Sistema de transmissió encriptat J2EE

CRYPTO MAGATZEM

Memòria

Autor: Francesc Gonzàlez Verdugo
Director: Jordi Ceballos Villach
Estudis: Enginyeria Tècnica de Sistemes
Juny de 2006

0 – Resum del Projecte

El projecte consisteix en una aplicació Web que actua com un magatzem central d'arxius, on els usuaris es poden connectar via https des de qualsevol localització, independentment del grau de seguretat que implementi la seva estació, i enviar arxius encriptats o sense encriptar amb les garanties màximes de seguretat i confidencialitat. Els fitxers restarien emmagatzemats a una BBDD on posteriorment, només usuaris acreditats amb una clau AES d'encriptació/desencriptació els podrien recuperar, també via HTTPS, i procedir a desencriptar-los a nivell local.

Qualsevol entorn de comunicacions ha de tenir present la seguretat en tot enviament d'informació que es faci cap a internet o amb origen Internet. Una vegada que la informació surt de les nostres xarxes corporatives, una plèiade de diferents agents poden capturar-la i trencar la seva confidencialitat.

El sistema d'enviament de fitxers es veurà complementat per un mòdul d'administració d'usuaris, on un administrador podrà editar tant perfils d'usuaris com usuaris per tal de variar les seves dades o permisos personals.

Total l'aplicació serà implementada a l'entorn J2EE utilitzant planes .jsp, Servlets i llenguatge java. Farem servir Apache Tomcat com a servidor web i MySQL com a servidor de BBDD.

Àrea del TFC: J2EE
Àmbits addicionals: Criptografia

1 - Taula de Continguts

0 Resum del projecte 2
1 Taula de Continguts 3
2 Introducció, objectius i planificació 5
2.1 Introducció 5
2.2 Objectius 5
2.3 Planificació 6
2.4 Requeriment Bàsics 6
3 Disseny de l'arquitectura 8
3.1 Introducció 8
3.2 MVC-Model-Vista-Controlador 8
3.3 Diagrama de fluxe dels servlets 10
4 Disseny de la BBDD 11
4.1 Diagrama ER 11
4.2 Diagrama Lògic 12
4.3 Descripció de les Taules 12
4.3.1 Taula Usuari 13
4.3.2 Taula UPLOADS 13
4.3.3 Taula Perfil 14
5 Disseny de classes 15
5.1 Classes del Model 15
5.1.1 Diagrama estàtic de classes 15
5.1.2 Mètodes de classe 16
5.1.2.a Classe “AdminUsuari” 16
5.1.2.b Classe “AdminPerfil” 17
5.1.2.c Classe “AppletCipher” 17
5.2 Classes del Controlador 18
5.3 Classes de la Vista 20
5.3.1 Arquitectura de la interfície (Vista) 20
5.3.2 Relació de pàgines 21
5.3.2.a Pàgines d'estructura 21
5.3.2.b Pàgines de continguts 21
5.3.3 Pàgines de l'aplicació 23
5.3.3.a Pàgines comuns 23
5.3.3.b Pàgines de l'Administrador 24

5.3.3.c Pàgines d'usuaris 35
6 Encriptació de fitxers 38
6.1 Anàlisi sobre l'operativa d'encriptació i desencriptació d'arxius 39
7 Valoració econòmica del projecte 41
8 Conclusions 42
9 Línies de desenvolupament futur 43
10 Bibliografia 44

2 – Introducció, objectius i planificació

2.1 Introducció

La tecnologia que ofereix J2EE es perfila des de fa uns anys com la gran competidora de l'entorn .NET de Microsoft al mercat de les solucions empresarials. Conèixer tant els components que conformen aquesta àrea de desenvolupament, com el seu funcionament intern, atorguen a un professional de la programació d'unes extraordinàries possibilitats d'èxit a la seva activitat laboral.

El fet de realitzar un TFC centrat a l'entorn J2EE i al llenguatge JAVA sense disposar prèviament d'una sòlida base de coneixements, més que considerar-se com un gran handicap, es pot veure com una magnífica ocasió per a endinsar-nos dins d'aquest apassionant àmbit de la programació distribuïda.

La utilització de components com per exemple: Servlets, planes .jsp, arquitectura de dos i tres capes, interfícies d'accés web, comunicacions encriptades i, el que és més important, l'aplicació pràctica a l'entorn actual de les IT dins l'univers empresarial, fan que la realització d'aquest TFC sigui un repte molt atractiu d'endegar.

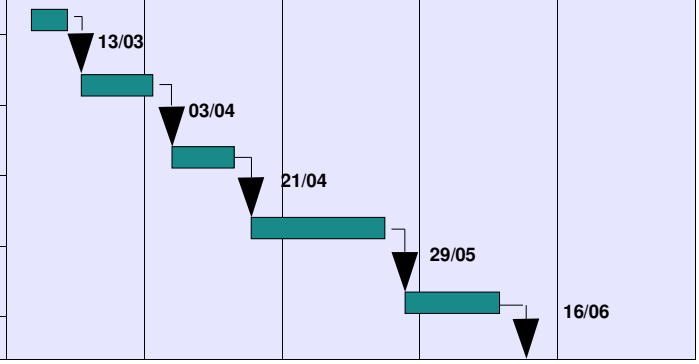
2.2 Objectius

El principal objectiu ha estat intentar arribar a assolir un nivell de coneixement mitjà de totes les tecnologies que s'utilitzaran durant el desenvolupament d'aquest treball que ens ha permès fer una primera incursió a l'entorn J2EE i sobretot, arribar a enllestir una aplicació plenament funcional, pràctica i d'interès per a una estructura empresarial o universitària.

2.3 Planificació

Al diagrama de Gantt que adjuntem, podem veure de manera esquemàtica la planificació del projecte al llarg del quadrimestre:

id	Tasca a lliurar	termini	principi	final
1	Pla de Treball	9 dies	04/03/06	13/03/06
2	Anàlisi i 1ª versió prot.	21 dies	13/03/06	03/04/06
3	Disseny i versió defi. Prot.	19 dies	03/04/06	21/04/06
4	Implementació	31 dies	21/04/06	29/05/06
5	Memòria i PPT	19 dies	29/05/06	16/06/06
6	Lliurament Final	1 día	16/06/06	16/06/06



2.4 Requeriments tècnics

La realització d'aquest projecte ha fet convergir tecnologies tant diferents com poden ésser, el model de programació de tres capes de l'entorn J2EE, la criptografia o les aplicacions web. Si fem un anàlisi més detallat, podríem establir les següents tres categories:

Llenguatge de programació

Tant l'aplicació Web com la de Control, s'han implementat a l'entorn J2EE, el que vol dir que hem fet servir planes .jsp i servlets per a oferir els serveis d'enviar/rebre arxius, encriptació dels mateixos i emmagatzematge a una BBDD. Tots aquests components han estat desenvolupats en JAVA i allotjats a un server Apache Tomcat.

