



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FI DE CARRERA

TÍTOL: Desenvolupament d'una Blogosfera

AUTOR: David Gómez Fernández

TITULACIÓ: Enginyeria Tècnica en Informàtica de Gestió

DIRECTOR: Xavier Raurell Guxens / Jordi Daudé Ventura

DEPARTAMENT: Llenguatges i Sistemes Informàtics

DATA: Gener de 2011

TÍTOL: Desenvolupament d'una Blogosfera

COGNOMS: Gómez Fernández NOM: David
TITULACIÓ: Enginyeria Tècnica en Informàtica de Gestió
ESPECIALITAT: Informàtica PLA:

DIRECTOR: Xavier Raurell Guxens / Jordi Daudé Ventura
DEPARTAMENT: Llenguatges i Sistemes Informàtics

QUALIFICACIÓ DEL PFC

TRIBUNAL

PRESIDENT

SECRETARI

VOCAL

Neus Català Roig Bernardino Casas Fernández Immaculada Massana Hugas

DATA DE LECTURA:

Aquest Projecte té en compte aspectes mediambientals: Sí No

PROJECTE FI DE CARRERA

RESUM (màxim 50 línies)

Aquest projecte pretén desenvolupar una blogosfera partint d'una eina base, *WordPress*. La blogosfera ha de ésser adaptada i dedicada per un col·lectiu de persones dintre d'un àmbit concret, els col·legiats del Col·legi d'Enginyers Industrials de Catalunya. Partint d'un blog normal, tracta de crear un portal web adaptat a les necessitats del col·lectiu abans esmentat.

Explica com dur a terme una feina dintre d'un ambient de treball, amb problemes reals a resoldre i amb la manca d'informació que això comporta.

També s'explica el perquè fer-ho amb *WordPress* i no amb altres eines, com integrar-lo dintre d'un sistema ja implementat i com connectar una eina de blogs a una totalment diferent per tal de transferir informació entre elles.

Paraules clau (màxim 10):

Blog	WordPress	CMS	PHP
IBM Portal	Quickr		

Proposta original

Descripció:

Aquest projecte intentarà fer un desenvolupament complet d'una blogosfera dintre d'un entorn real, com es el Col·legi d'Enginyers Industrials de Catalunya, per tal de mantenir informats, actualitzats i comunicats entre sí a tots els col·legiats amb les últimes tecnologies de informació.

Tractaré de desenvolupar un portal dinàmic on els enginyers industrials col·legiats puguin posar els seus articles i les seves activitats organitzades per departaments.

A més s'inclourà una opció de compra per els articles que els autors pugin en forma de llibre.

També hi haurà una part dedicada l'alumnat d'enginyers industrials on puguin tindre referències sobre l'actualitat de tot el referent amb la seva carrera, així com una borsa de treball per accedir al món laboral.

Tot això estarà connectat de forma que els usuaris registrats tinguin constància dels nous articles del seu interès mitjançant RSS i newsletters.

Les tecnologies escollides per fer el projecte són l'eina de gestió de continguts IBM Portal, Oracle com a motor de base de dades i Java com a llenguatge de programació dels mòduls.

Objectius:

- Aprendre a desenvolupar gestors de continguts per tal que els usuaris puguin fer les seves pròpies publicacions tot seguint una pauta.
- Connectar i mantenir de forma eficient bases de dades de grans dimensions.
- Aprendre a fer ús de diferents llenguatges de programació per mantenir de forma adient cada espai de treball.
- Resoldre un problema del món real, amb les necessitats corresponents, com és desenvolupar un projecte destinat a usuaris concrets.

Agraïments

Aquest projecte va dedicat al Col·legi d'Enginyers Industrials de Catalunya, ja que m'han donat l'oportunitat de començar una vida en el món laboral.

En especial vull agrair a Jesús i Xavier la seva ajuda, el seu suport i la seva confiança.

I a la resta d'amics i familiars que han estat al costat meu i que m'han donat els ànims que molts cops m'han faltat a mi.

Per últim, aquest projecte també va dedicat a Dani, Emilio i Virginia donat que han fet que els anys d'universitat fossin els millors viscuts fins al moment.

Moltes gràcies a tots.

Índex

1.	Introducció	9
1.1	Motivació	9
1.2	Descripció general.....	9
1.3	Quin és l'objectiu?	11
1.4	Justificació de l'eina escollida.....	12
1.4.1	IBM Portal. Perquè no?	12
2.	Els CMS. <i>WordPress</i>	13
2.1	Què és un CMS?.....	13
2.2	Com funcionen?	13
2.3	Tipus de CMS	13
2.4	<i>WordPress</i>	14
2.4.1	Què és?	14
2.4.2	Quins tipus hi ha.....	15
2.4.3	Instal·lació de <i>WordPress</i>	16
3.	Començament. De blog a multiblogging.....	17
3.1	Punt de partida.....	17
3.2	Funcionament del taulell de <i>WordPress</i>	17
3.2.1	Usuaris de <i>WordPress</i>	17
3.2.2	Menús	18
3.3	Conversió a <i>multisite</i>	24
3.3.1	Què és el <i>multisite</i> i per què serveix?	24
3.3.2	Estructura interna de <i>WordPress</i>	25
3.3.3	Activació del <i>multisite</i>	28
3.4	El <i>multisite</i> . Què tenim de nou?	31
4.	Portal inicial	34
4.1	Què tenim fins ara	34
4.2	Cóm mostrem el que tenim	34
5.	Model conceptual del sistema	37
6.	Menú jeràrquic de blogs.....	39
6.1	Què és?.....	39
6.2	Canvis en el model conceptual del sistema	39
6.3	Disseny de la solució. Cas d'ús AfegirDadesBlog	40
6.3.1	Cas d'ús AfegirDadesBlog	40
6.3.2	Responsabilitats del cas d'ús	41
6.3.3	Esdeveniments.....	41
6.3.4	Detecció de paràmetres.....	41
6.3.5	Contractes i diagrames de col·laboració	41
6.3.6	Visibilitats del cas d'ús AfegirDadesBlog.....	43
6.3.7	Model de components del CU AfegirDadesBlog	44
6.3.8	Aspectes destacats a l'hora de dissenyar	44
6.4	Implementació de la solució.....	44
6.4.1	Classe Jerarquia	44
6.4.2	Classe Site	45
6.4.3	Classe Blog.....	45
6.4.4	Sidebar.....	45
6.5	Interfície del cas d'us	47
7.	Pàgina principal. COEIC Home	48
7.1	Descripció	48
7.2	Canvis en el model conceptual del sistema	48
7.3	Disseny de la solució. Cas d'ús Introduir articles rellevants	49
7.3.1	Cas d'ús IntroduirArticlesRellevants.....	50
7.3.2	Responsabilitats del cas d'ús	50
7.3.3	Esdeveniments.....	50
7.3.4	Detecció de paràmetres.....	51
7.3.5	Contractes i diagrames de col·laboració	51
7.3.6	Visibilitats del cas d'ús IntroduirArticlesRellevants	54
7.3.7	Model de components del cas d'ús IntroduirArticlesRellevants	54
7.3.8	Aspectes destacats a l'hora de dissenyar	55
7.4	Cas d'us Esborrar articles rellevants	56

7.4.1	Cas d'ús EsborraArticles	56
7.4.2	Responsabilitats del cas d'ús	56
7.4.3	Esdeveniments	56
7.4.4	Detecció de paràmetres	56
7.4.5	Contractes i diagrames de col·laboració	57
7.4.6	Visibilitats del cas d'ús EsborraArticles	57
7.4.7	Model de components del cas d'ús EsborraArticles	58
7.4.8	Aspectes destacats a l'hora de dissenyar	58
7.5	Implementació de la solució	58
7.5.1	Classe Coeic	59
7.5.2	Classe Site	60
7.5.3	Classe Blog	60
7.5.4	Classe Post	60
7.5.5	Classe Usuari	60
7.5.6	Índex	60
7.6	Interfície del cas d'ús	61
8.	Quickr to WordPress	62
8.1	Descripció	62
8.2	Canvis en el model conceptual del sistema	62
8.3	Disseny de la solució. Cas d'ús ImportarArticle	63
8.3.1	Cas d'ús ImportarArticle	63
8.3.2	Responsabilitats del cas d'ús	63
8.3.3	Esdeveniments	63
8.3.4	Detecció de paràmetres	64
8.3.5	Contractes i diagrames de col·laboració	64
8.3.6	Visibilitats del cas d'ús ImportarArticle	66
8.3.7	Model de components del CU ImportarArticle	67
8.3.8	Aspectes destacats a l'hora de dissenyar	67
8.4	Implementació de la solució	68
8.4.1	Classe xuclablog	68
8.4.2	Classe Site	69
8.4.3	Classe Post	69
8.4.4	Classe PostQ	69
8.4.5	Classe PostW	69
8.4.6	Classe Blog	70
8.4.7	Classe Quickr	70
8.5	Una altre solució. Exportació dels articles	72
8.6	Interfícies del cas d'ús	73
8.6.1	Importació dels articles	73
8.6.2	Exportació dels articles	74
9.	Funcionalitats diverses implementades	75
9.1	Inserció, eliminació i actualització d'usuaris	75
9.2	Fitxa d'usuaris	76
9.3	Restricció de seguretat per la fase de proves	77
9.4	Personalització del tema	78
10.	Manual d'usuari	80
10.1	Requeriments del sistema	80
10.2	Manual d'instal·lació de <i>WordPress</i>	80
10.2.1	Instal·lació d'un blog normal	80
10.2.2	Activació del multisite	83
10.3	Instal·lacions posteriors a l'activació del <i>multisite</i>	85
10.4	Accés a l'aplicació	86
10.5	Manual del superadministrador	86
10.5.1	Cercar un lloc o un usuari	87
10.5.2	Veure un llistat dels blogs	87
10.5.3	Crear un blog nou	88
10.5.4	Creació d'usuaris	88
10.5.5	Activar / Desactivar temes	89
10.5.6	Modificar les opcions de la blogosfera	90
10.5.7	Assignació d'usuaris i rols a un blog	91

10.5.8	Esborrar un usuari d'un blog	92
10.5.9	Activar un plugin per tota la xarxa	92
10.6	Manual d'administrador.....	94
10.6.1	Vista visual del taulell d'administrador	94
10.6.2	Aplicació d'un tema	94
10.6.3	Ús del plugin COEIC Home	96
10.6.4	Ús del plugin Jerarquia de blogs.....	97
10.6.5	Afegir un widget.....	98
10.7	Manual de l'editor.....	99
10.7.1	Vista visual del taulell de l'editor.....	99
10.7.2	Afegir o editar una plana nova	99
10.7.3	Administració de comentaris	100
10.7.1	Importar articles de Quickr. Quickr To WordPress	101
10.8	Manual de l'autor i del col·laborador.....	102
10.8.1	Vista del taulell de l'autor	102
10.8.2	Vista del taulell del col·laborador.....	103
10.8.3	Afegir articles	103
10.9	Manual d'usuari del subscriptor.....	105
10.9.1	Els meus llocs	105
10.10	Altres opcions	105
10.10.1	Iniciar sessió.....	105
10.10.2	Menú meta un cop iniciada la sessió	106
10.10.3	Inserir, modificar i esborrar usuaris	106
10.10.4	Recollida d'articles per la fitxa d'usuaris.....	107
10.10.5	Exportar articles de Quickr a WordPress	107
11.	Glossari.....	109
12.	Annexa.....	112
12.1	Plugin Jerarquia de Blogs	112
12.1.1	Classe principal del plugin (jerarquia.php).....	112
12.1.2	Classe corresponent al controlador del plugin (classSite.php)	115
12.1.3	Classe corresponent al component Blog del plugin (classBlog.php).....	116
12.1.4	Arxiu que pinta el menú dintre del tema (sidebar.php)	120
12.2	Plugin COEIC Home	122
12.2.1	Classe principal del plugin (coec2.php).....	122
12.2.2	Classe site (classSite.php).....	130
12.2.3	Classe blog (classBlog.php)	132
12.2.4	Classe post (classPost.php).....	135
12.2.5	Pàgina principal del Theme, modificada per al plugin (index.php).....	137
12.3	Plugin Quickr To <i>WordPress</i>	141
12.3.1	Classe principal del plugin (quickrtowp.php).....	141
12.3.2	Classe Site (classSite.php).....	146
12.3.3	Classe Post (classPostGen.php).....	148
12.3.4	Classe PostQ (classPostQ.php).....	150
12.3.5	Classe PostWordPress (classPost.php)	152
12.3.6	Classe Blog (classBlog.php).....	154
12.3.7	Classe Quickr (classQuickr.php)	157
12.4	Quickr To <i>WordPress</i> . Versió Exportació d'articles des de Quickr.....	166
12.4.1	Interfície gràfica del portlet (articles.jsp)	166
12.4.2	Controlador del portlet (QuickrToWordPressPortlet.java).....	168
12.4.3	Classe Quickr (Quickr.java)	169
12.4.4	Connexió entre <i>Quickr</i> i <i>WordPress</i> (signinremot.php)	173
12.5	Funcionalitats diverses.....	174
12.5.1	Inserció, modificació i eliminació d'usuaris (usuaris.php)	174
12.5.2	Fitxa d'usuari (articles.php)	176
12.5.3	Restricció de seguretat a l'accedir a un article (single.php)	177
13.	Conclusions	178
14.	Bibliografia.....	179

1. Introducció

1.1 Motivació

He escollit aquest tema (*Desenvolupament d'una blogosfera*) donat que és un tema d'actualitat e important en el món dels sistemes d'informació.

Des de fa ja uns anys els blogs s'han anat convertint cada cop més en una eina de difusió d'informació molt utilitzada. El senzill mecanisme en el que es basen i la velocitat per introduir i difondre nous continguts fan que s'hagin convertit en tota una revolució en la comunicació.

Aquest fet ha produït una revolució social i un canvi de mentalitat. D'aquesta manera qualsevol pot fer-se la seva pàgina personal i compartir-la amb tothom, cosa que ha produït que hi hagi molta més comunicació entre la gent i molta més riquesa i varietat de continguts a Internet.

Tot això, juntament amb la necessitat del Col·legi d'Enginyers Industrials de Catalunya de recopilar les notícies i opinions dels enginyers han fet que hem decanti per escollir aquest tema.

1.2 Descripció general

El Col·legi Oficial d'Enginyers Industrials de Catalunya (d'ara en endavant COEIC), té la necessitat de fer arribar les notícies d'actualitat i altres temes d'interès als enginyers industrials. Per això mateix han decidit implementar un sistema de blogs on la gent pugui estar assabentada i on puguin participar i fer les seves propostes i valoracions del contingut. Fent que hi hagi una comunicació fluida entre tots ells.

Comissions

El COEIC té un departament especialitzat per els assumptes que volem tractar, aquest departament s'anomena Comissions. Les comissions són un punt de participació i retrobament format per un grup d'experts especialitzats en un àrea determinada.

Les àrees en les que es subdivideix són totalment independents les unes de les altres. Cadascuna tracta d'un tema diferent. D'aquesta manera obtenim les següents comunitats:

- *Acció i Cooperació*: s'encarrega de tractar temes socials portats a terme per ONGs.
- *Acció Professional*: l'objectiu és reforçar les relacions amb les administracions receptores del treball dels enginyers i amb altres institucions relacionades per incrementar la presència en les institucions.
- *Administracions Públiques*: està formada per enginyers que han treballat en administracions públiques i que pretenen oferir un millor servei a la societat.
- *Construcció*: representa al Col·legi davant d'altres institucions o estaments.
- *Cultura*: l'objectiu és organitzar i promocionar activitats en l'àmbit de la cultura.
- *Energia*: es dedica a preparar una resposta a les propostes del Pla d'Energia a Catalunya.
- *Esports*: s'encarrega de fomentar la participació de la massa social en activitats esportives com a activitat lúdica que fomenti la cohesió entre els socis.

- *Formació*: el seu objectiu és el d'organitzar activitats formatives en condicions de privilegi per al col·lectiu, en especial en temes professionals.
- *Gestió Empresarial*: té per objectius investigar, difondre i promoure les diverses tècniques de gestió empresarial.
- *Jubilats*: estimulen i desenvolupen activitats interessants tant per al seu col·lectiu com per als que els envolta.
- *Medi Ambient*: s'encarrega de debatre i difondre temes d'actualitat relacionats amb les millors tecnologies disponibles i la gestió de recursos existents.
- *Mobilitat i Transport*: crea una línia d'opinió que permet al Col·legi posicionar-se davant aspectes relacionats amb la mobilitat i el transport.
- *Noves Iniciatives*: té com a objectiu fer una primera aproximació a la carrera professional dels anomenats Enginyers Joves per tal de poder prestar uns serveis més pròxims des del Col·legi.
- *Packaging*: s'encarrega de donar suport al sector de l'enginyeria de l'envàs, l'emalatge i el processos d'envasat i aconduïment de productes.
- *Relacions amb les escoles d'enginyeria i les Universitats*: té per objectiu impulsar accions referents als nous plans d'estudis i al foment de la professió entre els més joves, vetllant per la igualtat d'oportunitats.
- *Seguretat*: intenta promoure, desenvolupar i difondre els coneixements sobre riscos i seguretats que afecten a les persones i al patrimoni en totes les activitats i mitjans que poden ser d'interès per als col·legiats i per la societat.
- *TIC*: pretén donar a conèixer l'ús de les TIC a les empreses per tal de millorar la seva productivitat i desenvolupament d'innovació.
- *Urbanisme*: té com a objectiu reforçar la figura de l'enginyer en el camp del urbanisme.

A cada comissió es troben una sèrie de col·legiats (els col·legiats són enginyers amb titulació), però cada un forma part d'una única comissió. A més cada comissió té un president, que no és més que un col·legiat que representa aquesta comissió.

A mode excepcional, algunes comissions tenen una subcomissió que correspon a una petita associació de la pròpia comissió. Un exemple d'això és la comissió de *Cultura* que té l'associació de la *Coral d'Enginyers* o la comissió d'*Esports* que té l'associació d'*Esports Intercol·legials*.

Col·legiats

Els col·legiats són els enginyers industrials titulats o a punt de titular-se que podran fer ús de la blogosfera. En podem distingir de dos tipus: els que són membres d'alguna comissió i els que no ho són.

Els col·legiats que són membres d'alguna comissió són els que tenen dret tant a pujar articles com a fer comentaris d'altres articles (siguin de la seva comissió o d'una altre). D'aquests en volem tenir enregistrada una fitxa bibliogràfica amb dades de contacte (nom, cognoms, mail, número de col·legiat), amb els articles que ha pujat, i amb una petita bibliografia de la seva trajectòria professional.

Els col·legiats que no són membres de cap comissió són aquells que la seva interacció amb la blogosfera es limita a poder llegir i comentar els articles pujats.

Articles

Els articles pròpiament dits, són els posts que conté cadascun dels blogs. Un post no es més que una nova entrada en un blog. En termes de la blogosfera, es cadascun dels articles que els col·legiats d'una comissió redacta dintre d'ella.

En podem distingir de diferents tipus d'articles depenent del contingut que tenen. Així doncs, no serà el mateix un article que presenti un llibre, que un article que informi amb un vídeo o un àudio (podcast) o un que simplement informi d'un succés.

RSS i Newsletters

Els RSS serveixen per mantenir informats a la gent, mitjançant un sistema de notificació cada cop que ha succeït alguna cosa. En el cas del COEIC, volem que tots els usuaris estiguin informats de cada tema d'interès que aparegui a la blogosfera. Per tant, caldrà implantar un sistema de RSS que avisi a l'usuari de l'aparició de nous articles a través del seu lector de RSS.

El cas dels Newsletters és diferent. Un Newsletter és una recopilació d'articles que s'han pujat durant un període de temps determinat, seria l'equivalent a una revista. D'aquesta manera el que volem és que, un cop al mes, tots els usuaris rebin un correu electrònic en el que se'ls informi de totes les noves notícies que s'han anat publicant durant el mes anterior.

Altres consideracions

Tenint en compte que el projecte és un intent de resoldre un problema real dintre d'un món real, no estic tractant un tema tancat, és a dir, les necessitats poden anar variant al llarg del temps i per tant poden haver-hi variacions i ampliacions en el problema a resoldre, tot depenent de la dificultat i de la temporalitat del projecte.

En el cas que es vagin incloent noves prestacions o hi hagin canvis significatius en els punts ja esmentats, s'informarà al llarg del projecte quan es tracti el tema.

1.3 Quin és l'objectiu?

El principal objectiu d'aquest projecte és el de resoldre un problema real amb tot el que això comporta. Tant les variacions en les necessitats de la gent, com la manca d'informació i les diferents suposicions semàntiques, són alguns dels problemes que s'intentaran combatre.

L'altre objectiu del projecte és realitzar un producte, modelant una eina base. Es tracta d'aconseguir millorar l'eina que fem servir per tal de personalitzar-la i treure el màxim rendiment per les nostres necessitats.

No estem fent un producte nou pròpiament dit, però tampoc pretenem imitar un altre, sinó que s'intenta fer un producte a mida.

1.4 Justificació de l'eina escollida

En un principi havíem seleccionat IBM Portal com a eina principal del projecte, com ho indica a la proposta original, a partir de la qual modelar la blogosfera, però finalment, després d'estudiar-la hem detectat que té moltes mancances que fan la feina molt feixuga. Per tant, ens hem decantat per el gestor de continguts *WordPress*.

1.4.1 IBM Portal. Perquè no?

IBM Portal es tracta d'una eina de pagament de gestió de portals web. Està destinada principalment a empreses que volen crear un portal personalitzat o una intranet per l'empresa.

Funciona mitjançant la creació de portlets, frameworks que componen la pàgina. Cada portlet conté diversos components que el componen.

Fa ús de tecnologia Java i permet connectar a bases de dades de diferents tipus, com Oracle o *MySQL*.

Encara que és una eina molt estructurada i senzilla d'utilitzar, a més de ser l'eina que usa el COEIC per desenvolupar el seu portal web i la seva intranet, té mancances que resulten difícils de gestionar.

IBM Portal porta una aplicació per crear blogs, però es troba molt limitada en quant a personalització, donat que el fet que estigui ja tot predefinit fa molt difícil la personalització en quant a poder millorar el que es dóna i fer-lo a mida.

A més, no permet fer una correcta gestió dels RSS, amb el que obteníem que una bona part dels propòsits que volíem aconseguir no els podíem realitzar correctament.

En cas d'escollir finalment aquesta eina no hauríem pogut fer ús de pràcticament cap dels seus beneficis, cosa que hauria provocat que es tingués que construir el producte final des de zero, quan això resulta inútil tenint en compte que hi ha formes de facilitar el treball.

Si no fos per els problemes abans esmentats, hagués estat la millor eina per els següents motius:

- L'ús del llenguatge Java fa que tinguem una gran flexibilitat i eficiència a l'hora de programar tots els mòduls.
- El poder connectar amb quasi qualsevol base de dades hauria fet que es pogués enllaçar amb l'actual base de dades interna del COEIC, evitant així la duplicitat de dades, estalviant espai de memòria i fent possible accedir a moltes dades ja existents. Igualment s'intentarà donar una possible solució a l'acoblament de dades amb l'eina finalment escollida.
- El fet de tenir tot integrat dintre d'una mateixa aplicació hauria fet que poguéssim fer ús d'altres components ja programats i d'altres avantatges com son els estils de la pàgina web final.

Per tant, després d'estudiar detingudament la possibilitat de fer ús d'aquesta eina i de veure que es perd més temps de la productivitat que es guanya, l'hem descartat per decidir-nos per *WordPress*.

2. Els CMS. *WordPress*

2.1 Què és un CMS?

Els CMS (*Content Manager System*) o Sistemes Gestors de Continguts són programes que permeten crear un estructura per la creació i administració de continguts per part de l'usuari.

Consisteixen en una interfície simple connectada a una base de dades on s'allotja tot el contingut del lloc. Es caracteritzen perquè separen de forma independent el contingut del disseny del lloc. D'aquesta manera es pot estructurar el contingut i fer el que volem amb ell i després aplicar-li un disseny o un altre, sense que hi hagin problemes d'incompatibilitat i sense tenir que modificar el format del contingut.

2.2 Com funcionen?

Els CMS sempre funcionen en el servidor web on estigui allotjat el portal. Per accedir, crear i modificar els continguts es fa mitjançant el portal web que es crea, donat que disposen d'un taulell d'administració que permet fer aquesta feina ràpida i de forma senzilla. En canvi, per afegir noves plantilles de presentació o per afegir plugins que millorin el contingut del portal, s'ha de fer ús del protocol FTP (*File Transfer Protocol*) per la pujada d'arxius.

Quan un usuari entra al portal que conté el CMS internament aquest primer accedeix a la base de dades per recuperar les opcions que té definides en quant a plantilla de presentació escollida, menús seleccionats i articles publicats i després pinta la plantilla de presentació amb els continguts corresponents que acaba de recuperar.

Normalment es componen per una jerarquia de nivells per als diferents usuaris, per diferenciar les capacitats que té cadascú dintre del portal.

D'aquesta manera ens trobem al cap de la jerarquia l'administrador o superadministrador (segons el CMS), que té accés a tota la configuració del portal, així com la capacitat d'escriure, esborrar o canviar la plantilla de presentació.

Per la part més baixa es troba l'usuari subscriptor, al que no se li permet fer res més que estar enregistrat i comentar les publicacions que es donin.

A part ens podem trobar per enmig altres tipus d'usuaris que tinguin la funció d'encarregar-se de gestionar una determinada part del portal o que puguin escriure on la resta no poden o que puguin fer coses que no poden els que estiguin per sota d'ells.

2.3 Tipus de CMS

Segons la seva funcionalitat hi ha diferents tipus de CMS. Al mercat també podem trobar molts CMS, tant de codi obert, com de pagament.

D'aquesta manera podem classificar els següents tipus de CMS:

- **Fòrums:** els més estesos juntament amb els blogs. Permeten crear fòrums de discussió entre la gent. Alguns CMS d'aquest tipus són: *BBPress* o *BuddyPress*
- **Wikis:** estan destinats a la compartició de coneixements col·laborant tots els usuaris a la realització dels continguts. Aquí podem trobar: *MediaWiki* o *TikiWiki*.

- **Ensenyament:** es fan servir per l'ensenyament online. Els més coneguts són *Joomla* i *Moodle*.
- **Blogs:** els CMS més utilitzats per tothom. La funció es la de crear pàgines personals on el creador doni la seva visió sobre algun tema, o escrigui el que vulgui per compartir-lo amb els demés. Els CMS més estesos d'aquesta branca són: *WordPress* i *b2evolution.net*.

Per la feina que nosaltres volem fer ens centrarem en l'últim tipus de CMS, els de blogs, donat que volem que un determinat col·lectiu de gent doni la seva opinió sobre diversos temes determinats.

Donat que tot CMS es comporta com un portal web, això limita les tecnologies que fan servir. Així ens trobarem amb els següents llenguatges de programació.

- El fet de ser via web ens obliga a fer ús del llenguatge *HTML* que ens permet crear pàgines estàtiques. És la base del portal i el que dona forma a les plantilles de presentació. Associat a *HTML* el més comú és utilitzar fulls d'estils (*CSS*).
- Com que el funcionament òptim és el de primer carregar les dades i després dibuixar el contingut pràcticament tots els gestors de continguts fan servir *PHP*, un llenguatge de programació pre-pàgina, és a dir, executa les funcions i a la web es mostra el contingut estàtic del resultat de les funcions realitzades. Encara que hi ha alguns CMS que fan ús de *Python* o de *Perl* per programar.
- Contínuament tindrem que fer accessos a bases de dades, per tant tots els CMS també necessiten un sistema gestor de bases de dades que els permeti disposar de les dades ràpidament. Com que *PHP* és el llenguatge més usat per la programació web, el més comú es fer ús de *MySQL* per enregistrar les bases de dades. També alguns gestors fan ús de *PostgreSQL* o *LDAP*.

Així doncs treballarem amb els llenguatges *HTML*, *CSS*, *PHP* i *SQL* per aconseguir un portal dinàmic i eficient.

2.4 WordPress

2.4.1 Què és?

Com he esmentat anteriorment, l'eina escollida per desenvolupar el projecte finalment ha estat *WordPress*.

WordPress és un CMS de creació de blogs que es va crear al 2004 per Matt Mullenweg i s'ha anat expandint fins a la data d'avui, on ja conta amb més de 200 milions d'usuaris i és considerat el CMS de creació de blogs més utilitzat i millor valorat.

Està desenvolupat en *PHP* i *MySQL* i és de codi obert. Això implica que sigui relativament fàcil de modificar i d'ampliar.

D'aquesta manera, un dels punts forts d'aquest CMS és la possibilitat de personalitzar els temes (plantilles de presentació) i de crear nous plugins que facin una ampliació de les possibilitats que té per defecte. A més, tant els temes, com els plugins es poden compartir entre els usuaris, de manera que qualsevol pot tenir un blog totalment actualitzat amb més funcions de les que ofereix el *WordPress* original, sense tenir cap mena de nocions sobre programació.

2.4.2 Quins tipus hi ha

WordPress ofereix dues opcions per crear un blog.

WordPress sense instal·lació

La més simple es tracta d'accedir a la web de *WordPress* (<http://www.wordpress.com>) i crear un nou usuari. Per aquest usuari s'associarà una URL (una direcció web) com a subdomini de *WordPress*.

Avantatges:

- L'usuari no es té que preocupar per cercar un hosting on allotjar el blog.
- No fa falta cercar cap domini.
- El seu funcionament és molt senzill.

Inconvenients:

- No es poden instal·lar nous plugins, només es pot fer ús dels que venen per defecte.
- No podem instal·lar noves plantilles de presentació, amb lo qual limitem la personalització que tindrà el blog.
- El blog estarà sotmès a un control de la temàtica de la que es parli. En el cas que intenti tractar de temes que incitin a la violència o similars seran censurats i esborrats.

Aquesta opció és recomanable per als usuaris que només volen la comoditat de poder escriure ràpidament sense fer ús de plugins ni temes personalitzats.

WordPress amb instal·lació

És la opció més habitual per els que volen fer ús de totes les qualitats de *WordPress*. És la més complexa.

Avantatges:

- Es pot instal·lar qualsevol tema i qualsevol plugin.
- Si es necessita podem modificar el codi font de *WordPress*.
- Tenim accés directe a la base de dades.
- Podem fer multiblogging.

Inconvenients:

- Hem de tenir un servidor dedicat o contractar un hosting per l'allotjament de la web.
- Hi ha que contractar un domini.
- Pot haver-hi problemes de seguretat.

Fins a la versió 2.7, la opció de *WordPress* amb instal·lació oferia dues modalitats diferents, *WordPress* normal i *WordPress* Mu, que permetia fer el multiblogging. Però a partir de la versió 3.0 *WordPress* Mu va desaparèixer i les seves funcionalitats van passar a formar part del *WordPress* normal.

Per el desenvolupament del projecte, com es lògic, fem servir *WordPress* amb instal·lació.

2.4.3 Instal·lació de WordPress

Una altre de les característiques més importants de *WordPress* és la facilitat de la instal·lació. Un cop tenim el nostre hosting i el seu domini associat hem de seguir els següents passos.

- 1) Primer hem de descarregar-nos des de la web oficial (<http://www.wordpress.org>). Dintre de l'arxiu comprimit que descarreguem ens trobem la carpeta amb tot el contingut de *WordPress*, que la tindrem que pujar mitjançant FTP al hosting que tinguem.
- 2) Posteriorment hem d'accedir al servidor *MySQL* i crear una nova base de dades, que serà la que usarem per al nostre portal.
- 3) Un cop que ja tenim els arxius al servidor i la base de dades creada tenim dues formes de continuar la instal·lació:
 - a) Editar l'arxiu *wp-config-sample.php*. En aquest cas, hem d'introduir els valors corresponents al nom de la base de dades, nom usuari, contrasenya i servidor en el que es troba. Al finalitzar tenim que reanomenar l'arxiu a *wp-config.php*.
 - b) Accedir a la instal·lació via web. Haurem de fer el mateix, però en un entorn gràfic. No sempre funciona bé.
- 4) Finalment accedim mitjançant la web al nostre blog, ens demanarà introduir una sèrie de dades personals de l'espai i tindrem tot instal·lat i configurat.

Aquesta instal·lació, és la que *WordPress* anomena “*Instal·lació en 5 minuts*”, donat que, si no hi ha cap problema el temps que es triga en instal·lar i configurar no dura més que això.

WordPress

A continuación deberás introducir los detalles de conexión con tu base de datos. Si no estás seguro de cuales son contacta con tu proveedor de alojamiento.

Nombre de la base de datos	<input type="text" value="wordpress"/>	El nombre de la base de datos en la que quieres que se ejecute WP.
Nombre de usuario	<input type="text" value="username"/>	Tu nombre de usuario de MySQL
Contraseña	<input type="text" value="password"/>	...y la contraseña de MySQL.
Host de la base de datos	<input type="text" value="localhost"/>	Si no funciona localhost tendrás que contactar con tu proveedor de alojamiento para que te diga cual es.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

Enviar

¹ Instal·lació via web de *WordPress*

3. Començament. De blog a multiblogging

3.1 Punt de partida

Comencem a resoldre el problema partint amb *WordPress* instal·lat a un servidor i amb un tema aplicat escollit per el COEIC.

En aquest cas el tema es diu *BluePrint* i com aspecte a destacar disposa d'un visor (*slider*) que fa una presentació amb imatges dels articles que hi ha actualment al blog. A més, tenim un panell d'administració del tema que ens permet, de entre altres coses, posar el logo de l'empresa a la capçalera, definir si volem mostrar o no el *slider* i afegir *banners* de publicitat.

Per la resta, és un tema totalment normal, el qual modificarem per adoptar el disseny que volem.

Ara mateix tenim que analitzar si amb el que tenim a hores d'ara ens és útil o si ens fa falta més.

Amb la instal·lació per defecte de *WordPress* podem tindre un blog senzill i altament personalitzable, i això ens aniria perfecte sempre que volem fer un blog, però l'objectiu és tractar de fer una blogosfera i amb el que tenim ara tenim les següents mancances:

- Per poder crear les diverses comissions hauríem de fer-les com si fossin categories del mateix blog, amb el que tindríem tots els articles junts, a més de que podríem tenir problemes per crear les subcomissions.
- Totes les comissions tenen que ésser exactament iguals. Això no és el que ens convé, donat que hi ha funcionalitats que només hi seran activades per determinades comissions.
- No podem fer distinció d'usuaris. Això és el que més ens limita de tot.
 - Si no podem fer una distinció d'usuaris, no podem dir que aquest usuari pertany a aquesta comissió o a aquesta altre.
 - Tampoc podríem anomenar un administrador de la comissió. En el cas de que volem donar permisos especials a un usuari en concret (com podria ser el cas del president de la comissió) tindria permisos especials per tota la blogosfera, cosa que seria altament perillosa per motius de seguretat.
- La pàgina principal sempre estaria formada per els últims articles publicats. Si volem destacar un article o un altre, això no ens ho permet. La filosofia del blog és mostrar articles en ordre cronològic invers, és a dir, de més nou a més antic.
- El resultat, no seria una blogosfera, sinó un blog amb molts articles.

Com podem veure, tenim inconvenients que ens dificulten continuar desenvolupant el projecte. Per tant, la solució per resoldre els problemes, és la de convertir el blog en un multiblog o *multisite*.

3.2 Funcionament del taulell de *WordPress*

3.2.1 Usuaris de *WordPress*

El que tenim una vegada hem instal·lat *WordPress* (sense fer l'activació del *multisite*) és el que utilitzarem més tard, un cop tinguem *multisite* actiu, per gestionar cadascun del blogs, així doncs, ja es pot veure el funcionament principal que té.

Primer de tot hem de saber quins tipus d'usuaris tenim, donat que no tothom té el mateix taulell. *WordPress* té definida una estructura jeràrquica d'usuaris que permet donar rols concrets per els nostres usuaris, segons les necessitats de cadascun d'ells.

D'aquesta manera podem assignar els següents rols als usuaris, ordenats de menys privilegis a més.

- **Subscriber:** usuari que només pot modificar el seu perfil. Res més.
- **Col·laborador:** pot fer el mateix que el subscriber, i a més pot escriure articles i modificar-los, però

no pot publicar-los, han de ser revisats abans de la seva publicació.

- **Autor:** pot fer el mateix que el col·laborador i a més pot publicar els seus articles sense que cap persona tingui que revisar-lo.
- **Editor:** té els mateixos privilegis que l'autor i a més pot revisar i publicar articles que han estat escrits per els col·laboradors. També li està permès crear pàgines¹ noves.
- **Administrador:** té accés total a totes les opcions del blog. Entre d'altres, pot:
 - Modificar, publicar i esborrar articles dels altres.
 - Crear i publicar els seus propis articles i pàgines.
 - Decidir quins comentaris es publiquen.
 - Canviar el tema de presentació del blog.
 - Instal·lar, activar i desactivar plugins
 - Administrar els usuaris que tingui.

3.2.2 Menús

Els menús que incorpora *WordPress* al seu taulell, com es normal, varia en funció del tipus d'usuari que estigui connectat. D'aquesta manera el taulell que tingui un subscriptor serà molt limitat, mentre que el taulell de l'administrador serà molt extens.

Com que, com ja he explicat, ens trobem en una estructura jeràrquica on cada rol superior conté les funcionalitats dels inferiors, explicaré el taulell de l'administrador, donat que és el més complex dels que tindrem.

Els menús es divideixen en tres seccions diferents.

- Opcions del tauler
- Opcions sobre els articles
- Opcions de configuració

Tauler



2

El Tauler és el menú principal de la part d'administració de *WordPress*. Es compon per les opcions Tauler i Actualitzacions

Opció Tauler

A l'opció Tauler ens trobem amb un resum del que tenim al nostre blog. Es divideix en les següents seccions:

- **Ara mateix:** mostra el nombre d'entrades que s'han publicat i el relacionat amb aquestes (etiquetes, categories, pàgines, comentaris...).

¹ Una pàgina és un article que es publica a part i al qual no es pot fer comentaris

² Vista del menú tauler

- **Editor ràpid:** permet a l'usuari escriure un article de forma ràpida, encara que manca d'algunes opcions.
- **Comentaris recents:** mostra els darrers comentaris que hem obtingut a les nostres entrades. Si tenim privilegis suficients també podem fer accions amb els comentaris com rebutjar-lo, respondre'l, editar-lo, marcar-lo com spam o enviar-ho a la paperera.
- **Esborrans recents:** ens permet veure els articles que tenim a mig fer i accedir a ells per acabar-los.

Les restants seccions són informatives, comuniquen a l'usuari noves notícies de l'entorn *WordPress*.

The screenshot shows the WordPress dashboard interface. At the top left is the 'Tauler' (Dashboard) header. The main area is divided into several sections:

- Ara mateix (Right Now):** A summary box showing statistics: 5 Entrades (posts), 0 Pàgines (pages), 2 Categories, and 6 Etiquetes (tags). It also shows the current theme 'Blueprint WP' with 1 giny (plugin) and the WordPress version '3.0.1'.
- Editor ràpid (Quick Editor):** A form for quickly creating a new post, including fields for 'Títol' (title), 'Penja/insereix' (upload/insert media), 'Contingut' (content), and 'Etiquetes' (tags). Buttons for 'Desa l'esborrany' (save draft), 'Reinicialitza' (refresh), and 'Publica' (publish) are visible.
- Esborrans recents (Recent Drafts):** A section indicating 'No hi ha esborrans en aquest moment' (no drafts at the moment).
- Comentaris recents (Recent Comments):** A section indicating 'Encara no hi ha comentaris.' (no comments yet).
- WordPress Blog:** A section for the latest news from the WordPress blog.

Opció Actualitzacions

En el cas de que hi hagi noves versions de *WordPress*, dels temes que tinguem instal·lats o de les extensions que utilitzem, en donarà l'opció d'actualitzar-ho automàticament a través d'aquest menú.

Entrades



2

El menú *Entrades* és el que permet escriure nous articles, modificar els existents i fer operacions relacionades amb aquests.

Opció Entrades

Quan anem a l'opció entrades ens trobem amb tots els articles que han estat publicats dintre del blog, tant si els hem escrit nosaltres o si els ha escrit un altre usuari.

