



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)

# **Mòdul d'autenticació PAPI per Drupal**

Treball Final de Màster  
Àrea web i comerç electrònic

Autor: Mateu Pascual Femenias

Consultor: Francisco Javier Noguera Otero

Gener de 2012

## Llicència de la memòria



Això és un resum fàcilment llegible del [text legal \(la llicència completa\)](#).

[Advertiment](#)

### Sou lliure de:

- copiar, distribuir i comunicar públicament l'obra
- fer-ne obres derivades
- fer un ús comercial de l'obra



### Amb les condicions següents:



**Reconeixement** — Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el licenciadador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).



**Compartir Igual** — Si altereu o transformeu aquesta obra, o en genereu obres derivades, només podeu distribuir l'obra generada amb una llicència idèntica a aquesta.

### Entenent que:

**Renúncia** — Es pot **renunciar** a alguna d'aquestes condicions si obteniu el permís del titular dels drets d'autor.

**Domini Públic** — Aquesta llicència no afecta a la situació de l'obra o algun dels seus elements quan es trobi en el **domini públic**, segons la llei vigent aplicable.

**Altres drets** — Els drets següents no queden afectats de cap manera per la llicència:

- Els vostres drets de repartiment just o **ús just**;
- Els drets **morals** de l'autor;
- Drets que altres persones poden ostentar sobre l'obra o sobre l'ús que se'n fa, com per exemple drets **de publicitat** o privacitat.

**Avis** — Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.

## Resum

El projecte “**Mòdul d'autenticació PAPI per Drupal**” és un projecte per incorporar al conegut gestor de continguts en programari lliure Drupal el sistema d'autenticació i autorització PAPI, amb el qual es pugui realitzar una autenticació Single Sign-On des d'un servidor d'autenticació extern a la plataforma i de forma distribuïda.

Un sistema d'autenticació i autorització centralitzat és aquell a on aquests processos es realitzen en el mateix element del procés sent impossible separar-los, en canvi, en un sistema distribuït l'autenticació i l'autorització es poden realitzar en elements diferents, fins i tot localitzats en diferents ubicacions, pel que l'autenticació es pot realitzar en un element i l'autorització en un altre diferent. En front a la senzillesa d'ús del primer es troba la seva limitació quan l'organització creix, sent necessari passar a un sistema distribuït que facilita l'escalat i la gestió en entorns oberts.

Drupal és un gestor de continguts que disposa d'un sistema d'autenticació centralitzat, rols i permisos que permet configurar una política d'accessos i permisos molt flexible, que es pot adaptar a la majoria de requeriments i necessitats dels usuaris i organitzacions. Drupal ofereix un conjunt d'apis que permet ampliar la funcionalitat del CMS i adaptar-lo completament a les particularitats de cada projecte, de fet, disposa d'una gran llibreria de mòduls llestos per integrar, gràcies a la comunitat que el suporta i el manté.

Entre els avantatges d'utilitzar el gestor de continguts Drupal podem destacar la facilitat de configuració, adaptació i manteniment, a més de la possibilitat que dona l'ús de les apis de programació per ampliar les seves funcionalitats i adaptar-lo a pràcticament qualsevol necessitat.

D'altra banda, PAPI és un sistema d'autenticació i autorització que proveeix una solució de seguretat als sistemes oberts. PAPI ofereix una arquitectura distribuïda que permet situar els diferents elements en diferents punts de la xarxa, això li dona una gran flexibilitat a l'hora d'integrar la seguretat en entorns d'organitzacions que no depenen d'un sol punt d'autenticació i simplifica la gestió global de la seguretat, a la vegada que ofereix una gran escalabilitat.

Els avantatges que aporta PAPI com a solució distribuïda per la autenticació i autorització dels usuaris són la independència de la tecnologia utilitzada, estar basada en l'ús de cookies, la independència de la IP d'origen, la limitació temporal d'accés als recursos, la mobilitat de l'usuari, la seva transparència, la compatibilitat amb altres sistemes de control d'accés i la compatibilitat amb els navegadors més habituals.

Amb la integració de PAPI i Drupal l'organització pot oferir una doble via d'autenticació dels usuaris, integrant d'una forma fàcil als usuaris externs dins de la seguretat de Drupal. Es pot aprofitar així els usuaris donats d'alta de forma centralitzada dins l'aplicació i també es poden integrar els usuaris externs a mesura que es van autenticant de forma distribuïda mitjançant PAPI.

Aquest projecte es publicarà al repositori de mòduls disponibles per Drupal i s'alliberarà el codi amb una llicència GNU/GPL, permetent així que la comunitat es beneficiï d'aquest mòdul i, a la vegada, que també contribueixi al seu manteniment i millora.

El mòdul complementa als ja existents que integren el sistema d'autenticació i autorització PAPI per a altres plataformes web que precisen una autenticació distribuïda. Entre aquests mòduls podem trobar plugins d'autenticació PAPI per DocuWiki , MediaWiki i WordPress.

## Índex de contingut

Llicència de la memòria.....	2
Resum.....	3
1. Introducció.....	5
Drupal: Gestor de Continguts en programari lliure.....	5
PAPI: Entorn d'autenticació i permisos.....	5
1.1 Objectius del projecte.....	6
1.1.1 Conèixer el disseny de l'arquitectura de Drupal.....	6
1.1.2 Conèixer el disseny de la infraestructura d'autenticació i autorització PAPI.....	9
1.1.3 Implementar un mòdul d'autenticació PAPI per Drupal.....	11
1.1.4 Publicar i documentar el mòdul desenvolupat baix una llicència lliure.....	12
2. Alternatives i solució triada.....	13
2.1 Gestors de continguts en programari lliure.....	13
2.2 Sistemes d'autenticació i permisos.....	13
3. Requisits, casos d'ús.....	15
3.1 Requisits del sistema.....	15
3.2 Casos d'ús.....	15
4. Interfícies d'usuari.....	18
5. Anàlisi de riscos.....	21
6. Pla de proves.....	22
7. Programari a utilitzar i llicències.....	23
7.1 Selecció de sistema operatiu i servidors.....	23
7.2 Sistema d'autenticació de Drupal.....	23
7.3 Components de PAPI.....	24
8. Planificació temporal.....	25
9. Pressupost.....	27
10. Disseny del sistema.....	28
10.1 Arquitectura de Drupal.....	28
10.2 Arquitectura de PAPI.....	30
10.2.1 Introducció.....	30
10.2.2 Missatges.....	32
10.3 Arquitectura del mòdul d'autenticació PAPI per Drupal.....	36
11. Desenvolupament.....	38
11.1 Planificació de les activitats de desenvolupament i integració del sistema.....	38
11.2 Desenvolupament.....	38
11.2.1 Preparació de l'entorn.....	38
11.2.2 Generació del codi font del component.....	42
11.2.3 Execució de les proves unitàries.....	46
11.3 Documentació.....	49
12. Implantació.....	50
12.1 Implantació del sistema.....	50
12.2 Proves.....	50
13. Manteniment.....	51
14. Conclusions.....	52
15. Llicències.....	54
16. Referències.....	55

# 1. Introducció

El projecte es basa en la integració del sistema d'autenticació i autorització PAPI [1] en el gestor de continguts Drupal [2]. Per això s'estudiarà cada un dels components del projecte per conèixer l'arquitectura i el disseny per poder-lo implantar amb èxit.

## Drupal: Gestor de Continguts en programari lliure

Drupal és un gestor de continguts en programari lliure, modular, multi propòsit i molt configurable, que permet d'una forma senzilla gestionar el contingut i el disseny d'un lloc web des del navegador web: publicar articles, imatges i altre tipus d'informació, a més de disposar d'altres serveis afegits com fòrums, blogs, galeries d'imatges, etc. Drupal, en comptes d'emmagatzemar els continguts del lloc web en arxius estàtics en el sistema d'arxius del servidor de forma fixa, organitza els seus elements i el contingut de les pàgines, mòduls i configuracions utilitzant un entorn web, emmagatzemant la informació en una base de dades, aprofitant els avantatges que aporta la web 2.0. D'altra banda, disposa d'un servei d'autenticació i assignació de permisos que permet controlar l'accés dels usuaris als diferents serveis i continguts del lloc web.

## PAPI: Entorn d'autenticació i permisos

PAPI és un sistema que proporciona un entorn d'autenticació i control d'accés per diversos serveis basats en la web i que permet controlar l'accés a recursos distribuïts en xarxa. L'autenticació de l'usuari es fa a nivell local mentre que els recursos són els qui autoritzen l'accés dels usuaris.. PAPI es basa en quatre components independents:

- **Servidor d'autenticació (AS):** Proporciona a l'usuari tots els permisos i autoritzacions als serveis que té accés des d'un servidor central.
- **Punts d'accés (PoA):** Realitzen l'autorització de l'accés actual de l'usuari als diferents serveis que ofereix l'organització, comprovant l'autenticació de l'usuari i els seus atributs.
- **Grup de Punts d'Accés (GpoA):** Permet centralitzar totes les polítiques de permisos i usuaris de l'organització.
- **Proxy PAPI:** Permet integrar en el sistema d'autenticació a recursos que només permeten autenticació per IP.

El propòsit d'aquest projecte és desenvolupar un mòdul que permeti integrar en un lloc web desenvolupat amb Drupal la infraestructura d'autenticació i autorització basada en PAPI.

## 1.1 Objectius del projecte

### 1.1.1 Conèixer el disseny de l'arquitectura de Drupal

En primer lloc coneixerem el gestor de continguts Drupal, la seva arquitectura i el sistema d'autenticació i permisos que utilitza.

Drupal és un gestor de continguts web (CMS – Content Management System) que ofereix un conjunt de funcionalitats per a la edició, emmagatzematge i publicació de diferents tipus d'informació utilitzant un servidor web i un navegador com a interfície de comunicació. Va sorgir com a alternativa a altres CMS ja existents i es va enfocar a optimitzar la combinació de flexibilitat i configurabilitat a l'hora de construir llocs webs de la forma més ràpida i senzilla possible.

La relació entre la flexibilitat d'un gestor de continguts i la seva configurabilitat és inversament proporcional [3], així, com més especialització es dona cap a un determinat sector (e-commerce, e-learning, wiki, blog, etc) més augmenten les seves prestacions i facilitat de configuració cap a aquest sector, però, a la vegada, disminueix la seva flexibilitat, el que dificulta determinades modificacions i implementacions de funcionalitats no previstes. En canvi, si el CMS és molt genèric i poc especialitzat augmenta la seva complexitat i dificultat de configuració. Per això Drupal intenta trencar aquesta relació oferint un entorn altament configurable que permet desenvolupar una gran varietat de llocs webs mantenint a la vegada una gran facilitat d'ús.

Per la realització d'aquest projecte s'utilitzarà la versió 7 de Drupal, que té una sèrie de canvis respecte a versions precedents. En aquesta versió es poden afegir camps al perfil dels usuaris i són més flexibles que els que es podien realitzar en versions anteriors.

Drupal estructura el contingut en un sèrie d'elements bàsics que formen les diferents capes que componen el gestor de continguts:

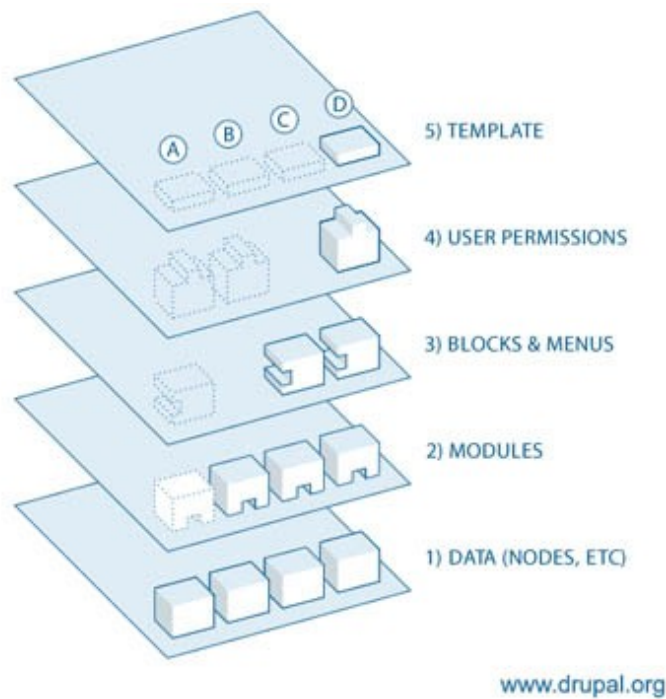
- 1. Nodes:** són els elements bàsics de Drupal, aquí és on s'emmagatzema la informació i el contingut del lloc web en el que s'anomena 'dipòsit de nodes', així com va creixent van augmentant el numero de nodes que s'emmagatzemen al dipòsit de nodes. Alguns exemples de Nodes poden ser: pàgines en llibres, temes de discussió en fòrums, entrades en blogs, nous articles, etc. Cada un d'aquests nodes té una ID única, un títol, una data de creació, un autor, un cos (body) i altres propietats depenent del tipus de node que es tracti.
- 2. Mòduls:** operen sobre els Nodes i aporten funcionalitat a Drupal permetent d'aquesta forma incrementar les seves capacitats i adaptar-les a les necessitats específiques de cada empresa o organització. Es poden considerar com a plugins que s'instal·len en el lloc web proporcionant noves funcionalitats. La instal·lació per defecte inclou una sèrie de mòduls que es poden activar o bé es poden descarregar del repositori els mòduls necessaris per cobrir les necessitats del lloc web. Existeixen tres tipus de mòduls en Drupal:
  - **Core (Nucli):** són els mòduls que venen per defecte amb Drupal. Alguns d'ells foren contribucions de la comunitat que finalment s'han incorporat com a estàndards de facto.
  - **Contributed (contribuïts):** són els mòduls que es comparteixen per a la comunitat de Drupal, tots s'alliberen sota la llicència pública GNU/GPL
  - **Custom (personalitzats):** Són mòduls creats pels desenvolupadors del lloc web.
- 3. Blocs i Menús:** Permeten estructurar i organitzar els diferents continguts del lloc web. Són els elements que allotgen els diferents mòduls i permeten accedir a l'usuari a la sortida generada i processada pels propis mòduls, a partir de la informació emmagatzemada en els nodes. L'estructura d'un lloc web desenvolupat amb Drupal consta de diferents regions, que inclouen la capçalera, el peu, les barres laterals, la secció principal amb el contingut,

etc. Dins de cada regió els blocs contenen i visualitzen la informació que proporcionen els diferents mòduls, com el menú de navegació, articles del blog, etc. En Drupal existeixen tres tipus de menú estàndard: enllaços primaris, enllaços secundaris i Menú de navegació. Els dos primers els crea i administra l'administrador o webmaster del lloc web i es mostren automàticament.

4. **Usuaris i Permisos:** Drupal disposa d'un registre d'usuaris i rols que especifiquen quines tasques poden realitzar i a quin contingut pot accedir cada tipus d'usuari, d'aquesta forma la seguretat i el control dels usuaris garanteixen la integritat de la informació emmagatzemada. Això vol dir que les operacions permeses sobre els elements que provenen de les capes inferiors (lectura, modificació, creació ...) es troben limitades per la capa de control d'usuaris i permisos de Drupal. Tots els visitants d'un lloc web Drupal es consideren usuaris, encara que no estiguin autenticats, en aquest cas l'usuari es considera anònim i se li assigna la ID 0. Els tipus d'usuari i els seus camps associats es defineixen des del nucli de Drupal, en la part de Perfil. Es poden crear diferents rols, a més dels que venen preestablerts per defecte, i després afegir o treure permisos segons la necessitat del lloc web.
5. **Plantilla:** Les plantilles o *Themes* són les que estableixen l'aparença gràfica, l'estil i els colors de la informació que es mostra a l'usuari, permetent així una total independència entre el contingut i la forma en que aquest es mostra. Aquesta separació entre informació i aspecte gràfic permet canviar el disseny i l'aspecte del lloc web sense modificar el contingut, el que és molt pràctic a l'hora de renovar la seva aparença. Les plantilles consten d'un arxiu PHP que defineix la sortida HTML i un o varis arxius CSS que defineixen les fonts, colors, posicions i altres estils.

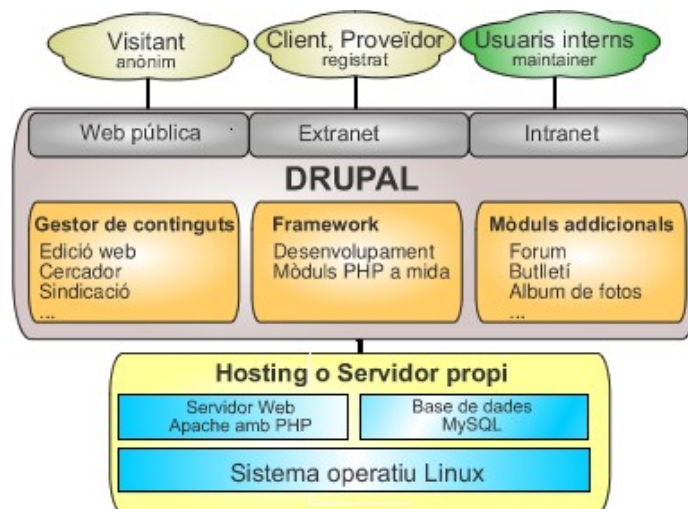
A part d'aquests cinc elements bàsics, que es poden veure a la figura següent, existeixen altres elements que complementen la funcionalitat de Drupal:

- **Cron:** Serveix per realitzar tasques bàsiques de manteniment de forma periòdica i desatesa. Es pot programar per netejar logs, re-indexar continguts, etc.
- **Taxonomia:** és el sistema mitjançant el qual Drupal classifica el contingut. És un dels mòduls del nucli de Drupal. Permet classificar els nodes segons agrupacions en categories, etiquetes, etc.



Entre les característiques de Drupal podem destacar:

- Framework i CMS molt escalable
- Optimitza els continguts per la indexació de Google i altres cercadors
- Codi font de molt bona qualitat
- API molt documentada per ampliacions a mida
- Creació i edició de pàgines des d'un editor gràfic wysiwyg
- Existeixen molts de components i mòduls addicionals
- Personalització del disseny del lloc web



La seguretat de Drupal ve donada per tres tipus d'elements:



- **Usuaris:** El mòdul user permet el registre d'usuaris, iniciar sessió i tancar sessió. Suporta rols, en els quals es defineixen els permisos que disposa cada usuari. Cada usuari pot personalitzar la seva configuració afegint camps com el correu electrònic, contrasenya, idioma, imatge a mostrar i regió.

