Proveïment de serveis segurs.

Jordi Pua Puñet
ETIS

Carlos Ares Angulo

13 de juny de 2008

RESUM

Aquest projecte és una mostra de les possibilitats de dotar de seguretat a les aplicacions web i d'arquitectura Client/Servidor basades en Java.

Els requeriments funcionals del projecte són els següents:

- El proveïdor de serveis espera peticions de serveis.
- La informació intercanviada entre el client, el proveïdor de serveis i el proveïdor d'identitat va signada i viatja de forma segura.
- Tant el proveïdor de serveis com el proveïdor d'identitat disposen de certificats digitals (autosignats) i el client ha de disposar d'un certificat digital vàlid i reconegut per una autoritat certificadora de confiança.
- El proveïdor d'identitat disposa d'una base de dades amb les dades dels serveis, clients i permisos i per tant es qui estableix si s'ha de resoldre positiva o negativament la petició.

L'objectiu final del projecte és la de disposar d'un exemple de com utilitzant les signatures digitals i els certificats digitals podem obtenir sistemes amb actors distribuïts en diferents actors i que en podem garantir la identitat i la seguretat de les dades en un present en el que les aplicacions i els serveis web han agafat molta embranzida i en aquest escenari és fàcilment previsible que en un futur immediat seran encara més presents i proliferes.

INDEX

1. INTRODUCCIÓ.

1.1. Justificació del TFC: punt de partida i aportació.

L'objectiu del projecte és el d'aconseguir una implementació que ens permeti observar la possibilitat i sobretot la necessitat que les aplicacions distribuïdes / web disposin d'un sistema robust per tal de poder garantir tant les identitats de tots els implicats en qualsevol intercanvi d'informació i també que aquesta informació que s'intercanvien estigui protegida es a dir que la informació que arriba és la que s'ha enviat.

En aquest projecte hi trobem tres actors, dos dels quals disposen d'una implementació i una lògica (el proveïdor de serveis i el proveïdor d'identitat) i el tercer actor podríem dir que només disposa de lògica (el client), tot i que en principi sembla que no tindríem que fer referència al client crec que aquest fet ens ajudarà a entendre millor els diferents processos que es duen a terme en els sistemes d'aquest projecte.

Un cop un usuari visita una pagina web des d'un navegador el proveïdor de serveis li fa arribar un applet (i la lògica d'aquest és la que anomenarem client dins del nostre sistema). l'usuari farà una petició de servei de forma anònima al proveïdor de serveis i aquest li retorna aquestes dades més un identificador de sessió en un missatge signat.

Quant el client rep les dades les tornar a signar amb la seva signatura i les fa arribar al proveïdor d'identitat que es qui s'encarregarà de verificar la identitat tant del proveïdor de serveis com del client i també de que les dades han arribat correctament. Finalment el proveïdor d'identitat verifica en la seva base de dades que l'usuari tingui permisos sobre el servei sol·licitat i d'aquesta manera pot construir el missatge de resposta i signar-lo abans d'enviar-lo al client.

Quant el client rep la resposta i simplement la reenvia al proveïdor de serveis, aquest comprova les signatures i concedeix o denega el servei en funció de les dades que li ha facilitat el proveïdor d'identitat i comprovant sempre que la concessió del servei no estigui caducada.

De la lectura dels paràgrafs anteriors haurem observat que en cap moment el proveïdor de serveis coneix la identitat del "client", es a dir per al nostre proveïdor de serveis es tractarà sempre de "clients" anònims, ni coneixerà la seva identitat ni enregistrarà cap dada que en faci referència, aquestes funcions queden delegades sobre el proveïdor d'identitat que és qui coneix la identitat tant del "client" com del proveïdor de serveis i per tant es qui traves de la comprovació i validació tant de signatures com de certificats pot garantir la identitat dels altres dos actors implicats.

1.2. Objectius del TFC

L'objectiu principal del projecte és el d'aprofundir sobre els coneixements de les tecnologies de seguretat que podem utilitzar o aplicar mitjançant Java en entorns distribuïts.

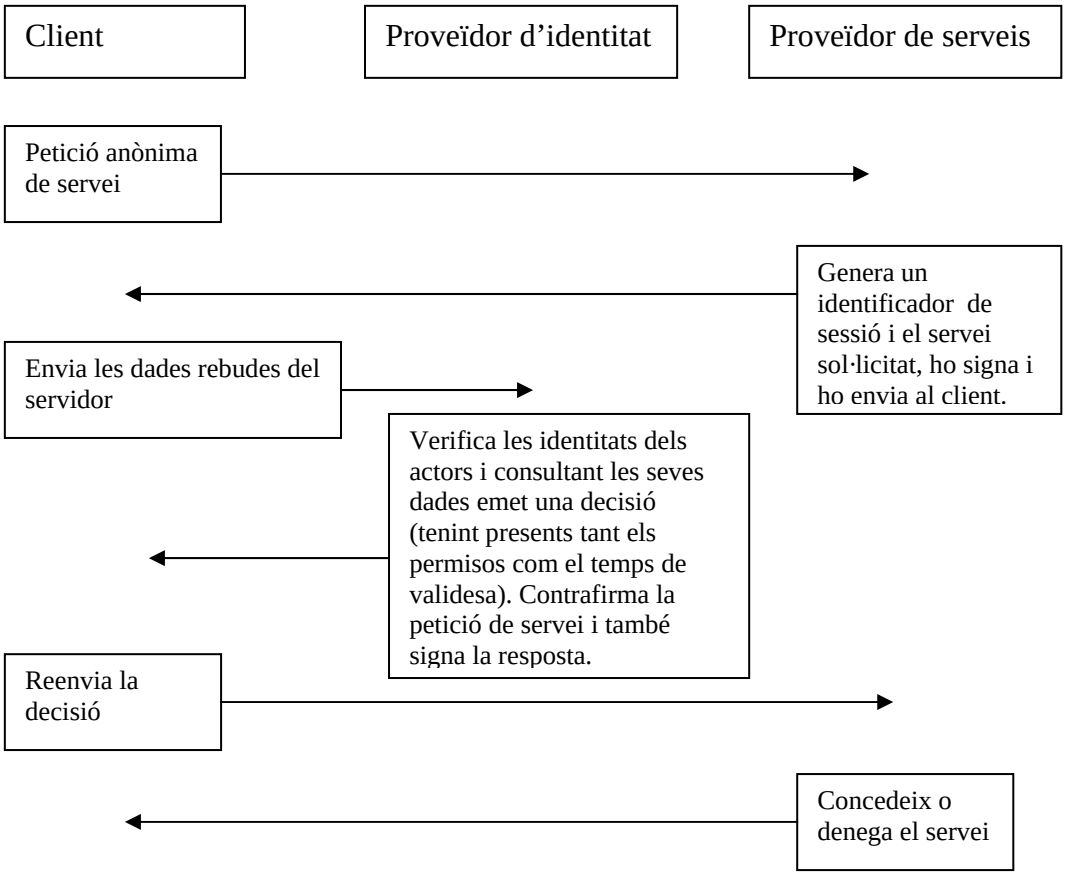
Pel que fa als objectius més concretament explicitats en la planificació d'aquest projecte són:

- Implementar la web del proveïdor d'identitat.
- Implementació del client.
- Implementació del proveïdor de serveis.
- Dissenyar i implementar les diferents interfícies que utilitzarem.
- Implementar els diferents sistemes de certificació digital que usarem.
- Garantir la seguretat del sistema i l'autenticitat tant de les comunicacions com dels comunicants. Per aconseguir aquests objectius utilitzarem certificats digitals (x.509), signatures digitals i transmissions signades. Els certificats ens ajudaran a garantir que cada actor és qui diu ser, les signatures digitals ens ajudaran a garantir que la informació que enviem és la que es rebrà i finalment les transmissions signades ens ajudaran a garantir que aquestes és facin amb seguretat de que no hi haurà sostracció de dades durant aquestes transmissions.
- Establir els sistemes i mètodes de comunicació segura entre els diferents actors.
- I finalment aconseguir un sistema de proveïment de serveis segurs amb garanties d'identitat tant del client com del proveïdor.
- Generar un conjunt de documents que conformin la totalitat d'aquest treball, des dels diferents codis fonts que el formaran, passant pels documents complementaris per a poder comprendre aquests codis font, fins a les diferents descripcions en llenguatge natural per tal de poder seguir l'evolució de totes les etapes del projecte.
- Personalment l'objectiu principal és el d'aconseguir transmetre que he estat capaç d'obtenir i consolidar tota una sèrie de coneixements al llarg de la carrera, concretament m'he centrat en una petita part de la seguretat en un entorn de xarxa ja que considero que és i ha de ser la base del desenvolupament de programari tant del present com del futur dins de qualsevol xarxa de telecomunicacions.

