

RCG: Creació d'un Repositori per Impartir Competències Genèriques

José Calvo Fullana, Joe Miró, Antoni Jaume-i-Capó

PFC d'Enginyeria en Informàtica

jcfjose@gmail.com, joe.miro@uib.es, antoni.jaume@uib.es

Resum— Amb l'entrada dels nous plans d'estudis les competències genèriques han adquirit una importància que enfronta als professors a la necessitat de cercar activitats que ajudin al seu ensenyament. Per aquest motiu s'ha proposat crear, emprant l'esforç col·laboratiu, un repositori que reculli totes aquestes activitats i les posi a disposició del professorat. D'aquesta manera ha sorgit el Repositori de Competències Genèriques, un espai col·laboratiu entre professors a on aquests poden crear activitats, cercar-les, classificar-les, comentar-les i votar-les, facilitant així la tasca de recollir i organitzar aquestes competències. La direcció del portal és <http://rcg.uib.es>.

través d'Internet que faciliti l'accés a les activitats existents a la resta de professorat. Aquests processos inclouen l'estudi de viabilitat del nou sistema, l'anàlisi i definició dels requeriments, el disseny de l'arquitectura, la seva implementació i el posterior manteniment necessari per a què la aplicació segueixi en actiu. Aquest repositori de material permet al professorat publicar activitats, cercar les que hagin publicat altres professors, comentar i valorar les ja existents. Poden cercar una activitat, descarregar-la, adaptar-la a la seva metodologia a les classes i utilitzar-la lliurement en la seva docència. Poden crear noves activitats, tant amb material nou com amb noves versions de les activitats emprades anteriorment, i afegir comentaris de com les fan servir, tot creant un sistema d'ensenyament col·laboratiu.

D'aquesta manera s'ajudarà al professorat en l'ensenyament de les competències genèriques, posant a la seva disposició un material que d'altra manera consumiria molt de temps per a trobar, valorar, recollir i organitzar de manera adequada.

I. INTRODUCCIÓ

Una de les novetats que apareixen als plans d'estudis de l'Espai Europeu d'Educació Superior és la importància explícita que es dona a les competències genèriques, aquelles que són necessàries en un titulat universitari però que no estan relacionades amb els seus coneixements tècnics. Degut a aquest motiu un dels problemes al que s'enfronten els professors dintre del marc de l'EEES és que es desconeix com ensenyar aquestes competències i com integrar-les en la docència.

Actualment ja existeixen moltes activitats dissenyades i llestes per fer-ne ús en l'ensenyança. El que és difícil és trobar-les, valorar-les, determinar si són adequades al nivell de l'assignatura, trobar la documentació associada, etc. Recollir i organitzar aquesta informació facilita que tots els professors, sense necessitat de tenir grans coneixements de la competència en sí, puguin integrar en les seves assignatures les activitats que faran que els seus alumnes obtinguin les competències transversals.

Al 2008 s'inicia un projecte d'innovació i millora de la qualitat docent per donar una solució a l'ensenyament d'aquestes competències, a on es decidí resoldre aquesta dificultat mitjançant l'esforç col·laboratiu de tots els professors, ja que quasi tots tenen alguna idea o han provat alguna activitat. Aquest fet fa que ja hi hagi tot una base de documents i activitats de gran interès que facilitaria la tasca a tots els professors interessats si tinguessin un fàcil accés a ells. És per aquest motiu que es proposa la creació d'un repositori per a gestionar d'una manera més senzilla aquestes activitats.

II. OBJECTIUS

La finalitat del projecte que aquí s'exposa ha estat realitzar tots els passos del procés del cicle de vida de software necessaris per aconseguir implantar una eina col·laborativa a

III. ANÀLISI

A. Requeriments

Durant la primera fase del projecte s'ha analitzat quins són els requeriments necessaris per a la creació de l'aplicació que dona aquest suport. Per a obtenir-los s'ha emprat la tècnica de recopilació de dades basada en entrevistes, ja que aquesta és la més utilitzada degut a la seva productivitat a l'hora de determinar les necessitats del sistema [1]. D'aquest anàlisi han sortit els requeriments d'usuari, que ens han permès saber els diferents tipus d'usuaris que interactuaran amb el sistema. A la Fig. 1 es pot veure el diagrama d'usuaris del repositori, que estableix la jerarquia que aquests tenen dintre del sistema.

