent capa que és amb la que es comunica.

- **Negoci:** Aquesta capa la configura el programari del gestor i és a on es reben les sol·licituds enviades per la capa de presentació. En aquí resideix la lògica principal que ha de seguir el sistema al disposar de les diverses regles que marquen la funcionalitat i la seguretat definida. Es comunica amb la següent capa per tal d'acomplir el seu propòsit, i retorna els resultats que genera a la capa de presentació.

- **Dades:** Aquesta capa és formada pel SGBD i és l'encarregada de gestionar les dades de caràcter permanent del sistema, emmagatzemant, recuperant, modificant o eliminant la informació segons les indicacions de la capa de la lògica de negoci.

En el diagrama de desplegament **(figura 33)** pot observar-se l'esquema general que segueix el sistema amb les divisions per capes.

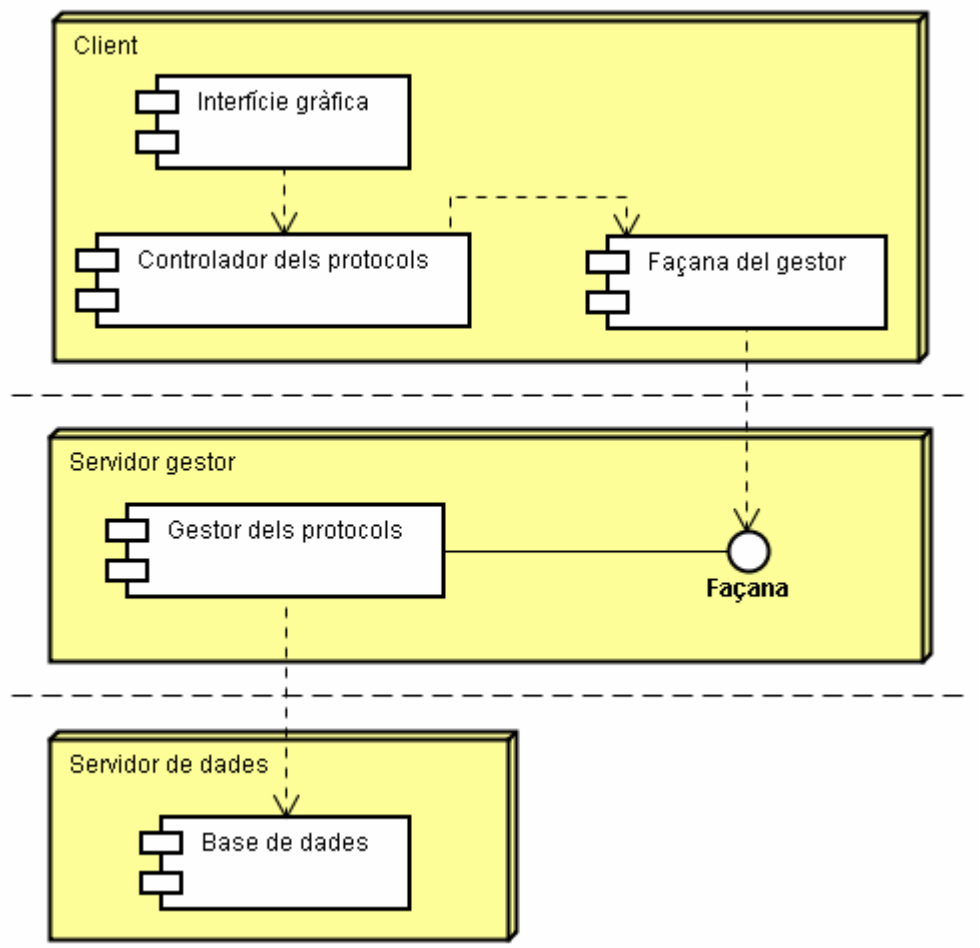

Figura 33. Diagrama de desplegament del sistema

9.3 Paquets i classes

En les següents seccions es mostra i descriu el disseny de les classes que conformen les aplicacions, juntament amb la indicació de la resta de classes amb les que es relacionen. S'apunta que, per tal d'estalviar espai, els gràfics amb els diagrames són força conceptuals i només intenten reflectir el funcionament global del projecte, obviant tot allò que no es pensa rellevant o mereixedor de ser destacat.

9.3.1 Client

A la pàgina següent **(figura 34)** es troba el disseny de classes de les aplicacions destinades als clients. Les classes pròpies dels clients són les situades dins dels paquets client i gui –pintats de color groc–. Alhora, les classes úniques utilitzades per al programa metge o per a l'aplicació pacient, es troben dins dels paquets del mateix nom –respectivament de color blau i vermell–.

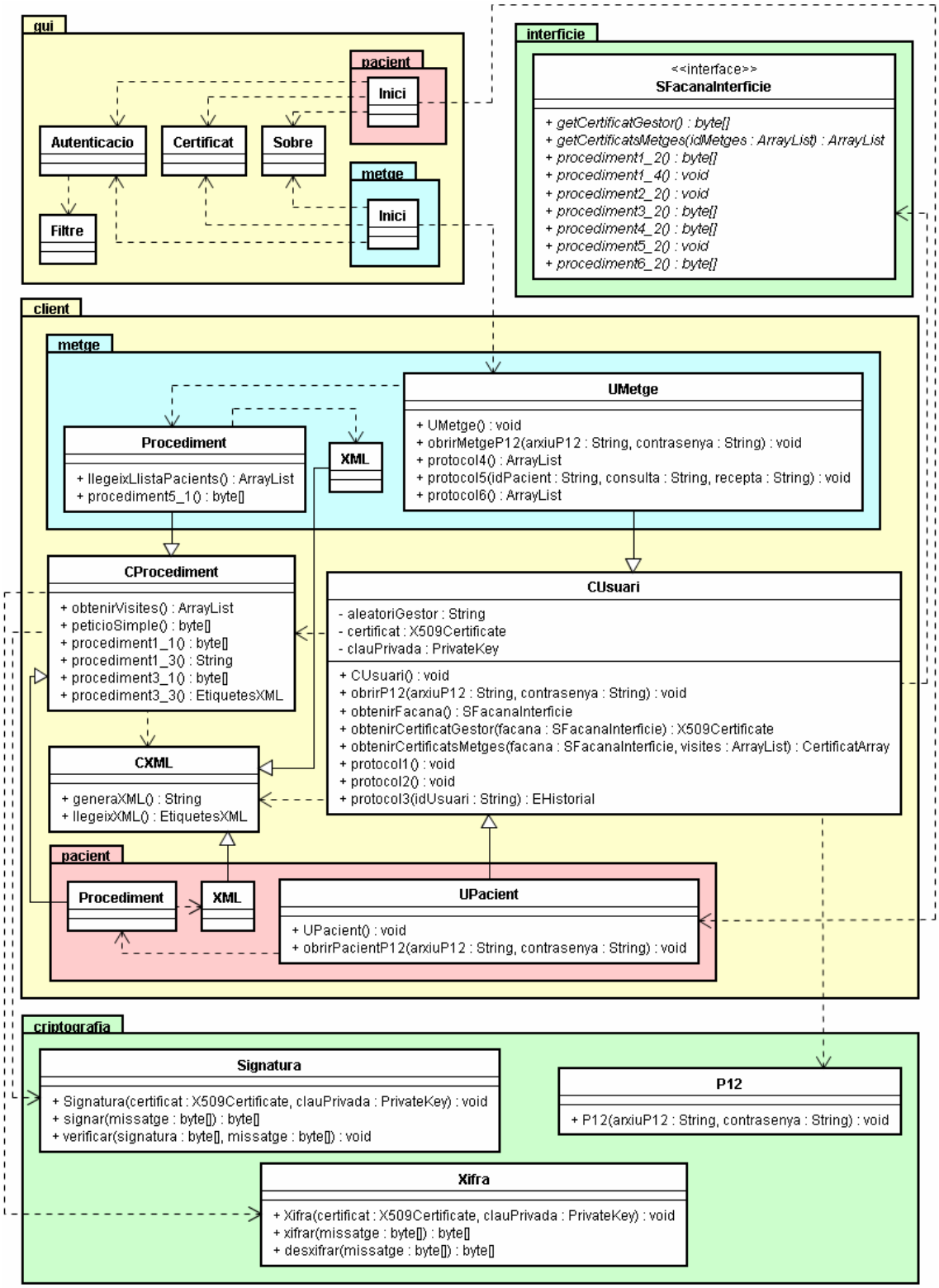

Figura 34. Diagrama de classes per a la part dels clients

Es procedeix a continuació amb la descripció d'aquestes classes:

- **client**
  - **CUsuari:** Classe principal d'usuari. Manté el certificat i la clau privada d'un usuari, així com el testimoni de sessió creat a l'autenticar-se. Compta amb mètodes per a obtenir la façana per a comunicar-se amb el gestor i els certificats d'aquest i els metges. També apareixen els protocols comuns a

metge i pacient definits en el capítol 4. Carrega, a més, l'arxiu de configuració dels clients.

- **CProcediment:** Classe contenidora dels procediments indicats al quart capítol per a poder executar els protocols de CUsuari.
- **CXML:** Classe encarregada de preparar les peticions cap al gestor i de llegir les seves respostes, totes elles formades amb documents XML tal i com s'ha definit al capítol 5.

- **metge**
  - **UMetge:** Classe principal de metge. Hereta la classe CUsuari i se li afegeixen els mètodes i protocols orientats a només ser executats per un metge.
  - **Procediment:** Classe amb els procediments marcats al capítol 4 per a l'execució dels protocols de UMetge. Hereta la classe CProcediment per tal de que UMetge pugui executar còmodament les funcions heretades de CUsuari.
  - **XML:** Classe destinada a preparar les peticions i llegir les respostes pròpies dels protocols de UMetge. Hereta la classe CXML per tal de que UMetge pugui executar còmodament les funcions heretades de CUsuari.