1.3. Enfocament i mètode seguit.

S'espera obtenir diferents components de programari que en conjunt formin una arquitectura Client/Servidor que ens permeti realitzar peticions de serveis i oferir aquests serveis de forma segura, es a dir tant el client com el servidor han de poder estar segurs que l'altre actor és qui diu ser, i per aconseguir aquesta verificació utilitzarem un tercer actor que ens proveirà i validarà els certificats digitals necessaris.

Una idea més esquemàtica dels resultats esperats podria ser la següent: un client vol disposar d'un servei específic i que li ofereixi un servidor concret, doncs be el servidor al que li sol·licitarà el servei, ha de poder tenir garanties de que el client és qui diu ser i que a més a més disposa dels permisos/privilegis suficients per a rebre aquest servei, i per una altra banda el client ha de poder tenir garanties que rep el servei sol·licitat i que aquest li facilita el proveïdor al qual li ha sol·licitat. Doncs be tots aquesta generació, intercanvi i comprovació de identitats i permisos la realitzarà un tercer actor que serà un proveïdor d'identitat.



1.4. Planificació.

En aquest punt de la memòria presentarem l'estructura del treball, la descomposició de tasques i la planificació temporal de la realització del treball.

L'estructura d'aquest Treball Fi de Carrera estarà dividida en les següents fases:

- ✓ Elaboració del Pla de Treball.
- ✓ Cerca d'informació sobre algorismes de generació i validació de certificats digitals.
- ✓ Cerca d'informació sobre les llibreries de les que disposa Java per a poder implementar els diferents algorismes escollits per a la implementació del treball.
- ✓ Realitzar un especificació bàsicament textual de les aplicacions a desenvolupar en aquest treball.
- ✓ Dissenyar tant el programa client com el programa servidor i del programa proveïdor d'identitat, a nivell de processos, comunicació i interacció entre ells.
- ✓ Dissenyar les diferents interfícies gràfiques d'usuari per a tots tres actors.

- ✓ Revisar i contrastar els dissenys dels programes i les interfícies per tal de completar les diferents opcions, missatges i altres components que tinguin que presentar les implementacions finals.

- ✓ Desenvolupament i implementació del programari tant client, servidor i proveïdor d'identitat.

- ✓ Implementació de la comunicació i interacció de les tres aplicacions.

- ✓ Elaboració de la memòria: disseny dels programes.

- ✓ Elaboració de la memòria: manuals d'usuari, jocs de proves, incidències i conclusions.

- ✓ Realitzar la presentació del projecte amb programari del tipus "PowerPoint".

I la planificació temporal de la realització del treball és la següent:

- ♦ **Període inici semestre a 6 de març**

  Elaboració del Pla de Treball.

- ♦ **Període 7 de març a 20 de març**

  Pla de treball revisat, fonaments i arquitectura.

  Cerca d'informació sobre algorismes de generació i validació de certificats digitals.

  Cerca d'informació sobre les llibreries de les que disposa Java per a poder implementar els diferents algorismes escollits per a la implementació del treball.

  Realitzar un especificació bàsicament textual de les aplicacions a desenvolupar en aquest treball.

- ♦ **Període de 21 de març a 11 d'abril**

  Disseny del producte.

  Dissenyar tant el programa client com el programa servidor i del programa proveïdor d'identitat, a nivell de processos, comunicació i interacció entre ells.

  Dissenyar la interfície gràfica d'usuari tant del programa client com del programa proveïdor d'identitat i la interfície en mode consola del programa servidor.

Revisar i contrastar els dissenys dels programes i les interfícies per tal de completar les diferents opcions, missatges i altres components que tinguin que presentar les implementacions finals.

- **Període de 12 d'abril a 15 de maig**

  Implementació parcial del producte.

  Desenvolupament i implementació del programari tant client, servidor i proveïdor d'identitat.

  Implementació de la comunicació i interacció de les tres aplicacions.

- **Període de 16 de maig a 13 de juny**

  Memòria.

  Elaboració de la memòria: disseny dels programes.

  Elaboració de la memòria: manuals d'usuari, jocs de proves, incidències i conclusions.

  Realitzar la presentació del projecte amb programari del tipus "PowerPoint".

- **Període de 14 de juny a 19 de juny**

  Presentació.

1.5. Productes obtinguts.

El desenvolupament d'aquest treball ha generat els següents productes:

Document de planificació: Document on s'especifiquen els objectius generals i específics del projecte. Es detalla un pla de treball concret amb dates fixades i s'informa de les interdependències entre tasques.

Document de l'anàlisi i disseny: Document on s'especifica el punt de partida del projecte, el context, el domini, els requisits i els processos que ha de complir el programari i per la part de disseny s'especifica com s'ha d'implementar les necessitats detectades durant la fase d'anàlisi.

Implementació: Conjunt de fitxers que són el producte final del projecte:

- Fitxers .jar i . lo que sigui executable
- El codi font de l'aplicació.
- Documentació detallada i normalitzada en format javadoc.

- Els fitxers on s'especifiquen les instruccions utilitzades per a generar els fitxers del primer apartat d'aquesta llista i també les instruccions OpenSSL que he utilitzat per al desenvolupament del projecte.
- Els scripts de creació de les taules i la inserció de dades necessàries a la base de dades a utilitzar.

Document de la memòria: Present document que inclou tots els anteriors menys l'apartat d'implementació.

1.6. Breu descripció dels altres capítols de la memòria.

El capítol 2 detalla la informació obtinguda a la fase d'anàlisi i una primera especificació que ens aproparà a la lògica i operativa del sistema.

El capítol 3 disseny de la lògica del client, disseny del proveïdor de serveis i disseny del proveïdor d'identitat.

El capítol 4 desenvolupament i implementació del programari que conforma el sistema del projecte. Implementació de la comunicació i interacció d'aquestes aplicacions.

2. FASE D'ANÀLISI I ESPECIFICACIÓ.

2.1. Cerca d'informació sobre algorismes de generació i validació de certificats digitals.

En el nostre treball utilitzarem OpenSSL, que ens permetrà generar i manipular certificats digitals i que també ens permetrà crear una autoritat certificadora, que més tard utilitzarem en la transmissió de dades per tal de garantir que les dades rebudes son les que ens van enviar es a dir ens permet garantir la no manipulació de les dades durant la seva transmissió, i finalment ens permetrà que el nostre proveïdor d'identitat pugui validar el certificat digital del client tant pel que fa la validesa de les seves dates com a la consulta de la llista de revocació que també generarem amb OpenSSL.

2.2. Cerca d'informació sobre les llibreries de les que disposa Java per a poder implementar els diferents algorismes escollits per a la implementació del treball.

En la nostra implementació hi tindríem fins a 3 algorismes cabdals (certificats, signatures, intercanvi de dades).

En el cas dels certificats digitals ja ha estat explicat en l'apartat anterior. Pel que fa a la signatura digital podem dir que:

En el nostre treball utilitzarem les llibreries externes anomenades Bouncy Castle, que ens permetrà generar i manipular signatures de tipus CMS/PKCS#7, amb les quals podrem signar les dades que vulguem transmetre, també podrem signar les pròpies transmissions, i finalment podrem validar les signatures de les dades rebudes. La signatura digital ens permet autenticar la font de les dades transmeses i també la no "lectura" d'aquestes dades per altres actors que no siguin els destinataris. I també ens permetran validar el certificat que ens presentarà el client.

En el cas de les comunicacions en el nostres projecte podem dir que:
Dins del nostre projecte necessitarem realitzar múltiples transmissions de diferents tipus de dades i davant d'aquesta necessitat el que farem serà utilitzar la tecnologia SOAP (utilitzarem una connexió HTTP POST convencional) que ens proporcionarà un mecanisme estàndard per a empaquetar les dades i deixar-les preparades per a la seva transmissió, es a dir ens facilitarà la tasca de transmetre tipus de dades complexes i també ens facilitarà la seva signatura digital.

2.3. Realitzar una especificació bàsicament textual de les aplicacions a desenvolupar en aquest treball.

En el nostre projecte hi apareixen tres actors principals:

Client:

- Escriu el nom del servei desitjat.
- Enviament de la petició de servei al proveïdor de forma anònima usant un canal segur.

- Manipulació del missatge signat rebut des del proveïdor de serveis que li permetrà contactar amb el proveïdor d'identitat extern.
- Transmissió del missatge de petició signat i també del seu certificat al proveïdor d'identitat, juntament amb les dades rebudes des del proveïdor de servei.
- Rebuda de la resposta des del proveïdor d'identitat i retransmissió de la mateixa al proveïdor de serveis.
- I depenent de la decisió presa pel proveïdor d'identitat el proveïdor de serveis ens facilitarà el servei o no.

Proveïdor de identitat:

Aplicació d'administració:

- Permetrà un accés amb nom d'usuari i contrasenya a la seva part d'administració.
- Permetrà donar d'alta usuaris, serveis i permisos dins de la seva part d'administració per tal de després poder prendre una decisió sobre les peticions rebudes.

