Esquema criptogràfic per exàmens electrònics segurs

Enginyeria Informàtica

Autor: Xavier Fustero Benavent

Consultor: Jordi Castellà Roca

22 de juny de 2004

La clau de la immortalitat
és viure una vida digna de ser recordada.

Bruce Lee.

1 Dedicatòria

La finalització d'aquest projecte dóna fi a tres anys d'aprenentatge a la UOC. Personalment ha estat una experiència molt enriquidora que m'ha permès apendre les noves tecnologies en una ciència de continu estudi com és la informàtica. Això ha estat possible gràcies a aquest tipus d'univertitats no presencials on pots combinar la vida professional amb la vida estudiantil.

Penso que tot l'esforç, paciència i dedicació per arribar al final d'aquesta enginyeria no ha estat només meva, sinó que hi ha molta més gent que ho ha fet possible.

Primer m'agradaria felicitar als coordinadors d'aquests estudis. Especialment felicito al *Jordi Castellà*, el meu tutor del projecte, per la seva dedicació i ajuda durant tot el desenvolupament d'aquest. També vull agraïr molt especialment l'ajuda de dos grans "gurús"del *L*atex, *Enrique Blanco* i *Josep Francesc Abril*, en l'elaboració d'aquesta memòria.

De forma especial, vull dedicar aquest projecte a tota la familia. Les coses importants per mi, ho són per ells.

Ja per últim, em queda agraïr-li moltíssim la paciència i suport moral d'aquests darrers tres anys a la meva parella. Gràcies *Renata*!

Índex

2 Descripció del projecte

L'objectiu d'aquest projecte és la implementació d'un esquema criptogràfic que permeti la realització d'exàmens electrònics de forma segura. Un examen electrònic es defineix com un examen on el seu enunciat i la seva resposta estan en format electrònic. Segons si la ubicació des d'on l'examen és respost per l'alumne està controlat o no ho està, la problemàtica per assegurar-ne la seva seguretat i evitar la còpia de les respostes, és més complexa. En aquest projecte ens centrarem en el cas més senzill, on l'alumne respon l'examen des d'un entorn controlat, p.e. una aula d'un centre de suport de la UOC. Un examen té un cicle de vida, és a dir, comença en el moment que el professor el redacta, pot passar per un procés de revisió i acaba quan es publica la nota final. Hi ha un seguit d'etapes. L'esquema presentat en el projecte garanteix mitjançant protocols i procediments, cadascuna de les etapes que es poden trobar en aquest cicle de vida. Els professors i els alumnes són els clients d'aquest esquema. Per cadascun d'aquests usuaris el projecte ha implementat un aplicatiu: "Aplicatiu Alumneï "Aplicatiu Professor". A més a més per guardar les dades que generen els alumnes (respostes als exàmens) i els professors (exàmens, correccions) s'ha implementat un sistema central anomenat "Gestor d'Exàmens"que controla el cicle de vida de l'examen. L'aplicatiu professor i l'aplicatiu alumne no interaccionen directament sinó que ho fan mitjançant el "Gestor d'Exàmens".

3 Nocions bàsiques de criptografia

En aquesta secció es descriuen nocions bàsiques de criptografia per situar al lector en el conjunt de protocols i processos criptogràfics que fem servir en la criptografia de clau pública o asimètrica.

En l'esquema de clau pública, cada usuari té una parella de claus: S_K *(secret key)* i la P_K *(public key)*. Per tal de vincular la identitat de l'usuari amb la clau pública emprem un certificat. Aquest certificat és un fitxer que conté la clau pública, l'identitat del seu propietari més la signatura d'una entitat de confiança que anomenem Autoritat de Certificació (CA). Les dades d'aquest fitxer segueixen l'estàndard X509.

Quant xifrem les dades, es crea un sobre digital. Aquest té els avantatges de la rapidesa dels criptosistemes simètrics (DES, 3DES, AES,etc.) i la facilitat de la gestió de claus dels criptosistemes asimètrics (RSA, ElGamal,etc.). Primer es crea una clau de sessió K d'un criptosistema simètric i es xifren les dades amb aquesta clau. A continuació, es xifra la clau K amb la clau pública del receptor (el "Gestor d'Exàmens"en aquest cas). A continuació, es concatena el text xifrat amb la clau K xifrada i s'envia. Com la clau K ha estat encriptada amb la clau pública del receptor, aquest és l'únic que pot obrir el sobre digital. Aquest últim desencripta K amb la seva clau pública i obté el missatge.

Els sistemes de xifrat amb claus asimètriques fan servir l'algorisme **RSA** que va ser publicat al 1978. El nom de l'algorisme és un acrònim construït amb els cognoms dels qui el van dissenyar: *Ronald L. Rivest, Adi Shamir* i *Leonard Adleman*. L'algorisme **RSA** transforma un enter T en un valor xifrat C fent servir la relació següent:

$$C = (T^E) mod P * Q \tag{1}$$

On P i Q són dos nombres primers grans (de més de 1024-bits si es vol que el xifrat sia segur) i on E és un nombre que és més gran que 1 i menor que $P*Q$. Aquest es tria amb la condició que E i *(P-1)*(Q-1)* siguin primers relatius, és a dir, que no tinguin factors primers comuns. *(P-1)*(Q-1)* producte de dos nombres parells és necessàriament parell i la condició anterior ens diu que E ha de ser senar. La funció que ens retorna T a partir de C ve donada per l'expressió:

$$T = (C^D) mod P * Q \tag{2}$$

On ara D , anomenat l'invers multiplicatiu de E és un nombre que satisfà la condició:

$$DE = 1(mod(P\text{-}1 * Q\text{-}1)) \tag{3}$$

El valor D es troba fàcilment buscant un enter X que verifiqui la condició:

$$X * (P\text{-}1) * (Q\text{-}1) + 1)/E = enter \tag{4}$$

Aquest enter s'agafa com D i així:

$$DE\text{-}1 = X * (P\text{-}1) * (Q\text{-}1) \tag{5}$$

és a dir,

$$DE = 1(mod(P\text{-}1) * (Q\text{-}1)) \tag{6}$$

Per a poder xifrar el missatge el que cal conèixer és E i $P*Q$, informació que és precisament la clau pública. Per a desxifrar el missatge el que cal conèixer és D i $P*Q$ que és la clau secreta. La seguretat del procediment reposa en que no existeix cap procediment conegut simple que permeti calcular D, P o Q si sols es coneix E i $P*Q$. Si P i Q són ambdós de 1024 bits, el temps de càlcul necessari amb els ordenadors actual és de l'ordre dels milers de milions d'anys.

El procediment que hem indicat permet assegurar que un missatge sols el pugui desxifrar el propietari de la clau pública que s'ha emprat per a xifrar-lo, cosa que ens resol el problema de la seguretat de la informació. Tanmateix en el problema de bescanvi de missatges queda encara pendent un problema: com pot assegurar-se el receptor d'un missatge xifrat de la identitat de l'emissor del missatge i de que el missatge original no ha estat modificat? Per a resoldre aquest segon problema es fa servir la tècnica de la signatura digital.

Una signatura digital és un senyal que sols pot donar una persona determinada, com ho és la signatura tradicional i que per tant la identifica de forma única i que al mateix temps garanteix la integritat del missatge. La signatura digital està regulada pel *Digital Signature Standard, DSS* que defineix l'algorisme de Signatura Digital (*DSA*). El procediment de construcció de la signatura digital d'un document és el següent:

1. L'emissor d'un missatge sotmet el missatge a un procés de hashing i crea així una clau unívocament associada al missatge (message digest), qualsevol canvi que es fés al missatge alteraria el resultat del procés de hashing, és a dir, el message digest. L'algorisme de hash que es fa servir està definit al *Secure Hash Standard(SHS)*, *FIPS 180 (Federal Information Process Standard).*

2. Xifra el message digest que ha obtingut amb la seva clau privada. Com que aquesta clau sols ell la coneix, aquest xifrat no és reproduïble per un tercer. Això és la seva signatura d'aquest missatge. Ara ja té els elements necessaris per a poder trametre el missatge signat. Afegeix la signatura al missatge, ho xifra tot amb la clau pública del destinatari i ho tramet.

3. Quan el destinatari rep el missatge, el desxifra amb la seva clau privada, desxifra la signatura amb la clau pública del remitent, sotmet el missatge al procés de hashing i compara el message digest que obté amb el que ha rebut. Si són iguals, té la garantia que el missatge no ha estat alterat i, puig que per a desxifrar la signatura ha fet servir la clau pública del remitent, té en principi la garantia de que el missatge ha estat emès pel remitent. Per assegurar que la clau pública del remitent pertany efectivament al remitent i no a algú altre que es vol fer passar per ell, així com que la dita clau no ha quedat fora d'ús pel motiu que sia, es fan servir les autoritats de certificació.

Una autoritat de certificació és un sistema independent en el que tothom confia que posa la seva signatura digital a les claus públiques dels usuaris a fi d'assegurar que una determinada clau pública pertany a un remitent determinat. Tots els usuaris de la xarxa fan confiança en els certificats emesos per l'Autoritat de Certificació.