Segons el tipus d'usuari que siguem podrem fer més o menys opcions amb ells. El més comú és que l'usuari sigui del tipus *Autor* o *Col·laborador*, els quals poden escriure i modificar els seus propis articles. Així doncs, dintre d'aquest panell, podran accedir a qualsevol article però només podrà editar els seus.

A aquesta secció també podrem fer cerques filtrades per data o per categories. O directament cercar un article per el seu títol.

¹ Vista del tauler de *WordPress*

² Menú d'entrades

Opció Afegix

Aquí es on pròpiament s'escriuen els articles. El contingut el podem escriure tant ajudats per una petita barra d'eines que ens permet fer modificacions de disseny, com en mode HTML.

A més d'escriure l'article podem afegir-li diversa informació addicional:

Afegeix una nova entrada

Penja/insereix Visual HTML

Camí:
Paraules comptades: 0

Extracte

Envia retroenllaços

Camps personalitzats

Discussió

Autor

- **Categoria:** podem assignar una categoria existent a l'article o crear-li un de nova. En el cas que no seleccionem cap, automàticament quedarà assignat a *Sense Categoria*.
- **Extracte:** petit resum de l'article per utilitzar-lo en els RSS, entre d'altres.
- **Etiquetes:** paraules que descriuen l'article, per poder fer cerques sobre temes en concret.
- **Camps personalitzats:** en el cas que necessitem introduir informació extra, podem crear un camp personalitzat.

Opció Categories

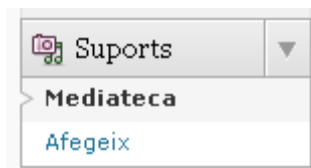
Les categories són les diverses seccions que podem afegir al nostre blog. D'aquesta forma podem agrupar els articles segons un tema concret.

Aquesta opció permet afegir noves categories, accedir a les existents i veure el nombre d'entrades que té cadascuna. També podem fer subcategories donant-li un pare a la nova categoria que creem.

Opció Etiquetes

Les etiquetes ens permeten marcar paraules clau d'un text. Dins d'aquest menú podrem crear-ne de noves i veure quines son les més usades.

Suports



1

Aquest menú ens permet afegir i gestionar els arxius multimèdia que volem introduir als articles.

A través de l'opció *Afegeix* podem pujar arxius de vídeo o de so. Tots els arxius que introduïm els podrem veure a l'opció *Mediateca*, i estaran disponibles per quan vulguem fer ús d'ells en algun article.

Enllaços

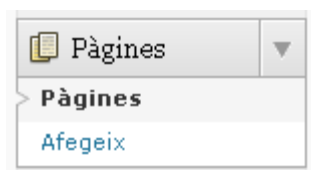


2

Aquesta secció permet veure els enllaços que es mostren a la pàgina principal del tauler.

Podem veure, afegir u organitzar els enllaços que volem.

Pàgines



3

Les pàgines, com ja he comentat anteriorment, són pàgines web amb contingut estàtic que ens permeten crear espais d'informació que no tenen que estar en constant actualització. Les podem utilitzar per fer una presentació de la web o per informar d'alguna notícia puntual, de la que no volem que es facin

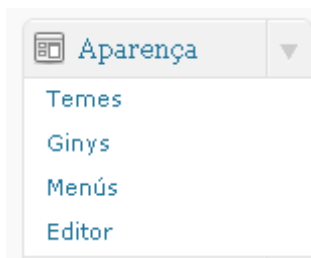
comentaris.

Aquesta secció ens permet crear i gestionar les pàgines que volem afegir al nostre blog.

Comentaris

A l'apartat de comentaris podem veure tots els comentaris que tenim al blog. Se'ns informa de l'article que correspon al comentari, la data i l'usuari que el va escriure i si tenim privilegis suficients podem esborrar-los o fer diverses opcions d'administració sobre ells.

Aparença



4

En aquesta secció podrem canviar completament el disseny visual del blog.

Opció Temes

Aquí ens trobem amb tots els temes que tenim instal·lat al nostre blog. Podem activar el que vulguem, veure com quedaria sense activar-lo (previsualitzar-lo) o esborrar-lo del sistema.

¹ Menú suports

² Menú enllaços

³ Menú pàgines

⁴ Menú d'aparença

Opció Ginys

Els ginys, també anomenats widgets, són aplicacions visuals molt petites que ens permeten incorporar al nostre disseny del blog noves funcionalitats.

La posició dels ginys dintre del blog, vindrà determinat per el tema que escollim. Els ginys tenen la peculiaritat que tenen que anar col·locats dintre d'un *sidebar*. Els *sidebars* són barres que posem dintre del tema per tal de poder afegir els menús o el ginys. Per tant, les opcions de col·locació dels ginys ens vendrà limitada per el nombre de *sidebars* que disposi el tema i per on es situïn aquests.

Els ginys més comuns i que es troben a la majoria dels blogs, són el giny del calendari, on mostra un calendari del mes amb els dies que s'han escrit articles remarcats, el giny meta, que permet accedir al taulell d'administració del blog, i el giny cerca, que permet cercar articles del blogs.

També podem incorporar ginys personalitzats amb el giny text, que ens permet posar el que vulguem en llenguatge HTML.

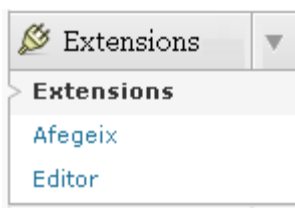
Opció Menús

Aquesta opció permet crear menús de navegació personalitzats, afegint categories o enllaços personalitzats.

Opció Editor

L'editor permet modificar els arxius font del temes. Ens mostra la llista d'arxius que componen el tema que estem utilitzant i ens permet accedir a ells i fer canvis. Aquesta opció no es molt aconsellable de fer-la servir, donat que ofereix una interfície simple i en la qual no podem fer gaire cosa.

Extensions



¹

Les extensions, o plugins, són petits programes que ens permeten fer noves funcionalitats.

No són el mateix que els ginys, mentre que els ginys es basen en funcions que afecten a la part visual del blog, les extensions ens serveixen per la part d'administració, per tant, només tindrà accés a elles i només les podrà instal·lar

un usuari amb privilegis d'administrador.

Les extensions no requereixen instal·lació, simplement cal descomprimir i copiar els arxius a la carpeta corresponent.

Aquesta serà una de les parts més importants del projecte, donat que les funcionalitats que hi hagi que afegir al blog, es farà a través d'extensions principalment.

Opció Extensions

Mostra totes les extensions que tenim al blog, tant les actives com les inactives

Opció Afegeix

Permet instal·lar noves extensions. Pregunta quin és l'arxiu que conté l'extensió (arxiu comprimit sempre) i la instal·la automàticament. Aquesta opció no és l'habitual per instal·lar extensions, en comptes d'això el més comú és copiar el directori d'arxius de l'extensió a la carpeta *wp-content/plugins*.

¹ Menú extensions

Opció Editor

Aquesta opció és exactament igual a l'opció editor que tenim a l'apartat d'*Aparença*. Permet editar els arxius font de les extensions.

Igual que passava amb l'opció editor dels temes, no és recomanable fer ús d'aquesta opció, donat que l'eina que tenim que utilitzar no és l'adient per programar.

Usuaris



1

El menú d'usuaris està disponible per tots els tipus d'usuari. A través de ell podem veure els usuaris que hi formen part del blog i, en cas de ser administrador, podem editar els perfils dels usuaris.

A través d'aquest menú un administrador pot donar o treure privilegis als diferents usuaris que tenim. Els altres tipus d'usuari poden accedir a l'opció *El vostre Perfil* per veure i modificar les seves dades personals.

Eines



2

WordPress ofereix dues eines, *Publica Això* i *Utilitzeu Això* per publicar directament el contingut d'un enllaç i per convertir etiquetes en categories respectivament.

A més dona l'opció d'importar i exportar els articles des de i a diferents formats, com són des de RSS (el més comú), des de altres blogs de *WordPress* o des

d'altres aplicacions.

Opcions



3

El menú d'opcions, és el que conté les opcions generals de configuració del blog.

Opció General

Conté opcions bàsiques del blog. Entre les coses que podem modificar a aquest menú ens trobem amb el títol del blog, el format de la data o el llenguatge en que volem que estigui la part d'administració.

Opció Escriptura

Aquí trobem les opcions pròpies que configuren l'escriptura dels articles. Podem configurar la categoria per defecte on volem que vagin els articles (per defecte *Sense Categoria*) o si volem que automàticament converteixi les emoticones en imatges.

Opció Lectura

Permet configurar opcions per quan anem a llegir entrades del blog.

¹ Menú d'usuaris

² Menú d'eines

³ Menú d'opcions

Opció Discussió

En aquest apartat podem configurar els comentaris dels articles. Es pot configurar de mode que es puguin fer arbres de comentaris o tots normals, es pot fer que no es puguin escriure comentaris, o fer un recull de paraules que permetin identificar un comentari automàticament com inadequat, per el cas que hi hagi paraules pujades de to, per exemple.

Opció Suport

Configura la mida de les imatges i la opció d'incrustació de les URLs dins de l'article.

Opció Privacitat

Permet variar la configuració de privacitat per tal de que el lloc web sigui accessible per part dels cercadors.

Opció Enllaços Permanents

Configura el mode en el que es generen els enllaços permanents, és a dir, les URL que dona accés a cadascú dels articles.

Aquesta opció és necessari tenir-la bé configurada per tal de millorar el posicionament SEO.

Els enllaços permanents els podem configurar donant com a nom de l'enllaç:

- L'identificador de l'article (forma predeterminada).
- El dia i el nom de l'article.
- El mes i el nom de l'article.

També es pot donar una forma personalitzada, que és el més aconsellable.

En el cas que tinguem extensions instal·lades, les opcions de configuració d'aquestes també acostumen a anar al menú d'opcions.

3.3 Conversió a *multisite*

3.3.1 Què és el *multisite* i per què serveix?

El *multisite* consisteix en convertir la instal·lació que tenim d'un únic blog, en una xarxa de blogs, en la que podem tenir tants blogs com vulguem.

WordPress permet aquesta conversió a partir de la versió 3.0, abans només era permès a la distribució *WordPress Mu*, com ja he comentat anteriorment.

Al convertir el nostre lloc un *multisite* (també dit multiblog o multilloc) podrem arreglar tots els problemes que havíem esmentat a l'apartat anterior.

Fer el *multisite* serveix per tenir diversos blogs, tots administrables des del mateix lloc, però cadascú independent dels altres. Aquest fet ens permetrà resoldre els problemes d'abans de la següent manera:

- Cada comissió ja no farà falta que sigui una categoria, sinó que podrà ser un blog independent. Això ens permetrà fer un altre ús de les categories, podrem dividir els articles de cada comissió segons la subcomissió a la que pertany (en el cas que hi hagin subcomissió) o podrem dividir-los també segons el tipus d'article que es tracti (si és articles de vídeo, de so o un article d'un llibre, per exemple).
- Com que cada blog és independent, podem activar les funcionalitats que vulguem als blogs que necessitem. De la mateixa manera que, en el cas que ens resulti necessari, podem tenir un tema

específic la comissió en concret.

- Ja podem fer la distinció d'usuaris. Encara que tots els usuaris formen part d'un únic conjunt, els podem destinar a les comissions que siguin necessàries. També podem donar privilegis determinats per comissions determinades.
- Per la pàgina principal avancem en el problema, donat que no mostraria els últims articles de tots els blogs, sinó que mostraria els últims de un blog en concret, el blog principal en aquest cas. Encara no és el que volem, així que el problema persisteix i el tindrem que solucionar d'una altra manera.
- Amb això ja comencem a tenir una aproximació a la definició de blogosfera.

El *multisite* de *WordPress* també presenta inconvenients.

Com que cada blog és independent dels altres, això fa que tinguem problemes de comunicació entre blogs, fa que sigui difícil accedir i manipular articles d'un blog a un altre.

Internament *WordPress* crea un nou conjunt de taules per cada blog que es crea. Aquest fet presenta una part positiva i una altra negativa.

- Com positiu, guanyem en velocitat d'accés a les dades. En cas de que tot estigués junt en un conjunt de taules faria que fos més lent a l'hora d'accedir a una dada en concret. Així tenim una indexació major, de forma que podem anar als articles de cada blog més ràpidament.
- En contraposició, augmenta l'espai utilitzat al servidor. Per cada blog que introduïm tenim un grapat de noves taules, amb l'espai que aquestes ocupen. Si tenim molts blogs, acabem amb una base de dades de dimensions considerables.
- També com a part negativa, tenim que cada cop que creem un nou blog es generen una sèrie de dades que corresponen a opcions predeterminades i característiques del blog. En molts casos moltes d'aquestes dades són iguals a tots els blogs, amb el que tenim dades duplicades, encara que a diferents taules. Aquest fet provoca ineficiència.

Intentar arreglar aquests inconvenients, provocaria desmuntar el sistema intern que ofereix *WordPress*, donat que hauríem de fer canvis en les estructures de taules. Aquest fet faria que en el cas que en un futur fem una actualització de *WordPress*, ens deixi de funcionar tot el que hem fet.

Per tant, hem d'adaptar-nos al que ens ofereix *WordPress* i aprofitar al màxim els avantatges per tal de minimitzar els inconvenients.

3.3.2 Estructura interna de *WordPress*

Fins ara, he explicat què és i què podem fer amb *WordPress*, i també què ens comporta convertir una instal·lació normal en un *multisite*.

Així doncs, ara mostraré que tenim, donat que una de les coses més importants per poder desenvolupar i comprendre cadascun dels mòduls que he de tractar, és conèixer a fons com funciona l'eina que fem servir.

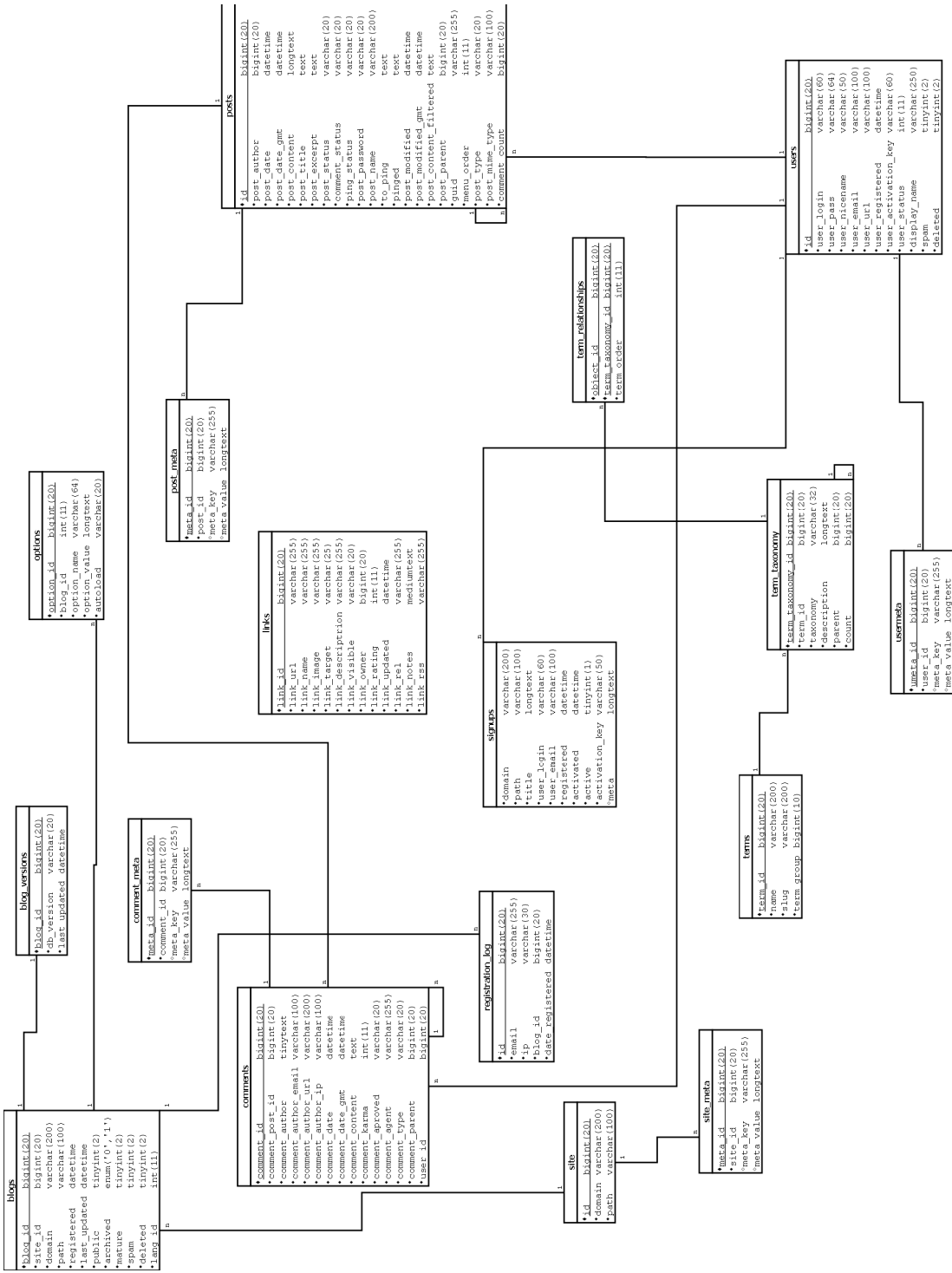
Les taules de les que disposem són les següents¹:

- **blogs:** guarda el id i la ruta en la que es troben els diferents blogs creats al fer el multilloc. Aquesta taula és única a la base de dades.
- **blog_versions:** relacionada amb la taula blogs, enregistra les diferents versions de cadascun dels blogs en cas de que n'hi hagués més d'una. També és única.
- **comments:** guarda tots els comentaris que s'han introduït en un blog. Es relaciona amb la taula de posts, amb la d'usuaris i amb si mateixa. Hi ha tantes taules comments com blogs.
 - Un comentari només pot ser escrit per un usuari i un usuari pot escriure molts comentaris.

¹ El nom de la taula realment serà `prefix_nomtaula`

- Un comentari és únic d'un article, però a un article hi ha de molts comentaris.
- Un comentari pot tenir diversos fills (comentaris del comentari) però només serà fill d'un únic comentari.
- **comment_meta:** enregistra informació clau dels comentaris.
- **links:** guarda enllaços d'interès que es mostren a la pàgina d'administració de *WordPress*. No té res a veure amb cap altre taula. Es repeteix a tots els blogs.
- **options:** emmagatzema les diferents opcions del blog, com el nom, la descripció, l'administrador d'aquest... També es repeteix a tots els blogs. Aquesta taula és un exemple d'ineficiència donat que podria ser única, donat que conté un camp *blog_id* que pot servir per identificar el blog corresponent, però no es fa cap ús d'ell.
- **posts:** aquesta és la taula més important. Guarda tots els articles del blog, tant els publicats, com els esborranys, com els que estan pendents de publicar-se i els esborrats. Es relaciona amb les taules *comments*, *postmeta*, *users* i amb si mateixa. Hi ha tantes taules *posts* com blogs.
 - Un post conté molta informació *postmeta*, i una informació *postmeta* és única d'un post.
 - El post l'escriu un usuari en concret i aquest usuari pot escriure tant pots com vulgui.
- **postmeta:** guarda informació dels diferents articles. Enregistra dades com la ruta de la imatge si en té alguna associada, les etiquetes que identifiquen el post o la categoria a la que pertany.
- **registration_log:** enregistra l'usuari administrador quan es crea el lloc.
- **signups:** usuaris que s'han registrat personalment. Aquesta taula és única
- **site:** conté els llocs creats, el seu domini principal i la ruta per accedir a ells. Normalment hi haurà només un únic *site*. Aquesta taula és única.
- **site_meta:** a aquesta taula està emmagatzemada tota la informació del lloc, com el seu nom, el correu de l'administrador i demés opcions. La taula també és única. Està relacionada amb la taula *site*.
 - Un *site* pot contenir molts *site_meta*, i un *site_meta* determinat correspon a un únic *site*.
- **terms:** enregistra les categories i les etiquetes del blog corresponent. Hi ha tantes taules *terms* com blogs. Es relaciona amb *term_taxonomy*.
 - A cada terme de la taula *terms* li correspon un únic *term_taxonomy* i un *term_taxonomy* pot ser de diferents termes.
- **term_relationships:** conté els posts als que estan associats cadascun dels termes de la taula *terms*.
- **term_taxonomy:** descriu el tipus de terme (categoria, enllaç o etiqueta) que correspon a cadascun del registres de la taula *terms*.
- **users:** conté tots els usuaris registrats a la blogosfera. Aquesta taula és única. Es relaciona amb les taules *posts*, *comments*, *usermeta* i *signups*.
- **usermeta:** enregistra informació personal relacionada amb els usuaris.

D'aquesta manera, el model conceptual que tenim amb *WordPress* amb l'opció *multisite* activada és el següent.



1 Model Conceptual de la base de datos

3.3.3 Activació del *multisite*

Al fer la instal·lació de *WordPress* teníem dues alternatives, fer-la de la forma més visual i modificant els arxius de configuració directament. Ara, per fer l'activació del *multisite* només en tenim una opció, que ve a ser una barreja de les dues opcions anteriors, en la que, per una part, tindrem que modificar els arxius de configuració i per altre, tindrem que seleccionar opcions d'un mode visual.

Primer de tot, hem de modificar l'arxiu de configuració de *WordPress* (*wp-config.php*) i afegir al final de l'arxiu la següent sentència:

```
define ('WP_ALLOW_MULTISITE', true);
```

Amb això li estem dient que volem fer que la nostra instal·lació de *WordPress* es converteixi en un multilloc, és a dir, li indiquem que activi aquesta opció per configurar-la.

Fet això, ens trobem amb un nou menú al tauler d'administració, dintre de *Eines*, tenim l'opció *Xarxa*



Aquesta opció ens explica el que tenim que fer per continuar amb l'activació del multilloc.

Primer haurem d'escollir el mode de creació del domini per accedir als blogs. Pot ser en subdominis o subdirectoris.

- **Subdominis.** La creació amb subdominis fa que, per cada blog nou que creem, per accedir a ell posem primer el nom del blog, seguit per la direcció principal de la web. Si accedim a la nostre web per la ruta *http://blogs.cat* i creem un nou blog que es diu *Esports*, llavors accedirem al blog *Esports* de la següent forma: *http://esports.blogs.cat*.

Aquesta opció no sempre es permet per motius evidents, si es dona el cas que la direcció de la nostra web ja està formada per un subdomini (com és el nostre cas) no podem fer un subdomini d'aquest. En aquests tipus de casos estem obligats a fer una estructura en subdirectoris.

- **Subdirectoris.** L'opció de crear l'estructura de blogs en subdirectoris sempre està disponible i en molts cassos es el més recomanable. Es diferencia de l'altre opció simplement en que, en comptes de posar el nom del nou blog al principi, el posarem al darrere, després de la URL del blog principal. D'aquesta manera, tot seguint l'exemple anterior tindrem que la ruta per accedir al blog *Esports* serà la següent: *http://blogs.cat/esports*.

Aquesta és l'opció que farem servir en el nostre cas, donat que la direcció del nostre blog principal és *http://blogs.eic.cat* i com he esmentat abans, com ja tenim subdominis no podem activar aquesta opció. A més, per el nostre cas, ens és molt més coherent fer una estructura de subdirectoris, donat que volem destacar que la nostra web és la actual i cadascun dels blogs que formaran el lloc estan directament relacionades amb la web principal, no són mons apart.

¹ Aspecte del nou menú de Eines

Un cop hem escollit la forma més adient per crear l'estructura que tindrà el nostre multilloc, hem de crear un nou directori dintre de la carpeta *wp-content* amb el nom *blogs.dir* que ens servirà per emmagatzemar tots els arxius que s'adjuntin (tant fotos, com vídeos o so) dintre dels articles nous. Aquests arxius es guardaran dintre de subcarpetes que formaran una jerarquia per accedir a elles primer amb un identificador del blog corresponent i després amb ordre cronològic, és a dir, primer es crearà un directori dintre de *blogs.dir* amb el id del blog on s'ha guardat l'arxiu, dintre d'aquest un altre amb l'any en que s'ha pujat i finalment un altre amb el mes corresponent. D'aquesta manera ens és senzill accedir a ells sense problemes.

Tot seguit, tal com ens indica *WordPress*, hem d'afegir les sentències següents al final de l'arxiu *wp-config.php*.

```
define( 'MULTISITE', true );
define( 'SUBDOMAIN_INSTALL', false );
$base = '/';
define( 'DOMAIN_CURRENT_SITE', '88.87.193.124' );
define( 'PATH_CURRENT_SITE', '/' );
define( 'SITE_ID_CURRENT_SITE', 1 );
define( 'BLOG_ID_CURRENT_SITE', 1 );
```

Amb això indiquem:

- Hem activat l'opció *MULTISITE*.
- No fem servir subdominis (en el cas que haguem seleccionat l'estructura de subdirectoris).
- La ruta a tot el sistema d'arxius de *WordPress* comença a partir de la /.
- El domini principal (el del primer blog) té aquesta direcció.
- La direcció al lloc inicial (i únic en el nostre cas) comença a partir de la /.
- L'identificador del lloc es 1.
- L'identificador del blog principal es 1 (podríem variar aquest valor, un cop tenim definits altres blogs).

Per últim afegim a l'arxiu *.htaccess*, les regles que ens donen.

Amb el fitxer *.htaccess* el que fem és definir directives de configuració de *Apache* per la instal·lació de *WordPress* perquè tingui un correcte funcionament.

Normalment aquest arxiu el trobarem a l'arrel de la instal·lació de *WordPress*, però podria ser que no estigués (cas en el que me trobat en la instal·lació en el servidor). En aquest cas hi ha que modificar l'arxiu *.htaccess* de configuració del directori on tenim instal·lat *Apache* o crear-lo com a nou dintre del directori arrel de *WordPress*.

```
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]

# uploaded files
RewriteRule ^([_0-9a-zA-Z-]+/)?files/(.+) wp-includes/ms-files.php?file=$2 [L]

# add a trailing slash to /wp-admin
RewriteRule ^([_0-9a-zA-Z-]+/)?wp-admin$ $1wp-admin/ [R=301,L]
```

```

RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^([_0-9a-zA-Z]+)/?(wp-(content/admin/includes).*) $2 [L]
RewriteRule ^([_0-9a-zA-Z]+)/?(.*\.php)$ $2 [L]
RewriteRule . index.php [L]

```

Amb això el que estem fent es activar el mode RewriteEngine d'Apache per tal de poder fer les redireccions corresponent de URLs. Un cop activat creem diverses regles d'escriptura per poder fer que WordPress faci correctament algunes operacions com són la pujada d'arxius o la redirecció correcta per als temes, plugins i demés informació a la qual no accedim mitjançant una URL concreta.

Un cop hem afegit tot només cal finalitzar sessió i tornar-la a iniciar perquè quedi totalment activada l'opció *multisite*.

Possibles problemes amb els que ens podem trobar

Tot i que l'activació del *multisite* és simple, és molt probable que ens trobem amb algun problema que pot aparèixer per problemes de configuració d'Apache o de MySQL.

Un dels cassos amb els que ens podem trobar, és el cas abans esmentat, en el que no tenim l'arxiu *.htaccess* dintre de la carpeta d'instal·lació de WordPress. En aquest cas, la solució és o bé modificar les regles dintre de l'arxiu de configuració d'Apache, tal com he comentat abans, o bé, crear l'arxiu *.htaccess* dintre del directori d'instal·lació de WordPress, tal i com hi hauria d'anar.

Altre problema comú és el que ens trobem un cop hem finalitzat l'activació del *multisite* sense cap problema, però a l'hora d'accedir a algun dels llocs nous que creem, ens surti una pàgina d'error.

Aquest problema és degut a que a la configuració d'Apache, no tenim activat el mode *mod_rewrite*. Aquest mode transforma les URLs que teòricament no existeixen, en URLs accessibles, això és, si volem entrar a la comissió d'Esports hauríem d'accedir, segons la norma d'accés per l'activació del *multisite* en subdirectoris, segons la ruta <http://blogs.eic.cat/esports>. Aquesta direcció, realment no existeix, no tenim típicament un arxiu *index* dintre d'una carpeta anomenada *esports*, sinó que té que accedir a l'arxiu *index* del tema aplicat per tal de pintar el contingut.

Per solucionar aquest problema, el que tenim que fer és canviar la configuració d'Apache. Hi ha que accedir a l'arxiu de configuració (arxiu *httpd.conf*) i modificar un parell de línies:

- Descomentem la línia següent, eliminant el # del principi. D'aquesta manera activarem el mode *mod_rewrite*.

```
# LoadModule rewrite_module modules/mod_rewrite.so
```

- Modifiquem el codi següent per tal de permetre les redireccions.

<Directory />		<Directory />
Options FollowSymLinks		Options FollowSymLinks
AllowOverride none		AllowOverride All
Order deny,allow		Order deny,allow

Deny from all	Allow from all
Satisfy all	Satisfy all
</Directory>	</Directory>

Amb això queden actius els serveis que necessita *WordPress* per fer les redireccions dels enllaços correctament.

3.4 El *multisite*. Què tenim de nou?

Ara que ja tenim activada l'opció *multisite* anem a veure què és el que tenim de nou respecte al que ja teníem anteriorment.

Noves taules

Com ja he comentat anteriorment quan vaig explicar l'estructura de la base de dades de *WordPress*, quan activem l'opció del *multisite* ens trobem amb que hi ha taules que es dupliquen tants cops com blogs tinguem i es creen d'altres noves que abans no i eren.

D'aquesta manera ens trobem que totes les taules que contenen informació sobre els articles, tals com *posts* i *comments*, i les taules relacionades amb aquestes, les tindrem més d'un cop, però amb el prefix variat, on l'identificador del blog acompanyarà al prefix al nom de la taula.

D'altra banda tenim taules que fins ara no eren necessàries i per tant no hi eren quan teníem la instal·lació normal. Aquestes taules són les següents:

- **blogs:** la necessitem per enregistrar cadascun dels nous blogs que creem. En el nostre cas, cadascuna de les comissions que tenim.
- **blogs_versions:** relacionada amb *blogs*.
- **registration_log:** relacionada amb *blogs* també. Serveix per saber la direcció de contacte de l'administrador del *blog*, i la data en que va se creat aquest *blog*.
- **signups:** usuaris registrats per web.
- **site:** informació de tot el lloc en general.
- **sitemeta :** relacionada amb *site*.
- **sitecategories:** relacionada també amb *site*.
- **users:** aquesta taula ja hi era en la instal·lació normal, però es veu alterada a l'opció *multisite*. Com que guardem tots els usuaris que estan registrats a la web, necessitem poder distingir-los per veure a quina comissió pertanyen en concret. Per això tindrem dos atributs més, que faran referència a la relació amb la taula *blogs*.
- **usermeta:** igual que amb la taula *users*, és refeta per poder guardar més dades.

Amb tot això quan creem un nou blog, per exemple la comissió *Esports*, farà el següent:

- La taula *blogs* assignarà un nou id la comissió d'*Esports*. Per exemple: id = 2.
- La taula *registration_log* salvarà la direcció de correu de l'administrador de la comissió, junt amb la data en que s'ha creat la comissió.
- Si l'administrador de la comissió és un usuari que encara no hi era a la base de dades les taules *users* i *usersmeta* enregistraran les seves dades.
- Per últim es crearà l'estructura de taules (*posts*, *comments*...) amb el prefix corresponent. Seguint l'exemple, si el prefix de les taules es *wp_*, el prefix de les taules que pertanyin a la comissió d'*Esports*, portaran el prefix *wp_2_*.

Nous menús

Amb l'activació del *multisite* i la posterior creació de nous blogs es crea un nou menú que només serà visible per un nou tipus d'usuari, el superadministrador.

El superadministrador es troba a l'esglaó més alt de la jerarquia d'usuaris. Mentre que els administradors tenen tots els privilegis dintre del seu blog, el superadministrador té tots els privilegis en tots els blogs que es creïn.

Així doncs, aquest tipus d'usuari pot accedir i modificar tot el que vulgui del lloc. Com es lògic, tenint en compte tota la responsabilitat que comporta el fet d'ésser superadministrador, només hi haurà un usuari que tingui aquest càrrec i es dedicarà bàsicament al manteniment correcte del lloc.



El menú de superadministrador té l'aspecte que es mostra a la imatge de dalt i ofereix les següents opcions:

- **Administrador:** permet crear i cercar blogs i usuaris.
- **Llocs:** mostra una llista de tots els blogs dels que disposem. A més ofereix diferents opcions sobre els blogs.
 - Crear un blog nou.
 - Esborrar un existent.
 - Editar el blog. Permet modificar opcions. Aquestes opcions es divideixen entre les que es són contingudes a la taula *blogs* (data de creació, URL, ...), les opcions pròpies del blog, que s'enregistren a la taula *options* del blog i altres com són la creació d'usuaris del blog.
 - Anar al panell de control intern del blog. El panell de control és el tauler propi de cada blog.
 - Desactivar. Posa el blog en suspens, a punt per esborrar, no es pot accedir a ell.
 - Arxivar. Guarda el blog sencer i no el mostra. Semblant a Desactivar, però en aquest cas sí que es pot anar al panell intern.
- **Usuaris:** mostra una llista de tots els usuaris que hi ha al lloc web. Per cada usuari podem veure el seu identificador, el nom d'usuari, el nom real, el rol, l'adreça de correu electrònic, la data en la que es va registrar i els blogs als quals pertany. A més podem editar o esborrar usuaris (excepte el superadministrador, que només podem editar-lo) i crear-ne de nous.
- **Temes:** ens mostra tots els temes que tenim instal·lats i ens permet deixar-los activats o desactivats perquè en puguin fer ús els administradors dels diferents blogs. En el cas que es desactivi un tema, aquest no serà visible (i per tant no es podrà utilitzar) per cap blog.
- **Opcions:** ens mostra diferents paràmetres que podem modificar referents a tota la xarxa de blogs que

¹ Menú de superadministrador

tenim. Entre d'altres trobem les opcions per canviar el nom del lloc web, el correu de benvinguda que reben els nous usuaris, els tipus de fitxers que poden penjar els usuaris o el idioma en que volem tenir configurat el taulell.

- **Actualitza:** permet fer una actualització de tota la xarxa de blogs automàticament, per si hem fet canvis en opcions generals.

4. Portal inicial

4.1 Què tenim fins ara

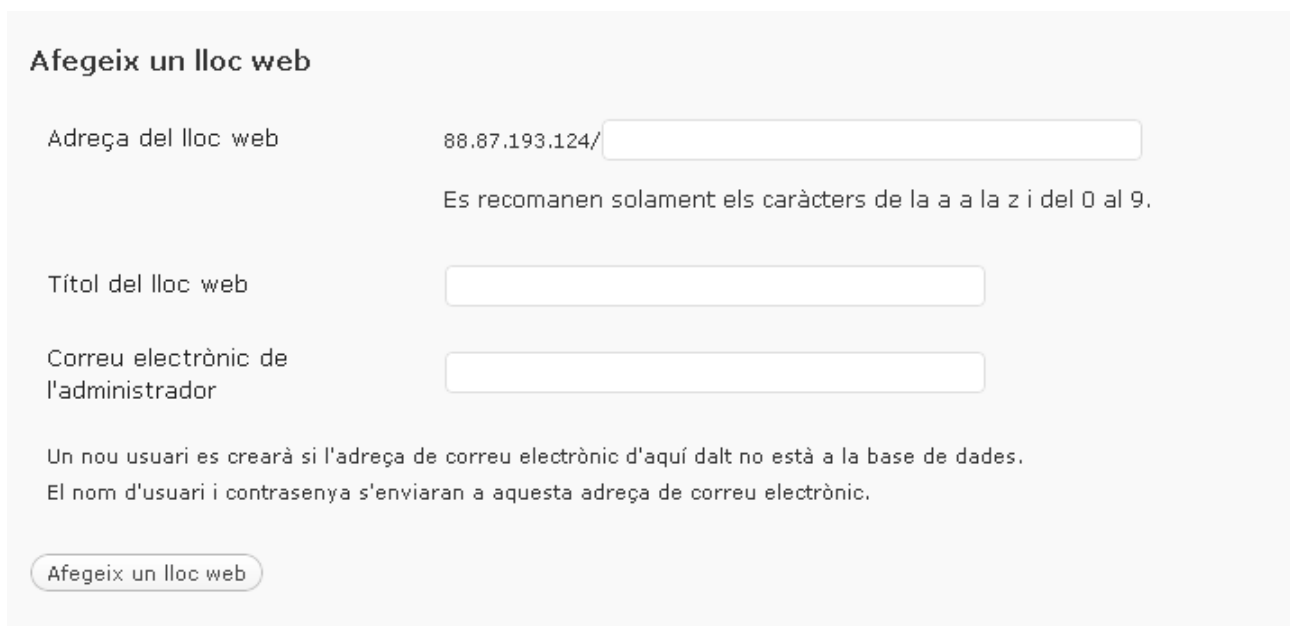
Un cop que hem finalitzat la instal·lació de *WordPress* i hem activat l'opció del *multisite*, el primer que hem de realitzar és començar a adaptar-lo a les nostres necessitats.

Com ja he explicat, els articles que volem guardar són escrits per enginyers pertinents a comissions. Cada comissió no és més que un departament dedicat a un tema en concret, on un grup d'enginyers qualificats i experts en l'àrea que es tracti, debaten e informen sobre notícies d'actualitat de la seva comissió i proposen millores per tot el col·lectiu.

A hores d'ara disposem d'un únic blog amb l'opció de *multisite* activa i amb el tema *BluePrint* aplicat. Llavors, seguint el que deia anteriorment de que cadascuna de les comissions és totalment independent de les altres, decidim aprofitar el *multisite* i crear un blog per cadascuna de les comissions de les que vulguem tenir el seu blog i així tractar cada comissió com un blog diferent.

Per crear les comissions, fem ús de l'explicat a l'apartat anterior i fem que un superadministrador s'encarregui, a través de l'opció *Administrador* del menú *Superadministrador* del seu taulell, de fer la creació de cadascun dels blogs corresponents.

Com inicialment no tenim introduït els usuaris, per cada nou blog es crearà un nou usuari que faci d'administrador d'aquest, tot indicant-li només el correu que tindrà. També podem fer que l'únic usuari que tenim, el superadministrador s'encarregui de ser l'administrador també dels blogs, per fer això hauríem d'indicar el correu del superadministrador allà on ens ho demana.



The screenshot shows the 'Add New Site' form in WordPress. The title is 'Afegeix un lloc web'. There are three input fields: 'Adreça del lloc web' (with the value '88.87.193.124/' and a note 'Es recomanen solament els caràcters de la a a la z i del 0 al 9.'), 'Títol del lloc web', and 'Correu electrònic de l'administrador'. Below the fields, there is a note: 'Un nou usuari es crearà si l'adreça de correu electrònic d'aquí dalt no està a la base de dades. El nom d'usuari i contrasenya s'enviaran a aquesta adreça de correu electrònic.' At the bottom, there is a button labeled 'Afegeix un lloc web'.

1

4.2 Cóm mostrem el que tenim

Ara, ja tenim creats tots els blogs que necessitem. Com que hem realitzat una activació del *multisite* en subdirectoris, per accedir a ells haurem d'anar mitjançant la ruta: http://domini/nom_blog. Com que el nostre domini és *blogs.eic.cat*, si volem accedir a la comissió d'*Esports*, escriurem <http://blogs.eic.cat/esports>.

Naturalment aquesta no és una manera útil per accedir als nostres blogs, donat que és una forma feixuga i, més important encara, si no tenim cap altre manera d'accedir a cadascun dels blogs, no estarem responent

¹ Menú per crear un nou blog

amb el que volem que sigui una blogosfera, és a dir, no estariem integrant tot el conjunt de blogs, sinó que estarien fent blogs independents amb domini comú.

Per tant, per oferir una manera fàcil i còmode per accedir a cadascuna de les comissions integrarem un menú a una barra lateral, que ens mostri i ens permeti anar a on vulguem.

Per configurar aquest menú el podríem fer de dues maneres, mitjançant els ginys o afegint-lo directament a la pàgina principal del tema.