- **pacient**
  - **UPacient:** Classe principal de pacient. Hereta la classe CUsuari i se li sumen els mètodes pensats únicament per als pacients. Un pacient no disposa de cap protocol especial, i per tant no s'afegeix cap en aquí.
  - **Procediment:** Classe destinada a contenir els procediments dels possibles protocols que s'introdueixin a UPacient. Per ara, al no existir protocols, és una classe buida, però és usada per UPacient per a executar els mètodes obtinguts de CUsuari ja que hereta de CProcediment.
  - **XML:** Classe pensada per a preparar les peticions i llegir les respostes de serveis. Al no tenir protocols propis, de moment és una classe buida, però UPacient la fa servir per a executar els mètodes obtinguts de CUsuari ja que hereta de CXML.

- **gui**
  - **Autenticacio:** Classe que defineix i controla el comportament de la finestra descrita a l'apartat 8.3.1 per a l'autenticació.
  - **Filtre:** Classe utilitzada per Autenticacio en el moment d'escollir un arxiu mitjançant la finestra emergent, ja que permet establir un filtre de manera que només es mostrin els arxius d'interès –en aquest cas, els arxius d'extensió .p12–.
  - **Certificat:** Classe que defineix i controla el comportament de la finestra descrita a l'apartat 8.3.7 amb la informació del certificat carregat.
  - **Sobre:** Classe que defineix i controla el comportament de la finestra descrita a l'apartat 8.3.8 amb informació sobre el programa.

  - **metge**
    - **Inici:** Classe que defineix i controla el comportament de la finestra amb la pantalla principal, així com de les pantalles que prenen com a base la primera, per a l'aplicació destinada als metges. Aquesta classe inicia l'aplicació dels metges.

  - **pacient**
    - **Inici:** Classe que defineix i controla el comportament de la finestra amb la pantalla principal, així com de les pantalles que prenen com a base la primera, per a l'aplicació destinada als pacients. Aquesta classe inicia l'aplicació dels pacients.

9.3.2 Servidor

Seguidament es mostra el disseny amb les classes situades al servidor **(figura 35)**. En aquest cas, les classes que pertanyen a aquesta part corresponen a les contingudes dins dels paquets servidor i bd –color lila–.

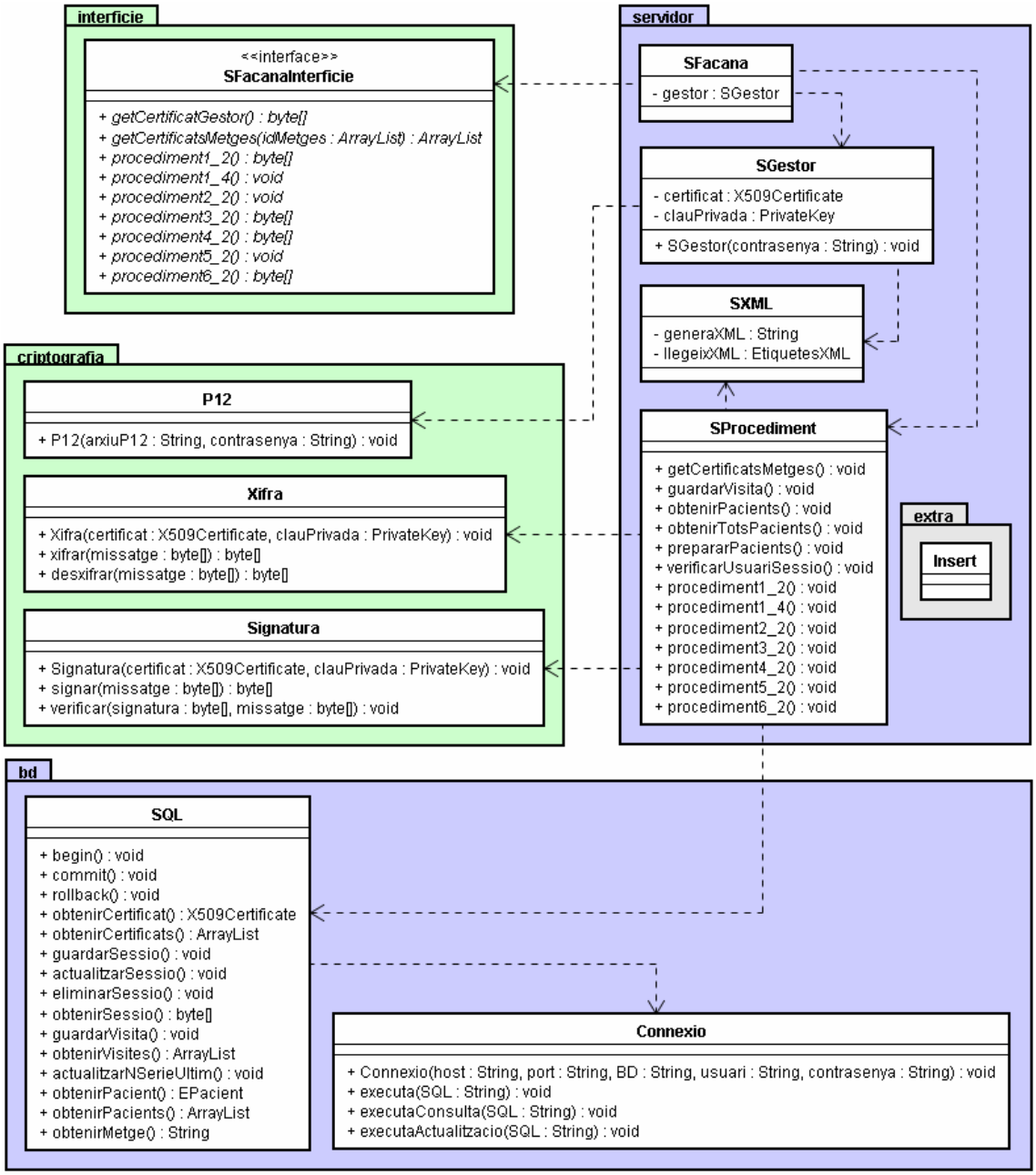

Figura 35. Diagrama de classes per a la part servidor

Tot seguit es descriuen les classes dels paquets afectats:

- **servidor**
  - **SGestor:** Classe destinada a carregar i mantenir el certificat i la clau privada del gestor. Carrega l'arxiu de configuració del servidor. Serveix també com a inici de l'aplicació del gestor.
  - **SFacana:** Classe que implementa la interfície SFacanaInterficie descrita en el següent apartat.

- **SProcediment:** Classe contenidora dels procediments indicats al quart capítol pensats per a ser executats pels gestor.
- **SXML:** Classe encarregada de llegir les peticions rebudes per part dels usuaris i de preparar les seves respostes, totes elles formades amb documents XML tal i com s'ha definit al capítol 5.

- **bd**
  - **Connexio:** Classe pensada per a establir una connexió amb el servidor de base de dades i executar les consultes que rebi. La connexió es realitza a través de l'usuari configurat per al gestor.
  - **SQL:** Classe contenidora de diversos mètodes per a preparar el conjunt de sentències SQL necessaris per a abordar la base de dades.

Dins del paquet servidor s'inclou un altre paquet anomenat extra, el qual posseeix la classe Insert. Aquesta és una classe que no intervé en el funcionament del sistema però s'aporta per a poder inicialitzar-lo còmodament, ja que compta amb un conjunt d'insercions a la base de dades amb tot d'informació necessària –usuaris, certificats, etc–. Aquesta seria una tasca a acomplir per l'administrador, i la connexió s'estableix utilitzant el seu usuari.

9.3.3 Paquets comuns

A les anteriors dues seccions s'han comentat les classes úniques per a clients i servidor, però als diagrames apareixen altres paquets –de color verd– amb classes fetes servir per totes dues parts. Comentar també que el sistema compta amb un altre paquet de nom util **(figura 36)**, el qual és usat per pràcticament totes les classes motiu pel que no s'ha inclòs.

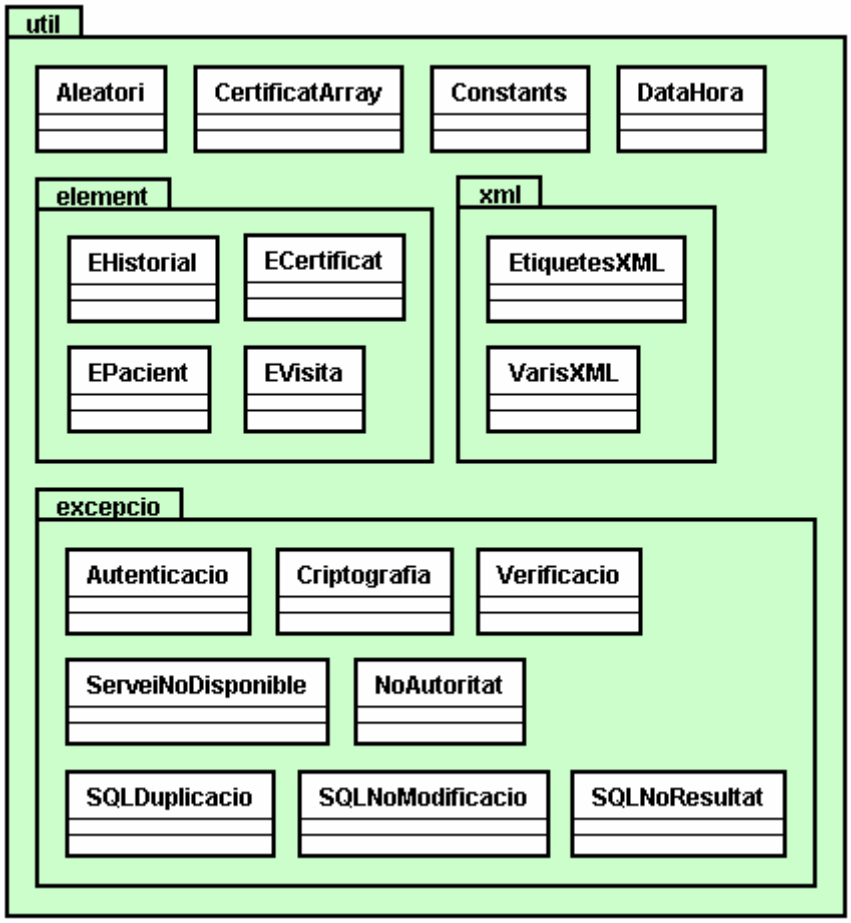

Figura 36. Paquet util

Totes les classes d'aquests paquets comuns es comenten a continuació:

- **interficie**
  - **SFacanaInterficie:** Classe que defineix la interfície de la classe usada com a façana del servidor. Compta amb les declaracions de tots els mètodes que un usuari pot requerir del gestor. Aquesta classe i la seva implementació SFacana, són el resultat d'aplicar el comentat al capítol 6 sobre RMI.