4 Introducció

4.1 Justificació del PFC i context en el qual es desenvolupa: punt de partida i aportació del PFC

Les xarxes de comunicacions d'abast global, com p.e. Internet, han permès que moltes activitats que es realitzaven de forma presencial es puguin realitzar de forma remota. El comerç electrònic podria ser l'exemple més significatiu, i un altre exemple és l'objectiu d'aquest projecte, la realització d'exàmens electrònics. Els professors poden redactar els enunciats dels exàmens des de casa seva, i guardar-los de forma segura en un sistema central que anomenem "Gestor dÉxàmens". Per la seva part, els alumnes poden descarregar-se els exàmens del sistema central i respondre'l. Cal matitzar que en aquesta primera fase els alumnes han de respondre els exàmens des d'un centre controlat. Quedaria per més endavant la possibilitat que els alumnes puguessin respondre els exàmens des de casa.

Les avantatges de la realització d'exàmens electrònics no es redueixen únicament a la independència espaial, ni temporal que s'ha esmentat amb els exemples anteriors. Hi ha altres avantatges com l'estalvi econòmic que es deriva de la impressió en paper dels enunciats. Un sistema electrònic permet realitzar aquestes tasques d'una forma més econòmica. En el cas de la UOC un altre punt important és la simplificació de la logística necessària en els períodes d'exàmens. Pels exàmens la UOC s'han d'habilitar diferents punts repartits per la geografia on hi ha els seus alumnes de manera que aquests puguin realitzar els exàmens. Això suposa que l'alumne ha d'indicar a quina seu realitzarà l'examen, i la UOC ha de fer arribar l'examen a aquesta seu. Un cop ha finalitzat l'examen, s'han de fer arribar les respostes als consultors per corregir-les. Actualment les respostes son escanejades i enviades en format digital als consultors. Estem doncs a mitges entre un examen convencional i un examen electrònic.

La realització de tot el cicle complert d'un examen electrònic simplificaria notablement la logística necessària, i suposaria un estalvi important.

Malgrat totes aquestes avantatges, també hi ha un seguit d'inconvenients o problemes que cal resoldre. Un d'aquests és la seguretat de tot el sistema. Els exàmens són un punt molt sensible en el sistema acadèmic. Cal garantir que cap de les parts pugui actuar de forma deshonesta sense que es detecti. L'objectiu d'aquest projecte és garantir un nivell de seguretat similar al que podem trobar en un examen convencional.

4.2 Objectius del PFC

L'objectiu d'aquest projecte és la implementació d'un protocol criptogràfic per la realització d'exàmens electrònics segurs a nivell d'aplicació. Per garantir el nivell de seguretat necessari s'han dissenyat un conjunt de protocols i processos criptogràfics per cada etapa del cicle de vida d'un examen. Per establir aquestes fites, observem què passa amb els exàmens presencials.
En un examen presencial, un alumne arriba a l' aula assignada i un o diversos professors entreguen els exàmens. Aquests porten algun segell acreditatiu de la universitat. Durant la realització d'aquests, els professors comproven la identitat dels estudiants demanant el carnet d'identitat i vigilen que no es copíin les respostes. Un cop acabat el temps de l'examen, entreguen les respostes als professors.
Totes aquestes pautes i algunes més són les que s'han d'assolir en aquest projecte. A continuació es detallen aquestes propietats de seguretat que garanteix l'esquema criptogràfic implementat.

Autenticitat

- Els alumnes han d'estar segurs de que els professors han realitzat els enunciats i les correccions dels exàmens
- Els professors han d'estar segurs de que els exàmens a corregir han estat fets pels alumnes.

Confidencialitat

- Els professors no tenen perquè saber qui és l'autor de la resposta
- Les respostes dels alumnes han de ser secretes i només visibles pels professors
- Els enunciats dels professors només poden ser accedits pels estudiants quant realitzin l'examen
- La correcció d'un examen i la seva revisió només pot ser accessible per l'autor.

Integritat de les dades

- Els professors han d'estar segurs de que les respostes dels alumnes no s'han modificat un cop entregades
- Els alumnes han d'estar segurs de que les correccions i/o revisions efectuades pels professors no s'han modificat
- Les dades guardades al "Gestor d'Exàmens"no es poden modificar.

No-repudi

- No és possible que un alumne negui que la resposta és seva, i tampoc que un professor negui que la correcció ha estat feta per ell.

Rebut de lliurament

- El "Gestor d'Exàmensl·lliura un rebut als alumnes conforme han fet un examen determinat.

4.3 Arquitectura i decisions de disseny

4.3.1 Arquitectura dels aplicatius

Com he descrit prèviament els usuaris de l'esquema són els professors i els alumnes. Cadascun d'aquests disposa d'un aplicatiu per poder realitzar els exàmens electrònics. A més a més hi ha un tercer aplicatiu per controlar el cicle de vida de l'examen que s'anomena ""Gestor d'Exàmens"". Per tant, s'han de crear tres aplicatius diferents:

- Aplicatiu Professor
- Aplicatiu Alumne
- Aplicatiu "Gestor d'Exàmens"

Les funcionalitats de cadascun, són les següents:

1. **Aplicatiu Professor**
    - Autenticar-se contra el "Gestor d'Exàmens"
    - Redactar un examen per cadascuna de les assignatures que el professor imparteix
    - Assegurar la validesa de l'examen
    - Enviar l'examen al "Gestor d'Exàmens"
    - Recuperar les respostes de l'examen
    - Verificar la validesa de les respostes
    - Corregir les respostes donant la nota final
    - Assegurar la validesa de les correccions
    - Enviar les correccions al "Gestor d'exàmens"
    - Obtenir les peticions de revisió de les correccions.
2. **Aplicatiu Alumne**
    - Autenticar als alumnes contra el "Gestor d'Exàmens"
    - Recuperar l'examen que guarda el "Gestor d'Exàmens"en una data determinada
    - Verificar la validesa de l'examen recuperat
    - Recollir la resposta de l'alumne
    - Enviar la resposta de l'alumne al "Gestor d'Exàmens"
    - Obtenir el rebut conforme ha lliurat l'examen
    - Accedir a la correcció de la seva resposta i veure 'n la nota final
    - Demanar la revisió d'una correcció d'examen.

- Autenticar els professors i els alumnes
- Rebre i emmagatzemar de forma segura els exàmens
- Lliurar els exàmens als alumnes
- Rebre i emmagatzemar de forma segura les respostes dels alumnes
- Lliurar el rebut de recepció de la resposta de l'examen als alumnes
- LLiurar les respostes dels exàmens als professors
- Rebre i emmagatzemar de forma segura les correccions de les respostes
- Lliurar als alumnes les correccions de les seves respostes
- Rebre la petició de revisió d'un examen.
- Lliurar als professors les peticions de revisió.

4.4 Llibreries criptogràfiques

Per a desenvolupar aquest projecte he escollit la llibreria criptogràfica **IAIK**. La raó per la que he escollit **IAIK** és per la seva facilitat d'ús, àmplia implementació de crides criptogràfiques i bona documentació.

4.4.1 Base de dades

La base de dades emprada és la de Microsoft Access amb connexió ADODB. La raó d'aquesta decisió ha estat el major coneixement d'aquesta BD i perquè és compatible amb Windows 2000 que és la plataforma de l'equip que s'ha emprat per desenvolupar el projecte. La base de dades consta de quatre taules:

- **tb_Enunciats**: és on guardem els enunciats dels exàmens dels professors
- **tb_Respostes**: és on guardem les respostes dels exàmens realitzades pels alumnes
- **tb_Correccions**: és on guardem les correccions que fan els professors dels exàmens
- **tb_Revisions**: és on guardem les revisions dels exàmens sol.licitades pels diferents alumnes

No s'elimina cap registre de les taules. S'ha près la decisió de guardar sempre els històrics i canviar només l'estat en que es troben els registres guardats. Per exemple, un examen guardat amb estat "PENDENT"a la taula **tb_Respostes**, un cop s'hagi corregit se li canviarà l'estat a ÇORREGIT". Crearem un nou registre a la taula **tb_Correccions** amb les dades d'aquest.

4.5 Enfocament i mètode seguit

A continuació descriuré cadascuna de les accions en cadascuna de les funcionalitats descrites anteriorment.

4.5.1 Redacció examen

- Aplicatiu professor:
  1. Crear un identificador únic de l'examen, *Id*, format per les dades següents:
     - *As*:Assignatura
     - *Cd*: Codi Assignatura
     - *Qu*: Quatrimestre
     - *Da*: Data
     - *Ns*: Número de sèrie.
  2. Obtenir l'enunciat de l'examen, E que intodueix el professor per teclat.
  3. Signar amb la clau privada del professor S_P l'identificador de l'examen *Id*, i l'enunciat de l'examen E: $Es = S_P[Id,E]$.
  4. Autenticar al professor davant del "Gestor d'Exàmensütilitzant la seva parella de claus (P_P,S_P).
  5. Lliurar les dades següents al "Gestor d'exàmens":(*Id,E,Es*).
- "Gestor d'Exàmens":
  1. Autenticar al professor.
  2. Verificar les dades que ha enviat el professor, és a dir, la signatura digital *Es*.
  3. Emmagatzemar les dades enviades per l'aplicatiu professor: (*Id,E,Es*).