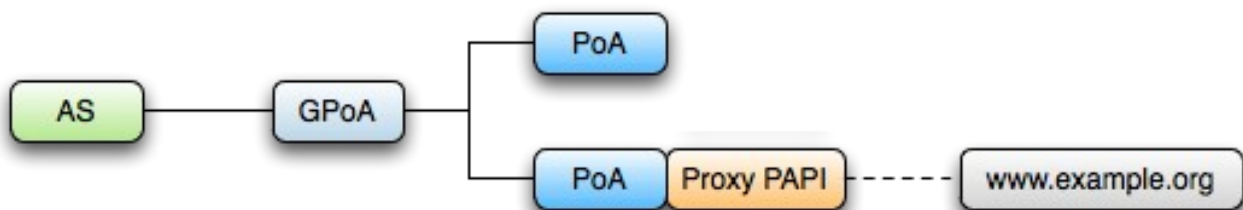
El nom d'usuari i la contrasenya s'encripten amb MD5 i s'emmagatzemen a la base de dades. Per autenticar-se l'usuari ha d'introduir el seu nom d'usuari i contrasenya, les quals son encriptades amb MD5 i comparades amb la que està a la base de dades. En cas que coincideixin el sistema li assigna una session ID única que l'identifica de forma unívoca i inicia la sessió, a la vegada que guarda la informació dins una Cookie, la qual no cap tipus d'informació personal, però actua com una clau per poder accedir a la informació emmagatzemada en el servidor.

- **Rols:** S'utilitzen per agrupar i classificar usuaris. Cada usuari pot tenir assignat un o més rols. Entre els rols per defecte es troben *anònim*, que és el que encara no s'ha autenticat iniciant sessió com a usuari; *usuari enregistrat* que és el que ja s'ha inscrit com a tal i ha iniciat sessió i *administrador*, que té tots els permisos activats. Es poden crear i modificar més rols a la pàgina de rols als quals se li assignen els permisos que es vol que tinguin.
- **Permisos:** La concessió d'un permís capacita als usuaris amb un rol concret realitzar una acció en el lloc web, per exemple, visualitzar, crear o editar continguts, administrar opcions d'un mòdul o utilitzar funcions concretes del lloc web. Quan un usuari inicia sessió se li assignen tots els permisos del rol o rols que té definit, lo que li permetrà accedir a tots els recursos dels que tingui el permís adequat.

### 1.1.2 Conèixer el disseny de la infraestructura d'autenticació i autorització PAPI

Posteriorment estudiarem l'entorn d'autenticació i permisos de Papi, quins elements ho componen, com interactuen entre ells i quins atributs s'intercanvien.

Papi és una tecnologia que permet desplegar una infraestructura d'autenticació i autorització i un sistema de Single Sign-On d'una forma fàcil i eficient, d'acord amb l'esquema següent:



Els elements que componen el sistema, com ja hem comentat anteriorment, són:

- **Servidor d'autenticació (AS)** o *Proveïdor d'identitat*, el qual és el que fa el procés d'autenticar l'usuari. D'una banda valida la seva identitat i, d'altra banda, li assigna els atributs que ha de presentar en els recursos protegits.

Les possibilitats en quant a programari que disposem a l'hora de desplegar un servidor d'autenticació PAPI són les següents:

- **AuthServer**, que és un Proveïdor d'Identitat que autenticarà als usuaris mitjançant la utilització d'un formulari HTML. Suporta diferents back-ends, com LDAP, POP3, etc. Actualment es troba en la seva versió 1.5 i està desenvolupat en Perl, amb una llicència GPL.

- **icGPoA**, que és un connector senzill que permet la integració amb la infraestructura PAPI d'un sistema d'autenticació i autorització que ja tinguem desplegat. La seva darrera versió és la 1.0 i està desenvolupat en PHP, amb una llicència GPL.
- **Punt d'Accés (PoA)**, també anomenat *Proveïdor de servei*, el qual realitza l'autorització de l'accés a un recurs protegit, comprovant l'autenticació de l'usuari i els seus atributs.

Per desplegar un Punt d'Accés que protegeixi els nostres recursos amb el sistema d'autenticació PAPI existeixen una gran quantitat de connectors per tot tipus d'escenaris i llenguatges de programació.

Entre ells podem trobar:

- Mòduls que s'integren dins del servidor Apache [4]
- Mòduls nadius per aplicacions com Wordpress [5], MediaWiki [6] o DocuWiki [7]
- Connectors nadius per CAS [8], OpenSSO [9] i simpleSAMLphp [10]

El projecte que es desenvolupa en aquest TFM és precisament la creació d'un PoA pel CMS Drupal, creant el mòdul natiu que connecti el sistema d'autenticació PAPI amb la gestió d'usuaris de Drupal.

- **Grup de Punts d'Accés (GPoA)**, que permeten centralitzar les polítiques d'autorització en un sol punt. Els diferents Punts d'Accés faran les peticions al GPoA i aquest gestionarà l'autenticació i autorització amb el Proveïdor d'identitat.

Els components que tenim a la nostra disposició per desplegar un GPoA en les instal·lacions de les empreses o organitzacions que volen protegir els seus recursos són:

- **easyGPoA**, és un component GPoA de PAPI senzill i fàcil de desplegar. Està en la versió 1.0 i està desenvolupat en PHP, amb una llicència GPL. Aquest component requereix un servidor web amb intèrprete PHP que disposi dels mòduls següents: **mcrypt** i **OpenSSL**.
- **PAPI::PoA**, Permet protegir zones del servidor Apache dins d'una infraestructura PAPI, a la vegada que inclou un Proxy de re-escritura, el qual permet als usuaris el poder accedir a un recurs extern des de les instal·lacions de l'empresa o organització. Està desenvolupat en Perl i va per la versió 1.5.2. La seva llicència també és GPL.
- **icGPoA**. També es pot utilitzar el connector icGPoA ja que ofereix una interfície de GPoA cap a un comportament que vulguem definir, donant així una solució més adaptable a les necessitats de les empreses o organitzacions que vulguin implementar un Grup de Punts d'Accés.
- **Proxy PAPI**, és un proxy HTTP amb re-escritura dels enllaços respecte a un recurs web extern. Això permet que recursos que només permeten autenticació per IP també es puguin integrar en una infraestructura d'autenticació i autorització.

Per integrar en un entorn de Single Sign-on basat en PAPI una sèrie de recursos que no siguin capaços d'integrar-se amb qualsevol tecnologia d'identitat digital, sinó que realitzen l'autorització basada en l'adreça IP de l'usuari o del seu nom d'usuari i contrasenya, s'utilitzarà un proxy PAPI.

Amb aquesta tecnologia es pot desplegar un accés protegit en el nostre entorn malgrat el recurs sigui extern.

Per a instal·lar-lo es pot utilitzar el connector **PAPI::PoA**.

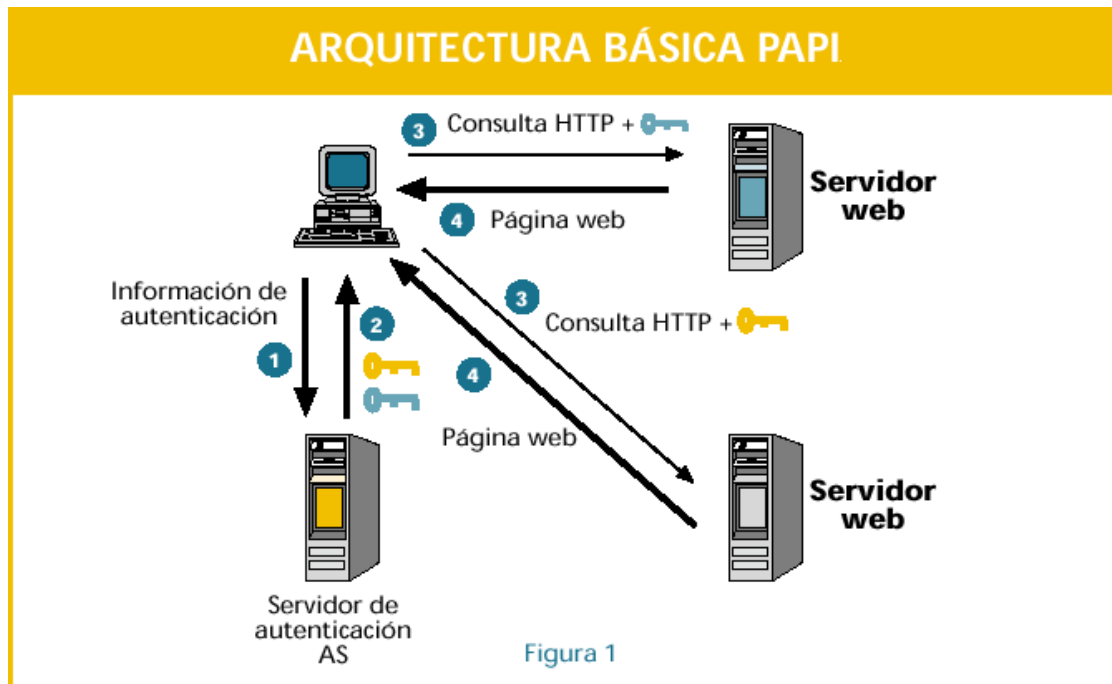
Per desplegar un sistema d'autenticació i autorització PAPI en una empresa o organització es

recomana utilitzar un **Servidor d'autenticació (AS)** i un **Grup de Punts d'Accés (GPoA)**.

En el AS es definiran els mètodes d'autenticació dels usuaris i s'emetrán els atributs de cada un d'ells, mentre que en el GPoA es definiran quins atributs s'emetrán a cada un dels recursos protegits per PAPI dins de l'organització.

Cada un dels recursos protegits hauran de tenir una relació de confiança amb el GPoA, que es definiran de forma senzilla en el desplegament de la infraestructura PAPI.

En la imatge següent es pot veure el procés d'autenticació:



1. L'AS realitza l'autenticació de l'usuari i comprova els recursos als quals té accés aquest usuari.
2. Per a cada recurs al que té accés l'usuari, l'AS li entrega una clau temporal encriptada, que li permetrà accedir a cada un dels recursos protegits mentre la clau sigui vàlida.
3. El PoA realitza el control d'accés al recurs restringit. Quan l'usuari fa una consulta HTTP al servidor amb el recurs protegit per un PoA, aquest comprova si l'usuari disposa d'una clau vàlida i si aquesta ha estat emesa per un AS vàlid, en tal cas el deixa passar i cada cert temps li entrega a l'usuari una nova clau per continuar entrant. Aquest procediment serveix perquè una mateixa clau no la utilitzin de forma simultània dos usuaris diferents, implicant així que es copii la clau i es porti a un altre ordinador. Una vegada la clau expira l'usuari s'ha de tornar a autenticar amb l'AS.

### 1.1.3 Implementar un mòdul d'autenticació PAPI per Drupal.

El pas següent és desenvolupar i implementar un mòdul per Drupal que utilitzi l'entorn d'autenticació de Papi.

Per això, una vegada analitzades les arquitectures i el funcionament de Drupal i de PAPI es

desplegarà, d'una banda, un Servidor d'Autenticació i un GPoA (o un PoA) per l'autenticació i autorització dels usuaris i, de l'altra banda, un sistema de gestor de continguts (CMS) Drupal per desenvolupar i provar el projecte en el seu conjunt.

Per realitzar aquest procés és necessari disposar d'un o varis servidors LAMP (Linux, Apache, MySQL i PHP) [11] a on es desplegaran tots els recursos necessaris per desenvolupar el mòdul i realitzar les proves.

#### 1.1.4 Publicar i documentar el mòdul desenvolupat baix una llicència lliure.

Finalment es documentarà el projecte, es triarà una llicència lliure pel codi i una altra per la memòria i s'alliberarà el projecte sota aquestes llicències.

El projecte es basa en el desenvolupament d'un mòdul per Drupal, el qual està desenvolupat en el llenguatge de programació PHP i disposa d'una llicència GNU/GPL [12]. D'altra banda, com ja hem vist abans, els diferents components i mòduls de PAPI també estan llicenciats amb GNU/GPL i desenvolupats en PHP, Perl o java, que també són llenguatges de programació lliures.

Aquesta llicència permet l'ús, estudi, modificació i distribució del codi font i de les seves modificacions, així com reconeix el dret dels usuaris a rebre o poder disposar del codi font. També obliga a redistribuir tot el codi i les seves modificacions amb la mateixa llicència GPL, pel que, en aquest cas, el projecte es publicarà baix la llicència GNU/GPL.

Pel que fa a la memòria del projecte, aquesta també s'ha de publicar sota una llicència lliure, sent les més comuns per a aquest tipus de documents les Creative Commons. (CC) [13].

Entre les diferents possibilitats que es troben en una llicència Creative Commons trobem:

- **Reconeixement (By):** En qualsevol explotació de l'obra autoritzada per la llicència s'ha de reconèixer l'autoria
- **No Comercial (Non Comercial):** L'explotació de l'obra queda limitada a usos no comercials.
- **Sense obres derivades (No Derivate Works):** L'autorització per a explotar l'obra no inclou la transformació per crear una obra derivada.
- **Compartir Igual (Share Alike):** L'explotació autoritzada inclou la creació d'obres derivades sempre que mantinguin la mateixa llicència al ser divulgades.

Aquestes quatre condicions combinades permeten escollir fins a 6 llicències diferents depenent de l'ús que l'autor vol que es faci de la seva obra.

En el cas de la memòria d'aquest projecte s'opta per una llicència de tipus BY-SA (Reconeixement i Compartir Igual), que permet l'ús comercial de l'obra i de les possibles obres derivades, sempre que la seva distribució es faci amb una llicència igual a la que regula l'obra original.



## 2. Alternatives i solució triada

### 2.1 Gestors de continguts en programari lliure

Per comparar altres alternatives pel projecte, malgrat aquest sigui exclusiu per Drupal, s'han analitzat altres gestors de contingut en programari lliure disponibles, amb l'objectiu de veure les diferències que hi pot haver entre ells i, tal vegada, servir de base per propers projectes d'implantació del sistema d'autenticació i autorització PAPI en aquests CMS.

- **Drupal.** Drupal és lliure, flexible, robust i disposa d'infinat de mòduls i complements que aporta la gent de la comunitat, repartida per tot el món. Permet desenvolupar tota classe de llocs web, ja siguin personals com empresarials i adaptar-lo a les necessitats de cada un.
- **Joomla.** un dels més populars gestors de contingut de codi obert, enfocat també al desenvolupament de forma fàcil de llocs web personals i corporatius amb una gran quantitat de mòduls, components i plugins addicionals.
- **WordPress.** una plataforma de publicació personal orientada a l'estètica, els estàndards web i la usabilitat. Està enfocat a la creació de blogs i disposa d'una gran comunitat de desenvolupadors i dissenyadors, els quals s'encarreguen de millorar i actualitzar les versions així com de crear plugins i temes d'ús lliure per la comunitat. La seva estructura es basa en la publicació d'entrades (articles) ordenats per data, així com altres funcionalitats gràcies als diferents plugins que es poden trobar.

### 2.2 Sistemes d'autenticació i permisos

- **PAPI.** un sistema d'autenticació i autorització que permet controlar l'accés a recursos distribuïts en xarxa.
- **CAS.** Central Authentication Service, que és un protocol de Single Sign-On que permet accedir a diverses aplicacions amb només una autenticació.
- **LDAP,** que permet l'accés a un servei de directori per cercar informació en un entorn de xarxa.
- **OpenId,** estàndard d'identificació digital descentralitzat, en el que un usuari s'autentica en una pàgina web i pot ser verificat per qualsevol servidor que suporti el protocol.
- **Shibboleth.** com un dels projectes de Internet2 que desenvolupa un entorn Single Sign-On que funcioni en entorns multi-organització, basat en la especificació SAM i desenvolupat pel grup OASIS.

Com que la idea del projecte és implantar un mòdul d'autenticació PAPI pel gestor de continguts Drupal és evident que per la realització i desenvolupament del projecte s'han escollit tant el gestor de continguts Drupal com el sistema d'autenticació i autorització PAPI.

D'una banda, Drupal disposa del seu propi mòdul d'autenticació i permisos i, gràcies a la col·laboració de la comunitat drupal.org s'han desenvolupat diversos mòduls que afegixen funcionalitats i més seguretat a aquest mòdul, Drupal disposa d'altres mòduls d'autenticació que es poden instal·lar sense gaire complicacions, entre ells:

- Bakery Single Sign-On System
- CAS
- LDAP
- Oauth
- OpenId

Tots aquest mòduls permeten realitzar un accés Single Sign-On, que és el que es pretén amb aquest mòdul,

integrant PAPI amb Drupal.

D'altra banda, PAPI és un entorn que proporciona accés als recursos als usuaris que ja s'han autenticat en el lloc d'origen, per tant permet utilitzar diferents recursos distribuïts sense necessitat d'autenticar-se en cada un d'ells. PAPI ja disposa de mòduls d'autenticació per altres plataformes en programari lliure, com per exemple MediaWiki, DokuWiki i Wordpress.

Amb aquest mòdul es facilitarà als usuaris de Drupal el procés d'autenticació i permisos distribuït en un entorn de xarxa amb l'ús de PAPI.

## 3. Requisits, casos d'ús

### 3.1 Requisits del sistema

Per poder implantar el sistema d'autenticació i autorització PAPI és necessari complir una sèrie de requisits que defineixen les principals característiques del sistema a nivell funcional [14]. Aquests requisits s'hauran de complir en el mòdul que es desenvoluparà en aquest projecte.