- **criptografia**
  - **P12:** Classe pensada per a carregar un arxiu PKCS#12 i extreure'n el certificat i la clau privada que conté.
  - **Xifra:** Classe encarregada de xifrar un missatge en clar o desxifrar un missatge xifrat.
  - **Signatura:** Classe encarregada de realitzar la signatura d'un missatge o de verificar la validesa d'una.

- **util**
  - **Aleatori:** Classe creada per a facilitar l'obtenció de valors generats aleatòriament.
  - **DataHora:** Classe creada per a facilitar l'obtenció de la data i l'hora.
  - **CertificatArray:** Classe ideada per a gestionar de forma simple un llistat de certificats.
  - **Constants:** Classe contenidora de diferents valors constants usats en diferents punts de les aplicacions.

- **xml**
  - **EtiquetesXML:** Classe pensada per a contenir els valors llegits d'un document XML.
  - **VarisXML:** Classe posseïdora de diferents funcions d'ús comú per a la resta de classes que treballen amb XML.

- **element**
  - **EHistorial:** Classe ideada per a contenir les dades que conformen un historial.
  - **EVisita:** Classe ideada per a contenir les dades que conformen una visita.
  - **EPacient:** Classe ideada per a contenir les dades d'un pacient.
  - **ECertificat:** Classe ideada per a contenir la informació inclosa en un certificat.

- **excepcio**

  El paquet excepcio conté diverses classes que defineixen diferents excepcions fetes servir per a controlar els possibles errors que es generin durant l'execució dels programes, tant des d'un punt de vista tècnic –si no es pot connectar a la base de dades, per exemple– com de seguretat seguint la lògica establerta –si un pacient intenta consultar un historial d'un altre pacient, per posar un cas–.

Capítol 10
Jocs de proves

10.1 Verificació de la PKI

La creació dels certificats mitjançant OpenSSL no és un procés que presenti complicacions, però es poden fer algunes petites proves que assegurin que els fitxers obtinguts són correctes.

Per a començar, es podria llegir el contingut dels certificats per tal de verificar que les dades que contenen corresponen amb les que es volien introduir. Es pot executar la comanda mostrada seguidament per a visualitzar, per exemple, la informació del certificat del gestor.

```
openssl x509 -in gestor.crt –text
```

El resultat d'aquesta instrucció es pot observar en el següent requadre.

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            89:fd:57:81:ef:cf:4e:24
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=ES, ST=Barcelona, L=Barcelona, O=UOC, OU=CAs,
➥CN=CA
        Validity
            Not Before: Oct 22 22:35:06 2007 GMT
            Not After : Oct 21 22:35:06 2008 GMT
        Subject: C=ES, ST=Barcelona, L=Barcelona, O=UOC,
➥OU=Gestors, CN=Gestor
        Subject Public Key Info:
```

```
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:e5:e3:ed:e8:c8:ea:02:f2:d0:17:7c:39:06:b8:
                    69:40:87:fc:db:0d:ea:56:d9:42:6e:78:fd:be:2e:
                    0c:78:ef:87:86:57:17:0b:8f:8c:68:c0:c8:d8:13:
                    74:90:59:18:e8:86:a3:fb:24:55:6b:bf:6d:e5:81:
                    34:79:7e:9b:31:10:38:46:a3:be:a5:f7:e9:8a:fa:
                    69:33:7b:2d:f0:51:16:ce:c5:b6:64:ef:66:64:f6:
                    fe:bb:a3:3b:65:b7:ce:46:eb:ea:18:22:c4:a0:e3:
                    7f:55:73:2c:dc:10:52:db:9e:d8:58:d1:71:8b:b4:
                    89:9f:61:77:f7:00:91:bd:47
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        10:c7:64:14:41:2a:91:fc:09:42:0a:b3:24:27:ee:bb:37:04:
        07:e4:21:4b:7b:d9:dc:d0:19:5f:7d:6f:ae:53:46:4e:4d:39:
        bb:1e:88:de:47:83:b2:2f:4d:94:55:33:86:53:47:fc:6b:a2:
        0e:a3:3f:d9:6e:6b:b9:30:19:3f:19:47:3f:d6:22:84:69:10:
        c3:cc:d2:a4:ca:e9:8c:28:8e:27:b5:77:c9:41:b7:10:0d:17:
        ff:e4:76:7e:85:ae:82:39:60:98:1b:80:01:e3:be:92:25:1f:
        27:a6:fb:bf:74:58:c2:20:b9:e7:11:e1:3b:7c:3a:33:87:a9:
        68:a7:48:f0:39:07:dc:bb:42:20:be:18:9b:36:71:86:10:39:
        e4:6f:58:36:64:c7:71:c6:53:29:2b:9e:08:4f:23:9c:07:ff:
        04:d4:77:f5:2e:7c:0c:ae:3d:02:de:49:ff:a4:fd:08:c1:96:
        95:92:43:2b:59:3b:24:91:ce:1d:cb:30:aa:e4:3a:d3:25:d7:
        9a:2f:01:1e:04:98:db:92:97:cf:05:25:70:79:35:94:db:b1:
        7c:9c:9e:e0:5c:8c:e2:58:62:80:5a:54:cd:5e:88:a5:9d:e4:
        d9:dc:83:83:b4:50:e7:54:e9:41:89:0b:73:f3:eb:73:6c:09:
        95:52:58:fa
-----BEGIN CERTIFICATE-----
MIICvDCCAaQCCQCJ/VeB789OJDANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJF
UzESMBAGA1UECBMJQmFyY2Vsb25hMRIwEAYDVQQHEwlCYXJjZWxvbmExDDAKBgNV
BAoTA1VPQzEMMAoGA1UECxMDQ0FzMQswCQYDVQQDEwJDQTAeFw0wNzEwMjIyMjM1
MDZaFw0wODEwMjEyMjM1MDZaMGYxCzAJBgNVBAYTAkVTMRIwEAYDVQQIEwlCYXJj
ZWxvbmExEjAQBgNVBAcTCUJhcmNlbG9uYTEMMAoGA1UEChMDVU9DMRAwDgYDVQQL
EwdHZXN0b3JzMQ8wDQYDVQQDEwZHZXN0b3IwgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAOXj7ejI6gLy0Bd8OQa4aUCH/NsN6lbZQm54/b4uDHjvh4ZXFwuPjGjA
yNgTdJBZGOiGo/skVWu/beWBNHl+mzEQOEajvqX36Yr6aTN7LfBRFs7FtmTvZmT2
/rujO2W3zkbr6hgixKDjf1VzLNwQUtue2FjRcYu0iZ9hd/cAkb1HAgMBAAEwDQYJ
KoZIhvcNAQEFBQADggEBABDHZBRBKpH8CUIKsyQn7rs3BAfkIUt72dzQGV99b65T
Rk5NObseiN5Hg7IvTZRVM4ZTR/xrog6jP9lua7kwGT8ZRz/WIoRpEMPM0qTK6Ywo
jie1d8lBtxANF//kdn6FroI5YJgbgAHjvpIlHyem+790WMIguecR4Tt8OjOHqWin
SPA5B9y7QiC+GJs2cYYQOeRvWDZkx3HGUykrnghPI5wH/wTUd/UufAyuPQLeSf+k
/QjBlpWSQytZOySRzh3LMKrkOtMl15ovAR4EmNuSl88FJXB5NZTbsXycnuBcjOJY
YoBaVM1eiKWd5Nncg4O0UOdU6UGJC3Pz63NsCZVSWPo=
-----END CERTIFICATE-----
```

També, amb la comanda següent, és factible realitzar la mateixa acció però amb la petició d'un certificat, en aquest cas la del pacient A.

```
openssl req -in pacientA.csr -text
```

De totes maneres, Openssl compta amb mètodes per tal d'oferir la possibilitat de verificar si aquests arxius s'han construït o no correctament. A continuació es pot veure la instrucció que comprova la fiabilitat de la petició del certificat del metge A, cosa especialment pràctica si s'usa abans d'acceptar aquesta petició i emetre el corresponent certificat.

```
openssl req -in metgeA.csr -verify -noout -config openssl.cnf
```

La següent és la comanda que confirma si el certificat originat ha estat signat per la CA de forma adequada.

```
openssl verify -CAfile CA.crt metgeA.crt
```

Aquest parell d'instruccions tenen com a sortida, si és que no hi ha cap inconvenient, un simple missatge fent constar que tot és correcte o, en cas de no ser així, informació indicant quin és el problema.

Una altra prova senzilla que es pot fer és arrencar el servidor web amb el que compta OpenSSL establint algun dels certificats per a la comunicació segura. La comanda que es pot observar a continuació permet realitzar l'acció descrita amb el certificat de la CA.