4.5.2 Resposta de l'examen

Per tal de que l'alumne respongui l'examen es realitzen els passos següents:

- Aplicatiu alumne:
  1. Autentica a l'alumne contra el "Gestor d'Exàmens"amb la seva parella de claus (P_A,S_A).
  2. Obtenir l'examen emmagatzemat al "Gestor d'Exàmens": (*Id,E,Es*).
  3. Verificar la signatura digital del professor de l'examen: *Es*
  4. Recollir la resposta R, que l'alumne entra pel seu teclat.
  5. Obtenir de forma aleatòria un identificador de resposta *Ie*
  6. Signar l'identificador de resposta *Ie*, la resposta R i la signatura de l'examen *Es* amb la clau privada de l'alumne S_A: $Rs = S_A[Es,Ie,R]$
  7. Enviar de forma segura *Ie,* R i *Rs* al "Gestor d'exàmens".
- "Gestor d'Exàmens": un cop es disposa de la resposta, es realitzen els passos següents:
  1. Verificar la signatura de la resposta *Rs* amb la clau pública de l'alumne P_A.
  2. Crear la capçalera de lliurament T amb els camps (Id,Ie,Tm). Tm és l'instant actual en que es realitza aquesta operació segons el rellotge del "Gestor d'Exàmens".
  3. Generar un rebut de lliurament *Ts* de la resposta R per part de l'alumne amb la clau privada del "Gestor d'Exàmens": $Ts=S_G[Es,R,T]$.

4. Enviar T i Ts a l'alumne.
5. Obtenir de forma aleatòria una clau de sessió K
6. Fer el sobre digital. Xifrar la signatura de la resposta Rs amb la clau K del pas anterior. Xifrar la clau K amb la clau pública del "Gestor d'exàmens" P_G: $X=S_G[X,Es,R]$.
7. Guardar de forma segura: (Id,E,Es,R,T,Ts,X,Xs).

- Aplicatiu Alumne: es verifiquen que les dades del rebut són correctes.

1. Verificar les dades del rebut Ts i la signatura del "Gestor d'Exàmens".
2. Guardar (T,Es,R,Ts) per demostrar que l'alumne ha realitzat l'examen.

4.5.3 Correcció de l'examen

Amb l'ajuda de l'aplicatiu professor, el professor corretgeig l'examen realitzant els passos següents:

- Aplicatiu professor:

1. Autenticar el professor contra el "Gestor d'Exàmens"amb la parella de claus del professor (P_P,S_P).
2. Obtenir les dades de l'examen i la resposta d'aquest: (Id,E,Es,R,X,Xs).
3. Verificar les signatures digitals següents: Es i Xs
4. Corretgir la resposta R i posar una nota N.
5. Signar la resposta de l'examen amb la clau privada S_P del parell de claus del professor (P_P,S_P): $Ns=S_P[Es,Xs,N]$.
6. Enviar N i Ns al "Gestor d'Exàmens".

- "Gestor d'Exàmens"

1. Verificar la signatura Ns.
2. Desxifrar X, obtenint Rs.
3. Emmagatzemar de forma segura (Id,E,Es,R,Rs,T,Ts,N,Ns).

4.5.4 Obtenció del resultat de l'examen

Amb l'ajuda de l'aplicatiu alumne, l'alumne realitza els passos següents per accedir a la seva nota final.

- Aplicatiu alumne

1. Autenticar l'alumne contra el "Gestor d'Exàmens"amb la seva parella de claus (S_A,P_A).
2. Obtenir la correcció de l'examen corresponent a l'identificador Ie, enviant el rebut Ts.
3. Verificar la signatura Ns.
4. Mostrar la nota N a l'alumne.

4.5.5 Revisió de l'examen

Amb l'ajuda de l'aplicatiu alumne, l'alumne realitza els passos següents per demanar una revisió del seu examen.

- Aplicatiu alumne

  1. Autenticar a l'alumne davant el "Gestor d'Exàmens"amb la seva parella de claus (S_A,P_A).
  2. Obtenir de forma aleatòria un identificador de revisió Ir
  3. Signar l'identificador de revisió Ir juntament amb les dades de l'examen amb la clau privada de l'alumne S_A. Rv=$S_A[Ir, Ts]$.
  4. Enviar Rv i Ir al "Gestor d'Exàmens".

  Amb l'ajuda de l'aplicatiu professor, el professor realitza els passos següents per veure les peticions de revisió d'un examen.

- Aplicatiu professor

  1. Autenticar al professor davant del "Gestor d'Exàmens"amb la parella de claus del professor (S_P,P_P).
  2. Obtenir les peticions de revisió de l'examen a partir de l'identificador d'examen *Id*.

4.6 Planificació del PFC

Taula resum de la planificació de les tasques

Nom tasca	Duració	Inici	Fi
PROJECTE CRIPTOGRAFIA	63 dies	25/03/2004	21/06/2004
Inici Projecte	1 dia	25/03/2004	25/03/2004
Requeriments	2 dies	25/03/2004	26/03/2004
Especificació	2 dies	29/03/2004	30/03/2004
Especificació dels tres models (A,P,G)	2 dies	29/03/2004	30/03/2004
Disseny	26 dies	31/03/2004	05/05/2004
Disseny dels tres models	20 dies	31/03/2004	27/04/2004
Documentació dels tres models	6 dies	28/04/2004	05/05/2004
Dissenys acabats	0 dies	05/05/2004	05/05/2004
Construcció models	26 dies	31/03/2004	05/05/2004
Sistemes criptogràfics	20 dies	06/05/2004	02/06/2004
Cerca de sistemes existents	1 dia	06/05/2004	06/05/2004
Creació claus i certificats	1 dia	07/05/2004	07/05/2004
Construcció protocol criptogràfic	20 dies	06/05/2004	02/06/2004
Comunicació	7 dies	03/06/2004	11/06/2004
Requeriments , anàlisi comunicació	1 dia	03/06/2004	03/06/2004
Estudi model XML	2 dies	04/06/2004	07/06/2004
Estudi tecnologia RMI	2 dies	08/06/2004	09/06/2004
Disseny del model RMI	2 dies	10/06/2004	11/06/2004
Jocs de proves	0 dies	11/06/2004	11/06/2004
Documentació	6 dies	14/06/2004	21/06/2004
Projecte acabat	0 dies	21/06/2004	21/06/2004

Taula 1: *Planificació projecte*

Aquesta taula és la planificació final que s'ha seguit. Originalment s'hi van afegir més tasques que s'han hagut d'eliminar per falta de temps. Però es detalla el temps dedicat al seu estudi i disseny. Aquest és el cas del model XML i el protocol RMI. Aquests s'han arribat a implementar però no s'ha aconseguit que funcionessin en el conjunt de l'apliació. Per aquest motiu no es reflecteixen en la taula.

4.7 Productes obtinguts i intal.lació

Al final d'aquest projecte he obtingut dues aplicacions diferents corresponent als dos tipus de clients: alumnes i professors. Aquestes corresponen a: *MenuAlumne* i *MenuProfessor*. Aquests dos aplicatius haurien de comunicar-se amb l'aplicatiu "Gestor d'Exàmens" però pel fet que ens falta implementar el protocol de comunicació, ho hem de córrer tot en un sol directori on es trobin totes les classes.
En el fitxer zip que s'entrega, **ffusteropfc.zip**, hi trobem els següents directoris:

- **aplicatius**: conté totes les classes que necessitem perquè funcionin els aplicatius clients: *MenuAlumne* i *MenuProfessor*
- **bd**: conté la base de dades *ExamensVirtuals.mdb*
- **data**: conté els subdirectoris *alumne, professor i ge* on es troben els certificats digitals emprats per cadascun d'ells
- **doc**: conté la memòria del projecte en format pdf i ps
- **src**: conté els arxius font de java

Els subdirectoris *alumne, professor i ge* del directori **data**, contenen els certificats que s'han necessitat per simular un alumne, un professor i un "Gestor d'Exàmens".

A continuació s'explica com s'han generat aquests certificats amb l'ajuda del openssl.

1. S'a creat una CA i tota la seva estructura: *CA.sh -newca*
2. - S'a posat validesa per un any: *openssl x509 -in CA/cacert.pem -days 365 -out noucacert.pem -signkey CA/private/cakey.pem*
   - *mv noucacert.pem CA/cacert.pem*
3. S'a generat un parell de claus pel CA creat: *openssl genrsa -des3 -out RootCA.key 1024*
4. S'a generat un fitxer de "certificate request": *openssl req -new -key RootCA.key -out RootCA.csr*
5. S'a autosignat aquest certificat: *openssl ca -policy policy_anything -out RootCA.crt -infiles RootCA.csr*
6. S'a creat un format pk12 per aquest fitxer: *openssl pkcs12 -export -in RootCA.crt -inkey RootCA.key -certfile demoCA/cacert.pem -out RootCA.p12*
7. S'a generat la resta de claus i les peticions de certificats de l'alumne, professor i ge (p.e): *openssl genrsa -des3 -out alumne.key 1024 openssl req -new -key alumne.key -out alumne.csr*

Instal.lació: aquesta aplicació no té un protocol de comunicació implementat. La màquina des de la que s'executin les proves ha de tenir plataforma *Windows* i s'ha de crear una connexió *ADOBD* amb la base de dades Microsoft Access que s'entrega.
Important: el path de la base de dades està en el codi. Per poder-ho executar copiarem la base de dades al directori: *c:*
CriptoServer