Servei web:

- Rebrà les dades signades i el certificat del client per tal de que les comprovi i emeti una autorització o una denegació de servei, tenint en compte el client, el proveïdor de serveis i el servei i la duració de l'autorització.
- Revisió de les diferents signatures rebudes i també els permisos del client en referència al servei sol·licitat, i finalment emet la seva resposta i ho signa amb el seu certificat, també retorna la petició signada pel proveïdor de serveis i amb una contrafirma seva.

Proveïdor de serveis:

- Permet donar d'alta nous serveis (que en realitat no seran operatius fins que l'administrador del proveïdor d'identitat el doni d'alta a la seva base de dades).
- Rep peticions anònimes a les quals respon junt amb el component que permetrà als clients contactar amb el proveïdor d'identitat per tal de poder continuar amb el procés.
- Rep des del client la resposta emesa pel proveïdor d'identitat i depenent de la decisió presa pel proveïdor d'identitat doncs concedirà o denegarà el servei sol·licitat pel client.

3. DISSENY DE LA LÒGICA DEL CLIENT, DISSENY DEL PROVEÏDOR DE SERVEIS I DEL PROVEÏDOR D'IDENTITAT.

Dissenyar tant la lògica de l'actor client (el que anomenem client no és una aplicació ni te implementació pròpia, es tracta d'un applet enviat pel proveïdor de serveis al navegador web que li sol·licita un servei, però per tal de poder observar millor la dinàmica de funcionament del sistema tractarem per separat la lògica pròpia del client i la del proveïdor de serveis), com el programa servidor i del programa proveïdor d'identitat, a nivell de processos, comunicació i interacció entre ells.

El proveïdor d'identitat un cop li arriben les dades de la petició realitzada pel client realitza una consulta a la seva base de dades per tal de poder saber si el client te permís sobre el servei sol·licitat i en el servidor sol·licitat, també n'obté la duració del propi permís per al servei.

Primer veurem el diversos fluxos de dades entre els diferents actors:

Del "client" al proveïdor de serveis:

El "client" envia el nom del servei sol·licitat.

Del proveïdor de serveis al "client":

El proveïdor de serveis li fa arribar el nom del servei sol·licitat i l'identificador de sessió generat, tot això signat.

Del "client" al proveïdor d'identitat:

El "client" li fa arribar al proveïdor d'identitat el nom del servei sol·licitat més l'identificador de sessió tot signat pel proveïdor de serveis i després signat pel "client".

Del proveïdor d'identitat al "client":

El proveïdor d'identitat li fa arribar al client el nom del servei sol·licitat i l'identificador de sessió tot això signat pel proveïdor de serveis i amb una contrasignatura del propi proveïdor d'identitat i també li afegeix l'acceptació o denegació del servei i la durada de l'autorització de servei i el proveïdor d'identitat ho torna a signar tot.

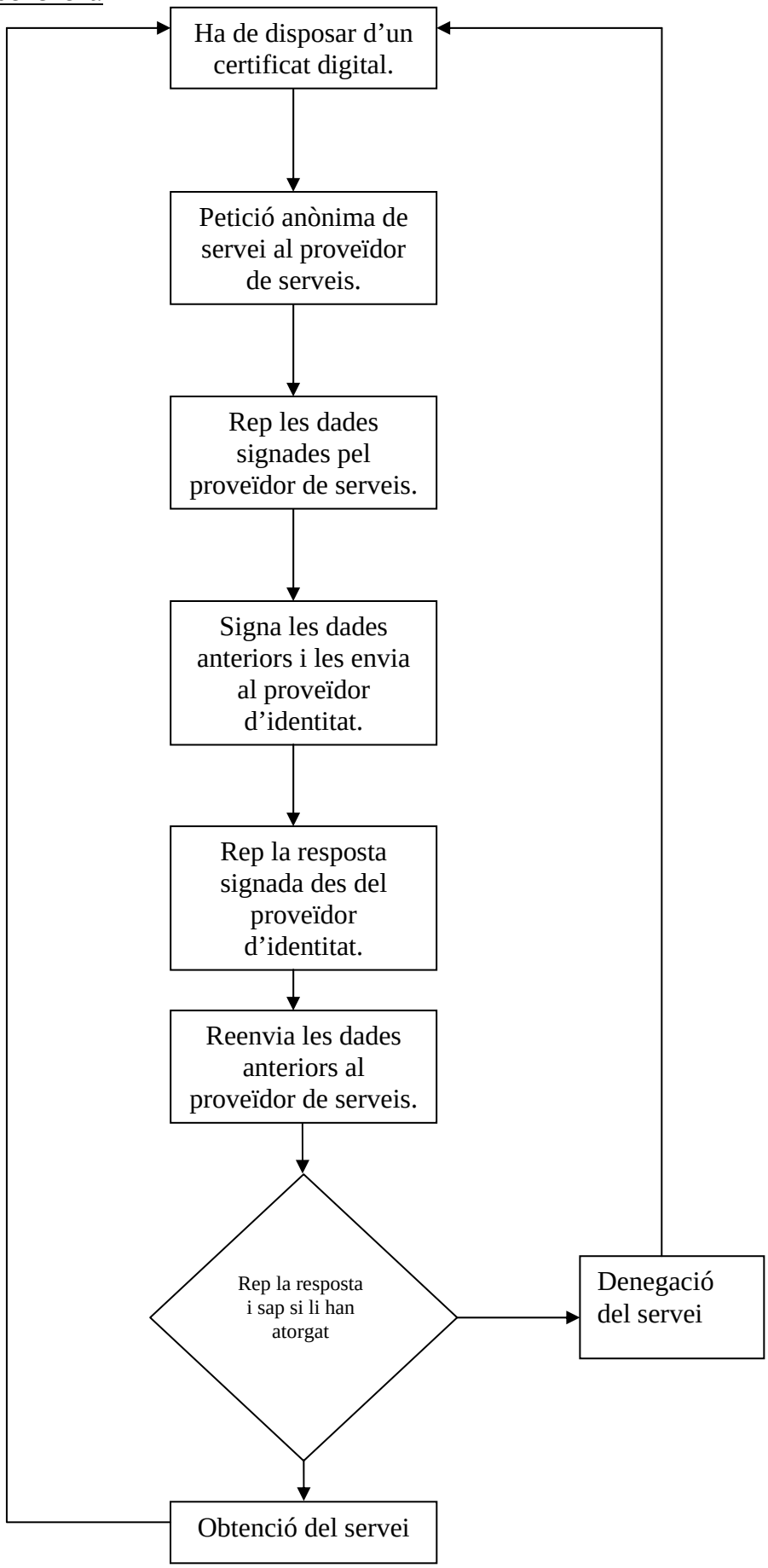
Del "client" al proveïdor de servei:

El "client" li reenvia les dades de l'apartat anterior al proveïdor de serveis.

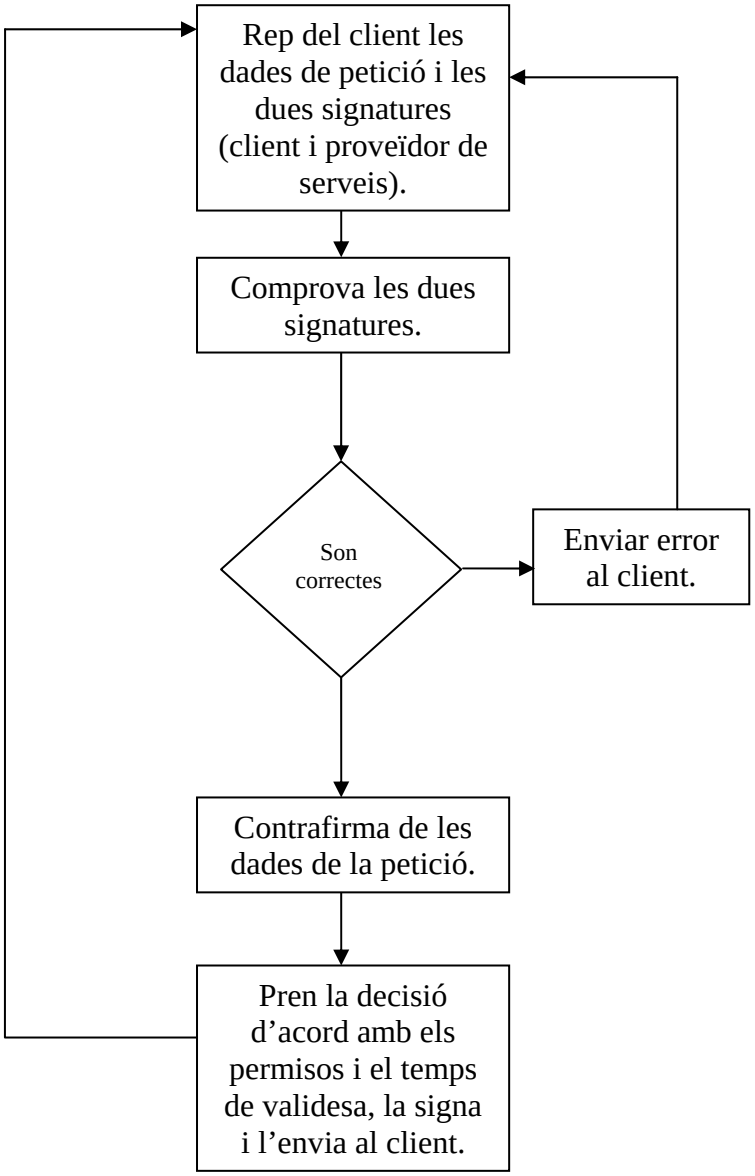
Del proveïdor de servei al "client":

El proveïdor de servei li envia al client el nom del servei sol·licitat i si li han atorgat el servei o no.