Menú mitjançant ginys

En el cas que volguéssim integrar-lo amb els ginys, hauríem de fer-ho de la següent manera:

- Primer de tot hauria de fer-ho un administrador (o superadministrador)
- Hauria d'anar al menú d'*Aparença*, l'opció ginys.
- Això ens limitaria la col·locació, donat que podríem ficar-ho en alguna de les barres que tinguem. Recordem, que les barres laterals venen definides per el tema, no s'encarrega *WordPress* de gestionar quantes barres tindrem.
- Un cop decidim la barra a la que volem posar el menú, agafaríem el giny *Text* i el ficaríem dins.
- Per últim quedaria omplir el giny de text, amb una llista amb totes les comissions.

El problema de fer això, és que tenim que repetir el procés per tots els blogs que tenim. Per tant és una feina feixuga, repetitiva i poc eficient. A més, en cas d'alguna variació d'algun nom de blog o afegit d'un de nou, hauríem de modificar molts cops el mateix.

Menú mitjançant el tema

Si volem afegir el menú mitjançant el tema que tinguem seleccionat, primer de tot hem de veure com funcionen els temes en *WordPress*.

WordPress té una estructura pròpia per crear els temes. Mitjançant una sèrie d'arxius, controlem cadascun dels espais de la pàgina i cadascuna de les pàgines a on es té que dirigir.

Les pàgines dels temes estan creades mitjançant HTML amb estils CSS, juntament amb una mica de programació PHP, en la que normalment, es faran servir funcions que *WordPress* té predefinides i que ajuden a fer diverses accions. Totes les funcions que té predefinides *WordPress*, les podem trobar al Codex.

Un tema, consta de les següents pàgines:

- **style.css:** aquest és el full d'estils. Està programat en llenguatge CSS. Serveix per donar-li color i forma a les etiquetes HTML.
- **index.php:** correspon a la pàgina principal del blog. És la primera que es carrega.
- **commnets.php:** plantilla per la part dels comentaris dels articles.
- **single.php:** plantilla que correspon a la lectura de cadascuna de les entrades.
- **page.php:** especifica com serà el disseny de les pàgines estàtiques que creem.
- **category.php:** per cadascuna de les categories.
- **author.php:** necessària per configurar una fitxa personalitzada de l'autor.
- **search.php:** pàgina per mostrar els resultats d'una cerca.
- **404.php:** plantilla per el disseny de la pàgina d'error.
- **footer.php:** defineix el contingut del peu de pàgina.
- **header.php:** especifica com serà el disseny de la capçalera de la pàgina.
- **sidebar.php:** permet definir les barres laterals que tindrem i el seu contingut per defecte. La posició

dels diferents sidebars les definim a través dels fulls d'estils.

Com que el que volem afegir té que anar a les barres laterals, haurem de modificar l'arxiu *sidebar.php*. Quan accedim a aquest arxiu dintre del nostre tema obtenim el següent:

```
<!-- LEFT SIDEBAR =====>-->
<div id="primary" class="widget-area">
  <?php if (!function_exists('dynamic_sidebar') || !dynamic_sidebar("Left
SideBar")) : ?>

    <?php endif; ?>

  </div>
<!--//END OF LEFT SIDEBAR =====>-->
```

Amb això indiquem:

- El codi `<div></div>` ens serveix per encapsular el contingut dintre d'un marc invisible. Els atributs `id` i `class` són els que defineixen la posició i tot el disseny del contingut. Aquests atributs estan definits dintre del full d'estils.
- El condicional `if` ens especifica el cas per defecte, és a dir, el que mostrarà en el cas que no afegim cap giny a la barra lateral a través del taulell d'administració de *WordPress*. En aquest cas no mostraria res.

Com que *WordPress* no ofereix una solució per crear menús amb multiblogs, hem d'idear un menú personalitzat.

Tenint en compte les necessitats que tenim referent a que podem tenir subcomissions (les quals seran també blogs independents) i amb la previsió d'oferir una opció atractiva i simple a l'usuari administrador que gestioni el contingut, he decidit implementar un petit plugin que, juntament amb el tema, generi una estructura jeràrquica per crear el menú de les comissions i subcomissions.

Llavors, finalment, l'opció escollida per la creació del menú, és una barreja de plugin i modificació del tema principal.

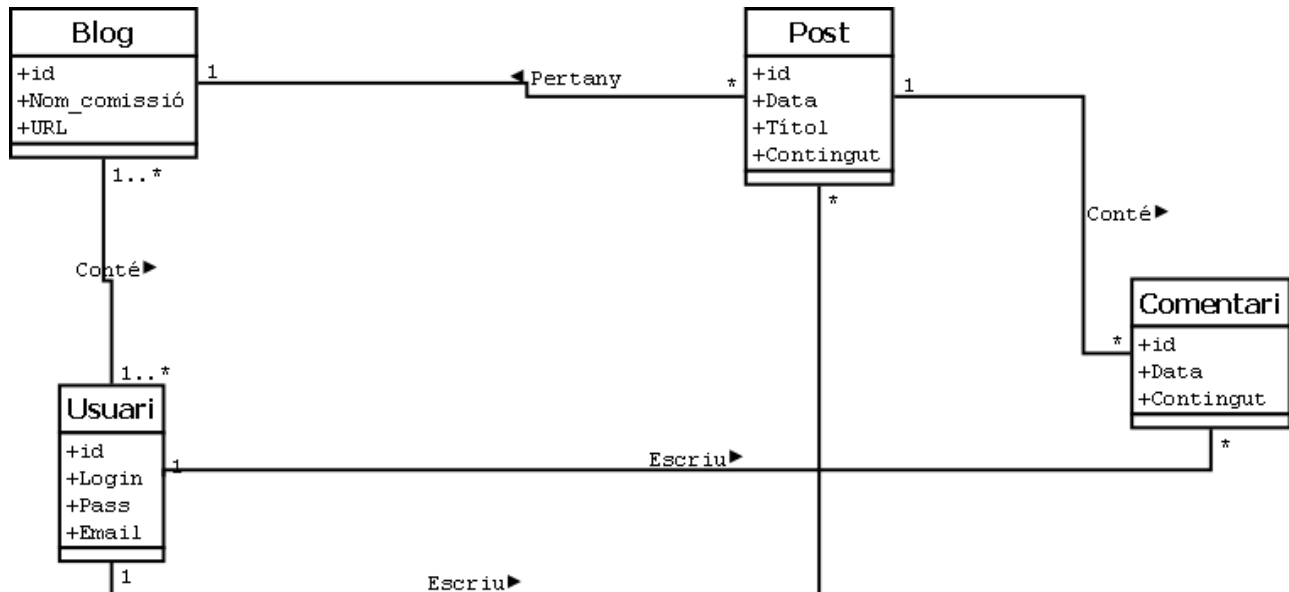
5. Model conceptual del sistema

Abans de començar a fer cadascun dels afegits cal fer l'estudi del problema actual.

Com que hem començat amb un problema ja començat, amb una eina que ja té una estructura pròpia i ben definida, el model conceptual inicial serà una adaptació a la realitat d'aquest model de partida.

A través d'aquest model inicial, a mesura que es vagin implementant noves funcionalitats el model conceptual inicial anirà expandint-se per tal d'anar-se adaptant als nous afegits.

Així doncs, el nostre model inicial es correspon amb els que ens ofereix *WordPress*, amb els components més adients per tal de resoldre el nostre problema.



1

El problema que ens ocupa té com avantatge que és reduït en nombre de conceptes i és molt clar i concís.

Partim amb aquest model conceptual, on tenim únicament els següents conceptes:

- **Blog:** cadascuna de les comissions que volem representar. Està identificat per un id i guardem el nom de la comissió i la direcció per la qual podem accedir a ell directament.
- **Post:** articles que s'escriuen dintre dels blogs. Identificats per un id, escrits a una data i amb un títol i un contingut.
- **Comentari:** comentaris que es fan sobre els articles. Només tenim present la data i el seu contingut.
- **Usuari:** cadascun dels enginyers que conformen la blogosfera. Identificats per un id, amb un nom d'usuari (*login*), una contrasenya (*pass*) i una direcció de correu electrònic (*email*).

Associacions:

- **Blog – Post:** un blog conté molts articles (o cap) i un article només pot estar a un blog.
- **Blog – Usuari:** un blog conté molts usuaris, com a mínim un (l'administrador del blog), i un usuari pot pertànyer a molts blogs, però com a mínim a un. Un cas excepcional és l'usuari superadministrador que pertany a tots els blogs.
- **Usuari – Post:** un usuari escriu molts articles (sense cap restricció) i un article és escrit per un únic usuari.

¹ Model Conceptual inicial del sistema

- **Usuari – Comentari:** un usuari pot escriure els comentaris que vulgui, però un comentari estarà escrit per un únic usuari.
- **Post – Comentari:** un article pot tenir molts comentaris (o cap) i un comentari en concret serà d'un únic blog.

En aquest model he omès conceptes que apareixen al model conceptual de base de dades que té *WordPress* com són les taules meta (amb informació addicional de registres d'altres taules) o el site, donat que no són informació rellevant i no el podem considerar conceptes relatius amb el problema que ens ocupa.

6. Menú jeràrquic de blogs

6.1 Què és?

Com he explicat abans, tenim la necessitat de generar un menú que ens permeti accedir a cadascuna de les comissions que tenim creades.

Aquest menú ha de tenir en compte que tenim tres tipus diferents de blogs, que haurem d'identificar d'alguna manera:

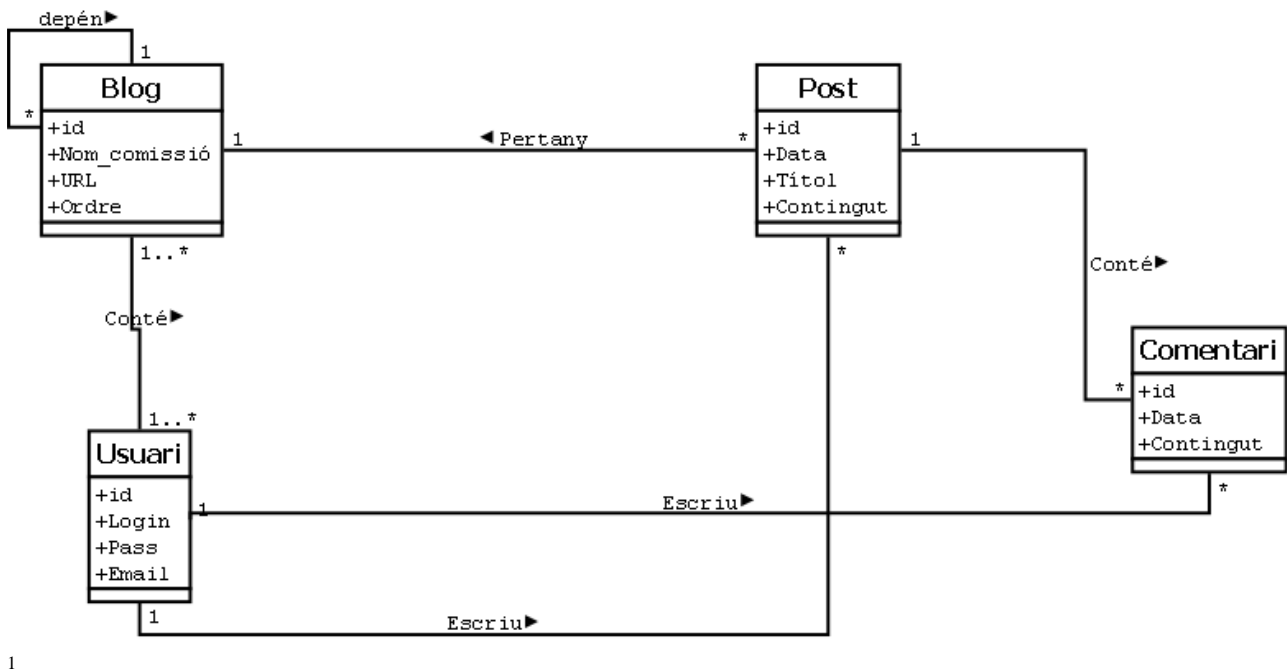
- **Blog principal:** és el blog que hem definit com el de presentació, per tant serà la *home* de la web.
- **Blog de comissió:** cadascuna de les comissions tindrà el seu blog independent. No depèn de cap altre blog.
- **Blog de subcomissió:** es tracta d'una comissió també, per tant és compost per un blog independent, però hi ha que identificar que pertany a una comissió en particular.

Aquest menú haurà de tenir en compte també el fet de que cadascun dels elements que el conformen tenen que anar a una posició específica, no poden anar a qualsevol lloc.

6.2 Canvis en el model conceptual del sistema

Com tenim una nova necessitat que afegir al sistema haurem de revisar el model conceptual que tenim fins ara per veure si tenim que afegir, esborrar o refer el que tenim, o si, per el contrari, no ens cal fer cap modificació i podem continuar amb el que ara mateix tenim.

L'afegit més important que hem d'avaluar és el fet de que hem d'identificar les diferents comissions. Ara mateix, des de el punt de vista del model conceptual actual, totes les comissions són exactament iguals, no en fem cap distinció en quant de si es tracta d'una comissió normal, d'una comissió subcomissió o de si es tracta del blog principal.



Per poder fer la distinció de comissions he decidit crea una nova associació reflexiva en el concepte blog.

¹ Model Conceptual amb l'afegit Jerarquia de Blogs

D'aquesta manera podem indicar que un blog depèn d'un altre (té un pare) i que un blog pare pot tenir tants fills com vulgui.

També es pot veure que hi ha una restricció que ens indica que tot blog tindrà un pare, donat que el blog principal serà el pare de tots, inclús de si mateix.

A més necessitem emmagatzemar un nou atribut dintre del concepte *Blog*, que és la posició que tindrà dintre del menú principal, aquest atribut és el que he anomenat *Ordre*.

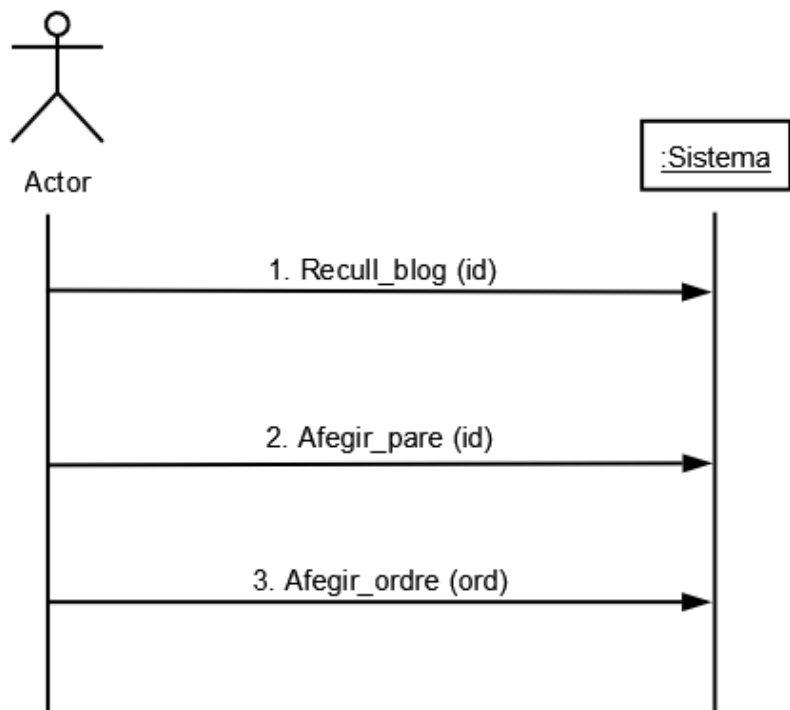
Restriccions semàntiques no contemplades al model conceptual

- El pare de cadascuna de les comissions serà la comissió a la que pertany (cas de comissió amb subcomissions).
- El blog principal tindrà com a pare a ell mateix.
- No pot haver-hi un blog sense ordre

6.3 Disseny de la solució. Cas d'ús AfegirDadesBlog

Per solucionar el problema, he decidit fer el disseny d'un cas d'ús anomenat AfegirDadesBlog, el qual recollirà les dades d'un blog, li afegirà el blog del que depèn i per últim també afegirà la posició del menú en la que es mostrarà.

6.3.1 Cas d'ús AfegirDadesBlog



¹ CU AfegirDadesBlog

6.3.2 Responsabilitats del cas d'ús

L'usuari demana al sistema fer la modificació de l'estructura d'un blog. A aquest blog recuperat li afegirem certa informació.

Primer recuperarem la informació del blog, un cop recuperada afegirem el blog del que depèn (el seu pare) i per últim la posició en la que anirà col·locat dintre del menú.

6.3.3 Esdeveniments

Recull_blog (id)

És el responsable de recuperar i mantenir actiu el blog escollit.

Afegir_pare (id)

És el responsable d'afegir el pare al blog que estem tractant.

Afegir_ordre (ord)

És el responsable d'afegir la posició dintre del menú en la que anirà el blog a tractar.

6.3.4 Detecció de paràmetres

Recull_blog (id)

- id: és l'identificador de la comissió a la que volem afegir la informació del pare i ordre.

Afegir_pare (id)

- id: és l'identificador de la comissió a la que pertany la comissió activa.

Afegir_ordre (ord)

- ord: és un numèric que indicarà la posició que tindrà el blog dintre del menú.

6.3.5 Contractes i diagrames de col·laboració

ES Recull_blog(id)

PRE:

- 1 – No hi ha cap blog actiu.
- 2 – Existeix al sistema un objecte b:Blog identificat per id

POST:

- 1 – L'objecte b:Blog es manté actiu.

PRE1: Cal per la correcta seqüenciació del ES.

PRE2: Aquesta PRE cal per què al diagrama de col·laboració s'assumeix que sempre es pot obtenir un Blog a partir del seu identificador.

POST1: Cal per la correcta comunicació entre els ES del cas d'ús.

Diagrama de col·laboració de l'ES Recull_blog(id)



ES Afegir_pare (id)

PRE:

- 1 – Tenim un blog actiu.
- 2 – Existeix al sistema un objecte bl:blog identificat per id.

POST:

- 1 – b té un nou concepte *pare* amb valor de bl .
- 2 – L'objecte b :Blog es manté actiu.

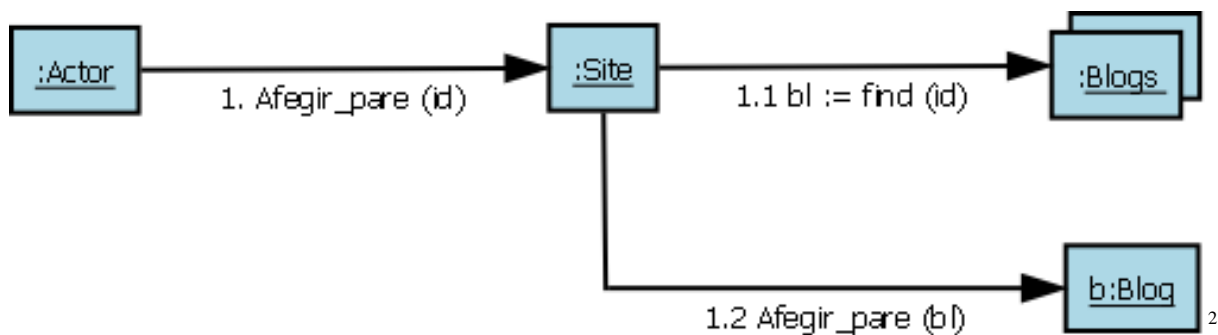
PRE1: Cal per la correcta comunicació entre els ES.

PRE2: Aquesta PRE cal per què al diagrama de col·laboració s'assumeix que sempre es pot obtenir un client a partir de l'identificador.

POST1: En el cas de que ja existeixi un concepte *pare* associat a b , es substituirà per el nou valor.

POST2: Cal per la correcta comunicació entre els ES.

Diagrama de col·laboració de l'ES Afegir_pare (id)



ES Afegir_ordre (ord)

PRE:

- 1 – Tenim un b :Blog actiu.
- 2 – ord és un valor numèric comprès entre 1 i el nombre màxim de blogs disponibles.

¹ Diagrama de col·laboració Recull_blog (id)

² Diagrama de col·laboració Afegir_pare (id)

POST:

1 – L'objecte b:Blog té un nou concepte *Ordre* amb el valor ord.

2 – No hi ha cap Blog actiu.

PRE1: Cal per la correcta comunicació entre els ES.

PRE2: Necessari perquè no s'insereixi un ordre superior al que podem mostrar.

POST1: Ens assegura que el valor de ord s'emmagatzemarà a l'atribut corresponent del component.

POST2: Cal per a la correcta comunicació entre les ES. Un cop el blog és inactiu podem tornar a cridar al cas d'ús per inserir l'ordre d'un altre blog diferent.

Diagrama de col·laboració de l'ES Afegir_ordre (ord)

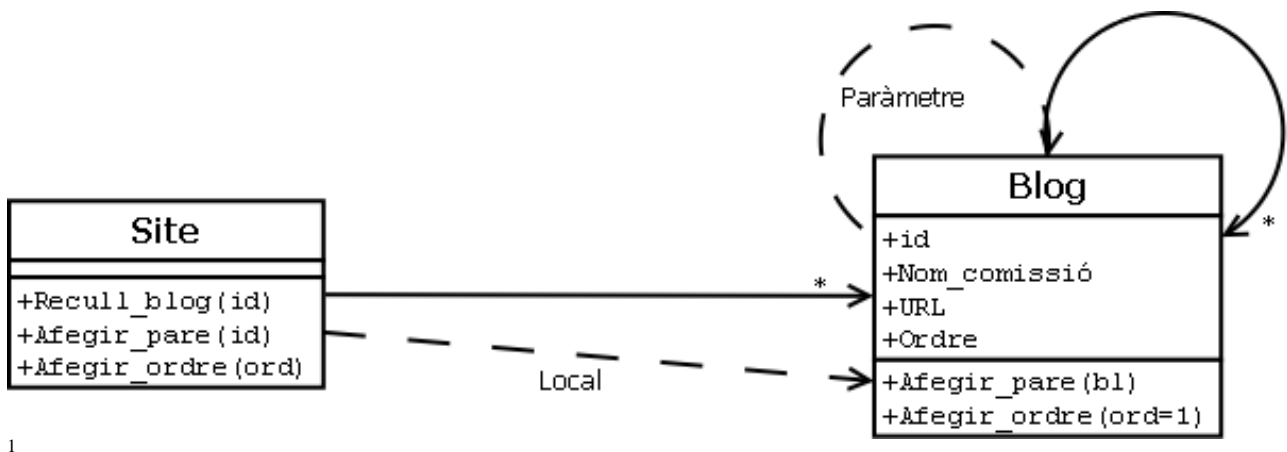


6.3.6 Visibilitats del cas d'ús AfegirDadesBlog

- **Visibilitats de paràmetre:** s'obtenen directament de les interfícies dels diferents components.
 - o Blog → Blog.
- **Visibilitats d'atribut:**
 - o Site → * Blog: Visibilitat de recuperació.
 - o Blog → * Blog: Visibilitat corresponent a la realització de l'associació del model conceptual.
- **Visibilitats locals:**
 - o Site → Blog: visibilitat de l'esdeveniment afegir ordre.

¹ Diagrama de col·laboració Afegir_ordre (ord)

6.3.7 Model de components del CU AfegirDadesBlog



6.3.8 Aspectes destacats a l'hora de dissenyar

A l'hora de dissenyar hi ha hagut certs aspectes que cal explicar per una millor comprensió de la solució proposada:

- He tingut la necessitat d'introduir un element controlador (*Site*), per tal de tenir un component que tingui accés a tot el sistema. El component *Blog* també era susceptible de ser controlador però ha estat descartat perquè no és únic e introduir un controlador nou hem permet facilitar la feina.
Aquest component, per tant, ha estat creat per conèixer tot el sistema i li assignem la responsabilitat d'expert. Com que es tracta d'un component inexistent en el model conceptual, estem violant el principi d'Espill, que indica que cada component del model es té que correspondre amb un concepte de la realitat.
- El primer que he tingut en compte a l'hora de realitzar el disseny ha sigut resoldre les encaixades passant els identificadors el més aviat possible, obtenint d'aquesta manera ràpidament objectes i treballant amb ells en quant podem.

6.4 Implementació de la solució

La implementació del plugin, un cop fet el disseny, és automàtica. Tal i com hem obtingut al model de components, hem d'implementar dues classes: la classe *Site*, que és el que fa de controlador de l'aplicació, i la classe *Blog*, que s'encarregarà de resoldre les peticions que li fem al component *Blog*.

La implementació sencera de les dues classes las podem trobar a l'annexa de programació del plugin *Jerarquia de Blogs*. Aquí, a més de les classes abans esmentades, ens trobem dos apartats més corresponents a la classe principal del plugin, *Jerarquia*, i amb l'arxiu de configuració de la barra lateral del tema que estem fent ús, *Sidebar*.

6.4.1 Classe Jerarquia

La implementació de la classe *Jerarquia*, és necessària donat que és el nucli del plugin de *WordPress*. Seguint les regles de programació de plugins de *WordPress*, tenim que crear una classe que funcioni per instal·lar i desinstal·lar el plugin, per definir les opcions de configuració d'aquest i per oferir a l'usuari administrador una finestra visual que permeti interactuar amb l'usuari.

Aquesta classe conté els següents mètodes:

Constructor

Aquest és el mètode que crida als demés. Defineix les funcions internes de *WordPress* per l'activació i desactivació del plugin e indica on trobem les opcions de configuració.

¹ Model de components del cas d'ús

Funcions install i uninstall

Aquestes dues funcions són les que ens permeten inserir noves taules a la base de dades en cas que fos necessari. S'executaran en quan s'activi i es desactivi el plugin a l'opció de plugins del taulell d'administració.

Funció config_page

És l'encarregada d'afegir les opcions de configuració al plugin. Definim el títol que té el plugin, el tipus d'usuari que pot fer ús d'ell i quina és la funció que s'utilitza per el cos del plugin.

Funció options_jer

Aquesta és la funció principal del plugin. És la part visual que es mostra a l'usuari perquè en faci ús del plugin.

En aquest cas, per oferir una manera còmode i simple de cara a l'usuari he decidit que el cas d'ús es mostri a mode de formulari que l'usuari ha d'omplir.

Amb això podem:

- Assegurar les PRE: Mitjançant un formulari amb camps en els que l'usuari només tingui que fer una selecció, ens assegurem que no s'introdueixin dades errònies i així mai fallarà l'encaixada que fem per recuperar l'objecte.
- Velocitat i comoditat per l'usuari: Amb el formulari podem realitzar el cas d'ús per a més d'un objecte a l'hora, amb el qual millorem i simplifiquem la feina per a l'usuari.

6.4.2 Classe Site

Aquesta és la classe corresponent al controlador que hem dissenyat anteriorment. Es compon per els mètodes abans explicats.

6.4.3 Classe Blog

És la realització del component amb el mateix nom. Conté els mètodes abans explicats, a més dels mètodes *getters* i *setters*.

A més han estat afegit d'altres atributs i mètodes per facilitar la inclusió del menú al tema. Aquests són els següents:

- Atribut fills: és un array on es guardarà quan faci falta els fills que té associats el blog en qüestió.
- Atribut vist: serveix per determinar si el blog ha estat mostrat per pantalla o no dintre del tema.
- Mètode path: recupera de base de dades la URL del blog.
- Mètode name: recupera de base de dades el nom del blog.

6.4.4 Sidebar

L'arxiu *Sidebar*, tal i com he explicat anteriorment, conté la informació que es mostrarà a les barres laterals. En el nostre cas, aquest *sidebar* és correspon amb el de la barra lateral esquerra, que és on volem mostrar el menú.

El funcionament d'aquest arxiu dintre del tema, en el nostre cas serà el següent:

- Recuperarem els blogs que tenim i els guardarem dintre d'un array.
- Ordenarem el contingut de l'array per tal de mostrar el menú en l'ordre indicat per l'usuari al plugin. Això ho aconseguim amb la funció *Quick_sort*.

- Pintem cadascun del blogs. Això ho aconseguim amb la funció `pinta`.

Per tant, hem definit dues funcions més, però que seran pròpies del tema, la funció `pinta` i la funció `quick_sort`.

Funció `pinta`

Aquesta funció s'encarrega de mostrar per pantalla l'array que tinguem amb tots els blogs. La funció mostrarà per pantalla el blog que toqui tractar, en el cas que no hagi ha estat ja inserit al menú, i, en el cas que tingui fills, farà una crida recursiva per encarregar-se d'inserir-los al menú.

Per aquesta funció és per la que ens ha fet falta crear els atributs nous *vist* i *fills* a la classe *Blog*. Gràcies a *vist* podem determinar si ja hem inserit el blog al menú, i gràcies a *fills* podem fer la crida recursiva sense cap problema.

Funció `quick_sort`

Aquesta funció implementa l'algorisme d'ordenació que porta aquest mateix nom. L'algorisme quick sort, o algorisme d'ordenació ràpida, és un algorisme recursiu d'ordenació d'arrays.

El seu funcionament és senzill. Indiquem un element de l'array com a pivot i posem els elements més petits a la seva esquerra i els elements més grans a la seva dreta.

D'aquesta manera, aplicant-lo recursivament aconseguim ordenar l'array d'una manera molt eficient, aconseguint un cost $O(n \cdot \log n)$ en el seu cas promig. El cas pitjor (amb cost $O(n^2)$) es dona quan el pivot es troba a l'extrem de l'array, però per això, he implementat l'array de forma que el pivot que agafi sigui l'element que es trobi a la meitat de l'array.

Gràcies a aquest algorisme aconseguim fer l'ordenació de l'array de blogs de forma molt eficient.

6.5 Interfície del cas d'us

Selecciona el pare de cadascun dels blogs existents

Blog	Pare	Ordre	Visibilitat
Accio i cooperacio	Home	1	Si
Acció professional	Home	2	Si
Administracions públiques	Home	3	Si
Construcció	Home	4	Si
Cultura	Home	5	Si
Energia	Home	6	Si
Esports	Home	7	Si
Formació	Home	8	Si
Gerència de riscos	Home	9	Si
Gestió empresarial	Home	10	Si
Jubilats	Home	11	Si
Medi ambient	Home	12	Si
Mobilitat i transport	Home	13	Si
Noves iniciatives	Home	14	Si
Packaging	Home	15	Si
Relacions amb les escoles d'enginyeria i les universitats	Home	16	Si
Seguretat, prevenció i protecció	Home	17	Si
TIC	Home	18	Si
Urbanisme	Home	19	Si
Enginyersmotorsport's	Mobilitat i transport	20	Si
Polígons Industrials	Construcció	21	Si
Club Bombeta	Home	22	Si
Impuls Pla Estratègic	Home	23	No

Actualitza

1

¹ Interfície del plugin Jerarquia de Blogs

7. Pàgina principal. COEIC Home

7.1 Descripció

Una de les necessitats més importants per al Col·legi d'Enginyers és la de tenir una pàgina principal personalitzada, on es mostrin els articles d'actualitat amb més rellevància, és a dir, els que consideren més importants.

Això, vist des del punt de vista de *WordPress*, es podria traduir com una plana estàtica on es mostrés certa informació que es consideri important. Això seria el que més s'ajustaria al que volem amb les opcions que ens ofereix *WordPress*.

Ara bé, no és el que volem. Amb la solució proposada per *WordPress* tindríem moltes limitacions i donaria moltes dificultats a l'usuari administrador. Un exemple de problema que ens podríem trobar, és el fet de voler actualitzar la plana, donat que hi hauria de canviar pràcticament tot el contingut sencer i de forma molt manual per tal d'obtenir una solució semblant al desitjat.

És per això que com a solució provaré de fer, d'una manera semblant al del menú, una barreja de plugin amb disseny del tema, que permeti fer exactament el que volem.

Aquesta solució es basa en la següent idea.

Definirem un blog com a blog principal, tal i com ja he suposat per fer el menú. Aquest blog, no pertanyerà a cap comissió en concret, no tindrà usuaris associats a ell amb permisos per escriure (excepte l'administrador i superadministrador) i no es podran fer publicacions en ell.

Les publicacions que es facin a aquest blog, seran els articles que volem que es mostrin a la plana principal i estaran controlades per el plugin que es dissenyarà.

D'aquesta manera aprofitem un dels inconvenients que té el *multisite*, com és el fet que cada blog sigui totalment independent.

Un altre problema amb el que ens trobem, és el fet de que, com ja he esmentat anteriorment, *WordPress* agafa els articles que s'han publicat i els presenta a l'usuari en ordre cronològic invers, és a dir, començant per el més nou, fins al més antic.

Per aquest motiu, és per el que caldrà també fer modificacions en el disseny del tema, perquè podem mostrar els articles que volem amb l'ordre que sigui més adient.

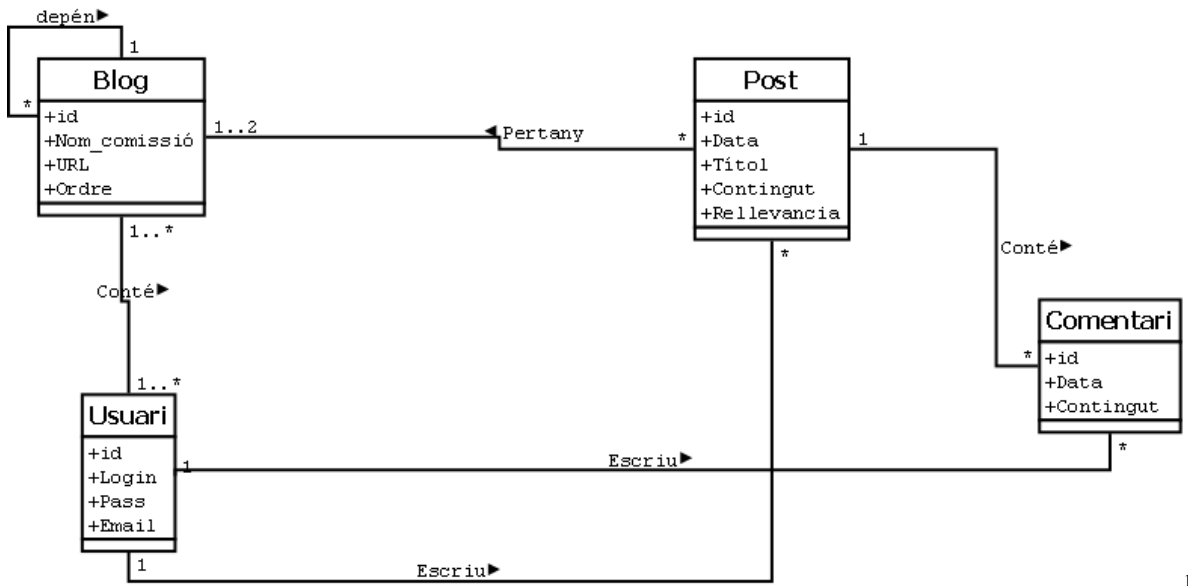
7.2 Canvis en el model conceptual del sistema

Com que anem a implementar una nova funcionalitat, tenim noves funcionalitats que abans no havíem contemplat a l'hora de fer el model conceptual. Per això, ara partirem del model conceptual que teníem i l'adaptarem a les necessitats que ara ens ocupen.

Un dels canvis que hem de tenir en compte és el de que ara, en alguns cassos, hi haurà articles que tinguin un nou concepte, la rellevància. Aquest concepte ens indicarà quina importància tindrà l'article dintre de la pàgina principal, es tractarà d'un valor numèric entre 1 i 5, on quan més petita sigui, menys important serà.

L'altre canvi que tenim que fer en el model és el fet que ara ens canvia la multiplicitat de l'associació Blog – Post, donat que fins ara teníem definit que un article només hi podia estar en un únic blog, ara tenim la possibilitat de que l'article es trobi a dos blogs diferents, l'original i el principal.

Per la resta, podem aprofitar el model que teníem fins ara.



Restriccions semàntiques no contemplades al model conceptual

- Un post pot pertànyer a dos blogs, però estarà limitat a que un dels blogs serà l'original i l'altre, obligatòriament serà el blog principal.
- La rellevància d'un post és opcional i serà un valor numèric comprès entre 1 i 5.

7.3 Disseny de la solució. Cas d'ús Introduir articles rellevants

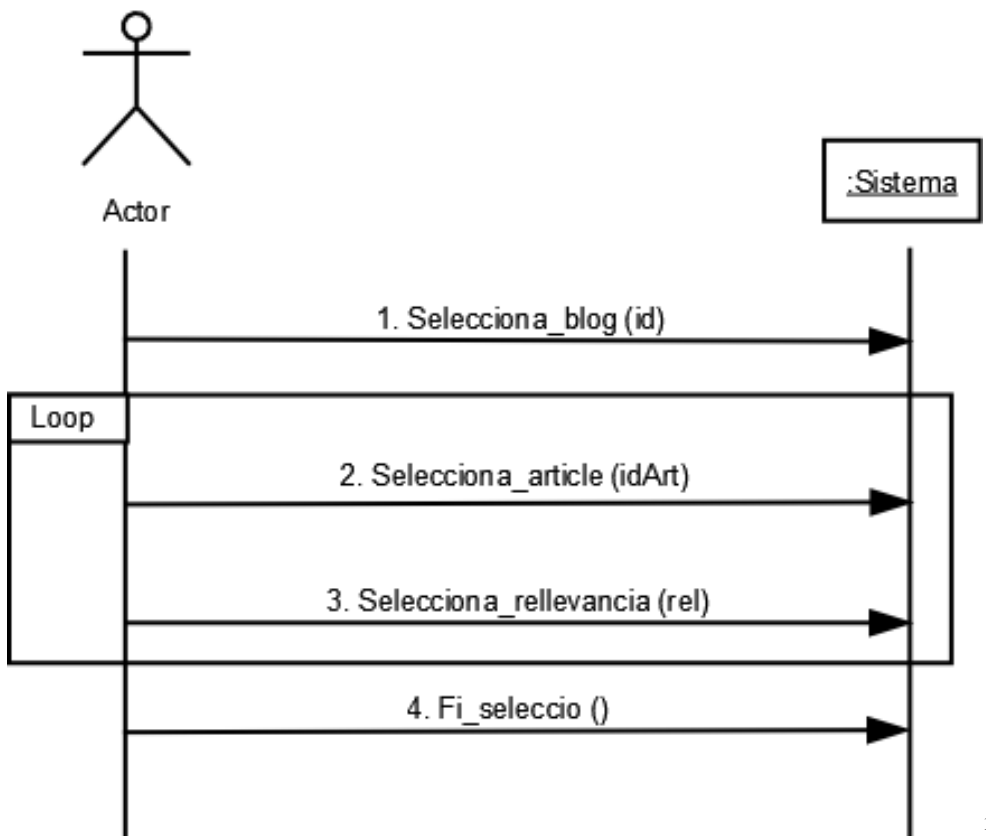
Per fer el disseny d'aquest nou problema, he creat un nou cas d'ús anomenat IntroduirArticlesRellevants.

Aquest cas d'ús, selecciona el blog del qual es vol extreure els articles, després es seleccionen els articles que volem afegir al blog principal i els assignem una rellevància.

Per últim afegirem els articles al blog principal.

¹ Model conceptual del cas d'ús

7.3.1 Cas d'ús IntroduirArticlesRellevants



7.3.2 Responsabilitats del cas d'ús

L'usuari demana al sistema introduir articles d'un blog determinat, dintre del blog principal. Per això recuperarem primer el blog en el que es troben els articles que volem.

Un cop tenim seleccionat el blog en qüestió, l'usuari demana al sistema l'article que li interessa.

Quan té seleccionat el article, li afegeix una rellevància per indicar-li la importància de l'article.

Després pot continuar afegint articles que estiguin dintre del mateix blog.

Quan finalitza d'introduir articles, el sistema guarda els articles seleccionats amb la rellevància escollida dintre del blog principal.

7.3.3 Esdeveniments

Selecciona_blog (id)

És el responsable de recuperar i mantenir actiu el blog origen dels articles.

Selecciona_article (idArt)

És el responsable de recuperar l'article que es vulgui traslladar al blog principal.

¹ Cas d'ús IntroduirArticlesRellevants

Selecciona_rellevància (rel)

És el responsable d'associar una rellevància a l'article seleccionat.

Fi selecció ()

És el responsable d'afegir els articles seleccionats al blog principal i finalitzar el cas d'ús.

7.3.4 Detecció de paràmetres

Selecciona_blog (id)

- id: és l'identificador de la comissió origen dels articles

Selecciona_article (idArt)

- idArt: és l'identificador de l'article que volem afegir.