5 Disseny UML de l'aplicació

5.1 Casos d'ús: Alumne

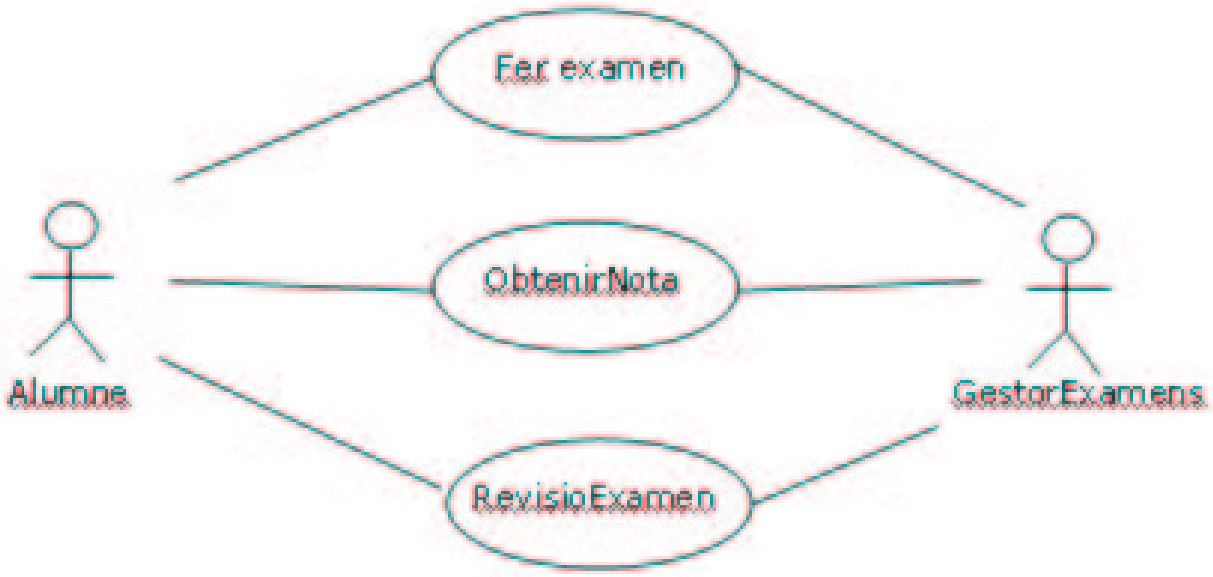

Figura 1: Cas Alumne

L'alumne mitjançant el seu aplicatiu pot realitzar les següents operacions:

- Fer un examen
- Obtenir la nota final
- Sol.licitar la revisió d'un examen.

5.1.1 Cas d'ús: FerExamen

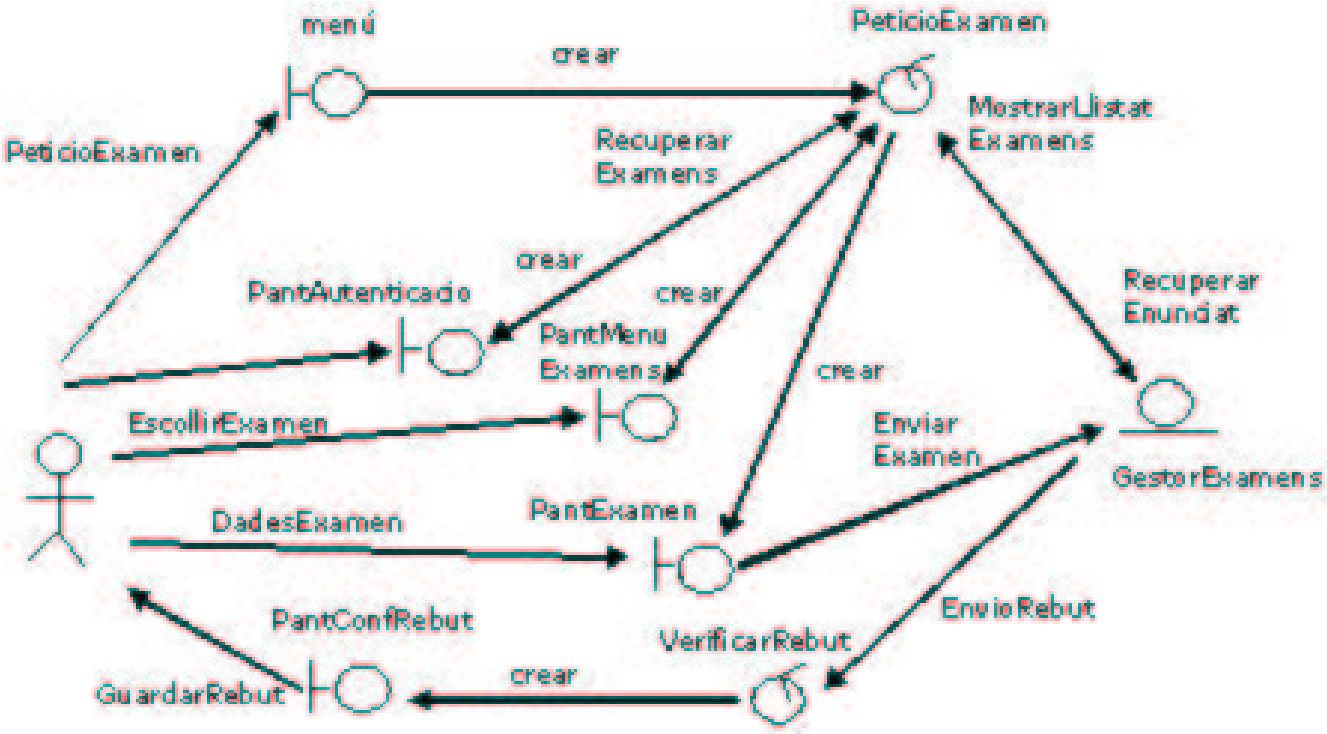

Figura 2: Cas FerExamen

- *Resum de la funcionalitat:* l'alumne s'identifica al "Gestor d'Exàmensï recupera l'examen que vol fer. L'alumne fa l'examen. Un cop acabat, envia l'examen al "Gestor d'Exàmens". Per últim rep el comprovant conforme ha enviat l'examen correctament.

5.1.2 Cas d'ús: ObtenirNota

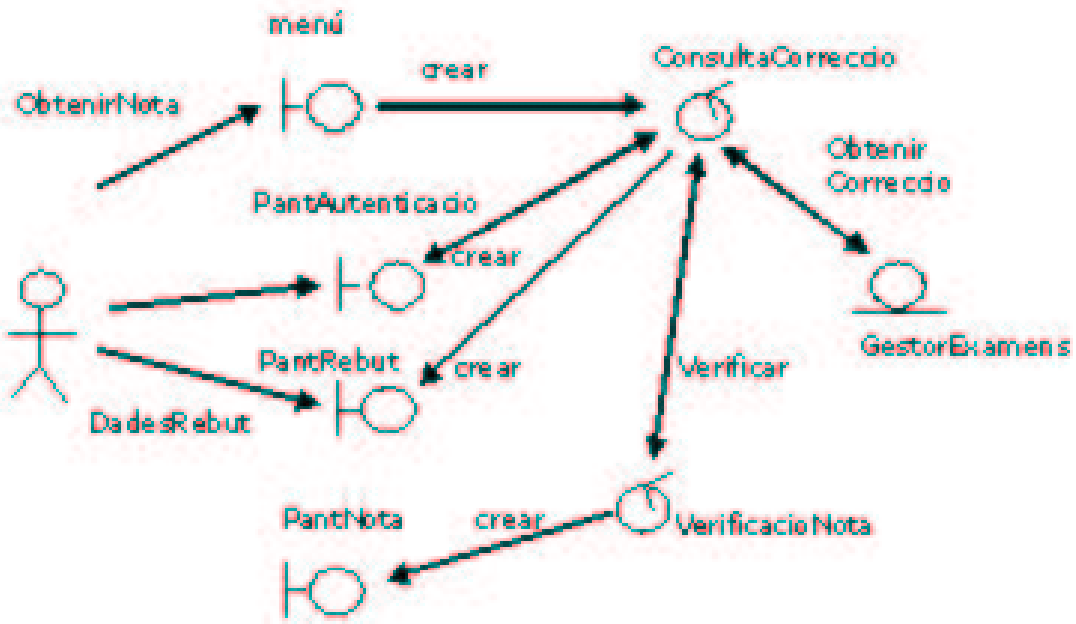

Figura 3: Cas ObtenirNota

- *Resum de la funcionalitat:* l'alumne ha de poder consultar la correcció dels seus exàmens i veure la nota. Per això, s'ha d'identificar al "Gestor d'Exàmens"qui li mostrarà la correcció de l'examen i la nota obtinguda.