```
openssl s_server -cert CA.crt -key CA.key -www -pass pass:medic
```

Pot comprovar-se el funcionament indicant en un navegador l'adreça `https://localhost:4433`. Com que és la primera vegada que es carrega el certificat de la CA indicat, el programa pregunta si es desitja confiar en ell –en molts navegadors també es permet visualitzar la informació identificadora que conté–, i en acceptar-lo, s'hi mostra diversa informació que s'estableix en la comunicació SSL.

10.2 Proves del sistema

Cada element desenvolupat per al sistema s'ha anat provant durant la seva creació, ja sigui de forma unitària o participant i comunicant-se amb altres parts del sistema, assegurant l'obtenció dels comportaments i resultats esperats.

Tot i que la majoria de possibles errors no són recuperables, s'ha tingut força cura amb la seva gestió generant missatges descriptius per a cada situació. Amb tot el programari desenvolupat, en els següents apartats s'intenta mostrar algunes de les accions més comunes que podria realitzar l'usuari metge i uns pocs dels errors que se'n podrien generar.

Es fa constar que el presentat en aquí només es una petita mostra de les situacions controlades, i que per a la resta d'aplicacions se segueix la mateixa pauta.

10.2.1 Autenticar-se

L'autenticació és la primera i única acció que es pot fer a l'iniciar el programa. La identificació s'acompleix amb la indicació d'un arxiu PKCS#12 pertanyent a un metge i la contrasenya "medic".

Figura 37. Introducció de les dades per a autenticar-se

Si el procés finalitza de forma correcta sense provocar cap incidència, es presenta un missatge amb la conformitat per a seguidament passar a la pantalla principal.

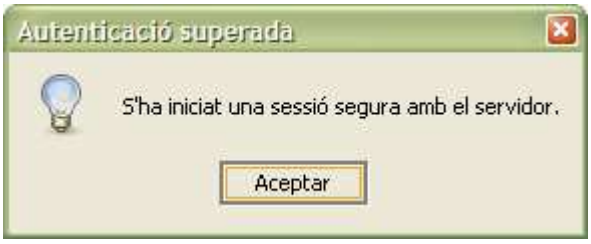

Figura 38. Missatge informant de l'establiment de sessió

Provant la robustesa de l'aplicació s'intenten forçar alguns errors, i així, si no s'introdueix la contrasenya correcta per a l'arxiu, a aquest no s'hi pot accedir i salta el següent missatge d'error.

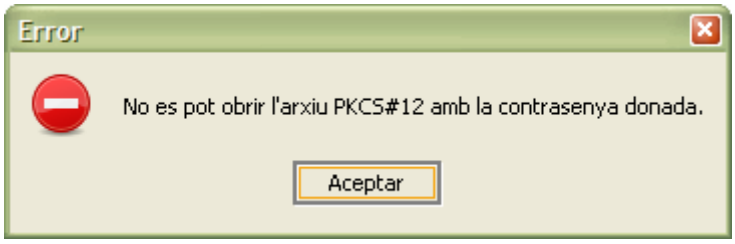

Figura 39. Missatge informant de la introducció d'una contrasenya no vàlida

L'aplicació metge només està pensada per a ser utilitzada pels usuaris que són metge, una comprovació aquesta, que es realitza en la pròpia aplicació client en el moment de carregar l'arxiu p12. El següent missatge es mostra quan s'intenta violar aquesta regla.

Figura 40. Missatge informant de la impossibilitat d'usar l'aplicació

Es prova de modificar el codi per tal de que el testimoni de sessió enviat pel gestor no coincideixi amb el que rep l'usuari, de manera que quan aquest el retorna al primer, el procés d'autenticació falla. Es pot veure aquesta situació en el següent missatge.

Figura 41. Missatge informant de no haver superat l'autenticació

10.2.2 Consultar un historial

Un metge que intenta consultar un historial pot obtenir diferents resultats segons les circumstàncies existents. Si prova d'accedir a les dades d'un pacient assignat al seu càrrec, el resultat és la obtenció de les dades al complert, tant les considerades com a públiques com les vistes com a privades. Les següents dues pantalles mostren l'historial de dos pacients assignats a un mateix metge, una situació vàlida per a mostrar totes les dades, encara que en aquest cas es troba que un dels historials no conté cap visita.

Figura 42. Autoritat total per consultar l'historial, però no conté visites

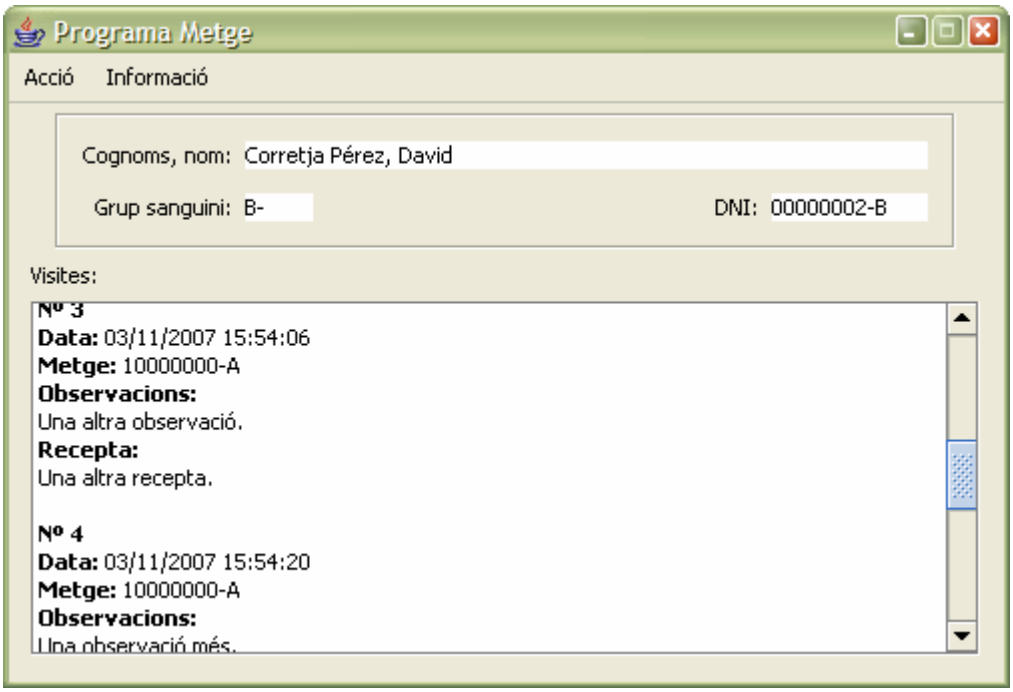

Figura 43. Autoritat total per consultar l'historial

També és possible que un metge provi d'obtenir l'historial d'un pacient que no té assignat, fet que condiciona que les dades obtingudes continguin només aquelles menys confidencials. La següent imatge és un exemple d'aquest estat, ja que el pacient que apareix no es tractat pel metge que mira l'informe mèdic.

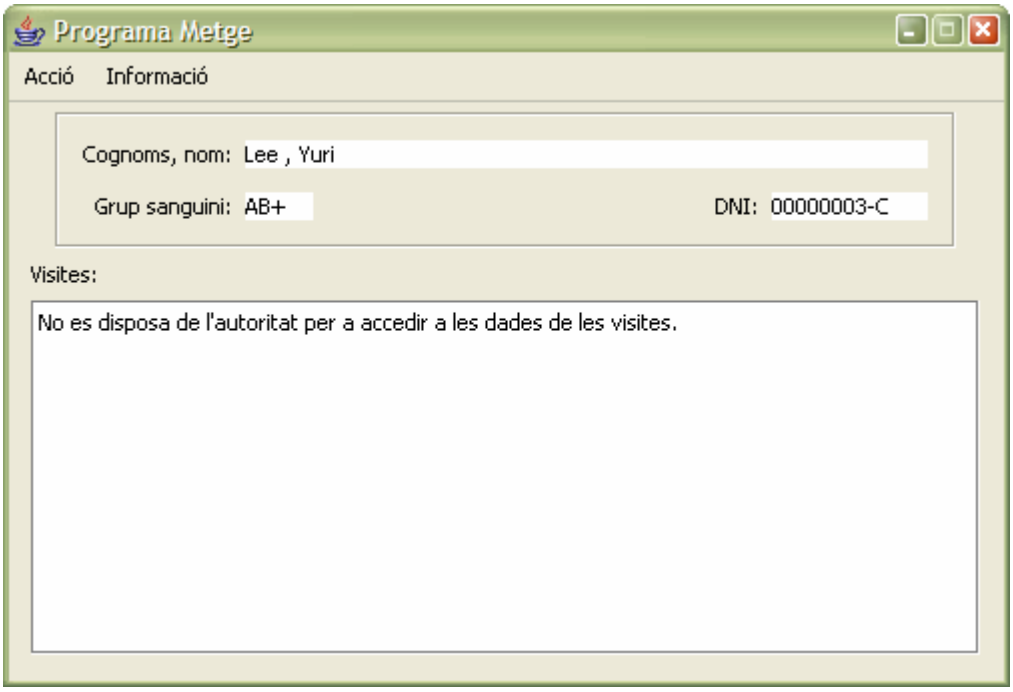

Figura 44. Autoritat parcial per consultar l'historial

Intentant provar a continuació algunes situacions que generin errors, es comenta que les dades mèdiques es troben signades, i podria comprovar-se la utilitat de la signatura en cas de que algú modifiqués aquesta informació accedint directament a la base de dades. Es canvia, per exemple, el grup sanguini d'un dels pacients, i l'error s'assenyala correctament mitjançant la següent finestra.

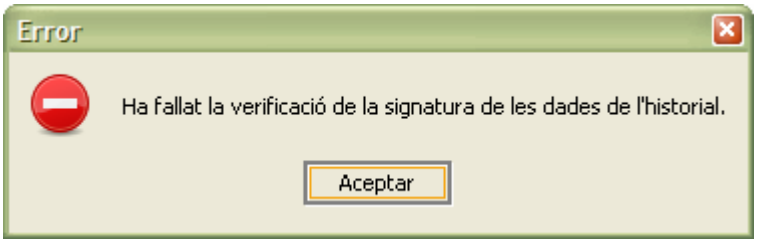

Figura 45. Missatge informant de l'error al verificar les dades

Un altre possible error és la detecció de que les visites no segueixen un ordre seqüencial, cosa que indicaria la falta d'alguna d'elles. En una situació com a l'anterior a on es comprova que les dades han estat modificades, no es pot fer gaire cosa més apart d'indicar la situació trobada, ja que no té gaire sentit presentar a l'usuari informació que pot ser falsa. La cosa canvia, però, amb una seqüència de visites a on l'error és que falten algunes d'elles; si les visites existents no contenen cap error encara es podrien aprofitar i ser mostrades.

Es prova d'accedir a la base de dades per eliminar directament alguna de les visites. S'escull concretament a un pacient amb 6 visites, i d'aquestes s'esborren la 4 i la 6, per exemple. El resultat al presentar l'historial, tal i com es visualitza a les següents imatges, és que s'ensenyen quines són les visites que falten, però la resta d'elles es mostren sense inconvenient.