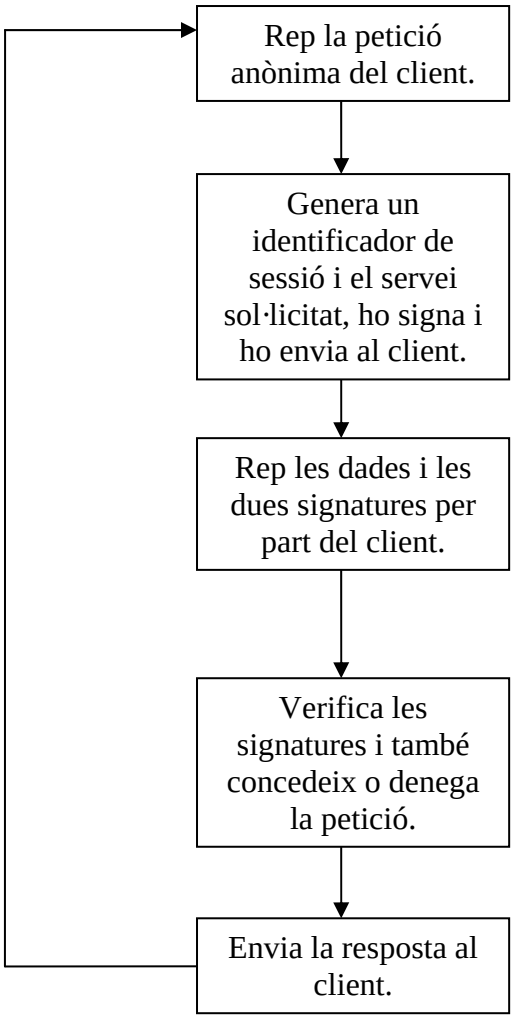
3.1. Lògica del client.



3.2. Programa Proveïdor d'identitat.

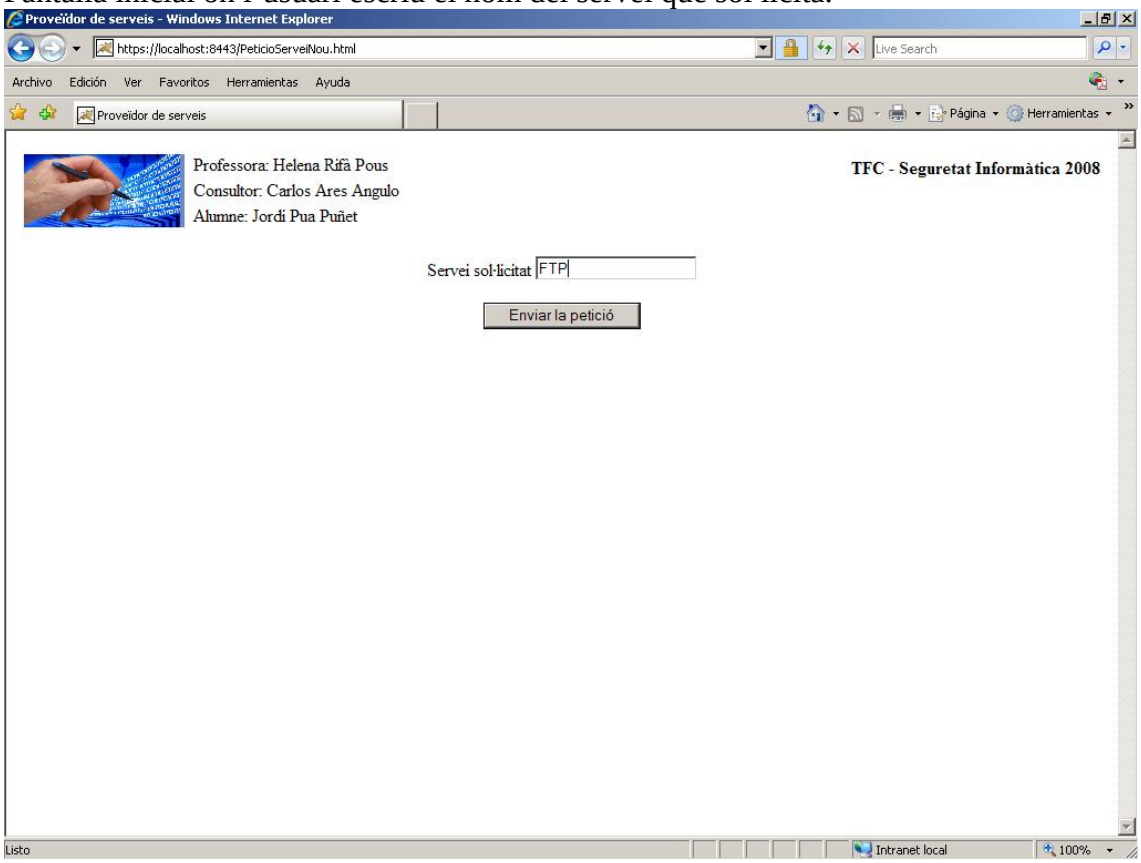


3.3. Programa Proveïdor de serveis.

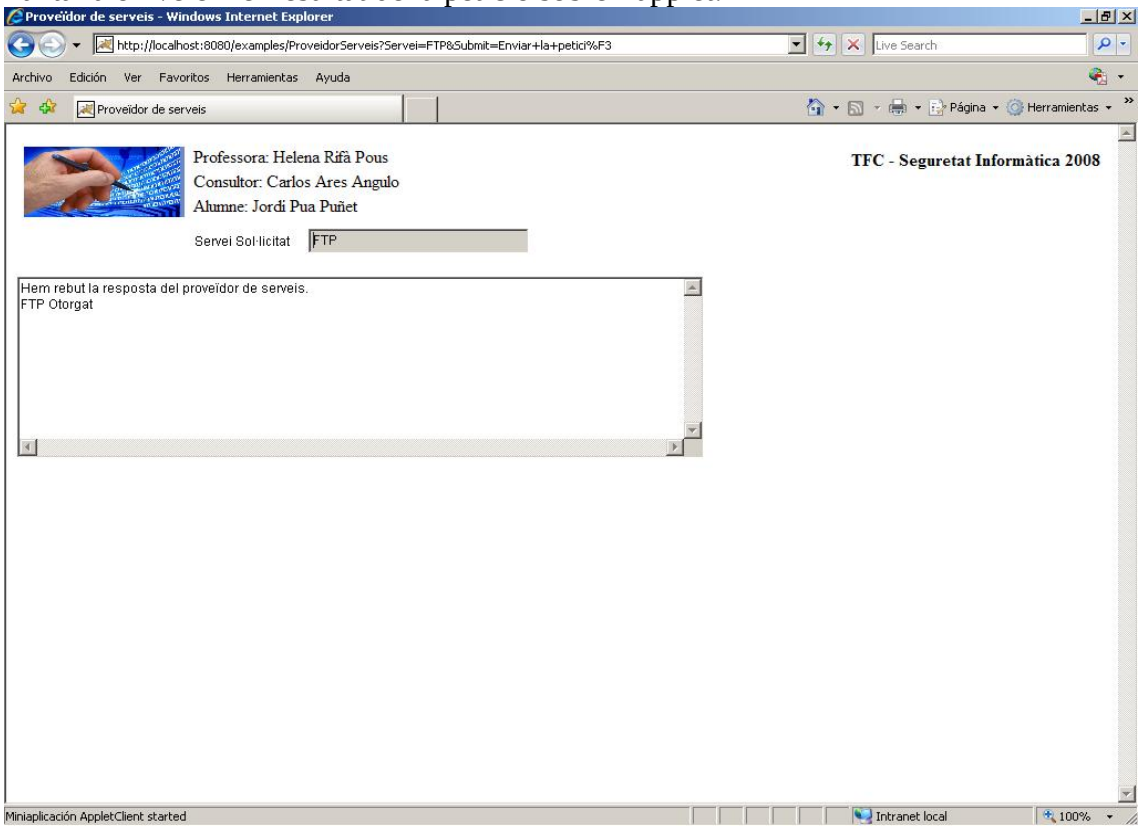


3.4. Disseny interfície del “Client”/Proveïdor de serveis.

Pantalla inicial on l’usuari escriu el nom del servei que sol·licita.

