Escenari segur client-servidor amb un proveïdor d'identitat i control d'accés extern

Jesús Gutiérrez Magallanes

ETIG/ETIS

Carlos Ares Angulo

13/06/2008

DEDICATÒRIA I AGRAÏMENTS

Dedico aquest projecte a aquelles persones que me han ajudat i suportat durant els últims quatre mesos. Entre elles es troben la meva família i els meus amics. Sense vosaltres no hagués sigut capaç d'arribar fins al final.

Agraeixo la paciència i ajuda del meu consultor, ja que hi ha hagut moments de desesperació en els quals m'ha sapigut encaminar cap a la direcció correcta.

RESUM

El projecte consisteix en desenvolupar un sistema que permeti a un client poder connectar-se a una pàgina web, validar-se sense contrasenya (amb un certificat idCat), i accedir als seus recursos.

En aquest escenari tenim quatre entitats ben diferenciades:

- *Client*: és la persona que es connecta des de un explorador d'Internet i que accedeix al servei.
- *Proveïdor de servei*: es tracta del servidor on s'ofereixen diferents recursos. En aquest cas permet accedir a: fotos, música, e-books, i jocs.
- *Proveïdor d'identitat*: es tracta del servidor que decideix si la petició que està fent el client es vàlida i, per tant, si aquest té permís per accedir al recurs sol·licitat.
- *Administrador:* és la persona que tindrà accés al servidor d'identitat mitjançant una pàgina web i definirà la relació entre els recursos als que tindran accés els clients. Per exemple: el client X tindrà accés a fotos però no a la resta.

L'objectiu final consisteix en que el proveïdor que ofereix els serveis es desentengui del procés de validació i sigui el proveïdor d'identitat qui faci aquesta tasca. Per tant, tenim un *escenari segur client-servidor* (xifrat amb el protocol SSL) *on el control d'accés es fa de manera externa*.

ÍNDEX DE CONTINGUTS

COS DE LA MEMÒRIA

INTRODUCCIÓ

JUSTIFICACIÓ DEL TFC I CONTEXT EN EL QUAL ES DESENVOLUPA

El mon de les tecnologies evoluciona de forma molt ràpida i van apareixent novetats en quan la forma de realitzar, segons quines tasques. Avui en dia, la gran majoria d'usuaris està acostumat a introduir el seu usuari i contrasenya quan intenta accedir a un recurs qualsevol de la xarxa Internet. Per exemple: quan volem accedir al nostre compte de correu o quan volem mirar com es troba el nostre compte bancari. Aquest sistema té una sèrie desavantatges:

- Molts dels recursos disponibles a la xarxa no t'obliguen a disposar d'una contrasenya robusta, és a dir, una contrasenya que sigui complicada d'obtenir per un hacker.
- Quan ens donem d'alta hem d'introduir una sèrie de dades i una contrasenya. Si utilitzem contrasenyes diferents, que és el més recomanable, ens hem de recordar a l'hora d'utilitzar-les.

El que proposa el meu TFC és utilitzar un sistema automàtic de validació com és la signatura electrònica. Realitzar una validació amb un certificat, realment no es tracta d'una millora ja que entitats oficials com Hisenda ja van servir aquest sistema. Però a nivell intern potser si que estem innovant ja que en aquest TFC la validació no la fa el proveïdor que ofereix el servei si no que la fa un altre proveïdor. Les millores que podria aportar aquest sistema són:

- Utilitzar un sistema molt segur en quant a validació de l'usuari.
- Poder utilitzar la mateixa forma de validar-se per qualsevol recurs de la xarxa.
- No caldria fer processos d'alta complicats. Podríem automatitzar el sistema per que agafés dades com el email del certificat.

Avui dia on l'ús d'Internet està molt generalitzat, és de vital importància mantenir una seguretat robusta quan ens connectem a la xarxa. Per aquest motiu penso que seria aconsellable utilitzar un sistema de validació com el que es presenta en aquest TFC.

OBJECTIUS DEL TFC

PREPARACIÓ DEL MATERIAL NECESSARI

Cal preparar un entorn de desenvolupament i de producció per poder programar l'aplicació i fer les proves. Això inclou instal·lar un Linux Ubuntu amb el Tomcat, Eclipse i la resta d'eines. Aquest mateix sistema em servirà tant per desenvolupar com fer simular de que es tracta d'un entorn de producció.

També caldrà obtenir un certificat digital idCat de l'Agència Catalana de Certificació. Aquest certificat serà suficient per poder validar i client i verificar si es troba a alguna llista de revocació per validar si es tracta d'un certificat vàlid.

S'ha de generar els certificats que s'utilitzaran per que els proveïdors puguin signar les dades i puguin establir comunicacions SSL. Amb l'eina Keytool ho podrem fer.

INSTAL·LACIÓ I CONFIGURACIÓ DE SOFTWARE

Caldrà fer la instal·lació del sistema operatiu (més software addicional necessari) dels host que utilitzarem. El sistema operatiu serà un Linux Ubuntu Desktop 8.04. amb les següents aplicacions addicionals:

- Servidor web Apache + Tomcat amb possibilitat de realitzar connexions segures SSL.
- Entorn de desenvolupament Java-Jdk.
- Base de dades Mysql on emmagatzemar la configuració que faci l'administrador del web.

FUNCIONALITAT DE COMPONENTS

- Client
    - Possibilitat de comunicar-se amb el proveïdor d'entitat mitjançant un applet.
    - Capacitat de poder seleccionar un certificat instal·lat al seu equip a través de l'applet.
    - Obtenció d'accés a un recurs mitjançant el certificat digital idCat.
    - Generació de codi XML signat.

- Proveïdor de servei
    - Capacitat d'acceptar peticions d'un client sense utilitzar un mecanisme d'autentificació.
    - Establiment de connexions segures i capacitat de signar dades i enviar-les a un client, i processar les dades signades d'aquest per poder determinar una resposta.
    - Capacitat de poder enviar un component web (applet)
    - Generació de codi XML signat.

- Proveïdor d'identitat
    - Possibilitat que el client sigui capaç de comunicar-se i processar les dades que rep.
    - Cal definir una base de dades on s'emmagatzemaran les dades.
    - S'utilitza una interfície per que l'administrador pugui gestionar les dades. Aquesta cal definir-la i ha d'operar de forma segura SSL.
    - Es capaç de validar la signatura del client mitjançant llistes de revocació.
    - Generació de codi XML signat.

JOC DE PROVES

Abans de l'entrega del projecte s'hauran de definir un joc de proves per donar com a bona la implantació:

- L'administrador pot gestionar les dades que hi ha al servidor d'identitat.

- El client es capaç de veure la web que ofereix el servei.
- L'acció del client es processa i el missatge que rep es correspon amb les condicions que ha definit l'administrador.
- Definició de resta de fluxos comuns que ens permeti verificar que l'aplicació funciona realment de forma adequada

PRODUCTE FINAL

El producte final està composat per:

- una memòria del projecte on hi aparegui un resum de tota la feina feta durant el semestre. Aquesta inclou un manual d'usuari, un manual d'instal·lació, i altres aspectes.
- Una presentació del producte on aparegui un resum de tota la feina realitzada. Ha de ser molt esquemàtica i visual.
- Un fitxer comprimit amb tot el codi font i un README on s'expliqui l'estructura i el contingut d'aquest fitxer.

RECERCA D'INFORMACIÓ

Serà convenient buscar informació referent a:

- Certificat idCat
- Instal·lació i configuració de software necessari
- Enviament d'applets a un client
- Signatura i validació de missatges en programació web
- Estructura del codi XML

MECANISMES DE SEGURETAT ESPERATS

- Protocol SSL: la comunicació que hi ha entre les diferents entitats que participen en el projecte: clients, administradors, proveïdors es connecten entre si de manera xifrada. D'aquesta manera evitem que s'intercepti d'informació per part d'un tercer. Per poder implementar aquest protocol cal l'ús de certificats, que seran generats amb l'eina Keytool.

- Certificat del client: aquest certificat serà emes per una autoritat de confiança: idCat. D'aquesta manera ens assegurarem que el client és qui diu ser. Per tant, no serà necessari que es validi amb un usuari i contrasenya, el certificat farà aquesta funció.

- Us de contrasenya: l'administrador haurà d'introduir un usuari y una contrasenya correcte per poder realitzar gestions. Abans, però, haurà de conèixer aquestes dades. Donarem per suposat que el procés d'alta d'un administrador és manual y que rep un mail del proveïdor d'identitat amb les seves credencials.

ENFOCAMENT I MÈTODE SEGUIT

En quan a la manera de planificar el projecte, crec que fent quatre entregues (PAC) anteriors, ha sigut una manera encertada de poder saber en quins aspectes s'estava equivocant l'estudiant, i poder corregir-los a temps. Després de cada entrega rebia un informe del consultor amb els aspectes que calia millorar. M'ha semblat una metodologia ideal.

Un dels grans problemes que he tingut ha sigut determinar el sistema per fer el debug i desplegament de les aplicacions. Inicialment estava utilitzant una màquina client amb Windows XP i feia el desplegament de forma manual. És a dir, primer depurava l'aplicació en XP i després passava el codi a un altre màquina, la de l'entorn de producció. Això m'ha ocasionat molts dolors de cap, però gràcies a les explicacions del consultor, he pogut corregir el problema. La solució ha passat per utilitzar una mateixa màquina per al desenvolupament i desplegament de l'aplicació.

En quant a la metodologia de treball sempre ha sigut la mateixa, recerca, recerca i més recerca. La gran majoria de dubtes els he resolt buscant informació a través d'Internet. També és cert que m'he recolzat molt en el consultor que ha sapigut ressoldre els meus dubtes..

Problemes que m'he trobat? Molts!. En l'apartat de *Decisions i evolució de la implementació del projecte* indicaré els canvis que he realitzat en quan al disseny a causa de trobar-me amb problemes que no he aconseguit resoldre.

Com a manera d'enfocar el problema, tal i com es pot apreciar en la planificació del projecte, primer m'he centrat en la gestió de recursos i després m'he dedicat a la resta de parts. Finalment he fet les proves i per acabar he redactat la memòria. També cal dir, que he anat anotant un document, a mode diari, tots els canvis que efectuava.

PLANIFICACIÓ DEL PROJECTE

Com que aquest projecte consta de diverses entregues, faré una planificació orientada a aquest aspecte. El projecte consta de sis entregues, i en cadascuna d'elles s'han d'assolir uns objectius.

PAC	Entrega	Tasques	Hores
1	06-03-08	Llegir i entendre enunciat	2
		Buscar informació relacionada amb la tecnologia SOAP i amb el dni-e, idCat, etc...	1
		Definició i planificació de les tasques	2
		Redactar document per entregar	4
2	20-03-08	Revisar i corregir errades que hagin sorgit del pla de treball inicial	3
		Definir amb més detall conceptes, arquitectura del sistema,	12

		tecnologies emprades, i eines	
		Redactar document per entregar	5
3	11-04-08	Estudi i proves per determinar l'ús de les tecnologies	5
		Definició exhaustiva del sistema que volem implementar	7
		Definició d'aplicacions necessàries i recerca de documentació per poder-les instal·lar i configurar	6
		Instal·lació i configuració del software necessari al servidor proveïdor de servei	7
		Anàlisis, disseny (comunicació, dades, pantalles, etc.) i programació d'aplicació web que permeti indicar al client quin recurs vol obtenir	8
		Redactar document per entregar	8
4	15-05-08	Estudi i proves per determinar l'ús de les tecnologies	5
		Instal·lació i configuració del software necessari al servidor proveïdor d'identitat	9
		Instal·lació i configuració del software necessari al client	2
		Anàlisis i disseny de base de dades	2
		Anàlisis, disseny (comunicació, dades, pantalles, etc.) i programació d'aplicació web que permeti gestionar les dades a l'administrador	12
		Definir i realitzar joc de proves	4
		Corregir errades	2

		Anàlisis, disseny i programació de component que s'enviarà al client i que fa es connecta amb el proveïdor d'identitat	10
		Redactar document per entregar i tenir acabat part del producte: gestió de recursos i clients, i sol·licitud de recursos	8
5	13-06-08	Anàlisis, disseny i programació de component que s'executarà al servidor d'identitat Cal programar la funció que permeti contrastar les dades del client i del proveïdor per prendre una decisió	12
		Completar programació d'aplicació web que s'executa al proveïdor de servei amb l'objectiu de poder rebre la resposta del client + resposta del proveïdor d'identitat i poder determinar si s'accepta o es denega el servei	10
		Definir i realitzar joc de proves definitiu	10
		Corregir errades	6
		Redactar memòria	8
		Manual usuari	6
		Manual instal·lació	4
6	19-06-08	Realitzar i preparar presentació	10

PRODUCTES OBTINGUTS

El resultat de tot el treball especificat en la planificació és aquest:

- Memòria: document on es reflexa el treball que he realitzat per desenvolupar el producte. Inclou manual d'usuari i d'administrador per a poder facilitar la feina de implantació i utilització de l'eina.