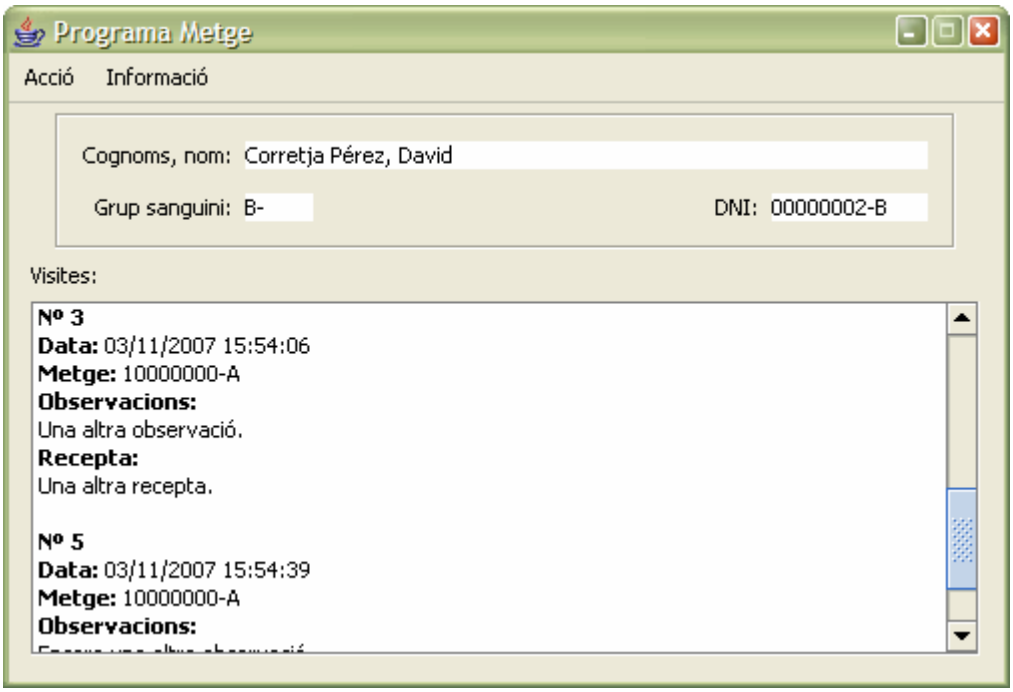

Figura 46. Historial amb falta de visites

Figura 47. Missatge informant de la falta de visites

També pot provar-se d'eliminar totes les visites, situació en la que directament és mostra que no es visualitza cap perquè no hi són.

Figura 48. Missatge informant de la falta de totes les visites

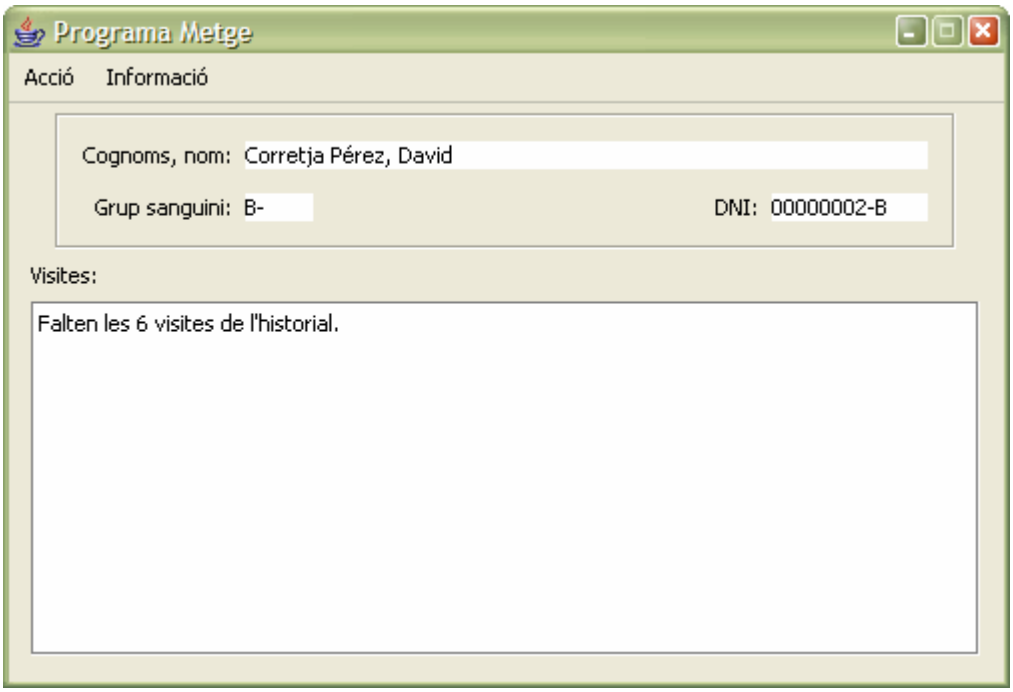

Figura 49. Historial amb totes les visites eliminades

10.2.3 Introduir una visita

Per a la introducció d'una visita és necessari inserir alguna dada pels camps d'observacions i recepta. Una vegada decidit a guardar allò escrit, el metge pot accionar l'emmagatzematge prement el corresponent botó.

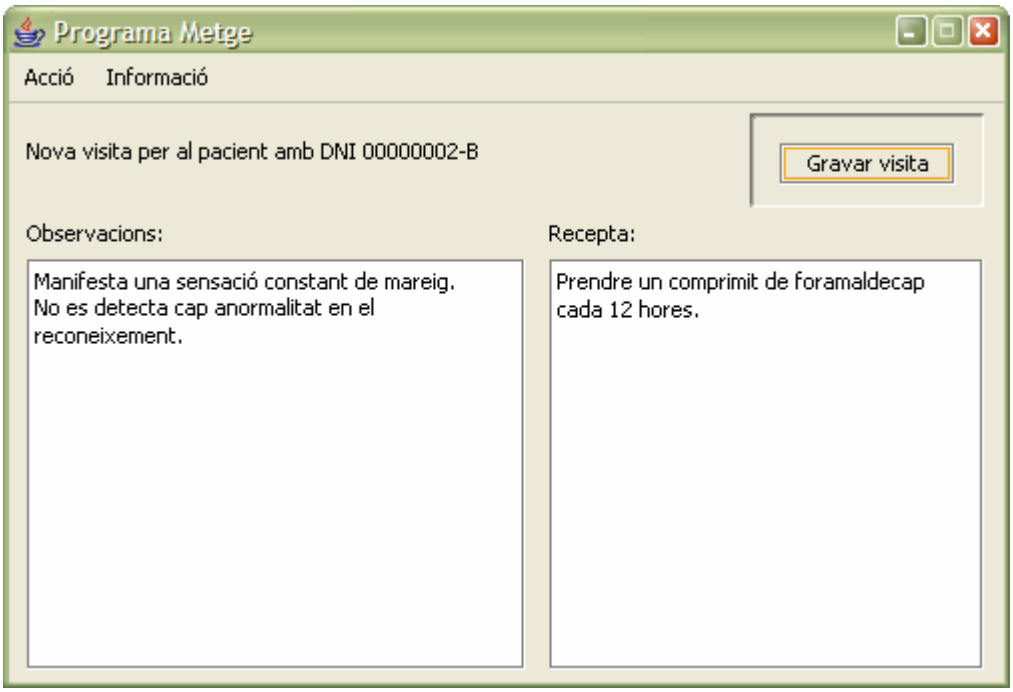

Figura 50. Introducció de les dades per a guardar una visita

Si l'acció s'assoleix sense problemes, es mostra el següent missatge informant del fet.

Figura 51. Missatge informant de la gravació de la visita

Si s'intenta consultar l'historial al qual s'acaba d'introduir la visita, es veu com el sistema funciona de forma correcta ja que segueix mantenint el compte de quines visites falten, tot i que anteriorment s'havien eliminat totes. Amb això, torna presentar-se el missatge explicant que falten les visites de la 1 a la 6, i finalment es mostra la nova visita inserida número 7.

Figura 52. Missatge informant de la falta de les primeres visites

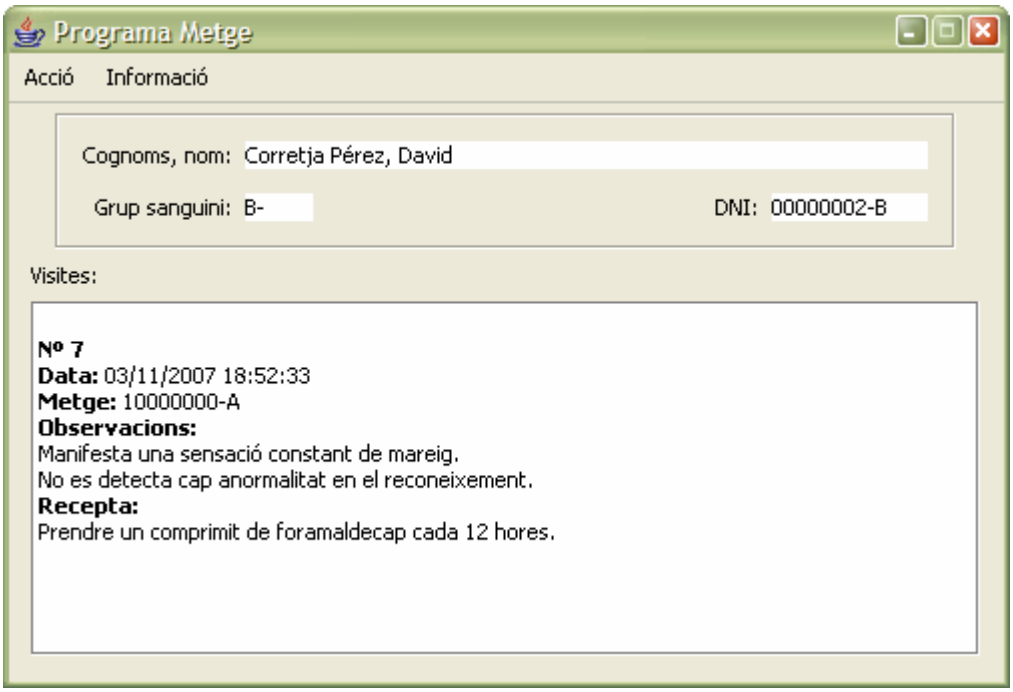

Figura 53. Historial amb la darrera visita introduïda

Provant alguna situació d'error, es força a que el metge triï un pacient que no resta al seu càrrec i intenti afegir-li una visita. El gestor controla correctament la situació i genera l'error.

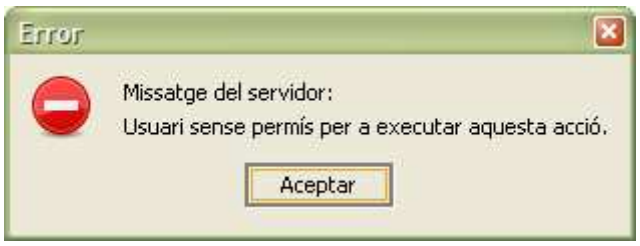

Figura 54. Missatge informant de la falta d'autoritat per a executar una acció

Capítol 11
Epíleg

11.1 Conclusions

Una vegada finalitzat el treball, pot dir-se que s'han assolit les fites marcades inicialment per al sistema d'historials mèdics plantejat, aconseguint els resultats i productes previstos durant la definició del projecte.

Satisfent l'objectiu principal d'aconseguir un sistema segur, s'han creat certificats digitals que identifiquen als usuaris i s'han establert diverses característiques i dissenyat diferents protocols criptogràfics, garantint en conjunt les bases de la seguretat de la informació descrites en el primer capítol:

- **Confidencialitat:** S'han dividit les dades dels historials en dos grups segons el nivell de privadesa i s'han establert diferents graus d'autoritat segons les circumstàncies i el tipus d'usuari que intenta tractar la informació:
    - Un pacient pot accedir a totes les seves dades així com el metge que l'atén.
    - Un metge que no resta al càrrec d'un pacient només li podrà consultar la seva informació més pública.
    - Amb qualsevol altra classe d'accés, variant la situació o l'actor que l'executa, simplement es denega obtenir cap dada rellevant del sistema.

  A més, la informació considerada com a més confidencial es guarda de forma xifrada, cosa que impedeix que sigui llegida en cas de que algú aconsegueixi esquivar els protocols definits accedint directament a la base de dades.

- **No repudi:** Aquesta característica s'ha considerat a l'hora de que un metge faci una inserció amb les dades d'una visita. Per tal de que aquest usuari no pugui repudiar allò

que ell ha escrit, s'emmagatzema la seva signatura digital de les dades originades creant un vincle inequívoc entre el metge i la visita.

- **Autenticació:** Els usuaris que accedeixen al sistema passen per un procés d'autenticació que, una vegada superat, crea una sessió que els permet identificar-se de forma certa per a les posteriors accions que realitzin.

  Addicionalment, les dades emmagatzemades a la base de dades de caràcter més crític són mantingudes de forma conjunta amb la signatura digital del gestor o el metge, fet que controla si les dades s'han modificat per algú sense autorització i, per tant, si aquesta informació és o no autèntica.

- **Integritat:** Es té en compte aquesta propietat per a les dades de més valor en quant a que s'emmagatzemen xifrades, situació que garanteix la seva integritat al no permetre la modificació sense permís. A banda del gestor, l'únic usuari al que se li dóna accés per a generar directament dades amb l'objectiu de ser guardades és el metge que introdueix una visita a l'historial d'un dels seus pacients.

Un dels altres objectius que es plantejaven era la necessitat d'obtenir un funcionament remot del sistema, la qual cosa s'ha aconseguit fent una divisió entre una part client i una altra part servidor. El gestor, connectat a Internet i oferidor dels diversos serveis, pot ser accedit sense problemes per qualsevol usuari amb accés a la xarxa que disposi de l'aplicació client.

Moltes de les dades fetes servir en el programari creat, tant d'informació útil per a l'usuari com de necessària per al funcionament en sí del sistema, cal mantenir-les de forma persistent, i això és el que s'ha acomplert al dissenyar i crear una base de dades que possibilita la conservació de tota aquesta informació.