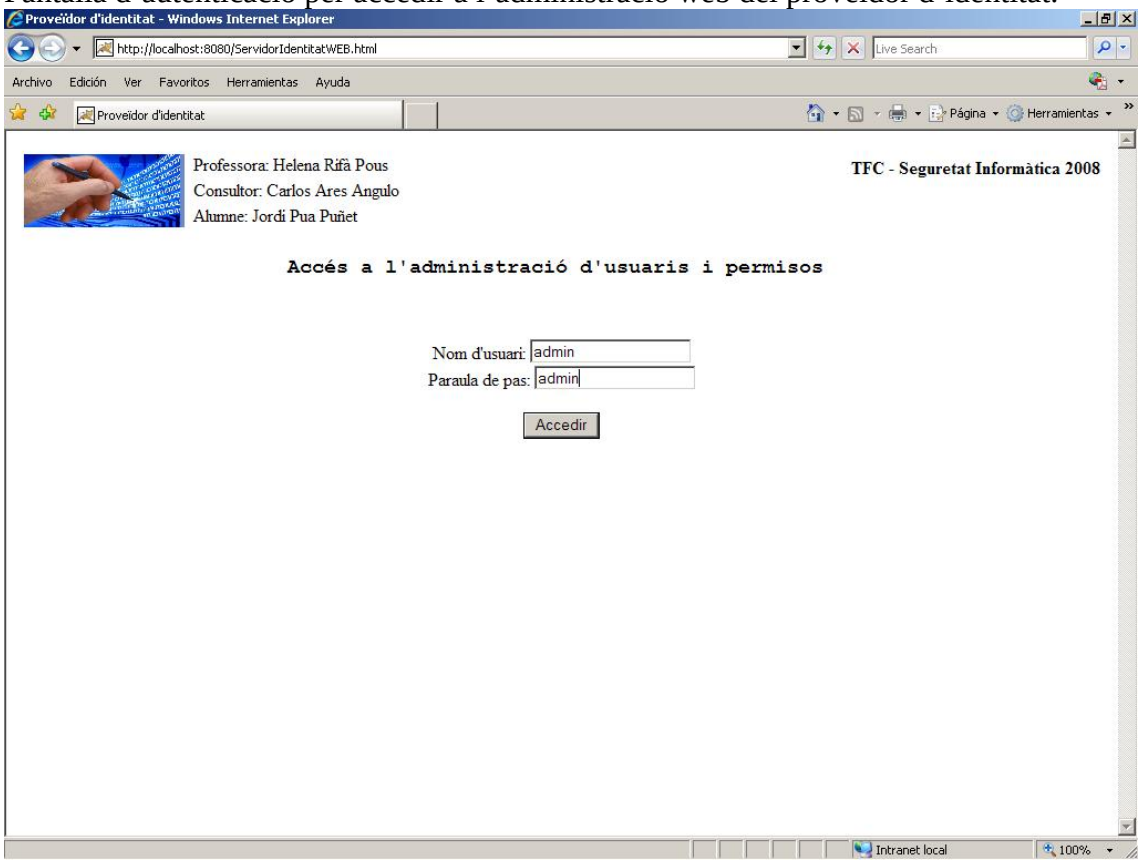


Pantalla on veiem el resultat de la petició sobre l’applet.

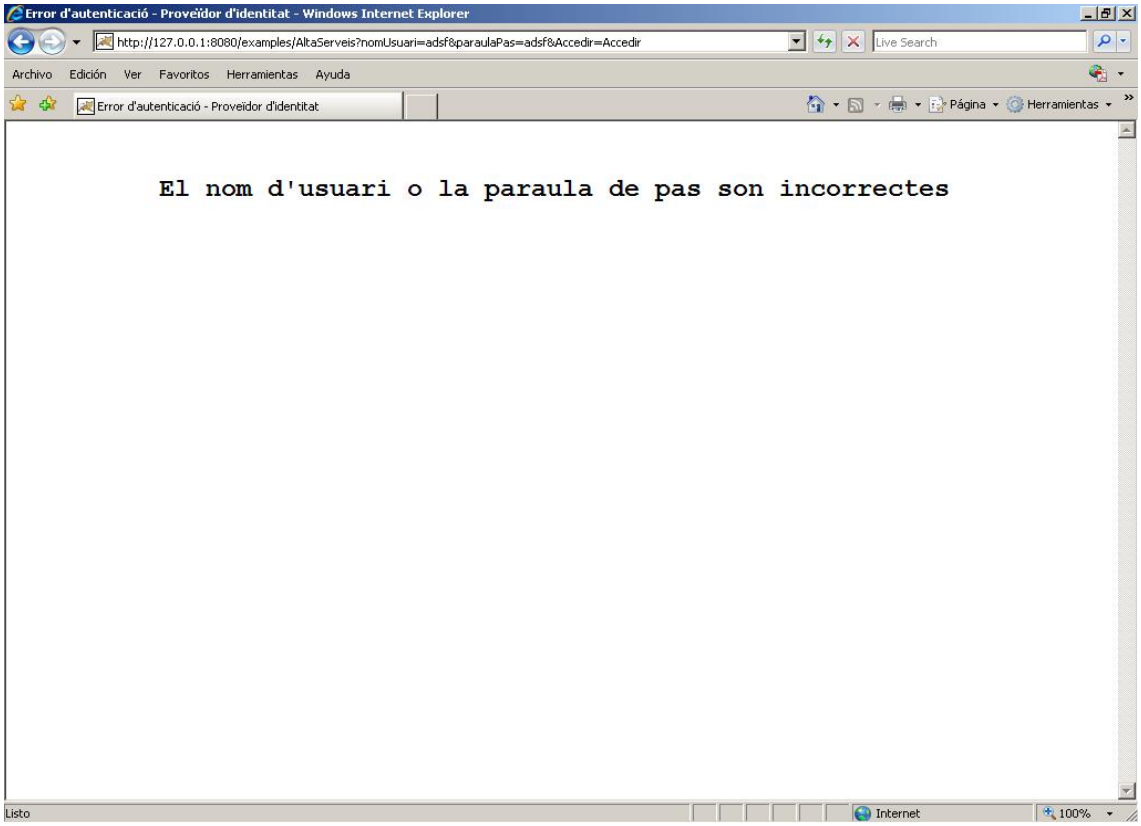


3.5. Disseny interfície del Proveïdor d'identitat.

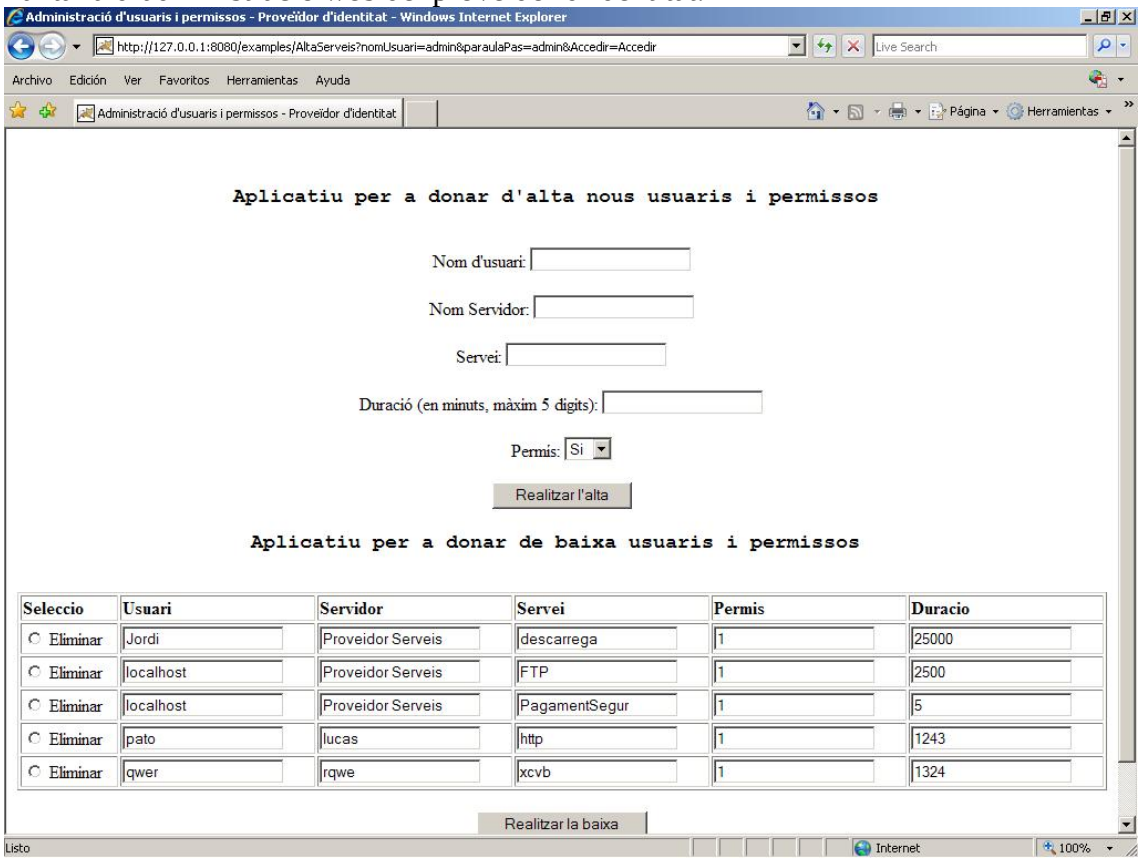
Pantalla d'autenticació per accedir a l'administració web del proveïdor d'identitat.