Entorn de desenvolupament

Com a eina bàsica hem utilitzat Eclipse 3.1.2 amb els plugins adients per a fer proves al server Tomcat. Respecte al llenguatge, JAVA amb el seu entorn de desenvolupament JDK 1.5.0_04.

Per tal de realitzar les operacions d'encriptació, s'han utilitzat les llibreries de JAVA Bouncy Castle i JCE de SUN, que implementen les funcionalitats necessàries per a realitzar les operacions de codificació i descodificació.

Sistema gestor de BBDD

Farem servir com a gestor bàsic MySQL, tot i que el projecte s'ha desenvolupat amb prou modularitat per tal d'afegir al futur qualsevol altre tipus de BBDD com puguin ésser ORACLE o semblants.

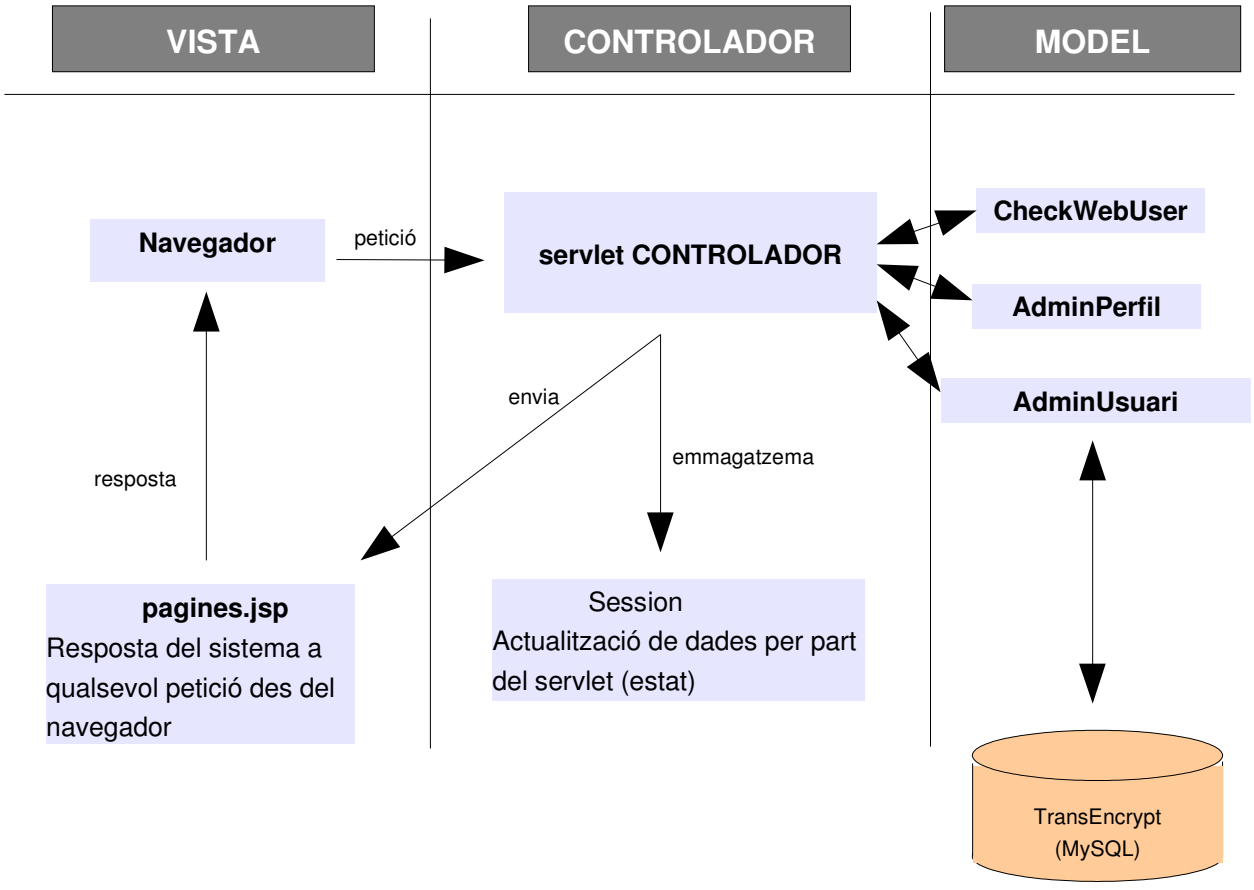
3 – Disseny de l'arquitectura de l'aplicació

3.1 Introducció

El desenvolupament de codi ha de respectar una sèrie de pautes que defineixen estructuradament tot el procés des que sintetitzem una idea primera fins que obtenim tots els components de l'aplicació compilats i funcionant plegats. Hem d'aplicar una lògica de disseny a tot el procés de creació d'una aplicació. Tot i que existeixen diferents patrons de disseny diferenciats per la complexitat que desenvolupen o per l'entorn cap a on son dirigits, un model destaca sobre els altres, MVC.

3.2 MVC – Model – Vista - Controlador

El gràfic que segueix reflecta l'aplicació del patró de disseny MVC al nostre projecte.