- Producte: consta de diverses parts. Una de elles es correspon amb el software necessari per fer funcionar el codi desenvolupat, i l'altre es tracta del propi codi.
- Presentació: diapositives que ens permeten descriure el treball fet de manera resumida i visual.

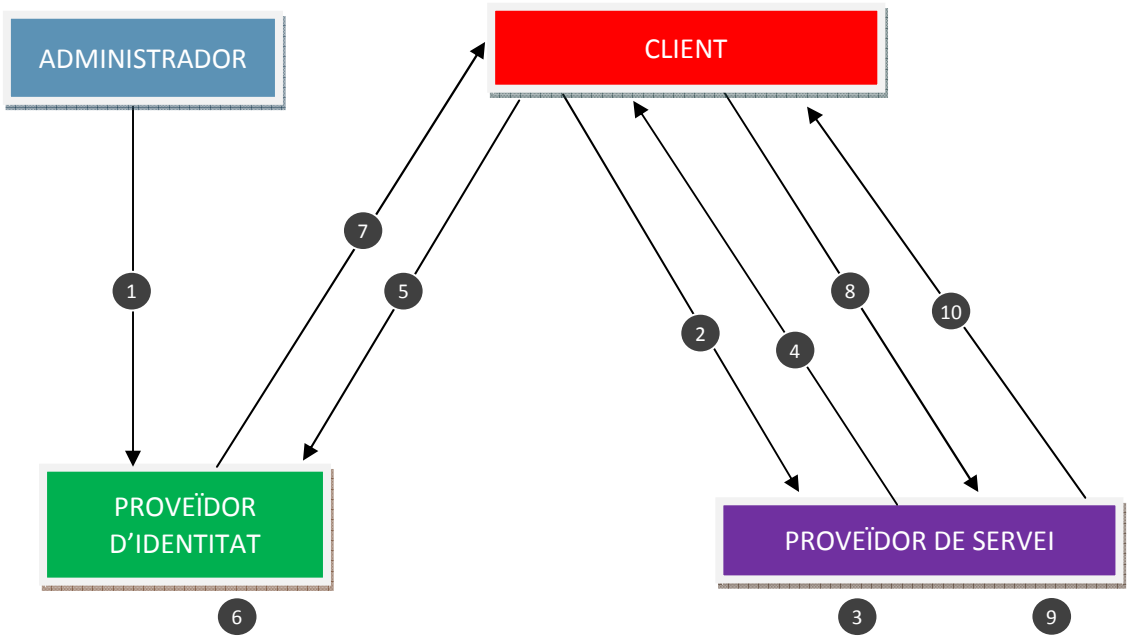
ANÀLISI I DISSENY

ESPECIFICACIÓ I DISSENY DEL PRODUCTE

El sistema està composat per 4 entitats: client, administrador, proveïdor d'identitat i proveïdor de servei. Bàsicament el sistema consisteix en una sèrie de peticions que fa un client, de forma anònima, el proveïdor de servei no sap qui està fent la petició, a un proveïdor de servei per obtenir accés a un recurs. Aquest es connecta a un proveïdor d'identitat que és qui s'encarrega de determinar si l'usuari pot accedir al recurs. Prèviament l'administrador haurà d'haver accedit al proveïdor d'identitat per administrar recursos, usuaris i permisos d'accés als recursos.

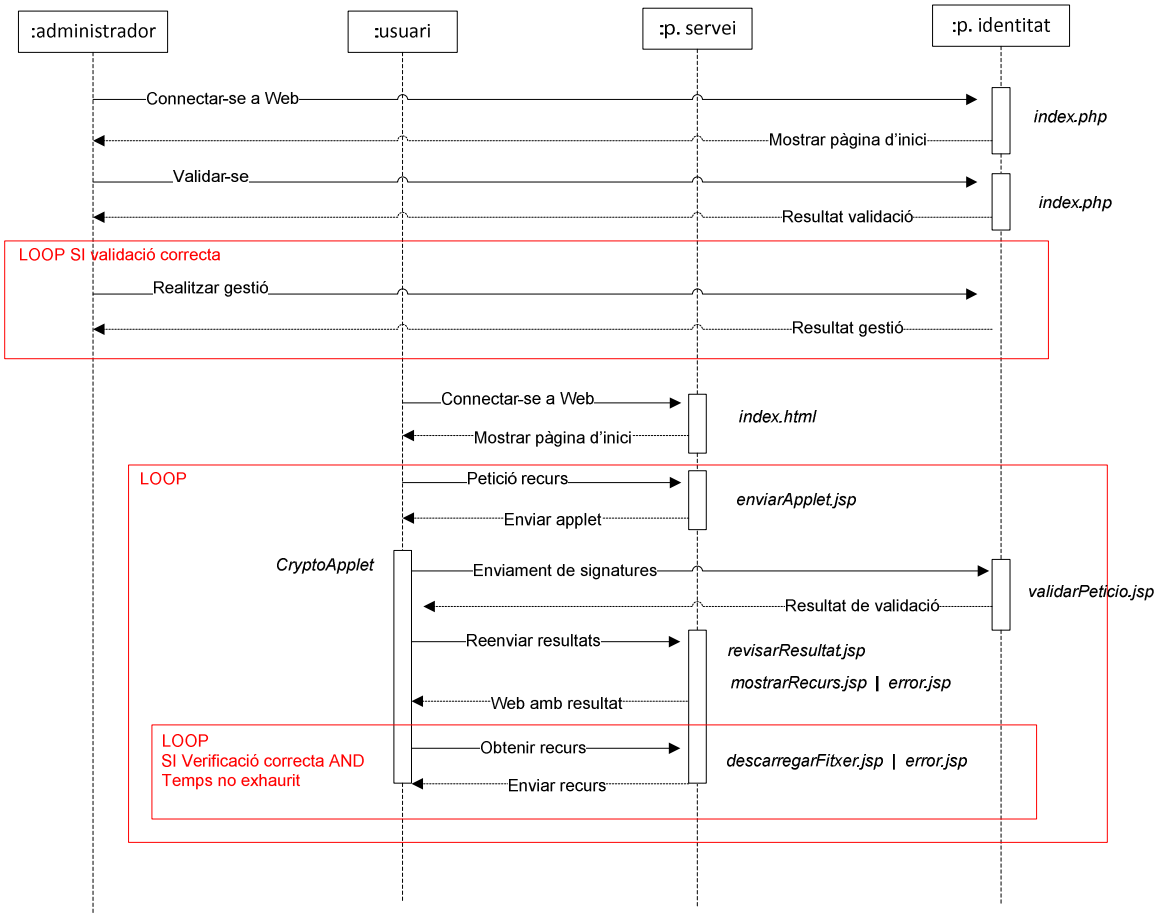
Per tal d'explicar el sistema de forma més gràfica utilitzaré alguns diagrames. Aquests s'utilitzen generalment per facilitar la comprensió de llargues quantitats de dades i la relació entre les diferents parts de les dades. Els diagrames poden generalment ser llegits més ràpidament que les dades en brut del que procedeixen. Per aquest motiu en aquest apartat realitzaré el diagrama de flux de dades i un de seqüència, que és el que més adequat per plasmar el sistema que es vol implementar.

DIAGRAMA DE FLUX DE DADES



1. Gestió usuaris, serveis i permisos.
2. Sol·licitar petició (anònima en un canal segur). No es necessària identificació i el servidor no sap qui està accedint.
3. Generar identificador i guardar-lo en la sessió junt amb l'hora en que s'inicia la transacció. Signar aquestes dades de petició i component i es genera una cadena XML amb aquestes dades.
4. Enviar petició signada + component, l'Applet rep com a paràmetre la cadena XML.
5. Enviar petició signada, en format XML, pel proveïdor de servei + petició signada pel client (mitjançant Javascript i HTML POST).
6. Verificar les dues signatures anteriors i verificar si els certificats dels signants són vàlids. Això inclou comprovar si el certificat del client es troba en alguna llista de revocació.
7. Enviar el resultat, mitjançant Javascript i HTML POST, generant un nou XML amb una sèrie de dades que s'explicaran en l'apartat següent.
8. Reenviar el resultat, cadena XML, al proveïdor de servei.
9. Verificar el resultat. Això inclou verificar que les signatures siguin correctes i que es corresponguin amb les dades de la petició feta.
10. Respondre amb acceptació o denegació del servei

SEQÜÈNCIA



En les subseccions següents es descriurà amb més detall els processos que intervenen en aquest diagrama. Cal tenir en compte que la comunicació entre els diferents components és segura, sota el protocol SSL. En l'apartat *de manual d'administrador* es descriu la manera d'aconseguir implementar aquest protocol.

ADMINISTRADOR <---> PROVEÏDOR D'IDENTITAT

- Connectar-se a Web: l'administrador es connecta a la web del proveïdor d'identitat per tal de poder gestionar els recursos i usuaris.

  Codi cridat: *index.php*

- Mostrar pàgina d'inici: l'aplicació s'encarrega de generar el codi HTML necessari per que l'administrador vegi la pàgina d'inici al seu navegador.

  Codi executat: *index.php*

- Validar-se: l'administrador introdueix un usuari i una contrasenya. De forma interna, el navegador envia aquestes dades al servidor mitjançant el mètode HTML POST.

  Codi cridat: *index.php*

- Resultat validació: l'aplicació comprova que l'usuari y contrasenya corresponen amb el que hi ha guardat a la taula "Administrador" del gestor de base de dades del servidor. Prèviament hem afegit a mà en la base de dades aquest usuari. La contrasenya es guarda codificada.

  En funció del resultat el jsp genera un codi o un altre:

  - Validació correcta: genera una pàgina HTML amb la web principal de gestions. A més es crea una sessió que guardarà el nom de l'usuari. Aquest nom es mostrarà a la part superior de la pantalla.
  - Validació incorrecta: genera una pàgina HTML que mostra un missatge indicant que la validació ha sigut incorrecta. L'administrador pot intentar validar-se un altre cop.

  Codi executat: *index.php*

- Realitzar gestió: un cop l'administrador s'ha validat correctament, ja pot començar a realitzar les gestions que cregui convenients. Cada cop que seleccioni una opció es crida al php que gestiona els recursos.

  Codi cridat: *index.php*

- Resultat gestió: cada gestió que faci ha de ser validada i mostra el resultat correcte o incorrecte de la gestió en HTML.

Codi executat: *index.php*

Cada cop que s'accedeix a una pàgina de gestió es comprova que hi ha creada la sessió del usuari. Això ens indicarà de que es tracta d'un administrador validat correctament.

USUARI <---> PROVEÏDOR DE SERVEI

- Connectar-se a Web: un cop l'administrador ha configurar els usuaris, recursos i permisos, l'usuari ja pot connectar-se a la web del proveïdor de servei per tal de poder accedir a un recurs.

  Codi cridat: *index.html*

- Mostrar pàgina d'inici: l'aplicació s'encarrega de generar el codi HTML necessari per que l'usuari vegi la pàgina d'inici al seu navegador.

  Codi executat: *index.html*

- Petició recurs: l'usuari disposa d'una sèrie de recursos als que pot accedir. Cada cop que s'accedeix a un recurs s'ha de realitzar un procés de validació.

  Codi cridat: *enviarApplet.jsp*

  Aquest codi s'envia mitjançant Javascript i fent ús del HTML POST.

- Enviar applet: l'aplicació genera un HTML que conté una etiqueta <APPLET> amb la direcció del servidor on es troba l'aplicació CryptoApplet. Aquesta s'encarregarà de mostrar una finestra on l'usuari podrà seleccionar el certificat amb el qual vol signar les dades.