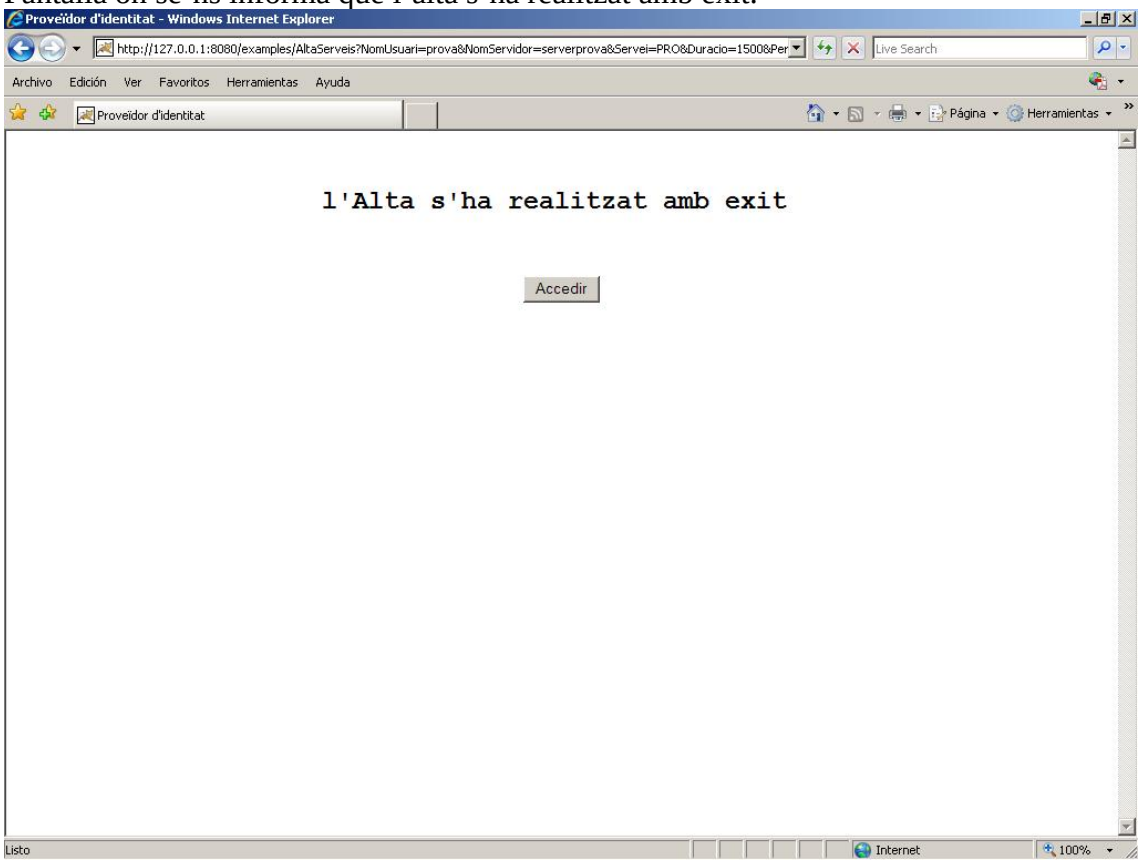
Pantalla d'error d'autenticació.



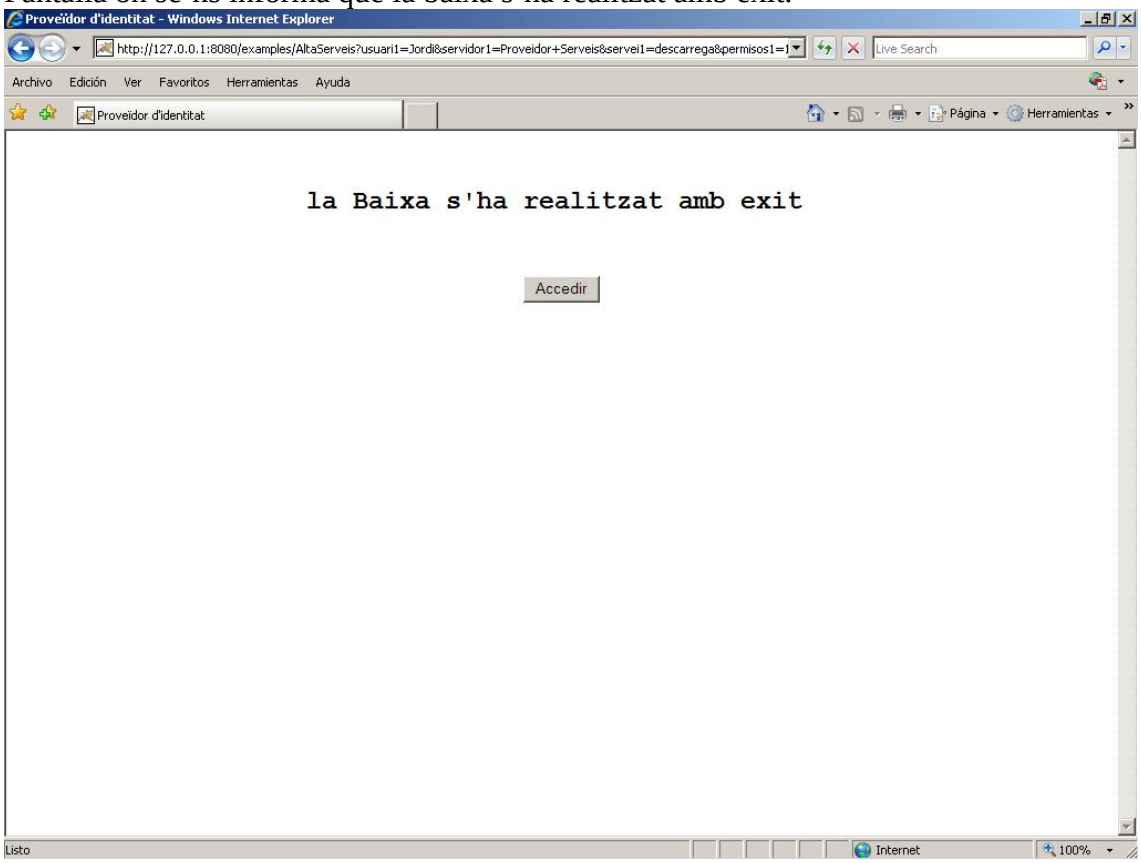
Pantalla d'administració web del proveïdor d'identitat.



Pantalla on se'ns informa que l'alta s'ha realitzat amb èxit.



Pantalla on se'ns informa que la baixa s'ha realitzat amb èxit.



3.6. Disseny de la base de dades del proveïdor d'identitat.

```
create database DBIdentitat;
use DBIdentitat;
drop table if exists ControlAcces;
create table ControlAcces(
Usuari varchar(64) not null,
Servidor varchar(64) not null,
Servei varchar(64) not null,
Permisos varchar(1),
Duracio varchar(5),
primary key(Usuari, Servidor, Servei)
)type=InnoDB;
drop table if exists ControlAdministrador;
create table ControlAdministrador(
NomUsuari varchar(64) not null,
ParaulaClau varchar(8) not null,
primary key(NomUsuari)
)type=InnoDB;
```

La base de dades del proveïdor d'identitat consta de dues taules:

Una taula anomenada ControlAdministrador en la que podrem trobar dos camps: el primer anomenat NomUsuari i el segon ParaulaClau, aquests dos camps s'utilitzen

per a guardar les parelles de noms d'usuari i paraules de pas que permetran accedir a la part web de l'administració de la base de dades del proveïdor d'identitat.

La segona taula s'anomena ControlAcces i consta de cinc camps:

Usuari: Identificador del client.
Servidor: Identificador del servidor.
Servei: Nom del servei sol·licitat.
Permisos: val 1 si es té permís i 0 si no es té permís.
Duracio: informa de la duració del permís atorgat expressat en minuts amb un màxim de 5 dígits.