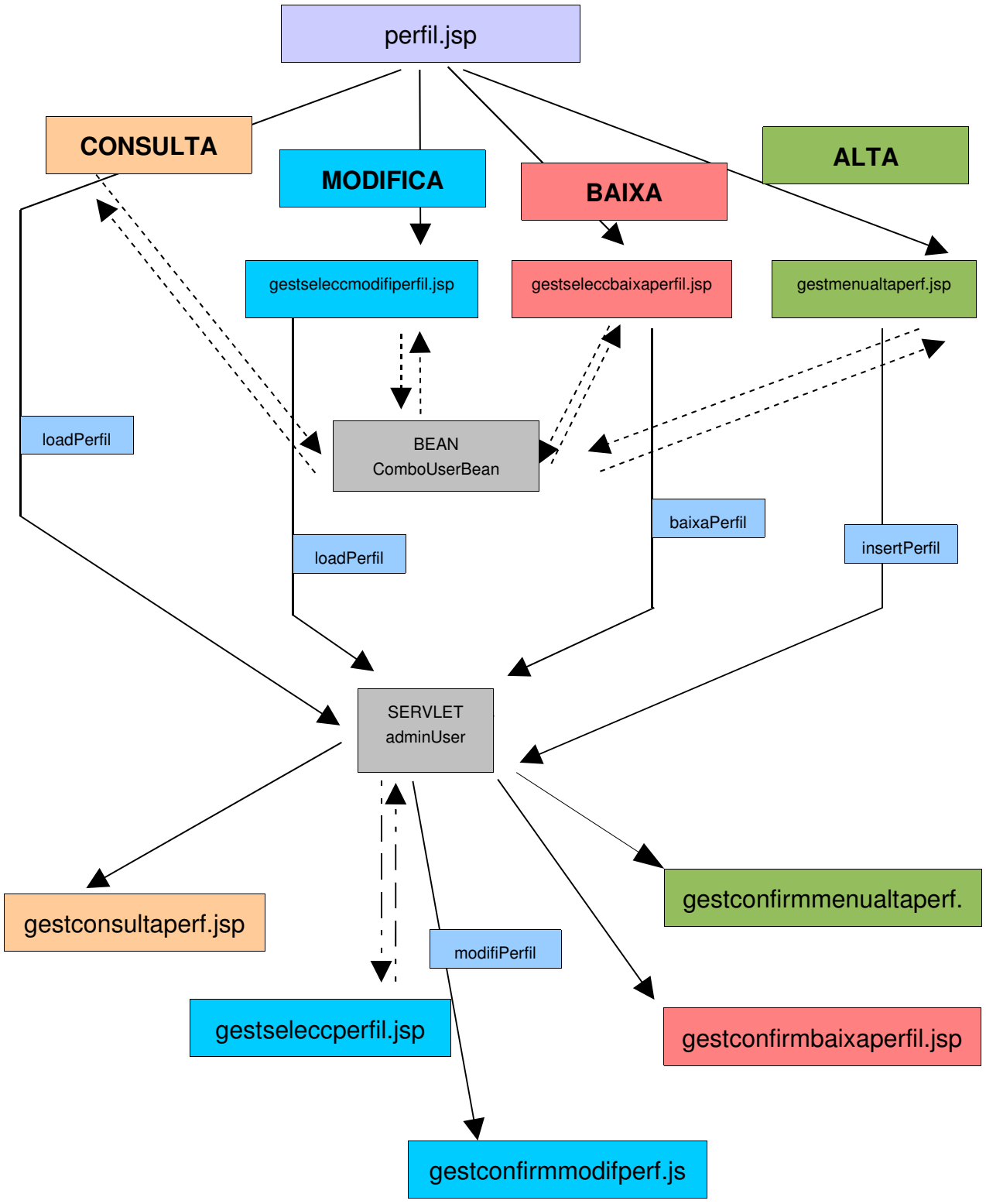
MVC o Model view Controler és un patró de disseny aportat originalment pel llenguatge SmalTalk a la Enginyeria de Software. El paradigma MVC consisteix en dividir les aplicacions en tres parts:

.-Controlador
.-Model
.-Vista

El **controlador** és l'encarregat de redirigir o assignar una aplicació (un model) a cada petició; el controlador ha de disposar d'un "mapa" de correspondències entre peticions i respostes (aplicació o model) que s'han de correspondre.
El **model** seria l'aplicació que respon a una petició, és la lògica del sistema. Una vegada realitzades les operacions necessàries, el flux de la petició transformada per aquestes operacions, torna cap al controlador que retorna els resultats a una **vista** (interfície visible) totalment deslligada de la lògica del sistema.

La separació en tres capes: presentació, lògica de negoci i accés a dades, és fonamental per tal que el desenvolupament d'arquitectures consistents, reutilitzables i de senzill manteniment, sigui cada vegada més estandarditzat i representi un estalvi de temps i monetari de cara a posteriors projectes.

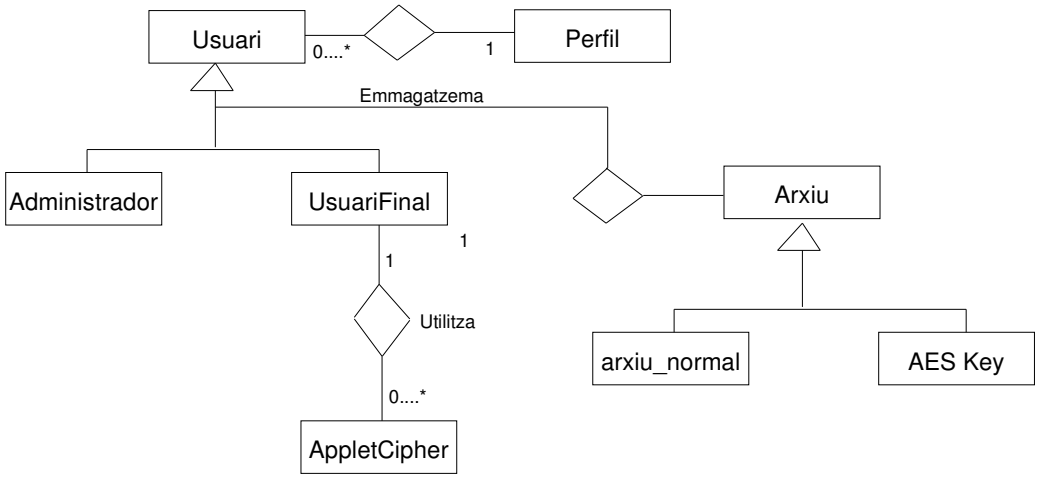
3.3 Diagrama de fluxe dels servlets



4 – Disseny de la BBDD

4.1 Diagrama ER

Dins aquest diagrama reflectirem tots aquells objectes que hauran de mantenir les seves dades una vegada l'aplicació sigui fora de funcionament. Els objectes presents seran les futures classes del **Model**, utilitzant-se la BBDD per desar els seus valors.