  Codi executat: *enviarApplet.jsp*

Es genera una sessió on es guarda l'identificador de petició y de recurs seleccionat. L'identificador de petició l'obté del SID de la sessió. Es genera un 'XML on s'haurà d'incloure la signatura de les dades guardades a la sessió. La signatura la realitzarem amb el keystore que es troba a /projecte/keystorePS

A mode d'exemple, podríem tenir les següents etiquetes:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <PeticionCliente>
    <IdPeticion>819CE0D7BB22E7166D308B3FE96ACE24</IdPeticion>
   <IdRecurso>musica</IdRecurso>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
                                        …
```

Els camps que apareixen són aquests:

- IdPeticion: identificador de petició.
- IdRecurso: identificador del recurs.
- Signature: signatura del proveïdor de servei i del client

Acumularem en una variable de text tot el XML generat i se'l passarem per paràmetre a l'applet. D'aquesta manera podrem obtenir posteriorment des del client del codi generat, amb la signatura del proveïdor de servei.

USUARI <---> PROVEÏDOR D'IDENTITAT

- Enviament de signatures: el codi *enviarApplet.jsp* anterior farà una petició HTML POST al servidor d'identitat. L'objectiu és verificar l'autenticitat de l'usuari i si té permisos per poder accedir al recurs que està sol·licitant.

  Codi cridat: *validarPeticio.jsp*

  L'applet rep per paràmetre la cadena XML, i torna a signar les dades amb la clau privada idCat.

- Resultat verificació certificats: el proveïdor d'identitat obté el XML i realitza les següents verificacions:
    - verifica que el certificat del client es correcte mirant que no es troba la llista de revocació.
    - Verifica que el Subject del certificat correspon amb algun usuari donat d'alta a la base de dades.
    - El servidor només confiarà en certificats que estiguin emesos per l' Agència Catalana de Certificació.
    - Només confiarà en peticions que provinguin del proveïdor de servei gutzworld.sytes.net

El servidor genera un nou XML amb la seva clau privada, que es troba a la carpeta /projecte/keystorePI, de les dades rebudes del client i de la decisió del procés de validació. El forma generat seguirà aquesta estructura:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <RespuestaProvIdent>
        <KeyDescProvServ>CN=gutzworld.sytes.net,OU=DepInformatica,O=Pro veedor de
        Servicio,L=Madrid,ST=Madrid,C=ES</KeyDescProvServ>
         <SubjectProvServ>gutzworld.sytes.net</SubjectProvServ>
        <KeyDescCliente>CN=JESUS GUTIERREZ MAGALLANES,2.5.4.5 =
        #13093436393431363939 58,2.5.4.42 = #13054a45535553,2.5.4.4 =
        #13144755544945525245 5a204d4147414c4c414e4553,OU=Vegeu
        https://www.catcert.net/verIDCat (c)03,C=ES</KeyDescCliente>
         <SubjectCliente>JESUS GUTIERREZ MAGALLANES</SubjectCliente>
        <IdPeticion>4D9E4A3A5E34C5B167CBFBA72142988E</IdPeticion>
        <IdRecurso>musica</IdRecurso>
        <Resultado>false</Resultado>
        <Tiempo>-1</Tiempo>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
                                    …
```

Els camps que apareixen són aquests:

- KeyDescProvServ:descripció del certificat del proveïdor de servei.
- SubjectProvServ: Subject del certificat del proveïdor de servei.

- KeyDescClient: descripció del certificat IdCat del client.
- SubjectCliente: Subject del certificat del client.
- IdPeticion: identificador de petició.
- IdRecurso: identificador del recurs.
- Resultado: resultat de la validació.
- Tiempo: si val -1 vol dir que el procés de validació no ha sigut correcte. Si es superior a 0 vol dir que l'usuari existeix i té associat un recurs.
- Signature: signatura del proveïdor d'identitat.

Codi executat: *validarPeticio.jsp*

USUARI <---> PROVEÏDOR DE SERVEI

- Reenviar resultats: el resultat de la validació, és a dir, de verificar si el certificat de l'usuari és correcte i si té accés per veure el recurs es reenvia al proveïdor de servei. L'applet li envia mitjançant el resultat mitjançant el mètode POST.

  Codi cridat: *revisarResultat.jsp*

- Web amb resultat: l'aplicació rep per paràmetre el resultat de la validació i analitza si és correcta o no:
  - Compara el id de petició i el id de recurs rebut amb el que havia guardat en el moment en que el client havia fet la petició.
  - Verifica que el Subject del proveïdor d'identitat es l'esperat. Només admet respostes d'aquest proveïdor d'identitat concret.
  - Verifica que el temps màxim per accés a un recurs no s'ha superat
  - Comprova que el resultat de la petició es true: positiu.

  En funció de les comprovacions anteriors, es donaran els següents resultats:

  - Validació correcta: genera un HTML amb la pàgina principal del recurs que el client volia accedir. Si torna cap enrere i vol accedir a un altre recurs, s'haurà d'efectuar un altre cop el procés de validació complet.

El llistat de fitxers disponibles s'obté mitjançant una funció que obté els fitxers que es troben en una carpeta del servidor. Aquests seran els que hagi pujat l'administrador.

El codi *revisarResultat.jsp* farà una redirecció a la pàgina *mostrarRecurs.jsp.*

- Validació incorrecta: genera un HTML amb un missatge que indica que el procés no ha sigut satisfactori. El client no pot accedir al recurs, però pot intentar accedir a un altre.

Codi executat: *revisarResultat.jsp* i *mostrarRecurs.jsp.*

- Obtenir recurs: Un cop el client és dintre de la pàgina principal del recurs pot obtenir els fitxers que vulgui, fent clic a sobre d'ells.
  - Codi cridat: *descarregarFitxer.jsp*

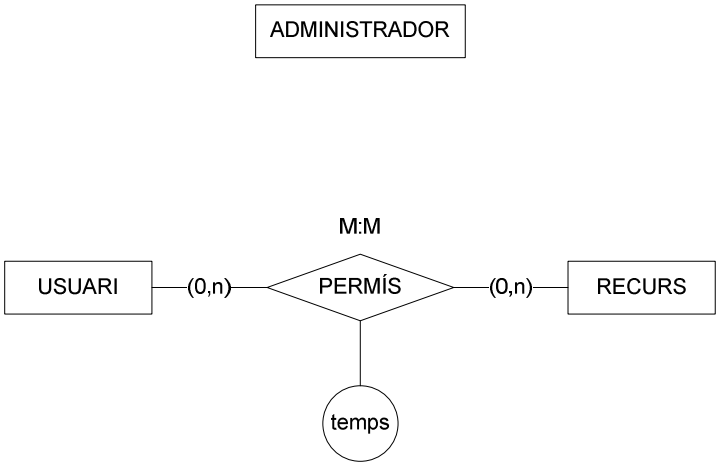
Aquest pas es fa per evitar que el fitxers puguin ser descarregats de forma directa mitjançant la url del fitxer. En aquest cas els fitxers no es troben en un lloc que no és públic al servei web.

- Enviar recurs: els servidor ofereix el fitxer, que es troba a la carpeta a la mateixa carpeta on apunta el servei FTP, per que l'usuari se'l descarregui. O sigui, comença el procés de descàrrega del recurs.
  - Codi executat: *descarregarFitxer.jsp* o *error.jsp*

Abans però es verifica que hi ha una sessió oberta, que el resultat de la validació és true i que no s'ha superat el temps màxim d'accés al recurs. SI es així, el codi *descarregarFitxer.jsp* fa una redirecció a la pàgina *error.jsp.*

MODEL CHEN

En aquest model es pretén visualitzar els objectes que pertanyen a la base de dades com entitats ,les quals tenen uns atributs, i quines relacions hi ha entre ells. Aquesta base de dades correspon a la que hi haurà allotjada al proveïdor d'identitat i on es guarda la relació que hi ha entre recursos i clients.



En aquest cas es tracta d'un model conceptual molt senzill. L'entitat ADMINISTRADOR no té cap relació amb la resta d'entitats per què només té la funció d'emmagatzemar els usuaris i contrasenyes que permeten validar-se en la web del proveïdor d'identitat.

ANÀLISIS DE LES ENTITATS

USUARI

Representa tots els usuaris que es donen d'alta en sistema. Aquest usuaris seran els que podran accedir i gaudir dels recursos.

Conté els atributs següents:

- **idUsuari:** identificador de l'usuari.
- nom: nom i cognoms de l'usuari. Aquest text apareixerà a la part superior de la web de servei un cop l'usuari s'hagi validat (gràcies al certificat)
- subject: aquest és el text que s'espera obtenir del certificat. Aquest text serà el que li permeti determinar al sistema qui és l'usuari que s'està validant. Per tant, el subject del certificat ha de coincidir amb el que introduïm aquí.

RECURS

Representa tots els recursos que es donen d'alta en sistema. Aquest recursos seran als que els usuaris podran accedir i gaudir.

Conté els atributs següents:

- **idRecurs:** identificador del recurs. Aquest text apareixerà sota de cada recurs que es visualitzi en la web del proveïdor de servei.
- descripció: text explicatiu del que contindrà aquest recurs.

ADMINISTRADOR

Representa als usuaris que podran accedir a la web del proveïdor d'identitat per gestionar els recursos i usuaris.

Conté els atributs següents:

- **idUsuari:** identificador de l'administrador. Aquest usuari s'haurà d'introduir en la web del proveïdor d'identitat per validar-se..
- contrasenya**:** aquesta contrasenya s'haurà d'introduir en la web del proveïdor d'identitat per validar-se.
- nom: nom i cognoms de l'administrador. Aquest text apareixerà a la part superior de la web d'identitat un cop l'usuari s'hagi validat (gràcies a l'usuari i la contrasenya).

ANÀLISIS DE LES RELACIONS

PERMÍS

usuari – **PERMÍS** – recurs

Aquest relació representa l'associació d'un usuari amb un recurs per a que tingui PERMÍS per accedir-hi.

Característiques:

- Un usuari pot ser que no tingui cap recurs associat.
- Un recurs pot ser que no estigui associat a cap client.
- No pot haver assignat un recurs a un client més d'una vegada.
- Mai es podrà esborrar un client o un recurs que estigui associat.

Atributs:

- **idRecurs:** identificador del permís. Junt amb el idUsuari formen la clau primària.
- **idUsuari:** identificador del permís. Junt amb el idRecurs formen la clau primària.
- temps: quantitat de minuts que un usuari pot accedir a un recurs, un cop validat. Si aquest temps s'exhaureix l'usuari tindrà que validar-se un altre cop per poder gaudir del recurs.

TAULES NORMALITZADES

USUARI (idUsuari, nom, subject)

RECURS (idRecurs, descripcio)

ADMINISTRADOR (idUsuari, contrasenya, nom)

PERMIS (idRecurs, idPermis, temps)

EXPLICACIONS ADICIONALS

ESTABLIR UNA CONNEXIÓ SEGURA SSL

Per que totes les comunicacions siguin xifrades haurem de configurar el Tomcat. S'haurà de generar una clau i un certificat pel servidor. Això ho aconseguirem amb l'eina keytool i les guardarem en la carpeta /projecte/. Cada proveïdor tindrà la seva pròpia clau, ja que a l'hora de signar interessa que en la signatura del XML aparegui el Subject de cadascun d'ells, segons el cas.

La configuració del Tomcat es bastant simple, només cal modificar el fitxer XML que té de configuració.

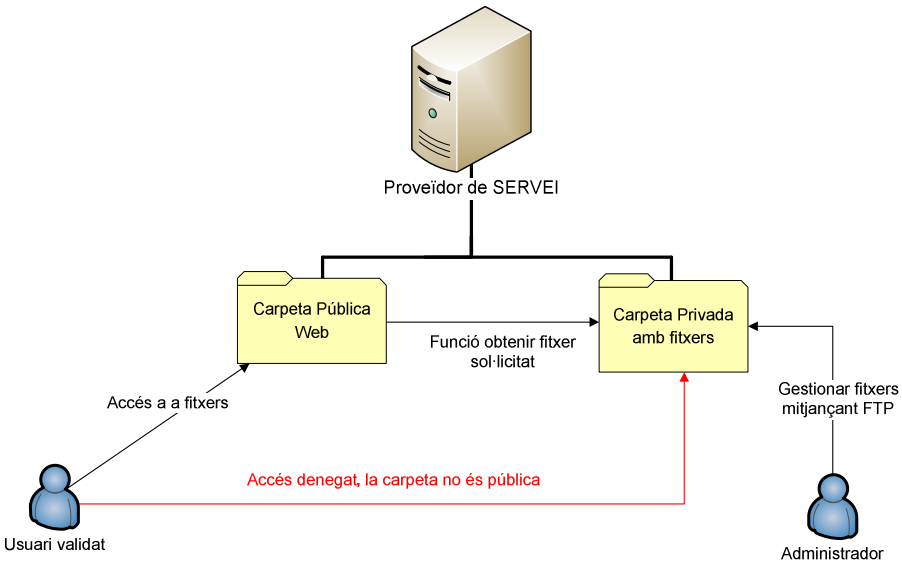
ACCÉS ALS RECURSOS (FITXERS)

Quan un usuari es valida amb el seu certificat i accedeix a la pantalla de recursos es pot descarregar una sèries de fitxers. Hi ha un procés que no hem explicat en el diagrama de seqüència i que és vital per que l'usuari pugui obtenir aquests fitxers.