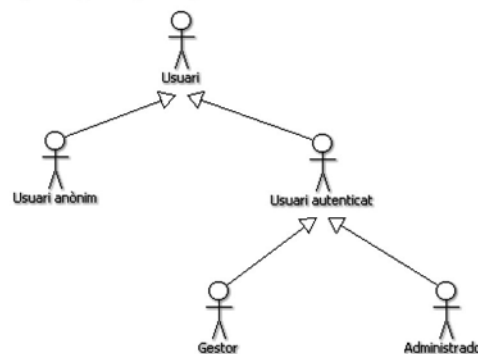


Fig. 1 Diagrama d'usuaris del sistema

Quan ja s'han definit els requeriments d'usuari aquests s'han emprat per treure els requeriments de sistema, necessaris per a la definició de les característiques que ha de tenir el sistema i les funcions que aquest ha d'oferir.

B. Funcionalitats del Repositori

La unitat principal del repositori és l'activitat. Una activitat és una tasca que es duu a terme en una assignatura per l'obtenció dels objectius específics d'aquesta, i a més, els objectius de la competència transversal. Els usuaris poden crear activitats al repositori, editar-les i adjuntar-hi arxius. Aquestes activitats consten del nom de l'activitat, les assignatures a les que va destinada i una descripció. Se li poden afegir etiquetes per a realitzar una classificació, especificant quines competències treballa l'activitat, a quin curs es pot compartir i a quins estudis va destinada, així com també comentaris per part d'altres usuaris.

Una altra possibilitat és la de crear *enquestes*, les quals es caracteritzen per una pregunta que formula l'usuari i un seguit de respostes que els altres integrants del repositori poden contestar. A més, de la mateixa manera que a les activitats, els usuaris també poden fer-hi comentaris.

La funcionalitat més importat al repositori són les *cerques*, les quals es realitzen a partir de la classificació per categories que ha fet l'autor de l'activitat. A l'usuari se li presenten diferents opcions per trobar el material, per a què pugui triar la que millor escaigui a les seves necessitats. També disposen d'una opció per realitzar cerques a altres repositoris i de la possibilitat de fer cerques predefinides per etiquetes. Aquesta és la cerca més precisa, a on a l'usuari se li presenten les diferents etiquetes existents per a cada categoria. Aquest selecciona les que vol que estiguin en els seus criteris de cerca i llavors se li mostra un llistat amb tot el material disponible per a les etiquetes elegides.

Finalment, el repositori permet l'ús tant del català com del castellà, i els usuaris tenen en tot moment la possibilitat de canviar d'idioma entre un i l'altre. A més, quan un usuari crea una activitat, aquest pot afegir una categoria que indica en quin idioma estan els arxius que hi adjunta.

C. Estudi de les Eines Existents

Per a realitzar aquest projecte ha estat necessari crear una eina que permeti gestionar el contingut que s'anirà recopilant durant el seu ús. Per aquest motiu s'ha elegit utilitzar com a tecnologia base un sistema gestor de continguts (Content Management System, CMS). Aquests sistemes són programes que permeten crear una estructura de suport per a la creació i administració de continguts, principalment a pàgines web, per part dels participants [2]. Els CMS donen la possibilitat de compartir qualsevol tipus d'arxiu, ja sigui text, vídeo, àudio, imatges... com a contingut web. A més, alguns poden crear fluxos de treball per a la creació de contingut i la seva aprovació.