Selecciona_rellevància (rel)

- rel: és un valor numèric comprès entre 1 i 5 que indica la importància de l'article.

7.3.5 Contractes i diagrames de col·laboració

ES Selecciona_blog (id)

PRE:

- 1- No hi ha cap blog actiu.
- 2- Existeix un b:Blog identificat per id.

POST:

- 1- L'objecte b:Blog es manté actiu.

PRE1: Cal per la correcta seqüenciació del ES.

PRE2: Cal per què al diagrama de col·laboració s'assumeix que sempre es pot obtenir un Blog a partir del seu identificador.

POST1: Cal per la correcta comunicació entre els ES del cas d'ús.

Diagrama de col·laboració de l'ES Selecciona_blog (id)



ES Selecciona_article (idArt)

PRE:

- 1- Existeix un blog actiu.
- 2- No hi ha cap article actiu.

¹ Diagrama de col·laboració del ES Selecciona_blog

- 3- Existeix un article identificat per idArt.
- 4- art encara no ha estat afegit.

POST:

- 1- b es manté actiu
- 2- Existeix un article art actiu.

PRE1: Cal per la correcta seqüenciació del ES.

PRE2: Cal per la correcta seqüenciació del ES.

PRE3: Necessari per què no podem afegir identificadors incorrectes.

PRE4: Cal per què no s'introdueixi dos cops el mateix article.

POST1: Cal per la correcta comunicació entre els ES.

POST2: Necessari per la correcta comunicació entre els ES.



ES Selecciona_relevància (rel)

PRE:

- 1- Hi ha un blog actiu.
- 2- Hi ha un article actiu.
- 3- rel és un valor numèric comprès entre 1 i 5.
- 4- Encara no s'ha introduït cap article amb aquesta rellevància.

POST:

- 1- L'article art té associat la rellevància rel.
- 2- art s'ha emmagatzemat al repositori d'articles rellevants.
- 3- Existeix un blog actiu.
- 4- No hi ha cap article actiu.

PRE1: Necessari per la correcta seqüenciació del ES.

PRE2: Cal per la correcta seqüenciació del ES.

PRE3: Necessari perquè no doni errors.

PRE4: Necessari per evitar redundància d'articles.

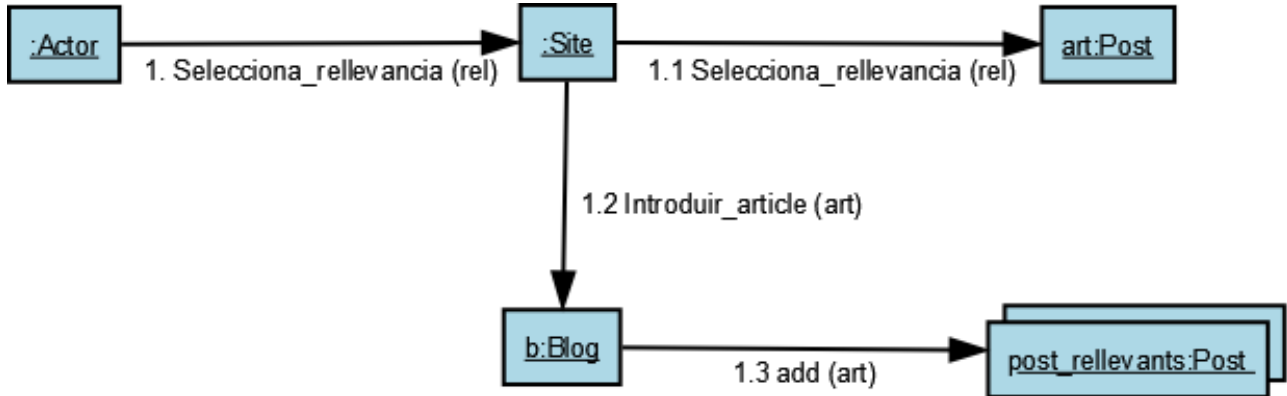
POST2: El necessitem per emmagatzemar temporalment els articles d'aquest blog que s'emmagatzemaran al blog principal.

¹ Diagrama de col·laboració del ES Selecciona_article

POST3: Necessari per la correcta comunicació entre ES.

POST4: El necessitem per poder tornar a iniciar el bucle en cas de voler afegir un nou article.

Diagrama de col·laboració de l'ES Selecciona_relevància (rel)



1

ES Fi_seleccio ()

PRE:

- 1- Hi ha un blog actiu.
- 2- A post_rellevants hi ha com a mínim un article.

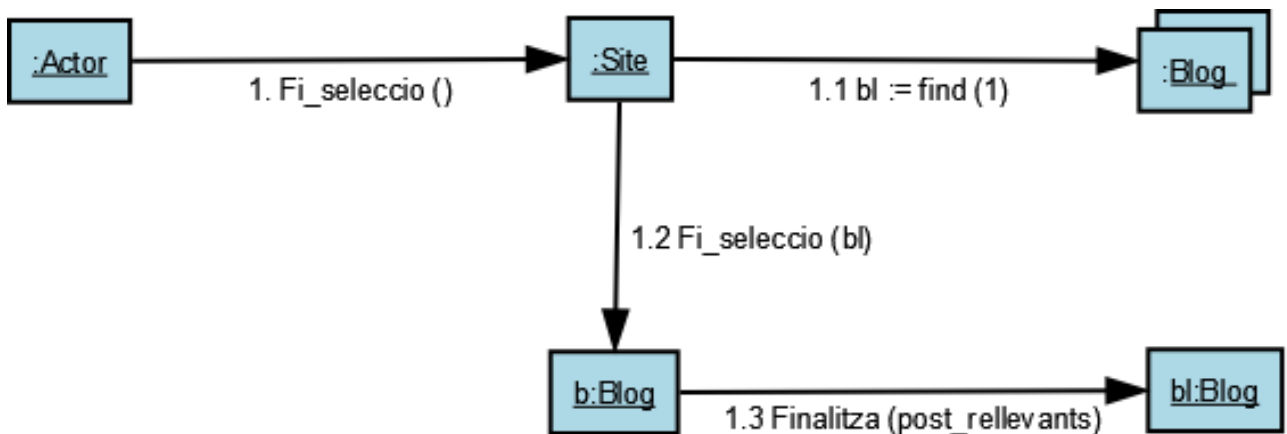
POST:

- 1- b deixa d'estar actiu.
- 2- Els articles de post_rellevants s'han introduït al blog principal.

PRE1: Necessari per la correcta seqüenciació dels ES.

PRE2: Ens serveix per assegurar-nos que els anteriors ES no han fallat.

POST1: Necessari per poder tornar a començar el cas d'ús.

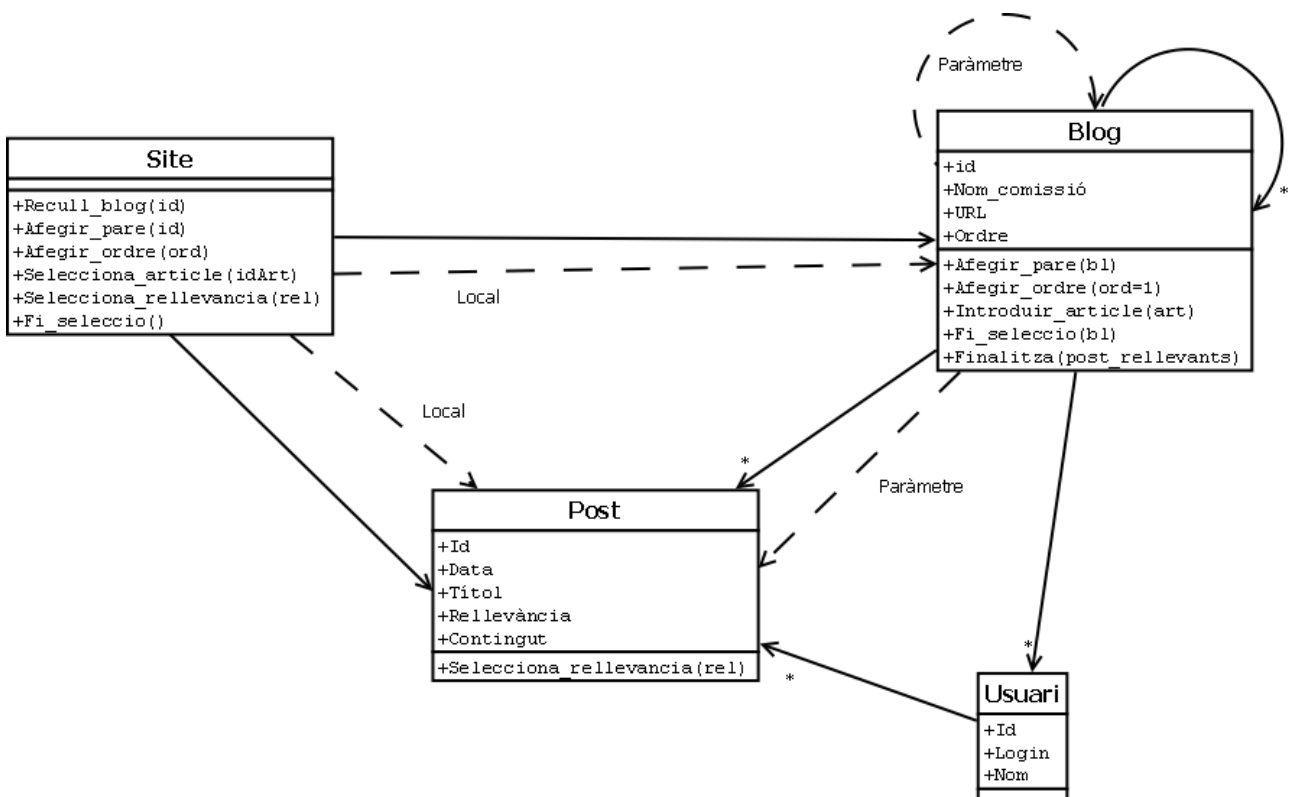


¹ Diagrama de col·laboració del ES Selecciona_relevancia

7.3.6 Visibilitats del cas d'ús IntroduirArticlesRellevants

- **Visibilitats de paràmetre:** s'obtenen directament de les interfícies dels diferents components.
 - o Blog → Blog.
 - o Blog → Post.
- **Visibilitats d'atribut:**
 - o Site → Post: visibilitat d'activació.
 - o Site → Blog: visibilitat d'activació.
 - o Blog → * Post: visibilitat d'emmagatzematge.
 - o Blog → * Usuari: realització de l'associació Blog – Usuari del model conceptual.
 - o Usuari → * Post: realització de l'associació Usuari – Post del model conceptual.
- **Visibilitats locals:**
 - o Site → Post: visibilitat de recuperació.
 - o Site → Blog: visibilitat de recuperació.

7.3.7 Model de components del cas d'ús IntroduirArticlesRellevants



¹ Diagrama de col·laboració del ES Fi_seleccio

² Model de components del cas d'ús

7.3.8 Aspectes destacats a l'hora de dissenyar

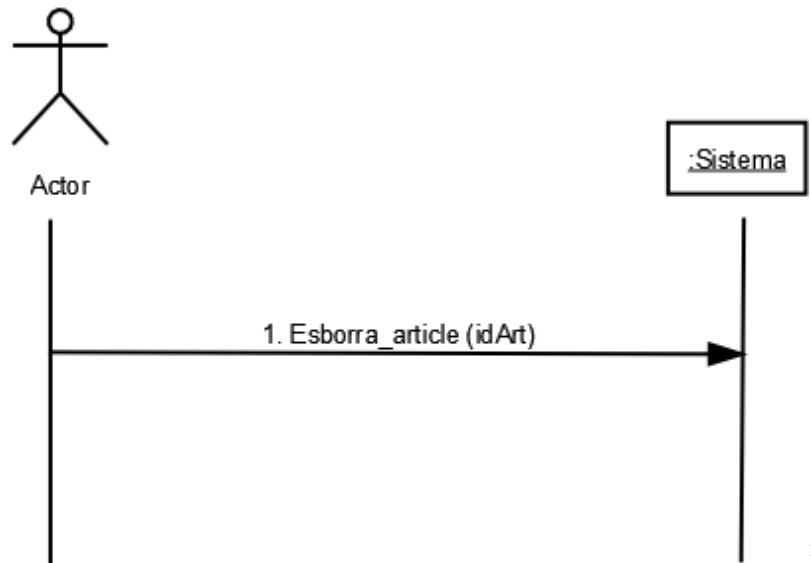
- Encara que els plugins dintre de *WordPress* són totalment independents, donat que tot pertany al mateix projecte, he decidit reaprofitar els components creats a l'anterior apartat. Així podem utilitzar de nou els components *Site* i *Blog*.
- Continuem amb el mateix controlador, *Site*, donat que és el que té coneixement de tot el sistema encara que violi el principi d'Espill.
- Introduïm dos components nous *Post* i *Usuari* que són realització dels conceptes amb mateix nom al model conceptual.
- *Post* ens permetrà recuperar les dades d'un article en concret.
- *Usuari* ens servirà per recuperar informació de l'usuari que escriu l'article. En aquest cas d'ús només en necessitem el seu identificador.
- Les encaixades són resoltes al principi de tots els diagrames de col·laboració per tal de poder treballar amb objectes en compte de amb identificadors.
- Com que a l'apartat anterior teníem un esdeveniment que ens permetia recollir un blog, al model de components no s'ha afegit el nou (*Selecciona_blog*) donat que es reaprofitarà l'esdeveniment *Recull_blog*.
- La necessitat d'introduir un quantificador d'importància als articles ens porta a introduir un concepte nou al component *Post*, aquest concepte és la *Rellevància* d'un article.
- A l'últim esdeveniment (*Fi_selecció*) com es pot comprovar, fem la recuperació d'un nou objecte blog, l'objecte que conté l'identificador igual a 1, que serà sempre el blog principal. En el nostre cas, és el blog on volem posar els articles rellevants.
- L'element fictici multiobjecte `post_rellevants:Post`, es crea per tal d'emmagatzemar en ell temporalment els articles que es vagin afegint en el cas d'ús.

7.4 Cas d'ús Esborrar articles rellevants

Igual que tenim la necessitat d'introduir els articles rellevants al blog principal, tenim la necessitat d'esborrar-los, donat que, com ja he explicat anteriorment, només permetem que hi hagi cinc articles al blog principal.

Aquest fet ens porta a la creació d'un nou cas d'ús per tal de que l'usuari seleccioni els articles que no vulgui que hi siguin a la pàgina principal i, en cas de no haver-hi cap article, introduir automàticament els cinc articles més recents per tal que sempre hi hagin articles a la pàgina principal.

7.4.1 Cas d'ús EsborraArticles



7.4.2 Responsabilitats del cas d'ús

L'usuari demana al sistema esborrar un article determinat de la pàgina principal. Per fer això li indica l'identificador de l'article.

El sistema s'encarrega d'esborrar-lo. Un cop ho ha eliminat comprova si hi ha articles dintre del blog principal. En cas que no hi hagi introdueix els cinc articles més recents, amb rellevància més alta l'article més nou, i més baixa el més antic.

7.4.3 Esdeveniments

Esborra_article (idArt)

És el responsable d'esborrar un article de dintre del blog principal.

7.4.4 Detecció de paràmetres

Esborra_article (idArt)

- idArt: és l'identificador de l'article que es vol esborrar.

¹ CU Esborra Article

7.4.5 Contractes i diagrames de col·laboració

ES Esborra_article (idArt)

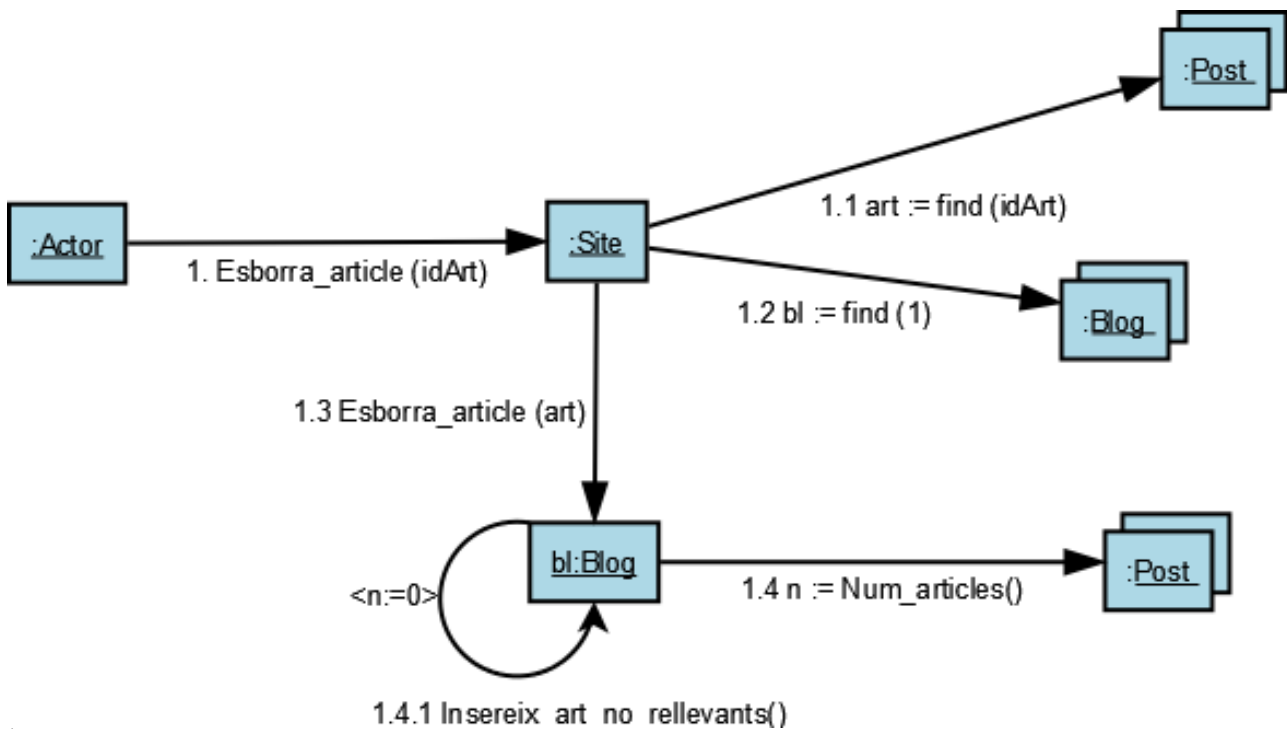
PRE:

- 1- Existeix un p:Post identificat per idArt

POST:

- 1- p:Post ja no existeix al blog principal.

PRE1: necessari per poder resoldre la encaixada.



7.4.6 Visibilitats del cas d'ús EsborraArticles

- Visibilitats de paràmetre:

- o Blog → Post.

- Visibilitats locals:

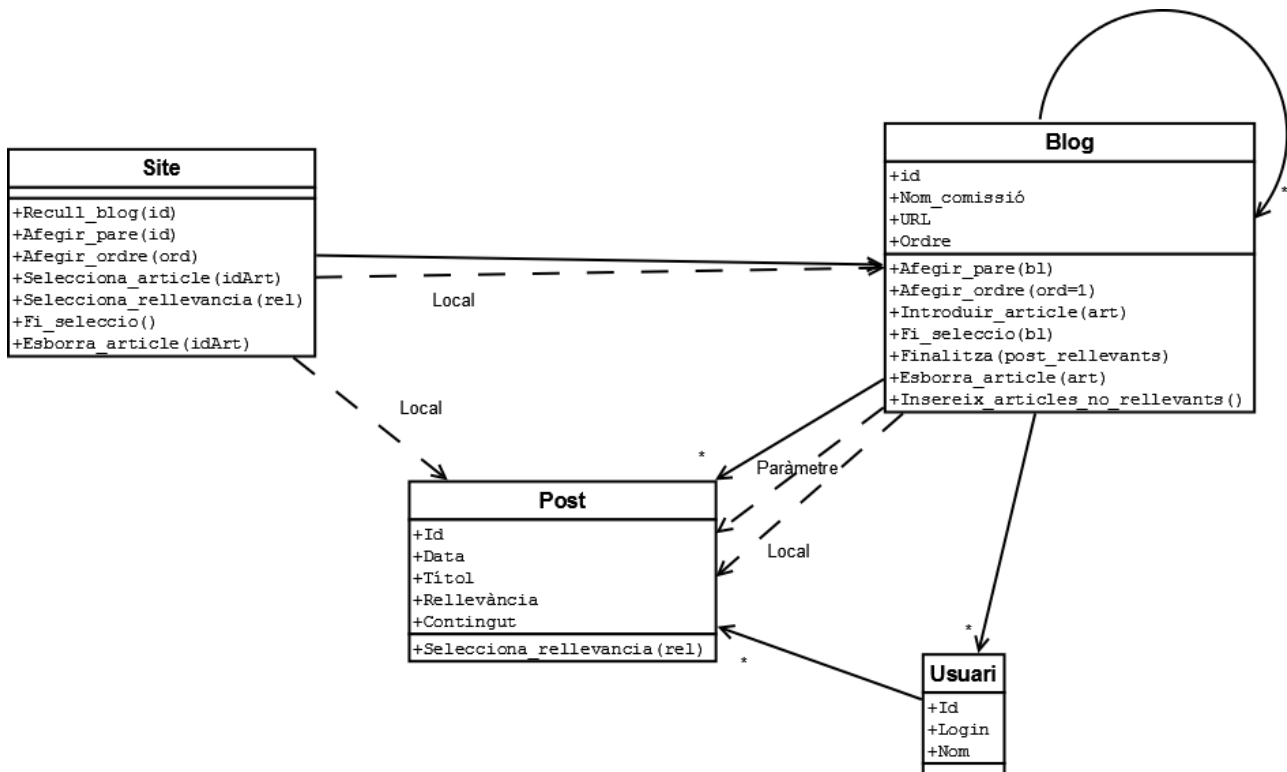
- o Site → Post: visibilitat de recuperació.
- o Site → Blog: visibilitat de recuperació.
- o Blog → Post: visibilitat de recuperació dels posts no rellevants.

- Visibilitats d'atribut:

- o Site → Blog.

¹ Esdeveniment de sistema Esborra_article

7.4.7 Model de components del cas d'ús EsborraArticles



1

7.4.8 Aspectes destacats a l'hora de dissenyar

- Per tractar el fet d'esborrar articles només necessitem d'un esdeveniment de sistema on l'usuari selecciona l'article que no vol que surti a la pàgina principal.
- Cada cop que s'esborra un article es comprova el nombre d'articles que resten al blog principal per si es dona el cas que es quedi sense articles.
- Com no podem deixar en blanc el blog principal, en cas que no tingui articles es carregaran els cinc articles més nous fent un ús automàtic del cas d'ús *IntroduirArticlesRellevants*.

7.5 Implementació de la solució

Igual que a l'anterior mòdul, un cop fet el disseny la implementació és ràpida. Com que hi hem hagut d'afegir nous components respecte a l'apartat anterior, hem de crear noves classes que permetin fer ús d'aquests. Així doncs, tenim dues noves classes al nostre projecte, *Post* i *Usuari*, que en son la realització dels components amb mateix nom (tal com indica el principi de fabricació pura).

A més, continuem tenint les classes ja creades (*Site* i *Blog*), les quals actualitzarem per introduir les noves funcionalitats.

Com estem tractant la implementació d'un nou plugin també haurem de fer ús d'una classe controladora per la instal·lació, configuració i visualització de les necessitats. En aquest cas tractem una nova classe anomenada *Coic*.

Per últim, estem desenvolupant una solució que afectarà al comportament normal de l'aplicació tant en la part d'administració com en la part visual. Per tant, tenim que fer modificacions al tema que estem fent ús per tal d'adaptar-lo als nous canvis. En aquest cas, es faran canvis a la pàgina principal del tema, això doncs

¹ Model de components del CU Esobrrar article

modificarem l'arxiu *index* del tema.

7.5.1 Classe Coeic

La implementació de la classe *Coeic*, igual que passava amb la classe *Jerarquia* a l'apartat anterior, se'ns fa absolutament necessària per complir amb les exigències de la programació de plugins de *WordPress*.

A través d'aquesta classe podrem activar el plugin, desactivar-lo i fer ús de la funcionalitat que ofereixen les demés classes. L'usuari interactuarà directament amb aquesta classe i el sistema traslladarà les exigències de l'usuari a la resta de components.

Trobem els següents mètodes:

Constructor

Defineix amb funcions de *WordPress* l'activació i desactivació del plugin i les opcions de configuració.

Funcions install i uninstall

S'encarreguen d'instal·lar les dades necessàries per el correcte funcionament del plugin. En aquest cas no necessitem fer ús d'aquestes funcions, donat que no introduïrem taules noves per satisfer les nostre necessitats, sinó que farem ús d'un blog en concret.

Funció config_page

Amb aquesta funció definim les opcions bàsiques que tindrà el plugin, com són el títol, tipus d'usuari que pot fer ús d'ell o la funció que inicialitza el plugin.

Funció options_plugin

Aquesta funció es defineix al mètode de configuració esmentat abans. Ens permetrà especificar tot el necessari per el correcte funcionament del plugin.

El funcionament que he proposat per la implementació és el següent:

- 1- **Control d'errors.** Primer de tot fem una comprovació dels articles que es decideixen afegir a la pàgina principal. Abans d'afegir un article al cas d'ús *IntroduirArticlesRellevants*, fem una comprovació d'errors per tal d'assegurar algunes precondicions que es marquen al esdeveniments. El control d'errors consta de les següents etapes:
 - a. **Control del màxim d'articles a la pàgina principal.** Si s'ha arribat a cinc el nombre d'articles continguts a la pàgina principal donarà un error i no introduirà l'article.
 - b. **Control de rellevàncies repetides.** Ens serveix per que no es doni el cas que introduïm dos articles amb la mateixa rellevància. Les rellevàncies són úniques dintre del blog.
 - c. **Control de no rellevància.** Igual que no podem tenir rellevàncies repetides, no podem tenir articles sense rellevància.
 - d. **Control d'articles repetits.** Per tal de no trobar-nos amb el mateix article dos cops dintre de la pàgina principal es fa un control dels articles. Considerem dos articles iguals si tenen el mateix títol i s'escriuen a la mateixa data per la mateixa persona.
- 2- **CU *IntroduirArticlesRellevants*.** Un cop hem comprovat que no tenim errors, podem anar a utilitzar el cas d'ús d'introducció d'articles per tal d'afegir els articles a la pàgina principal.
- 3- **CU *EsborrarArticles*.** Si hem escollit l'opció d'esborrar articles, procedim a fer la comprovació dels articles que s'han seleccionat i els hi apliquem el cas d'ús per esborrar-los.
- 4- **Paginació.** Amb l'objectiu de mostrar de forma clara els articles que es troben dintre de la blogosfera s'ha creat una paginació que ens permet veure els articles en conjunts de cinc.
- 5- **Filtració.** La forma d'escollir articles s'ha dividit en dos. L'opció predeterminada és la de mostrar tots els articles de la blogosfera per ordre cronològic, amb els més nous primer, però també es pot fer mitjançant una filtració dels articles per comissió, de forma que quan s'activa aquesta filtració només

apareixeran els articles corresponents a aquesta comissió, ordenats també de forma cronològica.

- 6- **Formulari d'adició.** Perquè l'usuari pugui fer un ús correcte de l'aplicació, he decidit que la interacció entre usuari i sistema per afegir articles es faci mitjançant un formulari. D'aquesta manera assegurarem més precondicions, com és el cas de les encaixades i de que no s'introdueixi una rellevància que no estigui compresa entre 1 i 5. El resultat d'aquest formulari és l'activació del cas d'ús d'introducció d'articles rellevants.
- 7- **Formulari d'eliminació.** Igual que amb el formulari d'adició, l'objectiu és assegurar precondicions i facilitar la feina a l'usuari. El resultat d'aquest formulari és l'activació del cas d'ús d'esborrar articles.

7.5.2 Classe Site

La classe *Site*, igual que amb el plugin anterior ens serveix de controlador de l'aplicació. Està formada per els mètodes realització dels esdeveniments que he dissenyat en els cassos d'ús anteriors.

7.5.3 Classe Blog

Reutilitzem la classe creada al plugin anterior i li afegim les noves funcionalitats. Així disposarà dels mètodes *Introduir_article*, *Fi_seleccio*, *Finalitza*, *Esborra_article* e *Insereix_articles_no_rellevants*, que fan referència als mètodes dissenyats als esdeveniments de sistema que hem creat al dos cassos d'ús que hem utilitzat.

Aquest cop no tenim que afegir nova informació en quant a atributs es refereix, donat que podem utilitzar dels que ja disposem.

7.5.4 Classe Post

Aquesta classe és nova dintre del sistema. És la realització del component amb mateix nom. Conté els mètodes i atributs especificats en aquest component. D'aquesta manera ens trobem amb els atributs *id*, *Data*, *Títol*, *Rellevància* i *Contingut* a més d'un atribut *Blog* fruit de l'associació amb aquest component.

Els mètodes implementats són els corresponents amb la implementació del component, a més dels mètodes *getters* i *setters* per accedir als atributs i els mètodes privats *data*, *tit*, *aut* i *cont*, que permeten la recuperació de les dades de la base de dades.

7.5.5 Classe Usuari

La classe *Usuari*, també és nova dintre del sistema. Implementa el component amb mateix nom. Per el moment no té mètodes específics i l'utilitzem per recuperar dades dels autors.

7.5.6 Índex

L'índex, com abans he explicat només varia per el blog principal. En ell es mostren, primer un *slider* amb les imatges destacades dels articles i després, es mostren els cinc articles més rellevants.

Com que la funcionalitat natural de *WordPress* (i de tots els blogs en general) és la de mostrar els articles en ordre cronològic invers, no ens mostraria els articles en l'ordre que voldríem.

A més, els articles que apareixeran a aquesta plana (els continguts al blog principal) seran una copia dels originals que es troben a la corresponent comissió, i per tal de no duplicar les imatges destacades aquestes no s'han introduït en els articles del blog principal i es recuperen de l'original.

La idea que es vol transmetre en el blog principal és la de mostrar les característiques més importants de l'article (*títol*, *autor*, *data* i *extracte*) recollides de l'article copiat i quan accedim a l'article, mostrar

l'original.

Per això el *permalink*, és a dir, la direcció URL per la que accedim a l'article, serà la de l'article original. Amb això aconseguirem que per l'usuari els articles continuïn sent únics i que els comentaris que s'escriuin sobre l'article, es facin sobre l'original directament.

Llavors, seguint les necessitats abans esmentades, l'arxiu índex ha quedat modificat i distingeix dues opcions:

- **Blog principal:** funció principal modificada. Recull els articles en ordre de rellevància i en mostra títol, autor, data i extracte de l'arxiu copiat i la imatge destacada i el *permalink* de l'original. Si accedim a un article, es mostra l'article original.
- **Altres blogs:** segueix la seqüència normal dels blogs per mostrar els articles.

7.6 Interfície del cas d'ús

Selecciona els posts que vulguis que surtin a la pàgina principal

Mostran't posts del 1 fins al 5 de 8

Totes les Comissions ▾

Títol	Comissió	Rellevància	Selecciona
Entrada 2 accio i coop	Acció i Cooperació	▾	<input type="checkbox"/>
Entrada accio i cooperacio	Acció i Cooperació	▾	<input type="checkbox"/>
Entrada accio prof	Acció Professional	▾	<input type="checkbox"/>
Segunda entrada de cultura	Cultura	▾	<input type="checkbox"/>
Entrada nueva cultura	Cultura	▾	<input type="checkbox"/>

Pàgina
1 ▾

Acceptar

1

Posts que surten actualment a la pàgina principal

Títol	Comissió	Rellevància	Esborrar
Enginyers Industrials de Catalunya	Gestió empresarial	5	<input type="checkbox"/>
El nou curs amb Bolonya	Formació	4	<input type="checkbox"/>
Produir energia és un joc	Cultura	3	<input type="checkbox"/>
Enginyers Industrials de Catalunya, un col·lectiu d'empreses	Gestió empresarial	2	<input type="checkbox"/>
Enginyeria i ràdio, al servei de les persones	Energia	1	<input type="checkbox"/>

Esborra

2

¹ Interfície corresponent al cas d'ús IntroduirArticlesRellevants

² Interfície corresponent al cas d'ús EsborrarArticle

8. Quickr to WordPress

8.1 Descripció

El COEIC disposa d'una eina actualment en la que algunes comissions poden escriure articles de forma semblant als blogs. Aquesta eina es diu *Quickr* i forma part de l'eina estudiada al principi del projecte, *IBM Portal*.

Quickr és una eina privada, tot el que no sigui membre col·legiat no pot entrar, així doncs interessa que els articles que es publiquin a *Quickr*, es publiquin també a *WordPress*, per tal de tenir una blogosfera completa i que aquesta sigui el centre de totes les discussions dels enginyers industrials.

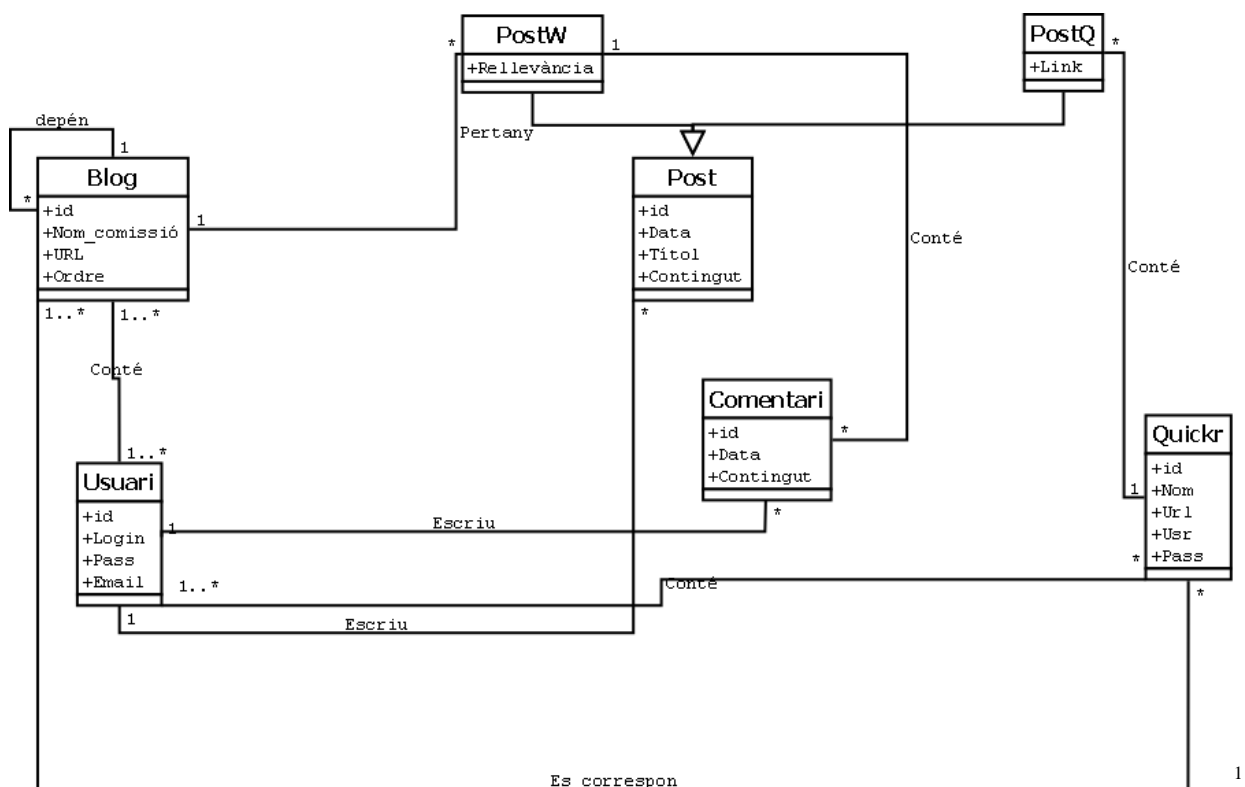
Per tant, com que no en tenim eines que ens permetin fer el que volem, hi ha que desenvolupar una nova funcionalitat per resoldre la necessitat que ens ocupa. Per fer-ho, aplicaré dos afegits, un a *WordPress* perquè l'usuari pugui importar els articles que li faci falta de *Quickr*, i una altre a *Quickr*, perquè també es pugui fer el mateix procés un cop s'hagi escrit un article.

8.2 Canvis en el model conceptual del sistema

Anem a introduir una funcionalitat que farà ús de dades externes al sistema que tenim. Es per això que hem d'introduir nous canvis al model conceptual del sistema.

Per una banda tindrem un nou concepte, que anomenaré *Quickr*. Representa una comissió establerta dintre d'aquest programa.

D'altre banda hem de tenir en compte que el que hem de recollir de *Quickr* són igualment articles (*Posts*) encara que la informació que tinguem d'ells no serà exactament la mateixa. Això ens comporta que canviem el concepte *Post* que ja teníem i el transformem a una estructura jeràrquica amb dos conceptes que tindran diversos atributs en comú, aquests dos conceptes s'anomenaran *PostW* (per als articles de *WordPress*) i *PostQ* (per als articles de *Quickr*).

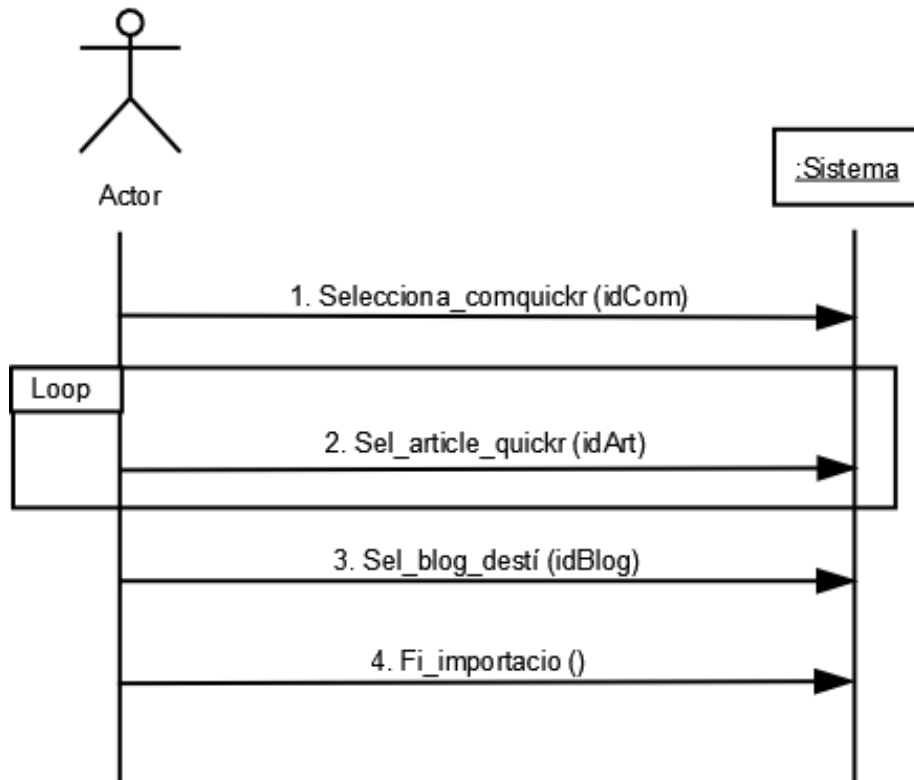


¹ Model conceptual corresponent a la funcionalitat *Quickr To WordPress*

8.3 Disseny de la solució. Cas d'ús ImportarArticle

Per tal de resoldre la primera de les noves funcionalitats que hi ha que implementar, la importació d'articles des de *Quickr*, he decidit fer un cas d'ús anomenat *ImportarArticle*. Aquest, ens permetrà seleccionar la comissió de *Quickr* que necessitem, els articles que volem afegir i el blog de *WordPress* on ha d'anar.

8.3.1 Cas d'ús ImportarArticle



8.3.2 Responsabilitats del cas d'ús

L'usuari demana al sistema introduir articles d'una comissió de *Quickr* determinada. Amb això accedim al RSS de la comissió.

Un cop hem accedit al RSS de la comissió de *Quickr* l'usuari selecciona els articles que vol importar. Quan té els articles que vol, selecciona el blog de *WordPress* en el que vol afegir els articles i finalment demana al sistema que s'encarregui de fer la importació dels articles.