5.1.3 Cas d'ús: RevisioExamen

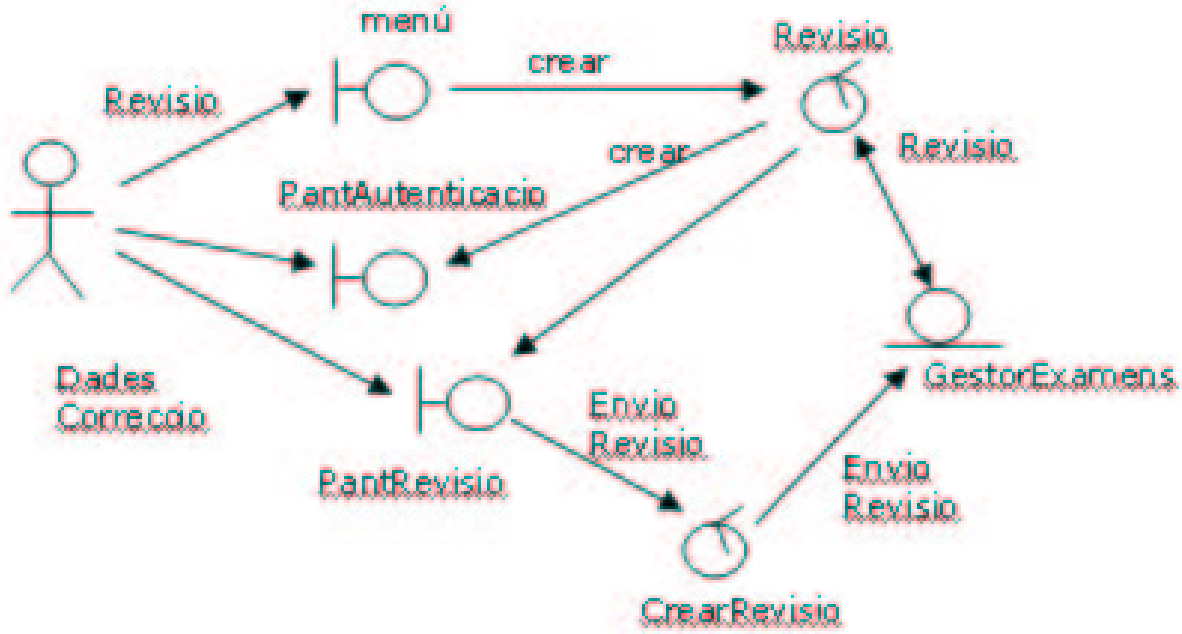

Figura 4: Cas RevisioExamen

- *Resum de la funcionalitat:* l'alumne s'identifica al "Gestor d'Exàmens". Aquest li mostra l'examen i l'alumne formalitza la petició de revisió.

5.2 Casos d'ús: Professor

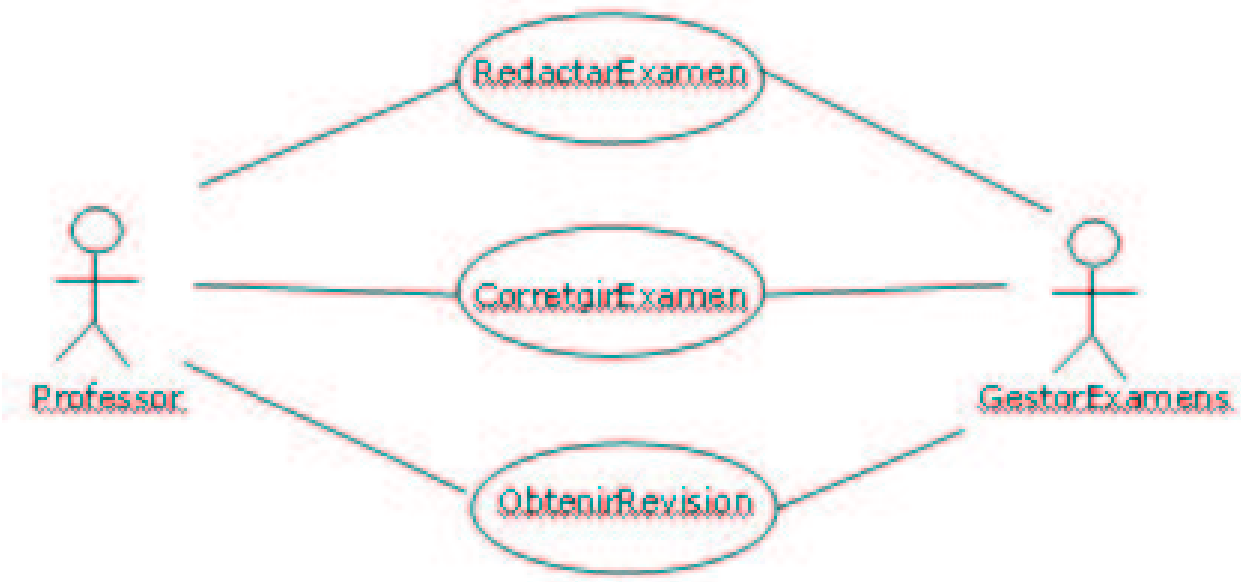

Figura 5: Cas Professor

El professor mitjançant el seu aplicatiu pot realitzar les operacions següents:

- Redactar un nou un examen
- Corregir un examen
- Mirar les revisions que li han demanat.

5.2.1 Cas d'ús: RedactarExamen

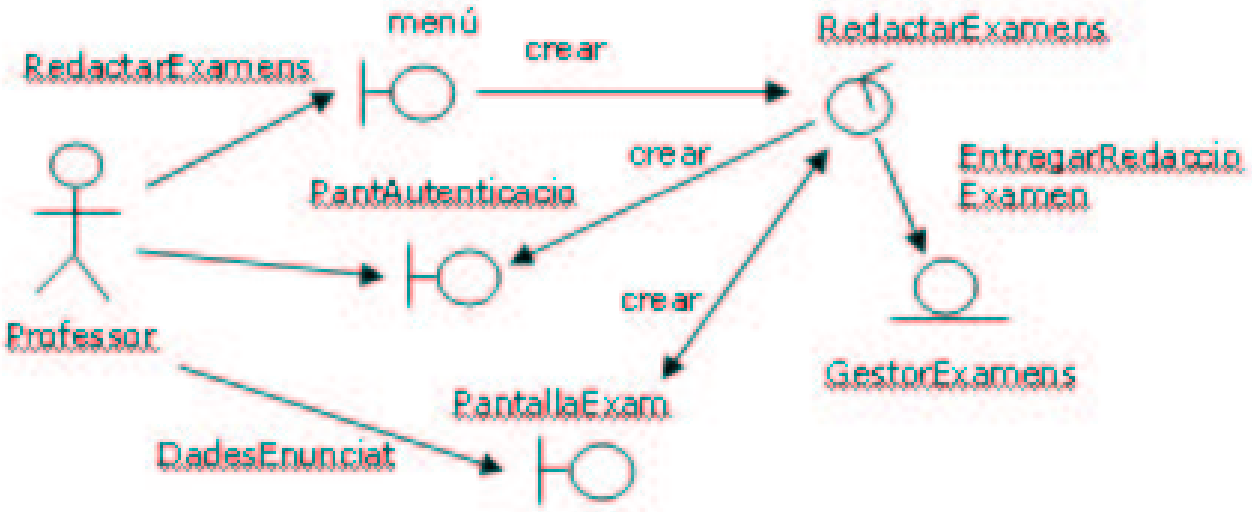

Figura 6: Cas RedactarExamen

- *Resum de la funcionalitat:* el professor s'autentica davant el "Gestor d'Exàmensï li lliura l'enunciat.

5.2.2 Cas d'ús: CorretgirExamen

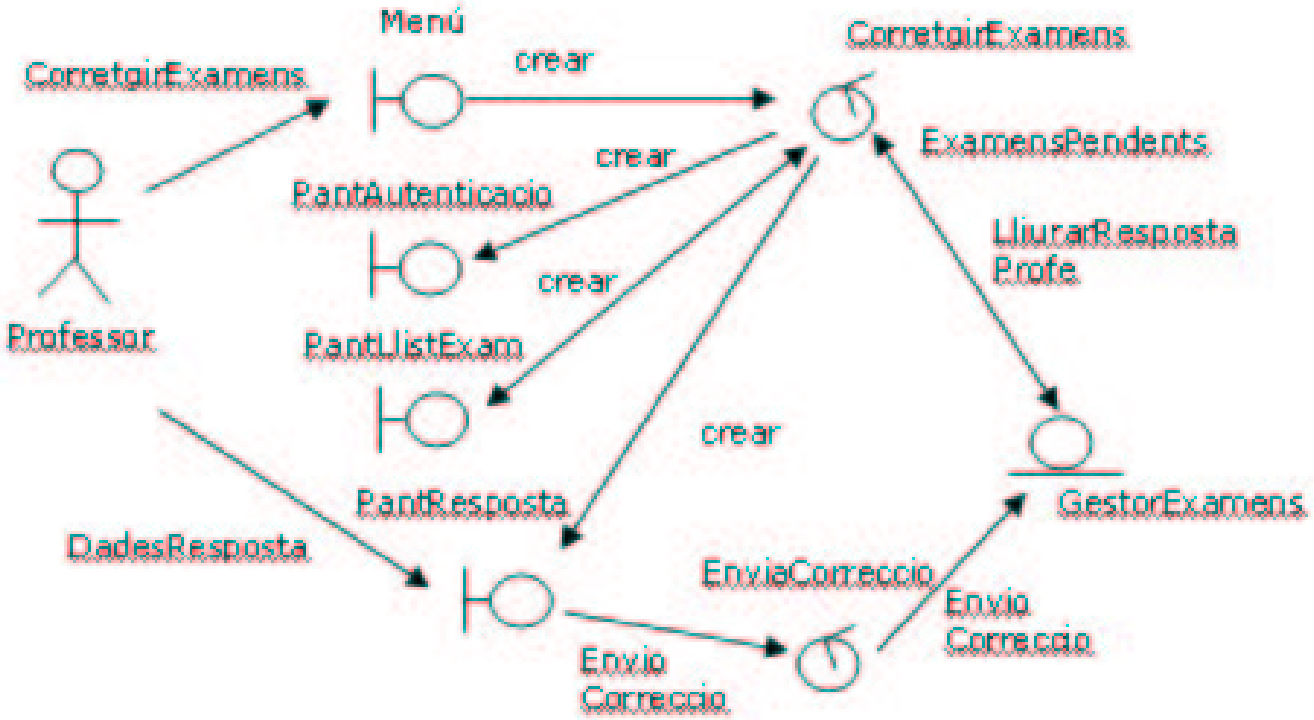

Figura 7: Cas CorretgirExamen

- *Resum de la funcionalitat:* el professor s'autentica davant el "Gestor d'Exàmens". Aquest permet que el professor pugui accedir a la llista de respostes del seu examen i corregir-les. Un cop corregides, l'aplicatiu professor les lliura al "Gestor d'Exàmens"perque aquest les guardi.