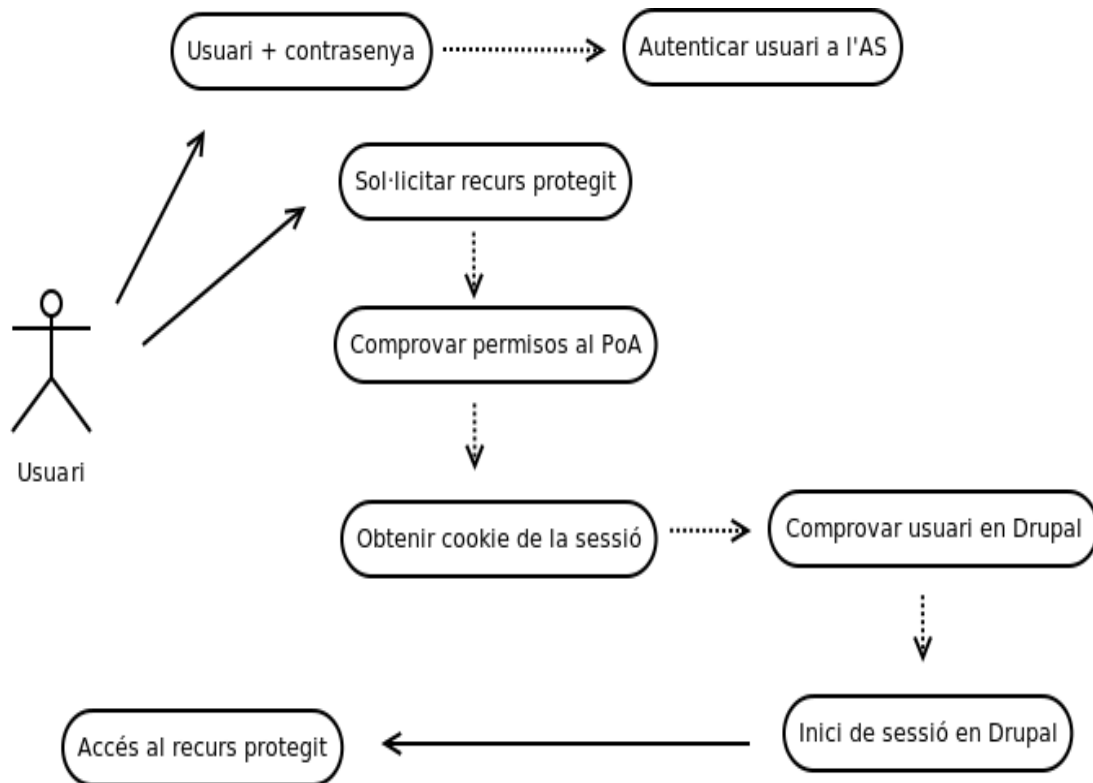
- Independència de la IP d'origen. El sistema de control d'accés ha de ser independent de la IP origen de la connexió.
- Accés als recursos durant un període limitat de temps. Una vegada autenticat l'usuari ha de tenir accés a tots els recursos dels quals disposi de permís sense que se li hagi de requerir noves autenticacions. Les sessions han de tenir una limitació de temps, transcorregut el qual sí que haurà de tornar a autenticar l'usuari. Aquest límit de temps s'ha de poder fixar de forma senzilla.
- Garantir la mobilitat de l'usuari. Els usuaris hauran de poder accedir als recursos als que estan autoritzats independentment del sistema de connexió utilitzat o de la seva localització en la xarxa. Els processos del sistema han de ser iguals tant per usuaris locals com remots.
- Transparent per l'usuari. Ha de requerir el mínim d'interacció possible per l'usuari.
- Compatible amb altres sistemes de control d'accés. El sistema s'ha de poder utilitzar conjuntament amb altres sistemes d'autenticació que ja estiguin implantats.
- Compatible amb els navegadors més usats. L'usuari ha de poder utilitzar el navegador que més li agradi, com a mínim dels més habituals: Explorer, Firefox, Chrome, Opera i Safari.
- Garantir la privacitat de l'usuari. El projecte ha d'implementar els mecanismes adequats per garantir l'anonimat en l'accés a determinats serveis.
- Facilitar les estadístiques en els proveïdors. El sistema ha de garantir que els accessos es puguin discriminar per usuari, per facilitar l'obtenció d'estadístiques d'accés fiables.

### 3.2 Casos d'ús

Els casos d'ús del sistema d'autorització i autenticació PAPI es basen en les necessitats d'accedir a un contingut protegit per un PoA o un grup de PoA (GPOA) i que l'usuari autenticat existeixi o no en Drupal

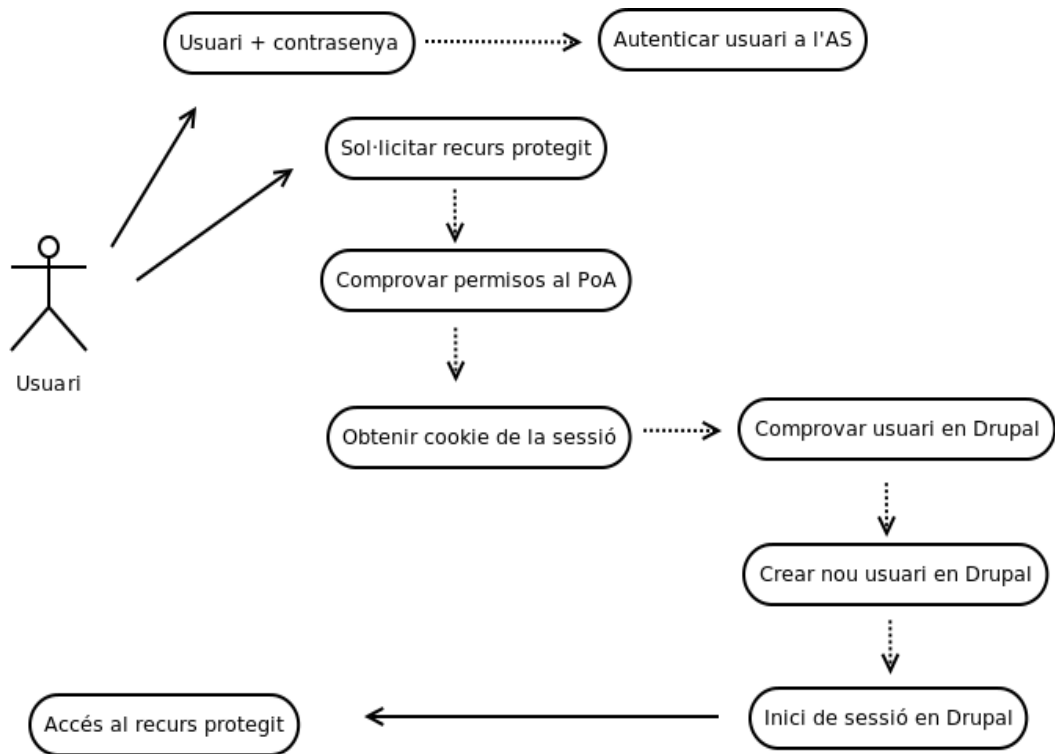
En aquest escenari es poden donar tres casos d'ús diferents:

- **Cas d'ús: Autenticació quan l'usuari existeix en Drupal.** Una vegada introduït el nom d'usuari i contrasenya es sol·licitarà l'autenticació a l'AS predefinit. Quan aquest retorna el control a Drupal es comprova que l'usuari existeix, en tal cas s'inicia la sessió i se li assignen els permisos d'acord al rol a què està assignat l'usuari en qüestió. Aquest procés d'autenticació serà totalment transparent per l'usuari. L'AS enviarà una cookie amb el contingut de l'autorització i el temps en què aquesta estarà activa.



- Cas d'ús. Autenticació quan l'usuari no existeix en Drupal.** El procés és similar a l'anterior, Una vegada introduït el nom d'usuari i contrasenya es sol·licitarà l'autenticació a l'AS predefinit. Quan aquest retorna el control a Drupal es comprova que l'usuari existeix, si no existeix el crearà i posteriorment iniciarà la sessió i se li assignen els permisos preestablerts. Aquest procés d'autenticació i creació de nou usuari serà totalment transparent per l'usuari. L'AS enviarà una cookie amb el contingut de l'autorització i el temps en què aquesta estarà activa.



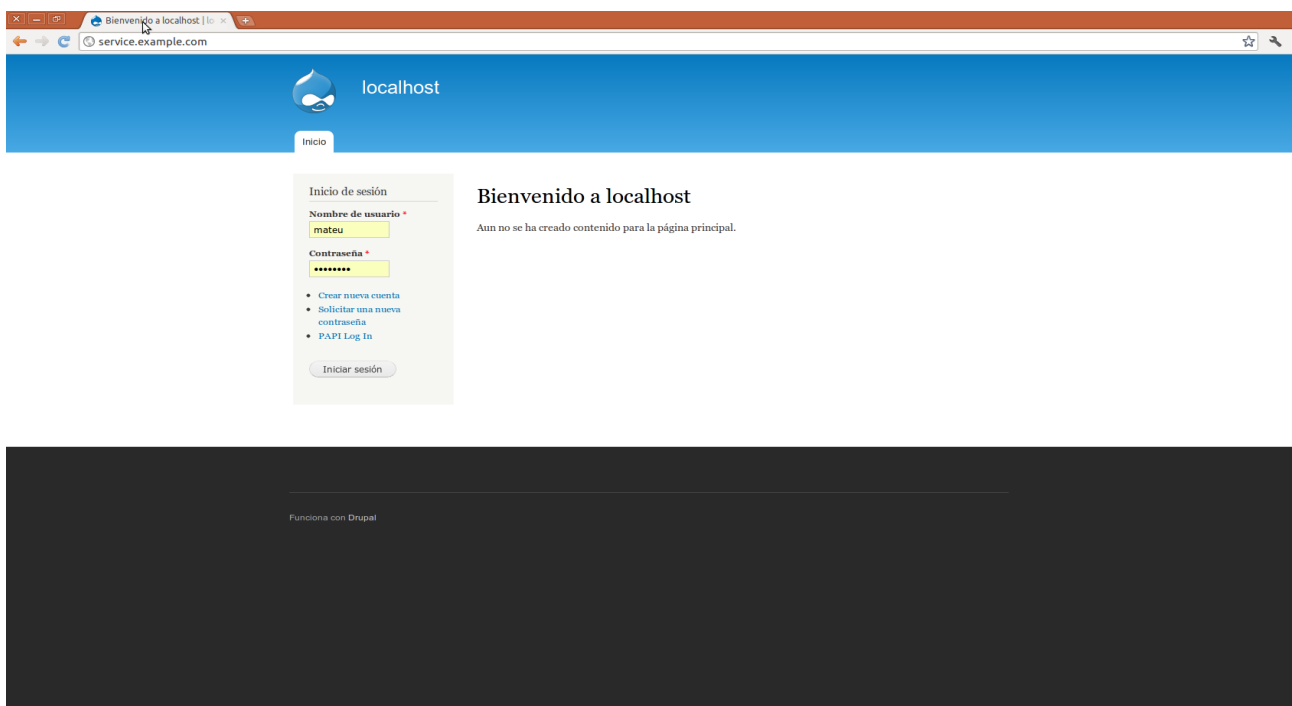


## 4. Interfícies d'usuari

Les interfícies d'usuari del projecte vindran donades per les aplicacions web que implementin aquest sistema d'autenticació i autorització PAPI per Drupal, ja que el procés d'introducció de nom d'usuari i contrasenya ve definit per defecte en Drupal i s'integra dins del *theme* de cada lloc web.

A mode d'exemple, als gràfics següents es pot veure la típica entrada de Drupal sol·licitant el nom d'usuari i contrasenya i la seqüència d'autenticació PAPI

En la imatge següent es mostra la pantalla de la pàgina d'inici de Drupal a on s'observa a la part dreta el mòdul User de Drupal que administra el registre d'usuaris i el sistema d'inici de sessió en Drupal, que conté un petit formulari per introduir nom d'usuari i contrasenya, un enllaç per crear un nou usuari, un enllaç per sol·licitar una nova contrasenya en cas d'haver-la oblidada i un enllaç per autenticar-se mitjançant el mòdul PAPI.



En aquest cas es seleccionarà l'opció *PAPI Login* que ens mostrarà la pantalla que mostra la imatge següent, a on surt un formulari per introduir usuari i contrasenya.

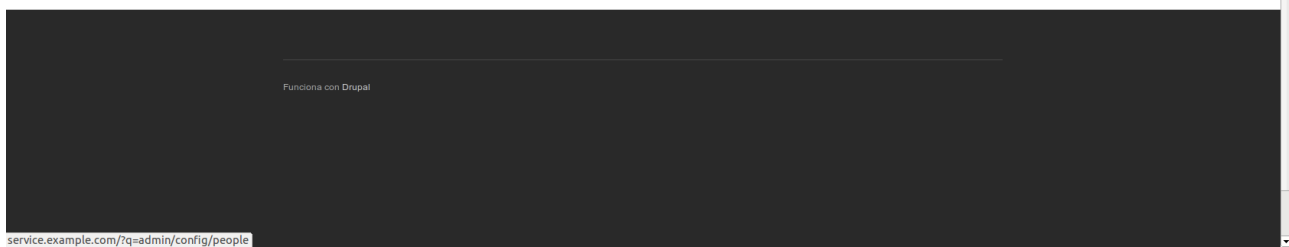
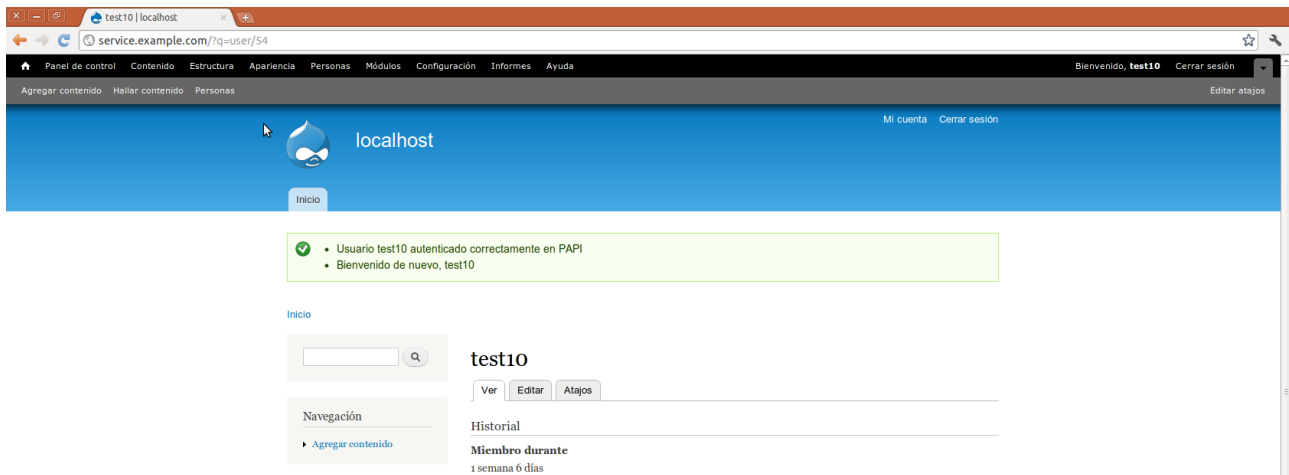
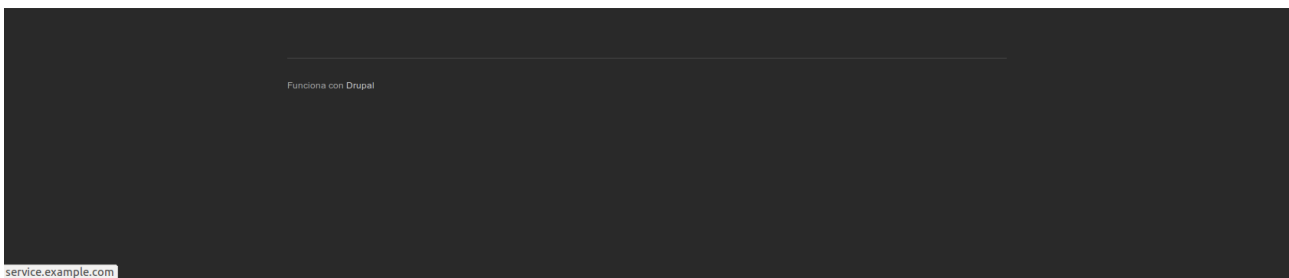
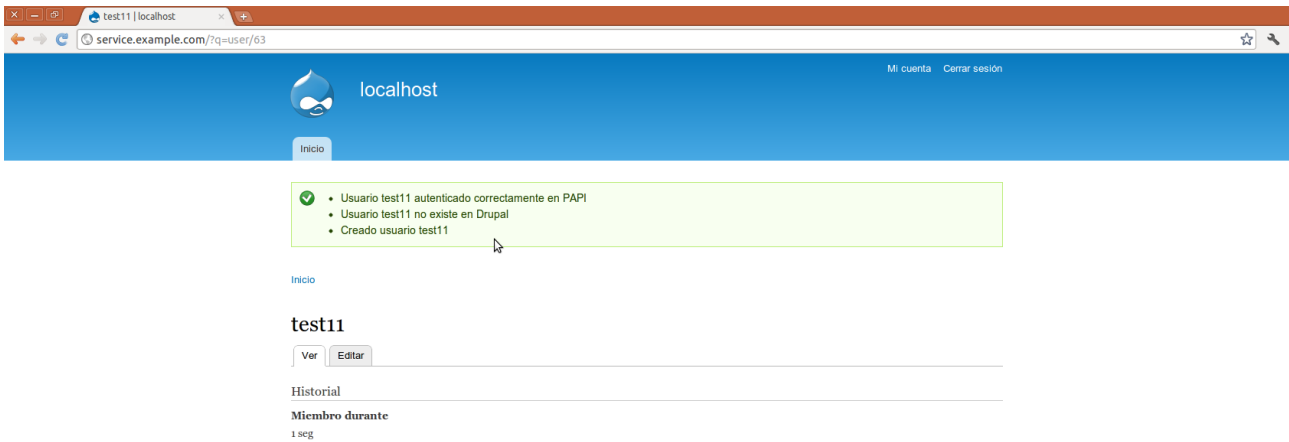


Per aquest prova s'utilitza el servidor d'autenticació icgPoA que en aquest entorn de prova valida l'usuari sempre que coincideixi el nom d'usuari i contrasenya. En entorns de producció s'utilitzarà un icgPoA real, que autenticarà als usuaris i serà el proveïdor d'autenticació qui el proporcionarà.