Per a decidir quin CMS emprar s'ha fet un estudi que determina quins cobreixen les necessitats del sistema, seguint el camí de la Universitat d'apostar per l'ús de programari lliure. Els dos sistemes analitzats han sigut Drupal i Joomla,

dos dels CMS més destacats dels darrer anys [3]. Aquests són paquets de programari lliure amb llicència GNU/GPL [4], escrits en PHP i emprant el sistema de base de dades MySQL per a emmagatzemar la informació. A la Fig. 2 es pot veure a mode il·lustratiu un exemple de les capes en l'arquitectura tecnològica de Drupal. Aquest manté les mateixes que Joomla, però a més implementa una capa d'abstracció que transforma les consultes SQL en consultes genèriques, permetent emprar diferents tipus de bases de dades sense haver de canviar el codi.

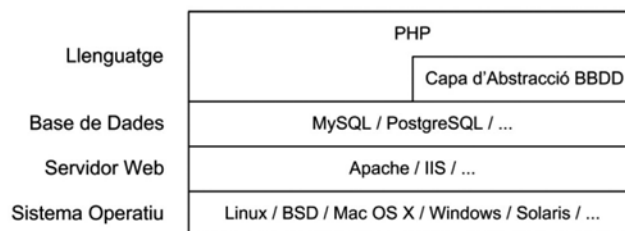


Fig. 2 Capes en l'arquitectura tecnològica de Drupal

A l'hora de realitzar la comparació, els aspectes que s'han tingut en compte s'han seleccionat per poder avaluar les capacitats dels dos sistemes en diferents àmbits necessaris a l'hora de crear el repositori, els quals es poden veure a la següent llista:

- Facilitat d'instal·lació
- Facilitat de creació d'una web
- Facilitat d'administració de contingut
- Flexibilitat gràfica i estructural
- Rols d'usuaris i flux de treball
- Suport multi idioma
- Extensibilitat i integració
- Manteniment del repositori i suport.

Els resultats obtinguts han determinat que Joomla és útil si es vol tenir un lloc senzill de manera ràpida i sempre que es pugui afrontar una categorització de contingut rígida i unes opcions de configuració i de disseny limitades. Per altra banda es pot usar Drupal si es vol alt rendiment, escalabilitat, un bon gestor de continguts i una flexibilitat de disseny significativa. Però això sí, s'ha d'estar preparat per a invertir molt de temps en aconseguir un lloc web professional.

D'aquesta manera s'ha elegit Drupal com a sistema gestor de continguts per al repositori, ja que compleix la major part dels requeriments obtinguts al principi de la fase d'anàlisi.

IV. DISSENY

La següent fase en la creació d'un software és la de disseny, a on es realitza el modelat del sistema en base als requeriments de sistema obtinguts, i a on s'extreu posteriorment el document de casos d'ús. Aquests serveixen per indicar com ha d'actuar el sistema amb l'usuari per aconseguir un objectiu específic, i és la tècnica més habitual per al modelat del comportament d'un sistema [5].

A. Descomposició del sistema

Per a realitzar un bon disseny és necessària una bona descomposició del sistema. Per aquest motiu la descomposició dels casos d'ús s'ha fet tenint en compte la *cohesió*, una mesura de la relació funcional entre els elements d'un mòdul que s'empra a la fase de disseny del cicle de vida d'un projecte de software [6]. Les funcionalitats dels mòduls amb una alta cohesió han d'executar entre ells una tasca ben definida en un programa. És a dir, les unitats de software que s'agrupen en un conjunt és degut a què estan relacionades en el desenvolupament d'una única funció específica.

Amb aquesta idea en ment s'han agrupat els diferents casos d'ús segons la funció bàsica que realitzen: l'autenticació, gestió d'usuaris, gestió de contingut i consultes. Aquests casos d'ús s'han dividit a la seva vegada en altres grups de subcasos d'ús, tal com es mostra a la Fig. 3. Aquí cada grup final conté els casos d'ús associats a aquesta funcionalitat. Per exemple, el CU2: *Gestió d'usuaris* és un grup final que conté els subcasos d'ús corresponents a les funcions relacionades amb els usuaris que interaccionen amb el sistema i que apareixen a la Fig. 1: crear un compte, validar-lo, denegar-lo, assignar un rol i modificar les dades personals.