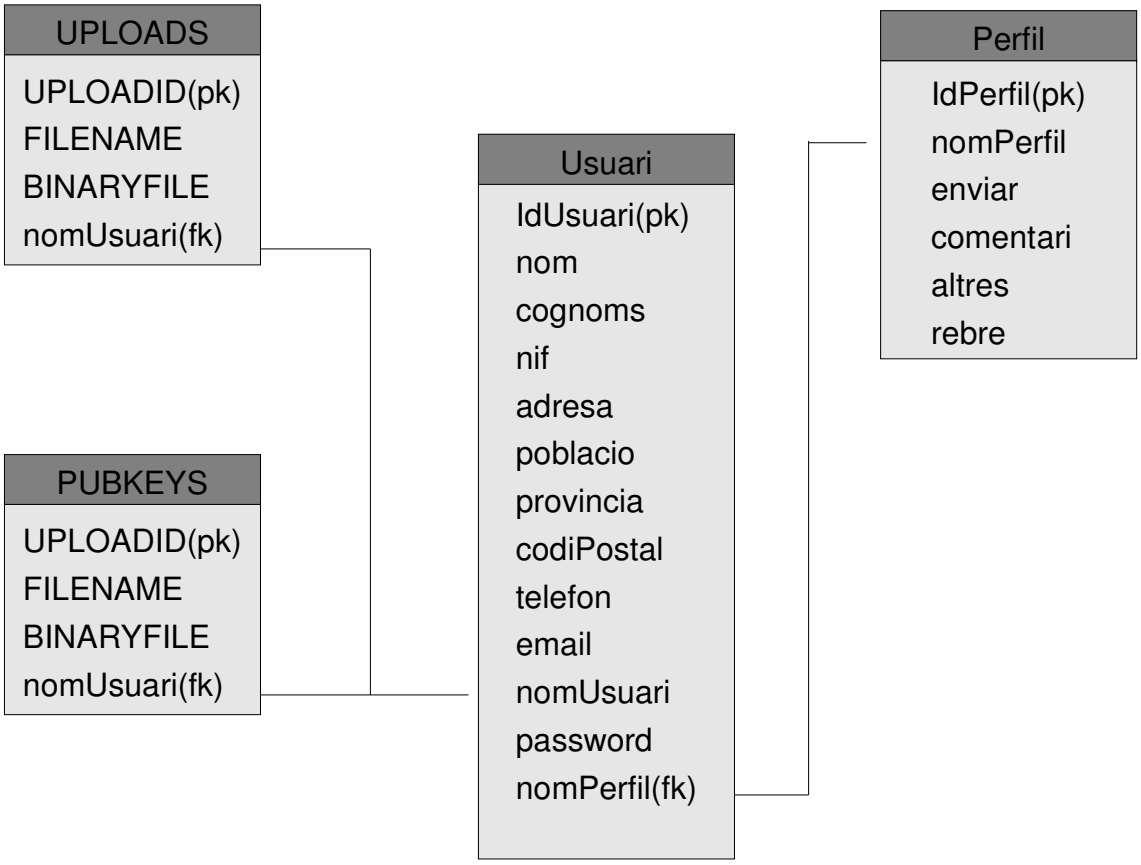


Les relacions d'herència s'han indicat a l'esquema per mostrar informació sobre les relacions entre les classes, tot i que posteriorment no devindran entitats definides.

4.2 Diagrama Lògic

Es farà servir aquest diagrama per poder apreciar, d'una manera més entenedora, les taules que composaran la BBDD i les relacions entre les mateixes.

Es pot apreciar com existeix una relació entre el camp “nomUsuari” de la taula “Usuari” i el camp “nomUsuari” de la taula “UPLOADS”, totes dues taules relacionen el mateix camp. Per altra banda, tindrem també una clara relació entre els camps “nomPerfil” i de les taules “Usuari” i “Perfil”, respectivament.



4.3 Descripció de les Taules

Obtindrem cada taula de la BBDD en funció de les entitats que hem identificat al diagrama ER: Usuari, UPLOADS i Perfil. Les relacions entre les entitats vindran donades per l'existència de claus foranies que connectin les diferents taules.

4.3.1 Taula Usuari

Aquesta taula mantindrà totes les dades dels usuaris que faran servir l'aplicació.

Camp	Tipus	Clau	Descripció	NULL
idUsuari	int(4)	Primària	Identificador de l'usuari	No
nom	varchar(50)		Nom de l'usuari	Si
cognoms	varchar(50)		Cognoms de l'usuari	Si
nif	varchar(9)		Nif identificació	Si
adreça	varchar(150)		Adreça de l'usuari	Si
poblacio	varchar(100)		Població de l'usuari	Si
provincia	varchar(100)		Província de l'usuari	Si
codiPostal	varchar(10)		Codi postal de l'usuari	Si
telefon	varchar(20)		Telèfon de l'usuari	Si
email	varchar(100)		Adreça electrònica de l'usuari	Si
nomUsuari	varchar(25)		Nom identificació de l'usuari	Si
password	varchar(20)		Clau d'identificació de l'usuari	Si
nomPerfil	int(4)	Forània	Tipus de perfil de l'usuari	No

4.3.2 Taula UPLOADS

Mitjançant aquesta taula, desarem totes les dades respecte als arxius encriptats que emmagatzemarem a la BBDD.

Camp	Tipus	Clau	Descripció	NULL
UPLOADID	varchar(10)	Primària	Identificador del fitxer	No
FILENAME	varbinary(8000)		Dades codificades de l'arxiu	Si
BINARYFILE	Longblob		Codificació d'un arxiu.	Si
nomUsuari	varchar(50)		Nom al format abans d'encriptar	Si

4.3.3 Taula Perfil

Aquesta taula mantindrà les dades referents als diferents tipus de perfils que existeixin al sistema.