5.2.3 Cas d'ús: Revisio

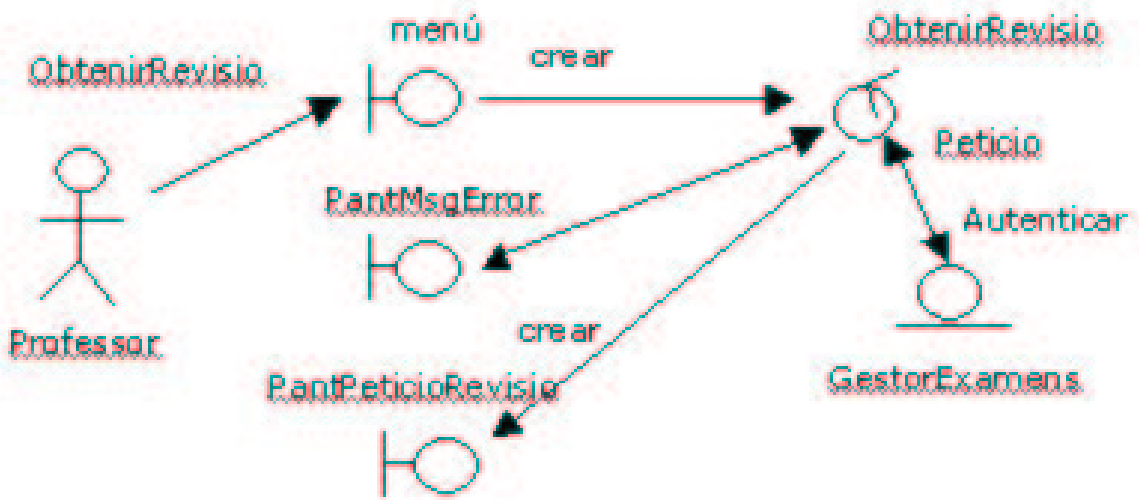

Figura 8: Cas Revisio

- *Resum de la funcionalitat:* el professor s'autentica davant el "Gestor d'Exàmensï aquest li lliura les peticions de revisió d'exàmens.

5.3 Casos d'ús: GestorExàmens

Figura 9: Cas GestorExàmens

GestioExamens es desglosa en els següents casos:

- Guardar la resposta d'un alumne
- Guarda la redacció d'un examen
- Lliurar un examen a l'alumne
- Recuperar l'enunciat d'un examen
- Lliurar la resposta al professor
- Mostrar un llistat d'enunciats d'exàmens
- Guardar correccions
- Enviar correccions
- Guardar una petició de revisió
- LLiurar les peticions de revisió
- Lliurar un llistat d'exàmens realitzats
- Recuperar les revisions.

5.3.1 Guardar Resposta Alumne

Figura 10: Cas GuardarRespostaAlumne

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"guarda a la seva base de dades la resposta entregada per l'alumne.

5.3.2 Cas d'ús: GuardarRedaccioExamen

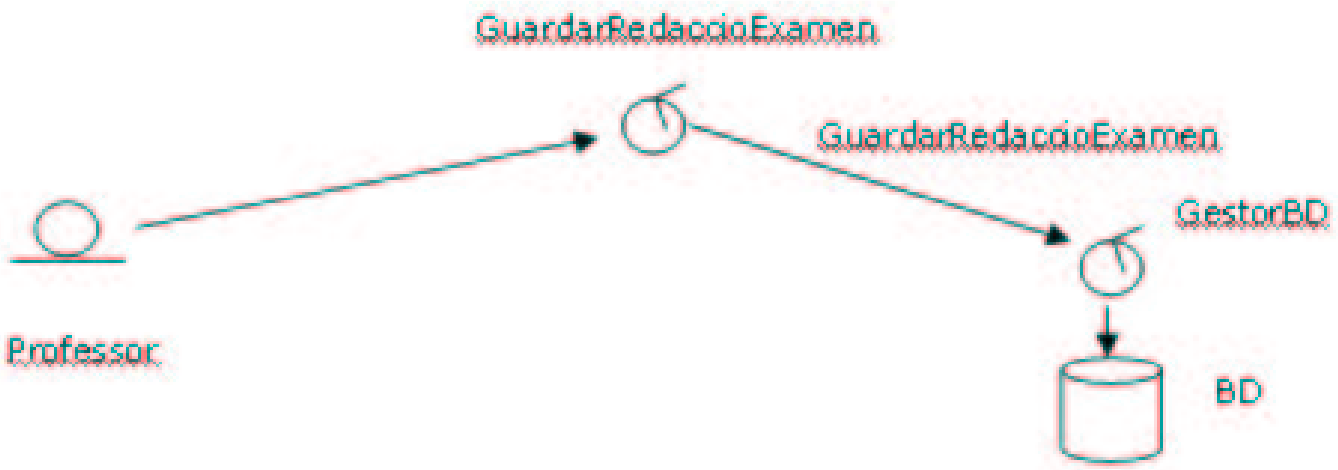

Figura 11: Cas GuardarRedaccioExamen

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"guarda un nou enunciat d'examen redactat per un professor, a la seva base de dades.

5.3.3 Cas d'ús: LliurarExamenAlumne

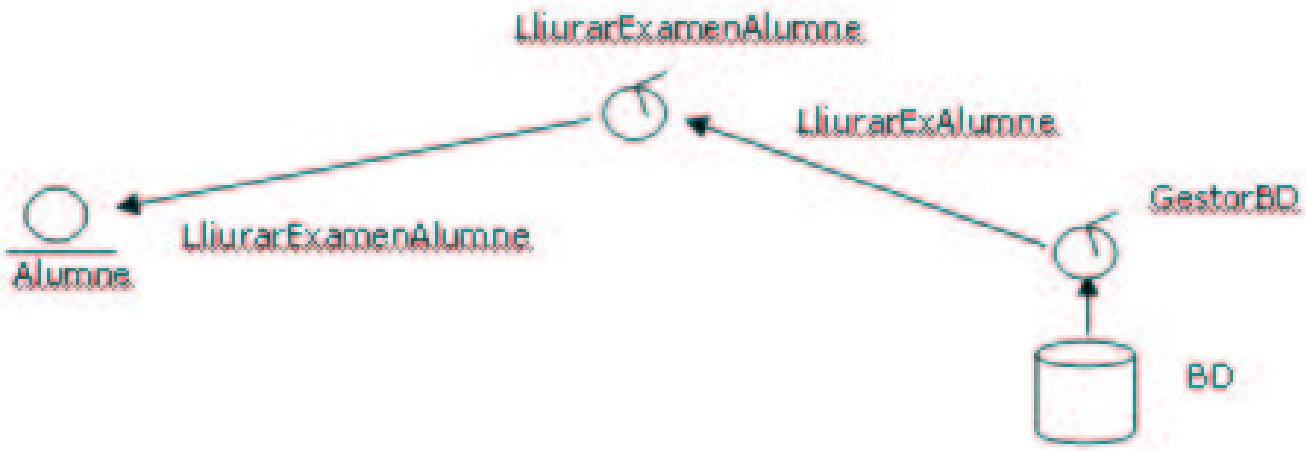

Figura 12: Cas LliurarExamenAlumne

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca un examen a la base de dades i el lliura a l'alumne. corresponent.

5.3.4 Cas d'ús: RecuperarEnunciat

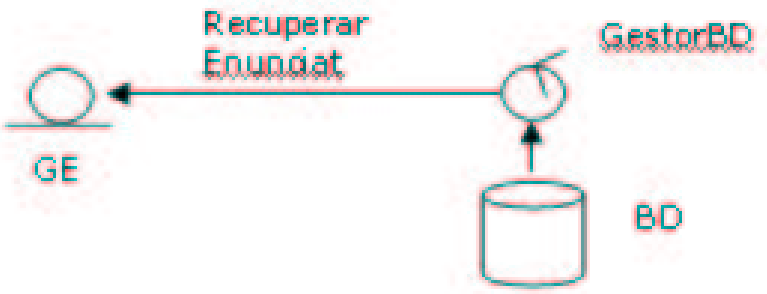

Figura 13: Cas RecuperarEnunciat

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"recupera l'enunciat d'un examen de la base de dades.