Una darrera fita era l'obtenció d'interfícies que permetessin utilitzar les aplicacions destinades als usuaris d'una manera senzilla, situació que s'ha assolit per a cadascun dels programes existents. Metges i pacients disposen de les seves respectives aplicacions amb les que interactuar i executar de forma fàcil les accions que els hi són permeses.

Comentar també que durant el disseny i la implementació del sistema s'ha estat treballant considerant la circumstància d'una ampliació realitzada en un futur, de manera que, dins del possible, no s'ha creat un sistema tancat sinó que allà on es podia s'ha intentat facilitar la realització de posteriors modificacions i afegiments.

11.2 Opinió personal

Per a realitzar aquest projecte ha estat necessari aplicar i utilitzar una àmplia varietat dels coneixements, eines i tecnologies apresos durant els transcurs de la carrera, circumstància que ha provocat que s'adquireixi una consolidació i major pràctica d'allò estudiat.

També s'ha hagut de fer front a diversos aspectes els quals mai, o pràcticament mai, s'havien fet servir de forma prèvia. Un d'aquests camps, de fet, és el pilar del treball realitzat: la seguretat. Tan sols es tenien unes idees molt vagues sobre criptografia, però, al plantejar l'elecció del camp del projecte, aquest es va considerar un tema prou interessant com per a estudiar-lo durant la seva realització.

A més dels conceptes de seguretat i les corresponents eines relacionades –OpenSSL i IAIK– també hi ha altres parts treballades per a les que no es tenien coneixements anteriors. Una d'aquestes ha estat, per exemple, l'ús de la llibreria JDOM per al maneig dels documents XML, o per dir un altre cas, la utilització de les llibreries Swing fetes servir per a les interfícies gràfiques.

En definitiva, es considera que amb el disseny i implementació del projecte s'ha aconseguit un assentament i aprofundiment dels camps tractats en els estudis de l'enginyeria així com d'altres treballats per a l'ocasió, motiu pel qual es té una valoració positiva del conjunt.

Annexos

Annex A
Preparació de l'entorn de treball

El projecte es desenvolupa i executa sobre un sistema Windows amb l'ajuda de les eines descrites a continuació. Totes elles, però, compten amb versió per a diverses plataformes, amb la qual cosa no hi hauria d'haver grans problemes en cas d'estimar-se o requerir-se el treball en un altre sistema operatiu d'ús comú. També convé remarcar, de la mateixa manera que s'indica l'opcionalitat del SO, que el següent programari no és vital per a fer un projecte com el present, i que amb diferents alternatives es poden aconseguir resultats igualment funcionals.

A.1 JDK

Java és el llenguatge de programació amb el que s'implementa el projecte. Per al desenvolupament s'han fet servir les llibreries proporcionades per Sun en el seu SDK, en concret la versió 1.5.0 del Java SDK Standard Edition. Aquest programari pot descarregar-se lliurement des de la web de Sun [http://java.sun.com/javase/downloads/index.jsp].

Seguint les instruccions del programa d'instal·lació el procés acabaria sense cap inconvenient, vigilant si de cas, per si no s'ha fet de forma automatitzada, que les variables globals del sistema `PATH` i `CLASSPATH` apuntin respectivament al directori dels executables i al de les classes incloses en el JDK.

A.2 IAIK

IAIK són una sèrie de llibreries enfocades a proporcionar mètodes i funcionalitats útils per a la criptografia. Aquestes APIs, gratuïtes sempre que no es facin servir amb finalitats comercials, poden ser descarregades des de la seva web [https://jce.iaik.tugraz.at/crm/main.php] una vegada registrats en ella. La versió usada és la 3.16 de l'IAIK/JCE Standard Edition.

En aquí, per tal de no tenir inconvenient en utilitzar forts nivells criptogràfics, cal destacar la necessitat de substituir dos paquets dedicats a la seguretat inclosos en el conjunt del programari que forma el JDK. Aquestes llibreries consten al Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files, del qual s'ha fet ús de la versió 6.0 disponible per a descarregar des de la web de Sun [http://java.sun.com/javase/downloads/index.jsp].

Per a poder desenvolupar i executar programari usant les llibreries IAIK convé instal·lar el paquet `iaik_jce_full.jar` dins les carpetes `%JAVA_JRE%\lib\ext` i `%JAVA_JDK%\jre\lib\ext` del JDK.

Els paquets de seguretat a substituir reben el nom de `local_policy.jar` i `US_export_policy.jar`, i es poden trobar a les carpetes `%JAVA_JRE%\lib\security` i `%JAVA_JDK%\jre\lib\security` del JDK.

A.3 OpenSSL

OpenSSL és la llibreria feta servir per a la creació de les claus i els certificats utilitzats en el sistema. En la implementació s'ha utilitzat la versió 0.9.8 del Win32 OpenSSL, la qual pot descarregar-se des de la web d'aquest projecte [http://www.openssl.org/source/]. El procediment d'instal·lació és senzill i automatitzat, de manera que no hauria de presentar cap problema.

En el requadre següent es mostra el contingut de l'arxiu de configuració `openssl.cnf` usat.