En la imatge anterior podem observar una sèrie de fitxers amb extensions .mid. Aquests fitxers es troben ubicats a una carpeta privada del servidor. L'accés a aquesta carpeta només està permès a l'aplicació web que s'executa al servidor i al servei de FTP. Per tant, per una banda, l'aplicació web mostrarà de manera automàtica el llistat dels fitxers que hi ha en aquesta carpeta, i per una altre, l'administrador podrà gestionar aquests fitxers via FTP.

En l'apartat d'anàlisi s'explica que l'administrador rebrà via mail un usuari i contrasenya per poder accedir a l'administrador de recursos del proveïdor d'identitat. Aquestes mateixes credencials seran vàlides per accedir al servidor FTP. Per tant, el proveïdor de servei, a més d'oferir un servei web, oferirà un servei FTP.

Faig un esquema per que quedi més clar:



Amb aquest sistema s'evita que l'usuari pugui obtenir els fitxers directament mitjançant la URL sense estar validat. Els links als que té accés l'usuari no apuntaran al fitxer del servidor, si no que criden a una funció que s'encarrega de proporcionar-li el fitxer.

ESTAT DE L'ART DE LES TECNOLOGIES

TECNOLOGIES QUE INTERVENEN

- Software:
  - Servidor Web HTTP Apache: permet allotjar pàgines web per a que els clients puguin connectar-se des de el navegador.
  - Tomcat: s'utilitza amb combinació amb l'Apache i ens proporciona suport de servlets i JSPs. Per tant, podrem desenvolupar aplicacions amb Java i executar-les en aquest servidor.
  - Mysql: gestor de base de dades que ens permet emmagatzemar informació en forma de taules. Guardarem la informació referent al clients, recursos i privilegis que l'administrador introdueixi.
  - Internet Explorer 7: programa que li permet a l'usuari accedir als recursos que li ofereix Internet. Tant el client com l'administrador necessitaran accedir a les pàgines que estan allotjades als proveïdors.
  - JDK Java: entorn de desenvolupament Java. També incorpora les classes necessàries per verificar si el certificat del client que rep el proveïdor d'identitat és vàlid o no.
  - J2EE: plataforma de desenvolupament necessària per programar els components que ens demanen en aquest projecte.

- Protocols:
  - SSL: protocols criptogràfics que proporcionen comunicacions segures en Internet. Les comunicacions entre client, administrador i proveïdors ha d'anar sempre xifrada amb aquest protocol.

- Llenguatges de programació:
  - HTML: llenguatge predominant per a la construcció de pàgines web. Serà necessari utilitzar-lo per programar les pàgines allotjades tant al proveïdor de servei com al d'identitat.

XMLDSig: l'utilitzarem per l'enviament de les dades entre els diferents components que interactuen en el sistema ja que permet compartir la informació de una manera segura, fiable i fàcil.

- JSP: programa que s'executarà als proveïdors i que s'encarregarà de generar les pàgines HTML, de forma dinàmica, que seran visibles pels clients.
- Javascript: ens permetrà realitzar el Submit dels formularis que utilitzem en els nostres JSPs, i que permetrà la comunicacions entre els diferents components.

- Llibreries i components
  - CryptoApplet: aplicació desenvolupada per la Universitat Jaume I, amb llicència GPL2, que ens permet poder treballar amb el certificat idCat. A més ofereix una interfície on el client pot seleccionar el certificat amb el que vol signar.
  - Java Base64: llibreria que ens permet codificar el text XML per poder-lo enviar entre pàgines.
  - Mysql: llibreria que ens permet connectar-nos a la base de dades que es troba al proveïdor d'identitat.

- Altres:
  - Infraestructura de Clau Pública (PKI): permet als usuaris autentificar-se davant a altres usuaris i utilitzar la informació dels certificats d'identitat per xifrar i desxifrar missatges, signar digitalment informació, garantir el no repudi d'un enviament, etc. El protocol SSL fa us d'aquest sistema per xifrar l' informació.
  - Signatura digital: es una tecnologia que produeix els mateixos efectes jurídics que la firma a mà d'un document físic, sent també admissible com a proba d'un judici, en funció de la legislació de cada país. En l'esquema proposat a l'apartat 1 podem veure com es signa informació diverses vegades.

DECISIÓ TEGNOLOGIA

- SO: en comptes d'utilitzar servidors Microsoft he optat per una alternativa lliure com és Linux. Podria haver fet la mateixa implementació del projecte en altres sistemes operatius, sempre que es compleixin una sèrie de requisits:
    - Client: que tingui navegador amb suport Java
    - Proveïdor servei: que disposi d'un servidor web amb suport de servlets.
    - Proveïdor identitat: que disposi d'un servidor web amb suport de codi jsp, i disposi d'un gestor de base de dades compatible amb Java.

- Servidor web: he escollit el Tomcat per que està bastant estès i hi ha bastant documentació. També hi ha alternatives bastant interessants com el JBoss o GlassFish.

- Llenguatge de programació: encara que no estic del tot familiaritzat amb Java, és un llenguatge que m'ofereix totes les eines necessàries per desenvolupar el projecte. He estat mirant alternatives com PHP5, però no he trobat molta informació referent a l'enviament de missatges XML + signatura. També he trobat problemàtic com fer l'enviament d'un applet que s'executi al client amb PHP.

IMPLEMENTACIONS DE REFERÈNCIA

- Accés a organismes de l'administració central

  http://www.cert.fnmt.es/index.php?o=cert#cert3

  Amb el certificat de la FNMT podem accedir a una sèrie de serveis sense la necessitat d'utilitzar usuari i contrasenya. El nostre cas és molt similar ja que el que volem validar es l'accés d'un usuari a un recurs. A més volem validar l'autenticitat de l'usuari.

- Buscador Google

  http://weblogs.java.net/blog/jitu/archive/2006/01/accessing_googl_1.html

  En aquesta pàgina podem veure l'ús del protocol SOAP i la tecnologia web service per realitzar una cerca a través del famós Google.

- Validar operacions bancàries

  http://www.kriptopolis.org/firma-digital-en-banca-on-line

  En aquesta web hi ha una noticia on diu que la entitat bancària ING permet validar operacions bancàries amb un certificat emès per la FNMT.

EINES QUE CAL UTILITZAR

- Eclipse: entorn de desenvolupament que em permetrà programar les aplicacions necessàries.
- Notepad+: aplicació que em permetrà inicialment fer la codificació de la pàgina HTML per determinar quin aspecte tindrà. Un cop la doni com vàlida la codificaré com a servlet.
- Wireshark: aplicació que ens permetrà monitoritzar quina informació s'està enviant y rebent des de el client. Pot ser útil si tenim algun tipus de problema amb la comunicació.
- gFTP: per pujar recursos al proveïdor d'identitat.

DECISIONS I EVOLUCIÓ DE LA IMPLEMENTACIÓ DEL PROJECTE

SOFTWARE I LLENGUATGE DE PROGRAMACIÓ

- Canvi de versió Ubuntu Server 6.0.6 per Ubuntu Desktop 8.04.
  La primera no ofereix possibilitats d'entorn gràfic i per tant no la podia utilitzar-la com a entorn de desenvolupament. A més, la nova versió software més actualitzat i correcions d'errades.

- Canvi de llenguatge JSP per PHP en l'administrador de recursos del proveïdor d'identitat
  És un llenguatge amb el que estic més familiaritzat i era una forma de guanyar temps. Aquest canvi no afecta en el funcionament del sistema, es només a nivell intern.

- Us de HTML en comptes de XML i XSLT

  Inicialment tenia pensat en programar les pàgines web amb XML i donar-les format amb XSLT. De fet, pensava que era l'única manera de poder enviar un fitxer XML amb les dades signades. Però a mesura que vaig anar investigant vaig veure que no era indispensable, a més de que seria dedicar massa temps en aprendre a formatejar les dades amb XSLT. Per tant, vaig optar per passar la cadena XML amb les dades firmades mitjançant HTML POST, guardant en un paràmetre HIDDEN les dades i fent un submit amb Javascript.

- No fer us de la tecnologia VMWARE

  El meu consultor em va dir que no era indispensable simular un entorn real on tingués un host pel proveïdor de servei, un altre pel d'identitat i un últim pel client. Per tant, una manera d'estalviar temps era implementar-ho tot sobre una mateixa màquina.

DISSENY

- Canvis en l'interfície en la gestió de recursos del proveïdor d'identitat

  Vaig eliminar l'apartat de registres ACTIVATS/DESACTIVATS. He cregut que no era una funcionalitat molt útil, tenint en compte el volum petit de dades que hem d'introduir en el moment d'alta d'un registre. A més, complicava una mica més la programació, sense aportar una gran millora.

  Abans

Ara

- No es poden afegir recursos nous, només poden ser 4: fotos, música, e-books, jocs

  Inicialment la idea era poder donar d'alta tants recursos como volguéssim en la gestió de recursos, però això complicava l'escenari, ja que l'administrador tenia que indicar d'alguna manera quines imatges eren les que es mostrarien en el proveïdor de servei. A més un cop donat d'alta el recurs estava la problemàtica de com fer per a que el proveïdor de servei es connectés al proveïdor d'identitat per anar a buscar aquestes imatges.

  Per falta de temps, el nombre de recursos i les seves fotos seran estàtiques. És a dir, la pàgina principal del proveïdor de servei no anirà a buscar les imatges i recursos disponibles al proveïdor de servei, si no que donarà per suposat que sempre hi ha els mateixos recursos: fotos, música, e-books i jocs.

Per tant, en el moment de definir els recursos el seu id haurà de correspondre amb el text que apareix sota de cada icona.

- Ús d'aplicació CryptoApplet en comptes de programar una aplicació pròpia

  Vaig intentar treballar amb el certificat idCat des de l'applet però no hi havia manera. El consultor em va recomanar que generés el meu propi certificat ja que la l'accés a aquest era bastant més complicat del que es pensava. Buscant per Internet vaig trobar un software que em permetia treballar amb ell. Es tracta d'un conjunt de llibreries desenvolupades en Java. Dona la casualitat que l'eina CryptoApplet ha sigut desenvolupada per la mateixa gent que ha programat l'eina de gestió del certificat idCat que ve en la clau: la Universitat Jaume I. És una eina lliure amb llicència GPL2, o sigui, que puc utilitzar-la.

  A part d'aquesta funcionalitat, l'eina CryptoApplet em permet generar signatura amb format XMLDSig, seleccionar el certificat amb el que vull signar les dades i tenir una vista prèvia de les dades que signaré.

- Eliminació de protocol SOAP

  Vaig fer provés amb aquest sistema y la veritat és que no era gaire complicat d'implementar. Però vaig veure que no afegia cap funcionalitat excepcional en el sistema que volia implementar. Com que no era requisit indispensable utilitzar aquest protocol vaig decidir fer ús del HTTP POST.

MANUALS

ADMINISTRADOR DE SISTEMES

REQUISITS

- Un PC amb, com a mínim, 512 MB de RAM i un disc dur igual o superior a 10GB.
- El cd d'instal·lació d'Ubuntu 8.04 Desktop
- Connexió a Internet de banda ampla ADSL
- Fitxer amb els fonts *gutzworld_producte.zip*

INSTAL·LACIÓ DE SISTEMA OPERATIU UBUNTU

- Instal·lar Ubuntu 8.04 - Desktop . En idioma espanyol i creació de l'usuari *gutz*
- En la instalació seleccionar serveis openssh i LAMP
- Passwd mysql: *enKriPTado*
- Configurar xarxa amb ip *192.168.1.50*
- El particionament del disc dur el fem automàtic.