5.3.5 Cas d'ús: LliurarRespostaProfe

Figura 14: Cas LliurarRespostaProfe

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca la resposta d'un examen a la base de dades i li lliura al professor.

5.3.6 Cas d'ús: MostrarLListatExamens

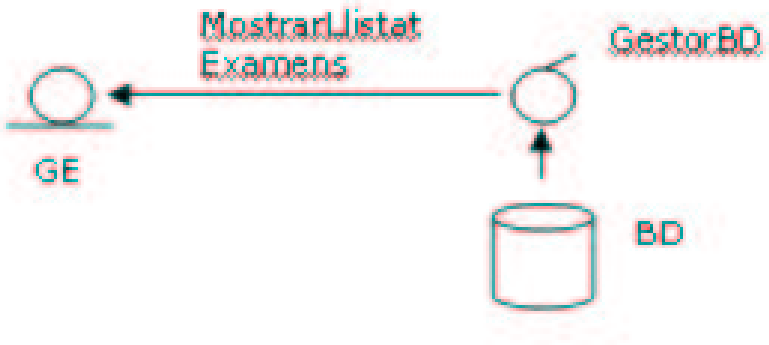

Figura 15: MostrarLlistatExamens

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca tots els enunciats dels exàmens a la base de dades .

5.4 Cas d'ús: GuardarCorreccions

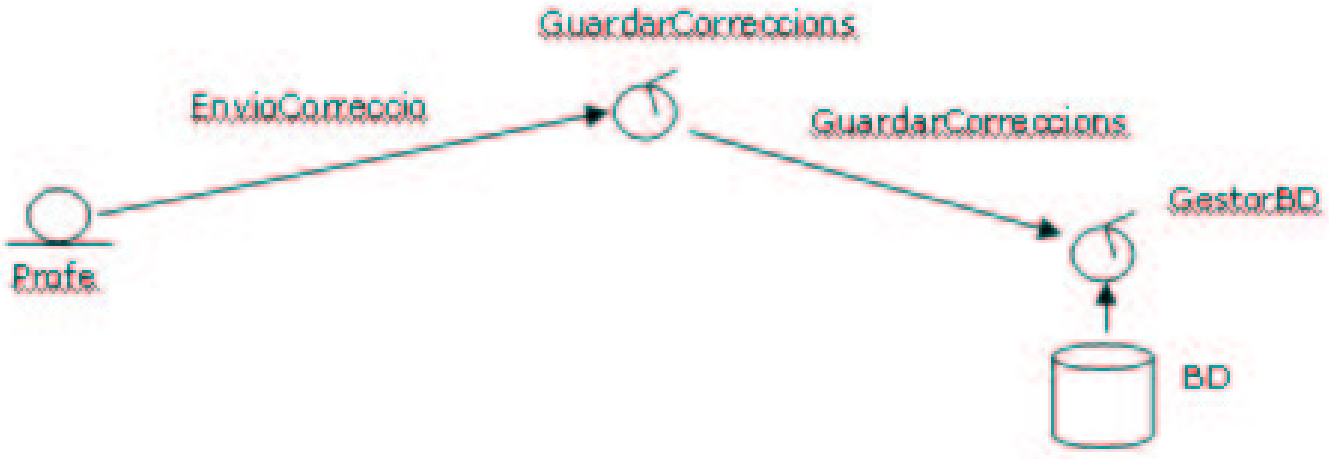

Figura 16: GuardarCorreccions

- *Resum de la funcionalitat:* El "Gestor d'Exàmens"guarda la correcció d'una revisió a la base de dades .

5.4.1 Cas d'ús: EnviaCorreccio

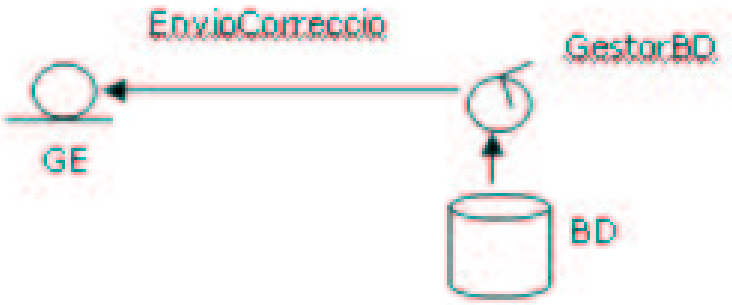

Figura 17: Cas EnviaCorreccio

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca la resposta d'un examen a la base de dades i l'entrega al professor.

5.4.2 Cas d'ús: PeticioRevisio

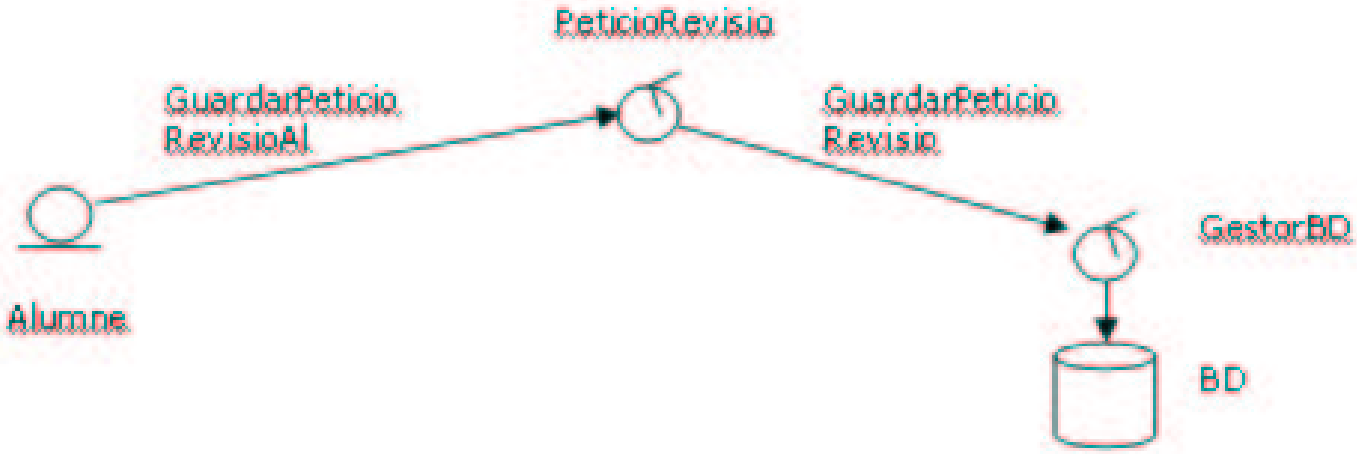

Figura 18: Cas PeticioRevisio

- *Resum de la funcionalitat:* El "Gestor d'Exàmens"guarda una nova petició de revisió a la base de dades.

5.4.3 Cas d'ús: LliurarPeticioRevisions

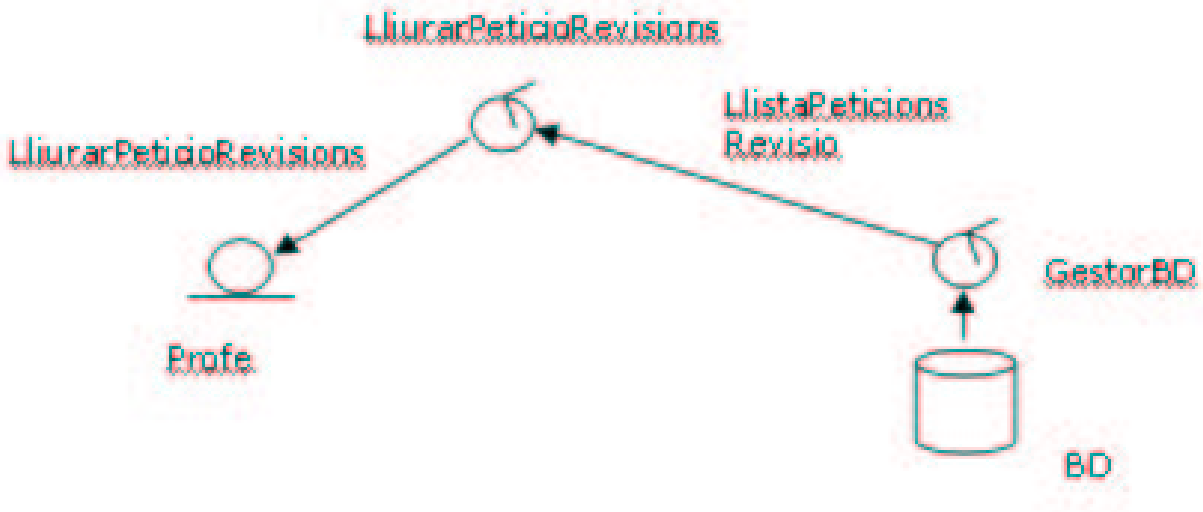

Figura 19: Cas LliurarPeticioRevisions

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca una revisió pendent de corregir a la base de dades i li lliura al professor.