B. Definició dels casos d'ús

Els casos d'ús s'han realitzat mitjançant una sèrie d'apartats que els defineixen, creant un document per a cada un d'ells a on es detallen les seves característiques. Primer apareix una *descripció* del cas d'ús tractat, especificant el que el sistema ha de permetre realitzar als usuaris que es defineixen. Després apareixen les *precondicions*, que descriuen l'estat del sistema abans d'executar-se el cas d'ús. Tot seguit es contemplen les *postcondicions* en cas d'èxit i en cas de fracàs, que descriuen respectivament l'estat del sistema després d'executar-se el cas d'ús en cas de finalitzar amb el resultat esperat per part dels usuaris principals i en cas de finalitzar sense aconseguir el resultat esperat. Després apareix la llista d'*usuaris* que empraran la funcionalitat, seguits per el *trigger* o acció que inicia la seva execució.

Al segon bloc es descriu la *seqüència normal* que segueix un cas d'ús, detallant pas a pas les accions que s'han de realitzar, per obtenir el resultat desitjat. De la mateixa manera, també es detallen les *excepcions*, mostrant una llista amb els passos i les accions que el sistema ha de realitzar en el cas que aparegui una situació que provoca una excepció.

Finalment, al darrer bloc apareixen el *súper cas d'ús*, que correspon al cas d'ús de nivell immediatament superior, i els *subcasos d'ús*, mitjançant una llista amb els de nivell immediatament inferior.

C. Diagrames dels casos d'ús

Una vegada que s'ha definit l'estructura que han de seguir els casos d'ús s'han creat cada un d'ells per a cada requeriment de sistema, fent servir UML per al disseny dels diagrames.

Aquest és un llenguatge gràfic de modelat de software emprat per especificar, dissenyar, construir i documentar sistemes, essent actualment el més conegut i utilitzat [7]. A la Fig. 4. es pot veure un exemple d'un diagrama de cas d'ús,

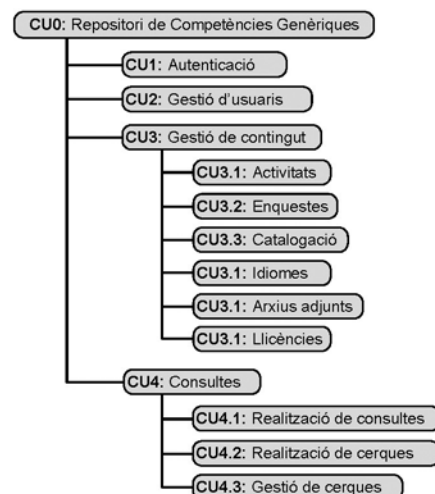


Fig. 3 Organització jeràrquica dels grups de casos d'ús.

concretament el que defineix el cas d'ús 0, i que apareix com a primer nivell a la Fig. 3.

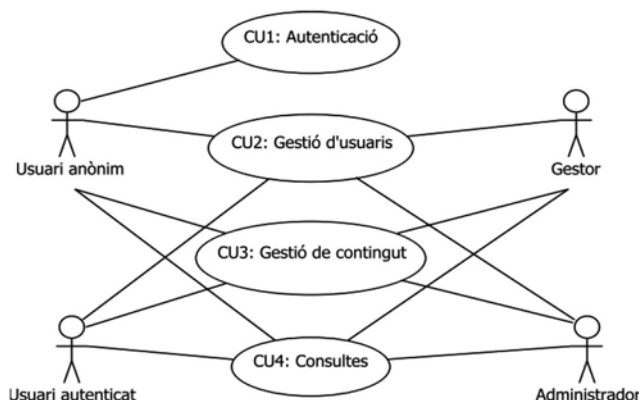


Fig. 4 Exemple per al diagrama del cas d'ús 0

V. IMPLEMENTACIÓ

La següent fase en el cicle de vida de creació del software és la implementació del nou sistema. Una vegada que ja s'ha fet l'estudi durant la fase d'anàlisi de quina és la millor eina per a la creació del repositori i s'han definit les seves funcions en la fase de disseny és necessari començar la seva implementació. Primer de tot s'ha realitzat l'instal·lació del CMS al servidor, per permetre així començar a configurar el gestor de continguts per a què compleixi amb els nostres requeriments.