INSTAL·LACIÓ TOMCAT Y J2EE

- Instal·lar paquets des de la consola

  sudo apt-get install sun-java6-jdk tomcat5.5 tomcat5.5-admin tomcat5.5-webapps openssl

- Accedir a la pàgina http://192.168.1.50:8180 i verificar que apareix la pàgina d'inici del Tomcat
- Afegir un usuari per poder accedir a la gestió del servidor
  - Editar el fitxer */etc/tomcat5.5/tomcat-users.xml* i afegir la següent línia:

    <user username="gutz" password="0181" roles="manager,admin"/>

- Editar el fitxer */etc/default/tomcat5.5* i canviar la següent línea:
  *TOMCAT5_SECURITY=**no***

- Configurar protocol HTTPS
  - Creació del directori /projecte on anira la clau privada que utilitzarà el servidor Tomcat.
  - Crear el certificat fent ús de la següent comanda:
    keytool -genkey -alias tomcat -keyalg RSA -keystore /projecte/keystore

    A continuació saltarà un assistent:

    *Escriba la contraseÃ±a del almacÃ©n de claves: **enKriptar***
    *Volver a escribir la contraseÃ±a nueva: **enKriptar***
    Â¿CuÃ¡les son su nombre y su apellido?
    *[Unknown]: **gutzworld.sytes.net***
    Â¿CuÃ¡l es el nombre de su unidad de organizaciÃ³n?
    *[Unknown]: **Dep. Sistemas***
    Â¿CuÃ¡l es el nombre de su organizaciÃ³n?
    *[Unknown]: **Servidor Tomcat***
    Â¿CuÃ¡l es el nombre de su ciudad o localidad?
    *[Unknown]: **Barcelona***
    Â¿CuÃ¡l es el nombre de su estado o provincia?
    *[Unknown]: **Barcelona***
    Â¿CuÃ¡l es el cÃ³digo de paÃ-s de dos letras de la unidad?
    *[Unknown]: **08023***
    Â¿Es correcto CN=gutzworld.sytes.net, OU=Dep. Sistemas, O=Servidor Tomcat, L=Barcelona, ST=Barcelona, C=08023?
    *[no]: **y***

    Escriba la contraseÃ±a clave para <tomcat>
    *(INTRO si es la misma contraseÃ±a que la del almacÃ©n de claves): **enKriptar***
    *Volver a escribir la contraseÃ±a nueva: **enKriptar***

- Canviem el fitxer de configuració */etc/tomcat5.5/server.xml* del tomcat per indicar-li la ubicació i la contrasenya y el keystore

```
<Connector port="8443" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        keystoreFile="/projecte/keystore" keystorePass="enKriptar"
        acceptCount="100" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS" />
```

- Creem el següent arxiu que serà l'encarregat de simular la llista de revocació:
  /proyecte/demoCA/index.txt

  Si no volem que un client es pugui connectar hem d'incloure el seu Subject.

- Per fer proves de forma local cal modificar el fitxer */etc/hosts* afegint la següent línia:

  192.168.1.50 gutzworld.sytes.net

  D'aquesta manera podem accedir des del navegador amb la direcció: *gutzworld.sytes.net,* simulant d'aquesta manera que ens connectem a la direcció pública.

CONFIGURACIÓ MYSQL I PHP

- Generar estructura de taules de Base de dades de Mysql
  - Crear el fitxer *backup.sql* amb el següent contingut:

```
-- MySQL Administrator dump 1.4
--
-- ------------------------------------------------------
-- Server version       5.0.38-Ubuntu_0ubuntu1.4-log
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
*/;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;


--
-- Create schema pidentidad
--

CREATE DATABASE IF NOT EXISTS pidentidad;
USE pidentidad;

--
-- Definition of table `administrador`
--

DROP TABLE IF EXISTS `administrador`;
CREATE TABLE `administrador` (
  `idUsuari` varchar(20) character set latin1 NOT NULL,
  `contrasenya` varchar(100) character set latin1 NOT NULL,
  `nom` varchar(200) character set latin1 NOT NULL,
  PRIMARY KEY  (`idUsuari`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

--
-- Dumping data for table `administrador`
--

/*!40000 ALTER TABLE `administrador` DISABLE KEYS */;
INSERT INTO `administrador` (`idUsuari`,`contrasenya`,`nom`) VALUES
 ('adminservicio','5f55a6ece505a982f1ea0f396442bf23','Administrador de
recursos');
/*!40000 ALTER TABLE `administrador` ENABLE KEYS */;


--
-- Definition of table `permis`
--

DROP TABLE IF EXISTS `permis`;
CREATE TABLE `permis` (
  `idRecurs` varchar(10) collate latin1_spanish_ci NOT NULL,
  `idUsuari` varchar(20) collate latin1_spanish_ci NOT NULL,
  `temps` int(10) unsigned NOT NULL,
  PRIMARY KEY  USING BTREE (`idRecurs`,`idUsuari`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

--
-- Definition of table `recurs`
--

DROP TABLE IF EXISTS `recurs`;
CREATE TABLE `recurs` (
  `idRecurs` varchar(10) collate latin1_spanish_ci NOT NULL,
```

```
  `descripcio` varchar(200) character set latin1 NOT NULL,
  PRIMARY KEY  (`idRecurs`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;



--
-- Definition of table `usuari`
--

DROP TABLE IF EXISTS `usuari`;
CREATE TABLE `usuari` (
  `idUsuari` varchar(20) collate latin1_spanish_ci NOT NULL,
  `nom` varchar(45) character set latin1 NOT NULL,
  `subject` varchar(200) character set latin1 NOT NULL,
  PRIMARY KEY  (`idUsuari`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

- Per aconseguir tenir l'estructura de la BD s'haurà de fer us de la següent instrucció:

  mysql -u root -p < backup.sql

  Ens preguntarà una contrasenya, introduïm la que vam escriure en el moment de la instal·lació: *enKriPTado*

- Configurar arxiu PHP de connexió a la base de dades
    - Editem el fixer /etc/php5/apache2/php.ini i descomentem la següent linia i afegim un 5:

      include_path = ".:/usr/share/php**5**"

    - Crear fitxer *conec.inc* que connecti amb la BD. Ho situarem a la carpeta anterior