8.3.3 Esdeveniments

Selecciona_comquickr (idCom)

És el responsable de seleccionar el blog *Quickr* del que es vol obtenir informació.

Sel_article_quickr (idArt)

És el responsable de seleccionar i recuperar les dades de l'article identificat per idArt

¹ Cas d'ús Importar Article

Sel_blog_destí (idBlog)

És el responsable de seleccionar i mantenir actiu el blog *WordPress* al qual hi aniran els articles importats.

Fi_importació ()

És l'encarregat de finalitzar la importació dels articles.

8.3.4 Detecció de paràmetres

ES Selecciona_comquickr (idCom)

- idCom: correspon a l'identificador de la comissió de la que volem fer ús.

ES Sel_article_quickr (idArt)

- idArt: és l'identificador de l'article de *Quickr*.

ES Sel_blog_destí (idBlog)

- idBlog: és l'identificador de la comissió de *WordPress*.

8.3.5 Contractes i diagrames de col·laboració

ES Selecciona_comquickr (idCom)

PRE:

- 1- No hi ha cap comissió activa.
- 2- Existeix un com:Quickr identificada per idCom.

POST:

- 1- com es manté activa.

PRE1: Necessari per a la correcta seqüenciació del cas d'ús

PRE2: Ens assegura que no falla l'encaixada.

POST1: La necessitem per la correcta comunicació entre ES.



ES Sel_article_quickr (idArt)

PRE:

- 1- Hi ha una comissió activa.
- 2- Existeix un objecte art:PostQ identificat per idArt.

¹ ES Selecciona_comquickr

3- Art encara no es troba al repositori d'articles per importar.

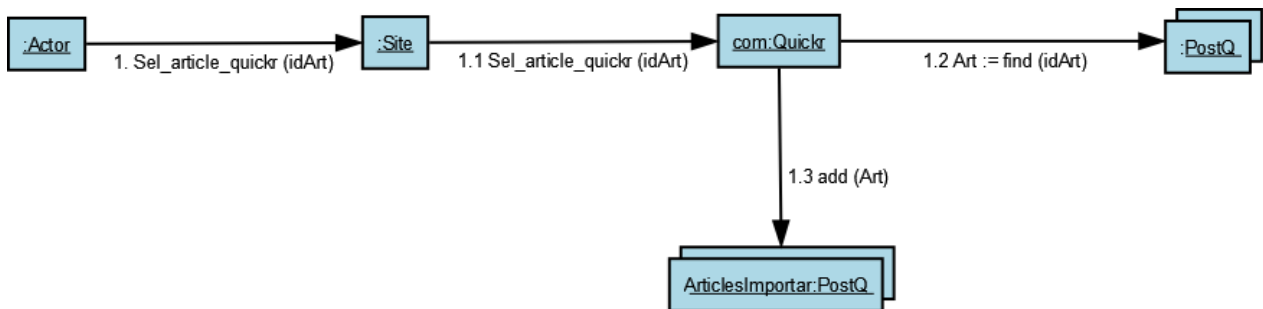
POST:

- 1- com es manté activa.
- 2- Art s'ha afegit al repositori d'articles per importar.

PRE2: Ens assegura que no falla l'encaixada.

PRE3: Serveix perquè no introduïm dos cops el mateix article.

POST2: Tindrem un repositori fragmentat que ens permetrà saber quins són els articles a importar.



1

ES Sel_blog_destí (idBlog)

PRE:

- 1- Hi ha una comissió activa.
- 2- Existeix un b:Blog identificat per idBlog.
- 3- Com a mínim hi ha un article al repositori d'articles per importar.
- 4- No hi ha cap blog actiu.

POST:

- 1- b esdevé actiu
- 2- Hi ha una comissió activa.

PRE2: Ens assegura que no falli l'encaixada.

PRE3: Serveix per assegurar que l'esdeveniment anterior és correcte.

POST1: Necessari per la correcta seqüenciació del cas d'ús.



2

¹ ES Sel_article_quickr

² ES Sel_blog_destí

ES Fi_importació ()

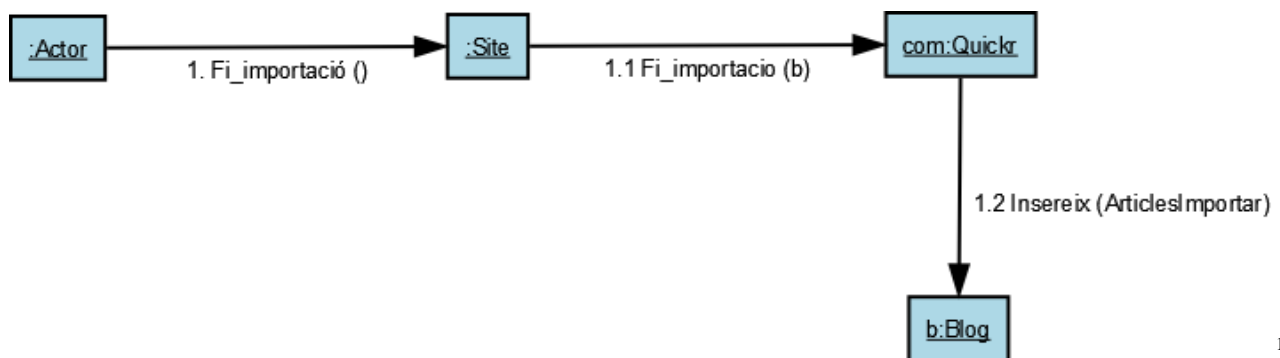
PRE:

- 1- Hi ha una comissió activa.
- 2- Hi ha un blog actiu.

POST:

- 1- Els articles del repositori d'articles a importar es troben al blog b.
- 2- No hi ha cap blog actiu.
- 3- No hi ha cap comissió activa.

POST1: Ens assegura que el cas d'ús ha finalitzat correctament.



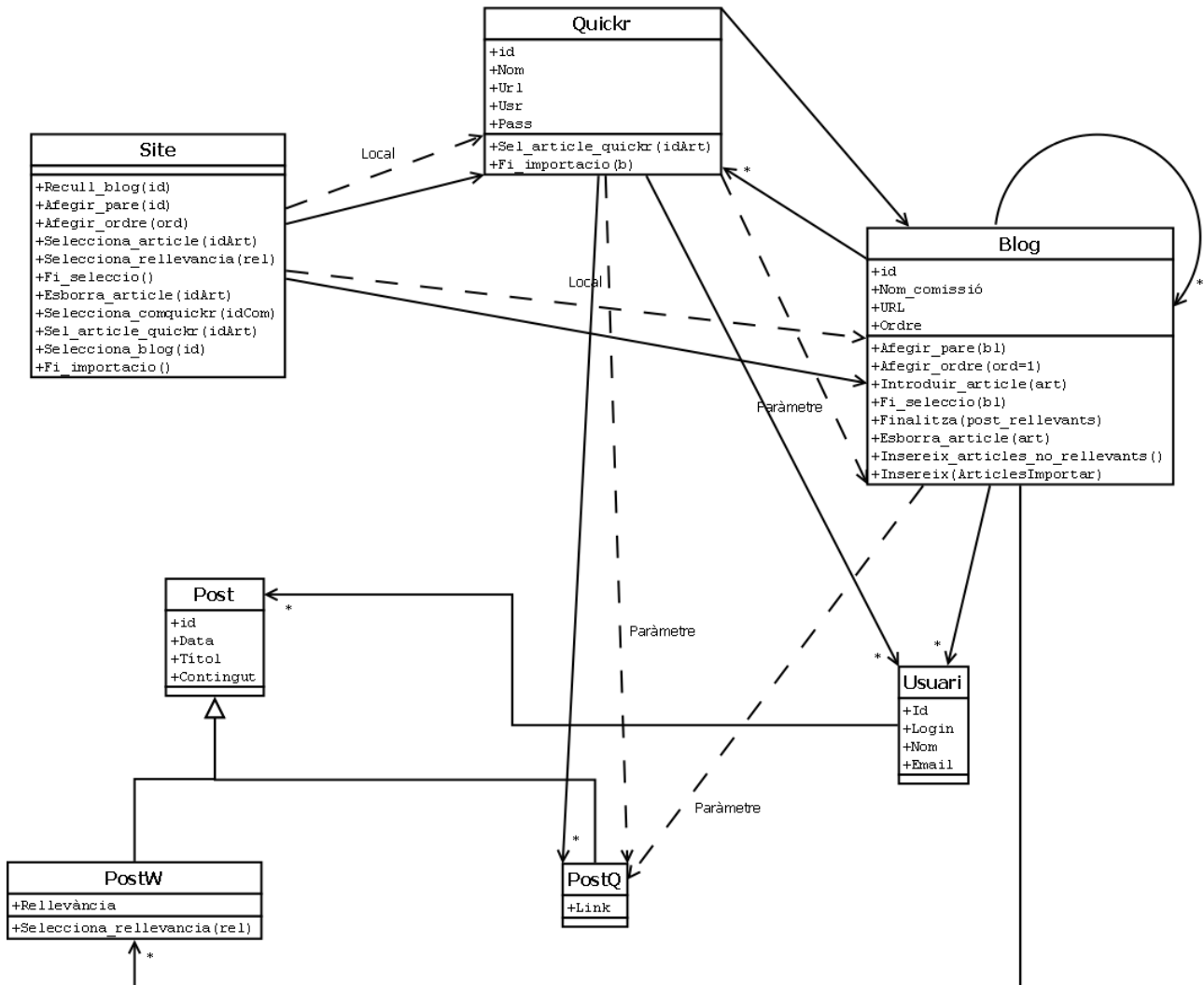
1

8.3.6 Visibilitats del cas d'ús ImportarArticle

- **Visibilitats de paràmetre:**
 - Blog → PostQ: extreta del diagrama de col·laboració del ES Fi_importacio.
 - Quickr → Blog: extreta del diagrama de col·laboració del ES Fi_importacio.
 - Quickr → PostQ: extreta del diagrama de col·laboració del ES Sel_article_quickr.
- **Visibilitats locals:**
 - Site → Blog: visibilitat de recuperació.
 - Site → Quickr: visibilitat de recuperació.
- **Visibilitats d'atribut:**
 - Site → Quickr: visibilitat d'activació.
 - Site → Blog: visibilitat d'activació.
 - Quickr → Blog.
 - Quickr → * PostQ: visibilitat d'emmagatzematge.
 - Blog → * Quickr: realització de l'associació del model conceptual.
 - Quickr → * Usuari: realització de l'associació del model conceptual.
 - Blog → * PostW: realització de l'associació del model conceptual.

¹ ES Fi_importació

8.3.7 Model de components del CU ImportarArticle



8.3.8 Aspectes destacats a l'hora de dissenyar

- Com havíem fet abans, continuem fent ús dels mateixos components per tal de tenir un sistema complet.
- Continuem fent ús del mateix component controlador del sistema, *Site*.
- Els canvis al model conceptual ens obliguen a introduir canvis també a l'hora de dissenyar.
 - o Tenim un nou concepte Quickr, que es tradueix en un nou component amb el mateix nom.
 - o La introducció de l'estructura jeràrquica dels articles (*Post*) ens porta a distingir també en el disseny entre articles que són de *WordPress* i articles que provenen de *Quickr*.
- Al model de components, com que estem representant exclusivament els canvis en aquest cas d'ús s'ha obviat la inclusió del concepte *Comentari*, donat que no es fa servir directament al cas d'ús. Es parteix de la base del model de components de l'apartat anterior, per tant, es mantenen els mètodes

anteriors i les visibilitats provinents del model conceptual que afecten en aquest cas d'ús.

8.4 Implementació del la solució

A l'hora d'implementar aquesta nova funcionalitat ens trobem amb diversos problemes que fins ara no teníem. El fet de tenir que accedir a una aplicació externa a *WordPress* i que a més, aquesta eina sigui privada, fa que la implementació no sigui tan automàtica com ha sigut en els cassos anteriors.

A més, per adaptar el plugin a les necessitats que ens pertoquen tenim que fer modificacions en les classes creades anteriorment i crear-ne de noves.

Així doncs, per el desenvolupament d'aquest plugin ens trobem amb les classes ja creades anteriorment *Site* i *Blog*, les quals hem de ampliar, la classe *Post* es converteix en genèrica per poder tractar els articles tant de *WordPress* com de *Quickr*. A més implementem altres classes noves com son la pròpia del plugin per *WordPress*, les classes *PostW* i *PostQ*, extensions de *Post* i la classe *Quickr*. Totes elles són la realització dels elements del disseny amb el mateix nom.

El nom de les classes als plugins a la implementació varia respecte al nom dels components, violant així el principi de fabricació pura de disseny, però és essencial per tal de que pugui funcionar correctament a *WordPress*, donat que no pot haver-hi dues classes amb el mateix nom. Per tant el nom de les classes passen a ser el seu nom propi, tot seguit de la inicial del plugin al que fan referència (Q per el plugin actual, H per el plugin COEIC Home i J per el plugin Jerarquia de blogs).

8.4.1 Classe xuclablog

Aquesta classe es la principal del plugin. Com tot plugin implementat a *WordPress*, requereix un plugin principal que serà el que faci d'interfície per a l'usuari, el que ens permeti instal·lar les taules que necessiti el plugin i el que faci les accions.

Aquesta classe consta dels següents mètodes:

Constructor

Igual que en els altres plugins s'encarrega de fer les accions principals del plugin: instal·lació, desinstal·lació i adició al menú.

Funcions install i uninstall

A diferència del plugins que havíem implementat fins ara, per aquest requerim d'una taula on guardem les comissions de *Quickr* a les que podem accedir per tal d'importar els articles.

Per aquest motiu utilitzarem la funció *install* per tal de crear aquesta nova taula dintre de la base de dades amb uns valors inicials ja introduïts. En el cas que es vulguin afegir més comissions de *Quickr* o esborrar algunes de les existents hi haurà que accedir a aquesta taula que creem ara mateix.

La taula en qüestió s'anomenarà *xucla* i contindrà els localitzadors i el nom de les comissions a les que tinguem els articles que vulguem importar.

La funció *uninstall* esborrarà la taula i tot el seu contingut.

Funció *config_page*

Aquest funció, com he explicat a l'hora d'implementar les anteriors funcionalitats ens serveix per configurar les opcions principals del plugin.

A diferència dels anteriors plugins, el nombre d'usuaris que podrà fer aquesta feina no serà només el superadministrador i l'administrador del blog, sinó que hi haurà un altre tipus d'usuari que podrà fer ús d'això. Aquest usuari es tracta dels editors, que, tal com he explicat anteriorment és el que pot escriure articles i planes al seu blog sense restriccions. Per això, com que serà un nombre molt limitat d'usuaris, se li afegeix aquesta opció.

Funció options_xucla

El cos del plugin està format de forma seqüencial per tal de que es correspongui amb el correcte funcionament del cas d'ús.

D'aquesta manera, amb una interfície clarificadora i restrictiva aconseguim complir la majoria de les precondicions que especificàvem en el disseny.

Per tant el funcionament del cos de cara a l'usuari és el següent:

- **Selecció de la comissió d'origen.** Primer de tot, ha de seleccionar la comissió d'origen de la que voldrà recuperar els articles pertinents a *Quickr*. Aquesta selecció estarà restringida per els registres de la taula que hem creat anteriorment. D'aquesta manera assegurem les encaixades i per tant, ens evitem errors.
- **Selecció d'articles.** Un cop s'ha seleccionat la comissió d'origen, es carregaran els articles que pertanyen a la comissió. Dels articles es mostren el títol, l'autor que el va publicar i la data de publicació, ja que són els camps més identificatius dels articles. A més es permet accedir a l'article original, per tal de visualitzar-lo i així assegurar-se de que es selecciona l'article correcte.
- **Selecció del blog destí.** Per últim, un cop s'han seleccionat els articles que es volen importar, hi haurà que seleccionar els blog al que aniran destinats. Aquesta funcionalitat tindrà dos casos diferents, el cas en el que l'usuari que està fent el procés sigui el superadministrador, cas en el que podrà afegir els articles al blog que vulgui, i el cas en que l'usuari actiu sigui l'administrador del blog o un editor, que com és lògic, no permetem que pugui posar l'article allà on vulgui, sinó que només podrà afegir l'article a la seva comissió.

Un cop fet tot el procés es crida a l'últim esdeveniment del cas d'ús, que es correspon amb el botó *Inserta* de la interfície gràfica del plugin.

8.4.2 Classe Site

La classe *Site*, tal i com hem vist fins ara, serà l'encarregada de fer de controlador de l'aplicació, capturarà els esdeveniments dels sistema i els transmetrà als components adients.

Està formada per tant per els mètodes de captura dels esdeveniments del cas d'ús, a més de tenir els que abans havíem fet servir.

8.4.3 Classe Post

En aquest cas la classe *Post* ja no es refereix a la informació que conté un article de *WordPress*, tal i com la havíem tractat fins ara, sinó que passa a ser una classe genèrica de *Posts*. Aquí tindrem els conceptes comuns entre els articles de *WordPress* i els de *Quickr*.

Els mètodes que conté són simplement els de recuperació i modificació dels atributs privats que la formen.

8.4.4 Classe PostQ

La classe *PostQ* és la realització del component amb el mateix nom. Per tant, s'encarrega de representar els articles que estan continguts a *Quickr*. Com és una classe heretada de *Post*, conté els seus mateixos conceptes, a més dels nous que identifiquen a un article de *Quickr*, com és el link de l'article.

Els mètodes que conté aquesta classe també són els de recuperació i modificació dels atributs.

8.4.5 Classe PostW

PostW és la classe que representa els articles que s'han escrit a *WordPress*, per tant, és la classe que havíem

tingut fins ara. Per això, conté les propietats que havíem tractat en anteriors apartats, com és la rellevància. També està format per els mètodes *getters* i *setters* exclusivament.

8.4.6 Classe Blog

La classe *Blog*, és la mateixa que hem utilitzat fins ara, per tant, conté tots els mètodes que havíem implementat fins al moment, a més d'un nou mètode, *Insereix*

Mètode *Insereix*

El mètode *Insereix* s'encarrega d'importar els articles de *Quickr* que s'han seleccionat per inserir. Per fer-ho, rep com a paràmetre el repositori d'articles per importar a *WordPress* i els tracta individualment.

Primer de tot crea un article de *WordPress* amb les dades que necessita (títol, autor, contingut...) i comprova si l'article ja hi és dintre del blog.

En cas que no estigui dintre del blog l'afegirà, i en cas contrari actualitzarà l'article, donat que partim de la base que dos articles amb el mateix títol i escrits a la mateixa data (la data compren fins el segon en el que va publicat) són els mateixos.

D'aquesta manera ens evitem tenir articles repetits dintre del mateix blog i fem que si un article es modifica a través de *Quickr* i es vol tornar a importar, simplement actualitzarà el que ja hi era.

8.4.7 Classe *Quickr*

La classe *Quickr* és una classe nova dintre de l'aplicació, que correspon a la realització del component amb el mateix nom. Tal i com he explicat abans, *Quickr*, representa cadascuna de les comissions que es troben dintre de l'aplicació de *IBM* que porta aquest mateix nom.

Per accedir als articles que es publiquen a cadascuna d'aquestes comissions, el que fem és accedir al *RSS* que es generen. Per tant, haurem de tractar arxius *XML*

El major problema amb el que ens trobem a l'hora d'implementar és que el *RSS* al que tenim que accedir és privat i per tant, fa falta autenticació per poder-lo veure. És per això que farem ús de dues classes ja implementades, *Snoopy* i *cURL*, que ens permetran accedir al contingut de les pàgines que requereixin autenticació.

Per el procés d'importació d'articles, ens interessa agafar tot l'article i prepara-ho per enviar a *WordPress*. Amb això vol dir que tenim que fer els següents passos:

- Accedir primer al *RSS*.
- Per cada article accedir a la *URL* que el conté.
- Capturar tot el contingut *HTML* i, en el cas que tingui imatges, descarregar-nos les imatges
- Pujar les imatges al servidor de *WordPress*.
- Modificar el contingut de l'article per tal d'indicar-li quina serà la nova ubicació de les imatges de les que disposa.

Per realitzar tot el procés abans esmentat i per capturar els esdeveniments del sistema del cas d'ús he implementat els següents mètodes a la classe *Quickr*:

Sel_article_quickr

Pertany al diagrama de col·laboració del esdeveniment de sistema amb el mateix nom. S'encarrega de cercar l'article corresponent dintre del repositori d'articles del blog i afegir-lo al repositori d'articles que s'importaran.

Fi_importacio

Correspon a l'esdeveniment amb el mateix nom. S'encarrega de dir-li al blog de *WordPress* corresponent

que té que inserir els articles abans seleccionats.

Feed

Aquest és un mètode privat que fa ús de la classe *Snoopy* per fer l'autenticació de la pàgina que conté el RSS i, un cop accedit, genera un objecte *XML* que retorna.

EntradesFeed

Aquest mètode s'encarrega de tractar l'objecte *XML* abans creat i genera cadascun dels objectes *PostQ* que formen el repositori d'articles de la comissió *Quickr*.

Dataesp

S'encarrega de modificar la data obtinguda al *XML* i la posa en format europeu o en format de *WordPress*.

CapturaHTML

Accedeix a la direcció URL on es troba un article i captura el contingut HTML del mateix. Com es tracta també d'accedir a una zona privada, fem ús de la classe *Snoopy* per autenticar-se.

ModPost

Cerca al contingut de l'article si conté imatges. En cas que així sigui les descarrega, i modifica el contingut del codi HTML de l'article.

El resultat d'aquest procés és un article totalment preparat per importar a *WordPress*.

Searching

Mètode que cerca les etiquetes IMG que hi hagi a un article. En el cas que trobi una o més imatges, retorna un array amb la posició que ocupa al contingut de l'article, la ruta de la imatge i la direcció completa per poder accedir a la ella.

Salvaimatge

Aquest mètode s'encarrega de descarregar una imatge i pujar-la al servidor. Per fer aquest procés requerim tant de la classe *Snoopy* com de la classe *cURL*.

Per poder accedir a una imatge hi ha el problema que no demana autenticació, i si volem accedir sense haver iniciat sessió a *Quickr*, no ens permet veure la imatge.

Per resoldre aquest problema fem ús de la classe *cURL*. L'única solució que tenim per accedir a la imatge és iniciar sessió, per tant amb *cURL*, creem un accés remot a *Quickr* amb el que poder fer login sense necessitat d'anar a la pàgina web del portal *Quickr*.

Un cop aconseguim autenticar-nos tornem accedir a la imatge, la descarreguem i seleccionem el destí del servidor on volem que la guardi.

El resultat de tot aquest procés és el retorn de la direcció URL de la imatge al servidor *WordPress* per poder modificar el contingut de l'article e indicar-li la nova direcció de la imatge.

Replaceimg

Reemplaça una cadena de caràcters per una altre. En el nostre cas ens referim a les imatges que hem descarregat i guardat al servidor, modifiquem el contingut de l'article.

Formataccent

Modifica els caràcters del contingut de l'article per tal d'adaptar-lo a format HTML, per el cas dels accents.

8.5 Una altre solució. Exportació dels articles

Per tal de que hi hagi una associació completa entre les comissions de *WordPress* i les comissions de *Quickr*, s'ha decidit implementar una funcionalitat semblant a la importació d'articles, però des de l'altre banda.

Aquesta nova funcionalitat a de permetre fer l'exportació dels articles des de *Quickr* fins al blog *WordPress* pertinent.

Tal i com podem veure al model conceptual, tenim que una comissió de *Quickr* es correspon amb una o més d'una comissió de *WordPress*. Això és per el cas de les subcomissions, donat que a *Quickr* les comissions i subcomissions es troben dintre del mateix conjunt, mentre que a *WordPress*, les estem tractant com a blogs diferents. Per tant, normalment tindrem que l'associació *Blog – Quickr* normalment serà de 1 a 1 i només en alguns cassos especials serà diferent.

Com que el procés que hem fet servir per importar els articles ha sigut el d'accedir a un *RSS*, podem reaprofitar-lo per aquest cas també. Així doncs, no hem de crear un nou cas d'ús, donat que estarem fent pràcticament el mateix, l'únic que amb una interfície diferent.

Per tant, la creació d'aquesta nova solució es limita a la implementació d'un nou component a *Quickr*, en forma de portlet, que oferirà una interfície semblant a la que hem fet servir per la importació.

El procés serà el següent:

- **Selecció d'articles per exportar a *WordPress*.** La interfície que es mostra per poder seleccionar els articles és la mateixa que per la importació, donat que es recuperen els articles del mateix *RSS*.
- **Selecció del blog de *WordPress* al que es farà l'exportació.** Aquest blog, com abans he explicat està limitat, sent en al majoria de cassos un únic blog. Els blogs disponibles per enviar estan definits a la pròpia comissió *Quickr*.
- **Crida al cas d'ús *ImportarArticle*.** Un cop s'ha seleccionat tot el necessari es crida a *WordPress* directament per utilitzar el cas d'ús creat anteriorment, fent així que no hi hagi que tornar a implementar tot el procés de nou.

Per tant l'única implementació que es té que fer a *Quickr* és la referent a la interfície que permeti fer la selecció de tot el necessari.

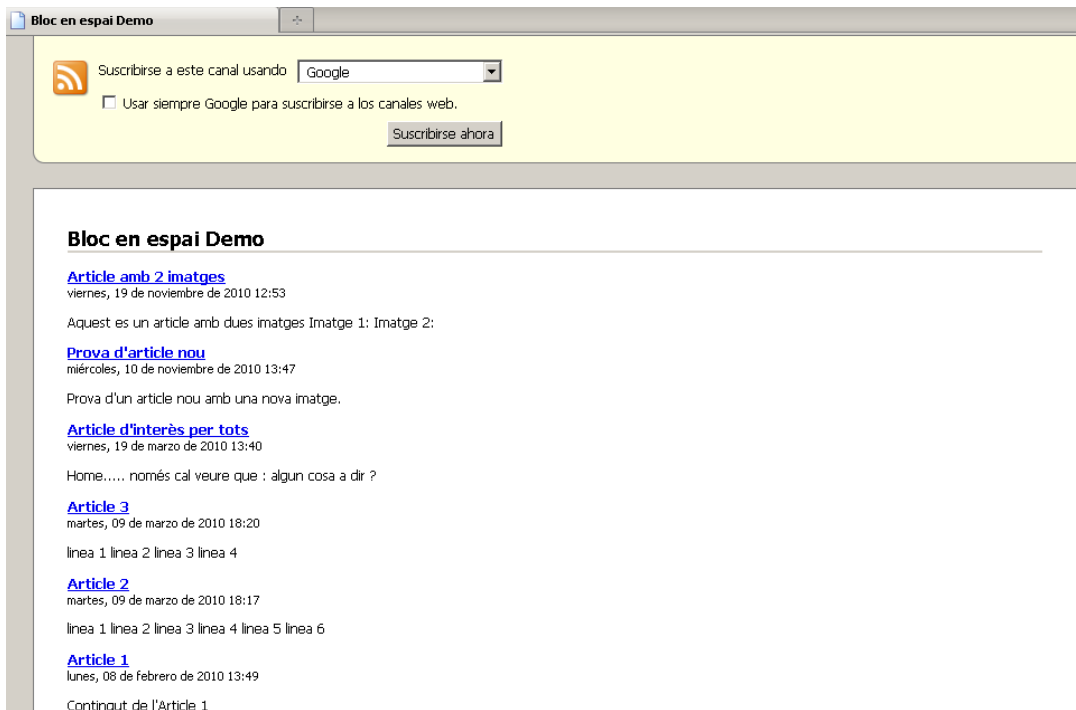
El procés d'implementació és una adaptació de la classe *Xuclablog* de *WordPress* a un portlet de *Quickr* en llenguatge *Java*.

Per realitzar la crida al plugin de *WordPress* ho farem creant un *iframe* ocult en el que es carregarà la URL del plugin i per el que es passaran les dades pertinents per paràmetre.

La repercussió del portlet al plugin que havíem implementat, resulta simplement en una nova opció a l'hora de fer l'acció del formulari. Ara el plugin *Quickr To WordPress* interpretarà a l'hora de cridar al cas d'ús, de si es tracta d'una importació d'articles o d'una exportació des de *Quickr*.

8.6 Interfícies del cas d'ús

8.6.1 Importació dels articles



Bloc en espai Demo

Suscribirse a este canal usando Usar siempre Google para suscribirse a los canales web.

Bloc en espai Demo

[Article amb 2 imatges](#)
viernes, 19 de noviembre de 2010 12:53

Aquest es un article amb dues imatges Imatge 1: Imatge 2:

[Prova d'article nou](#)
miércoles, 10 de noviembre de 2010 13:47

Prova d'un article nou amb una nova imatge.

[Article d'interès per tots](#)
viernes, 19 de marzo de 2010 13:40

Home..... només cal veure que : algun cosa a dir ?

[Article 3](#)
martes, 09 de marzo de 2010 18:20

linea 1 linea 2 linea 3 linea 4

[Article 2](#)
martes, 09 de marzo de 2010 18:17

linea 1 linea 2 linea 3 linea 4 linea 5 linea 6

[Article 1](#)
lunes, 08 de febrero de 2010 13:49

Contingut de l'Article 1

1



Selecciona l'origen dels blogs

Transferencia de posts de Quickr

Selecciona els articles que vulguis que s'agefeixin al blog

Els articles actuals a Quickr son els següents

Títol	Autor	Data	
Article amb 2 imatges	wpsadmin	19-11-2010	<input type="checkbox"/>
Prova d'article nou	wpsadmin	10-11-2010	<input type="checkbox"/>
Article d'interès per tots	wpsadmin	19-03-2010	<input type="checkbox"/>
Article 3	wpsadmin	09-03-2010	<input type="checkbox"/>
Article 2	wpsadmin	09-03-2010	<input type="checkbox"/>
Article 1	xavier.raurell	08-02-2010	<input type="checkbox"/>

Selecciona el destí del post

2

¹ Exemple de Feed a recuperar amb articles de Quickr

² Aspecte del plugin QuickrToWordPress per importar

8.6.2 Exportació dels articles

Enginyers Industrials de Catalunya

Comissió de Construcció Finalitzar sessió: wpsadmin wpsadmin

Volver a Inicio Ayuda

Comissió de Construcció Personalizar

Novetats Calendari Arxiu Documentació Membres Blog **Exportació**

Articles a exportar a WordPress

Títol	Autor	Data	
Article amb 2 imatges	wpsadmin	19-11-2010	<input type="radio"/>
Prova d'article nou	wpsadmin	10-11-2010	<input type="radio"/>
Article d'interès per tots	wpsadmin	19-03-2010	<input type="radio"/>
Article 3	wpsadmin	09-03-2010	<input type="radio"/>
Article 2	wpsadmin	09-03-2010	<input type="radio"/>
Article 1	xavier.raurell	08-02-2010	<input type="radio"/>

Accio

1

¹ Aspecte del plugin QuickrToWordpress per exportar.

9. Funcionalitats diverses implementades

A més de les funcionalitats que han estat implementades fins ara, el COEIC desitja personalitzar més l'eina *WordPress* per tal de limitar algunes funcionalitats i ampliar d'altres que permetin una integració completa amb tot el sistema que tenen.

Per tant he desenvolupat altres funcionalitats menors, que explicaré a continuació.

9.1 Inserció, eliminació i actualització d'usuaris

WordPress ofereix la opció de registrar usuaris mitjançant formularis que té implementats internament. Per fer ús d'aquesta opció es requereix activar abans aquesta opció al panell d'ajustaments del taulell.

De la mateixa manera, des de *WordPress* es pot tenir un control absolut dels usuaris, afegint informació personal, editant-la i eliminant els usuaris que es vulgui en cas que sigui administrador.

Tot i que *WordPress* dona aquestes facilitats, es volen desactivar, per tal de tenir un control d'usuaris més global.

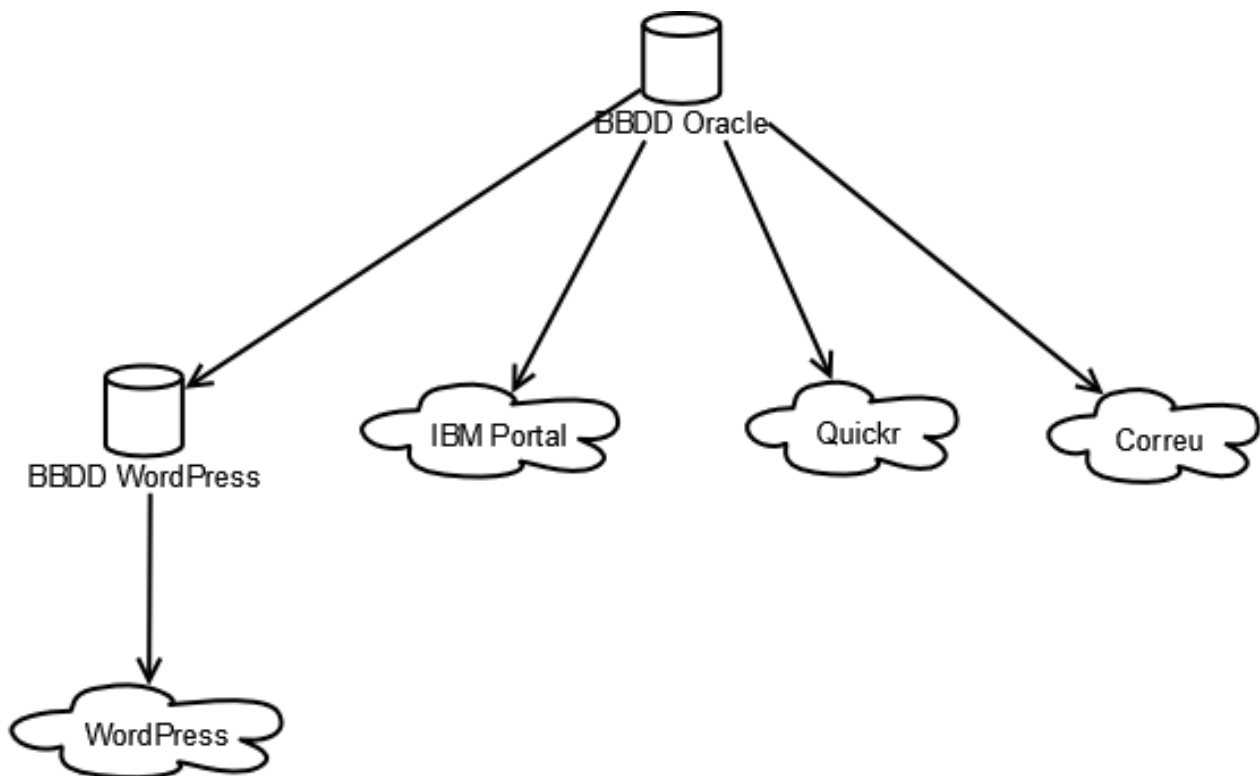
El motiu d'això és perquè el COEIC té actualment, i vol continuar tenint, la seva base de dades d'usuaris centralitzada. Tots els enginyers industrials col·legiats que es troben a la base de dades *Oracle* central, tenen el mateix nom d'usuari tant per accedir al portal d'enginyers, com per accedir a *Quickr*, com per accedir al seu correu. D'aquesta manera, si un enginyer industrial nou es dona d'alta queda enregistrat en tot el sistema.

Si es vol continuar mantenint aquesta situació, s'ha d'evitar que els usuaris es puguin registrar lliurement i tampoc han de poder modificar les seves dades, així, si volen canviar la contrasenya d'entrada, per exemple, quedarà modificada per tot el sistema. La restricció per part dels usuaris els obliga (tal i com succeeix actualment) a notificar al COEIC en el cas de voler modificar les seves dades.

Per aconseguir aquest objectiu he pres una sèrie de mesures:

- **Inhabilitació del registre d'usuaris.** Aquesta mesura ja hi és configurada per defecte a *WordPress*, encara que per assegurar-se hi ha que accedir com usuari superadministrador i, al taulell de superadministrador tenim que anar al menú d'opcions. Allà ens trobem l'opció *Paràmetres de registre*, on ha d'ésser seleccionada l'opció *El registre està desactivat*.
- **Inhabilitació del perfil d'usuari.** El taulell de *WordPress* incorpora un menú, *Usuari*, que permet accedir al perfil de l'usuari logat i modificar les seves dades personals. Com abans he explicat, això no ens convé, per tant hi ha que desactivar-lo. Per fer-ho, hi ha que accedir al nucli de *WordPress*, concretament a l'arxiu *menu.php* i deshabilitar la opció del menú d'usuari. D'aquesta manera no mostra el menú d'usuaris i per tant, no permet fer modificacions.
- **Implementació d'una solució.** Per poder registrar usuaris, actualitzar-los o esborrar-los, he decidit implementar unes funcions que s'encarreguin de fer-ho automàticament. Les funcions s'encarreguen de recuperar les dades que s'han enviat prèviament mitjançant un *trigger* i executen les accions necessàries per dur a terme la feina.
 - o **Inserció d'un usuari.** Per fer la inserció d'un usuari rebem el DNI (que serà l'identificador a la base de dades de *WordPress*), el nom d'usuari, el password encriptat i la direcció de correu. Un cop fet això es crida a la funció interna de *WordPress* *wp_insert_user* que s'encarrega d'introduir l'usuari.
 - o **Actualització d'un usuari.** L'actualització d'un usuari és semblant a la inserció. Rebem el login i el password i/o el nou correu i actualitzem l'usuari mitjançant la funció *wp_update_user*.
 - o **Eliminació d'un usuari.** Per esborrar un usuari només hem de rebre el seu login, llavors recuperem el seu identificador i cridem a la funció *wp_delete_user*.

Fent això aconseguim l'objectiu de mantenir els mateixos usuaris arreu de tot el sistema.



1

9.2 Fitxa d'usuaris

El COEIC té una fitxa per cadascun dels usuaris que té a la seva base de dades. Com a conseqüència d'això es vol fer que al accedir a la fitxa de l'usuari, si aquest a escrit articles a la blogosfera, es mostri una llista dels articles que ha escrit amb el títol de l'article, la data, el blog en el que s'ha escrit i l'enllaç directe a l'article.

Per poder recuperar els articles que ha escrit un usuari a la blogosfera, he decidit implementar un nou arxiu (*articles.php*) dintre del cos de *WordPress* que retornarà tots els articles que ha escrit aquest usuari.

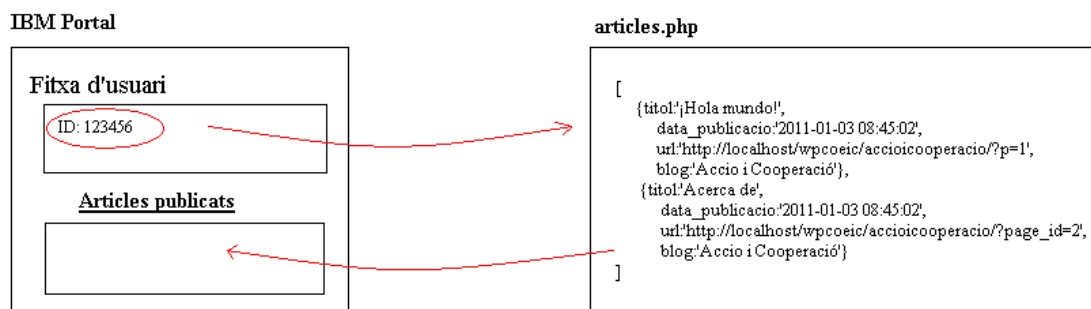
Per fer-ho la pàgina creada rebrà l'identificador de l'usuari del qual es vol consultar la informació i es cercaran tots els articles que ha escrit en tota la blogosfera.

Com que la fitxa d'usuaris es troba dintre del portal de IBM que té el COEIC, hi ha que crear una comunicació entre totes dues eines.

La connexió que he decidit implementar ha sigut mitjançant la rebuda de paràmetres per mètode GET, recuperant l'identificador de l'usuari. Per retornar el resultat de la cerca cap al portal IBM, he decidit fer el retorn via *JSON*, perquè així es puguin recuperar les dades des del portal.

La fitxa d'usuari no s'inclou, donat que format part del portal web del COEIC i no ha estat desenvolupada per aquest projecte.