Si l'autenticació en el AS no és correcta es mostrarà un avís, com es pot veure a la imatge següent:



En el cas que la autenticació sigui correcta es comprova que l'usuari existeixi en Drupal, en cas de no existir el mòdul crea un usuari nou i inicia la sessió, si l'usuari ja existeix simplement inicia sessió i mostra el missatge corresponent, tal com es pot veure a les imatges següents:



## 5. Anàlisi de riscos

El desenvolupament de qualsevol projecte implica una sèrie de riscos. És important fer-ne una anàlisi i avaluació per intentar detectar els problemes que poden sorgir i la forma de resoldre'ls i valorar si el cost de que es produeixen poden fer inviable el projecte.

Els riscos que poden aparèixer poden ser de tipus econòmic, tècnic o tecnològic, organitzacional, extern, etc. Lo més important és identificar-los i quantificar-los, tenint previstes les mesures correctores que es poden aplicar per tal de minimitzar la seva influència en el desenvolupament del projecte.

Els riscos més probables que podrien aparèixer són:

- **Riscos de gestió:** Els problemes amb la planificació temporal del projecte poden afectar tant al cost econòmic com a la viabilitat del mateix. Un retard considerable en el desenvolupament del projecte pot crear problemes de credibilitat per part del client i de la comunitat, així com un increment de la tensió entre els professionals que hi estan implicats i un increment del cost final del producte.
- **Riscos tècnics:** Un plantejament equivocat del disseny o dels requeriments del projecte pot provocar problemes en el desenvolupament, implantació i manteniment posterior del mòdul. És molt important, previ a l'inici del desenvolupament, conèixer a fons el disseny de l'arquitectura de Drupal i com maneja l'autenticació i els permisos dels usuaris, així com el disseny de l'arquitectura del sistema d'autenticació i autorització PAPI. Coneixent les possibilitats i limitacions que ofereix cada producte es minimitza la probabilitat de tenir problemes.
- **Riscos de viabilitat:** Un projecte neix d'una idea, es desenvolupa i creix d'acord les expectatives i l'èxit que suposa la seva implantació. En el cas del programari lliure (i també del privatiu) un dels riscos principals és que una vegada finalitzada la fase de desenvolupament aquest no tingui una continuïtat i acabi per morir-se. El gran avantatge del programari lliure és que alliberant el codi es pot generar una comunitat d'usuaris que el mantinguin viu i vagin corregint els errors i millorant el producte.

## 6. Pla de proves

Per acabar la fase d'anàlisi es realitzarà l'especificació del pla de proves, que ens serviran per establir si el sistema compleix amb els requisits establerts en les fases anteriors. Es realitzaran proves de sistema a diferents nivells:

- **Proves unitàries.** Es comprovarà cada un dels elements que formen el projecte. D'una banda es comprovarà la implementació del sistema d'autenticació PAPI de forma independent, instal·lant un o Servidor d'autenticació i un PoA, es donaran d'alta uns quants usuaris i es configuraran una sèrie de permisos als recursos restringits. D'altra banda, s'implementarà un gestor de continguts Drupal i es comprovarà el seu funcionament i l'autenticació que porta per defecte.
- **Proves d'integració.** Es configurarà el mòdul desenvolupat en el projecte perquè s'autentiqui en el AS de PAPI i s'accedirà als recursos restringits dels PoA's definits en el punt anterior.
- **Proves de sistema.** Es comprovarà que cada un dels elements que intervenen en el sistema funcionen correctament i de forma coordinada: Sistema operatiu, servidor web, base de dades, intèrpret de PHP, OpenSSL, Servidor d'autenticació, PoA, Drupal, etc.
- **Proves d'implantació.** Una vegada tot el sistema funcioni d'una manera correcta es farà la implantació completa de tots els elements en altres servidors, per comprovar que no hi ha problemes en el procés d'implantació.
- **Proves d'acceptació.** Els usuaris finals provaran el sistema autenticant-se i accedint a recursos restringits d'acord als seus permisos. Per aquesta fase es seleccionaran una sèrie d'usuaris i es crearan les seves credencials el servidor AS. En principi aquesta prova no hauria de tenir cap complicació de cara als usuaris ja que tot el procés d'autenticació i autorització és totalment transparent per ells. Una vegada han introduït el seu nom d'usuari i contrasenya és el mòdul el que fa totes les comprovacions a l'AS i al PoA.

Els requisits necessaris per realitzar les diferents proves són:

- **Servidor web.** El servidor web estarà instal·lat en una màquina que farà les funcions de servidor web, servidor de bases de dades, PHP i Drupal. El sistema operatiu que s'utilitzarà serà una distribució GNU/Linux, possiblement Debian o Ubuntu. El servidor web ha de suportar PHP i ha de tenir activats els mòduls OpenSSL i Mycrypt, en aquest cas s'utilitzarà Apache2
- **Xarxa amb connexió a Internet.** Les proves es faran en un entorn de xarxa amb connexió a Internet, podent tenir servidors i clients a diferents ubicacions.
- **Navegadors web.** Per connectar amb el servidor els usuaris ho faran per http, utilitzant un dels navegadors més extesos en l'actualitat, com Mozilla Firefox, Chrome o Safari. Els navegadors utilitzats han de tenir suport Javascript i cookies.
- **Servidors d'autenticació i PoA.** S'instal·laran en els servidors adequats de la xarxa els components necessaris del sistema d'autenticació i autorització PAPI.

Les diferents proves es realitzaran en els diferents escenaris i casos d'ús plantejats anteriorment, els resultats dels quals es reflectiran en una taula per la seva posterior comprovació.

## 7. Programari a utilitzar i llicències

### 7.1 Selecció de sistema operatiu i servidors

El projecte s'instal·larà en un servidor accessible per donar accés als usuaris. D'una banda es necessita un servidor web per poder servir les peticions de les pàgines del lloc web. D'altra banda, al estar codificat en php el servidor web ha de ser capaç de processar aquest llenguatge. Finalment, es necessita un sistema gestor de bases de dades per mantenir el contingut del lloc web.

El servidor ha de estar en una màquina que disposi d'un sistema operatiu GNU/Linux, preferentment una versió Debian o Ubuntu, amb el servidor web Apache, base de dades MySQL i llenguatge de programació PHP amb el mòdul OpenSSL.

Tant el sistema operatiu a utilitzar com el servidor web i el gestor de bases de dades han de ser amb llicència lliure GNU/GPL, pel que les diferents opcions escollides són:

- **Sistema operatiu:** GNU/Linux Ubuntu 11.10
- **Servidor web:** Apache2
- **Base de dades:** MySQL
- **Mòdul d'encryptació.** OpenSSL

### 7.2 Sistema d'autenticació de Drupal

Drupal disposa d'un sistema d'autenticació d'usuaris que es complementarà amb el sistema d'autenticació i autorització PAPI.

Drupal accepta usuaris autenticats i anònims. Els usuaris anònims tenen el valor del seu ID a zero, mentre que els autenticats el tenen a valor 1 i els assigna els permisos mitjançant rols predefinitos. Aquests permisos es poden modificar segons les necessitats de cada moment del lloc web.

D'altra banda, com ja s'ha comentat, existeixen mòduls que afegixen més seguretat al sistema d'autenticació i que permeten un Single Sign-On basat en diferents sistemes d'autenticació, entre els que podem trobar:

- LDAP
- SAML
- Oauth
- OpenId
- Bakery

El gestor de continguts s'instal·larà en el servidor web escollit i donarà accés al seu contingut als usuaris que realment estiguin autoritzats a accedir-hi. La versió que s'utilitzarà pel desenvolupament d'aquest projecte és la Drupal 7.

Drupal està desenvolupat en PHP i pot utilitzar bases de dades MySQL i Postgress, malgrat també disposa de connectors per altres SGBD. Està publicat sota llicència lliure GNU/GPL i mantingut per una comunitat que es troba a [www.drupal.org](http://www.drupal.org).

### 7.3 Components de PAPI

Es poden trobar solucions en diferents llenguatges de programació per implantar un sistema d'autorització i autenticació PAPI. En el cas que ens ocupa Drupal està desenvolupat en PHP, pel que s'escolliran els elements de PAPI que també estiguin desenvolupats en aquest llenguatge.

Els diferents components de PAPI desenvolupats en PHP els podem veure al quadre següent:

	En distribución oficial	Servidor de autenticación	GPoA	PoA	Proxy PAPI	Tecnología
easyPoA	1.5			X		
easyGPoA	1.5		X			PHP
icGPoA	1.5	X	X			
phpPoA	1.5			X		

Aquests components, disponibles en PHP, es poden trobar en la pàgina [www.papisoftware.net](http://www.papisoftware.net) i estan distribuïts per RedIRIS [15].

El sistema PAPI es compon de:

- **Servidor d'Autenticació (AS)**, que és a on es defineixen els mètodes d'autenticació dels usuaris i els seus atributs. El component que utilitzarem en el projecte és icGPoA
- **Grup de Punts d'Accés (GPoA)** que és a on es defineixen els atributs de cada un dels recursos a protegir. El component a utilitzar serà easyGPoA.
- **Punt d'Accés (PoA)** és el que desenvoluparem al mòdul, que comprovarà l'autenticació de l'usuari i els seus atributs per poder accedir als recursos. El component a utilitzar serà phpPoA.

Tots aquests components de PAPI es distribueixen sota llicència GPL, pel que el mòdul a desenvolupar també es publicarà sota llicència GPL.

Per desenvolupar aquest projecte s'utilitzaran els següents components de PAPI:

- **icGPoA**. S'utilitzarà com a Servidor d'Autenticació. Pel desenvolupament d'aquest projecte estarà ubicat al directori arrel de Drupal, ja que l'AS ha de ser accessible, i autenticarà als usuaris que el seu nom d'usuari i contrasenya siguin iguals. Quan s'utilitzi en mode de producció s'haurà de seleccionar el Servidor d'Autenticació que l'organització hi confii.
- **EasyPoA**. S'utilitzarà com a Punt d'Accés al que se comunicarà el mòdul del projecte per autenticar un usuari. S'ubicarà en un directori no accessible directament pels usuaris, per tant, en aquest cas estarà a /var/EasyPoA.



## 8. Planificació temporal

La distribució temporal del projecte s'ajustarà a les hores totals de dedicació incloses en el conveni de cooperació educativa per la realització de pràctiques universitat-empresa, firmades en aquest cas amb l'empresa PRISE, i que comprenen 185 hores de dedicació en la modalitat 100% virtual.

Aquest projecte final de màster es realitzarà en la modalitat semestral, amb data d'inici, a efectes pràctics, en el mes d'octubre de 2011 i, tenint en compte que el treball final de màster resultant d'aquestes pràctiques haurà d'estar finalitzat abans de febrer de 2012, es preveu que les pràctiques finalitzin durant el mes de gener de 2012, tenint per tant, una duració aproximada de tres mesos.

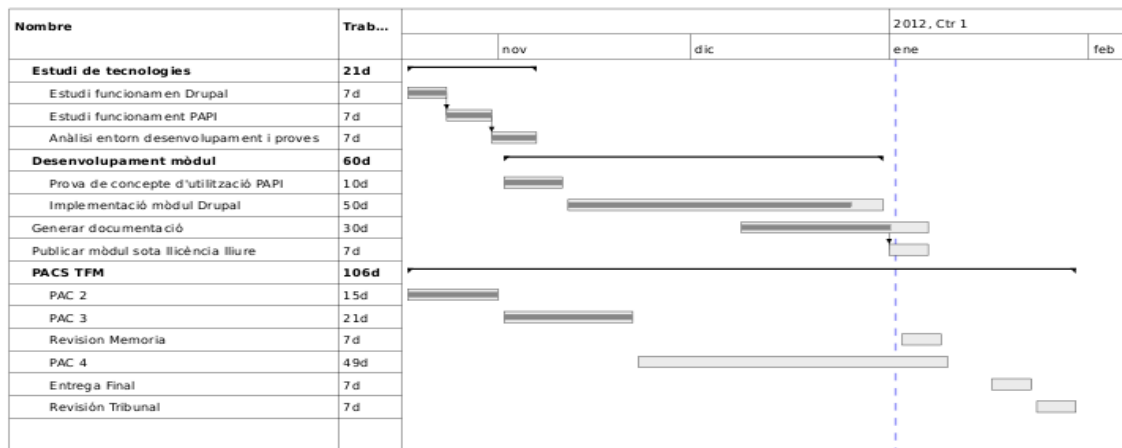
Atenint-nos a la planificació de l'assignatura Projecte Final de Màster i la seva temporització, la data de lliurament final és el dia 22 de gener de 2012, amb un lliurament de la memòria el dia 8 de gener, per tal que el consultor la revisi. Per tant, la distribució temporal de la pràctica externa s'adaptarà per què en la data de lliurament ja hagin finalitzat totes les tasques exigides i enviat al tutor de pràctiques el codi final amb el projecte i la seva documentació.

Per comprovar que el projecte s'adapta al ritme exigít s'aniran enviant regularment al tutor de pràctiques i al consultor de l'assignatura, si aquest ho requereix, el treball realitzat fins al moment. Aquestes dates no es poden especificar ja que dependrà de com avanci el desenvolupament del projecte, però se n'enviaran com a mínim dues o tres durant la fase de desenvolupament.

La planificació detallada del projecte, d'acord amb la descripció de tasques incloses a l'apartat anterior i al calendari de l'assignatura quedaria com es mostra a continuació:

- 17 d'octubre 2011: Lliurament de la PAC1 l'aula de pràctiques externes
- 23 d'octubre 2011: Final estudi funcionament Drupal
- 30 d'octubre 2011: Final estudi funcionament PAPI
- 31 d'octubre 2011: Lliurament PAC2 de TFM
- 6 de novembre 2011: Final anàlisi entorn desenvolupament del projecte
- 10 de novembre de 2011: Final prova de concepte d'utilització de PAPI
- 11 de novembre de 2011: Inici desenvolupament mòdul de Drupal
- 21 de novembre 2011: Lliurament PAC3 TFM
- 30 de desembre 2011: Final desenvolupament del projecte
- 06 de gener 2012: Final generació de documentació del projecte
- 06 de gener 2012: Publicació del mòdul sota llicència lliure
- 08 de gener 2012: Revisió memòria per part del consultor
- 09 de gener 2012: Lliurament PAC4 TFM
- 22 de gener 2012: Lliurament final

També es pot veure aquesta planificació en el següent diagrama de Gantt:



WBS	Nombre	Inicio	Fin	Trabajo	Duración	Desperdicio	Coste	Asignado a	% Completado
1	<b>Estudi de tecnologies</b>	<b>oct 17</b>	<b>nov 6</b>	<b>21d</b>	<b>20d 1h</b>	<b>84d</b>	<b>0</b>		100
1.1	Estudi funcionamen Drupal	oct 17	oct 23	7d	7d	84d	0		100
1.2	Estudi funcionament PAPI	oct 23	oct 30	7d	7d	84d	0		100
1.3	Anàlisi entorn desenvolupament i proves	oct 30	nov 6	7d	7d	84d	0		100
2	<b>Desenvolupament mòdul</b>	<b>nov 1</b>	<b>dic 30</b>	<b>60d</b>	<b>59d 1h</b>	<b>30d</b>	<b>0</b>		90
2.1	Prova de concepte d'utilització PAPI	nov 1	nov 10	10d	10d	80d	0		100
2.2	Implementació mòdul Drupal	nov 11	dic 30	50d	50d	30d	0		90
3	Generar documentació	dic 8	ene 6	30d	30d	16d	0		80
4	Publicar mòdul sota llicència lliure	dic 31	ene 6	7d	7d	23d	0		0
5	<b>PACS TFM</b>	<b>oct 17</b>	<b>ene 29</b>	<b>106d</b>	<b>104d 1h</b>		<b>0</b>		0
5.1	PAC 2	oct 17	oct 31	15d	15d	90d	0		100
5.2	PAC 3	nov 1	nov 21	21d	21d	69d	0		100
5.3	Revision Memoria	ene 2	ene 8	7d	7d	21d	0		0
5.4	PAC 4	nov 22	ene 9	49d	49d	20d	0		0
5.5	Entrega Final	ene 16	ene 22	7d	7d	7d	0		0
5.6	Revisión Tribunal	ene 23	ene 29	7d	7d		0		0

## 9. Pressupost

El projecte es desenvolupa íntegrament utilitzant programari lliure i de codi obert, pel que el cost de les llicències és nul.

El servidor web a on s'allotja el gestor de continguts també utilitza programari lliure, així s'utilitzarà un sistema LAMP (Linux, Apache, MySQL i PHP), pel que el cost de les llicències també serà nul.

El cost total del projecte vindrà donat pel cost de la ma d'obra utilitzada. En aquest cas s'estima que les hores totals que cal dedicar a l'estudi, execució i posada en funcionament del projecte serien d'unes 185. El cost/hora pot variar entre 40-70€.

El detall del cost i les hores utilitzades podria ser aquest:

- cost del maquinari del servidor: Variable, segons el preu de mercat actual de servidors. Un servidor amb un processador core i5, 8GB de memòria i disc de 2 TB podria estar al voltant del 2.000€
- Cost de llicència del sistema operatiu: nul
- Cost de llicència del servidor web: nul
- Cost de llicència del sistema gestor de bases de dades: nul
- Cost de llicència del
- Cost de llicència Drupal: nul
- Cost del sistema d'autenticació PAPI: nul
- Estudi de tecnologies: 30 h.
- Anàlisi entorn desenvolupament i proves: 14 h.
- Desenvolupament del mòdul: 100 h.
- Generar documentació: 30 h.
- Implementació del mòdul sota llicència lliure, proves unitàries i d'integració: 12 h.

Aquests costos són estimatius a falta d'un pressupost rigorós i un estudi del mercat informàtic per valorar el cost del maquinari i el cost/hora de la ma d'obra necessària per desenvolupar el projecte, tenint en compte que no és un projecte comercial sinó que es tracta d'un treball acadèmic.

## 10. Disseny del sistema

Abans de començar amb el desenvolupament i implantació del projecte estudiarem el disseny i l'arquitectura de PAPI i de Drupal.