```
<?php
function Conectarse()
{      if (!($link=mysql_connect("localhost","root","enKriPTado")))
       {  exit();
       }
       if (!mysql_select_db("pidentidad",$link))
       {  exit();
       }
       return $link;
}
?>
```

    - Reinitciem el servidor mysql: */etc/init.d/mysql restart*
    - Reinitciem apache: */etc/init.d/apache2 restart.*

- Fer segur el protocol HTML
    - Creem carpeta */projecte/pidentitat* on aniran els certificats
    - Accedim a la carpeta
    - Activem el mode ssl per Apache: ***a2enmod ssl***

      Apareixerà el missatge:

      Module ssl installed; run /etc/init.d/apache2 force-reload to enable.

    - Creem els certificats:
        - ***openssl genrsa -des3 -out server.key 1024***

Generating RSA private key, 1024 bit long modulus

.....++++++

.....++++++

e is 65537 (0x10001)

Enter pass phrase for server.key: ***enKriptar***

Verifying - Enter pass phrase for server.key: ***enKriptar***

- ***openssl req -new -key server.key -out server.csr***

Enter pass phrase for server.key: ***enKriptar***

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

*Country Name (2 letter code) [AU]:****ES***

*State or Province Name (full name) [Some-State]:****Barcelona***

*Locality Name (eg, city) []:****Mollet***

*Organization Name (eg, company) [Internet Widgits Pty Ltd]:****GutzWorld***

*Organizational Unit Name (eg, section) []:****Dep. Informatica***

*Common Name (eg, YOUR name) []:****Jesús Gutiérrez***

Email Address []:

Please enter the following 'extra' attributes

to be sent with your certificate request

*A challenge password []:****enKriptar***

*An optional company name []:****enKriptar***

- ***openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt***

  Signature ok
  subject=/C=ES/ST=Barcelona/L=Mollet/O=GutzWorld/OU=Dep. Informatica/CN=Jes\xFAs Guti\xE9rrez
  Getting Private key
  Enter pass phrase for server.key: ***enKriptar***

- Els col·loquem ens les carpetes corresponents:

  cd /projecte/pidentitat
  mkdir certs
  mkdir private

  mv server.crt certs
  mv server.key private

- Afegim el contingut just abans de l'última línia del fitxer: */etc/apache2/sites-available/default*

  SSLEngine on

  SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire

  SSLCertificateFile /projecte/pidentitat/certs/server.crt
  SSLCertificateKeyFile /projecte/pidentitat /private/server.key

- Reinitzar Apache y després connectar-se a la pàgina https://gutzworld.sytes.net i veure que carrega correctament. Cal recordar que cada cop que reinitciem el servidor HTTPS ens demanarà la Phasfrase, que és la clau que hem introduït a l'hora de generar el certificat, en aquest cas: *enKriptar.*

CREACIÓ D'USUARI ADMINISTRADOR

La forma de crear el compte que utilitzarem per a que l'administrador pugui gestionar els usuaris és una mica manual, però vàlida.

En el moment de generar l'estructura de la base de dades amb el fitxer *backup.sql* també hem aprofitat per crear un usuari en la taula d'administrador que es diu *adminservicio*.

Per poder modificar la contrasenya d'aquest usuari, he generat un codi PHP que actualitza el password i el xifra amb una funció de xifratge. A l'hora de validar un usuari, xifrem la contrasenya que ha introduït i comparem amb la que hi ha guardada en BD. Aquesta mena d'algoritmes no admeten retorn, és a dir, un cop fem el xifratge, si no sabem la clau que s'ha utilitzat per xifrar mai ho podrem desxifrar. Les passes que calen seguir son aquestes:

Hem de modificar el fitxer */var/www/index.php* i canviar el següent:

//$sql="UPDATE pidentidad.administrador SET contrasenya = '". md5("**0181**") ."' WHERE idUsuari = 'adminservicio'";

//$result=mysql_query($sql,$link);

Descomentem les línies de codi i posem la contrasenya que ens interessi. Carreguem la pàgina principal del gestor de recursos: https://gutzworld.sytes.net. Un cop carregada, ja haurà actualitzat les dades. Tornem a comentar les línies i ja està.

COPIAR EL CODI FONT DE L'APLICACIÓ

Un cop descomprimit el fitxer *gutzworld_producte.zip* haurem de copiar el següents fitxers:

- El contingut de la carpeta *web/pidentitat/php* a la ruta següent: */var/www*.
- El contingut de la carpeta *web/pidentitat/jsp* a la ruta següent: */var/lib/tomcat5.5/webapps/pidentitat*.
- El contingut de la carpeta *web/pservei/jsp* a la ruta següent: */var/lib/tomcat5.5/webapps/pservei*.

ACCÉS A L'APLICACIÓ DES DE L'EXTERIOR

- S'ha d'obrir el por 443 i el 8443 del NAT del router per que a través d'Internet siguem capaços de connectar-nos a la nostra aplicació.

- En el meu cas, el router no té una ip fixa, i per tant, pot canviar de tant en tant. Una manera de solucionar aquest problema és configurant el servei No-Ip i cada cop que canviï la ip del router actualitzi el domini gratuït *gutzworld.sytes.net* per a que apunti a la nova ip.

SERVEI FTP

- Instalem el servidor: *apt-get install vsftpd*
- Configurem el fitxer */etc/vsftpd.conf*

  # Habilitar el acceso a usuarios anónimos. Para mayor seguridad poner NO.

  anonymous_enable=NO

  # Permitir el acceso de usuarios locales a sus respectivas carpetas privadas:

  local_enable=YES

  # Permitir el modo escritura:

  write_enable=YES

  # Mascara del directorio:

  local_umask=022

  # Mensaje de bienvenida:

  ftpd_banner=Bienvenidos al Servidor FTP de este sitio.

  # Enjaula a los usuarios dentro de su propio directorio personal. Mejora la seguridad.

  chroot_local_user=YES

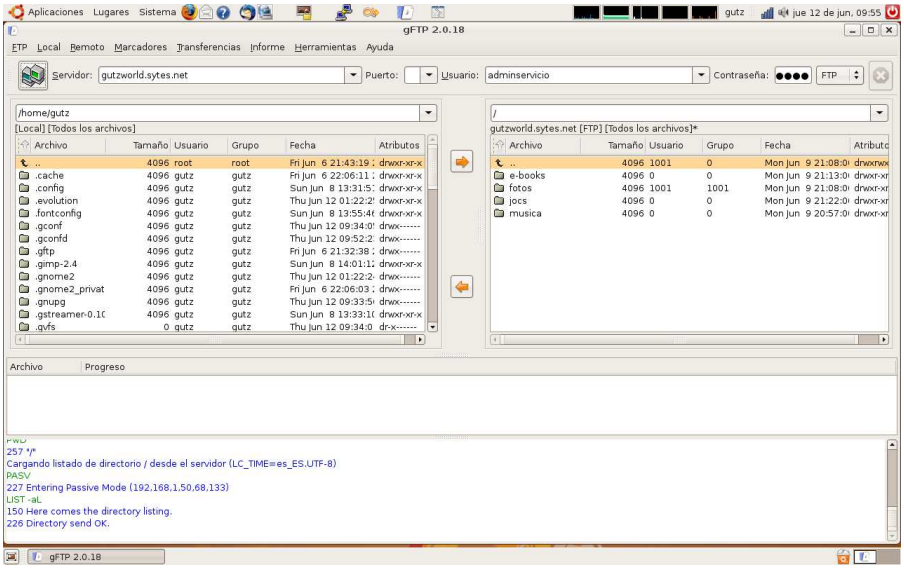
- Creem l'usuari : *adduser adminservicio.* Utilitzem el mateix passwd que hem utilitzat per a l'usuari administrador de la gestió de recursos.
- Editem el fitxer /etc/passwd

  *adminservicio:x:1001:1001:,,,:**/projecte/ftp**:/bin/bash*

- Creem el directory */projecte/ftp*
- Canviem permisos
    - *chown adminservicio:root /projecte/ftp/*
    - *chmod 770 /projecte/ftp/*
- reinitciar el servidor ftp : */etc/init.d/vsftpd restart*

GUARDAR RECURSOS AL SERVIDOR DE FITXERS

Com hem comentat en apartats anteriors, el número de recursos està limitat a quatre: fotos, musica, e-books, i jocs. Per tant, haurem de crear 4 carpetes en el servidor ftp. De tal manera que la estructura quedarà així:



Podeu treballar amb el programa gFTP. S'instal·la amb *apt-get install gftp*

Caldrà que utilitzem com a direcció ftp: *gutzworld.sytes.net* i com a usuari *adminservicio*. La contrasenya es la mateixa que hem definit durant el procés de creació d'aquest usuari.

CLIENT I ADMINISTRADOR DE RECURSOS

REQUISITS

- Un PC amb, com a mínim, 512 MB de RAM i un disc dur igual o superior a 10GB.
- Tarja gràfica amb suport d'una resolució de com a mínim 1024x768.
- Sistema operatiu Windows XP amb el Internet Explorer 7 instal·lat
- Connexió a Internet de banda ampla ADSL
- Clauer amb el certificat idCat.

INSTAL·LACIÓ DEL CERTIFICAT IDCAT

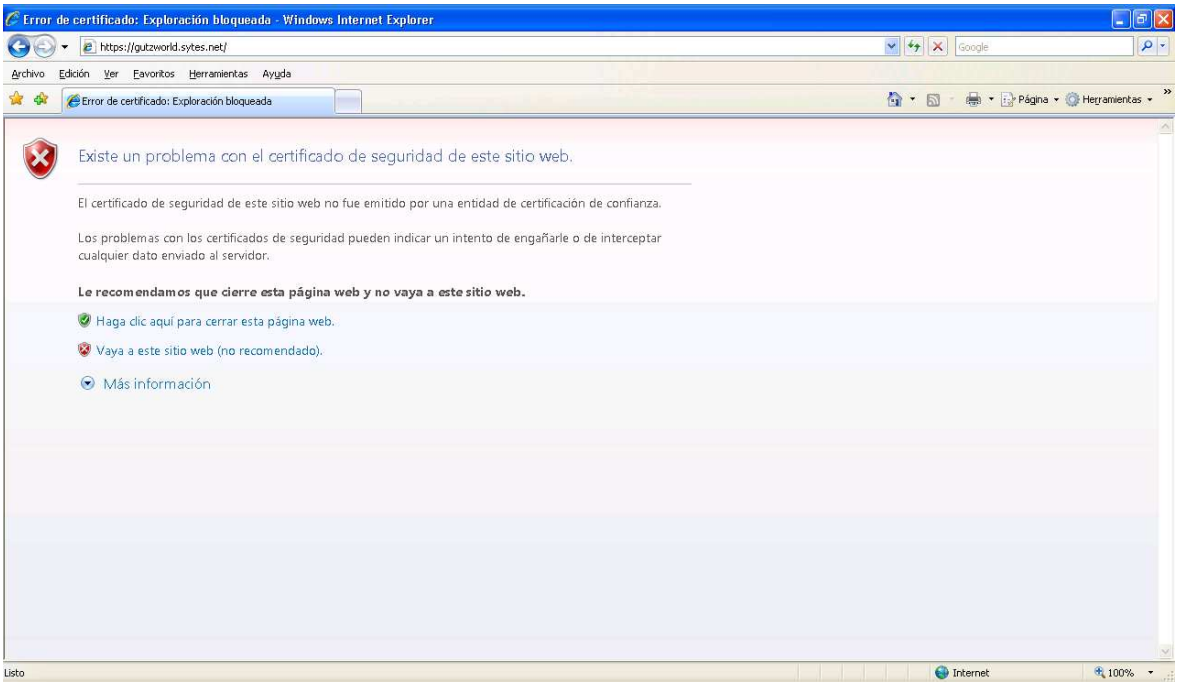
Hem de realitzar les següents passes per fer-ne ús:

- Introduïm en el clauer idCAT en el port USB de l'equip.
- Instal·lem el programari adient al sistema operatiu WinXP que trobarem dins el clauer . En aquest cas es tracta del executable: *x:\Programari clauer\windows\ setup-clauer-idCAT.exe*
- Accedim al Gestor del clauer idCAT (Inici / Programes / CATCert / Clauer idCAT) i canviem la paraula de pas actual per una de nova que ena sigui fàcil de recordar.
- A partir d'aquest moment ja podem utilitzar el vostre certificat idCAT per identificar-nos i realitzar tràmits electrònics.
- Per provar que la instal·lació ha sigut correcta disposem d'una plana web on podem signar dades amb el nostre certificat: http://www.catcert.net/web/cat/6_3_prova.jsp

GESTIONAR RECURSOS

Abans que el client pugui gaudir dels recursos, l'administrador haurà de fer una rel·lació del recursos que pot disposar cada client. Per fer-ho s'haurà de connectar a la plana de gestió de recursos https://gutzworld.sytes.net

Es normal que quan ens connectem per primer cop aparegui un missatge similar a aquest:



Això succeeix per què estem utilitzant un certificat autosignat per xifrar el protocol HTTP. Seleccionem *Vaya a este sitio web (no recomendado).*

Apareixerà una pantalla de login:

Cal ntroduïr el següent usuari i contrasenya:

usuari: *adminservicio*

contrasenya: *0181* (pot ser que sigui diferent si l'heu canviada en l'apartat d'Administrador)

Com es pot comprovar el color de fons de les pàgines és de color verd, això ens ajudarà a identificar les pàgines que són del proveïdor d'identitat.

PÀGINA D'INICI

Des de aquesta plana pot seleccionar la gestió que vol fer: usuaris, recursos, permisos