4. DESENVOLUPAMENT I IMPLEMENTACIÓ DEL PROGRAMARI QUE CONFORMA EL SISTEMA DEL PROJECTE.

4.1. Fitxers amb el codi font i altres fitxers relacionats amb la implementació.

Els fitxers corresponents a aquest apartat s'entreguen fora d'aquesta memòria.

4.2. Eines utilitzades en el desenvolupament.

Java

J2SDK-1.6.0_06.
J2RE1.6.0_06.

Un cop realitzades les instal·lacions de Java hem d'assegurar-nos que tant variable de sistema Classpath conté el camí de la carpeta lib de la instal·lació i que la variable de sistema Path conté tant el camí de la carpeta lib com de la carpeta bin de la instal·lació.

Servidor web

Apache-SSL 1.3.27.
Apache Tomcat 6.0.

Un cop realitzades les instal·lacions hem de verificar que la variable de sistema Path conté el camí de la carpeta lib de la instal·lació. També hem de modificar el fitxer web.xml de Tomcat per tal d'afegir els noms i classes dels nostres servlets (AltaServeis, ProveidorIdentitat i ProveidorServeis), per exemple:

```
<servlet>
  <servlet-name>AltaServeis</servlet-name>
  <servlet-class>AltaServeis</servlet-class>
</servlet>
```

També necessitem modificar el fitxer server.xml per tal de que el nostre Tomcat escolti tant el port 8080 com el port 8443 per a les pagines segures i també hem d'indicar-li a Tomcat que voldrem seguretat i autenticació per a les pagines segures.

```
<Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />

<Connector protocol="org.apache.coyote.http11.Http11Protocol"
        port="8443" minSpareThreads="5" maxSpareThreads="75"
        enableLookups="true" disableUploadTimeout="true"
        acceptCount="100"  maxThreads="200"
        scheme="https" secure="true" SSLEnabled="true"
        keystoreFile="D:/.keystore" keyAlias="applet" keystorePass="jpua08"
        clientAuth="false" sslProtocol="TLS"/>
```

Seguretat

OpenSSL 0.9.8d.

Després de realitzar la instal·lació ens hem d'assegurar que la variable de sistema Path contingui el camí de la carpeta bin de la instal·lació i també hem de modificar el fitxer openssl.cnf per tal de poder crear i utilitzar correctament la nostra autoritat certificadora de confiança, donat que les modificacions son una mica extenses adjunto el fitxer openssl.cnf modificat al conjunt de fitxers del meu projecte.

Bouncy Castle 1.39.

Ens hem de descarregar els fitxers que conformen el conjunt de llibreries i classes de la distribució Bouncy Castle i els copiarem a la carpeta lib de la instal·lació del JDK de Java, també tindrem que afegir la seva ubicació a la variable de sistema Classpath, per exemple:

C:\Archivos de programa\Java\jdk1.6.0_06\lib\bcprov-jdk16-139.jar;

Si volem deixar la instal·lació preparada per a un us més còmode des del codi font de Java podem afegir BouncyCastle com a proveïdor de seguretat dins del fitxer java.security

security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider

BBDD

MySQL 5.0.
MySQL Administrator 1.1.
MySQL QueryBrowser 1.1.
MySQL ConnectorJava 3.1.12.

Un cop realitzades les instal·lacions ens hem d'assegurar que la variable de sistema Path inclou el camí de la carpeta bin de la instal·lació, també hem de copiar el connector que ens hem descarregat a la carpeta lib de la instal·lació de Tomcat i hem d'incolure aquest camí a la variable de sistema Classpath. I finalment per tal de poder crear taules i utilitzar-les a les nostres aplicacions hem de crear-nos un usuari amb suficients permisos per a poder tenir accés a la nostra base de dades.

ANT 1.7.0

Un cop realitzada la instal·lació ens hem d'assegurar que la variable de sistema Path conté el camí de la carpeta bin de la instal·lació.

IDE

Eclipse 3.2.2.

4.3. Tecnologies utilitzades i presa de decisions.

En aquest apartat presentarem algunes de les tecnologies utilitzades en el desenvolupament i també explicitar les decisions preses durant el desenvolupament del projecte.

Per tal de poder realitzar aquest projecte hem utilitzat elements externs a la pròpia API de Java:

- OpenSSL per als aspectes relacionats amb els certificats digitals i també amb l'autoritat de certificadora de confiança.
- MySQL per als aspectes relacionats amb la base de dades que utilitza el proveïdor d'identitat.
- Bouncy Castle per a poder gestionar tots els aspectes relacionats amb les signatures i certificats digitals des del propi codi Java.
- Com a conseqüència de que part del desenvolupament es basa en serveis web i en servlets hem necessitat utilitzar el servidor web Apache i el servidor web Tomcat per als servlets.

Pel que fa a les signatures i certificats digitals he utilitzat OpenSSL per a generar els certificats autosignats i les signatures digitals tant del proveïdor de serveis com del proveïdor d'identitat. En el cas del certificat no autosignat que utilitza el "client" he optat per generar una autoritat certificadora també amb OpenSSL i la seva corresponent llista de revocació per tal de que els processos de comprovació i validació sobre aquest certificat del "client" puguin funcionar correctament (com ja s'ha comentat en apartats anteriors ens servirem de OpenSSL per a generar els certificats, signatures i l'autoritat certificadora a més a més de la corresponent llista de revocació, però les comprovacions i validacions dels mateixos ho farem des del codi font amb el suport de les llibreries Bouncy Castle).

Per tal de poder realitzar els processos de signatura, validació i comprovació que s'han de dur a terme des del codi Java dels diferents actors he usat les llibreries Bouncy Castle que ens permeten complir amb les especificacions presentades en l'enunciat d'aquest projecte. En relació també a les signatures realitzades amb les llibreries Bouncy Castle es fan en modalitat atached.

5. VALORACIÓ ECONÒMICA.

La valoració econòmica te dos vessants principals una la dels recursos humans i l'altre els recursos tecnològics.

En aquest cas la primera part no és aplicable donat que és el treball fi de carrera i el cost en hores que s'han emprat s'anul·la amb l'aprenentatge obtingut.

Per la part de recursos tecnològics el programari utilitzat per fer l'aplicació és programari lliure i ja es disposava del maquinari necessari per tant no s'han realitzat inversions ni en programari ni en maquinari.

6. CONCLUSIONS.

Amb el procés d'especificació i anàlisi de requeriments he pogut posar en pràctica molts dels coneixements adquirits en assignatures com Enginyeria del programari i Seguretat en xarxa de computadors.

A la fase d'implementació a més d'aplicar coneixements que ja tenia he adquirit d'altres dels que no disposava i que en algun moment m'han fet endarrerir-me en la meva planificació.

En contraposició amb els endarreriments he obtingut un avantatge per l'ús de J2EE en combinació amb altres aplicacions i tecnologies ja que m'han permès observar i utilitzar la integració d'aquestes diverses aplicacions i tecnologies, tot això m'ha permès superar la complexitat d'una aplicació distribuïda que pot arribar a desbordar fàcilment un programador.

En aquest projecte he posat en pràctica la integració de tecnologies com J2EE, Servlets i accés a base de dades mitjançant JDBC amb les eines que s'han descrit en el punt 4.2 d'aquesta mateixa memòria.

7. GLOSSARI.

Certificats digitals: document digital mitjançant el qual un tercer confiable (una autoritat de certificació) garanteix la vinculació entre la identitat d'un subjecte o entitat i la seva clau pública.

Autoritat de certificació: una autoritat de certificació, certificadora o certificant (AC, o CA per les seves sigles en anglès Certification Authority) és una entitat de confiança, responsable d'emetre i revocar els certificats digitals o certificats, utilitzats en la signatura electrònica, amb els quals s'empra la criptografia de clau pública. Jurídicament és un cas particular de prestador de serveis de certificació.

Signatures digitals: és un mecanisme de xifrat per autentificar informació digital. El mecanisme utilitzat és la criptografia de clau pública per això aquest tipus de signatura també rep el nom de signatura digital de clau pública.

x.509: és un estàndard UIT-T per a infraestructures de claus publiques (en angles, Public Key Infrastructure o PKI). X.509 especifica, entre altres coses, formats estàndard per a certificats de claus publiques i un algorisme de validació de la ruta de certificació.

Applet: Una miniaplicació (applet en anglès) és un component de programari que corre sobre el context d'un altre programa, com ara un navegador web. Les miniaplicacions acostumen a desenvolupar una funció molt concreta que no pot ser usada de forma independent.

Servlet: Les miniaplicacions de servidor (anglès servlets) són objectes Java executats per un servidor d'aplicacions i que responen a invocacions HTTP, servint pàgines dinàmiques. El contingut generat pot ser un fitxer de qualsevol tipus, la majoria de vegades HTML. Un objecte Servlet és capaç de rebre una invocació i generar una resposta en funció de les dades de la invocació, de l'estat del propi sistema i les dades a què pugui accedir.