### 10.1 Arquitectura de Drupal

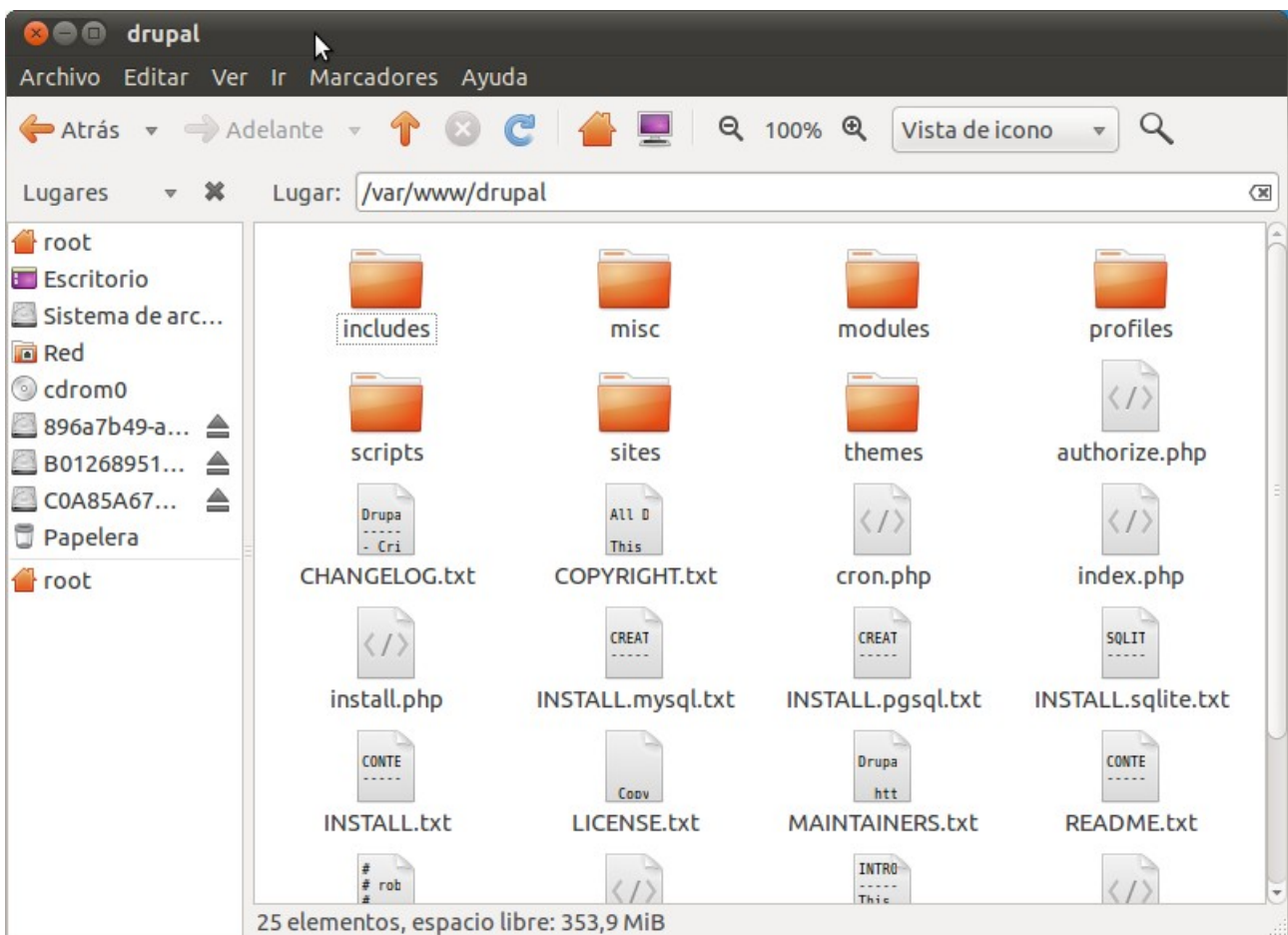
Drupal és un gestor de continguts en programari lliure que permet la creació i manteniment de llocs web d'una forma senzilla i pràctica. Està desenvolupat en php i el seu codi està alliberat en el portal [www.drupal.org](http://www.drupal.org).

A l'apartat 1.1.1 hem vist l'arquitectura i el disseny dels components de Drupal i com es relacionen entre ells. En aquest apartat em centraré en la part dels mòduls contribuïts. Com estan estructurats els mòduls i com es pot interactuar amb el sistema mitjançant aquestes apis és lo que analitzaré a continuació.

Drupal s'instal·la en un directori accessible web i conté una sèrie d'arxius i carpetes amb el codi, imatges i arxius de configuració necessaris pel seu funcionament, amb la següent estructura de arxius:

- **includes:** conté el nucli de Drupal amb les llibreries de funcions comuns.
  - **includes/database:** conté el codi per la lògica del sistema d'abstracció de bases de dades
  - **includes/filetransfer:** conté el codi necessari per realitzar transferències d'arxius mitjançant ftp i ssh.
- **misc:** conté els fitxers javascript i icones disponibles per a utilitzar-los en l'instal·lació de Drupal
- **modules:** conté els mòduls del nucli de Drupal.
- **profiles:** conté els diferents perfils d'instal·lació per un lloc web. L'objectiu d'un perfil d'instal·lació és habilitar els diferents mòduls del nucli segons les necessitats de cada lloc web.
- **scripts:** conté scripts per executar diferents tasques predefinides, com per exemple, comprovar sintaxis, cron, etc.
- **sites:** conté les configuracions i modificacions realitzades mitjançant el formulari de configuració de mòduls o temes.
  - **sites/default:** manté la configuració per defecte del lloc web, a l'arxiu `default.settings.php`
    - **sites/default/files:** conté els arxius que es puguen al lloc web, com imatges, texts, documents, etc. Ha de tenir permisos d'escriptura pel servidor.
  - **sites/all:** Conté informació, arxius de configuració i codi dels mòduls i temes que no pertanyen al nucli de Drupal, d'aquesta manera es separen del mòduls i temes propis de Drupal i que varien en cada nova versió.
    - **sites/all/modules:** conté els mòduls, contribuïts o personalitzats, que hem instal·lat o modificat al lloc web.
    - **sites/all/themes:** conté les plantilles gràfiques, contribuïts o pesonalitzats que hem instal·lat o modificat al lloc web.

- **themes:** conté el motor de les plantilles gràfiques per defecte de Drupal.
- **cron.php:** arxiu en codi php que s'utilitza per realitzar tasques periòdiques de manteniment i càlcul d'estadístiques.
- **install.php:** arxiu d'instal·lació i configuració de Drupal.
- **update.php:** modifica l'esquema de la base de dades després d'actualitzar la versió de Drupal.
- **robots.txt:** implementa l'exclusió estàndard de robots.
- **authorize.php:** administra operacions sobre fitxers autoritzats.



Drupal necessita una base de dades per gestionar tota la informació i contingut del lloc web. Està preparat per connectar-se a diferents bases de dades, entre les quals es troben MySQL, que és la que utilitzaré en aquest projecte i la que normalment es sol utilitzar, PostgreSQL i SQLite. Per la instal·lació i configuració de cada una d'aquestes bases de dades hi ha un arxiu d'ajuda al directori arrel de Drupal, i són aquests:

- INSTALL.mysql.txt
- INSTALL.pgsql.txt
- INSTALL.sqlite.txt

Drupal disposa d'un sistema d'apis de programació que realitzen la majoria de funcions bàsiques i d'intercanvi d'informació entre el nucli i el programador. Una API és una interfície de programació

d'aplicacions i està composta de llibreries de funcions que donen accés a determinades dades o funcionalitats del sistema sens necessitat de conèixer la seva estructura interna.

Les api de Drupal [16] aporten un sistema de comunicació senzill i fiable amb els diferents mòduls del nucli de Drupal, amb el sistema gestor de la base de dades. Consisteixen en una sèrie de 'hooks' que consisteixen en funcions PHP que permeten als programadors interactuar amb el sistema, reaccionar als events i personalitzar el comportament de Drupal. Amb aquest sistema es poden modificar comportaments i funcionalitats de Drupal sense necessitat de modificar els arxius del nucli de Drupal. Les api es localitzen al directori *includes* i formen part del nucli de Drupal. Es troben en forma de mòduls escrits en php amb l'extensió *.inc*.

Entre les apis més interessants trobem:

- Module system. Permet als mòduls interactuar amb el nucli de Drupal.
- Database abstraction layer. Permet utilitzar diferents servidors de base de dades sense necessitat de modificar el codi dels mòduls.
- Menu system. Defineix els menús de navegació i redirigeix les peticions de les pàgines cap a lese URL corresponents.
- Form generation. Funcions per processar i mostrar formularis HTML.
- File upload system. Funcions per manejar el sistema d'arxius i pujar o baixar arxius.
- Field API. Per adjuntar camps de dades personalitzats a les entitats de Drupal.
- Search system. Per manejar el sistema de cerca global de Drupal.
- Node access system. Funcions per determinar els drets d'accés dels usuaris a cada node.
- Theme system. Funcions i plantilles per la interfície d'usuari i la seva implementació en temes.

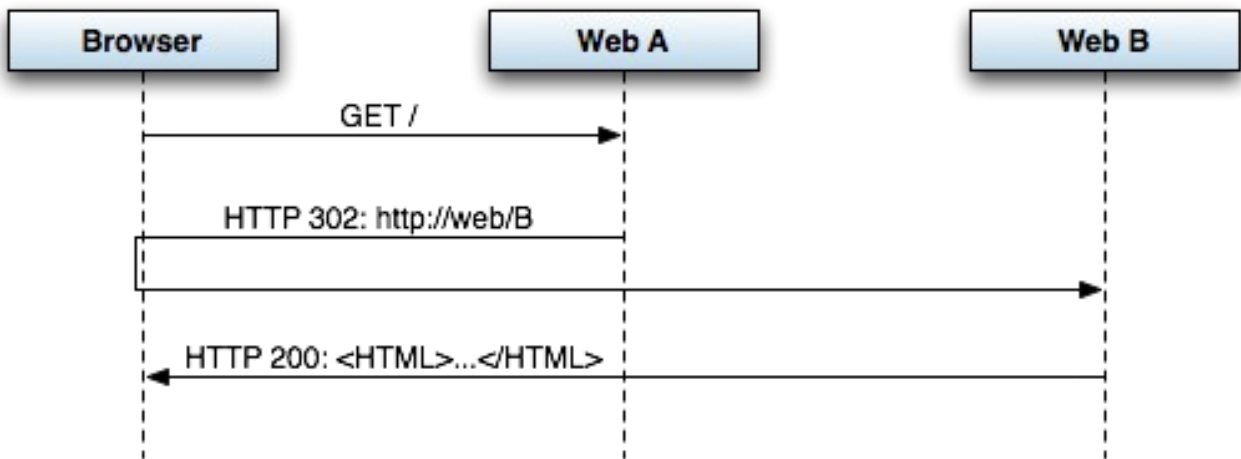
Drupal també te una sèrie de constants predefinides i variables globals que es poden utilitzar directament en tots els mòduls.

## 10.2 Arquitectura de PAPI

### 10.2.1 Introducció

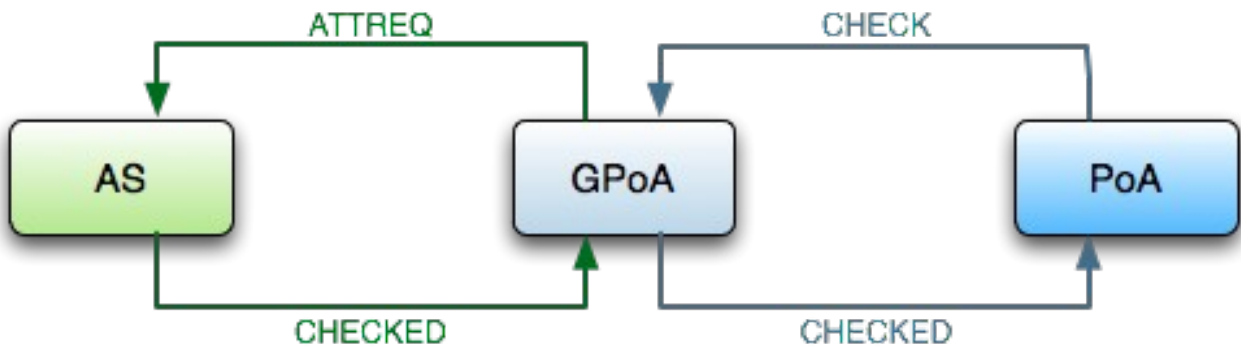
PAPI és un sistema que facilita l'accés, a través d'Internet, a recursos d'informació restringits a usuaris autoritzats. Disposa de mecanismes d'autenticació i autorització que identifiquen als usuaris i retornen el control al sol·licitant de l'accés. Aquests mecanismes s'han dissenyat per tenir la màxima flexibilitat a fi de proporcionar a les organitzacions que utilitzin un esquema d'autenticació propi i d'aquesta manera les dades es mantenen dins del seu propi àmbit, a la vegada que els proveïdors d'informació disposen de les dades suficients per poder realitzar estadístiques.

Els mecanismes de control d'accés són totalment transparents per l'usuari i compatibles amb la majoria de navegadors web i sistemes operatius. PAPI utilitza procediments HTTP estàndards, pel que es pot utilitzar per la pràctica totalitat de sistemes que requereixen serveis d'identitat digital i control d'accés. No requereix l'ús de maquinari o programari específic, garantint així que els usuaris puguin accedir des de qualsevol lloc a recursos d'informació dels que tinguin els permisos adequats.



Els missatges del protocol de PAPI es transmeten utilitzant mètode GET o POST sobre HTTP mentre que, d'altra banda, les respostes seran redireccions 302 HTTP, incloent a més cookies en el cas que l'usuari hagi d'emmagatzemar algun tipus d'informació útil pel protocol.

Al diagrama següent es pot veure el conjunt de missatges que defineixen el protocol de PAPI:



Tots aquests missatges no són enviats directament entre els diferents components, sinó que la comunicació es basa en el navegador de l'usuari. D'aquesta forma l'usuari, per accedir a un component B des d'un component A. En la seqüència següent veurem les peticions que s'intercanvien el navegador de l'usuari i els diferents components A i B que intervenen en el procés

1. El navegador envia una petició al component A

```
http://web/A?arg1=X&arg2=Y
```

2. El component A retorna la resposta següent:

```
HTTP/1.1 302 Found
Date: Sun, 02 Mar 2008 10:16:02 GMT
Server: Apache/1.3.31 (Unix)
Location: http://web/B?arg3=Z
Transfer-Encoding: chunked
```

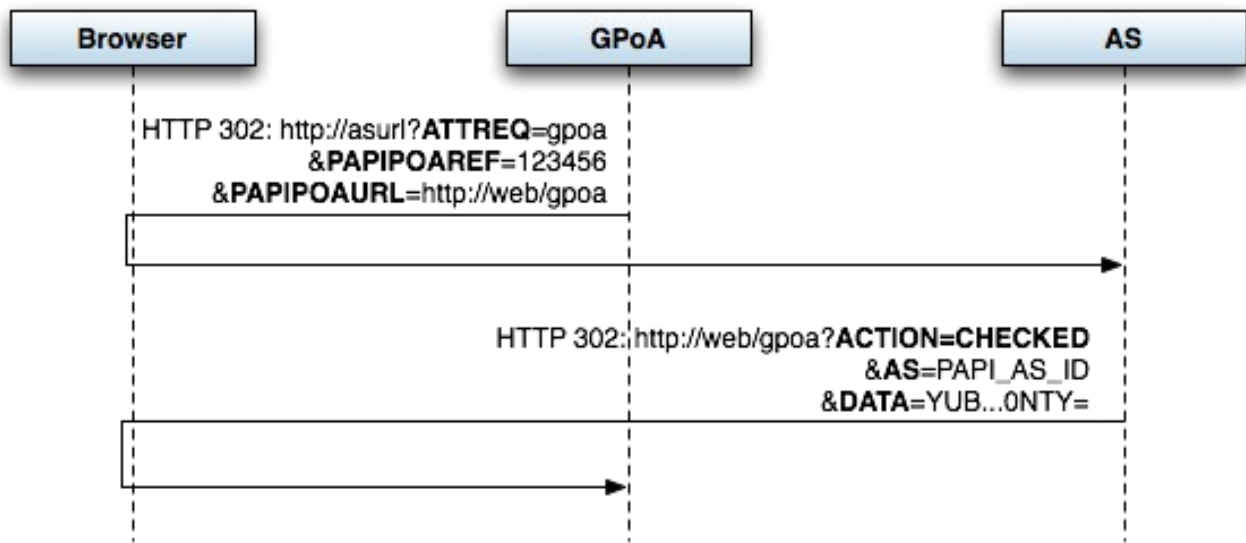
- El navegador, al rebre un missatge de resposta 302 torna a enviar automàticament una nova petició, la que s'indica en la capçalera Header, en aquest cas:

`http://web/B?arg3=Z`

### 10.2.2 Missatges

#### Petició d'autenticació: ATTREQ - CHECKED

Correspon a una petició d'atributs per part d'un component PAPI, normalment un GPoA o un PoA. En el cas que l'usuari no s'hagués autenticat prèviament, l'autentica per poder obtenir els seus atributs.



#### Missatge ATTREQ

El format del missatge és el següent:

`http://asurl?ATTREQ=%PoA-Service-ID%&PAPIPOAREF=%value%&PAPIPOAURL=%value%`

a on:

- ATTREQ** tindrà el valor de l'identificador (ID) del GPoA que ha realitzat la petició.
- PAPIPOAREF** és un valor que estableix el GPoA i que li serà retornat en la resposta.
- PAPIPOAURL** és la URL original que ha generat la redirecció a l'AS. Aquest paràmetre és molt important ja que permet a l'AS establir quina asserció ha de retornar i a quina URL ha de generar la resposta (mitjançant una redirecció HTTP).