¹ Representació del sistema d'emmagatzematge d'usuaris del COEIC



1

9.3 Restricció de seguretat per la fase de proves

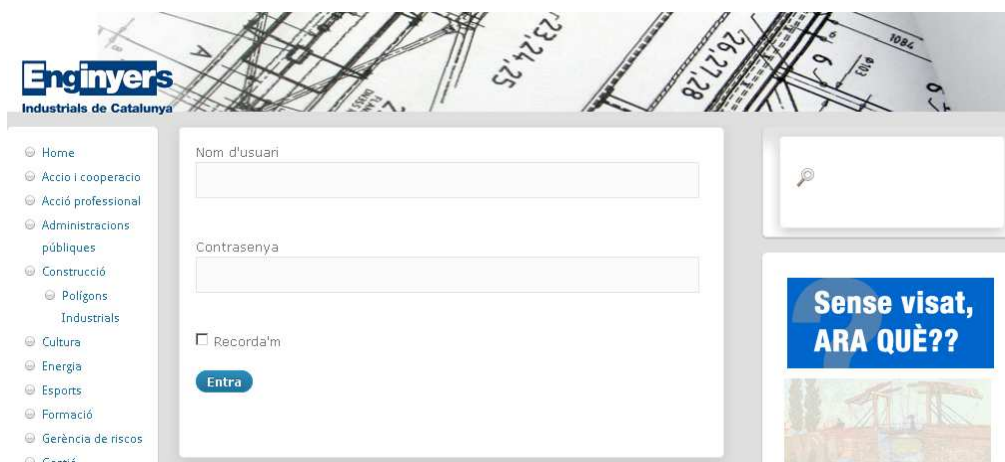
Inicialment, en una fase de proves de funcionament de l'aplicació, el COEIC vol mantenir un accés tancat a la blogosfera.

Aquest accés tancat consistirà en que només els usuaris registrats poden veure els articles publicats i fer comentaris sobre ells. El que podrà veure qualsevol persona que entri a la blogosfera seran les pàgines principals de les diferents comissions, on es mostren un petit resum dels articles que hi ha publicats.

Per aplicar les restriccions de seguretat, el que he fet ha sigut modificar la pàgina que mostra els articles del tema que tenim aplicat (*single.php*).

D'aquesta manera, cada cop que s'intenti accedir a un article, el sistema comprovarà si l'usuari està logat o no, i, en cas de que no estigui, oferirà l'opció d'iniciar sessió. Així si intenten accedir usuaris que no són membres del COEIC no podran veure el contingut dels articles.

Aquesta mesura de seguretat està indicada per tal de controlar el comportament del usuaris que fan ús de la blogosfera. Així es vol evitar que articles d'opinió personal que puguin ser contradictoris a l'opinió del col·lectiu del COEIC surtin a l'exterior i donin una imatge poc adequada del col·legi.



2

¹ Funcionament de la connexió entre la fitxa d'usuaris i WordPress

² Mostra d'intent d'accés a un article sense iniciar sessió

9.4 Personalització del tema

Com abans he explicat, el tema seleccionat per donar un aspecte agradable a la blogosfera, *BluePrint*, no es més que una carcassa que acabariem modificant per adaptar-lo a les nostres necessitats.

Per tant, s'han realitzats canvis al contingut, alguns ja explicats, que són els següents:

- **Capçalera:** la capçalera escollida és dinàmica i sincronitzada amb la pàgina principal del col·legi (*www.eic.cat*). Amb això el que vull dir, és que es tracta no d'una imatge incrustada, sinó d'una direcció URL, que conté la imatge actual que es fa servir com capçalera a la web del col·legi.
- **Barra lateral esquerra:** com he explicat anteriorment la barra lateral esquerra està destinada única i exclusivament a mostrar el menú jeràrquic que s'ha implementat anteriorment i que ens serveix per anar d'una comissió a una altre.
- **Barra lateral dreta:** la barra lateral dreta, està formada per una sèrie de widgets que ofereixen diverses opcions.
 - o **Cerca:** primer de tot tenim un widget de cerca per quan sigui necessari.
 - o **Publicitat:** hi ha configurat, a través de les opcions que ofereix el tema, una sèrie de banners de publicitat que informen als usuaris sobre temes del col·legi.
 - o **Twitter:** accés directe al compte de *Twitter*, del col·legi. Aquest widget ofereix les últimes notícies publicades a *Twitter* per part del COEIC.
 - o **Opcions de login:** aquest widget és personalitzat. Només es mostra quan un usuari ha iniciat sessió, llavors fa una distinció entre usuaris.
 - **Subscriptors:** en cas de ser-hi dintre d'un blog o article, mostra l'opció de subscriure's als RSS de les entrades del blog o del comentaris dels articles i l'opció de desconnectar-se. En cas de ser-hi dintre de la *Home*, només mostrarà l'opció de desconnectar-se.
 - **Resta d'usuaris:** en cas de ser-hi dintre d'un blog o d'un article mostra les mateixes opcions que als subscriptors amb l'afegit de l'opció de poder anar al seu taulell. En el cas de ser-hi a la *Home* pot anar al seu taulell (només administradors) i desconnectar-se.
- **Peu de pàgina:** ha estat modificat perquè sigui igual al de *www.eic.cat*, per així oferir informació personal sobre el col·legi.
- **Índex:** tal com ja he explicat anteriorment, la pàgina principal també ha estat modificada per tal de que es mostri diferent en cas de que estiguem a la *Home*, o a una comissió qualsevol.
- **Article:** com he explicat a l'apartat anterior, la pàgina del contingut d'un article ha estat modificada per tal que només pugin accedir els usuaris que han iniciat sessió.
- **Slider:** ha estat modificat el comportament per tal d'adaptar-lo a la pàgina principal.

- Home
- Acció i cooperació
- Acció professional
- Administracions públiques
- Construcció
 - Pòligons Industrials
- Cultura
- Energia
- Esports
- Formació
- Gerència de riscos
- Gestió empresarial
- Jubilats
- Medi ambient
- Mobilitat i transport
 - Engineersmotorsp
- Noves iniciatives
- Packaging
- Relacions amb les escoles d'enginyeria i les universitats
- Seguretat, prevenció i protecció
- TIC
- Urbanisme
- Club Bombeta



**Sense visat,
ARA QUÈ??**

**Agenda d'activitats
TARDOR HIVERN**
OCTUBRE, NOVEMBRE, DESEMBRE, GENER

**XXX
Cicle de Música dels
Engineers Industrials
de Catalunya**



Opcions de login

- Taulell
- Surt

**Engineers
Enginyeria**

Els reis arriben al COEIC!!!
<http://engineersclubbomb...>
yesterday · reply

Felic 2011 engineers!!
6 days ago · reply

Declaració per a la millora de les administracions i funció pública
<http://t.co/Uz5r1zX> via @AddThis
9 days ago · reply

S'obre la convocatòria de projectes emprenedors per presentar-se al concurs Fem Indústria: <http://www.femindustria...>
9 days ago · reply

twitter Join the conversation

¹ Aspecte de la Home amb un administrador logat i sense articles introduïts

10. Manual d'usuari

Per tal de poder fer un ús correcte de l'aplicació i no hi hagi cap problema per part de cap usuari, explicaré a través d'aquest manual d'usuari com s'ha d'instal·lar en una màquina local i com fer servir la blogosfera per *WordPress*.

Com que a *WordPress* hi ha diferents tipus d'usuari, les opcions que es podran fer servir varien segons el nivell que tinguin, per tant, partiré del manual per l'ús d'un usuari tipus superadministrador, que té control total de l'aplicació.

A mida surtin opcions que puguin usar altres usuaris, les introduiré en el manual del tipus d'usuari corresponent.

Tant aquest manual, com tot el projecte, estan fets per tal que el seu funcionament sigui perfecte a les versions de *WordPress*, 3.0.3 i 3.0.4.

10.1 Requeriments del sistema

Perquè es pugui fer ús de la blogosfera de *WordPress* en un ordinador en local, cal tenir instal·lats prèviament *MySQL*, *Apache* i *PHP* per tal de poder fer ús dels llenguatges de programació *SQL* i *PHP*.

Per el sistema operatiu *Windows* podem fer ús d'alguna eina *WAMP* (*Windows*, *Apache*, *MySQL*, *PHP*), que englobi tots dos programes, com es el cas de *EasyPHP* (<http://www.easyphp.org>).

En el cas de *Linux* ens podem fer amb alguna eina *LAMP* (*Linux*, *Apache*, *MySQL*, *PHP*) com pot ser el cas de *Xampp* (<http://www.apachefriends.org/es/xampp.html>).

També hi ha l'opció d'instal·lar per separat les diferents aplicacions. Els requisits mínims que demana *WordPress* és la versió de *MySQL 4.0* i la versió de *PHP 4.2*, a més de tenir un navegador (mínim *Internet Explorer 6*).

10.2 Manual d'instal·lació de *WordPress*

10.2.1 Instal·lació d'un blog normal

Un cop hem tenim instal·lat i configurat els requeriments bàsics hem de procedir a descarregar-nos *WordPress* des de la pàgina oficial (<http://www.wordpress.org>) o fer ús de la carpeta *WordPress* dels arxius adjunts al projecte.

En cas de descarregar, al finalitzar la descàrrega ens trobem amb un arxiu comprimit que conté tota l'eina *WordPress*.

Per procedir a la instal·lació el descomprimim a la carpeta *www* de *Apache*.

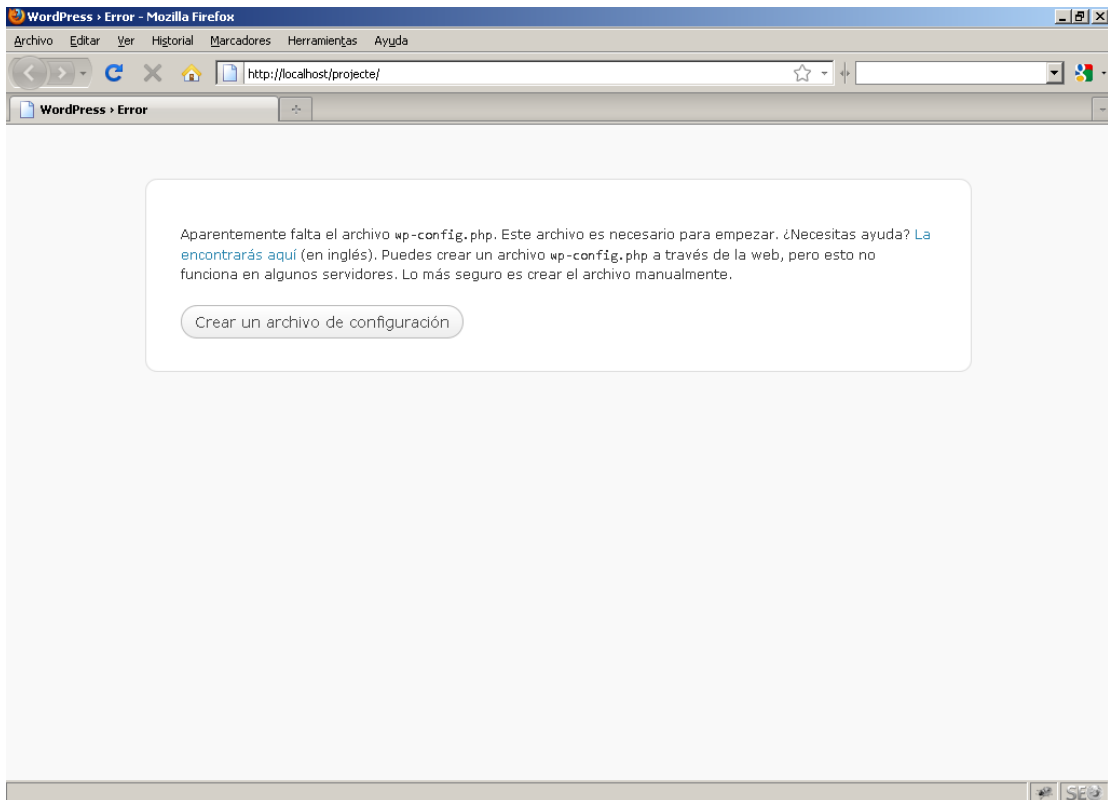
Abans de començar a instal·lar, hem de crear una base de dades nova dintre del servidor *MySQL*, per tal de que pugui guardar les taules corresponents de *WordPress*.

Com a exemple usarem una base de dades anomenada *projecte* i el directori amb *WordPress* rebrà el mateix nom.

Ara que ja tenim creada la base de dades ja podem començar a instal·lar *WordPress*. Per fer-ho, obrim una finestra del navegador i accedim a la direcció on tenim el directori de l'eina. <http://localhost/projecte>.

Al entrar, en sortirà la finestra següent, indicant-nos que hem de crear l'arxiu de configuració.

Premem el botó que hi ha i continuem la instal·lació.



A continuació ens mostra una pantalla informativa que ens explica la informació que haurem d'afegir sobre el nostre blog. Aquesta informació és:

- **Nom de la base de dades.** En el nostre cas *projete*.
- **Nom d'usuari de la base de dades.** Usuari de *MySQL*. Si no hem creat cap, podem fer ús de l'usuari *root*. En tot cas ha de ser un usuari que tingui els privilegis suficients com per crear taules i fer modificacions sobre elles.
- **Contrasenya de la base de dades.** Contrasenya de l'usuari que hem seleccionat anteriorment.
- **Host de la base de dades.** Com que estem treballant en local, serà *localhost*.
- **Prefixe de la taula.** Aquest serà el prefixa que tindran les taules que formen l'estructura interna de *WordPress*. En el nostre cas posarem el prefixa per defecte, *wp*.

També ens informa que en cas que falli la instal·lació automàtica, podem fer ús de la instal·lació manual, que consisteix en modificar l'arxiu *wp-config-sample.php*, ficar en ell les dades corresponents i, finalment, reanomenar l'arxiu a *wp-config.php*.

Com volem instal·lar automàticament, procedim a clicar el botó i omplim el formulari següent amb les dades que he explicat anteriorment.

¹ Pàgina inicial de instal·lació de *WordPress*



A continuació deberàs introduir els detalls de connexió amb la teua base de dades. Si no estàs segur de quins són contacta amb el teu proveïdor d'hosting.

Nombre de la base de dades	<input type="text" value="projecte"/>	El nom de la base de dades en la que vols que s'executi WP.
Nombre de usuari	<input type="text" value="root"/>	Tu nom de usuari de MySQL.
Contrasena	<input type="text"/>	...i la contrasenya de MySQL.
Host de la base de dades	<input type="text" value="localhost"/>	Si no funciona <code>localhost</code> tindràs que contactar amb el teu proveïdor d'hosting per saber qual és.
Prefiix de taula	<input type="text" value="wp_"/>	Si vols executar diverses instal·lacions de WordPress en una sola base de dades canvia això.

Enviar

Un cop omplert el formulari premem el botó d'enviar.

Si no hi ha hagut cap error ens portarà a una finestra que ens donarà pas a la instal·lació. Això ens portarà a un nou formulari on hem d'omplir informació corresponent al blog que estem creant.

- **Títol del blog.** Nom que tindrà el nostre blog. *Desenvolupament d'una blogosfera.*
- **Nom d'usuari.** Aquest primer usuari serà el superadministrador de l'aplicació. Per defecte, *admin.*
- **Contrasena.** Clau d'accés de l'administrador.
- **Correu electrònic.** Adreça de correu de l'usuari que ens servirà per enviar-nos la contrasenya en cas que la oblidem.

¹ Formulari de configuració de *WordPress*



Bienvenido

¡Bienvenido al famoso proceso de instalación de WordPress de cinco minutos! Tal vez quieras leer tranquilamente la [Documentación del archivo Léeme](#). En caso contrario, rellena los datos más abajo y en seguida estarás utilizando la plataforma de publicación personal más potente y extensible del mundo.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	<input type="text" value="Desenvolupament d'una blogo"/>
Nombre de usuario	<input type="text" value="admin"/> <small>Los nombres de usuario sólo pueden tener caracteres alfanuméricos, espacios, guiones bajos, guiones, puntos y el símbolo @.</small>
Password, dos veces	<input type="password" value="....."/> <input type="password" value="....."/> <small>Se generará un password automático si lo dejas en blanco.</small> Fuerte <small>Tu contraseña debe tener al menos siete caracteres. Para que tu contraseña sea segura, usa mayúsculas, minúsculas, números y símbolos como ! " ? \$ % ^ &).</small>
Tu correo electrónico	<input type="text" value="david@david.com"/> <small>Comprueba bien tu dirección de correo electrónico antes de continuar.</small>
<input checked="" type="checkbox"/> Permitir que mi sitio aparezca en motores de búsqueda como Google y Technorati.	
<input type="button" value="Instalar WordPress"/>	

1

Un cop omplert aquest formulari, si no hi ha hagut cap problema a la pantalla següent ens informará que ja s'ha instal·lat *WordPress* i que ja podem accedir a ell.

10.2.2 Activació del multisite

Ara que ja tenim *WordPress* instal·lat hem de procedir a activar el mode *multisite* per poder crear la blogosfera. Per fer-ho hem de seguir una sèrie de passos.

Primer de tot editarem l'arxiu de configuració de la instal·lació de *WordPress* actual, l'arxiu *wp-config.php*.

L'edició de l'arxiu consisteix en escriure una sentència a sota de la línia que diu */* ¡Eso es todo, deja de editar! Feliz blogging */*.

La sentència és: *define ('WP_ALLOW_MULTISITE', true);*

Amb això li estem indicant que l'opció multisite queda activada. Guardem i tornem al blog, on haurem de tancar la sessió que tinguem activa i tornar a entrar a iniciar sessió.

¹ Formulari de configuració del blog.

Ara tindrem una nova opció al menú *Eines*, l'opció *Xarxa*. Aquesta serà la que ens permetrà activar l'opció *multisite*.



1

Com estem fent ús d'una instal·lació en local de *WordPress*, només podem fer una activació del *multisite* en subdirectoris. En cas que féssim la instal·lació en un servidor, tindríem l'opció d'activar el *multisite* en forma de subdominis.

Per continuar amb l'activació del *multisite*, fem clic al botó d'instal·lació.

La següent finestra ens indica els passos que hem de seguir i les modificacions que tenim que fer als arxius per tal de fer l'activació. Aquestes són les següents.

- **Crear un directori anomenat *blogs.dir* dintre del directori *wp-content*.** Això ens permetrà emmagatzemar els arxius que es guardin quan inserim un nou objecte (imatge, video, so) a un article. Dintre d'aquest directori s'organitzarà per l'identificador del blog i dintre de cada subdirector, per la data en que es va crear l'objecte.
- **Modificar l'arxiu *wp-config.php*.** Sota de l'edició anterior que havíem fet, haurem d'introduir el codi que ens proporciona.
- **Modificar l'arxiu *.htaccess*.** L'arxiu *.htaccess* ens serveix per configurar regles d'escriptura que *Apache* interpreta per tal de redireccionar URLs en el nostres cas. Llavors haurem de modificar el nostre *.htaccess* amb el codi que ens proporciona. En cas que no tinguem aquest arxiu, el creem e inserim aquest codi.

Per últim, un cop haguem guardat tots els canvis només ens quedarà tornar a tancar sessió i iniciar-la de nou per tal de que tot quedi activat.

¹ Opció *Xarxa* del menú *Eines*

Desenvolupament d'una blogosfera Nueva entrada Hola, admin | Cerrar

Crear una red de sitios de WordPress

Activando la red

Completa los siguientes pasos para activar la función de creación de red de sitios.

Atención: Recomendamos realizar una copia de seguridad de tus archivos `wp-config.php` y `.htaccess` existentes.

1. Crea el directorio `blogs.dir` en `C:\EasyPHP-5.3.2i\www\projecte\wp-content`. Este directorio se usa para almacenar los archivos subidos por los sitios adicionales y el servidor web, debe poder escribir en él.
2. Añade lo siguiente a tu archivo `wp-config.php` en `C:\EasyPHP-5.3.2i\www\projecte\` **sobre** la línea que dice `/* ¡Eso es todo! ¡Hay que editar nada más! ¡Feliz blogging. */`:

```

define( 'MULTISITE', true );
define( 'SUBDOMAIN_INSTALL', false );
$base = '/projecte/';
define( 'DOMAIN_CURRENT_SITE', 'localhost' );
define( 'PATH_CURRENT_SITE', '/projecte/' );
define( 'SITE_ID_CURRENT_SITE', 1 );
define( 'BLOG_ID_CURRENT_SITE', 1 );

```
3. Añade lo siguiente a tu archivo `.htaccess` en `C:\EasyPHP-5.3.2i\www\projecte\`, reemplazando las reglas existentes de WordPress:

```

RewriteEngine On
RewriteBase /projecte/
RewriteRule ^index\.php$ - [L]

# uploaded files
RewriteRule ^([_0-9a-zA-Z-]+)/?files/(.*) wp-includes/ms-files.php?file=$2 [L]

# add a trailing slash to /wp-admin
RewriteRule ^([_0-9a-zA-Z-]+)/?wp-admin$ $1wp-admin/ [R=301,L]

RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(wp-(content|admin|includes).*) $2 [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(.*)\.(php)$ $2 [L]
RewriteRule . index.php [L]

```

Una vez que completes estos pasos tu red estará activa y configurada. Deberás acceder de nuevo. [Acceder](#)

Un cop fet això, si no hi ha hagut cap problema, tindrem el *multisite* activat.

10.3 Instal·lacions posteriors a l'activació del *multisite*

Per poder utilitzar tot el contingut de la blogosfera tal i com ha estat explicat hi ha que fer la instal·lació (com es *WordPress*, copiar els arxius) dels següents afegits. Els afegits es proporcionen en una carpeta a part anomenada *Afegits*.

- **Tema.** Dintre de la carpeta *Tema* tenim el tema corresponent que tenim que utilitzar, comprimit. Aquest arxiu es correspon amb el tema *Blueprint per COEIC*. Hi ha que descomprimir aquest arxiu dintre de la carpeta `wp-content/themes`.
- **Plugins.** Els plugins els trobem dintre de la carpeta `plugins`. Es corresponen amb els plugins *COEIC Home*, *Jerarquia de blogs* i *QuickrToWordPress*. Cadascun dels arxius que es troben es tenen que descomprimir a la carpeta `wp-content/plugins`.
- **Altres.** Dintre de la carpeta *Altres* ens trobem amb dos arxius que corresponen al control d'usuaris via *trigger* i a la recuperació d'articles via *Json*. Tots dos arxius hi ha que posar-los al directori arrel

¹ Configuració del *multisite* de *WordPress*

de la instal·lació de *WordPress*.

- **Modificació del nucli de *WordPress*.** Per últim haurem de fer una petita modificació del nucli de *WordPress* per tal d'enfortir la seguretat de la blogosfera. Hem de copiar l'arxiu adjunt *menu.php* contingut a la carpeta *Altres* i reemplaçar-lo per el que hi ha actualment a la carpeta *wp-admin* de la instal·lació de *WordPress*.

Es recorda que està comprovat per la versió 3.0.3 i 3.0.4 de *WordPress*, no s'assegura el correcte funcionament per versions posteriors.

10.4 Accés a l'aplicació

Si hem seguit els passos anteriors ara mateix ja tindrem un multiblog, però amb un únic blog.

Per accedir a la nostra blogosfera accedirem a través de la plana principal. Seguint l'exemple explicat tota l'estona, tenim que anar a <http://localhost/projete> per accedir a la blogosfera.

Tal i com ho tenim ara mateix per accedir a la part d'administració, al taulell, que és la part que ens interessa realment podem fer-ho de dues maneres:

- **Accés per *widget*.** En cas de tenir configurat un *widget* d'accés (per defecte es troba activat), tenim l'opció d'accedir directament per aquí
- **Accés directe.** En qualsevol dels casos sempre podrem entrar via URL amb la següent direcció. <http://localhost/projete/wp-admin>.

Meta

- [Acceder](#)
- [RSS de las entradas](#)
- [RSS de los comentarios](#)
- [WordPress.org](#)

1

Opciones de login

- Taulell
- Surt

2

10.5 Manual del superadministrador

Un cop entrem al taulell, en el cas que siguem super administradors ens trobem amb un menú dedicat a l'administració dels blogs.



3

¹ Widget Meta del tema per defecte de *WordPress*.

² Widget Meta del tema personalitzat per el COEIC.

³ Menú de superadministrador

10.5.1 Cercar un lloc o un usuari

El menú *Administrar* ens permet fer la cerca d'un blog que tinguem creat i d'un usuari que tinguem introduït. A més, ens permet crear tant un blog nou, com un usuari nou.



10.5.2 Veure un llistat dels blogs

A través del menú de llocs podem veure els blogs que tenim creats. Per cada blog podem veure:

- **Ruta d'accés al blog.** URL directe al blog.
- **Darrera actualització.** Últim cop que es va escriure un article, una plana...
- **Registat.** Data en la que va ésser creat el blog.
- **Usuaris.** Usuaris que conformen aquest blog.

A més, sobre cadascun dels blogs podem:

- **Editar les opcions del blog.** Permet l'edició de totes les característiques del blog.
- **Anar al taulell.** Podem canviar el blog actiu per anar a un altre.
- **Visita del blog.** Accés a la part visual del blog.

I per tots els blogs menys el principal:

- **Desactivar.** Desactivació del blog per tal que no s'utilitzi.
- **Esborrar.** Eliminació del blog.



¹ Opció Administrar del menú de superadministrador.

10.5.3 Crear un blog nou

A través del formulari que disposem al menú *Llocs* podem crear nous blogs. Per crear-los necessitem:

- **Direcció del blog.** URL d'accés al blog.
- **Títol del blog.** Nom que tindrà.
- **Correu electrònic de l'administrador.** Correu de la persona que administrerà el blog. Podem posar el correu d'un usuari existent, amb lo qual el farem administrador del blog, o podem ficar un de nou, llavors crearem un nou usuari amb el nom del blog.

Per continuar amb l'exemple crearem un parell de blogs més, administrats per el mateix superadmin. Aquest blogs seran *Blog1* i *Blog2*.

Añadir sitio

Dirección del sitio

Solo se recomiendan los caracteres a-z y 0-9.

Título del sitio

Correo electrónico del administrador

Se creará un usuario nuevo si el correo electrónico no está en la base de datos
El nombre de usuario y la contraseña se enviarán a este correo electrónico.

2

10.5.4 Creació d'usuaris

De manera predeterminada tenim desactivat el registre d'usuaris, però podem crear-los com superadministrador.

Els usuaris els podem crear de dues maneres:

- **Menú d'usuaris.** Dintre del menú d'usuaris de superadministrador ens trobem, tal i com ho fèiem amb els blogs, amb un llistat dels usuaris que tenim actualment (els quals els podem editar) o amb un formulari d'adició d'usuaris, que ens permetrà afegir un nou usuari. Per fer-ho necessitem.
 - o **Nom d'usuari.** Nom amb el que l'usuari farà login.
 - o **Direcció de correu electrònic.** Correu de l'usuari, la contrasenya se li enviarà a aquesta direcció.
- **URL.** Aquesta opció ha estat implementada per ser usada per triggers. Mitjançant l'accés directe a <http://localhost/usuaris.php?accio=insert&login=nom&pass=password&email=mail> podem insertar un nou usuari. Per fer-ho necessitem.

¹ Menú Llocs, llista de blogs

² Creació d'un nou blog

- **Nom d'usuari.** Usuari amb el que farem login. Hi ha que substituir *nom* per el nom d'usuari.
- **Contrasenya.** Contrasenya de l'usuari encriptada en base64. La substituïm per *password*.
- **Correu electrònic.** Direcció de correu de l'usuari. Es substitueix per *mail*.

Usuarios Añadir nuevo

Buscar usuarios

Acciones en lote

<input type="checkbox"/>	ID	Nombre de usuario	Nombre	Correo electrónico	Registrado	Sitios
<input type="checkbox"/>	1	admin - Super admin	admin	david@david.com	10/01/2011	localhost/projecte/ localhost/projecte/blog1/ localhost/projecte/blog2/

Acciones en lote

Añadir usuario

Nombre de usuario

Correo electrónico

Se enviarán el nombre de usuario y la contraseña a esta dirección de correo electrónico.

Per prosseguir crearem els cinc usuaris següents: *Administrador, Editor, Autor, Col·laborador, Subscriptor*.

10.5.5 Activar / Desactivar temes

Podem decidir desactivar temes que tinguem per tal que cap blog en pugui fer ús d'ells. Per defecte, tots els temes hi són activats.

Per poder executar aquesta opció tenim que anar al menú *Temes*. Com que no ens fa falta el tema per defecte (*Twenty Ten*) el desactivarem i en canvi activarem el tema del *COEIC*.

¹ Creació d'usuaris per menú

Temas de la red

Los temas deben activarse en tu red antes que estén disponibles para cada sitio.

Aplicar cambios

Activar	Tema	Versión	Descripción
<input checked="" type="radio"/> Sí <input type="radio"/> No	COEIC	2.0	
<input type="radio"/> Sí <input checked="" type="radio"/> No	Twenty Ten	1.1	The 2010 theme for WordPress is stylish, customizable, simple, and readable — make it yours with a custom menu, header image, and background. Twenty Ten supports six widgetized areas (two in the sidebar, four in the footer) and featured images (thumbnails for gallery posts and custom header images for posts and pages). It includes stylesheets for print and the admin Visual Editor, special styles for posts in the “Asides” and “Gallery” categories, and has an optional one-column page template that removes the sidebar.

Aplicar cambios

Total

Temas instalados: 2
Temas disponibles: 1

1

10.5.6 Modificar les opcions de la blogosfera

Podem canviar diverses opcions que afecten a tota la xarxa de blogs. El menú *Opcions* del superadministrador ens permet accedir a elles.

Les opcions que més ens pot interessarà variar són:

- **Nom de la xarxa.** Podem canviar el nom de la xarxa de blogs.
- **Correu electrònic del superadministrador.** A aquesta direcció serà a la que vagin destinats tots els correus de verificació i administració.
- **Perfil d'usuari per defecte.** Això ens servirà per definir de quin tipus d'usuari seran els nous usuaris. Per defecte és *Subscriber*.
- **Permetre nous registres.** Aquí podem definir si volem que es puguin crear nous usuaris i si aquests poden tenir privilegis per crear nous blogs. Per defecte està desactivat.
- **Botons de pujada d'arxius.** Haurem de modificar els tipus d'arxiu que volem deixar pujar en els articles. Per defecte no hi ha cap tipus activat. És aconsellable activar la pujada d'imatges, obligatori dintre del tema *COEIC* per al seu correcte funcionament.
- **Espai per pujades del blog.** Espai màxim que destinem als arxius pujats per als usuaris.
- **Mida màxima dels arxius.** Mida màxima dels arxius que es pugen als articles.
- **Idioma predeterminat.** Definim l'idioma que tindran de forma predeterminada els blogs. Per defecte es anglès, així que hi ha que variar-ho.

¹ Menú d'activació de temes.

10.5.7 Assignació d'usuaris i rols a un blog

Per fer l'assignació d'usuaris a un blog, tenim que anar al menú *Llocs / Sitios* i, en el blog en el que volem afegir els usuaris, fer clic en l'opció *Editar*.

Un cop dintre de les opcions del blog ens trobem amb l'opció *Afegir usuari nou*, on, posant el nom d'usuari, podem afegir-lo al blog corresponent, amb un rol d'usuari determinat.

Un usuari es pot trobar a més d'un blog i cadascun dels blogs pot tenir un rol diferent.

Per exemple, afegirem els usuaris *Administrador*, *Autor* i *Colaborador* al blog *Blog1* i els usuaris *Editor* i *Subscriptor* al blog *Blog2*, amb els rols corresponents als seus noms.

The image shows two sections of a web interface for managing site users. The top section, titled 'Usuarios del sitio', contains a table with three columns: 'Usuarios', 'Perfil', and 'Contraseña'. The table lists four users: 'admin' (N/A profile, N/A password), 'administrador' (Administrador profile, empty password), 'colaborador' (Colaborador profile, empty password), and 'autor' (Autor profile, empty password). Below the table is an 'Actualizar opciones' button. The bottom section, titled 'Añadir usuario nuevo', contains instructions: 'Introduce el nombre de usuario de un usuario existente y haz clic en "Actualizar opciones" para añadirlo.' Below this are two input fields: 'Acceso de usuario:' (empty) and 'Perfil:' (dropdown menu with 'Suscriptor' selected). Below these fields is another 'Actualizar opciones' button.

Usuarios	Perfil	Contraseña
admin	N/A	N/A
administrador	Administrador	<input type="text"/>
colaborador	Colaborador	<input type="text"/>
autor	Autor	<input type="text"/>

Actualizar opciones

Añadir usuario nuevo

Introduce el nombre de usuario de un usuario existente y haz clic en "Actualizar opciones" para añadirlo.

Acceso de usuario:

Perfil:

Actualizar opciones

¹ Menú d'afegir usuaris a un blog.

10.5.8 Esborrar un usuari d'un blog

Igual que podem afegir usuaris a un blog, tenim l'opció d'esborrar-los d'ells. Per esborrar-los tenim dues opcions:

- **Opcions del blog.** Igual que per afegir usuaris, ens permet també esborrar-los seleccionant-los de la llista d'usuaris que ens dóna i fent clic a *Actualitzar opcions*.
- **Menú usuaris del blog.** A cada blog també tenim un menú d'usuaris que ens ofereix una llista dels usuaris que estan adherits al blog. Per cada usuari tenim l'opció editar i esborrar. Aquesta funció també la poden fer servir els administradors del blog.

Per exemple, esborrarem del blog principal els usuaris *Administrador*, *Editor*, *Autor* i *Colaborador*.

Usuarios

Todos (6) | Administrador (1) | Suscriptor (5)

Acciones en lote [v] Aplicar Cambiar perfil a... [v] Cambiar

<input type="checkbox"/>	Nombre de usuario	Nombre	Correo electrónico	Perfil	Entradas
<input type="checkbox"/>	admin		david@david.com	Administrador	1
<input checked="" type="checkbox"/>	administrador		administrador@aaaa.aaa	Suscriptor	0
<input type="checkbox"/>	autor		autor@aaa.aaa	Suscriptor	0
<input type="checkbox"/>	colaborador		colaborador@aaaaa.aaa	Suscriptor	0
<input type="checkbox"/>	editor		editor@aaaaa.aaa	Suscriptor	0
<input type="checkbox"/>	suscriptor		suscriptor@aaa.aaaa	Suscriptor	0

Acciones en lote [v] Aplicar

Esobrrar un usuari

10.5.9 Activar un plugin per tota la xarxa

Per activar un plugin per tota la xarxa hem de ser exclusivament superadministradors, en cas que vulguem activar un plugin només per un blog, podem ésser administradors normals.

L'activació de plugins es realitza des de el menú *Plugins / Extensions*. Aquest menú ens proporciona un llistat de tots els plugins que hi ha disponibles per activar. Aquests plugins es corresponen amb els que hem instal·lat amb anterioritat.

Per instal·lar un plugin només hi ha que fer clic al botó *Activar*. En cas de voler activar per tots els blogs, haurem de fer *Activar per tota la xarxa*.

Continuant amb l'exemple, activarem *PostsIniciais* i *Jerarquia de blogs* per el blog principal i *Quickr To WordPress* per tota la xarxa.

¹ Llistat d'usuaris d'un blog



Añadir nuevo

Todos (5) | Inactivos (5) | Actualización disponible

Buscar Plugins Instalados

Acciones en lote

<input type="checkbox"/> Plugin	Descripción
<input type="checkbox"/> Akismet Activar Activar para la red Editar Borrar	Akismet checks your comments against the Akismet web service to see if they look like spam or not. You need an API key to use it. You can review the spam it catches under "Comments." To show off your Akismet stats just put <code><?php akismet_counter(); ?></code> in your template. See also: WP Stats plugin . Versión 2.4.0 Por Automattic Visitar la web del plugin
<input type="checkbox"/> Hello Dolly Activar Activar para la red Editar Borrar	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. Versión 1.5.1 Por Matt Mullenweg Visitar la web del plugin
<input type="checkbox"/> Jerarquia de Blogs Activar Activar para la red Editar Borrar	Organitzador de la estructura jeràrquica dels blogs Versión 1.0 Por David Gómez Visitar la web del plugin
<input type="checkbox"/> PostsIniciais v2 Activar Activar para la red Editar Borrar	Selecció de posts per a la pàgina principal Versión 3.0 Por David Gómez Visitar la web del plugin
<input type="checkbox"/> Quickr to WordPress Activar Activar para la red Editar Borrar	Xuclador de blogs de Quickr Versión 2.0 Por David Gómez Visitar la web del plugin

1

¹ Menú d'activació de plugins

10.6 Manual d'administrador

El manual d'administrador serveix per tot usuari administrador d'un blog, encara que també en pot fer ús l'usuari superadministrador.

Per un usuari tipus administrador d'un blog, el menú que té és pràcticament el mateix al que teníem quan no hi havíem fet encara l'activació del *multisite*.

10.6.1 Vista visual del taulell d'administrador

Aquest és l'aspecte que ofereix el taulell de l'administrador.

The screenshot shows the WordPress administrator dashboard. The top bar includes the WordPress logo, the site name 'Blog 1', and user information 'Hola, administrador | Cerrar sesión'. The left sidebar contains a menu with 'Escritorio' selected, and other options like 'Entradas', 'Multimedia', 'Enlaces', 'Páginas', 'Comentarios', 'Apariencia', 'Usuarios', 'Herramientas', 'Ajustes', and 'Blueprint Options'. The main area is titled 'Escritorio' and contains several widgets: 'Ahora mismo' (Now) with a table of content and debate statistics; 'Publicación rápida' (Quick publish) with a form for title, content, and tags; 'Espacio de almacenamiento' (Storage space) showing 10MB used of 10MB allowed; 'Comentarios recientes' (Recent comments) with a comment from 'Sr WordPress'; 'Últimos borradores' (Recent drafts) showing none; and 'WordPress Blog' with a news item about WordPress 3.1 Release Candidate 2.

10.6.2 Aplicació d'un tema

Per fer l'aplicació d'un nou tema a un blog hem d'anar al menú *Aparença*. Aquí ens trobem amb l'opció *Temes*, que serà la que ens permeti activar un tema.

Per activar-lo haurem de donar-li a activar en el tema que vulguem. També podem veure com queda sense activar-lo fent clic a *Vista prèvia*.

Per el nostre projecte, activarem el tema *COEIC*.

Adicionalment, el tema seleccionat té altres opcions que es poden configurar. Aquestes es troben dintre d'un nou menú anomenat *Blueprint Options*.

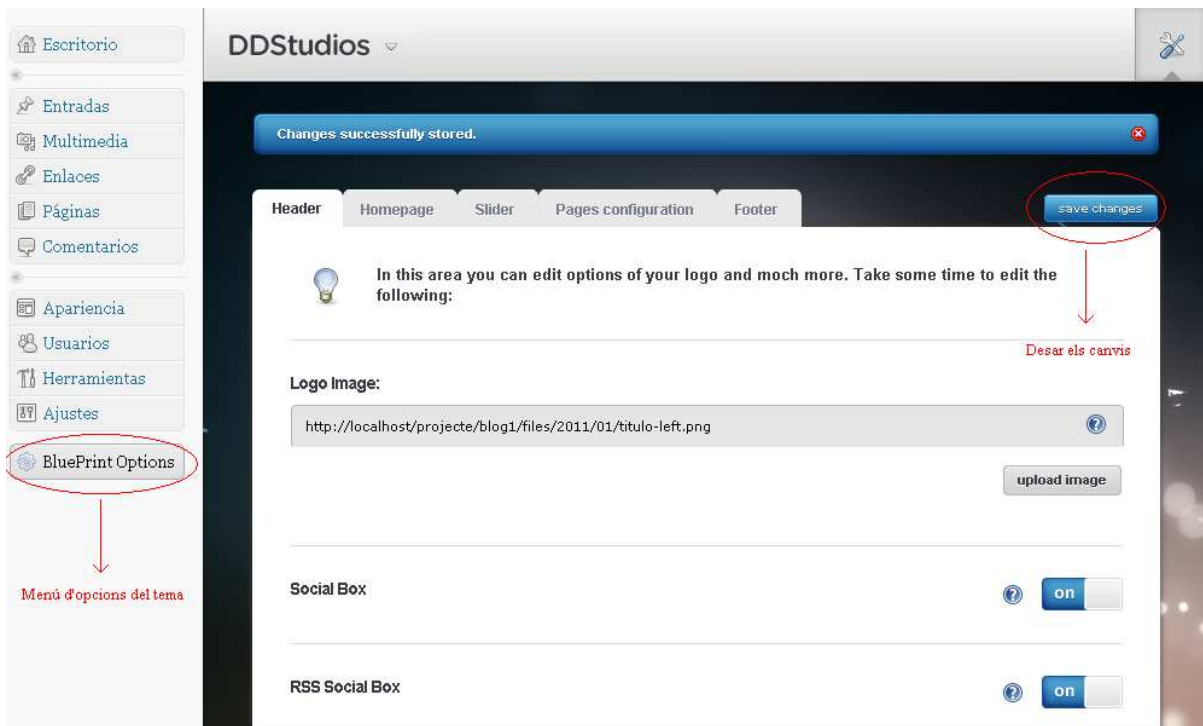
D'aquestes opcions ens interessaven activar les següents:

- **Logo image.** Aquí ens permet afegir una nova imatge per al logotip de la capçalera. Per exemple, carregarem la imatge que es troba dintre del directori *Altres* de l'adjunt *Afegits*. Un cop seleccionada la imatge, aquesta es pujarà al servidor.