Drupal ve amb una sèrie de capacitats implementades al seu nucli, fet que ha ajudat a elegir aquest sistema ja que moltes d'elles satisfan gran part dels requeriments establerts. A més, de cara als requeriments que no es satisfan nativament, permet una gran flexibilitat i adaptabilitat gràcies a la possibilitat d'ampliació mitjançant mòduls complementaris. A continuació es detallen les funcionalitats que s'hi poden trobar, juntament amb el procés que s'ha seguit per crear-les.

A. Gestió d'usuaris

Primer de tot, els usuaris s'han de poder registrar al repositori. Aquesta funcionalitat s'ha satisfet mitjançant el mòdul de Drupal *User*, que administra el registre d'usuaris i el sistema d'inici de sessió, permetent que així qualsevol usuari pugui crear un compte al repositori. Degut a què és un mòdul de sistema no es pot modificar el codi, i s'han d'utilitzar les opcions de configuració que proporciona Drupal per adaptar-lo a les pròpies necessitats.

En el nostre cas s'ha creat un formulari que els nous usuaris han d'omplir amb les seves dades: nom, llinatges, universitat i estudis. Una vegada omplit, aquest compte queda a l'espera de que un gestor el validi, moment en el qual el nou usuari rebra una confirmació de que ja pot accedir al repositori.

B. Rols i permisos

Els rols d'usuari permeten ajustar al detall la seguretat i l'administració del sistema. Al repositori s'han creat quatre tipus diferents de rols, alguns dels quals s'assignen automàticament quan l'usuari fa certes accions, i altres s'han d'assignar manualment. Concretament, els tipus de rol d'usuari que s'han creat són quatre: *usuari anònim*, *usuari autènticat*, *gestor* i *administrador*.

L'usuari anònim s'empra per als usuaris que no tenen compte o que encara no s'han autènticat. Quan un usuari accedeix amb el navegador al repositori automàticament se li assigna a aquella sessió un identificador, juntament amb el tipus de rol. En aquest cas, mentre que l'usuari no s'identifiqui la seva sessió té el rol d'usuari anònim. D'aquesta manera, l'usuari veu les activitats del repositori, però no els seus arxius adjunts.

L'usuari autènticat s'assigna automàticament a tots els usuaris que s'autèntiquen al repositori. Els usuaris autènticats tenen la possibilitat d'accedir al seu compte i canviar les seves dades, crear activitats i editar-les, pujar arxius, veure les activitats dels altres usuaris juntament amb els arxius adjunts i descarregar-los. També poden crear enquestes i votar-hi.

El rol de gestor l'ha d'assignar manualment un usuari administrador, i s'empra per a què els usuaris amb aquest rol puguin acceptar o denegar les peticions de nou compte d'usuari. No es vol que tothom pugui accedir al repositori simplement creant-se un compte, per aquest motiu existeix el rol de gestor: quan un usuari crea un compte aquest queda a l'espera de que un gestor el validi, el qual rep un correu avisant-lo de que hi ha un nou usuari al repositori. Aquesta funcionalitat s'ha aconseguit amb el mòdul *User registration notification*, el qual permet notificar a certs rols quan un nou usuari ha creat un compte.

Finalment, el rol d'administrador és el que permet utilitzar qualsevol tipus d'opció al repositori, i és aquest el que s'empra quan s'han de configurar funcions noves al sistema.

C. Creació i edició de contingut

Els usuaris registrats han de poder crear contingut al repositori: activitats i enquestes. Per aconseguir-ho s'han configurat dos tipus de contingut nous a Drupal mitjançant el mòdul de sistema Node, el qual permet enviar i mostrar

contingut. En el cas de les enquestes també és necessari un altre mòdul de sistema, *Poll*, que permet capturar vots sobre diferents temes en forma de preguntes d'opció múltiple. Aquests continguts es creen adherits a una de les sis llicències *Creative Commons*, gràcies al mòdul *Creative Commons Lite*.