8. RECURSOS.

8.1. Bibliografia consultada.

- Programación en Java 2. Anaya Multimedia. John Zukowski.
- Tecnología de servidor con Java: Servlets, JavaBeans y JSP, Eidos (2000).
- Java 1.2 Al descubierto, autor Jamie Jaworski, editorial PRENTICE HALL.
- Core Java 2 volumen II – Características avanzadas (Sun microsystems), autors Cay S. Horstmann i Gary Cornell, editorial PEARSON PRENTICE HALL.
- Manual de referencia J2EE, autor Jim Keogh, editorial McGraw Hill.

8.2. Recursos d'Internet.

- http://java.sun.com
- www.programacion.com
- http://es.wikipedia.org
- www.lawebdelprogramador.com
- www.bouncycastle.org
- http://tomcat.apache.org
- www.mysql.com
- www.openssl.org
- www.java.sun.com
- www.w3schools.com

9. ANNEX 1. SCRIPTS DE SEGURETAT

Creació del magatzem de claus del proveïdor de serveis i l'exportació del certificat del proveïdor de serveis:

- keytool -genkey -keystore magatzem.serveis -alias serveis
- keytool -export -keystore magatzem.serveis -alias serveis -file serveis.cert

Creació del magatzem de claus del proveïdor d'identitat i l'exportació del certificat del proveïdor d'identitat:

- keytool -genkey -keystore magatzem.identitat -alias identitat
- keytool -export -keystore magatzem.identitat -alias identitat -file identitat.cert

Conjunt d'instruccions per a la creació de la nostra Autoritat Certificadora de confiança:

- openssl genrsa -rand -des3 -out cakey.pem 1024
- openssl req -new -x509 -days 365 -key cakey.pem -out cacert.pem -config openssl.cnf

Conjunt d'instruccions per a la creació de les claus i certificats de servidor, que tot i que no utilitzarem en el nostre projecte he decidit crear-los per tal de poder disposar de tot l'entorn necessari per a la funcionalitat de la nostra Autoritat Certificadora de confiança:

- keytool -genkey -alias servidor -keypass jpua08 -validity 365 -storepass jpua08
- keytool -certreq -alias servidor -file servidor.csr -keypass jpua08 -storepass jpua08
- openssl ca -config openssl.cnf -in servidor.csr -out servidor.csr.pem -keyfile cakey.pem -cert cacert.pem
- openssl x509 -in servidor.csr.pem -out servidor.csr.der -outform DER
- keytool -storepass jpua08 -alias servidor -keypass jpua08 -import -file servidor.csr.der

Creació de les llistes de revocació de certificats de la nostra Autoritat Certificadora de confiança per tal de poder fer una correcta gestió dels certificats emesos per la mateixa:

- openssl ca -gencrl -out llistarev.crl
- openssl ca -gencrl -out llistarev.pem

Conjunt d'instruccions per a la creació de les claus i certificats de client generats amb la nostra Autoritat Certificadora de confiança i que utilitzarà el nostre "client" per a realitzar les seves signatures i facilitar el seu certificat digital:

- keytool -genkey -alias applet -keypass jpua08 -validity 365 -storepass jpua08
- keytool -certreq -alias applet -file applet.csr -keypass jpua08 -storepass jpua08
- openssl ca -config openssl.cnf -in applet.csr -out applet.csr.pem -keyfile cakey.pem -cert cacert.pem

- openssl x509 -in applet.csr.pem -out applet.csr.der -outform DER
- keytool -storepass jpua08 -alias CA -keypass jpua08 -import -file cacert.pem
- keytool -storepass jpua08 -alias applet -keypass jpua08 -import -file applet.csr.der

10. ANNEX 2. SCRIPTS DE CONFIGURACIO

```
01. ##linia de comentari
02. ##camí del keystore del servidor de serveis
03. C:\\Certificats\\magatzem.serveis
04. ##alias del keystore del servidor de serveis
05. serveis
06. ##paraula de pas del keystore del servidor de serveis
07. proveidorserveis
08. ##camí del keystore del servidor d'identitat
09. C:\\Certificats\\magatzem.identitat
10. ##alias del keystore del servidor d'identitat
11. identitat
12. ##paraula de pas del keystore del servidor d'identitat
13. proveidoridentitat
14. ##camí del keystore del client
15. D:\\.keystore
16. ##alias del keystore del client
17. applet
18. ##paraula de pas del keystore del client
19. jpua08
20. ##nom d'usuari de la nostra base de dades
21. admin
22. ##paraula de pas de la nostra base de dades
23. admin
24.
```

Aquestes línies s'han de guardar en un fitxer (que s'adjunta amb el conjunt de fitxers del projecte) anomenat "Configuracio.conf" i s'ha d'ubicar una copia del mateix en el directori d'instal·lació del Tomcat i una altra copia en el directori que contingui l'escriptori de Microsoft Windows de l'usuari actiu.

11. ANNEX 3. GUIÓ / ESQUEMA DE DESPLEGAMENT

A continuació llistaré els arxius presents en l'entrega del projecte i un petit guió/esquema on s'explica com desplegar i utilitzar les diferents aplicacions que conformen el conjunt del projecte.

Nom d'arxiu	Tipus d'arxiu	Ubicació	Utilitat de l'arxiu
BaseDades.sql	Script	----------	Conté la definició de la nostra base de dades i també les instruccions per a incloure les dades inicials.
SignarAPPLET.bat	Script	----------	Conté les instruccions necessàries per a crear i signar l'arxiu que contindrà el nostre applet.
Configuracio.conf	Script	-Directori d'instal·lació del Tomcat. -Directori que contingui l'escriptori de l'usuari actiu.	Conté dades sobre ubicacions d'arxius, alies i contrasenyes que necessitaran les diferents aplicacions del projecte.
ProveidorIdentitat.bat	Script	Directori on s'ha de crear el keystore del proveïdor d'identitat (apareix en l'arxiu Configuracio.conf)	Conté les instruccions necessàries per a crear el keystore i el certificat del proveïdor d'identitat.
ProveidorServeis.bat	Script	Directori on s'ha de crear el keystore del proveïdor de serveis (apareix en l'arxiu Configuracio.conf)	Conté les instruccions necessàries per a crear el keystore i el certificat del proveïdor de serveis.
Entitat_Client_CRL.bat	Script	Directori on es vulgui crear l'Autoritat Certificadora de confiança. (aquest ha d'estar reflectit dins l'arxiu openssl.cnf)	Conté les instruccions necessaries per a crear la nostra Autoritat Certificadora de confiança i les signatures i certificats de servidor i client.
Openssl.cnf	Script	Directori on es vulgui crear l'Autoritat Certificadora de confiança.	Conté la configuració necessària per al nostre openssl.

AltaServeis.class	Servlet	Directori de desplegament dels servlets del Tomcat indicat en l'arxiu web.xml	Conté el servlet que ens permetrà realitzar l'administració de la base de dades del proveïdor d'identitat.
ProveidorIdentitat.class	Servlet	Directori de desplegament dels servlets del Tomcat indicat en l'arxiu web.xml	Conté el servlet del proveïdor d'identitat.
ProveidorServeis.class	Servlet	Directori de desplegament dels servlets del Tomcat indicat en l'arxiu web.xml	Conté el servlet del proveïdor de serveis.
AppletClient,jar	Applet	Directori de desplegament de pagines web segures del Tomcat indicat en l'arxiu web.xml	Conté l'applet que es descarrega al navegador de l'usuari i per tant la lògica del client.
ServidorIdentitatWEB.html	Pàgina web	Directori de desplegament de pagines web no segures del Tomcat indicat en l'arxiu web.xml	L'utilitzem per accedir a l'administració de la base de dades del proveïdor d'identitat
PeticioServeiNou.html	Pàgina web	Directori de desplegament de pagines web segures del Tomcat indicat en l'arxiu web.xml	És la pagina web que utilitzarà l'usuari per tal de realitzar les seves peticions de servei.
firma-digital.jpg	Imatge	-Directori Imatges dins del directori de desplegament de pagines web segures del Tomcat indicat en l'arxiu web.xml. -Directori Imatges dins del directori de desplegament de pagines web segures del Tomcat indicat en l'arxiu web.xml	Imatge que apareix en les pagines web que s'utilitzen dins del projecte.
AltaServeis.java	Codi font	----------	Codi font de l'administració de la base de dades del proveïdor d'identitat.
ProveidorIdentitat.java	Codi font	----------	Codi font del servlet del proveïdor

			d'identitat.
ProveidorServeis.java	Codi font	----------	Codi font del servlet del proveïdor de serveis.
AppletClient.java	Codi font	----------	Codi font de l'applet.

Un cop ubicats els arxius dels servlets, l'applet, les pàgines web, la imatge i també els arxius Configuracio.conf i openssl.cnf ja es pot realitzar l'execució de qualsevol de les parts del projecte, es a dir podem realitzar l'administració de la base de dades del proveïdor d'identitat o realitzar peticions de servei per part dels usuaris.