¹ Taulell de l'administrador.

- **Ads HTML.** Aquesta opció es troba a la pestanya *Homepage* i ens permet configurar els anuncis que posarem a barra lateral dreta. Els anuncis tenen que estar en format *HTML*. En el nostre cas substituïrem el codi de l'arxiu *Anuncis.txt* del directori *Altres* de l'adjunt *Afegits*, per el que es proporciona per defecte.

Fet això, guardem els canvis amb l'opció *save changes* i ja tindrem el tema de l'aplicació actiu i configurat per el seu ús correcte.



¹ Activació d'un tema.
² Configuració del tema.

10.6.3 Ús del plugin COEIC Home

El plugin *COEIC Home* només està activat per el blog principal, per tant, només en podran fer ús el superadministrador i l'administrador del tema principal.

Perquè tingui un correcte funcionament, abans de fer el primer ús del plugin, hi ha que esborrar els articles introduïts al blog (veure apartat *Esborrar Articles*).

Introduir articles

Per introduir nous articles dintre de la pàgina principal haurem de seguir els passos següents:

- **Cercar l'article corresponent.** Per defecte es mostren tots els articles de la blogosfera ordenats per data, però es pot filtrar per la comissió de la que es vulgui afegir l'article.
- **Introduir rellevància.** Hi ha que introduir una rellevància entre 1 i 5 que indicarà el grau d'importància de l'article, sent 5 el més important.
- **Seleccionar per introduir.** Hi ha que marcar l'article com seleccionat per introduir al blog principal marcant-lo.
- **Acceptar la introducció.** Quan haguem seleccionat tots els articles que vulguem només hi ha que fer clic al botó Acceptar.

Restriccions:

- No poden haver dos articles amb la mateixa rellevància.
- El mateix article no pot ser-hi més d'un cop al blog principal.
- És indispensable marcar l'article a introduir.
- No poden haver-hi més de cinc articles ni seleccionats, ni dins del blog principal.

Esborrar articles

Per fer l'eliminació d'articles existents en el blog hi ha que seguir els següents passos:

- Seleccionar l'article o articles a esborrar.
- Fer clic al botó *Esborra*.

En cas que s'esborrin tots els articles del blog principal, automàticament s'introduiran els cinc articles més nous de la blogosfera per tal que no hi estigui buit.

Selecciona els posts que vulguis que surtin a la pàgina principal

Mostran't posts del 1 fins al 2 de 2

Totes les Comissions

Títol	Comissió	Rellevància	Selecciona
iHola mundo! Blog 2			<input type="checkbox"/>
iHola mundo! Blog 1			<input type="checkbox"/>

Pàgina
1

Acceptar

Selecció de la comissió

Marca per afegir

Selecció de la rellevància

Posts que surten actualment a la pàgina principal

Títol	Comissió	Rellevància	Esborrar
iHola mundo! Blog 1		5	<input type="checkbox"/>
iHola mundo! Blog 2		3	<input type="checkbox"/>

Esborra

Marca per esborrar

10.6.4 Ús del plugin Jerarquia de blogs

Igual que l'anterior plugin, *Jerarquia de blogs* només hi serà actiu al blog principal.

Aquest plugin ens permet tenir ordenat i jerarquitzat el menú de la blogosfera. Per fer-ho hem de seguir els següents passos.

- **Per cada blog, primer de tot hem de seleccionar si té un blog pare.** En cas que sigui així, hi ha que indicar de quin blog es tracta.
- **Seleccionem l'ordre que ocuparà en el menú.**
- **Escollim si volem que sigui visible o no.**
- **Premem el botó *Actualitza*.**

Restriccions:

- Per defecte el blog pare serà la *Home*, el que significa que no es tracta d'una subcomissió.
- Si hi ha dos blogs amb el mateix ordre es mostraran per ordre del seu identificador.
- El primer cop que es faci ús del plugin hi ha que actualitzar, encara que el ordre que es mostri ja sigui el que desitgem. Si no ho fem, els blogs es trobaran en estat *No visible*.

¹ Plugin COEIC Home.

Selecciona el pare de cadascun dels blogs existents

Blog	Pare	Ordre	Visibilitat
Blog 1	Home	1	Si
Blog 2	Blog 1	2	Si

Actualitza

10.6.5 Afegir un widget

Adicionalment, els administradors poden configurar i afegir widgets a les barres laterals. Per fer-ho tenen que anar al menú *Aparença, Ginyes / Widgets*.

Aquí ens trobem amb diferents widgets que podem afegir. Els més comuns són *Categories* i *Pàgines*. En casos excepcionals també és pot fer servir el giny *Text*.

Per afegir el widget, hi ha que arrastrar-ho cap a la barra lateral desitjada. Es recomana posar-ho a la barra lateral dreta.

The screenshot shows the WordPress administration interface for 'Blog 1'. The top navigation bar includes 'Nueva entrada', 'Hola, administrador', and 'Cerrar sesión'. The left sidebar contains various menu items, with 'Widgets' highlighted. The main content area is titled 'Widgets disponibles' and contains a list of widgets that can be dragged to the right sidebar. The right sidebar has four sections: 'Left SideBar', 'Right SideBar', 'Footer Widget', and 'Sidebar Pages'. The 'Widgets disponibles' section includes:

- Archivos**: Un archivo mensual de las entradas de tu sitio.
- Categorías**: Una lista o desplegable de categorías.
- Buscar**: Un formulario de búsqueda para tu sitio.
- Calendario**: Un calendario de las entradas de tu sitio.
- Comentarios recientes**: Los comentarios más recientes.
- Enlaces**: (No description visible)

¹ Plugin Jerarquia de blogs.

² Menú de widgets.

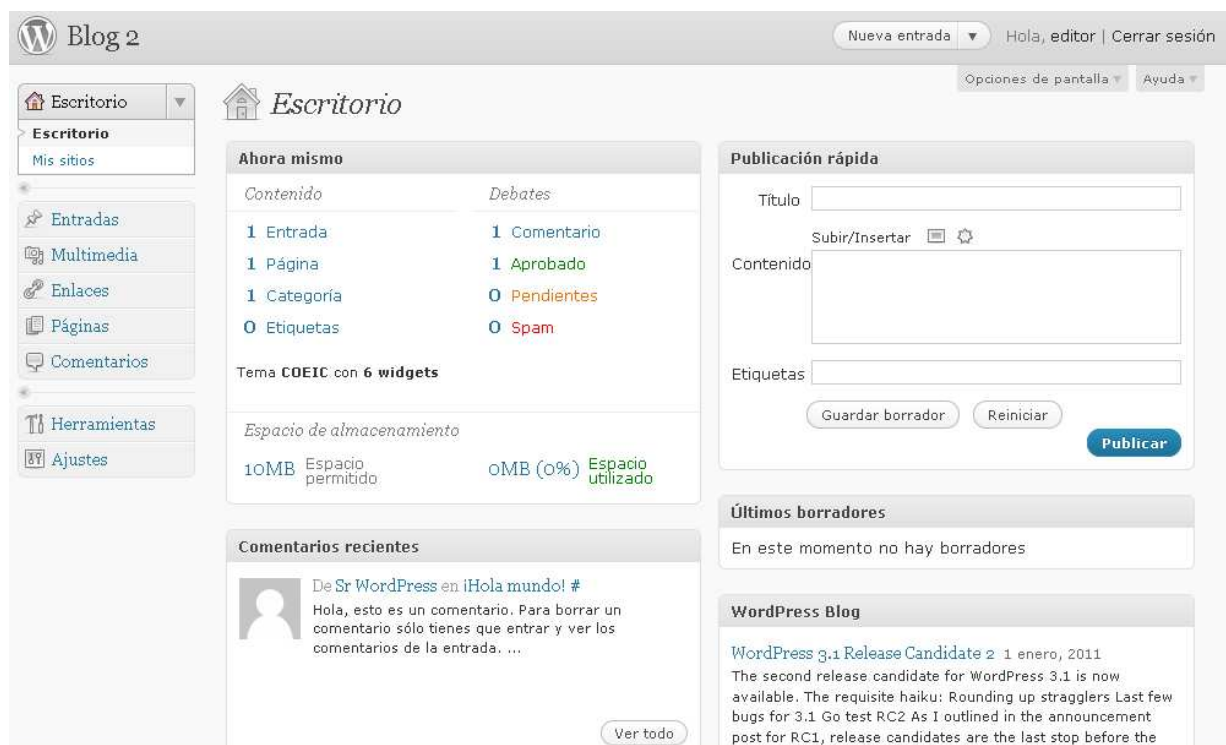
10.7 Manual de l'editor

Com en els anteriors cassos, el manual de l'editor serveix tant per editors, administradors i superadministradors.

Els apartats explicats a continuació corresponen a les noves opcions que es poden fer a partir d'aquest nivell d'usuari.

10.7.1 Vista visual del taulell de l'editor

Aquest és l'aspecte que ofereix el taulell de l'editor.



The screenshot shows the WordPress dashboard for a user named 'Hola, editor'. The interface is in Spanish. On the left, there is a sidebar menu with options like 'Escritorio', 'Entradas', 'Multimedia', 'Enlaces', 'Páginas', 'Comentarios', 'Herramientas', and 'Ajustes'. The main content area is divided into several sections: 'Ahora mismo' (Now) showing a summary of content (1 Entrada, 1 Página, 1 Categoría, 0 Etiquetas) and comments (1 Comentario, 1 Aprobado, 0 Pendientes, 0 Spam); 'Publicación rápida' (Quick publish) with fields for title, content, and tags, and buttons for 'Guardar borrador', 'Reiniciar', and 'Publicar'; 'Espacio de almacenamiento' (Storage space) showing 10MB permitted and 0MB used; 'Comentarios recientes' (Recent comments) with a comment from 'Sr WordPress'; 'Últimos borradores' (Recent drafts) showing no drafts; and 'WordPress Blog' with a news item about WordPress 3.1 Release Candidate 2.

10.7.2 Afegir o editar una plana nova

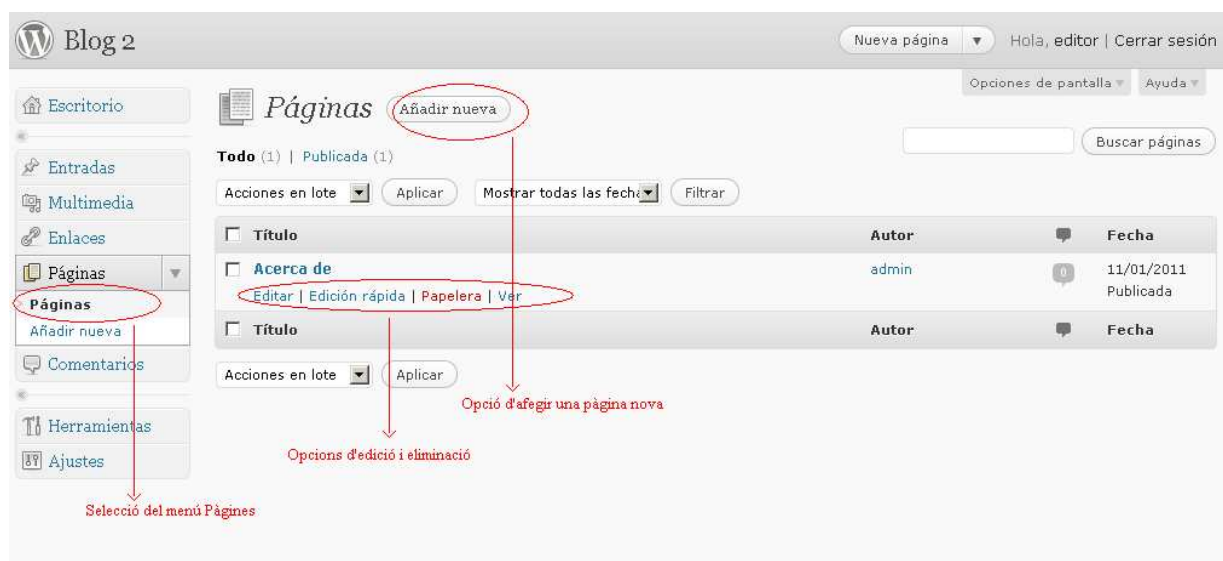
Els editors poden escriure planes noves, que els administradors podran fer que es mostrin a través dels widgets.

Per afegir una plana nova hem de seguir els següents passos.

- **Anar al menú *Pàgines*.** Aquí veurem les pàgines que hi ha actualment dintre del blog. Per defecte tenim la plana *Acerca de*.
- **Prémer el botó *Afegir nova*.** A partir d'aquí, omplir la plana és igual que afegir un article (Veure secció *Afegir articles*).
- **Opcions addicionals.** A més de les opcions típiques que ofereix l'adició d'un article també hem podem afegir d'altres propietats per la pàgina.
 - o **Establir una jerarquia.** Si la plana depèn d'una altre.
 - o **Indicar l'ordre en el que té que anar.** Per mostrar en el menú en el que l'incloem.
 - o **Utilitzar una plantilla.** Per donar-li un estil determinat.

¹ Taulell de l'editor.

Per editar o esborrar una plana existent, només tenim que seleccionar l'opció que volem al passar per damunt de la plana que desitgem dintre de la llista de planes del blog.



10.7.3 Administració de comentaris

Un editor d'un blog també pot administrar el comentaris. Això significa que pot aprovar, esborrar o marcar com *spam* els comentaris que formen el blog.

Per fer-ho té que anar al menú al menú *Comentarios*. Un cop allà, es mostra una llista del comentaris que hi ha al blog, fent referència a l'article al qual pertany.

Les opcions que tenim per els comentaris ens serveixen per lo següent.

- **Rebutjar.** Marca el comentari com que no ha estat aprovat, i per tant no ho mostra. El comentari continua estant dintre del blog.
- **Respondre.** Resposta directa del comentari.
- **Edició ràpida i editar.** Permet la manipulació del contingut del comentari.
- **Spam.** Marca el comentari com *spam*, per tant els pròxims comentaris que escrigui aquest autor no seran publicats. El comentari s'esborra del blog.
- **Enviar a la paperera.** Esborra el comentari. Deixa d'aparèixer al blog però l'usuari pot continuar escrivint comentaris.

¹ Menú d'administració de pàgines.

Comentarios

Opciones de pantalla Ayuda

Todos | Pendientes (0) | Aprobado | Spam (0) | Papelera (0)

Buscar comentarios

Acciones en lote Aplicar Mostrar todos los comentarios Filtrar

Autor	Comentario	En respuesta a
<input type="checkbox"/> Sr WordPress localhost/proyecto/	11/01/2011 at 9:42 Hola, esto es un comentario. Para borrar un comentario sólo tienes que entrar y ver los comentarios de la entrada. Entonces tendrás la opción de editar o borrar. Rechazar Responder Edición rápida Editar Spam Enviar a la Papelera	¡Hola mundo! #
Autor	Comentario	En respuesta a

Acciones en lote Aplicar

1

10.7.1 Importar articles de Quickr. Quickr To WordPress

Els editors dels blogs poden fer ús també del plugin *Quickr To WordPress*. Per fer-ho, tenen que anar al menú d'opcions i fer clic en l'única opció que tenen disponible, *QuickrToWordPress*.

Primer hi ha que carregar la comissió *Quickr* on es contenen els articles desitjats. Per fer-ho la seleccionem del desplegable que tenim.

Posteriorment, seleccionem els articles que volem importar i, si no som superadministradors, fem clic al botó importa, donat que no tindrem cap més opció que la que ja surt a la selecció del blog destí.

Fet això, els blogs que haguem seleccionat s'importaran al nostre blog.

Selecciona l'origen dels blogs

Prova0

Transferencia de posts de Quickr

Selecciona els articles que vulguis que s'afegeixin al blog

Els articles actuals a Quickr son els següents

Títol	Autor	Data	
Article amb 2 imatges	wpsadmin	19-11-2010	<input type="checkbox"/>
Prova d'article nou	wpsadmin	10-11-2010	<input type="checkbox"/>
Article d'interès per tots	wpsadmin	19-03-2010	<input type="checkbox"/>
Article 3	wpsadmin	09-03-2010	<input type="checkbox"/>
Article 2	wpsadmin	09-03-2010	<input type="checkbox"/>
Article 1	xavier.raurell	08-02-2010	<input type="checkbox"/>

Selecciona el destí del post

Accio i cooperacio

Inserta

2

¹ Menú d'administració de comentaris

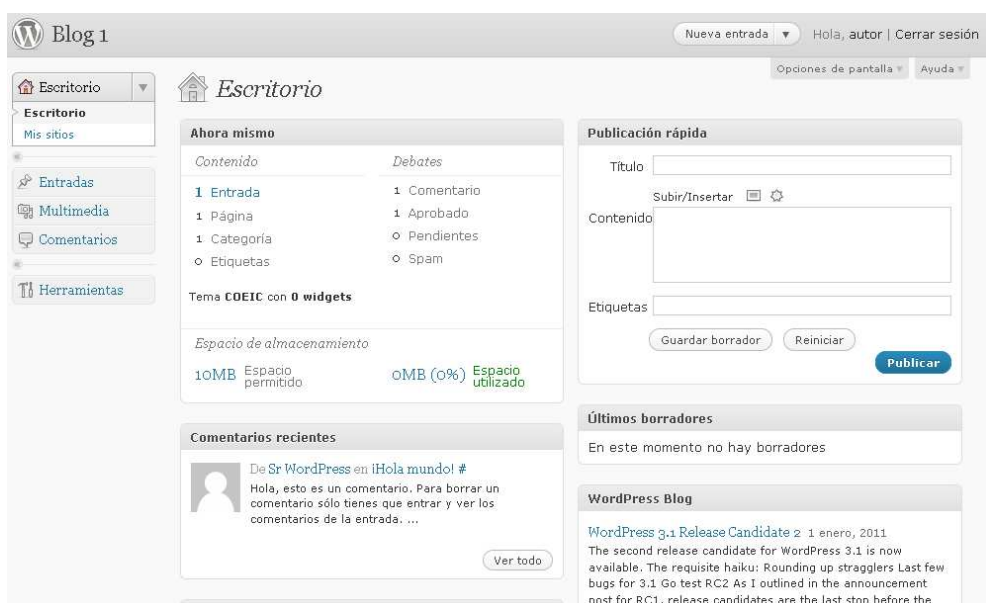
² Plugin QuickrToWordPress

10.8 Manual de l'autor i del col·laborador

Tant l'autor com el col·laborador, són usuaris molt semblants entre ells. La restricció que té un enfront l'altre, és el fet d'escriure articles i quedin aprovats automàticament (autor) o que faci falta aprovació (col·laborador). A més l'autor pot fer ús del a llibreria multimèdia, mentre que el col·laborador només pot escriure articles amb text sense adicions.

10.8.1 Vista del taulell de l'autor

L'aspecte del taulell de l'autor és el següent.



The screenshot displays the WordPress author dashboard. At the top, it shows the site name 'Blog 1', a user menu with 'Nueva entrada', 'Hola, autor', and 'Cerrar sesión', and utility links for 'Opciones de pantalla' and 'Ayuda'. The main area is divided into several sections:

- Escritorio (Dashboard):** A sidebar on the left with navigation links for 'Escritorio', 'Entradas', 'Multimedia', 'Comentarios', and 'Herramientas'.
- Ahora mismo (Right now):** A central widget showing a summary of content and discussions. It lists 1 entry, 1 page, 1 category, and 0 tags under 'Contenido'. Under 'Debates', it shows 1 comment, 1 approved, 0 pending, and 0 spam.
- Publicación rápida (Quick post):** A form on the right for creating a new post, including fields for 'Titulo', 'Contenido', and 'Etiquetas', along with 'Guardar borrador', 'Reiniciar', and 'Publicar' buttons.
- Espacio de almacenamiento (Storage space):** A widget showing '10MB Espacio permitido' and '0MB (0%) Espacio utilizado'.
- Comentarios recientes (Recent comments):** A widget displaying a comment from 'Sr WordPress' with a 'Ver todo' link.
- Últimos borradores (Recent drafts):** A widget indicating 'En este momento no hay borradores'.
- WordPress Blog:** A widget showing a recent blog post titled 'WordPress 3.1 Release Candidate 2' dated January 1, 2011.

¹ Taulell de l'autor.

10.8.2 Vista del taulell del col·laborador

L'aspecte del taulell del col·laborador és el següent.



10.8.3 Afegir articles

L'adició d'articles nous es fa des de el menú *Entrades*. Aquí ens trobarem amb un llistat dels articles que hi ha al blog, de les quals l'únic que podem fer és veure-les, sempre i quan no estigui escrita per nosaltres.

Per afegir un nou article, hi ha que prémer el botó *Afegir nova*. Llavors, redactarem l'article de la següent manera.

- **Introducció del títol.** Al primer espai amb el que ens trobem és el dedicat per el títol de l'article.
- **Contingut.** El contingut el podem escriure tant en format visual, amb l'ajuda d'una barra d'eines per donar estil a la tipografia de l'article, com en format *HTML*.
- **Inserir imatges.** Per inserir imatges hem de fer clic al botó *Afegir una imatge*, llavors seleccionem la imatge que desitgem inserir. Al seleccionar la imatge ens sortiran les propietats d'aquesta i la mida que ha de tenir, omplim el formulari amb els valors adients. Si aquesta es la imatge principal, seleccionarem l'opció *Utilitzar com imatge destacada*. Finalment, premem el botó *Inserir a l'entrada* i ja formarà part de l'article.
- **Extracte.** Si volem fer un petit resum de l'article, hi ha escriure'l en aquest apartat. Aquest serà el que es mostri al blog. En cas de deixar-ho buit, automàticament es mostraren els 150 primers caràcters.
- **Categories.** Si l'article forma part d'una categoria en concret, el fem a aquest apartat.
- **Etiquetes.** Aquí posarem les paraules més destacades de l'article.
- **Imatge destacada.** Si no hem inserit cap imatge abans, podem definir aquí quina serà la imatge destacada. Aquesta imatge serà la que es mostri a la pàgina principal del blog i al visor de presentació d'imatges. En cas de que hi hagi una imatge que no volem, hi ha que donar-li a *Treure imatge destacada*.

¹ Taulell del col·laborador.

Añadir una imagen



Nombre de archivo: bombeta1.gif
Tipo de archivo: image/gif
Fecha de subida: 12 enero, 2011
Dimensiones: 940 × 198

Editar imagen

Título * bombeta1

Texto alternativo
Texto alternativo (alt) de la imagen, por ejemplo "La Mona Lisa"

Leyenda

Descripción

URL del enlace
 Ninguna URL del archivo URL de la entrada
Introduce una URL para el enlace o clic sobre el actual.

Alineación Ninguna Izquierda Centrar Derecha

Tamaño Miniatura (150 × 150) Medio (300 × 63) Grande Tamaño completo (940 × 198)

Insertar en la entrada [Usar como imagen destacada](#) [Borrar](#)

Guardar todos los cambios


1

Añadir nueva entrada

Introduce el título aquí

Enlace permanente: <http://localhost/proyecto/blog1/2011/01/12/> [Editar](#)

Subir/Insertar Visual HTML



Ruta p
Contador de palabras: 0 Borrador guardado a las 11:49:15.

Extracto
Los extractos son resúmenes opcionales de tu contenido hechos "ex-profeso" que puedes usar en tu tema. [Aprende algo acerca de los extractos manuales.](#)

Enviar trackbacks
Enviar trackbacks a:
(Separar varias URLs con espacios)
Los trackbacks son un modo de avisar a sistemas antiguos de que les has enlazado. Si enlazas a otros sitios creados con WordPress recibirán un aviso automático gracias a los [pingbacks](#), sin tener que hacer nada.

Campos personalizados
Añadir nuevo campo personalizado:

Nombre	Valor
<input type="text"/>	<input type="text"/>

[Añadir un campo personalizado](#)

Los campos personalizados se pueden usar para añadir metadatos adicionales a una entrada y luego [mostrarlos en tu tema](#).

Publicar


[Guardar borrador](#) [Vista previa](#)

Estado: **Borrador** [Editar](#)
Visibilidad: **Público** [Editar](#)
[Publicar inmediatamente](#) [Editar](#)

[Mover a la papelera](#) [Publicar](#)

Categorías
Todas las categorías Más utilizadas
 Sin categoría

Etiquetas de las entradas
[Añadir nueva etiqueta](#) [Añadir](#)
Separa las etiquetas con comas.
[Elige entre las etiquetas más utilizadas](#)

Imagen destacada

[Quitar la imagen destacada](#)

2

¹ Inserció d'una imatge.

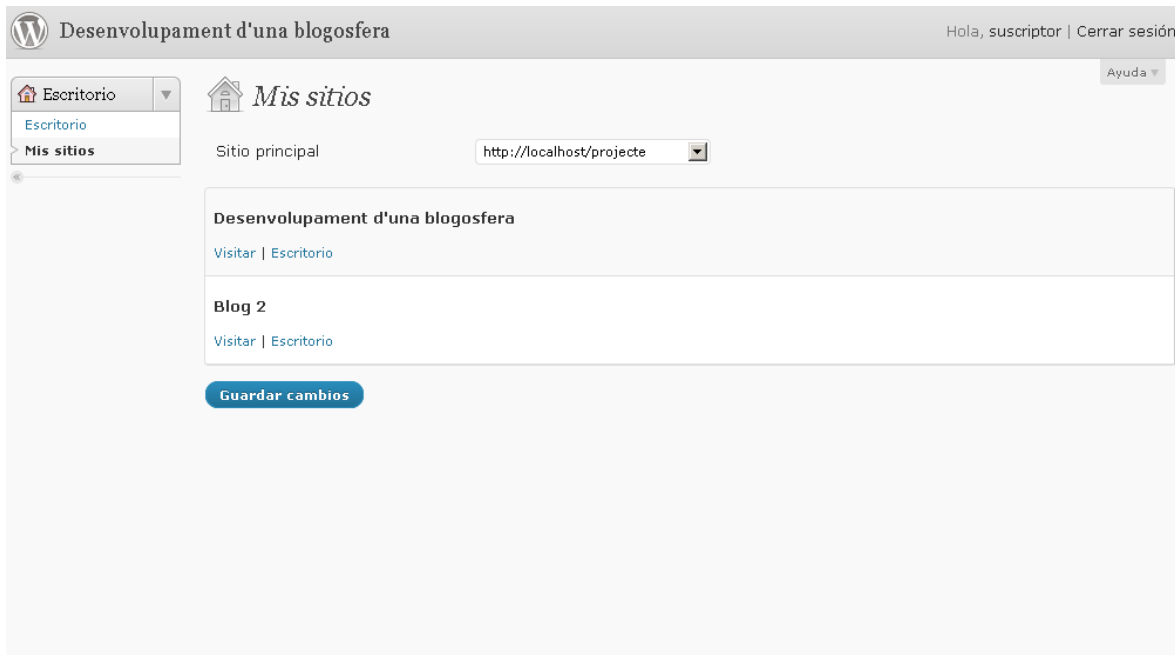
² Inserció d'un article.

10.9 Manual d'usuari del subscriptor

El subscriptor no pot fer res, només pot veure els llocs en els que es troba, per tant només tindrà l'opció *Els meus llocs*, que serà l'última opció que tenen tots els demés usuaris.

10.9.1 Els meus llocs

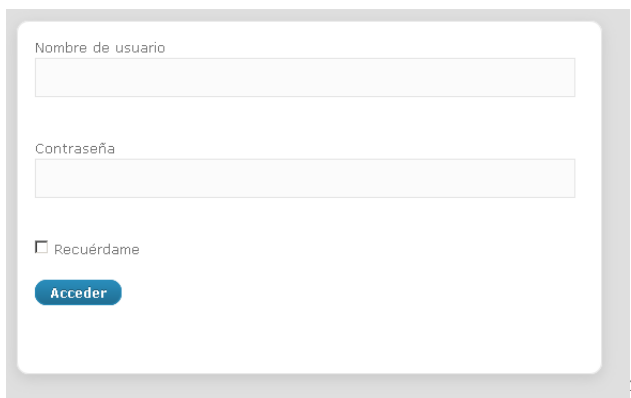
El menú *Els meus llocs* mostra els blogs en els que l'usuari està adherit. A través d'aquest menú els pot visitar o anar al taulell d'ell i configurar quin serà el seu blog principal.



10.10 Altres opcions

10.10.1 Iniciar sessió

Per iniciar sessió dintre de la blogosfera hi ha que tractar d'accedir a algun article. Llavors ens sortirà el formulari de login. Al fer clic a accedir s'iniciarà sessió i podrem accedir a les opcions dels menú meta.



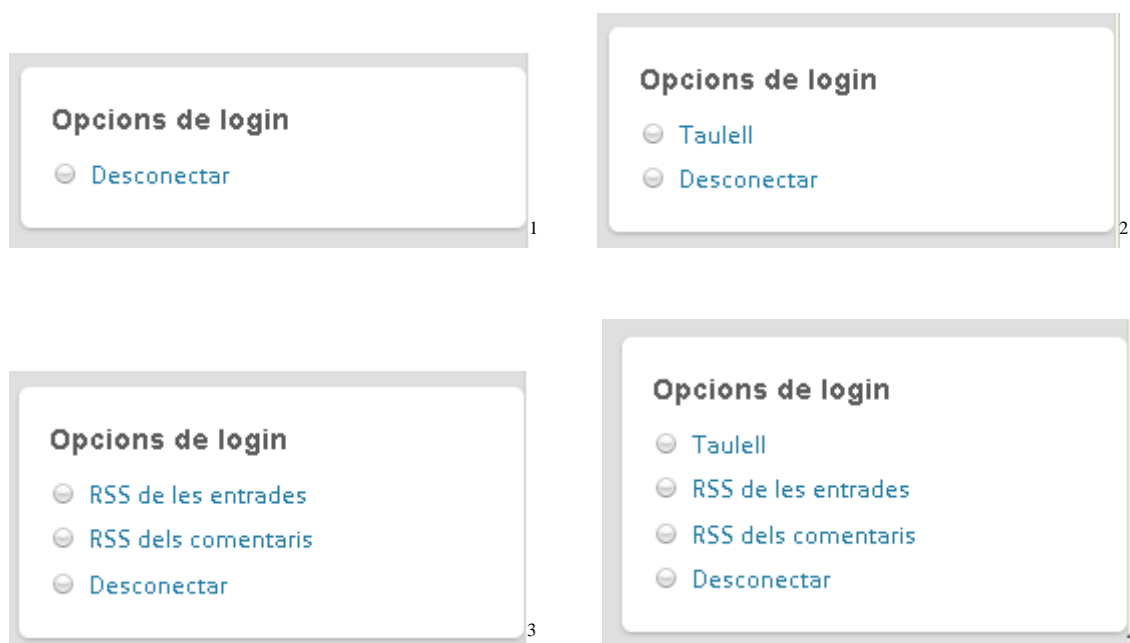
¹ Menú *Els meus Llocs*

² Formulari d'iniciar sessió

10.10.2 Menú meta un cop iniciada la sessió

El menú meta, varia segons el blog i el tipus d'usuari, així tindrem les següents opcions.

- **Blog principal, usuari no administrador.** Es pot accedir al taulell o desconnectar-se.
- **Altres blogs, usuari no subscriptor.** Es pot accedir al taulell, subscribir-se als RSS de comentaris i entrades i desconnectar-se.
- **Blog principal, usuari no administrador.** Només es pot desconnectar.
- **Altres blogs, usuari subscriptor.** Pot subscribir-se als RSS de comentaris i articles i desconnectar-se.



10.10.3 Inserir, modificar i esborrar usuaris

Aquesta opció no és pròpia de cap usuari en concret, sinó que correspon a un procés automàtic que s'executarà cada cop que s'hagi de fer una acció amb els usuaris.

La crida automàtica es fa via URL, fent una crida a l'arxiu *usuaris.php* on s'indica l'acció de la següent manera:

- **Inserció d'un usuari.** Farem una crida del tipus: *http://localhost/projete/usuaris.php?accio=insert&login=log&pass=password&email=mail*. On els paràmetres *log* correspondrà al nom d'usuari, *password* a la contrasenya de l'usuari i *mail* a la direcció de correu de l'usuari. Hi ha que tenir en compte que el *password* ha d'estar encriptat en *base64*.
- **Modificació de les dades d'un usuari.** La crida serà la mateixa que en la inserció, però canviarem *insert* per *update*. Amb el *login* indiquem de quin usuari es tracta i el *password* i/o el mail que li indiquem serà el que nou que vulguem posar. No és necessari posar tots dos, només el que es vulgui

¹ Menú meta del blog principal, per als usuaris no administradors.

² Menú meta del blog principal, per als usuaris administradors.

³ Menú meta d'un blog, per als subscriptors.

⁴ Menú meta d'un blog, per als usuaris no subscriptors.

canviar. En cas de modificar la contrasenya, aquesta haurà d'estar encriptada en *base64*.

- **Eliminació d'un usuari.** Per esborrar un usuari la crida ha de ser la següent: *http://localhost/projete/usuarios.php?accio=delete&login=log*. Estant el *login* codificat en *base64* per restriccions de seguretat.

10.10.4 Recollida d'articles per la fitxa d'usuaris

La recollida d'articles també es fa de forma automàtica i no depèn d'un usuari en concret. Aquesta haurà de cridar-se mitjançant la següent adreça URL: *http://localhost/projete/articles.php?id=identificador*. On *id* correspondrà a l'identificador de l'usuari del que volem recuperar els articles que ha escrit.

El resultat de la crida és una resposta en format *JSON*, per tant hi ha que tractar-la per mostrar-la.

10.10.5 Exportar articles de Quickr a WordPress

L'exportació d'articles està disponible per tota persona que pugui fer ús de *Quickr* e inserir articles en ell. Per fer-ho haurem de seguir els següents passos des de l'aplicació *Quickr*.

- **Accedir a Quickr.** Mitjançant la direcció *http://espais.eic.cat* entrarem a l'aplicació *Quickr* del COEIC. A aquesta direcció haurem de fer login amb el nom d'usuari i contrasenya corresponents.

espais.eic.cat

Los espacios virtuales de los ingenieros. La forma más sencilla de interactuar y compartir contenidos.



ID de usuario:

Contraseña:

Iniciar sesión

Cancelar

1

- **Accedir a la comissió corresponent.** El següent pas serà accedir a la comissió a la que tenim accés a través del menú *Mis Lugares*.

¹ Accés a l'aplicació *Quickr*.

IBM Lotus Quickr

Visitar un lugar y empezar a colaborar con otros.



Crear un lugar

Gestionar lugares

Trabajar con plantillas

Administración de sitio

Obtener más información



Conectores de Lotus

Favoritos **Mis lugares** Lugares públicos Todos

Ordenar por: Título Propietario

[Ocultar detalles](#)

Acte de benvinguda 10-2008

Propietario: • [Anna Alfonso](#)

TMP-Demo backup

Propietario: • [wpsadmin wpsadmin](#)

Prensa Col·legis Professionals

Propietario: • [Cristina Reche](#)

Reunions Comunicació

Propietario: • [Eva Díaz](#)

Grup Química Verda

Propietario: • [Eva Díaz](#)

Reunions Continguts Portal

Propietario: • [Maite Sanz](#)

Comissió (Beta)

Propietario: • [wpsadmin wpsadmin](#)

Grup Fem Indústria

Propietario: • [Jordi Sellés](#)

Pulse para obtener tarjeta personal

Comissió de Construcció

Espai de treball de la Comissió de Construcció

1

- **Accés a l'exportació.** Fent clic a la pestanya *Exportació* de la comissió, accedirem al component *QuickrToWordPress* en la seva versió d'exportació.
- **Exportar els articles corresponents.** Aquí seleccionem els articles que volem exportar a *WordPress* e indiquem el blog en el que aniran. Finalment, fem clic al botó *Exportar*.

Enginyers Industrials de Catalunya

Comissió de Construcció

Finalitzar sessió: wpsadmin wpsadmin

Volver a Inicio Ayuda

Comissió de Construcció Personalizar

Novetats Calendari Arxius Documentació Membres Blog **Exportació**

Articles a exportar a WordPress

Títol	Autor	Data	
Article amb 2 imatges	wpsadmin	19-11-2010	<input type="radio"/>
Prova d'article nou	wpsadmin	10-11-2010	<input type="radio"/>
Article d'interès per tots	wpsadmin	19-03-2010	<input type="radio"/>
Article 3	wpsadmin	09-03-2010	<input type="radio"/>
Article 2	wpsadmin	09-03-2010	<input type="radio"/>
Article 1	xavier.raurell	08-02-2010	<input type="radio"/>

Accio

Exportar

2

Fent això exportarem a *WordPress* els articles que vulguem.

¹ Selecció de la comissió *Quickr*.

² Selecció de l'article a exportar a *WordPress*.

11. Glossari

Banner: espai dedicat dintre d'una pàgina web per posar algun anunci, normalment publicitari.

Base de dades: conjunt de dades pertinents a un mateix context i emmagatzemats sistemàticament per al seu posterior ús. Es componen per taules principalment i es poden fer indefinides consultes de diferents tipus sobre aquestes.

Blog: espai web que es va actualitzant periòdicament i que recopila de forma cronològica els articles o textos d'un o varis autors.

Blogosfera: terme que agrupa al conjunt de blogs dintre d'un espai.

Categoria: desglossament d'articles que formen part d'un tipus de temàtica determinada.

CMS: *Content Management System* (sistema de gestió de continguts). És un programa que permet crear una estructura de suport per a la creació i administració de continguts, normalment pàgines web.

Codex: documentació oficial de *WordPress*.

Codi Font: conjunt de línies de text que componen un programa. Són instruccions que ha de seguir l'ordinador per interpretar les comandes de l'usuari cap al programa.

Codi Obert: el codi obert o *Open Source*, és el terme que es dona als programes que han estat desenvolupats de forma lliure. No tenen fins lucratiu sinó que tenen la finalitat de ser utilitzats per a beneficis pràctics, per compartir amb la resta d'usuaris.

Comissió: àrea de coneixement formada per un grup d'experts en la matèria.

CSS: els *Cascading Style Sheets*, fulls d'estils en cascada ó fulls d'estils, és un llenguatge per definir la presentació d'un document estructurat escrit en HTML o XML. Donen color i forma a cadascuna de les etiquetes del document.

Domini: direcció a través de la qual es pot accedir a l'espai web.

Espill: principi de disseny de software que indica que un element de la realitat es té que correspondre amb un, i només un, element del model. No poden existir elements al model que no existeixin a la realitat.

Etiqueta: paraula o conjunt petit de paraules que serveix per identificar el tema sobre el que tracta un article. Útil per fer indexació i, per tant, per fer una cerca ràpida d'articles.

Framework: és una estructura conceptual i tecnològica de suport definida que serveix com a base perquè un altre projecte pugui ser desenvolupat.

Fòrum: aplicació web que dona suport a discussions o a opinions en línia.

FTP: *File Protocol Transfer*, protocol de transferència d'arxius. Protocol que serveix per pujar o baixar arxius d'un espai web.

Getter: mètode de recuperació d'atributs privats d'un mètode.

Giny: petita aplicació per la part visual del blog. S'afegeix a les barres laterals.

Home: en termes de pàgines web es refereix a la pàgina principal, la primera a la que entrem quan ens dirigim a una plana web.

Hosting: el *Hosting* o allotjament web és un servei que proveeix als usuaris d'Internet d'un sistema per poder emmagatzemar informació, imatges o qualsevol contingut al que es pugui accedir via web.