Drupal per defecte mostra una llista d'arxius adjunts a cada node creat, però l'estil que els hi aplica no especifica visualment quin tipus d'arxiu és. El que s'ha fet és, mitjançant el mòdul *shiny upload*, afegir una icona a cada arxiu segons el seu tipus MIME. A la Fig. 5. es veu un exemple de com apareixien els arxius adjunts a l'inici i com apareixen emprant aquest mòdul:

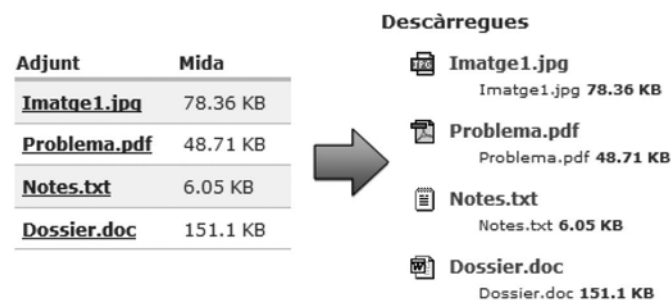


Fig. 5 Arxius adjunts emprant icones segons el tipus MIME.

Finalment, de cara a l'edició d'aquest contingut, s'ha optat per configurar el mòdul *Wysiwyg*, que permet que els usuaris puguin editar contingut amb un editor visual.

D. Consultes i cerques

És evident que de res serveix anar creant contingut si després no es té cap mètode per a poder visualitzar-ho. Tot el contingut de Drupal una vegada es crea automàticament ja està llest per a ser consultat. El problema ve en que aquest contingut s'ha d'agrupar d'alguna manera per a tenir-ho tot al mateix lloc i fer així les consultes més senzilles i ràpides. Per solucionar-ho s'ha emprat el mòdul *Views*, que permet crear llistes personalitzades i consultes a la base de dades. Amb aquest s'ha creat un llistat amb tot el contingut ordenat per data de creació, i mostrant el títol, el text del contingut, les paraules clau associades, el nom dels arxius adjunts i els comentaris que han escrit altres usuaris.

També és possible emprar el cercador del repositori. El motor de cerca de Drupal manté un índex de les paraules trobades al contingut del lloc, i aquest es va actualitzant automàticament cada vegada que es crea un nou contingut.



Fig. 6 Exemple de núvols d'etiquetes creats al repositori.

A més del motor de cerca, i per a què l'usuari tingui una visió general del tipus de material que pot trobar i la quantitat de material que hi ha a cada categoria, se li proporciona també un núvol d'etiquetes. Aquest consisteix en una representació visual de les etiquetes usades per a catalogar el contingut, ordenades alfabèticament i a on les més utilitzades es representen amb un tamany de lletra més gran que la resta. Aquesta funcionalitat s'aconsegueix mitjançant el mòdul *Tagadelic*, que crea distints núvols d'etiquetes com els que apareixen a la Fig.6.

L'inconvenient que té aquest mòdul és que no permet de forma nativa la traducció del contingut que mostra. Per solucionar-ho s'ha canviat part del codi modificant la funció *theme_tagadelic_weighted*, quedant com es mostra a continuació:

```
function theme_tagadelic_weighted($terms) {
    $output = '';
    foreach ($terms as $term) {
        if (module_exists("i18ntaxonomy")) {
            $terms = i18ntaxonomy_localize_terms($terms);
        }
        $output .= l($term->name,
            taxonomy_term_path($term),
            array('attributes' =>
                array('class' => 'tagadeliclevel',
                    $term->weight,
                    'rel' => 'tag'))
            . "\n";
    }
    return $output;
}
```

Codi 1. Modificació del mòdul Tagadelic per admetre traduir etiquetes.

D'aquesta manera quan el mòdul processa cada terme primer comprova si està activat el mòdul *i18ntaxonomy* (que permet la traducció de vocabulari) i l'empra per traduir cada un dels termes abans de crear la sortida i mostrar-ho per pantalla.