```
################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes

# Passwords for private keys if not present they will be prompted
➥for
input_password    = medic
output_password   = medic

# This sets a mask for permitted string types. There are several
➥options.
# default: PrintableString, T61String, BMPString.
# pkix : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or
➥UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or
➥UTF8Strings
# so use this option with caution!
string_mask       = nombstr

################################################################
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = ES
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full
➥name)
stateOrProvinceName_default     = Barcelona

localityName                    = Locality Name (eg, city)
localityName_default            = Barcelona

0.organizationName              = Organization Name (eg,
➥company)
0.organizationName_default      = UOC

organizationalUnitName          = Organizational Unit Name
➥(eg, section)
organizationalUnitName_default  = Gestors

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

dnQualifier                     = D.N.I or N.S.S.
#dnQualifier_default            = 00000000-A

#emailAddress                   = Email Address
#emailAddress_max               = 40

################################################################
[ req_attributes ]
challengePassword = A challenge password
```

```
challengePassword_min = 4
challengePassword_max = 20
```

Figura 55. Arxiu de configuració per a OpenSSL

A.4 Eclipse

Eclipse és l'entorn de treball escollit per a facilitar la feina de programació. La versió usada de l'Eclipse SDK és la 3.1.2, de lliure descàrrega des de la web del producte [http://www.eclipse.org/downloads/].

La instal·lació d'aquesta IDE és simple i no comporta cap complicació, però cal tenir en compte que en les opcions del programa se li han d'indicar les rutes de directori del JDK que es farà servir. Si el JDK ja es troba present en el sistema abans d'instal·lar l'Eclipse, aquest ja detecta de forma automàtica els directoris facilitant encara més el procés.

A.5 MySQL

MySQL és el sistema gestor de base de dades triat per a gestionar i emmagatzemar la múltiple informació generada en el sistema. Aquest SGBD és gratuït i pot ser descarregat des de la seva web [http://dev.mysql.com/downloads/mysql/5.0.html]. La versió concreta usada és la 5.0.27.

Per a ser utilitzat des de Java també és necessari instal·lar el MySQL Connector Java, el qual pot ser descarregar des de la web [http://dev.mysql.com/downloads/connector/]. La versió utilitzada ha estat la 3.1.14.

A.6 MySQL Workbench

MySQL Workbench és una eina per a la creació i documentació de bases de dades, permetent dissenyar d'una forma visual i senzilla l'estructura de les taules i les seves relacions, i generant de manera automàtica el codi SQL necessari per a obtenir-la. El programa compta amb dues versions, una de gratuïta i una altra de comercial, aquesta darrera amb funcionalitats extres. La versió usada de MySQL Workbench és la 5.0.9, la qual pot trobar-se a la web del programa [http://dev.mysql.com/downloads/workbench/5.0.html].

A.7 JDOM

JDOM són un conjunt de llibreries gratuïtes pensades per a facilitar el treball a l'hora de fer servir la tecnologia XML. La versió feta servir de JDOM és la 1.1, la qual pot descarregar-se des de la seva web [http://www.jdom.org/dist/binary/].

Per a poder utilitzar-les convé situar-les a la carpeta corresponent del JDK, o modificar el CLASSPATH de manera que pugui ser accedida sense problemes.

A.8 JUDE

JUDE és un programa per al modelatge UML que ajuda a generar els diagrames per al disseny d'un sistema. L'aplicació disposa de versió gratuïta i comercial, amb més o menys funcionalitats. La versió usada del JUDE Community és la versió 5.1, la qual pot descarregar-se mitjançant un senzill registre previ des de la web del programa [http://jude.change-vision.com/jude-web/product/community.html].

A.9 Jigloo

Jigloo és un plugin per a Eclipse capaç de treballar tant amb les llibreries SWT com Swing –les usades en aquest projecte–, pensat per a facilitar el treball per a la realització de les interfícies gràfiques. La seva utilització és gratuïta sempre que no s'utilitzi amb finalitats comercials. La versió utilitzada és la 4.0.3, la qual pot ser descarregada des de la seva web [http://www.cloudgarden.com/jigloo/].

Per a instal·lar-lo cal col·locar els seus arxius dins del directori `plugin` de l'Eclipse, i aquest, a l'iniciar-se, automàticament l'integrarà dins del seu entorn.

Annex B
Desplegament i posada en marxa

En els següents apartats s'indiquen les accions que s'han de dur a terme per a executar les aplicacions. S'assenyala que per a tal fita és necessari disposar de les llibreries fetes servir al llarg del desenvolupament del projecte, com poden ser les d'IAIK per a la criptografia. A l'annex A ja s'indica com instal·lar els principals elements requerits i en aquí no es reitera amb aquest tema.

B.1 Arxius adjunts

Acompanyant a aquesta memòria s'inclouen tot un conjunt d'arxius generats per al projecte, els quals s'estructuren de la següent manera:

Directori	Descripció
`/bin/gestor`	Arxius binaris amb l'aplicació del gestor.
`/bin/metge`	Arxius binaris amb l'aplicació del metge.
`/bin/pacient`	Arxius binaris amb l'aplicació del pacient.
`/doc`	Aquesta memòria amb la documentació del projecte.
`/pki`	Certificats i PKCS#12 utilitzats en el sistema.
`/project`	Arxiu .zip amb el projecte de l'IDE Eclipse.
`/scr`	Scripts amb el codi per a la creació de la base de dades i els seus usuaris.
`/scr/gestor`	Arxius de codi font corresponents a l'aplicació del gestor.
`/scr/metge`	Arxius de codi font corresponents a l'aplicació del metge.
`/scr/pacient`	Arxius de codi font corresponents a l'aplicació del pacient.

Taula 12. Estructura dels arxius del projecte

B.2 Creació de la base de dades

Comptant amb que el SGBD ja es trobi instal·lat i posat en marxa, pot crear-se la base de dades utilitzada pel sistema executant l'script `/src/BD.sql`, el contingut del qual es mostra tot seguit.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE DATABASE IF NOT EXISTS `medic` ;
USE `medic`;

-- -----------------------------------------------------
-- Table `medic`.`Usuari`
```

```
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Usuari` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Usuari` (
      `DNI` VARCHAR(10) NOT NULL ,
      PRIMARY KEY (`DNI`) ) ;


-- -----------------------------------------------------
-- Table `medic`.`Pacient`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Pacient` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Pacient` (
      `Usuari_DNI` VARCHAR(10) NOT NULL ,
      `Nom` VARCHAR(20) NOT NULL ,
      `Cognom1` VARCHAR(20) NOT NULL ,
      `Cognom2` VARCHAR(20) NULL ,
      `Metge_DNI` VARCHAR(10) NULL ,
      INDEX fk_Pacient_Usuari_index (`Usuari_DNI` ASC) ,
      PRIMARY KEY (`Usuari_DNI`) ,
      INDEX fk_Pacient_Metge_index (`Metge_DNI` ASC) ,
      CONSTRAINT `fk_Pacient_Usuari` FOREIGN KEY (`Usuari_DNI` )
➥REFERENCES `medic`.`Usuari` (`DNI` ) ON DELETE CASCADE ON UPDATE
➥CASCADE ,
      CONSTRAINT `fk_Pacient_Metge` FOREIGN KEY (`Metge_DNI` )
➥REFERENCES `medic`.`Metge` (`Usuari_DNI` ) ON DELETE SET NULL ON
➥UPDATE CASCADE ) ;


-- -----------------------------------------------------
-- Table `medic`.`Metge`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Metge` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Metge` (
      `Usuari_DNI` VARCHAR(10) NOT NULL ,
      `Nom` VARCHAR(20) NOT NULL ,
      `Cognom1` VARCHAR(20) NOT NULL ,
      `Cognom2` VARCHAR(20) NULL ,
      INDEX fk_Metge_Usuari_index (`Usuari_DNI` ASC) ,
      PRIMARY KEY (`Usuari_DNI`) ,
      CONSTRAINT `fk_Metge_Usuari` FOREIGN KEY (`Usuari_DNI` )
➥REFERENCES `medic`.`Usuari` (`DNI` ) ON DELETE CASCADE ON UPDATE
➥CASCADE ) ;


-- -----------------------------------------------------
-- Table `medic`.`Certificat`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Certificat` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Certificat` (
      `Usuari_DNI` VARCHAR(10) NOT NULL ,
      `Certificat` BLOB NOT NULL ,
      INDEX fk_Certificat_Usuari_index (`Usuari_DNI` ASC) ,
      PRIMARY KEY (`Usuari_DNI`) ,
      CONSTRAINT `fk_Certificat_Usuari` FOREIGN KEY
➥(`Usuari_DNI` ) REFERENCES `medic`.`Usuari` (`DNI` ) ON DELETE
➥CASCADE ON UPDATE CASCADE ) ;
```

```
-- -----------------------------------------------------
-- Table `medic`.`Sessio`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Sessio` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Sessio` (
      `Usuari_DNI` VARCHAR(10) NOT NULL ,
      `Sessio` BLOB NOT NULL ,
      INDEX fk_Sessio_Usuari_index (`Usuari_DNI` ASC) ,
      PRIMARY KEY (`Usuari_DNI`) ,
      CONSTRAINT `fk_Sessio_Usuari` FOREIGN KEY (`Usuari_DNI` )
➥REFERENCES `medic`.`Usuari` (`DNI` ) ON DELETE CASCADE ON UPDATE
➥CASCADE ) ;


-- -----------------------------------------------------
-- Table `medic`.`Historial`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Historial` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Historial` (
      `Pacient_DNI` VARCHAR(10) NOT NULL ,
      `NSerieUltim` INT NOT NULL ,
      `SignaturaNSerieUltim` BLOB NOT NULL ,
      `GrupSanguini` VARCHAR(3) NULL ,
      `SignaturaDades` BLOB NULL ,
      INDEX fk_Historial_Pacient_index (`Pacient_DNI` ASC) ,
      PRIMARY KEY (`Pacient_DNI`) ,
      CONSTRAINT `fk_Historial_Pacient` FOREIGN KEY
➥(`Pacient_DNI` ) REFERENCES `medic`.`Pacient` (`Usuari_DNI` ) ON
➥DELETE CASCADE ON UPDATE CASCADE ) ;


-- -----------------------------------------------------
-- Table `medic`.`Visita`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medic`.`Visita` ;

CREATE  TABLE IF NOT EXISTS `medic`.`Visita` (
      `NSerie` INT NOT NULL ,
      `Historial_DNI` VARCHAR(10) NOT NULL ,
      `Metge_DNI` VARCHAR(10) NOT NULL ,
      `Data` VARCHAR(20) NOT NULL ,
      `Visita` BLOB NOT NULL ,
      `Signatura` BLOB NOT NULL ,
      INDEX fk_Visita_Historial_index (`Historial_DNI` ASC) ,
      INDEX fk_Visita_Metge_index (`Metge_DNI` ASC) ,
      PRIMARY KEY (`NSerie`, `Historial_DNI`) ,
      CONSTRAINT `fk_Visita_Historial` FOREIGN KEY
➥(`Historial_DNI` ) REFERENCES `medic`.`Historial`
➥(`Pacient_DNI` ) ON DELETE CASCADE ON UPDATE CASCADE ,
      CONSTRAINT `fk_Visita_Metge` FOREIGN KEY (`Metge_DNI` )
➥REFERENCES `medic`.`Metge` (`Usuari_DNI` ) ON DELETE NO ACTION
➥ON UPDATE CASCADE ) ;



SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Figura 56. Script per a la creació de la base de dades

També és necessari crear els usuaris que s'hi poden connectar, per a la qual cosa cal executar l'script guardat a l'arxiu `/src/usuarisBD.sql`. En el quadre mostrat a continuació pot veure's el seu contingut.