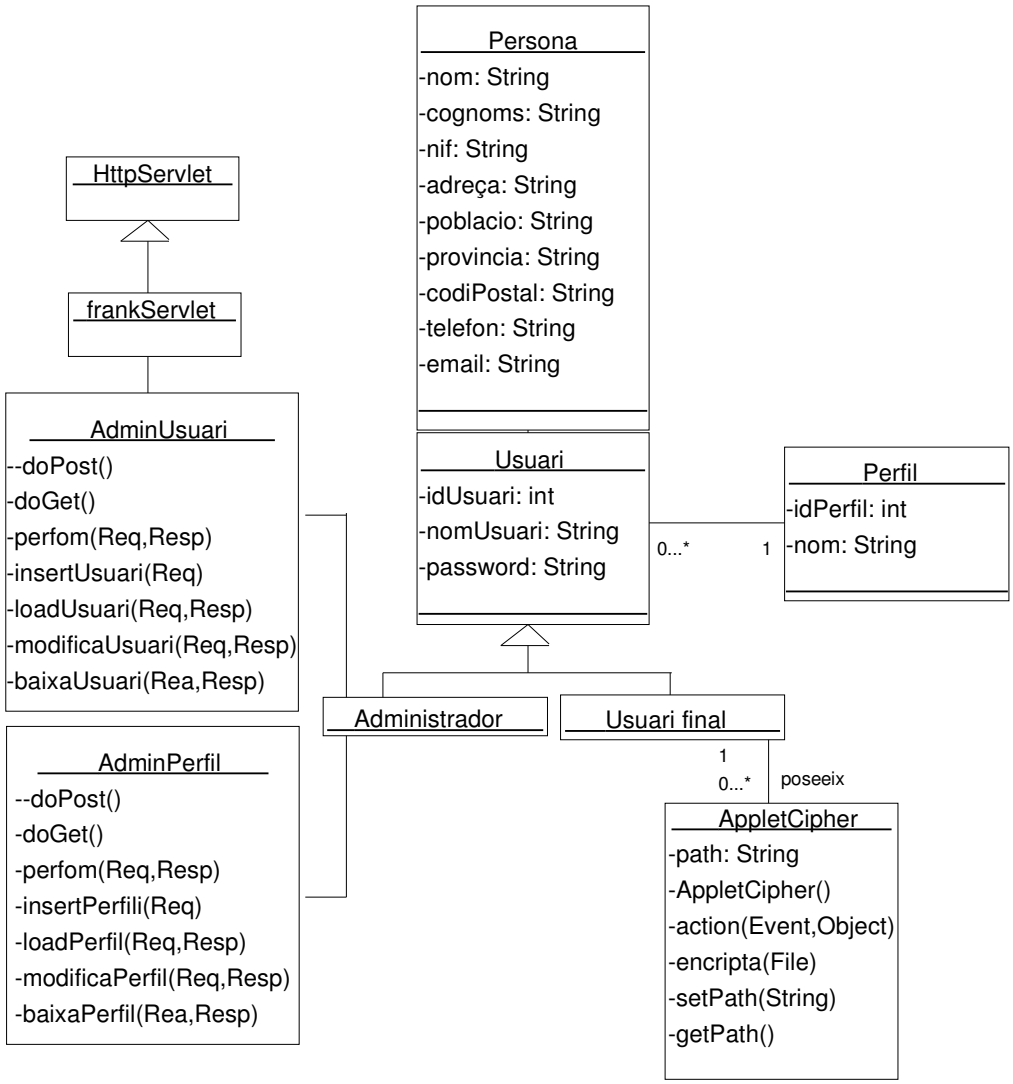
Camp	Tipus	Clau	Descripció	NULL
idPerfil	int(4)	Primària	Codi del perfil	No
nomPerfil	varchar(50)		Denominació del perfil	No
enviar	Tinyint(1)		Boolean enviar arxius	Si
comentari	varchar(300)		Comentaris afegits	Si
altres	varchar(300)		Comentaris afegits	Si
rebre	Tinyint(1)		Boolean enviar arxius	Si

5 – Disseny de Classes

5.1 Classes del Model

5.1.1 Diagrama estàtic de classes

Aquest diagrama reflectirà aquelles classes de l'arquitectura MVC que pertanyen al **Model**, parlem de les classes d'entitat; AdminUsuari, AdminPerfil, a més a més de les classes que, obligatòriament, ens caldran per accedir a la BBDD, o sigui, . Dins l'esquema, reflectirem relacions d'herència per tal de fer-ho més entenedor, malgrat que posteriorment no devindran en classes efectives amb el seus mètodes i atributs.



5.1.2 Mètodes de classe

Cada classe que s'ha identificat anteriorment vindrà conformada amb tota una sèrie de mètodes que es faran servir tant per construir objectes de la mateixa classe, com per atorgar valors als atributs de la mateixa.

5.1.2 a Classe "AdminUsuari"

L'objecte usuari ha d'ésser totalment modificable, hem de disposar de mètodes que puguin editar el valor dels seus atributs.

- *public void doPost(HttpServletRequest request, HttpServletResponse response)*
Utilitzarem aquest mètode per implementar accions referents a la funcionalitat Post del servlet.

- *public void doGet(HttpServletRequest request, HttpServletResponse response)*
Utilitzarem aquest mètode per implementar accions referents a la funcionalitat Get del servlet.

- *perform(HttpServletRequest request, HttpServletResponse response)*
Cridarem una implementació concreta dels request response.

- *insertUsuari(HttpServletRequest request)*
Fem una alta d'usuari utilitzant el request.

- *loadUsuari(HttpServletRequest request, HttpServletResponse response)*
Selecciona un usuari de la BBDD a partir del seu codi d'usuari passat pel request.

- *modifiUsuari(HttpServletRequest request, HttpServletResponse response)*
Modifica un usuari de la BBDD prenent com a dada de localització, el seu codi d'usuari passat pel request.

- *baixaUsuari(HttpServletRequest request, HttpServletResponse response)*
Esborra un usuari de la BBDD prenent com a dada de localització, el seu codi d'usuari passat pel request.

5.1.2 b Classe “AdminPerfil”

L'objecte usuari ha d'ésser totalment modificable, hem de disposar de mètodes que puguin editar el valor dels seus atributs.

- *public void doPost(HttpServletRequest request, HttpServletResponse response)*
Utilitzarem aquest mètode per implementar accions referents a la funcionalitat Post del servlet.

- *lpublic void doGet(HttpServletRequest request, HttpServletResponse response)*
Utilitzarem aquest mètode per implementar accions referents a la funcionalitat Get del servlet.

- *perform(HttpServletRequest request, HttpServletResponse response)*
Cridarem una implementació concreta dels request response.

- *insertPerfil(HttpServletRequest request)*
Fem una alta de perfil utilitzant el request.

- *loadPerfil(HttpServletRequest request, HttpServletResponse response)*
Selecciona un usuari de la BBDD a partir del seu codi de perfil passat pel request.