E. Cerques a altres repositoris

Per aconseguir aquesta funcionalitat s'ha emprat el mòdul *Services*, una API per crear web services i així poder intercanviar dades entre els distints repositoris. Emprant un web service es pot aconseguir que una aplicació externa tingui accés als serveis que ofereix el repositori, sense haver de intercanviar usuaris ni claus d'accés a la pròpia base de dades.

Per a l'intercanvi de dades s'ha emprat XML-RPC, un protocol de crida a procediments remots que fa servir XML per codificar les crides HTTP com a mecanisme de transport de dades [8]. A la Fig. 7. es pot veure el modelat del comportament que es segueix en aquestes cerques:



Fig. 7 Flux de dades durant una petició al web service.

El client fa una cerca escrivint les paraules clau a cercar i envia la petició. Aquesta es connecta al web service que ofereix el servidor, rep la petició i fa la cerca a la base de dades, de manera totalment independent de la transacció rebuda. Al obtenir les dades aquest retorna un document XML al repositori client, a on es visualitzen els resultats cercats.

Per a complementar aquesta estructura de petició-resposta, i ja que no s'ha trobat cap mòdul que s'ajusti a les nostres necessitats, s'ha creat *RCG Share*. Aquest nou mòdul permet capturar les paraules clau a cercar, crear la petició al web service, rebre el document XML de resposta, processar-lo i visualitzar els resultats. Basant-se en el protocol XML-RPC, permet la connexió a altres repositoris modificant la funció *search* de Drupal. El codi de la funció principal es pot veure a continuació:

```
function rcg_share_search($keys) {
    $web_services = array();
    //Recorrem i cerquem a cada un dels web services.
    $find = array();
    foreach ($web_services as $url) {
        $results = xmlrpc($url, 'search.nodes', $keys);
        if (xmlrpc_error()) {
            $error = xmlrpc_error();
        } else {
            $find = array_merge($find, $results);
        }
    }
    return $find;
}
```

Codi 2. Funció principal del mòdul de cerques RCG Share.

La variable *\$web_services* és un array que emmagatzema les direccions del servidor de cada repositori habilitat per rebre peticions. Així, el mòdul accedeix a la variable, i per cada direcció que hi troba realitza una connexió XML-RPC al web service *search.nodes* amb les paraules clau sol·licitades. Si aquesta petició resulta en error (per exemple, si no ha trobat resultats) es guarda per a què Drupal pugui mostrar un missatge posteriorment de que no hi ha resultats. En cas de que no hi hagi error, es van afegint els resultats a la variable *\$find* fins que no queden més repositoris per consultar. Llavors es retorna aquesta variable creada amb l'estructura necessària per a què Drupal l'empli per a visualitzar els resultats.

F. Puntuació de contingut

Els usuaris registrats també poden votar el contingut creat, gràcies al mòdul *Fivestar*. Aquest afegeix un sistema de vots a cada activitat que es crea al repositori, com es pot veure a la Fig. 8:



Fig. 8 Exemple de vots per a una activitat.

G. Vocabulari i sistema de classificació

Per a realitzar la classificació s'ha decidit utilitzar etiquetes, paraules clau associades al contingut que donen una gran flexibilitat a l'hora de fer cerques i catalogacions. El mòdul emprat ha estat *Taxonomy*, configurat per a què quan un usuari crea una activitat li apareixi una pàgina a on pot triar les categories que vol assignar-hi, basant-se en les que ja hi ha definides.

Les categories definides són cinc: competència transversal, estudis, assignatura, curs i idioma del contingut adjunt, basant-se en les etiquetes de competències transversals definides en el Llibre Blanc del títol de grau en Enginyeria Informàtica [9]. Les categories tenen un conjunt inicial d'etiquetes, exceptuant la d'assignatura, que és de format lliure. Tal com es veu a la Fig. 9, és el propi usuari el que pot escriure el que vulgui, deixant llibertat en el títol de l'assignatura.