#### Resposta CHECKED a un ATTREQ

Les respostes que emet un AS a aquest tipus de missatge són:

- En el cas que l'usuari s'hagi autenticat correctament, es retorna una redirecció 302 HTTP amb el format següent:



```
%PAPIPOAURL%?AS=%AS-ID%&ACTION=CHECKED&DATA=%value%
```

a on

- **AS** és la ID de l'AS que ha enviat la resposta.
- **ACTION** és el tipus de resposta que envia, en aquest cas CHECKED.
- **DATA** és el valor que conté la informació que requeria el POA.
- En el cas que l'usuari no s'hagi autenticat correctament o s'hagués produït un error, es retorna una redirecció 302 HTTP amb el mateix format però el camp DATA conté un error que informa al PoA de l'error. Com es pot veure, en el paràmetre DATA d'una resposta es retornen els atributs de l'usuari, també anomenats assercions d'usuari. L'AS de PAPI permet que l'asserció sigui diferent per a cada un dels PoAs, amb el que s'evita enviar més atributs i més informació de la necessària pel control d'accés en el recurs protegit. El valor de DATA s'obté de passar a base 64 el valor obtingut de xifrar la següent cadena de text amb la clau privada RSA de l'AS.

```
userAssertion:expiryTime:currentTime:KEY
```

a on

- **userAssertion** és l'asserció de l'usuari pel component que ha enviat la petició. El format de l'asserció és el següent:  
%LISTA\_ATRIBUTOS%@%AS-ID%
- **expiryTime** és el temps en segons des de l'1 de gener de 1970 en el que l'asserció deixa de ser vàlida.
- **currentTime** és el temps en segons des de l'1 de gener de 1970 en que ha estat generada l'asserció.
- **KEY** és el valor del paràmetre PAPIPOAREF que es va rebre en el missatge ATTREQ.

En el cas que l'AS vulgui indicar que l'usuari no té autorització per a accedir al GPoA o per indicar que s'ha produït un error en l'autenticació el valor de userAssertion ha de ser 'ERROR'.

La llista d'atributs (%LISTA\_ATRIBUTOS%) en una asserció no té un format definit, però la informació que conté segueix l'estructura següent:

- Nom d'atribut
  - Llista de valor de l'atribut

Dins d'una mateixa federació o Single Sign-On s'ha d'establir un acord sobre els separadors de cada un dels elements que componen l'asserció. Es recomana utilitzar els següents:

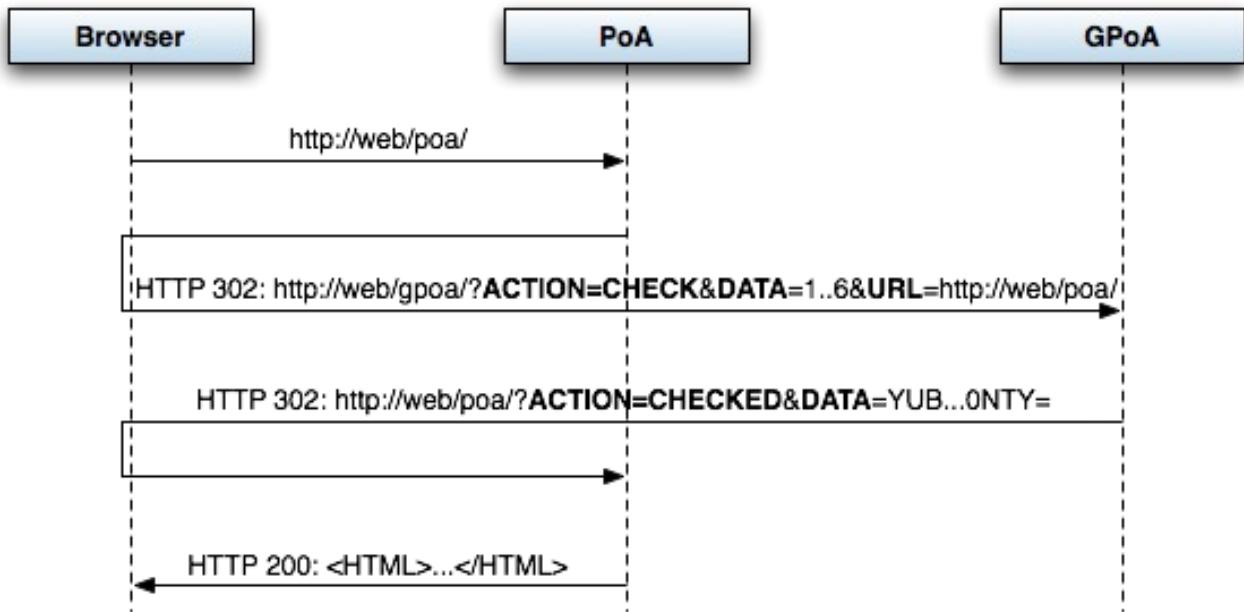
- Separador d'atributs: ,
- Separador atribut-valor: =
- Separador de valors en un atribut: |

D'aquesta forma quedaria una asserció de la següent manera:

```
name1=valorA,name2=valorB|valorC@PAPI_AS_ID
```

**Petició de decisió d'autorització: CHECK – CHECKED**

Un PoA envia un missatge CHECK a un GPoA per comprovar si l'usuari té potencialment drets d'accés als recursos protegits per aquest PoA. En cas de que sigui així, obté informació d'autenticació i una sèrie d'atributs, pel que a més pot realitzar la seva pròpia política d'autorització en base a ella.



**Missatge CHECK**

El format del missatge és el següent:

```
http://%gpoa-url%/AuthLocation?ACTION=CHECK&DATA=%value%&URL=%PoA-url%
```

a on:

- **gpoa-url** és la direcció web del GPoA al qui se li envia el missatge.
- **ACTION** és el tipus de missatge que s'envia al GPoA.
- **DATA** és una cadena alfanumèrica que s'ha de retornar en la resposta.
- **URL** és la direcció de resposta a la que s'ha de fer la redirecció 302 HTTP en el GPoA.

**Resposta CHECKED a un CHECK**

El GPoA envia un missatge CHECKED com a resposta al missatge anterior:

- En el cas que l'usuari tingui una cookie Hcook en el GPoA, se li envia una resposta correcta amb el format següent:

```
%PoA-url%?ACTION=CHECKED&DATA=%value%
```

a on:

- **PoA-url** és la direcció web que el PoA ha enviat en el camp URL del missatge CHECK.
- **ACTION** és el tipus de resposta que envia, en aquest cas CHECKED.
- **DATA** és el valor que conté la informació que requeria el PoA. El valor de DATA s'obté de passar a base 64 el valor obtingut de xifrar la següent cadena de text amb la clau privada RSA del GPoA:

```
userAssertion:expiryTime:currentTime:KEY
```

a on:

- **userAssertion** és l'assertió de l'usuari pel PoA que ha enviat la petició.
- **expiryTime** és el temps en segons des de l'1 de gener de 1970 en el que l'assertió deixa de ser vàlida.
- **currentTime** és el temps en segons des de l'1 de gener de 1970 en que ha estat generada l'assertió.
- **KEY** és el valor del paràmetre PAPIPOAREF que es va rebre en el missatge ATTREQ.

En el cas que l'AS vulgui indicar que l'usuari no té autorització per a accedir al PoA o per indicar que s'ha produït un error en l'autenticació el valor de userAssertion ha de ser 'ERROR'.

### Selecció del Proveïdor d'Identitat

En el cas que un usuari no tinguis un context de seguretat en el GPoA, aquest ha de mostrar una pàgina web en la qual l'usuari pot elegir en quin Proveïdor d'Identitat (AS) es vol autenticar. Aquesta pàgina s'anomena *Wher Are You From? (WAYF)* i s'encarrega de redirigir a l'usuari al seu Proveïdor d'Identitat mitjançant un missatge ATTREQ



Per a indicar-li al GPoA quin és el Proveïdor d'Identitat on s'ha d'autenticar l'usuari s'utilitza el següent paràmetre opcional dins del missatge CHECK:

- **PAPIHLI** és l'identificador de l'AS o proveïdor d'identitat PAPI a on s'ha de redirigir a l'usuari en cas que no existeixi un context de seguretat en el GPoA.

Així, quan el GPoA no rep aquest paràmetre PAPIHLI en una petició ha de mostrar la pàgina WAYF? La qual acabarà repetint la mateixa petició que ha rebut afegint-li aquest paràmetre.

### 10.3 Arquitectura del mòdul d'autenticació PAPI per Drupal

El mòdul a desenvolupar està programat en llenguatge PHP i està basat el mòdul contribuït de Drupal simpleSAMLphp\_auth [17], que realitza l'autenticació de l'usuari amb el sistema d'autenticació SAML.

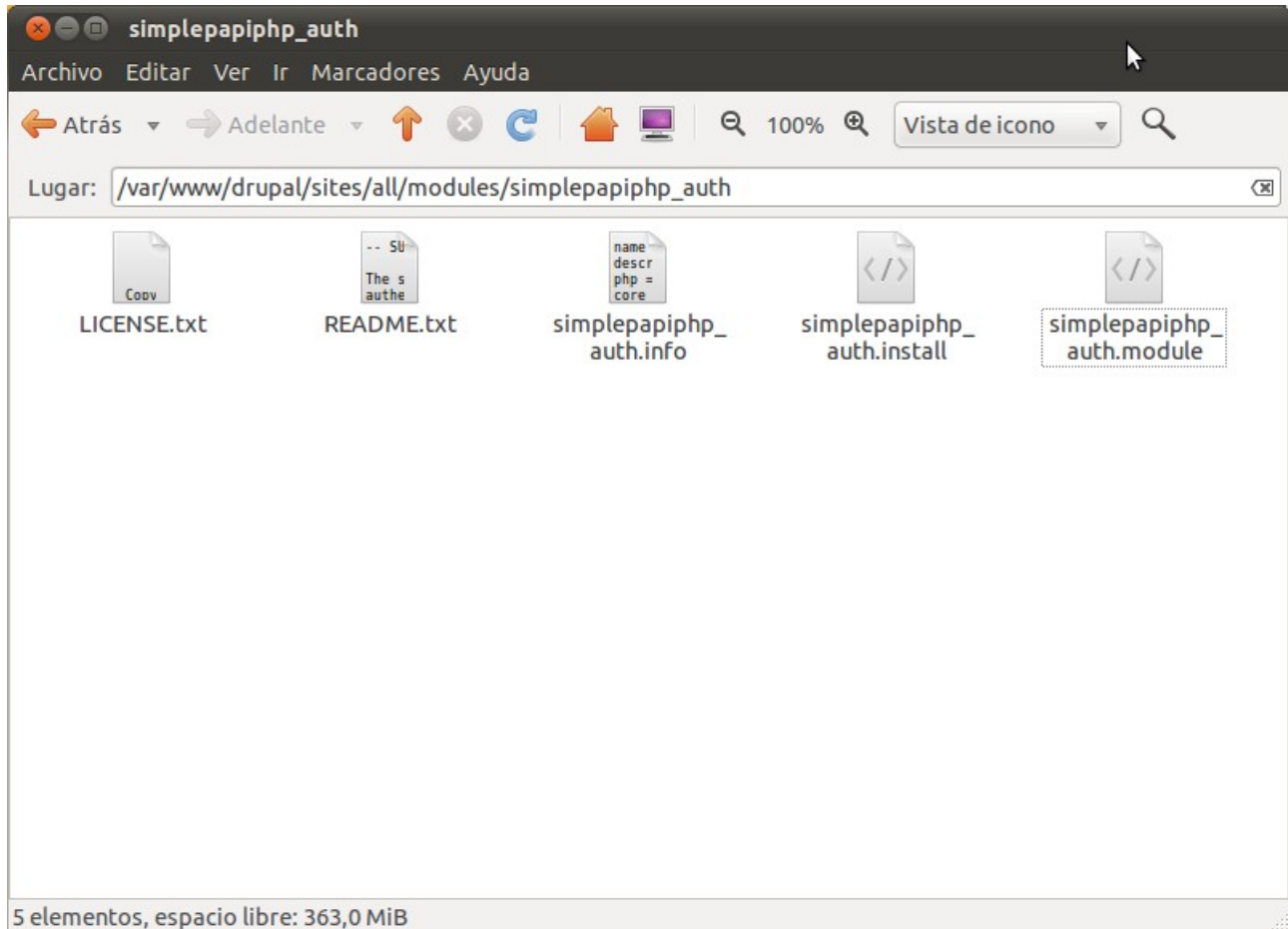
El mòdul consta d'una sèrie de funcions que realitzen el procés de comunicació amb el punt d'accés PAPI, capturar l'usuari i iniciar sessió. També hi ha una funció que genera el formulari de configuració del mòdul i una per modificar el mòdul d'autenticació de Drupal per què ofereixi l'opció d'autenticar-se amb PAPI.

Entre les funcions disponibles al mòdul podem destacar les següents:

- **simplepapi\_auth\_init()**. Funció d'inici del mòdul, defineix les variables principals i crea un nou objecte EasyPoA
- **simplepapi\_auth\_settings()**. Funció que genera el formulari de configuració del mòdul. Crea i actualitza les variables de sistema que emmagatzemen els valors de configuració.
- **simplepapi\_auth\_form\_alter(&\$form, \$form\_state, \$form\_id)**. Modifica el mòdul d'autenticació de Drupal perquè aparegui l'enllaç a l'autenticació PAPI.
- **simplepapi\_auth\_login()**. És la funció que connecta amb EasyPoA per autenticar l'usuari posteriorment realitzar les accions adequades segons la resposta de l'AS.
- **\_simplepapiphp\_auth\_get\_authname()**. Captura el nom d'usuari que retorna l'AS de

## PAPI per poder iniciar sessió en Drupal

Els mòduls contribuïts de Drupal consten d'una sèrie d'arxius que contenen el codi, la configuració, la llicència, instruccions del mòdul i un instal·lador/des-instal·lador. Aquests arxius es desen a una carpeta amb el nom del mòdul ubicada a `/sites/all/modules/simplepapiphp_auth`, com es pot veure a la imatge següent:



Els arxius que formen el mòdul són:

- **LICENSE.txt**. Texte legal amb la llicència GNU/GPL del mòdul.
- **README.txt**. Arxiu que conté indicacions d'instal·lació, configuració i funcionament del mòdul.
- **simplepapiphp\_auth.info**. Arxiu de configuració de Drupal.
- **simplepapiphp\_auth.install**. Arxiu amb codi PHP per instal·lar i des-instal·lar el mòdul.
- **simplepapiphp\_auth.module**. Codi font del mòdul en llenguatge PHP.

# 11. Desenvolupament

## 11.1 Planificació de les activitats de desenvolupament i integració del sistema.

Seguint la planificació general del projecte la fase de desenvolupament s'inicia a mitjans de novembre i finalitza a finals de desembre. En aquest període s'han decidit les alternatives del sistema operatiu i programes a utilitzar i s'ha implantat l'entorn de desenvolupament del mòdul.

Els elements necessaris pel desenvolupament del projecte són:

- Servidor web amb sistema operatiu GNU/Linux, en aquest cas he optat per la versió 10.10 d'Ubuntu
- Servidor web Apache2 versió 2.2.20
- PHP, versió 5.3.3-1ubuntu9.6
- Servidor de base de dades MySQL versió 5.1.58
- CMS Drupal versió 7.9
- Servidor d'autenticació icgPoA de PAPI, versió 1.5
- Punt d'accés EasyPoA de PAPI, versió 1.5
- mòdul mcript de PHP
- mòdul OpenSSL de PHP
- PEAR i el paquet PEAR::Log.
- Editor de text. He utilitzat la versió 2.30.3 de Gedit amb l'opció de realçar la sintaxi PHP activada.
- Navegador web, he utilitzat la versió 14.0.835.202 de Google Chrome, així com la versió 3.6.23 de Mozilla Firefox.

## 11.2 Desenvolupament

La programació del mòdul es basa en l'adaptació del mòdul contribuït simpleSAMLphp\_auth que es troba al repositori de Drupal i que realitza una funció similar al que estic desenvolupant.

El mòdul simpleSAMLphp\_auth autentica l'usuari de forma distribuïda mitjançant el sistema d'autenticació SAML i consta d'una sèrie de funcions que connecten amb el Service Provider (SP) de SAML simplesamlphp. Per tant, s'ha adaptat aquest mòdul perquè en comptes de connectar-se amb l'SP de SAML es connecti al PoA de PAPI EasyPoA.

### 11.2.1 Preparació de l'entorn

El procés d'instal·lació i configuració de tots els elements que intervenen en el procés és el següent:

#### 1. Instal·lar servidor web Apache2

```
# sudo apt-get install apache2
```

#### 2. Assignar permisos de lectura-escritura al directori /var/www per l'usuari www-data

```
# sudo chmod ug+rx /var/www
```

### 3. Instal·lar servidor mysql

```
# sudo apt-get install mysql-server
```

### 4. Instal·lar php5

```
# sudo apt-get install php5
```

### 5. Instal·lar mòdul Apache mysql

### 6. Instal·lar mòdul PHP5 gd

### 7. Instal·lar mòdul PHP5 mysql

### 8. Instal·lar mòdul PHP5 mycrypt

### 9. Instal·lar mòdul PHP5 OpenSSL

### 10. Instal·lar pear

```
# sudo wget http://pear.php.net/go-pear  
# sudo php go-pear.phar  
# sudo apt-get install php-pear  
# sudo pear install Log
```

### 11. Crear la base de dades MySQL per utilitzar amb Drupal

```
# mysql -user=root -password=*****  
mysql>create database bd_drupal;  
mysql> exit
```

### 12. Crear infraestructura de servidor web de prova en la màquina local. Per això es definirà un nom de domini de proves que es correspongui amb localhost al arxiu */etc/hosts*.

```
127.0.0.1 service.example.com
```

### 13. Afegir un servidor virtual a apache que apunti directament al directori arrel de Drupal, modificant l'arxiu de configuració */etc/apache2/sites-available/default*.

```
<VirtualHost *:80>  
  
    ServerName service.example.com  
    DocumentRoot /var/www/drupal  
  
</VirtualHost>
```

### 14. Instal·lar Drupal 7.9

Descarregar *Drupal-7.9.tar.gz* de <http://drupal.org/project/drupal>

Descomprimir a `/var/www/drupal`

Amb el navegador web anar a <http://service.example.com>. S'iniciarà la instal·lació de Drupal, seguir les indicacions d'instal·lació i configuració.

### 15. Instal·lar icgpoa

Descomprimir l'arxiu `icgpoa.zip` adjunt a aquesta documentació en el directori arrel de Drupal. Aquest arxiu l'ha proporcionat el tutor de pràctiques per disposar d'un sistema senzill d'autenticació en un entorn de proves, a on s'autentica l'usuari quan el nom d'usuari i la contrasenya són iguals. En un entorn de producció s'haurà de modificar aquest `icgpoa` perquè autèntiqui amb usuari i contrasenya real o bé designar un AS d'una organització d'autenticació.