- *modifiPerfil(HttpServletRequest request, HttpServletResponse response)*
Modifica un usuari de la BBDD prenent com a dada de localització, el seu codi de perfil passat pel request.

- *baixaPerfil(HttpServletRequest request, HttpServletResponse response)*
Esborra un usuari de la BBDD prenent com a dada de localització, el seu codi de perfil passat pel request.

5.1.2 c Classe “AppletCipher”

Aquesta classe proporcionarà tots els mètodes per a la codificació d'arxius.

- *AppletCipher()*
Constructor de la classe.

- *action(Event evt, Object arg)*
Mètode principal de la classe .

- *encripta (File desFile)*
Mètode que realitza tota l'operativa de l'encriptació.

- *setPath(String path)*
Setter per a afegir la ruta cap a l'arxiu a encriptar.

- *getPath()*
Getter per a obtenir la ruta cap a l'arxiu a encriptar.

5.2 Classes del Controlador

El **controlador** serà conformat per un conjunt de classes que heretaran d'una única classe que actuarà com a servlet Controlador, quines funcionalitats s'encarregaran de redirigir totes les peticions que es facin des de la interfície de l'usuari, a **vista**, cap al **model** on es cridaran les classes per connectar amb la BBDD i preparar les planes .jsp que posteriorment es presentaran a l'usuari novament a la **vista**.

Respecte als mètodes comuns que presenten els servlets controladors, poden ésser molt variats però a nivell d'orientació podem indicar uns de bàsics:

- *service(HttpServletRequest request HttpServletResponse response)*
Mètode bàsic del servlet que recollirà tant les sol·licituds fetes des de la plana web com emetrà les respostes en forma de planes .jsp

- *init(...)*
Mètode que ens permetrà inicialitzar una connexió amb les classes del **model** i obrir un objecte de connexió amb la BBDD.

- *doPost(...)*
Aquest mètode implementarà les planes .jsp com a resposta a requeriments HTTP POST.

- *doGet(...)*

Idem de l'anterior per a requeriments HTTP GET.

- *destroy(...)*

El mètode destroy tancarà la connexió amb la BBDD eliminant l'objecte creat pel mètode init.

5.3 Classes de la Vista

5.3.1 Arquitectura de la interfície (Vista)

Respecte a la Vista, hem d'indicar que mantindrem cinc seccions fixes estructuralment que seran la **capçalera**, el **menú lateral**, el **cos central,** la **identificació d'usuari** i els **Annexes Laterals**. Aquestes estructures mantindran la seva distribució i dimensions, tot i que el seu contingut variarà en funció del tipus d'usuari que estigui connectat i de les opcions de que aquest disposi.
Acte seguit, exposem un exemple del que podrien ésser una plana .jsp del projecte amb codi de crida entre una plana jsp al servletControlador i una altre on es procedeix a cridar una plana jsp amb dades proporcionades pel mateix servlet.

5.3.2 Relació de pàgines

5.3.2 a Pàgines d'estructura

Ja hem vist a l'apartat anterior la divisió d'estructures fixes que sempre apareixeran a les planes .jsp, aquestes seran: - header.jpg, gestmenu.jsp, menu.jsp, banners.jsp

5.3.2 b Pàgines de continguts

Per a cada pàgina definida al document d'anàlisi disposarem d'una plana .jsp, a continuació llistem totes les que apareixeran al sistema:

Mòdul	Pantalla d'anàlisi	fitxer
Administració	Pantalla inicial	**Gestmenu.jsp**
	Alta perfil	**GestMenuAltaPerf.jsp**
		GestConfirmMenuAltaPerf.jsp
	Consulta perfil	**GestSeleccConsultaPerfil.jsp**
		GestConsultaPerf.jsp
	Modifica perfil	**GestSeleccModifPerfil.jsp**
		GestSeleccPerfil.jsp
		GestConfirmModifPerf.jsp
	Baixa perfil	**GestSeleccBaixaPerfil.jsp**
		GestConfirmBaixaPerf.jsp
	Alta usuari	**GestMenuAltaUser.jsp**
		GestConfirmMenuAltaUser.jsp
	Consulta usuari	**GestSeleccConsultaUser.jsp**
		GestConsultaUser.jsp
	Modifica usuari	**GestSeleccModifUserl.jsp**
		GestSeleccUser.jsp
		GestConfirmModifUser.jsp
	Elimina usuari	**GestElimiUser.jsp**
		GestQuestionarElimi.jsp
		GestConfirmElimiUser.jsp
	Baixa usuari	**GestSeleccBaixaUser.jsp**
		GestConfirmBaixaUser.jsp

Mòdul	Pantalla d'anàlisi	Fitxer
Connexió	Pantalla inicial	**login.jsp**
	Redirecció d'error	**Error.jsp**

Mòdul	Pantalla d'anàlisi	Fitxer
Gestió d'arxius	Pantalla inicial	**GestMenu.jsp**
	Enviament fitxers	**DataBaseUpload.jsp**
	Descàrrega fitxers	**AdminBBDDUser.jsp**
	Desencriptació	**Desencriptaradmin.jsp**
		Desencriptaradmin2.jsp

5.3.3 Pàgines de l'aplicació

5.3.3 a Pàgines de comunes

login.jsp

En aquesta pantalla d'accés a l'aplicació únicament disposarem com a elements rellevants, els controls d'usuari i contrasenya

5.3.3 b Pàgines de l'Administrador

gestmenualtaperfil.jsp

Des d'aquesta pantalla i amb el popup inferior, podrem donar d'alta un nou perfil.

gestselecccconsultaperfil.jsp

Aquesta funcionalitat ens permetrà visualitzar les dades d'un perfil determinat.

gestseleccmodifPerfil.jsp

D'aquesta manera podrem, previ escollir un perfil del combo, accedir a la pantalla per modificar les dades determinades d'un perfil.

gestseleccBaixaPerfil.jsp

Amb el popup inferior, escollirem un perfil determinat per tal de donar-lo de baixa.

admin.jsp

Aquesta pantalla ens permet visualitzar els arxius que cada usuari te pujats a la BBDD, tanmateix com fer un download d'aquest.