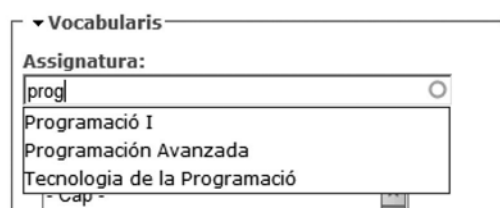


Fig. 9 Mostra de la catalogació segons l'assignatura amb format lliure.

H. Aspecte gràfic

S'ha optat per una distribució de tres columnes, ja que és un esquema àmpliament utilitzat i no suposa un disseny estrany de cara a l'usuari. A la Fig. 10, es pot veure el disseny del repositori. La columna de l'esquerra conté el menú de navegació, la columna de la dreta mostra varies seccions com el canvi d'idioma, el cercador, etc., i a la columna central és a on apareix tota la informació.



Fig. 10 Mostra del disseny del repositori a on es pot veure la distribució de tres columnes que s'ha emprat.

VI. CONCLUSIONS

Amb aquest projecte aquí presentat ha sorgit el Repositori de Competències Genèriques, un espai col·laboratiu entre professors a on aquests poden crear activitats, cercar-les,

classificar-les, comentar-les i votar-les. Poden consultar activitats creades per altres professors, descarregar-se el contingut, adaptar-lo a la seva metodologia de treball i utilitzar-lo en la seva docència. El recent inici de l'EEES requereix de contribucions com la que s'ha implementat en aquest projecte, i hem pogut comprovar que aquesta eina era necessària.

Per altra banda, com a tot projecte de software, el seu cicle de vida no acaba després de la implementació i la seva posterior instal·lació. Per a què el projecte pugui mantenir-se és necessari el seu manteniment, de manera que l'eina no quedi desfasada. Aquesta darrera fase contempla les modificacions del producte, tant de codi com de documentació, les possibles migracions a un nou entorn, el suport als usuaris, etc. A més també s'han de tenir en compte les modificacions necessàries en cas de desitjar noves funcionalitats.

Finalment, és necessària l'activitat continuada al repositori per part dels administradors i gestors, ja que són aquests els encarregats d'acceptar o denegar les peticions d'alta de nous usuaris al sistema. D'aquesta manera el repositori podrà seguir en marxa permetent l'accés a nous usuaris amb nou material, facilitant així al professorat la tasca de recollir i organitzar les competències genèriques.

REFERÈNCIES

- [1] E. Amengual, X. Gilabert, A. Mas, *Tècniques de recopilació de dades*. Apunts d'Enginyeria del Software I. Tema 6, diapositiva 5. Universitat de les Illes Balears.
- [2] Wikipedia, la enciclopèdia lliure (2011) Sistema de Gestión de Contenidos. [Online]. Disponible: http://es.wikipedia.org/wiki/Sistema_de_gestion_de_contenidos
- [3] Packt Publishing, Llista de guanyadors i premis otorgats als Annual Open Source CMS Awards. [Online]. Disponible: <http://www.packtpub.com/article/open-source-cms-award-previous-winners>.
- [4] Free Software Foundation, GNU General Public License. Versió 3. [Online]. Disponible: <http://www.gnu.org/copyleft/gpl.html>.
- [5] G. Fontanet, *Requeriments d'usuari i de sistema*. Apunts d'Enginyeria del Software II, pàgina 3. Universitat de les Illes Balears.
- [6] E. Amengual, *El procés de Disseny del Sistema d'Informació II*. Apunts d'Enginyeria del Software I. Tema 10, diapositiva 80. Universitat de les Illes Balears.
- [7] Object Management Group. Getting Started with UML. [Online]. Disponible: <http://www.uml.org/>
- [8] Wikipedia, la enciclopèdia lliure, XML-RPC. [Online]. Disponible: <http://ca.wikipedia.org/wiki/XML-RPC>
- [9] J. Casasnovas, *Libro Blanco. Título de grado en Ingeniería Informática*. ANECA. Juny 2005.