5.4.4 Cas d'ús: LlistatExamensFets

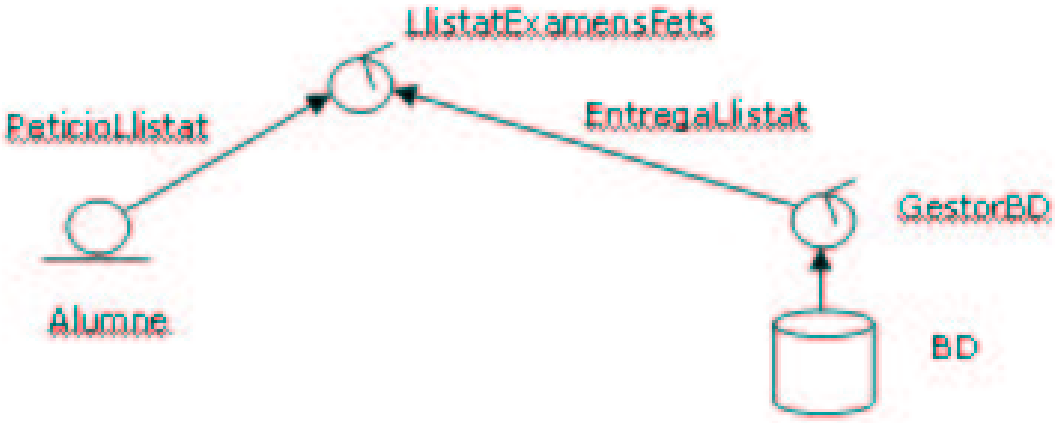

Figura 20: Cas LlistatExamensFets

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca els exàmens realitzats per l'alumne a la base de dades i els hi mostra.

5.4.5 Cas d'ús: RecuperarRevisio

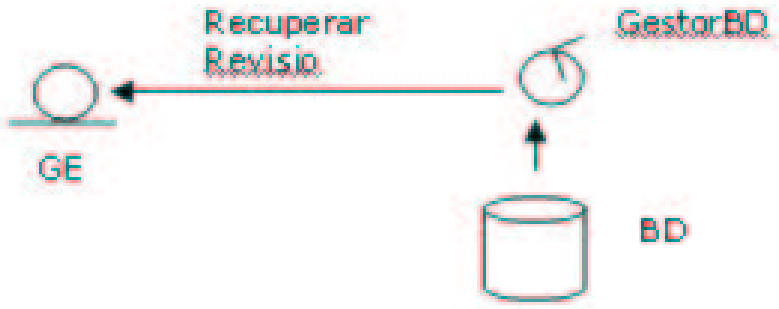

Figura 21: Cas RecuperarRevisio

- *Resum de la funcionalitat:* el "Gestor d'Exàmens"busca la revisió d'un examen a la base de dades.

5.5 Conclusions i linies futures

En aquest projecte s'ha implementat un esquema criptogràfic per realitzar exàmens electrònics de forma segura. Aquest esquema es base en la criptografia de clau pública o asimètrica. Per garantir el nivell de seguretat necessari que requereix aquest tipus d'activitats, s'han dissenyat un conjunt de protocols i processos que garanteixen els quatre conceptes clau de la seguretat de la informació:

- **Confidencialitat**: és la propietat que assegura que només els qui estan autoritzats tindran accés a la informació
- **Integritat**: és la a propietat que assegura la no alteració de la informació
- **autenticació**: és la propietat que fa referència a la identificació
- **no-repudi**: és la propietat que assegura que cap part no pugui negar cap compromís o acció presos anteriorment.

Però per garantir aquesta seguretat, no n'hi ha prou amb aquest protocol. Necessitem una gesti'øcorrecta de les dades per part del "Gestor d'Exàmens" .

Amb el disseny de la base de dades que s'ha fet, no és possible fer una bona gestió dels exàmens. No s'ha dissenyat una taula per guardar els alumnes matriculats ni una taula per gaurdar els professors que imparteixen docència a la universitat. Amb aquest esquema, un alumne situat en un punt controlat podria redactar un nou examen o un professor podria resoldre un examen. Igualment, quant mostrem enunciats d'exàmens o respostes realitzades pels alumnes, es mostren totes.

Per millorar aquesta gestió de les dades, faria falta redissenyar la base de dades per tal de tenir més control sobre els alumnes i els professors.

A continuació s'especifica un nou disseny de la base de dades per millorar aquesta gestió:

- **tb_examens**: existeix actualment. Guardem els enunciats dels exàmens.
- **tb_professors**: guardariem els professors que imparteixen docència a la universitat
- **tb_alumnes**: guardariem els alumnes matriculats a la universitat
- **tb_respostes**: existeix actualment. Guardem les respostes dels professors.
- **tb_correccions**: existeix actualment. Guardem les correccions que fan els professors
- **tb_revisions**: existeix actualment. Guardem les peticions de revisió
- **redacta**: seria una relació (1,N) tb_professor - tb_examens
- **imparteix**: seria una relació (1,N) tb_professor tb_assignatures
- **matriculat**: seria una relació (M,N) tb_alumnes - tb_assignatures
- **realitza**: seria una relació ternària (1,1,1) on guardariem la data i la nota entre tb_examens - tb_respostes - tb_alumnes
- **solicita** : seria una relació (1,N) entre tb_alumnes - tb_revisions
- **te_correcio**: seria una relació (1,0..1) entre tb_correccions - tb_revisions

- **te_correccio**: seria una relació (1,1) entre tb_respostes - tb_correccions.

Un altre aspecte a millorar i referent a la base de dades, seria pensar en migrar la base de dades a un gestor de base de dades més fiable que el Access com per exemple *Orcle* o *mysql*. Access és limitat en quant als camps de tipus *text* o tipus *memo*. El servidor BD podria oferir serveis en un port determinat i dáquesta manera l'accés es faria a través d'una url.

Un altre aspecta a millorar en aquest aplicatiu és la implementació d'un protocol de comunicació entre els diferents aplicatius clients i l'aplicatiu "Gestor d'Exàmens". S'ha estudiat la tecnologia *"Java Remote Method Invocation (Java RMI)"* ja que amb la nostre arquitectura de tres aplicacions hi ha una arquitectura d'objectes distribuïts . Aquesta tecnologia està disponible per la plataforma *Java 2 Standard Edition (J2SE)*, que és el llenguatge amb el qual s'ha desenvolupat el projecte. Però per falta de temps no s'ha pugut implementar.

APer últim, comentar que aquest aplicatiu es podria millorar amb una capa de presentació on els usuaris es puguéssin moure a través d'una GUI.

6 Glossari

Autenticació: fa referència a la identificació. És el nexe d'unió entre la informació i l'emissor d'aquesa.

Autoritat de certificació: tercera part fiable que emet certificats, és a dir, documents electrònics que acrediten l'autenticitat de les claus públiques dels usuaris pertanyents al domini de l'autoritat.

Certificat: document electrònic consistent en una còpia de la clau pública d'un usuari signada pel gestor del directori de claus públiques o per una tercera part fiable.

Clau de xifratge: clau que s'utilitza per a xifrar la informació i obtenir les propietats de confidencialitat i integritat.

Confidencialitat: no ha de ser possible que tercers puguin interceptar els nostres missatges i assabentar-se del que diem.

Criptografia: ciència que estudia les tècniques matemàtiques utilitzades per a la protecció de la informació.

Criptografia de clau compartida: grup de criptosistemes que basen la seva seguretat en una sola clau; emissor i receptor fan servir tant per a xifrar com per a desxifrar.

Criptosistema: mètode que permet desxifrar un text en clar per obtenir-ne un text desxifrat inintel.ligible.

Criptosistema de clau pública: criptosistema en què cada usuari té una clau pública i una de privada; amb aquest criptosistema una parella d'usuaris es pot comunicar confidencialment sense compartir una clau secreta.

DES *(Data Encryption Standard)*: criptosistema de xifratge de blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits.

ElGamal: criptosistema de clau pública i basat en el problema del logaritme discret.

Falsificació: atac criptoanalític que pretén obtenir la signatura d'un cert missatge sense la intervenció del signatari.

Factorització: en matemàtiques, el factorial d'un nombre n és el producte dels enters positius menors o iguals a n.

Integritat: cal que la informació que viatja per la xarxa no es pugui alterar sense que es detecti.

No-repudi: no ha de ser possible que un usuari que ha firmat digitalment un document electrònic pugui negar-ho.

Problema del logaritme discret: problema que consisteix a invertir l'exponenciació modular i que actualment es considera intractable.

RSA: criptosistema de clau pública molt utilitzat, publicat per Rivest, Shamir i Adleman l'any 1978; es basa en el problema de factorització.

Signatura digital: procediment per a signar documents en format electrònic que consisteix en un algorisme de signatura privat del signatari i un algorisme públic per a la verificació de la signatura.

Sobre digital: tècnica que combina la criptografia de clau compartida i la de clau pública per tal d'aprofitar la velocitat de la primera i la flexibilitat de la segona.

Verificació: comprovació que una signatura és vàlida, és a dir, que ha estat efectuada pel pretès signatari. Ha de ser possible per a totom, és a dir, no ha de requerir coneixement de paràmetres secrets.

7 Bibliografia

Referències

[Brett McLaughlin, B.McL. (1997)] Brett McLaughlin, B.McL. (2000). Java and XML. *0´* REILLY.

[Deitel and Deitel (1998)] Deitel and Deitel (1998). Cómo programar en Java. *P*rentice Hall, PHH.

[George Reese, G.R. (1997)] George Reese, G.R. (1997). Database Programming with JDBC and Java. *0´* REILLY.

[UOC (2003)] UOC (2003). Apunts de l'assignatura de criptografia. Universitat oberta de Catalunya.