La mateixa pantalla ens serveix per visualitzar les claus AES que els usuaris han desat a la taula de claus.

desencriptaradmin.jsp

L'administrador accedirà a aquesta plana per realitzar operacions de desencriptament d'arxius d'usuaris. Disposarà d'un primer control per seleccionar els arxius que prèviament a desat a nivell local, i un altre per escollir una clau AES d'usuari que també ha desat.

gestmenualtauser.jsp

Popup desplegable on podrem introduir les dades d'un nou usuari escollint el perfil determinat al que pertany.

gestselecccconsultauser.jsp

Pantalla on obtindrem totes les dades d'un usuari.

gestseleccmodifiUser.jsp

Aquesta funcionalitat ens permet editar totes les dades d'un usuari determinat.

gestseleccBaixauser.jsp

Aquest popup ens permet donar de baixa definitivament un usuari de la bbdd.

5.3.3 b Pàgines d'usuaris

dadespersonalsuser.jsp

Aquesta pantalla li permetrà modificar les dades a l'usuari, una vegada realitzada l'acció, prement sobre “Modificar”, els canvis seran desats a la BBDD.

adminbbddUser.jsp

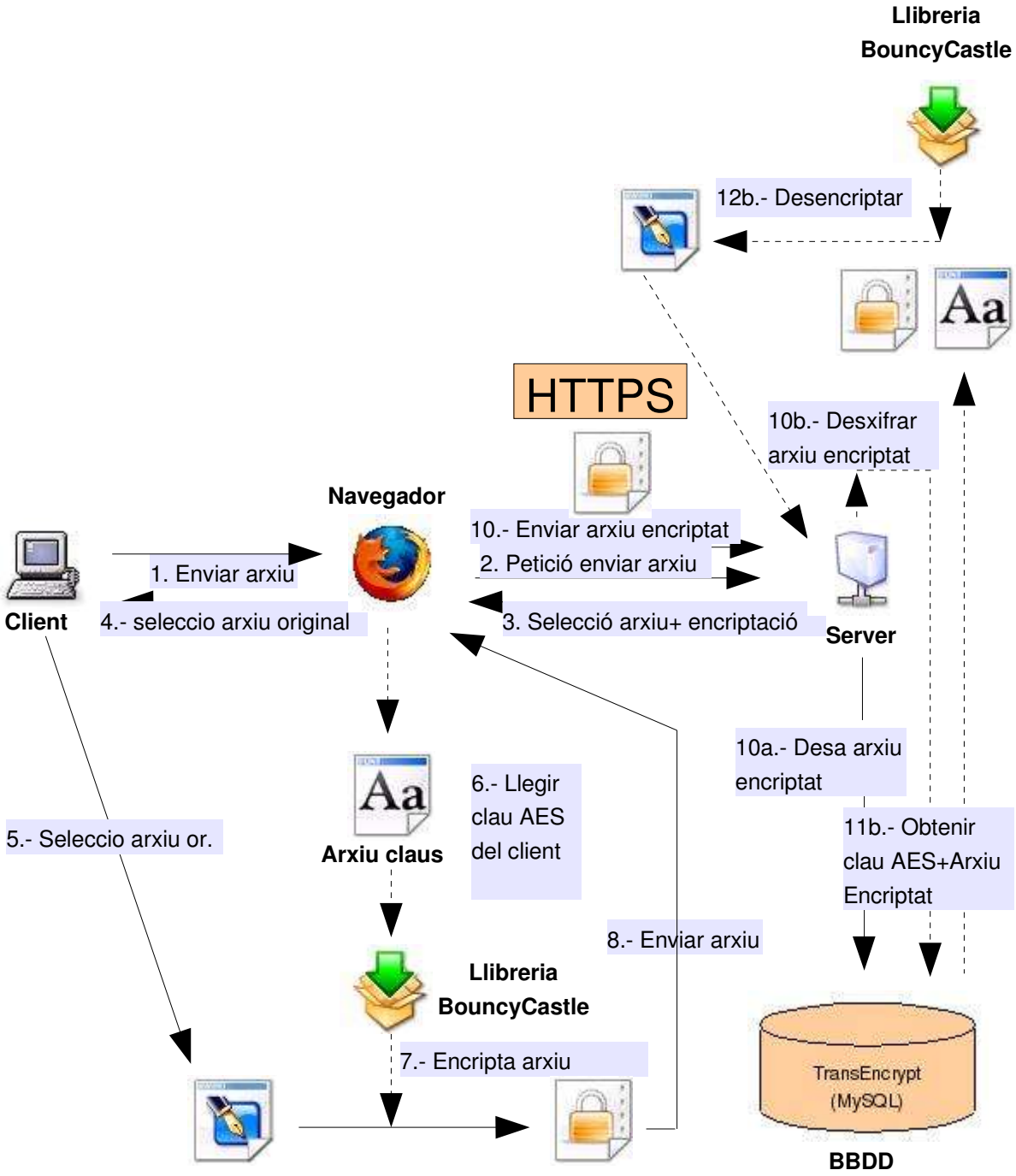
Aquesta funcionalitat li permet a l'usuari visualitzar tots els arxius que ha pujat al servidor. També li serà permès fer downloads d'aquest arxius per a esser posteriorment visualitzats o desencriptats.

DatabaseUpload.jsp

Aquesta pantalla li presenta a l'usuari la funcionalitat de pujar arxius a la bbdd.

6 – Encriptació de fitxers

A aquest gràfic podem veure el sistema que utilitzarem per tal de realitzar l'enviament d'arxius encriptats al server. El procés de descàrrega de fitxers seria l'invers.



El suport que dona el JDK a la criptografia es divideix en dos grans blocs, el JCA “Java Cryptography Architecture” i el JCE “Java Cryptography Extension”, la primera part ens defineix les bases del suport criptogràfic, i la segona ens proveeix dels algorismes necessaris per a poder encriptar i desencriptar dades.

Degut a les lleis dels Estats Units, que prohibeixen exportar software d'encriptació de dades, el JCE no ve inclòs al JDK i està restringit l'accés al mateix mitjançant la web de SUN, malgrat això, existeixen paquets de terceres parts desenvolupats fora dels EUA que implementen les especificacions del JCE i no son subjectes a restriccions legals. Un exemple seria BOUNCYCASTLE, que serà la llibreria que farem servir al nostre projecte, i en especial la implementació que fa sobre l'algorisme AES.