### 16. Instal·lar i configurar EasyPoA

Descarregar `EasyPoA.tar.gz` del lloc web de prise [18] <http://www.prise.es/codes/index.php/p/papi-easypoa/downloads/>

Instal·lar el directori EasyPoA a `/var`. Aquest directori ha de ser accessible des del servidor però no pels usuaris del lloc web, a fi que mai es puguin servir les classes PHP que formen EasyPoA

EasyPoA requereix d'un servidor web i un intèrpret PHP. Necessita tenir instal·lats i activats els mòduls següents:

- Mòdul Crypt de PHP
- Mòdul OpenSSL de PHP
- El paquet PERL::Log

Aquests paquets ja haurien d'estar instal·lats, tal com s'indica als punts 8, 9 i 10.

EasyPoA consta dels següents arxius i directoris:

- **EasyPoA/**: Directori que conté les classes PHP que formen EasyPoA.
- **EasyPoA/keys**: Directori que conté l'arxiu `_GPoA_pubkey.pem`. Aquest arxiu té la clau privada que proporciona `icgPoA`.
- **EasyPoA/logs**: Directori que conté l'arxiu `PAPIEasyPoA.log`.
- **EasyPoA/PAPIPoA.ini**: És l'arxiu de configuració de EasyPoA.
- **LICENSE-2.0.txt**: Fitxer amb la llicència Apache v2.0.
- **Release-Notes.txt**: Fitxer amb l'històric de les notes de les versions de EasyPoA

La configuració d'EasyPoA es troba a l'arxiu `PAPIPoA.ini` i consta de las següents seccions:

- **Secció Global [Global]**: Defineix tots els paràmetres comuns per tots els recursos que es protegeixen amb la mateixa instància d'EasyPoA. Els paràmetres definits a aquesta secció s'heretaran al reste de seccions que es corresponguin amb la configuració de cada recurs, a menys que siguin re-definits.
  - **Lcook\_Timeout**: Defineix el temps de vida en segons de la sessió.
  - **URL\_Timeout**: Defineix el temps màxim que pot passar entre que l'EasyPoA demana aun GPoA i aquest contesta.
  - **GpoA\_URL**: URL del GPoA al qua demanarà EasyPoA si no té dades sobre



l'usuari que tracta d'accedir al recurs.

- **Pubkeys\_Path:** Ruta del directori a on es troba la clau pública del GPoA. El fitxer ha de tenir el nom *\_GpoA\_pubkey.pem*.
- **Attribute\_Separator:** Caràcter utilitzat per separar els atributs en l'assertió enviada des del GPoA a EasyPoA.
- **LogLevel:** Nivell del log d'EasyPoA. Els diferents valors que pot agafar són:
  - **PEAR\_LOG\_EMERG:** Systema no usable.
  - **PEAR\_LOG\_ALERT:** Requereix axxió immediata.
  - **PEAR\_LOG\_CRIT:** Condició crítica.
  - **PEAR\_LOG\_ERR:** Condició d'error.
  - **PEAR\_LOG\_WARNING:** Condició d'avís.
  - **PEAR\_LOG\_NOTICE:** Normal però significat.
  - **PEAR\_LOG\_INFO:** Informatiu.
  - **PEAR\_LOG\_DEBUG:** Missatges de nivell de depuració.
- **LogFile:** Ruta i nom de l'arxiu on s'enviaran els missatges de log d'EasyPoA.
- **Secció de recursos [ServiceID].** Aquestes seccions defineixen el recurs a protegir per EasyPoA. *ServiceId* és l'identificador del recurs i el que proporciona el nom a la secció. Cada una d'aquestes seccions ha de tenir, com a mínim, el paràmetre *location* que indica el path del recurs en el servidor web que es vol protegir. La resta de paràmetres seran els mateixos de la secció [Global] en cas que es vulguin modificar.
  - **Location:** Path del recurs en el servidor web que es desitja protegir

Per instal·lar correctament EasyPoA es concediran permisos de lectura i execució pel directori EasyPoA i el es contingut a l'usuari que executa el servidor web, en el cas d'Apache en linux és www-data.

```
chmod -R ugo+r /usr/local/lib/php/EasyPoA
chmod a+x /usr/local/lib/php/EasyPoA
```

La configuració de l'arxiu de configuració PAPIPoA.ini serà:

```
[Global]
Lcook_Timeout      = 60
URL_Timeout        = 200
GPoA_URL            = http://service.example.com/icgpoa/icgpoa.php
Pubkeys_Path       = /var/EasyPoA/EasyPoA/keys
Attribute_Separator = ","
Value_Separator    = "="
LogLevel           = PEAR_LOG_DEBUG
LogFile            = /var/EasyPoA/EasyPoA/logs/PAPIEasyPoA.log

[Drupal]
Location = /
```

Finalment, copiarem la clau pública de l'icgPoA a l'arxiu `/var/EasyPoA/EasyPoA/keys/_GpoApubkey.pem`. Aquesta clau pública la trobarem al valor de la variable `$pubKey` de l'arxiu `/icgpoa/icgpoa.php`.

### 11.2.2 Generació del codi font del component

Com ja he comentat anteriorment el codi font del component està basat en el mòdul contribuït `simpleSAMLphp_auth`, disponible al repositori de Drupal. S'han aprofitat l'estructura i les funcions del mòdul per adaptar-les als requeriments i funcions de PAPI.

L'arxiu amb el codi font del mòdul és `simplepapipahp_auth.module` i està codificat en PHP. A continuació enumeraré les funcions que s'han modificat i quines han estat aquestes modificacions.

- **`simplepapipahp_auth_menu()`**. Executa la el formulari de configuració del mòdul, d'una banda, i el d'autenticació d'usuari, de l'altra, segons els paràmetres del formulari que es configuren a la funció `simplepapipahp_auth_form_alter()`.

```
function simplepapipahp_auth_menu() {
  watchdog('papidebug', "_simplepapipahp_auth_menu", NULL, WATCHDOG_DEBUG);
  $items = array();
  $items['admin/config/people/simplepapipahp_auth'] = array(
    'title' => 'Ajustes del mòdul de autenticació simplePAPIphp',
    'description' => 'Control the various settings of the simplePAPIphp authentication module',
    'page callback' => 'drupal_get_form',
    'page arguments' => array('simplepapipahp_auth_settings'),
    'access arguments' => array('administer simplePAPIphp authentication'),
    'type' => MENU_LOCAL_TASK | MENU_NORMAL_ITEM,
  );
  $items['papi_login'] = array(
    'title' => 'Logon to the site',
    'description' => 'Provides a site login page',
    'page callback' => 'simplepapipahp_auth_loginpage',
    'access callback' => TRUE,
    'type' => MENU_CALLBACK,
  );
  return $items;
}
```

- **`simplepapipahp_auth_loginpage()`**. Aquesta funció crea una nova instància de EasyPoA i autentica l'usuari. És la que connecta amb EasyPoA i rep l'assertió amb l'usuari autenticat.

`$_simplepapipahp_auth_as->checkAccess()` comprova que l'usuari té els permisos adequats, en tal cas comprova si l'usuari existeix a Drupal. Si existeix, inicia sessió, en cas contrari, crea l'usuari extern a Drupal i inicia sessió. El codi està dins d'un bloc `try/catch` per facilitar la captura d'excepcions.

```
function simplepapipahp_auth_loginpage() {
  watchdog('papidebug', "simplepapipahp_auth_loginpage", NULL, WATCHDOG_DEBUG);
  global $user;
  global $_simplepapipahp_auth_as;
  global $_simplepapipahp_auth_papi_attributes;
  // if (!simplepapipahp_auth_isEnabled()) {
  //   // Exit without initializing.
  //   drupal_set_message(t('We\'re sorry this feature is not yet enabled.'));
  //   return;
  // }
  $output=null;
  if ($user->uid == 0 ) {
    watchdog('papidebug', "simplepapipahp_auth_autentica", NULL,
```

```

WATCHDOG_DEBUG);
    try{
        $papiauthname='';
        $simplepapiphp_auth_as = new
EasyPoA(variable_get('simplepapiphp_auth_authsource'));
        $simplepapiphp_auth_papi_attributes =
        $simplepapiphp_auth_as->getUserData();
        echo "<META HTTP-EQUIV='refresh' CONTENT='0'>";
        if($simplepapiphp_auth_as->checkAccess()){
            $papiauthname =
            $simplepapiphp_auth_get_authname();
            drupal_set_message('Usuario ' . $papiauthname. '
autenticado correctamente en PAPI', 'status');
            $user = user_external_load($papiauthname);
            if ($user) {
                $msg = t("Bienvenido de nuevo, " . $user-
>name);
            }
            drupal_set_message($msg, 'status');
        }else{
            $msg = t("Usuario " . $papiauthname. " no
existe en Drupal");
            drupal_set_message($msg, 'status');

            user_external_login_register($papiauthname,'simplepapiphp_auth');
            $msg = t("Creado usuario " . $papiauthname);
            drupal_set_message($msg, 'status');
        }
    }

} catch(Exception $e){
    $msg = "<h1>Error en la ejecución</h1>";
    $msg .= "<p>". $e->getMessage(). "</p>\n";
    $msg .= "<pre>". print_r($e-
>getTraceAsString(), true). "</pre>\n";
    //
    drupal_set_message($msg, 'error');
} catch(CryptException $cryptException){
    $msg = "<h1>Error criptográfico en la ejecución de
papi_EasyPoA</h1>";
    $msg .= "<p>[" . $cryptException->getMessage(). "]</p>\n";
    $msg .= "<pre>". print_r($cryptException-
>getTraceAsString(), true). "</pre>\n";
    drupal_set_message($msg, 'error');
} catch(EasyPoAException $poaException){
    $msg = "<h1>Error en la ejecución de papi_EasyPoA</h1>";
    $msg .= "<p>". $poaException->getMessage(). "</p>\n";
    $msg .= "<pre>". print_r($poaException-
>getTraceAsString(), true). "</pre>\n";
    drupal_set_message($msg, 'error');
} catch(ConfigurationException $confException){
    $msg = "<h1>Error en la configuración de EasyPoA</h1>";
    $msg .= "<p>". $confException->getMessage(). "</p>\n";
    $msg .= "<pre>". print_r($confException-
>getTraceAsString(), true). "</pre>\n";
    drupal_set_message($msg, 'error');
}
}
else {
    drupal_goto('user/' . $user->uid);
}
return $output;
}

```

- **simplepapiphp\_auth\_init()** Implemeta el hook d'inici. Es defineixen les variables locals, es

referencia la classe i crea una nova instància d'EasyPoA.

```
function simplepapiphp_auth_init() {

    watchdog('papidebug', "simplepapiphp_auth_init", NULL, WATCHDOG_DEBUG);
    global $user;
    global $_simplepapiphp_auth_as;
    global $_simplepapiphp_auth_papi_attributes;
    global $_simplepapiphp_auth_papi_config;
    global $_simplepapiphp_auth_papi_version;

    require_once('/var/EasyPoA/EasyPoA/EasyPoA.class.php');

    watchdog('papidebug', "simplepapiphp_auth_autentica", NULL,
WATCHDOG_DEBUG);
    $_simplepapiphp_auth_as = new
EasyPoA(variable_get('simplepapiphp_auth_authsource'));
}
```

- **simplepapiphp\_auth\_settings()** Genera el formulari de configuració del mòdul. Utilitza l'api de Drupal per crear variables de sistema que contenen els valors assignats per l'usuari. Els paràmetres que configura són:
  - Activació de l'autenticació PAPI.
  - Directori d'instal·lació del GPOA per defecte.
  - Identificació del recurs a protegir.
  - Atribut de PAPI que s'utilitzarà com a nom d'usuari.
  - Atribut de PAPI que s'utilitzarà com a direcció de correu electrònic de l'usuari.
  - Forçar https per enllaços de login.

```
function simplepapiphp_auth_settings() {
    watchdog('papidebug', "simplepapiphp_auth_settings", NULL, WATCHDOG_DEBUG);
    global $_simplepapiphp_auth_papi_version;

    if (!empty($_simplepapiphp_auth_papi_version) ) {
        $ver = explode('.', $_simplepapiphp_auth_papi_version);
        if ( !($ver[0] >= 1 && $ver[1] >= 5) ) {
            drupal_set_message(t("Please upgrade SimplePAPIphp. You are using
%ssp_version", array('%ssp_version' => $_simplepapiphp_auth_papi_version)),
'warning');
        }
    }
    $form = array(
        'simplepapiphp_auth_activate' => array(
            '#type' => 'checkbox',
            '#title' => t('Activar autenticación PAPI'),
            '#default_value' => variable_get('simplepapiphp_auth_activate', FALSE),
            '#description' => t('Activar esta función antes de configurar el módulo
puede provocar el bloqueo de Drupal.'),
        ),
        'simplepapiphp_auth_installdir' => array(
            '#type' => 'textfield',
            '#title' => t('Directory de instalación del GPOA por defecto:
/var/EasyPoA'),
            '#default_value' => variable_get('simplepapiphp_auth_installdir',
'/var/EasiPoA'),
            '#description' => t('Directorio base del GPOA EasyPoA. Path absoluto sin la
barra final.'),
        ),
    ),
```

```

'simplepapiphp_auth_authsource' => array(
  '#type' => 'textfield',
  '#title' => t('Identificación del recurso a proteger, por defecto: Drupal'),
  '#default_value' => variable_get('simplepapiphp_auth_authsource', 'Drupal'),
  '#description' => t("Identificación del recurso a proteger. Nombre de la
sección que contiene el parámetro 'Location' en el archivo de configuración
PAPIPoA.ini"),
),
'simplepapiphp_auth_user_name' => array(
  '#type' => 'textfield',
  '#title' => t("Atributo de simplePAPIphp que se usará como nombre de
usuario, por defecto 'uid' "),
  '#default_value' => variable_get('simplepapiphp_auth_user_name',
'eduPersonPrincipalName'),
  '#description' => t('Ejemplo: <i>uid</i>.'),
  '#required' => TRUE,
),
'simplepapiphp_auth_mailattr' => array(
  '#type' => 'textfield',
  '#title' => t("Atributo de simplePAPIphp que se usará como dirección de
correo, por defecto 'mail'"),
  '#default_value' => variable_get('simplepapiphp_auth_mailattr', 'mail'),
  '#description' => t('Ejemplo: <i>mail</i><br />.'),
),
'simplepapiphp_auth_forcehttps' => array(
  '#type' => 'checkbox',
  '#title' => t('Forzar https para enlaces de login'),
  '#default_value' => variable_get('simplepapiphp_auth_forcehttps', TRUE),
  '#description' => t('Debería activarse para sitios en producción.'),
),
);
return system_settings_form($form);
}

```

- **simplepapiphp\_auth\_form\_alter(&\$form, \$form\_state, \$form\_id).** Modifica els formularis de login d'usuari i de configuració del compte d'usuari amb els valor de PAPI Login.

```

function simplepapiphp_auth_form_alter(&$form, $form_state, $form_id) {
  watchdog('papidebug', "simplepapiphp_auth_form_alter", NULL, WATCHDOG_DEBUG);
  if ($form_id == 'user_login_block') {
    $link = l('PAPI Log In', 'papi_login');
    $links = $form['links']['#markup'];
    $links = str_replace('</ul>', '<li class="papi">' . $link . '</li></ul>',
$links);
    $form['links']['#markup'] = $links;
  }

  if ($form_id == 'user_account_form') {
    $link = l('PAPI Log In', 'papi_login');
    $links = $form['links']['#markup'];
    $links = str_replace('</ul>', '<li class="papi">' . $link . '</li></ul>',
$links);
    $form['links']['#markup'] = $links;
  }
}

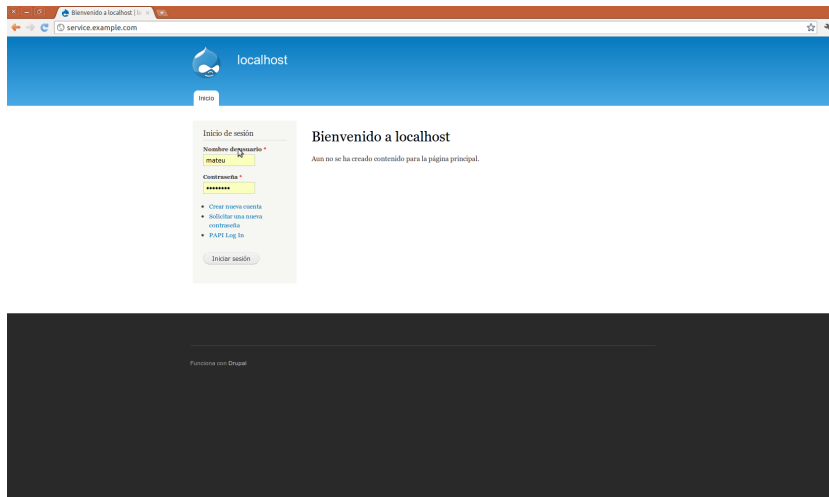
```

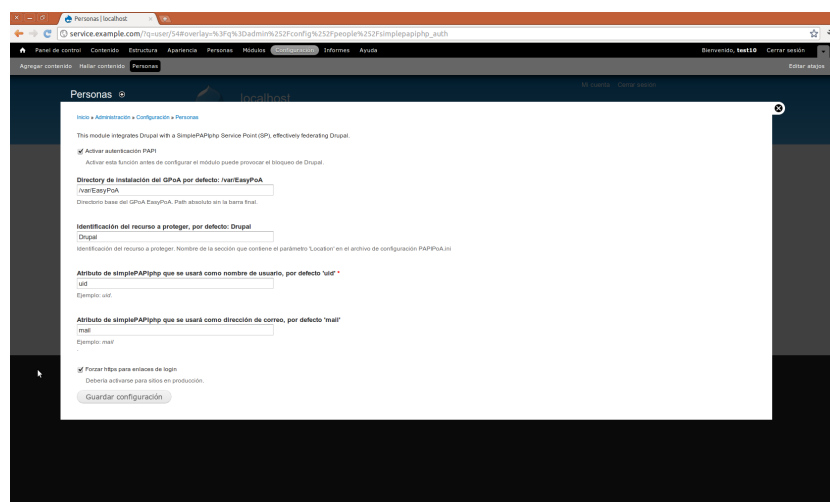
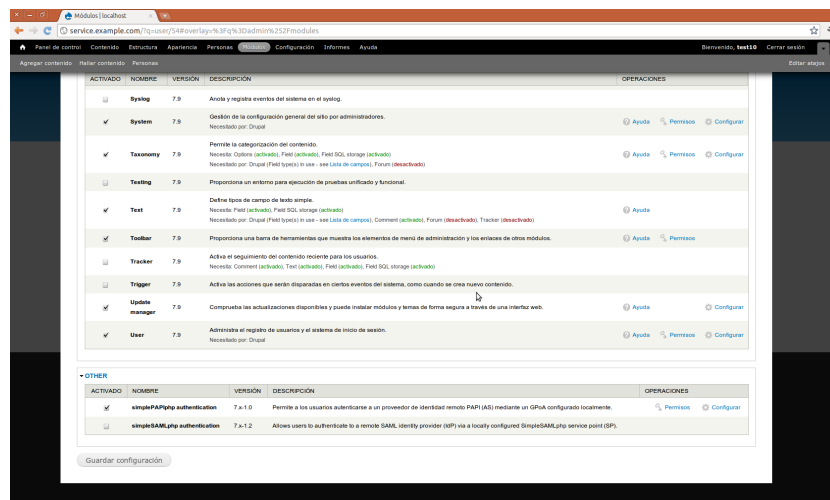
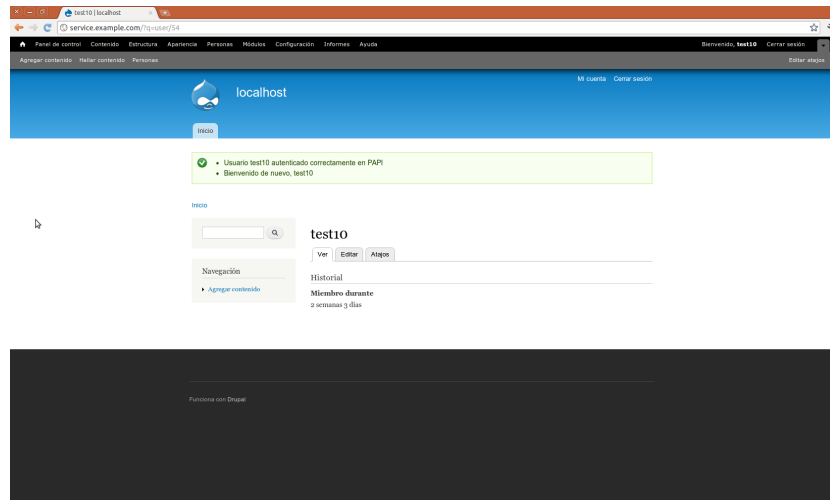
La resta de funcions bàsicament no s'han modificat més que en els noms de les variables, ja que no influeixen en l'accés a les llibreries de EasyPoA ni afecten al funcionament del mòdul una vegada modificat.