```
-- ---------------------------------
-- Usuari administrador
-- ---------------------------------

GRANT ALL ON medic
TO administrador IDENTIFIED BY 'medic';

-- ---------------------------------
-- Usuari gestor
-- ---------------------------------

GRANT SELECT ON medic.certificat
TO gestor IDENTIFIED BY 'medic';

GRANT SELECT, UPDATE ON medic.historial
TO gestor;

GRANT SELECT ON medic.metge
TO gestor;

GRANT SELECT ON medic.pacient
TO gestor;

GRANT SELECT, INSERT, UPDATE, DELETE ON medic.sessio
TO gestor;

GRANT SELECT ON medic.usuari
TO gestor;

GRANT SELECT, INSERT ON medic.visita
TO gestor;
```

Figura 57. Script per a la creació dels usuaris de la base de dades

Per finalitzar, una vegada creada la base de dades i els comptes d'usuari d'aquells que la poden manipular, convé omplir-la amb informació per a poder-la fer servir. Com s'ha explicat a l'apartat 9.3.2, al paquet extra, situat a la vegada dins del paquet servidor, s'hi troba la classe Insert destinada a aquest propòsit. Remarcar que, per a funcionar, aquesta classe espera trobar els arxius dels certificats al directori `/pki`. Pot ser executada fàcilment a través de l'arxiu `/bin/gestor/omplirBD.bat`.

B.3 Iniciar el servidor

Per a iniciar el servidor tan sols cal cridar la classe `servidor.SGestor` situada a la carpeta `/bin/servidor`. La classe espera rebre com a paràmetre la contrasenya de l'arxiu PKCS#12 del gestor, que com sempre per a totes les contrasenyes, es recorda que és "medic".

Al fitxer de configuració `/gestor/configuracioServidor.xml` s'hi ha de trobar la ruta de la classe stub. Per defecte s'ha indicat la URL `file:/C:/PFC/bin/servidor/`, però hauria d'editar-se en cas de que el sistema s'estigui executant des d'un altre lloc.

Per a tenir un servidor funcional també cal que el servidor de base de dades i el servidor RMI s'estiguin executant, ja que sinó el gestor no podrà operar amb les dades ni tampoc podrà rebre peticions dels usuaris.

Pot utilitzar-se l'arxiu `gestor/iniciar.bat` per a posar en marca el servidor RMI i, a continuació, el gestor.

B.4 Iniciar els clients

S'han creat dues aplicacions diferenciades per a metges i pacients, i respectivament s'ha situat cadascuna als directoris `/bin/metge` i `/bin/pacient`. El punt d'entrada que permet iniciar l'execució es troba a les classes `gui.metge.Inici` i `gui.pacient.Inici`.

Per a poder fer funcionar sense problemes els programes, cal fer accessibles un parell de llibreries encarregades de la interfície gràfica situades a la carpeta `/lib` del directori arrel dels dos usuaris. Simplificant la posada en marxa, s'inclou l'arxiu `/metge/iniciar.bat` i `/pacient/iniciar.bat` que ja obté aquestes llibreries i executa les classes esmentades a l'anterior paràgraf.

Es tindran unes aplicacions funcionals sempre i quan el gestor estigui executant-se, ja que d'una altra manera els clients no podran utilitzar el programa.

Annex C
Bibliografia i referència

IAIK

Especificació de l'API
http://javadoc.iaik.tugraz.at/iaik_jce/current/

Java

Especificació de l'API
http://java.sun.com/j2se/1.5.0/docs/api/

OpenSSL

Documentació per a l'ús del programa
http://www.openssl.org/docs/apps/openssl.html

OpenSSL command-line HOWTO
http://www.madboa.com/geek/openssl/

SQL

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

RMI

Getting started using Java RMI
http://java.sun.com/j2se/1.5.0/docs/guide/rmi/hello/hello-world.html

Swing

Greating a GUI with JFC/Swing
http://java.sun.com/docs/books/tutorial/uiswing/index.html

Jigloo Swing Tutorial
http://www.cloudgarden1.com/swing_tutorial/index.html

XML

XML Tutorial,
http://www.w3schools.com/xml/default.asp

JDOM API Specification
http://www.jdom.org/docs/apidocs/

Easy Java/XML integration with JDOM,
http://www.javaworld.com/javaworld/jw-05-2000/jw-0518-jdom.html

Altres

Wikipedia
http://www.wikipedia.org/

Annex D
Glossari

API
Sigles del terme anglès Application Programming Interface, interfície de programació d'aplicacions. Conjunt de funcions de programació proporcionades per una llibreria o un sistema operatiu i orientades a oferir certs serveis concrets funcionant com a capa d'abstracció.

Autoritat de certificació
Veure CA.

Autoritat de registre
Veure RA.

CA
Sigles del terme anglès Certification Authority, autoritat de certificació. Entitat d'una PKI encarregada d'emetre certificats considerats de confiança per al sistema destinats a usuaris.

Certificat digital
Document contenidor de les dades que identifiquen al propietari del mateix, la seva clau pública i la signatura digital de la CA que valida i autentica tot l'anterior.

Clau privada
Clau criptogràfica utilitzada per a desxifrar i signar documents digitals. Només és disponible per al seu propietari.

Clau pública
Clau criptogràfica utilitzada per a xifrar i verificar la signatura de documents digitals. Resta disponible a tothom.

Infraestructura de clau pública
Veure PKI

JDK
Sigles del terme anglès Java Development Kit, kit de desenvolupament en Java. Part de l'SDK de la companyia Sun dedicat a la creació d'aplicacions en llenguatge Java per a l'execució en la seva màquina virtual.

PKCS
Sigles del terme anglès Public-Key Cryptography Standards, estàndards de criptografia de clau pública. Conjunt d'especificacions destinades a delimitar un marc global per a la informació utilitzada en la criptografia de clau pública.

PKI
Sigles del terme anglès Public Key Infrastructure, infraestructura de clau pública. Conjunt d'elements necessaris per a crear i gestionar certificats digitals amb una base de criptografia de clau pública.

RA
Sigles del terme anglès Registration Authority, autoritat de registre. Entitat pertanyent a una PKI encarregada d'aspectes com la verificació de la identitat dels usuaris o la publicació dels certificats.

SDK
Sigles del terme anglès Software Development Kit, kit de desenvolupament de programari. Conjunt de tecnologies i eines que permeten la creació d'aplicacions per a un sistema o una plataforma concreta.

Signatura digital
Resultat d'una tècnica criptogràfica que pretén aconseguir l'equivalent dins del món virtual a la signatura comuna escrita i les seves propietats de seguretat. Per tal de signar un missatge és necessària la clau privada de qui signa, i qui rep aquest document pot fer ús de la clau pública del signatari per comprovar l'autenticitat del missatge.

TTP
Sigles del terme anglès Trusted Third Party, tercera part de confiança. Entitat en la que confien dues parts facilitant la interacció entre aquestes i donant seguretat a la seva transacció. Dins una PKI, una CA és una TTP.