6.1 Anàlisi sobre l'operativa d'encriptació i desencriptació d'arxius

Els sistema que es seguirà per realitzar l'operativa d'enviament d'arxius a la BBDD restarà composat per dos funcionalitats bàsiques:

1) Creació d'un canal de comunicacions segur entre el client i el server mitjançant comunicacions encriptades SSL (https).
2) Mitjançant Applets propietat del server i signats per una CA de confiança, accedirem prèvia acceptació de l'applet pel client, a l'arxiu objecte d'encriptació a nivell local i realitzarem un procés de codificació, utilitzant principalment la llibreria OpenSource BouncyCastle.

D'aquesta manera utilitzarem un xifrat doble que atorgarà un grau mol elevat de seguretat a la transmissió de dades entre usuari i server.

Endinsant-nos a la codificació, passarem a analitzar els mètodes cridats per tal de realitzar l'operativa d'encriptació i desencriptació:

```
public static byte [] decrypt(byte [] cipherText, byte [] key) {

     try {
         IvParameterSpec ivSpec = new IvParameterSpec(iv);
         SecretKey secretKey = new SecretKeySpec(key, "AES");
         Cipher aes = Cipher.getInstance("AES/CBC/PKCS5Padding");
         aes.init(Cipher.DECRYPT_MODE, secretKey, ivSpec);
         byte[] plainText = aes.doFinal(cipherText);
         return plainText;
     } catch(Exception e) {
         errorExit("Decryption failed");
     }
     return null;
}

public static byte [] encrypt(byte [] plainText, byte [] key) {

     try {
         IvParameterSpec ivSpec = new IvParameterSpec(iv);
         SecretKey secretKey = new SecretKeySpec(key, "AES");
         Cipher aes = Cipher.getInstance("AES/CBC/PKCS5Padding");
         aes.init(Cipher.ENCRYPT_MODE, secretKey, ivSpec);
         byte[] cipherText = aes.doFinal(plainText);
         return cipherText;
     } catch (Exception e) {
         e.printStackTrace();
         errorExit("Error in encryption:" + e.getMessage());
     }
     return null;
}
```

Podem veure com creem un objecte de tipus SecretKey passant-li com a paràmetre la AES key que serà la que seleccionarà l'usuari amb el Jfile de Swing. Posteriorment creem un Cipher del tipus AES/CBC/PKCS5Padding, que és inicialitzat en modes ENCRYPT o DECRYPT segons el mètode cridat.

Per últim desem les dades finals a un array de bytes que, posteriorment i fent ús de FileOutpuStream, convertim a un arxiu primitiu o un encriptat amb terminació .aes.

7 – Valoració econòmica del projecte

El projecte ha representat una inversió molt extensa en hores de treball que difícilment es podrien extrapolar al mon comercial dins l'àmbit d'una empresa de serveis o a l'activitat professional d'un programador Freelance. Podríem parlar al voltant d'unes 3 hores diàries durant un període de 100 dies, o sigui, 300 hores valorades en uns 50 euros, que representen un total de 15.000 €.

Aquesta quantitat no és orientativa de la inversió que pot representar per a una empresa desenvolupar un projecte com el que hem realitzat a aquest TFC. La formació i adquisició de coneixements per part del programador ha estat gradual des del començament fins a la finalització del projecte. per tant seria obvi pensar que un programador senior en la tecnologia J2EE hauria realitzat tot el desenvolupament en un període de temps molt inferior al que hem invertit nosaltres.

8 – Conclusions

L'abast de les tecnologies a utilitzar ha estat molt ampli i de caire força diferenciat. Des de programació bàsica java de funcionalitats a servlets, passant per la creació de beans amb funcions molt definides, jsp's o la creació d'applets per accedir a la màquina de client, amb tota la complexitat que això comporta.
El concepte de la “reutilització de codi” s'ha experimentat en tota la seva dimensió a l'hora de implementar funcionalitats com: Upload, Download d'arxius, visualització d'arxius a BBDD o directament, tot el procés d'encriptació.

9 – Línies de desenvolupament futur

- L'aplicació web s'hauria de complementar amb funcionalitats d'encriptació més variades.

- Nivells de complexitat o algorismes a escollir.

- La identitat del client s'ha d'assegurar fiablement. Passarem d'una encriptació de clau simètrica a una altre d'asimètrica sense que representi un augment de complexitat al xifrat. D'aquesta manera obtenim una garantia addicional de l'origen de les dades enviades.

- Una vegada realitzada la connexió amb el client, un applet ha de realitzar una auditoria de seguretat a la màquina client per tal de detectar possible malware: keyloggers, troians, backdoors, virus en general, etc.

- Un applet ha de permetre signar gràficament amb un llapis electrònic i afegir com a . jpeg la signatura a l'arxiu encriptat.

- Opcionalment es poden afegir d'altres mostres d'autenticitat, com poden ésser:

    - Signatures de veu amb arxius .wav
    - Empremta dactilar amb dispositiu adient.
    - Enviament d'una copia de DNI digital amb els arxius pujats al servidor.

10 – Bibliografia

Encriptació en JAVA
http://www.javaworld.com/javaworld/javaqa/2003-05/01-qa-0509-jcrypt.html
http://java.sun.com/developer/JDCTechTips/2004/tt0116.html
http://www.bouncycastle.org/
http://cephas.net/blog/2004/04/01/pgp_encryption_using_bouncy_castle.html
http://www.aviransplace.com/index.php/archives/2004/10/12/using-rsa-encryption-with-java/
http://www.artima.com/forums/flat.jsp?forum=121&thread=116230
http://www.aviransplace.com/index.php/archives/2004/10/12/using-rsa-encryption-with-java/3/
http://www.bouncycastle.org/specifications.html

J2EE
http://java.sun.com/j2ee/1.4/docs/tutorial/doc/
http://www.theserverside.com/
http://www.programacion.com/java/tutoriales/J2EE/
http://www.webopedia.com/TERM/J/J2EE.html
http://www.mundotutoriales.com/tutoriales_j2ee-mdpal14064.htm

Applets
http://javaboutique.internet.com/
http://www.freewarejava.com/applets/index.shtml< br>http://www.javafile.com/

Beans
http://www.javazoom.net/jzservlets/download4j/download4j.html
http://www.javazoom.net/jzservlets/uploadbean/uploadbean.html

Css i HTML
http://www.w3schools.com/
http://www.desarrolloweb.com/
http://www.programacionweb.net/
http://webdesign.about.com/