### 11.2.3 Execució de les proves unitàries

Les proves unitàries s'han realitzat d'acord als casos d'ús especificats al capítol 3.2.

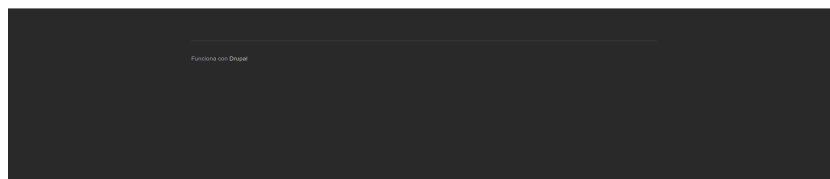
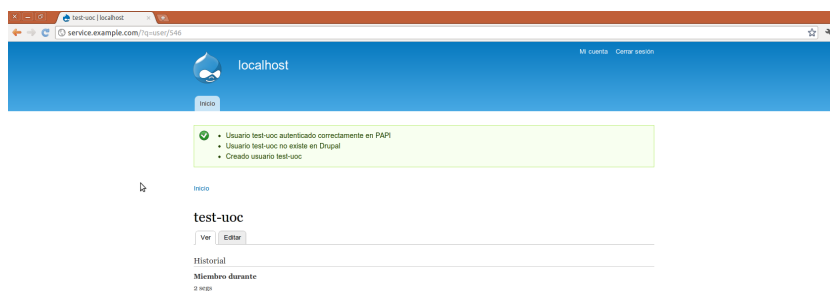
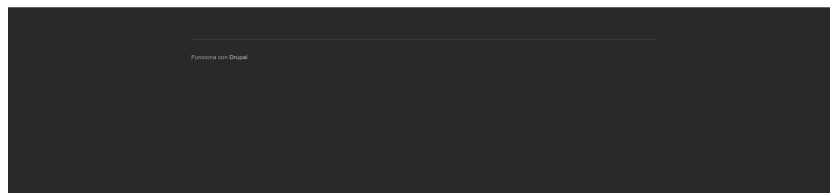
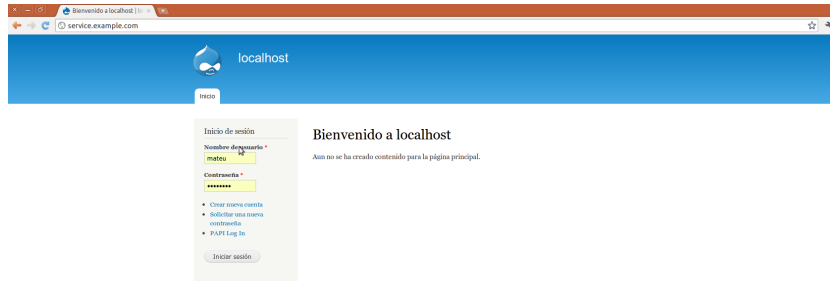
- **Escenari 1**, que correspon al cas d'ús 'autenticació quan l'usuari existeix en Drupal'. Drupal té un un usuari creat, *test10*, amb privilegis d'administrador. Es realitzarà una autenticació PAPI amb aquest usuari i es comprovarà que funcioni correctament tant l'inici de sessió com el formulari de configuració del mòdul. Com hem comentat abans, perquè l'AS autentiqui correctament l'usuari, el nom d'usuari i la contrasenya han de ser iguals, per tant, usuari és *test10* i contrasenya també és *test10*.





Com es pot veure a les imatges, tant l'autenticació com la configuració del mòdul funcionen correctament.

- **Escenari 2**, que correspon al cas d'ús 'autenticació si l'usuari no existeix a Drupal'. En aquest cas s'autenticarà amb l'usuari *test-uoc*, el qual no existeix a Drupal. Es comprovarà que crea l'usuari i i inicia sessió a Drupal de forma correcta. La contrasenya perquè autèntiqui correctament l'usuari és *test-uoc*.



Com es pot veure a les imatges, ha creat correctament l'usuari test-uoc i ha iniciat sessió amb



aquest usuari. En aquest cas, al no tenir permisos d'administrador no mostra la barra de menús de configuració.

### 11.3 Documentació

Simultàniament al desenvolupament del mòdul s'ha anat generant la documentació del projecte. Aquesta documentació consta dels següents elements:

- **Codi del mòdul.** Codi en PHP del mòdul.
- **Arxiu README.txt**, ubicat juntament amb la llicència, el codi, l'arxiu de configuració i l'arxiu d'instal·lació/des-instal·lació. Conté indicacions sobre la instal·lació i configuració del mòdul.
- Arxiu INSTALL.txt, amb indicacions de com instal·lar i configurar el mòdul
- **Memòria del projecte**, que és aquest mateix document.
- Presentació del projecte, en format lliure que consta d'una sèrie de diapositives que expliquen el projecte.
- Vídeo de presentació, a on es mostra la presentació del projecte i una demostració del seu funcionament, acompanyat de les explicacions de l'autor.
- Apendix.pdf. Document que conté els texts legals de les llicències del mòdul (GNU/GPL) i de la memòria (CC BY-SA)

## 12. Implantació

Per implantar el mòdul en un entorn de producció, a l'igual que en l'entorn de desenvolupament, s'haurà d'habilitar un servidor a on s'hauran d'instal·lar i configurar tots els elements necessaris pel seu correcte funcionament. Al ser un component de Drupal possiblement la majoria de requeriments que es necessiten ja estaran instal·lats i configurats, només seria precís instal·lar i configurar el PoA amb EasyPoA i seleccionar un Servidor d'Autenticació que realitzés el procés d'autenticació dels usuaris. Aquest AS podria estar al mateix servidor o bé estaria ubicat en qualsevol lloc accessible per web.

### 12.1 Implantació del sistema

Els elements necessaris per la implantació del sistema són:

- **Servidor**, que disposi d'una distribució actualitzada del sistema operatiu GNU/Linux.
- **Servidor web** Apache2 versió 2.2.20 o superior
- **PHP**, versió 5.3.3-1ubuntu9.6 o superior amb els mòduls mcrypt i OpenSSL instal·lats i activats.
- **Servidor de base de dades MySQL** versió 5.1.58 o superior
- **Servidor d'autenticació icgPoA** de PAPI, versió 1.5, en el cas que es vulgui fer l'autenticació en el propi servidor, en cas contrari tenir accés a un servidor d'autenticació PAPI extern.
- **Punt d'accés EasyPoA** de PAPI, versió 1.5, instal·lat i configurat tal com s'ha descrit al capítol 11.2.1 apartat 11.
- **PEAR** i el paquet PEAR::Log.
- **Drupal versió 7.x**, possiblement també funcioni en altres versions, però s'hauria de comprovar.
- **Mòdul contribuït simplePAPIphp\_auth**, que és el mòdul desenvolupat en aquest projecte.

### 12.2 Proves

Abans de la implantació definitiva del sistema s'haurà de comprovar que tots els elements que intervenen en el procés funcionin de forma correcta i que els resultats que s'obtenen són els esperats.

En cas que es produeixin incidències es prendrà nota i es tractaran de resoldre.

## 13. Manteniment

Tot projecte una vegada finalitzat, provat i implantat amb èxit necessita un manteniment, ja sigui per depurar errors, afegir noves funcionalitats o adaptar-lo a diferents requeriments. L'avantatge d'utilitzar codi obert i una llicència lliure és que quan s'allibera el codi aquest queda a disposició de qualsevol persona o organització que el vulgui utilitzar i es té total llibertat per utilitzar, estudiar, distribuir, modificar i redistribuir el codi modificat sense cap limitació, en contra dels programes privats, a on l'únic autoritzat a modificar i distribuir el programa és la persona o organització propietària del copyright.

Aquest projecte, seguint la filosofia del programari lliure i com ja s'ha comentat anteriorment, es publicarà en el repositori de drupal.org juntament amb el seu codi font i la documentació necessària per instal·lar-lo i configurar-lo, deixant la porta oberta a que qualsevol desenvolupador de la comunitat pugui aprofitar el codi per modificar-lo, ampliar-lo o utilitzar-lo per la finalitat que consideri adequada i d'aquesta forma contribuir en el manteniment del mòdul.

D'altra banda, Rediris és una xarxa de recerca espanyola que proporciona serveis avançats de comunicació i seguretat a la comunitat científica i universitària espanyola i entre les seves activitats es troba el projecte PAPI. Aquest sistema d'autenticació i autorització ja s'ha implementat com a plugin per a altres aplicacions, com per exemple, DokuWiki, Media Wiki i WordPress, i ara estarà disponible també per Drupal.

## 14. Conclusions

Com ja he comentat al capítol 2.2 Drupal té al seu repositori de mòduls contribuïts una sèrie de mòduls per l'autenticació i autorització Single Sign-On. El desenvolupament d'aquest projecte permetrà disposar d'una opció més per garantir l'accés autoritzat a recursos distribuïts en xarxa, utilitzant el sistema d'autenticació i autorització PAPI.

El sistema que utilitza PAPI és mitjançant cookies temporals que a més impedeixen que es puguin copiar a altres equips. Quan el GPoA detecta que una cookie ha caducat o no procedeix de l'ordinador origen denega el servei i provoca que l'usuari s'hagi de tornar a autenticar.

El codi del mòdul està basat en un mòdul de Drupal que integra l'autenticació distribuïda SAML en PHP. Aquest mòdul s'ha adaptat als requisits i característiques de PAPI per realitzar l'autenticació Single Sign-On en un AS. Disposa d'un formulari de configuració i d'un enllaç que activa l'autenticació PAPI. El procés d'autenticació és molt senzill, una vegada introduïts l'usuari i contrasenya, si PAPI els valida retorna el control a Drupal i aquest comprova si aquest usuari existeix, en tal cas inicia la sessió amb els permisos assignats a l'usuari, en cas contrari el dona d'alta i inicia la sessió.

Amb aquest mòdul es pretén que l'autenticació web per utilitzar recursos distribuïts sigui més senzilla per part dels usuaris i la gestió de permisos centralitzada sigui més efectiva per les corporacions.

Amb PAPI es garanteix que només els usuaris autenticats i amb autorització accediran als recursos protegits, i amb el sistema de cookies temporals que s'utilitza s'evita que aquestes es puguin copiar a un altre equip.

La integració amb Drupal ofereix una opció més a l'hora de desenvolupar un lloc web d'altres prestacions i fiabilitat en programari lliure.

Entre les millores que s'obtindran amb l'ús d'aquest mòdul podem destacar:

- Millora de la seguretat en un entorn de recursos distribuïts
- Simplificació de l'autenticació per part de l'usuari
- Gestió de permisos i autoritzacions centralitzat per a corporacions
- Autoritzacions basades en cookies i amb data de caducitat

La forma d'implantar aquesta solució en els diferents projectes que es desenvolupin amb el CMS Drupal es realitzaran de forma senzilla gràcies a la utilització d'aquest mòdul, ja que només serà necessari configurar els paràmetres adequats per accedir als diferents Servidors d'Autenticació i atorgar els permisos a cada recurs per donar-li o denegar-li l'accés dels usuaris.

Aquest mòdul es podrà adaptar a una gran quantitat d'escenaris possibles facilitant el procés d'autenticació i autorització dels usuaris facilitant també la labor dels desenvolupadors de projectes web que utilitzin el gestor de continguts Drupal i que vulguin dotar al seu lloc web d'una seguretat àmpliament provada en altres entorns crítics de seguretat.

Durant el procés d'estudi i disseny del projecte han anat sorgint una sèrie de dificultats, bàsicament de concepte, les qual s'han anat superant de forma satisfactòria gràcies a l'ajuda i consells del tutor de pràctiques. En el procés de desenvolupament també han aparegut dificultats degut a la complexitat del projecte, la majoria de les quals s'han resolt de forma satisfactòria. Un dels problemes més importants que s'han superat ha estat que la pàgina es quedava en blanc i era necessari recarregar-la per obtenir el resultat adequat.

L'objectiu inicial d'aquest projecte era desenvolupar un mòdul per Drupal que permetés l'autenticació distribuïda a través del sistema d'autenticació i autorització PAPI. Per aconseguir-lo

s'ha desenvolupat un mòdul en PHP basat en un projecte similar ja existent que autenticava l'usuari amb el sistema d'autenticació SAML modificant el codi per accedir a les llibreries de PAPI. L'objectiu s'ha complert amb èxit, com es pot observar en el capítol de proves. El projecte és millorable, quedant pendent de realitzar altres funcions que donarien més funcionalitat al mòdul, entre les quals destacaria

- Assignació de permisos per usuari.
- Revisió dels permisos d'usuari quan s'inicia la sessió.
- Poder elegir si crear usuari o no una vegada autenticat a PAPI i l'usuari no existeix a Drupal.

El mòdul s'ha publicat al repositori de la comunitat de Drupal perquè tothom el pugui utilitzar. Al ser un projecte desenvolupat en programari lliure, els components utilitzats disposen de llicències GNU/ GPL i estar desenvolupats en PHP permetran que es puguin anar corregint errades i afegir noves funcionalitats que, a la vegada, podran revertir a la comunitat. D'aquesta manera també es garanteix la continuïtat del projecte i que el seu manteniment estigui suportat per la comunitat drupal.org.

D'altra banda, no es pot oblidar que darrera el projecte PAPI es troba RedIris, que disposa dels recursos suficients per mantenir i millorar PAPI. RedIris dona suport a les diferents adaptacions o plugins que s'han realitzat fins ara per altres plataformes, com DokuWiki, MediaWiki i WordPress.

Per finalitzar, només resta comentar que, com a experiència personal, la realització d'aquest projecte ha suposat un gran repte: he hagut d'estudiar, entendre i treballar uns conceptes nous, integrar dues plataformes diferents i documentar tot el procés en un temps limitat. Però, com tot gran esforç, la finalització amb èxit del projecte ha suposat una enorme satisfacció personal, reforça els coneixements adquirits durant el procés d'anàlisi, disseny i desenvolupament, consolidant aquests coneixements i donant-me la base per poder-los aplicar en futurs projectes. També me permetrà encarar els reptes futurs amb el convenciment que, a l'igual que aquest i independentment de la seva complexitat, els podré superar amb èxit. D'altra banda, alliberar el codi font del projecte permetrà que la gent que ho necessiti el pugui utilitzar, estudiar i modificar d'acord a les seves necessitats.

## 15. Llicències

Les llicències del projecte són:

- El codi del projecte està llicenciat sota llicència lliure GNU/GPL
- La memòria del projecte està llicenciada sota llicència Creative Commons BY-SA

El text legal d'ambdues llicències està disponible en l'arxiu adjunt [apèndix.pdf](#)

## 16. Referències

- [1] <http://www.papisoftware.net>
- [2] <http://www.drupal.org>
- [3] <http://www.cursosdrupal.com/content/introduccion>
- [4] <http://www.apache.org>
- [5] <http://www.wordpress.org>
- [6] <http://www.mediawiki.org>
- [7] <http://www.dokuwiki.org>
- [8] <http://www.jasig.org/cas>
- [9] <http://java.net/projects/opensso/>
- [10] <http://simplesamlphp.org/>
- [11] <http://es.wikipedia.org/wiki/LAMP>
- [12] <http://www.gnu.org/licenses/licenses.ca.html>
- [13] <http://www.creativecommons.org>
- [14] <http://e-spacio.uned.es:8080/fedora/get/tesisuned:IngInf-Rcastro/Documento.pdf>
- [15] <http://www.papi.rediris.es>
- [16] <http://api.drupal.org>
- [17] [http://drupal.org/project/simplesamlphp\\_auth](http://drupal.org/project/simplesamlphp_auth)
- [18] [www.prise.es](http://www.prise.es)