El funcionament de cadascuna de les seccions és molt similar i instituïu.

SECCIÓ USUARIS

A la plana central podem veure una sèrie d'opcions:

- Llistat d'usuaris disponible: usuaris que han sigut donats d'alta i que poden accedir als recursos del proveïdor de servei sense cap problema.

  En el llistat apareix el id d'usuari seguit del seu nom i cognoms. Si volem modificar un usuari cal fer clic sobre del seu id , es tracta d'un vincle. S'obrirà un altre pantalla on podem modificar les dades i guardar-les.

- Operacions:
  - Alta usuaris: Si seleccionem aquesta opció i polsem la fletxa que hi ha a mà dreta podrem accedir a un altre pantalla on crearem un nou usuari.

Si volem tornar a la pàgina principal, hi ha un vincle a mà esquerra, sota el títol de la secció, on posa *<< tornar*

- Eliminar seleccionats: a mà esquerra de cada element de les llistes anteriors hi ha un requadre que permet seleccionar l'element. Aquesta operació ens permet eliminar els usuaris seleccionats.

SECCIÓ RECURSOS

A la plana central podem veure una sèrie d'opcions:

- Llistat de recursos disponibles: recursos que han sigut donats d'alta i que poden ser visitats pels usuaris que estiguin donats d'alta i que tinguin associat aquest recurs.
    - En el llistat apareix el id del recurs, que ha de coincidir amb el títol de la web del proveïdor de servei que apareix sota de cada recurs, seguit de la descripció i del directori on es troben els recursos.
    - Si volem modificar un recurs cal fer clic sobre del seu id , es tracta d'un vincle. S'obrirà un altre pantalla on podem modificar les dades i guardar-les.
- Operacions possibles:
    - Alta recurs: permet accedir a un altre pantalla on crearem un nou recurs
    - Eliminar seleccionats: a mà esquerra de cada element de les llistes anteriors hi ha un requadre que permet seleccionar l'element. Aquesta operació ens permet eliminar els recursos seleccionats.

Per canvis de disseny hem decidit que els recursos sempre seran els mateixos. Per tant, haurem de crear 4 recursos. Per tant, l'apartat recurs hauria de quedar així:

Si volem tornar a la pàgina principal, hi ha un vincle a mà esquerra, sota el títol de la secció, on posa *<< tornar*

SECCIÓ PERMISOS

A la plana central podem veure una sèrie d'opcions:

- Llistat de permisos disponibles: associacions d'un recurs amb un client que han sigut donades d'alta. Això permetrà a un usuari donat d'alta que visiti el recurs associat, que també ha d'estar donat d'alta.
    - En el llistat apareix el id de l'associació, seguit del id del client, id del recurs i temps que disposa el client per gaudir del recurs. Un cop exhaurit aquest temps no podrà accedir al recurs. S'haurà de tornar a validar.
- Operacions possibles:
    - Alta permís: permet accedir a un altre pantalla on crearem una nova associació entre un client i un recurs.
    - Eliminar seleccionats: a mà esquerra de cada element de les llistes anteriors hi ha un requadre que permet seleccionar l'element. Aquesta operació ens permet eliminar els permisos seleccionats.

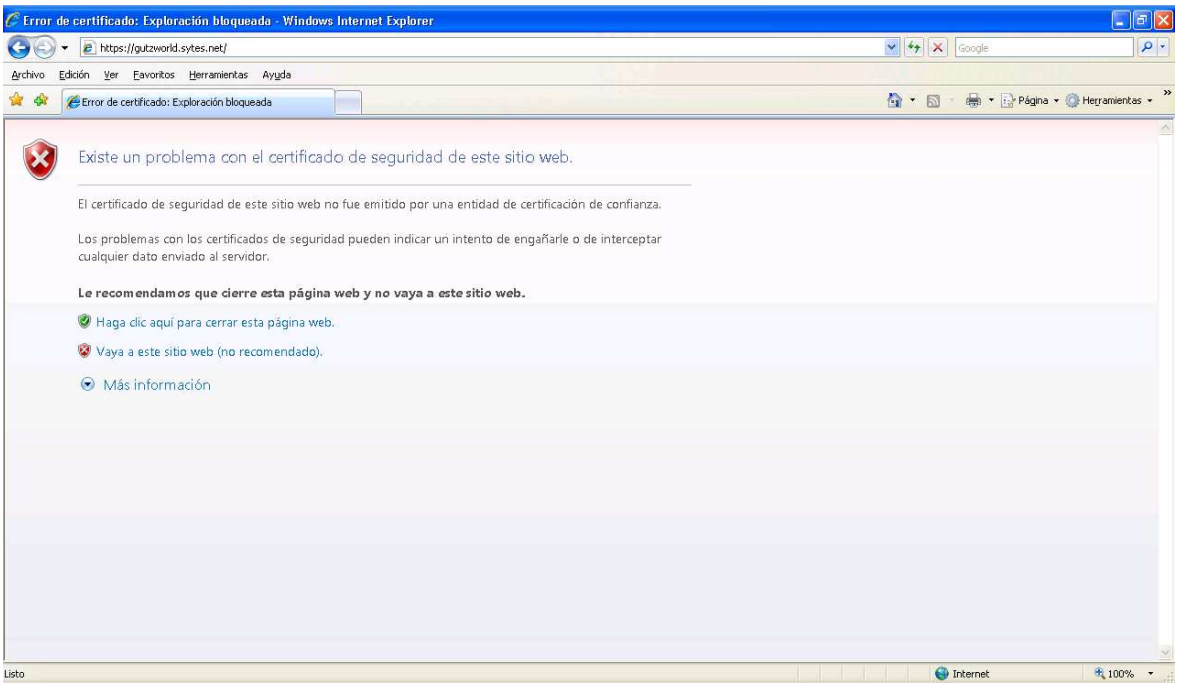
Si volem tornar a la pàgina principal, hi ha un vincle a mà esquerra, sota el títol de la secció, on posa *<< tornar*

PROVEÏDOR DE SERVEIS

El client ja pot gaudir del recursos. Cal recordar que ha de tenir inserit el clauer en el port usb, si no, no podrà validar-se amb el certificat.

La pàgina d'accés als recursos és: https://gutzworld.sytes.net:8443/pservei

Es normal que quan ens connectem per primer cop aparegui un missatge similar a aquest:



Això succeeix per què estem utilitzant un certificat autosignat per xifrar el protocol HTTP. Seleccionem *Vaya a este sitio web (no recomendado).*

PÀGINA D'INICI

Ara ja deuríem d'estar en la plana principal. Com es pot comprovar el color de fons de les pàgines és de color blau, això ens ajudarà a identificar les pàgines que són del proveïdor de serveis. A més, a dalt tenim el títol "Proveïdor de serveis" que ens ubica perfectament en quin proveïdor som.

En el centre tenim els 4 recursos disponibles. Hem de fer clic en ells per poder entrar en la secció.

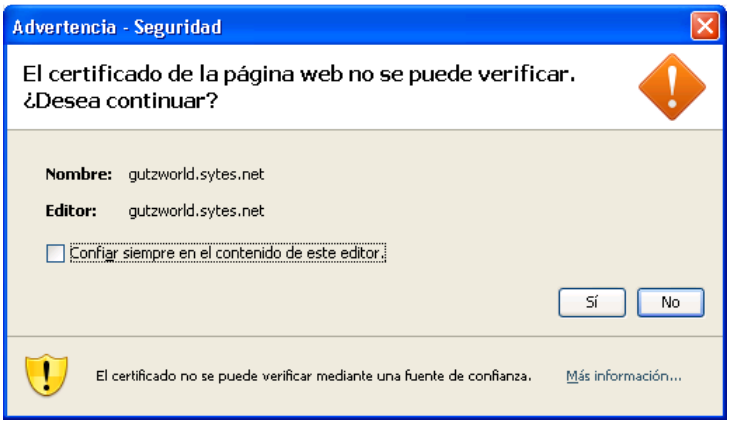
A la part baixa de la pàgina tenim un text que indica que es tracta del projecte de final de carrera i l'autor. Aquest té un vincle que permet accedir a una pàgina on es fa una breu descripció del projecte.

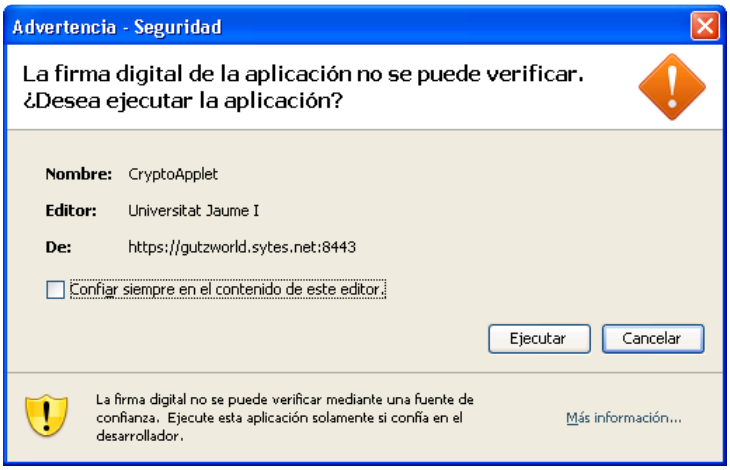
DESCRIPCIÓ DEL PROJECTE

En aquesta plana podem veure una breu descripció del projecte. Ara, la imatge amb el títol de la secció és a la part esquerra de la pantalla. Això ens serveix per identificar on ens trobem. Sota hi ha un vincle "<< tornar" que ens permet tornar a la pàgina d'inici.

PETICIÓ DE RECURS

Un cop hem fet clic en un dels recursos, apareix una finestra indicant-nos que s'executarà un programa en el nostre ordinador, li diem que si. Primer però ens indica que el programa prové d'un servidor que no és de confiança. Aquest missatge està relacionat amb el fet de que el certificat del servidor és autosignat.

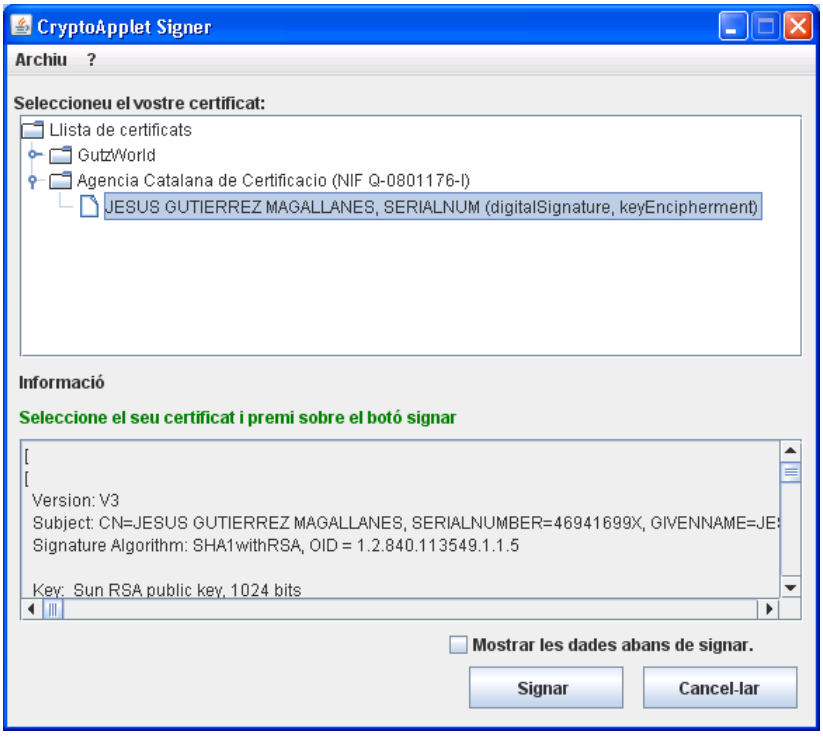




En el primer cas diem que *Sí* i en el segon *Ejecutar*

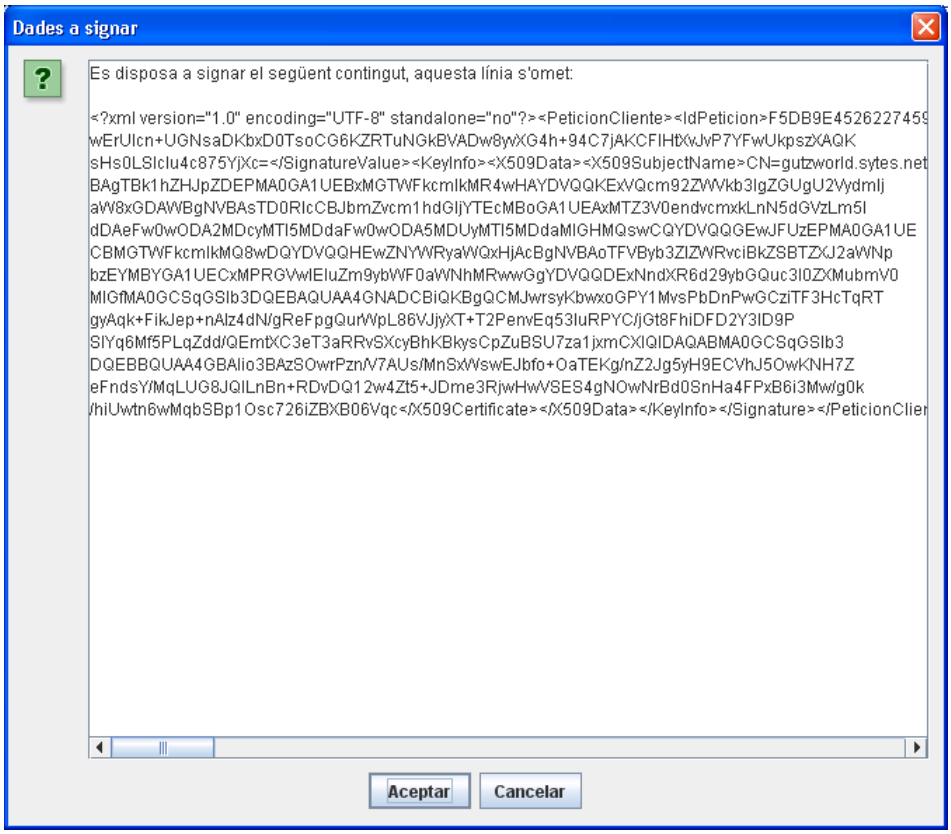
Ara apareix una finestra on indica que s'està realitzant el procés de validació:

Pocs segons després apareix una finestra on hem de seleccionar el certificat amb el que volem signar:



Seleccionem certificat idCat que tinguem instal·lat.

Si marquem la opció *Mostrar les dades abans de signar* podrem veure les dades que volem signar:



En aquesta plana apareixen les dades en XML. Si ens fixem en el codi podem extraure la següent informació:

- Identificador de petició

  <IdPeticion>F5DB9E45262274598A5E73CDBF93005B</IdPeticion>

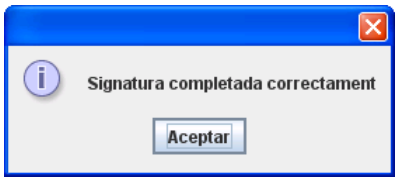
- Identificador de recurs seleccionat
  <IdRecurso>fotos</IdRecurso>

- Informació de la signatura del proveïdor d'identitat
  <X509SubjectName>CN=gutzworld.syte.net,OU=Dep Informatica,O=Proveedor de Servicio,L=Madrid,ST=Madrid,C=ES</X509SubjectName>

Fem clic en acceptar i tornarem a la pantalla de selecció de certificat. El procés de signatura començarà automàticament. Si no haguéssim marcat l'opció de *Mostrar les dades abans de signar* hauríem d'haver fet clic en el botó *Signar*.

A continuació apareix una finestra que ens demana la clau de la nostra clau privada. És la que hem introduït el moment d'instal·lació del certificat .

Si tot el procés ha sigut correcte rebrem el següent missatge:

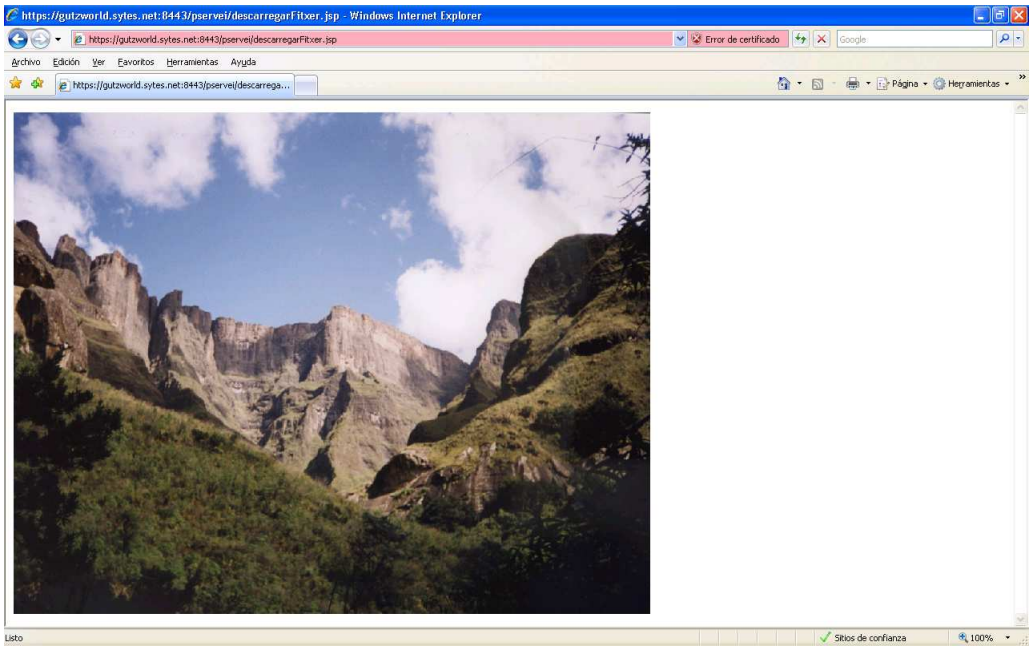


Farem clic en *Aceptar* i ja podrem accedir a l'apartat de recursos.

ACCÉS AL RECURS

Un cop dintre del recurs veurem que apareix un llistat de fotos:

Fent clic en qualsevol d'elles la podrem visualitzar:



Si volem accedir a un altre foto, hem de tirar cap enrere, i seleccionar un altre foto.

Si s'excedeix el temps màxim d'accés a un recurs rebrem un missatge com aquest:

Per seleccionar un altre recurs, un cop dintre de la secció fotos, hem de prémer el vincle << *tornar*

Es repetirà el procés de petició explicat anteriorment.

Si no tenim permís per accedir al recurs rebrem un missatge com aquest:

INFORME DE FUNCIONAMENT DEL PRODUCTE

Aquest apartat té per objectiu demostrar les funcionalitat assolides del producte. Per tant es definiran una sèrie de fluxes d'ús habituals del producte per demostrar el seu correcte funcionament

GESTIONAR RECURSOS

VALIDACIÓ ADMINISTRADOR

Des de la pàgina https://gutzworld.sytes.net

- Intentar entrar amb els camps en blanc
- Col·locar el nom d'usuari correcte però la contrasenya incorrecta i al contrari.

En tots dos casos rebo el mateix missatge: *Usuari o contrasenya incorrecta*

- Entrem a la secció *usuaris,* copiem la url,
    - https://gutzworld.sytes.net/index.php?seccion=a1c28da1af69ba622c94f7e3bd95814c
    - tanquem sessió i intentem pegar la url que havíem copiat. No podem accedir a la secció ja que hem tancat la sessió i ens hem de validar de nou.
- Intento validar-me des de dos navegadors diferents, en ordinadors diferents, amb el mateix usuari. El funcionament és correcte. Si un dels dos tanca la sessió, l'altre pot continuar treballant.

GESTIONAR RECURSOS

- Ens validem de nou, ara entrem a la secció d'usuaris i donem un d'alta. Provem de deixar cadascun dels camps en blancs, almenys un cop.

  En tots els casos, quan li donem a continuar rebem el següent missatge: *Alguna de les dades que has introduït està en blanc*

  Introdueixo totes les dades i premo Continuar. El missatge ara és diferent: *Les ddades s'han GUARDAT correctament.*
- Premo el vincle <<Tornar que es troba sota la imatge de secció, i torno a la pantalla inicial de secció. Veig com s'ha afegit el registre que acabo de crear.

- Ara intentaré afegir un usuari que tingui el mateix id o nom de certificat d'un que ja existeixi. El missatge rebut és: *El id d'usuari que has introduït ja existeix.*

- Intentem eliminar un usuari sense seleccionar cap: *No has seleccionat cap usuari*

- Eliminem un usuari, o dos i verifiquem que ho fa correctament

- Intentem eliminar un usuari que està associat a un permís. El sistema no hem deixa. Rebo el següent missatge: *Has seleccionat usuaris que estan associats a algun permís.*
- Repeteixo els fluxes anteriors per a la gestió de recursos i de permisos, i el resultat és satisfactori.

PROVEÏDOR DE SERVEIS

GESTIÓ DE RECURSOS

Per verificar que la integració entre el que defineix l'administrador i el que visualitza el client faré les següents proves:

CONTROL MÀXIM DE TEMPS D'UN RECURS:

- Associaré un recurs amb temps màxim d'un minut.

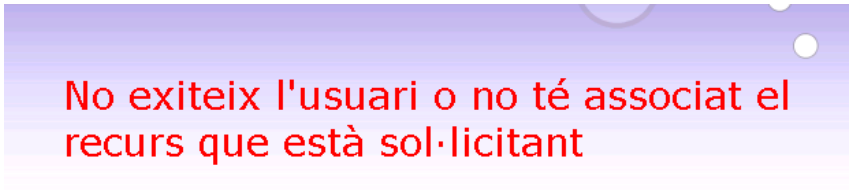
- Entraré en el recurs

- Visitaré un recurs

- Esperaré un minut
- Tornaré a visitar un altre recurs

ASSOCIACIÓ D'UN RECURS AMB UN CLIENT

- Eliminem l'associació entre el recurs *música* i el usuari *jgutz.*
- Intentem entrar un altre cop al recurs

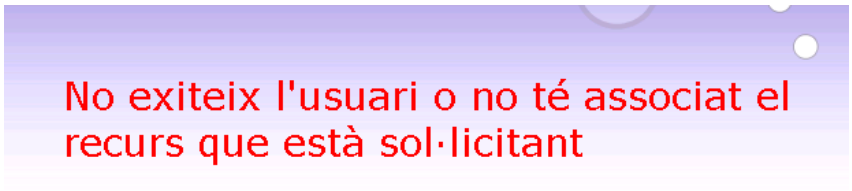


Com es pot veure, ara el client no pot accedir a aquest recurs.

COMPROVACIONS DE CERTIFICAT

- Eliminem totes les associacions creades amb l'usuari *jgutz,* l'esborrem. El creem de nou amb un Subject que no correspon amb el de la seva clau privada.

- Associem el recurs *musica* amb l'usuari que acabem de crear

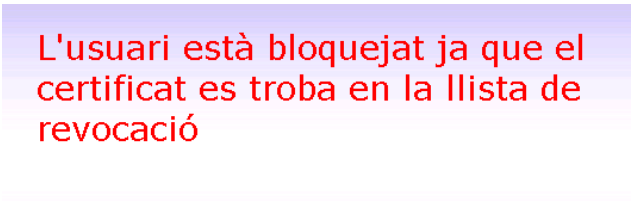


El sistema realitza la detecció de l'usuari a través del Subject. Si el Subject del certificat no apareix en la *llista d'usuaris disponibles* serà com si no estigués donat d'alta. Per tant, cal que coincideixi el Subject del certificat amb el Subject que donem d'alta.

Ara esborrarem l'associació creada i l'usuari creat. Crearem un de nou amb el Subject correcte. Associarem el recurs *e-books* amb el nou *usuari*.

Ara provarem que el control de llistes de revocació funciona:

- Editarem el fitxer /pojecte/demoCA/index.txt que es troba al servidor d'identitat, i afegirem el Subject del certificat que no volem que sigui vàlid.
- Tractarem de'accedir al recurs *e-books.*



RESTA DE COMPROVACIONS

Finalment, he fet una sèrie de comprovacions a nivell de codi, jugant amb les entrades i sortides de certes funcions Java:

- Validació de signatures

  Per exemple agafem el text que ens mostra l'applet CrytpApplet, abans de signar-lo, el copiem a un fitxer de text amb extensió xml, eliminant algun caràcter de la signatura, i se'l passem com a paràmetre al constructor de la classe Validate.

  Si cridem a la funció ValidarDoc, ens retornarà un false.

- Validació de proveïdor de servei

  El proveïdor d'identitat només espera peticions del servidor de servei *Proveïdor de Servei.* *S*i fem el mateix que en la prova anterior, però modifiquem en el xml aquesta informació de la signatura, i cridem a la funció *ServidorSoap.ObtenerRespuest(msg)*, ens haurà de retornar un missatge d'error.

CONCLUSIONS

Tal i com comentava al resum inicial, crec que el projecte que he desenvolupat pot ser molt útil de cara al futur. No només a nivell funcional si no també en quant a seguretat. Les tecnologies evolucionen i el la tendència del futur consisteix en disposar d'una eina ràpida, eficient i segura. Crec que aquests aspectes s'han complert.

Pràcticament totes les tecnologies que s'han implementat en aquest projecte eren noves per mi. Tinc experiència en la programació de pàgines web amb PHP però mai ho havia fet amb Java i Jsp. També tinc experiència en la instal·lació d'un servidor web LAMP (Linux+Apache+Mysql+PHP) però mai havia instal·lat l'aplicació Tomcat. Tampoc havia treballat amb codi XML y menys amb l' integració d'applets. En quan al llenguatge Java, l'havia tocat en assignatures de la carrera, però mai havia desenvolupat servlets. El tema de certificats també l'havia estudiat a l'assignatura de Seguretat de Xarxes però mai havia he fet ús d'ells en un programa. Per tant, he tingut que reforçar els meus coneixements en quant a servlets, XML, SOAP, Tomcat i gestió de certificats.

En resum, penso que ha sigut una experiència molt enriquidora. El treball de recerca d'informació ha sigut vital per aconseguir els objectius finals. M'ha ajudat a reforçar gran part dels meus coneixements i a aprendre noves tecnologies.

GLOSSARI

- Components
  - Client: persona que fa us del servei mitjançant el seu ordinador personal. Necessita un navegador web per accedir a la web del proveïdor.
  - Proveïdor de servei: aplicació que s'executa en un servidor a la que accedeix el client per demanar recursos. L'accés és anònim.
  - Proveïdor d'identitat: aplicació que s'executa en un servidor que s'encarrega de determinar quins recursos poden ser visitats per un client.
  - Administrador: persona que determina quins clients i a quins recursos poden accedir aquests. Es defineixen mitjançant una pàgina web que es troba allotjada al proveïdor d'identitat. La configuració definida es guarda a una base de dades allotjada al servidor.

- Peticions
  - Accés anònim: petició que fa un client a un servei sense la necessitat d'introduir nom d'usuari ni contrasenya. Això és possible gràcies a l'ús de certificats digitals, en concret el idCat.
  - Signar petició: consisteix en signar digitalment unes dades per enviar-les a un destinatari. D'aquesta manera, quan el destinatari rep el missatge pot verificar la signatura.

- Seguretat
  - Verificar signatura: quan un destinatari rep un missatge comprova, mitjançant una sèrie de mecanismes, que l'autor del missatge es qui esperava.
  - Contrafirma: consisteix en tornar a signar una petició que ja ha sigut signada.
  - Verificació de certificat: mitjançant llistes de revocació es pot determinar l'estat d'un certificat.

- Llistes de revocació (CRL): és un arxiu que conté una llista de certificats que ja no són vàlids per motius determinats, els seus números de sèrie i les dates de revocació.
- Canal segur: consisteix en xifrar la comunicació amb el protocol SSL entre el client i el proveïdor, impedint d'aquesta manera que terceres persones interceptin la informació que s'està enviant.

- Altres
  - Applet: és un component d'una aplicació que s'executa en el context d'un altre programa, en aquest cas des de un navegador web. Serà el client qui rebi aquest applet per poder realitzar comunicacions amb el protocol SOAP.
  - SOAP: permet que dos objectes en diferents processos puguin comunicar-se entre ells intercanviant dades XML. Un dels objectes es troba al client (applet) i l'altre el servidor d'identitat (servlet). La signatura també anirà inclosa en les dades XML (format XMLDSig).
  - Vmware: software que ens permet simular l'execució de diferents màquines, amb el seu sistema operatiu corresponent, en un mateix ordinador físic. En aquest cas , és on simulem l'entorn de producció.
  - Keystore: magatzem on es guarden el certificats instal·lats al sistema.

BIBLIOGRAFIA

- Tomcat y J2EE

  http://www.gestcon.com/wiki/index.php?title=TOMCAT

  http://vladimir.prie.to/content/como-instalar-tomcat-en-ubuntu

- Creació de certificat

  http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html

- Gui tools mysql

  http://dev.mysql.com/downloads/gui-tools/5.0.html

- Buscar informació de com guardar les contrasenyes amb HASH

  http://phpsec.org/articles/2005/password-hashing.html

- Sessions php

  http://www.webestilo.com/php/php12e.phtml

- HTTPS en Apache per a PHP

  https://help.ubuntu.com/ubuntu/serverguide/C/httpd.html

- Instal·lar connector MYSQL

  http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.6.tar.gz/from/pick#mirrors

- Generar sessions en JSP i obtenir el SID

  http://www.jsptut.com/Sessions.jsp

- Generar signatura y generar funció de validació per comprovar signatura

  http://www.comunidadjava.com/tec/programacionapifirmadigitaljava/

  http://java.sun.com/javase/6/docs/technotes/guides/security/xmldsig/XMLDigitalSignature.html#wp268799

- Parsejar codi XML

  http://www.onjava.com/pub/a/onjava/2002/06/26/xml.html

  http://www.desarrolloweb.com/articulos/xml-document-object-modal-java.html

- Connexió amb base de dades en JSP

  http://www.chuidiang.com/java/mysql/EjemploJava.php

- Xpath

  http://joanju.info/2008/03/15/parsear-archivos-xml-con-xpath-y-java/

  http://www.onjava.com/pub/a/onjava/2005/01/12/xpath.html

- Instal·lar servidor Ftp

  http://www.guia-ubuntu.org/index.php?title=Servidor_de_FTP