HTML: llenguatge de programació d'etiquetes. És el llenguatge base per a la creació de pàgines web.

HTTP Basic Authentication: protocol de seguretat d'HTML per el qual demana autenticació a l'hora d'accedir a una direcció Web.

Id: identificador únic per un blog, usuari, opció...

Indexació: crear un índex per tal de aconseguir cerques més ràpides.

Java: llenguatge de programació orientat a objectes desenvolupat per *Sun Microsystems*.

Jerarquia de nivells: sistema d'administració de permisos d'usuaris, quan més alta és la jerarquia, més coses pot fer l'usuari. A *WordPress* distingim entre SuperAdministrador, Administrador, Editor, Autor, Col·laborador i Subscriptor.

Json: llenguatge de programació que serveix per l'intercanvi de dades.

LDAP: protocol de nivell d'aplicació que permet l'accés a un servei de directori ordenat i distribuït per cercar informació a una xarxa. També es considera una base de dades a la que es poden realitzar consultes.

Multiblogging: és la característica de tenir diversos blogs diferents associats a un mateix espai de treball.

Multilloc: igual a multiblogging.

Multisite: igual a multiblogging.

MySQL: sistema gestor de bases de dades de lliure distribució.

Newsletter: els *Newsletters* o butlletins informatius són publicacions distribuïdes de forma regular. En el nostre cas es tracta de butlletins amb els darrers articles introduïts a la blogosfera.

Oracle: sistema gestor de bases de dades relacionals.

Perl: llenguatge de programació orientat a objectes.

PHP: llenguatge de programació prepàgina, interpretat i orientat a objectes destinat a la programació de pàgines web.

Plugin: petit programa que es desenvolupa per afegir-lo a un altre. Serveixen per ampliar o millorar les funcions que hi ha per defecte.

Portal web: és un espai web que serveix per oferir a un usuari una sèrie de recursos i serveis.

Portlet: component que s'afegeix a una pàgina de portal web. Està format per una interfície d'usuari i una part de codi que el gestiona.

Post: són cadascú dels articles que es publiquen en un blog. Poden contenir, a més de text, imatges, so i vídeos.

PostgreSQL: sistema gestor de base de dades relacional de codi lliure.

Python: llenguatge de programació orientat a objectes que es caracteritza per la filosofia de programar amb sintaxis neta per tal de tenir un codi llegible.

Quick Sort: mètode d'ordenació d'elements recursiu de forma ràpida.

Quickr: aplicació de portal elaborada per IBM que permet tenir diferents portals interactius per als usuaris.

RSS: és un format XML que s'utilitza per difondre informació actualitzada freqüentment als usuaris subscriptes a la font de continguts. Permet mostrar continguts sense necessitat de cap navegador, fent ús de software especial per llegir els RSS.

SEO: procés per millorar la visibilitat d'una pàgina web dintre dels cercadors d'Internet.

Servidor Web: es tracta d'un programa dissenyat per transferir textos complexos com pàgines web.

Setter: mètode de modificació de dades d'atributs privats.

SGBD: són les sigles de *Sistema Gestor de Bases de Dades*, es tracta de programes dedicats a servir d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.

Sidebar: component d'un tema que s'encarrega de gestionar el contingut de les barres laterals.

Slider: petita aplicació que implementa un visor d'imatges.

Spam: missatges no desitjats, de remitent desconegut que acostumen a ser de tipus publicitari.

SQL: llenguatge d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions amb elles.

Subdirectori: una estructura en subdirectoris és aquella a la que s'accedeix a la web mitjançant la fórmula `http://domini/subdirectori`.

Subdomini: una estructura en subdominis és aquella a la que s'accedeix a la web mitjançant la fórmula `http://subdomini.domini`.

Taulell: escriptori de gestió i administració de *WordPress*.

Tema: els temes o plantilles de presentació corresponen al conjunt d'arxius que serveixen per donar l'aspecte físic a la pàgina.

URL: seqüència de caràcters que segueixen un estàndard per anomenar recursos a Internet. Els recursos poden ser pàgines web (HTML), servidors (FTP) ó servidors de correus entre altres.

Widget: petita aplicació que s'incorpora dintre de una barra lateral.

Wiki: espai web on les pàgines del qual poden ser editades per múltiples voluntaris a través d'un navegador web. Es diferencien dels blogs en que els blogs un únic usuari escriu l'article i a la wiki, pot ser un usuari el que crea l'article, però la resta d'usuaris el poden modificar per afegir, esborrar o actualitzar la informació que conté.

Wp-content: directori intern de *WordPress* en el qual es guarden els fitxers i directoris corresponents als temes, plugins, llenguatges i arxius pujats per els usuaris.

XML: llenguatge d'etiquetes que serveix per definir conceptes i el seu contingut. Un dels usos més generalitzats és per produir *RSS*.

12. Annexa

12.1 Plugin Jerarquia de Blogs

Arxius corresponents al plugin Jerarquia de Blogs

12.1.1 Classe principal del plugin (jerarquia.php)

```
<?php
/*
Plugin Name: Jerarquia de Blogs
Plugin URI: URL
Description: Organitzador de la estructura jerarquia dels blogs
Author: David Gómez
Version: 1.0
Author URI: URL
*/

class jerarquia{
    var $opt;
    var $table_name;
    // Mètode constructor del plugin.
    function jerarquia() {
        //Variable global de WordPress per permetre l'accés a base de dades.
        global $wpdb;
        //Accés a la instal·lació del plugin.
        register_activation_hook(__FILE__, array(&$this, 'install'));
        //Accés a la desinstal·lació del plugin.
        register_deactivation_hook(__FILE__, array(&$this, 'uninstall'));
        //Posició del menú al menú.
        add_action('admin_menu', array(&$this, 'config_page'));
    }

    // Instal·lació del plugin. No necessita instal·lació.
    function install() {}

    // Desinstal·lació del plugin. No necessita desinstal·lar res.
    function uninstall() {}

    // Pàgina de configuració.
    function config_page() {
        /**Definició de les opcions de configuració:
        - Títol de la pàgina.
        - Títol del menú.
        - Nivell d'usuari que pot fer ús. 10 = superadministrador.
        - Nom del menú.
        - Acció del menú. En el nostre cas crida a la funció options_xucla.
        */
        add_options_page(
            'Jerarquia de Blogs',
            'Jerarquia de Blogs',
            10,
            'Jerarquics',
            array(&$this, 'options_jer')
        );
    }

    //Cos del plugin
    function options_jer() {
        include('classSite.php');
        include('classBlog.php');
        //Definim la variable global d'accés a base de dades per quan ens faci
        falta.
    }
}
```



```

global $wpdb;
?>
<h3>Selecciona el pare de cadascun dels blogs existents</h3>
<form method="post" name="selectpare">
  <table border="0">
    <tr><td><b>Blog&nbsp;</b></td><td><b>Pare&nbsp;</b></td><td><b>Ordre&nbsp;</b></
b></td><td><b>Visibilitat&nbsp;</b></td></tr>
  <?php
  //Recuperem els identificadors dels blogs.
  $sql_comi = "SELECT blog_id, path
              FROM ".$wpdb->prefix."blogs";
  $blogs = $wpdb->get_results($sql_comi);
  $sql_n = "SELECT COUNT(blog_id) AS nblog
           FROM ".$wpdb->prefix."blogs";
  //Número de blogs creats.
  //$nblog = get_blog_count();
  $nblog = $wpdb->get_var($sql_n);
  foreach ($blogs as $b){
    //Per cada blog, excepte el principal guardaem el nom i el id.
    if($b->blog_id != 1){
      $blog[$b->blog_id] = new BlogJ($b->blog_id);
    }
  }
  echo "<input type='hidden' name='accio' value='1'>";
  //Per cada blog que tenim amb informació el mostrem i
  //mostrem també un desplegable amb els blogs que pot escollir com pare.
  foreach ($blog as $nb){
    $pare = $nb->getPare();
    $ordre = $nb->getOrdre();
    echo "<tr><td>".$nb->getNom()."</td>";
    echo "<td><select style='width:380px' name = 'par_" . $nb->
    >getId().">";
    echo "<option value=1";
    if(is_null($pare) or $pare == 1)
      echo " selected ";
    echo ">Home</option>";
    foreach ($blog as $p){
      //No mostrem l'opció de seleccionar-se a si mateix. Provacaria un
error.
      if($p->getId() != $nb->getId()){
        echo "<option value=" . $p->getId();
        //Marquem com seleccionat el que té actualment com pare.
        if($pare == $p->getId()) echo " selected";
        echo ">". $p->getNom()."</option>";
      }
    }
    echo "</select></td>";
    echo "<td><select name = 'orden_" . $nb->getId().">";
    for($i = 1; $i < $nblog; $i++){
      echo "<option value='". $i. "'";
      if($i == $ordre)
        echo " selected ";
      echo ">". $i. "&nbsp;</option>";
    }
    echo "</td>";
    echo "<td><select name = 'visible_" . $nb->getId().">";
    echo "<option value='si'";
      if($nb->getVisibilitat() == "si") echo " selected ";
    echo ">Si</option>";
    echo "<option value='no'";
      if($nb->getVisibilitat() == "no") echo " selected ";

```

```

        echo ">No</option>";
    echo "</select>";
    echo "</tr>";
}
?>
<tr><td colspan="2">&nbsp;</td></tr>
<tr><td colspan="2" align="right">
    <input class="button-primary" type="submit" value="Actualitza">
</td></tr>
</table>
</form>
<?php
////////// Acció del formulari //////////
//Creem un nou controlador que administri tot.
//Per cada element el recollim al controlador,
//li afegim el pare i li afegim l'ordre.
//Per últim afegim la visibilitat del blog.
$site = new SiteJ();
if(isset($_POST['accio']) && $_POST['accio'] == 1){
    foreach ($blog as $accbl){
        $par = "par_". $accbl->getId();
        $pa = $_POST[$par];
        $ordree = "orden_". $accbl->getId();
        $or = $_POST[$ordree];
        $vis = "visible_". $accbl->getId();
        $v = $_POST[$vis];
        $site->Recull_blog($accbl->getId());
        $site->Afegir_pare($pa);
        $site->Afegir_ordre($or);
        $site->Afegir_visibilitat($v);
    }
    ?>
    <form name="fr" method="post" action=""><input type="hidden"
name="rec" value="1"></form>
    <script language="javascript">document.fr.submit();</script>
    <?php
    }
} //Fi del mètode d'opcions.
}
$jer = new jerarquia();

```

12.1.2 Classe corresponent al controlador del plugin (classSite.php)

```
<?php
/**
    Classe controlador.
*/
class SiteJ{

    //Atribut privat que correspon a un objecte Blog.
    private $b;

    /**
        Mètode constructor, crea un nou controlador.
    */
    public function __construct(){
    }

    /**
        Mètode de recollida d'un nou blog a partir del identificador.
    */
    public function Recull_blog($id){
        $this->b = new BlogJ($id);
    }

    /**
        Mètode per afegir el pare a un blog.
    */
    public function Afegir_pare($id){
        $pare = new BlogJ($id);
        $this->b->Afegir_pare($pare);
    }

    /**
        Mètode per afegir l'ordre d'un blog al menú.
    */
    public function Afegir_ordre($ord){
        $this->b->Afegir_ordre($ord);
    }

    /**
        Mètode per afegir la visibilitat del blog al menú.
    */
    public function Afegir_visibilitat($v){
        $this->b->Afegir_visibilitat($v);
    }
}
?>
```

12.1.3 Classe corresponent al component Blog del plugin (classBlog.php)

```
<?php
/**
 * Classe que crea un nou objecte de tipus Blog. Conté les dades del blog.
 * Dades del blog:
 *   @id: identificador del blog a transformar.
 *   @path: ruta URL d'accés al blog.
 *   @nom: títol del blog.
 *   @pare: blog del que depèn.
 *   @fills: array amb els blogs que depenen d'ell.
 *   @vist: flag que indica si el blog ha sigut pintat o
 *   no al menú (0 = no pintat).
 *   @visibilitat: indica si el blog es visible al menú o no.
 */

class BlogJ{
    private $id;
    private $nom;
    private $path;
    private $pare;
    private $ordre;
    private $fills;
    private $vist;
    private $visibilitat;

    /**
     * Mètode constructor, crea un nou Blog.
     * @id: id del blog a crear.
     * @pa: path del blog (opcional).
     */
    public function __construct($id){

        $this->id = $id;
        $this->nom = BlogJ::name();
        $this->path = BlogJ::path();
        $this->pare = BlogJ::pare();
        $this->ordre = BlogJ::ordre();
        $this->visibilitat = BlogJ::visi();
        $this->vist = 0;

    }

    /**
     * Mètode que retorna un array amb els blogs fills.
     */
    public function getFills(){
        global $wpdb;
        //Recuperem els ids dels blogs.
        $sql_comi = "SELECT blog_id
                    FROM ".$wpdb->prefix."blogs";
        $blogs = $wpdb->get_results($sql_comi);
        foreach ($blogs as $bl){
            //Per cada blog generem un objecte amb la seva informació.
            $pa = new BlogJ($bl->blog_id);
            $pare_bl = $pa->getPare();
            //Cas en el que el blog sigui fill del que tractem.
            if($pare_bl == $this->id){
                //Es guarda a l'array. Si no existeix l'array el creem.
                if(count($this->fills) == 0) $this->fills = array();
                array_push($this->fills, $bl->blog_id);
            }
        }
    }
}
```

```

    }
    return $this->fills;
}

/**
 * Mètode privat que cerca el nom del blog i el retorna.
 * En cas de que es tracti del blog principal el titula "Home".
 */
private function name(){
    global $wpdb;
    $id = $this->id;
    if($id == 1) return "Home";
    switch_to_blog($id);
    $optname = "'blogname'";
    $sql = "SELECT option_value
            FROM ".$wpdb->prefix."options
            WHERE option_name = ";
    $sql_name = $sql.$optname;
    $nom = $wpdb->get_var($sql_name);
    restore_current_blog();
    return $nom;
}

/**
 * Mètode privat que cerca i retorna el pare del que depèn el blog.
 * Retorna el pare actual del blog.
 */
private function pare(){
    global $wpdb;
    $id = $this->id;
    switch_to_blog($id);
    $optname = "'pare'";
    $sql = "SELECT option_value
            FROM ".$wpdb->prefix."options
            WHERE option_name = ";
    $sql_pare = $sql.$optname;
    $pare = $wpdb->get_var($sql_pare);
    restore_current_blog();
    return $pare;
}

/**
 * Mètode que afegeix un blog com a pare del blog actual.
 * @bl: objecte Blog que es vol posar com a pare.
 */
public function Afegir_pare($bl){
    global $wpdb;
    $idPare = $bl->getId();
    switch_to_blog($this->id);
    //Cerquem el pare que te actualment.
    //En el cas que no tingui pare l'afegim, en cas contrari l'actualitzem.
    if(!isset($this->pare) or is_null($this->pare) or ($this->pare == "")){
        $sql_ins = "INSERT INTO ".$wpdb->prefix."options
                    (option_name, option_value, blog_id)
                    VALUES ('pare', ".$idPare.", 0)";
        $wpdb->query($sql_ins);
    }
    else{
        $sql_up = "UPDATE ".$wpdb->prefix."options
                  SET option_value = ".$idPare."
                  WHERE option_name = 'pare'";
    }
}

```

```

        $wpdb->query($sql_up);
    }
    //Marquem l'atribut pare amb el nou pare.
    $this->pare = $idPare;
    restore_current_blog();
}

/**
 * Mètode que afegeix la posició del menú al blog.
 * @ord: ordre que tindrà el blog dintre del menú.
 */
public function Afegir_ordre($ord = 1){
    global $wpdb;
    switch_to_blog($this->id);
    //Cerquem l'ordre actual que té.
    //En cas que no tingui ordre el creem, en cas contrari l'actualitzem.
    if(!isset($this->ordre) or is_null($this->ordre) or ($this->ordre ==
"")){
        $sql_inzor = "INSERT INTO ".$wpdb->prefix."options
                    (option_name, option_value, blog_id)
                    VALUES ('ordre', ".$ord.", 0);";
        $wpdb->query($sql_inzor);
    }
    else{
        $sql_upord = "UPDATE ".$wpdb->prefix."options
                    SET option_value = ".$ord."
                    WHERE option_name = 'ordre'";
        $wpdb->query($sql_upord);
    }
    $this->ordre = $ord;
    restore_current_blog();
}

/**
 * Mètode que selecciona si un blog és visible al menú o no.
 * @v: visibilitat (si o no)
 */
public function Afegir_visibilitat($v = "si"){
    global $wpdb;
    switch_to_blog($this->id);
    if(!isset($this->visibilitat) or is_null($this->visibilitat) or ($this->visibilitat=="")){
        $sql_insvi = "INSERT INTO ".$wpdb->prefix."options
                    (option_name, option_value, blog_id)
                    VALUES ('visibilitat', '".$v."', 0);";
        $wpdb->query($sql_insvi);
    }
    else{
        $sql_upvi = "UPDATE ".$wpdb->prefix."options
                    SET option_value = '".$v.'"
                    WHERE option_name= 'visibilitat'";
        $wpdb->query($sql_upvi);
    }
    $this->visibilitat = $v;
    restore_current_blog();
}

```

```

/**
 * Cerca i retorna l'ordre que ocupa al menú
 */
private function ordre(){
    global $wpdb;
    $or = get_blog_option($this->id, 'ordre');
    return $or;
}

/**
 * Cerca i retorna la URL del blog.
 */
private function path(){
    global $wpdb;
    $id = $this->id;
    $sql_path = "SELECT path
                FROM ".$wpdb->prefix."blogs
                WHERE blog_id = ".$id.";";
    $path = $wpdb->get_var($sql_path);
    return $path;
}

/**
 * Cerca i retorna la visibilitat del blog al menú.
 */
private function visi(){
    global $wpdb;
    $v = get_blog_option($this->id, 'visibilitat');
    return $v;
}
}
?>

```

12.1.4 Arxiu que pinta el menú dintre del tema (sidebar.php)

```
<!-- Barra lateral esquerra -->
<div id="primary" class="widget-area">
  <div class="box">
    <ul>
      <?php
        //Incloem la classe blog.
        global $blog_id;
        include_once ("classBlog.php");
        global $wpdb;
        //Recupera tots els ids dels blogs.
        switch_to_blog(1);
        $sql_comi = "SELECT blog_id
                    FROM ".$wpdb->prefix."blogs";
        $blogs = $wpdb->get_results($sql_comi);
        //Inicialitza l'array $mb, que conté la informació de cada blog.
        foreach ($blogs as $b){
          $mb[$b->blog_id] = new BlogT($b->blog_id);
          //Guarda en un nou array tots els blogs, menys la home i els blogs
no visibles.
          if(count($a) == 0) $a = array();
          if($b->blog_id != 1 and $mb[$b->blog_id]->getVisibilitat() == "si")
array_push($a, $mb[$b->blog_id]);
        }
        if(count($a) != 0){
          //Ordenem el blogs per el camp 'ordre' mitjançant l'algorisme Quicksort.
          Quick_sort($a,0,count($a)-1);
        }
        //Pintem l'enllaç a la home.
        echo "<li><a href='". $mb[1]->getPath()."'>". $mb[1]-
>getNom()."</a></li>";
        //Pintem cada blog.
        if(count($a) != 0){
          for($i=0;$i<count($a);$i++){
            //Només es crida al mètode de pintar els blogs principals.
            if($a[$i]->getPare() == 1 or $a[$i]->getPare() == 0)
              pinta($a[$i], $mb);
          }
          restore_current_blog();
        }
      ?>
    </ul>
  </div>
</div>
<!-- // Fi barra lateral esquerra -->
<?php

/**
 * Funció recursiva que pinta un element del menú amb els seus submenús.
 * @pare: element del menú a tractar.
 * @arbl: array amb tots els blogs.
 */
function pinta(&$pare, &$arbl){
  //La funció s'executarà sempre que no s'hagi visitat l'element.
  if($pare->getVist() != 1){
    //Cas en que l'element té fills i no es tracta de la home.
    if(count($pare->getFills()) != 0 and $pare->getPare() != 0){
      //Es mostra el pare, es crea un submenú i es torna a cridar
      // a la funció per pintar als fills.
      echo "<li><a href='". $pare->getPath()."'>".
        $pare->getNom()."</a></li>";
      $pare->setVist(1);
    }
  }
}
```



```

        echo "<ul>";
        $ar_fil = $pare->getFills();
        foreach($ar_fil as $fill){
            $fill = $arbl[$fill];
            pinta($fill, $arbl);
        }
        echo "</ul>";
    }
    //Cas en que no tingui fills. Es pinten directament.
    else{
        echo "<li><a href='". $pare->getPath(). "'>".
            $pare->getNom(). "</a></li>";
        $pare->setVist(1);
    }
}
}

/**
Funció recursiva d'ordenació Quicksort. Ordena l'array de blogs per el
camp ordre.
@array: array que volem ordenar.
@inici: posició inicial a ordenar.
@fi: posició final a ordenar.
*/
function Quick_sort(&$array,$inici,$fi){
    //Recuperem la posició de la meitat de l'array.
    //El mètode floor de php redondeja un float per la part baixa.
    $mitad=$array[floor(($inici+$fi)/2)]->getOrdre();
    //Guardem la posició d'inici i final.
    $i=$inici;
    $j=$fi;

    do{
        //Cerquem els elements més grans que la meitat (començant
        //per l'inici).
        while ($array[$i]->getOrdre()<$mitad) $i++;
        //Cerquem els elements més petits que la meitat (començant per
        //el final).
        while ($array[$j]->getOrdre()>$mitad) $j--;
        //Si no s'han creuat i i j.
        if ($i<=$j){
            //Canviem les posicions per posar els més petits a un costat
            //i els més grans
            //a l'altre.
            $temp=$array[$i];
            $array[$i]=$array[$j];
            $array[$j]=$temp;
            $i++;
            $j--;
        }
    } while ($i<=$j);
    //Si encara no hem acabat d'ordenar l'array (cas en que j no ha arribat
    //a l'inici i i no ha arribat al final) continuem cridant a quicksort
    //però amb l'array que ens queda per ordenar.
    if ($inici<$j) Quick_sort($array, $inici, $j);
    if ($i<$fi) Quick_sort($array, $i, $fi);
}
?>

```

12.2 Plugin COEIC Home

Arxius corresponent al plugin COEIC Home

12.2.1 Classe principal del plugin (coeic2.php)

```
<?php
/*
Plugin Name: PostsInicials v2
Plugin URI: URL
Description: Selecció de posts per a la pàgina principal
Author: David Gómez
Version: 3.0
Author URI: URL
*/
class coeic {
    var $opt;
    var $table_name;
    // Mètode constructor del plugin.
    function coeic() {
        //Variable que permet l'accés a base de dades.
        global $wpdb;
        //Funció d'instal·lació del plugin.
        register_activation_hook(__FILE__, array(&$this, 'install'));
        //Funció de desinstal·lació del plugin.
        register_deactivation_hook(__FILE__, array(&$this, 'uninstall'));
        //Afegim el plugin a les opcions d'administració
        add_action('admin_menu', array(&$this, 'config_page'));
    }

    // Opcions de configuració del menú plugin.
    function config_page() {
        /**Definició de les opcions de configuració:
        - Títol de la pàgina.
        - Títol del menú.
        - Nivell d'usuari que pot fer ús. 10 = superadministrador i
          administrador.
        - Nom del menú.
        - Acció del menú. En el nostre cas crida a la funció options_xucla.
        */
        add_options_page('COEIC Home', 'COEIC Home', 10, 'COEIC_options2',
array(&$this, 'options_plugin'));
    }

    //Cos del plugin. Configuració del que ha de mostrar i del que ha de fer el
    plugin.
    function options_plugin() {
        //Definim la variable global d'accés a base de dades.
        global $wpdb;
        //Acció per quan s'ha escollit la opció de afegir articles al plugin.
        if (isset($_POST['form'])) {
            if(isset($_POST['nb'])) {
                $site = new SiteH();
                //nb2 = comptador dels posts que es mostren.
                $nb2 = $_POST['nb'];
                //Consulta de recuperació del número d'articles que hi ha
                actualment al plugin.
                $sql_comp = "SELECT COUNT(*)
                    FROM ".$wpdb->prefix."posts
                    WHERE post_type='post' AND
                    post_status='publish'";

                $cnt = 1;
```

```

//Creem un array de 5 elements per guardar les rellevàncies
//que ja han estat introduïdes. Inicialment tindran totes les
//relevàncies valor '0', que vol dir que no s'han introduït a cap
//article.
while ($cnt <= 5){
    $arr_rel[$cnt] = '0';
    $cnt++;
}
//Consulta de recuperació de les rellevàncies ja introduïdes.
$sql_r = "SELECT meta_value
        FROM ".$wpdb->prefix."postmeta
        WHERE meta_key = 'relevancia'";
$rest = $wpdb->get_results($sql_r);
//Marquem a 1 les rellevàncies que ja hi són introduïdes.
foreach ($rest as $rstd){
    $arr_rel[$rstd->meta_value] = '1';
}
//Número d'articles que hi ha actualment.
$comp = $wpdb->get_var($sql_comp);

//Recorregut per tots els articles del formulari per cercar
//els que han estat seleccionats.
$blog = $_POST['filtra'];
if($blog != 0) $site->Recull_blog($blog);
for($in=1; $in<$nb2; $in++){
    //Cas en el que es trobi amb un article seleccionat.
    if(isset($_POST['seleccionat_'].$in)){
        $seleccionat = "si";
        //Cas en que tingui una rellevància introduïda.
        if(isset($_POST['rel_'].$in)){
            //Guardem la rellevància de l'article. Valor entre 1 i 5.
            $rel = $_POST['rel_'].$in;
            if($rel!=0){
                //La rellevància ja ha estat introduïda anteriorment.
                if($arr_rel[$rel] == '1'){
                    //Marquem que hi ha un error de rellevància ja introduïda.
                    $err_relint = 1;
                }
                //Rellevància lliure.
                else{
                    //No hi ha error de rellevància introduïda.
                    $err_relint = 0;
                    //Marquem la nova rellevància com introduïda.
                    $arr_rel[$rel] = '1';
                }
            }
        }
    }
}
//Cas en el que no s'ha introduït cap rellevància per l'article.
else{
    //Marquem que hi ha un error de rellevància no introduïda.
    $err_relint = 2;
}

//Guardem l'identificador de l'article.
if(isset($_POST['idpost_'].$in))
    $post = $_POST['idpost_'].$in;
//Guardem l'identificador del blog.
if(isset($_POST['idblog_'].$in))
    $blog = $_POST['idblog_'].$in;

//ES del cas d'us. Recollida de blog i selecció de l'article

```

```

//corresponent.
if((isset($_POST['filtra']) &&
($_POST['filtra'] == 0)) or !isset($_POST['filtra']))
    $site->Recull_blog($blog);
    $site->Selecciona_article($post);
    //Consulta que comprova si es troba l'article ja
    //introduït al plugin.
    $pnuevo = new Post($post);
    switch_to_blog(1);
    $sql_comprova = "SELECT COUNT(*) AS comp
                    FROM ".$wpdb->prefix."posts
                    WHERE post_title = '".
                        $pnuevo->getTitol()."
                    and post_date='". $pnuevo->getData()."
                    and post_author='". $pnuevo->getAutor()-
                        >getId()." and post_status = 'publish'";
    //Shies guarda si l'article és o no dintre del plugin (0 = no).
    $shies = $wpdb->get_var($sql_comprova);
    restore_current_blog();
    //Cas en que no hi hagin errors.
    if($comp < 5 && $rel!=0 && $shies == 0 && $err_relint == 0){
        //Introduïm la rellevància a l'article.
        $site->Selecciona_rellevancia($rel);
    //Cas en que no hi hagi un filtrar per comisions, finalitza el CU.
    if((isset($_POST['filtra']) && ($_POST['filtra'] == 0))
    or !isset($_POST['filtra']))
        $site->Fi_seleccio();
    }
    //Cas en que hi hagi errors.
    else{
        echo "<font color='red'><b>";
        //Error de màxim d'articles ja introduït.
        if($comp >= 5)
            echo "La quantitat de posts actuals a la pàgina
principal ja s'acaba de 5, elimina posts per afegir-ne de nous<br>";
        //Error d'article sense rellevància.
        if($rel==0)
            echo "Has intentat introduir un post sense
rellevància<br>";
        //Error d'article ja existent al plugin.
        if($shies!=0)
            echo "Ja existeix aquest post a la pàgina
principal<br>";
        //Error d'article amb rellevància ja introduïda.
        if($err_relint == 1){?>
            <script type="text/javascript">
                alert("Ja existeix un post amb aquesta rellevància")
            </script>
            <?php
        }
        //Error d'article sense rellevància.
        if($err_relint == 2){?>
            <script type="text/javascript">
                alert("Post sense rellevància")
            </script>
            <?php
        }
        echo "</b></font>";
    }
}
}
}

```

```

//Cas de filtració per comissió. Es crida al mètode Fi_seleccio un cop
//s'han introduït tots els articles.
    if(((isset($_POST['filtra'])) && ($_POST['filtra'] != 0)
        && $seleccionat == "si")
        or ($c != 0 && $seleccionat == "si")){
        $site->Fi_seleccio();
    }
    //Crida a la destructora de Site.
    unset($site);
}
} //Fi de l'acció afegir articles.

//Acció d'esborrar articles del plugin.
if (isset($_POST['form2'])) {
    $form2 = $_POST['form2'];
    if(isset($_POST['a'])) {
        //Nombre d'articles del plugin (més 1).
        $a2 = $_POST['a'];
        //Recorregut per els articles del plugin per comprovar si tenen
        //actiu el check d'esborrar.
        for($sil=1; $sil<$a2; $sil++){
            //Cas en que estigui seleccionat l'article.
            if(isset($_POST['elim_.$sil'])){
                if(isset($_POST['id_.$sil'])){
                    //Identificador de l'article.
                    $id = $_POST['id_.$sil'];
                    $site = new SiteH();
                    $site->Esborra_article($id);
                }
            }
        }
    }
} //Fi acció d'esborrar articles.
switch_to_blog(1);
//Array amb la informació de les comissions.
$sql_bl = "SELECT blog_id
          FROM ".$wpdb->prefix."blogs
          WHERE blog_id != 1;";
//Array amb els identificadors de cadascun dels blogs existents.
$ar_blogs = $wpdb->get_results($sql_bl);
foreach ($ar_blogs as $z){
    //Creem un array amb tota la informació de cadascun dels blogs.
    $arr_blogs[$z->blog_id] = new BlogH($z->blog_id);
}

//Paginació.
$i2 = 2;
//Número total d'articles.
$totale = 0;
//Cas en que no hi hagi cap filtre.
if(!isset($_POST['filtra']) || $_POST['filtra'] == 0){
    //Contem el número total d'articles de la blogosfera.
    foreach ($ar_blogs as $za){
        switch_to_blog($za->blog_id);
        $sqlcontapost = "SELECT COUNT('id') as num
                        FROM ".$wpdb->prefix."posts
                        WHERE post_status = 'publish'
                        and post_type = 'post';";
        $rst = $wpdb->get_var($wpdb->prepare($sqlcontapost));
        $totale += $rst;
    }
}

```

```

        restore_current_blog();
    }
}
//Cas en que hi hagi algun filtre.
else if($_POST['filtra'] != 0){
    //Contem el nombre d'articles del filtre.
    $c = $_POST['filtra'];
    switch_to_blog($c);
    $sqlcontapost = "SELECT COUNT('id') as num FROM ".
        $wpdb->prefix."posts
        WHERE post_status = 'publish'
        and post_type='post'";
    $totale = $wpdb->get_var($wpdb->prepare($sqlcontapost));
    restore_current_blog();
}
//Articles per plana.
$postpag = 5;
//Número total de planes.
$pags = intval($totale / $postpag);
if(($totale % $postpag) != 0){
    $pags += 1;
    //Articles que van a l'última plana.
    $postultpag = ($totale % $postpag);
}
//Comptadors.
$i2 = 1;
//Comptador de tots els posts.
$nb = 1;
$in = 0;

//Plana actual.
if(isset($_POST['pag'])) $pagact = $_POST['pag'];
else $pagact = 1;
//Limit d'articles a mostrar a la consulta.
if($pagact == 1 or $pagact == 0) $lim = 0;
else $lim = ($pagact*$postpag)-$postpag;
$limfin = $lim+$postpag;
if($lim > $totale+1 or $limfin > $totale){
    $limfin = $totale;
    $lim = $totale - $postultpag;
}
//Fi paginació.
//Formulari d'adició d'articles al plugin.
?>
<h3> Selecciona els posts que vulguis que surtin a la pàgina
principal</h3><p>
<font size="-3"><i>Mostran&#39;t posts del <?php echo ($lim+1).> fins al
"$.limfin." de "$totale?></i></font>
<form name="totposts" action="" method="post">
<table border="0">
<tr><td colspan="4"><?php
//Menú desplegable per la filtració per comissions.
echo "<select name='filtra' onchange='this.form.submit()'>
<option value='0' ">
//Cas en que estigui seleccionada l'opció "Totes les Comissions".
if($comi == 0) echo " selected ";
echo ">Totes les Comissions</option>";
$om = 2;
//Mostrem la resta de blogs amb value igual al seu identificador de blog.
foreach ($arr_blogs as $zz){
    echo "<option value='".$zz->getId().">";

```

```

        if($c == $zz->getId()) echo " selected ";
        echo ">". $zz ->getNom(). "</option>";
    }
    echo "</select>";
?>
</td></tr>
<tr>
    <td><b>T&iacute;tol</b></td>
    <td><b>Comissi&oacute;</b></td>
    <td align="center"><b>Rellev&agrave;ncia</b></td>
    <td><b>Selecciona</b></td></tr>
<?php
//Acció de filtració per comissió.
//Seleccionem els articles de la comissió corresponent.
if(isset($_POST['filtra']) and ($_POST['filtra'] != 0)){
    $comi = $_POST['filtra'];
    $comision = $comi;
    switch_to_blog($comi);
    $sqla = "SELECT id, post_date, '". $comi.'" as nblog
            FROM ".$wpdb->prefix."posts
            WHERE post_status = 'publish' and
            post_type = 'post'";
    restore_current_blog();
}
//Sense filtració per comissió.
else{
    $sql_bl = "SELECT blog_id
            FROM ".$wpdb->prefix."blogs
            WHERE blog_id != 1";
    //Array amb els identificadors de cadascun dels blogs existents.
    $ar_blogs = $wpdb->get_results($sql_bl);
    $cnt = 1;
    //Per cada comissió, recuperem els articles.
    foreach ($ar_blogs as $b){
        //Fem el canvi de blog per recollir els articles.
        switch_to_blog($b->blog_id);
        //Cas en que tenim més d'una comissió, fem un UNION
        //per ercuperar tots els articles.
        if ($cnt != 1) $sqla .= " UNION (";
        //Consulta de recuperació d'articles.
        $sqla .= "SELECT id, post_date, '". $b->blog_id.'" as nblog
                FROM ".$wpdb->prefix."posts
                WHERE post_status = 'publish' and
                post_type = 'post' ";
        restore_current_blog();
        if ($cnt != 1) $sqla .= " ) ";
        $cnt++;
    }
}
$sqla .= " ORDER BY post_date DESC
        LIMIT ".$lim." , ".$postpag." ";
$result = $wpdb->get_results($sqla);
//Comptador per ficar els cinc primers articles a un array.
$cntcinc = 0;
foreach ($result as $resultado){
    $s2 = new SiteH();
    $s2->Recull_blog($resultado->nblog);
    $s2->Selecciona_article($resultado->id);
    //$pt = new Post2($resultado->id, $resultado->nblog);

    $pt = $s2->getP();

```

```

$blogpt = $s2->getB();
//Guardem els cinc articles més nous per el cas en que no hi hagi
//articles a la pàgina principal.
if($cntcinc < 5){
    $sarrcinc[$cntcinc] = $pt;
    $cntcinc++;
}
echo "<tr><td>".$pt->getTitol()."</td><td>".
    $blogpt->getNom()."</td>";
//Selecció de la rellevància.
echo "<td align='right'><select name='rel_'. $nb.'">
    <option value='0' selected>&nbsp;</option>
    <option value='1'>&nbsp;1&nbsp;</option>
    <option value='2'>&nbsp;2&nbsp;</option>
    <option value='3'>&nbsp;3&nbsp;</option>
    <option value='4'>&nbsp;4&nbsp;</option>
    <option value='5'>&nbsp;5&nbsp;</option>

    </select></td>";
echo "<td align='right'><input type='checkbox'
name='seleccionat_'. $nb.'" value='". $nb.'"></td></tr>";
echo "<input type='hidden' name='idpost_'. $nb.'"
value='". $resultado->id.'">";
echo "<input type='hidden' name='idblog_'. $nb.'"
value='". $resultado->nblog.'">";
//Aumentem el comptador d'articles.
$nb++;
unset($s2);
}
//Passem un camp ocult per identificar el formulari corresponent.
echo "<input type='hidden' name='form' value='true'>";
//Enviem el nombre de blogs que hem mostrat fins ara.
echo "<input type='hidden' name='nb' value='". $nb.'">";
?>
<tr><td><b>P&agrave;gina</b></td><td colspan="3">&nbsp;</td></tr>
<?php
//Paginació (formulari).
echo "<tr>
    <td>
        <select name='pag' onchange='this.form.submit()'>
            <option value='0' ";
            if(($pagact == 0) or (!isset($_POST['pag']))) echo " selected ";
            echo ">&nbsp;</option>";
            $cont = 1;
            while($cont <= $pags){
                echo "<option value='". $cont.'" ";
                if($cont == $pagact) echo " selected ";
                echo ">". $cont."</option>";
                $cont++;
            }
        echo "</select>
    </td>
    <td colspan='3' align='right'>
        <input type='submit' class='button-primary'
value='Acceptar'>
    </td>
</tr>";

?>
</table>
</form>

```



```

<?php
//Fi formulari d'adició d'articles al plugin.
//Recuperem els articles actuals que hi son al plugin.
$sqlrec = "SELECT DISTINCT *
          FROM ".$wpdb->prefix."posts as p INNER JOIN ".$wpdb->
>prefix."postmeta as pm
          ON p.id = pm.post_id
          WHERE p.post_type='post' and p.post_status = 'publish'
          ORDER BY pm.meta_value DESC";

$array_posts = $wpdb->get_results($sqlrec);
?>
<h3> Posts que surten actualment a la pàgina principal</h3><p>
<?php
//Formulari de mostra d'articles i possibilitat d'esborrar.
?>
    <form method="post" action="" name="esobo">
        <table border="1">
            <tr><td><b>Titol</b></td><td><b>Comissió</b></td><td><b>Rellev
&agrave;ncia</b></td><td><b>Esborrar</b></td></tr>
            <?php
                $a = 1;
                //Recorregut per cadascun els articles del plugin per recuperar tota
la informació.
                foreach ($array_posts as $post){
                    if($post->meta_key == "relevancia") $rel = $post->meta_value;
                    else $rel = 0;
                    if($rel != 0){
                        $sql_cm = "SELECT meta_value
                                FROM ".$wpdb->prefix."postmeta
                                WHERE post_id = ".$post->ID." AND meta_key =
'comissio'";
                        $com = $wpdb->get_var($sql_cm);
                        echo "<tr><td>".$post->post_title."</td><td>".$com."</td><td
align='center'>".$rel."</td><td align='right'><input
name='elim_".$a."'></td></tr>";
                        echo "<input type='hidden' name='id_".$a."' value='".$post-
>ID."'>";
                        $a++;
                    }
                }
            <?php
                echo "<input type='hidden' name='form2' value='true'>";
                echo "<input type='hidden' name='a' value='".$a."'>";
            ?>
            <tr><td colspan="4" align="right"><input class="button-primary"
type="submit" value="Esborra"></td></tr>
        </table>
    </form>
</p>

<?php
//Fi formulari d'esborrar articles.

}
//Fi cos del plugin.
}
?>

```

12.2.2 Classe site (classSite.php)

Classe corresponent al controlador de l'aplicació. No s'han inclòs els mètodes getter i setter.

La classe al complet amb tots els mètodes es troba a l'arxiu del plugin.

```
<?php
/**
 * Classe controlador.
 */
class SiteH{

    //Atribut privat que correspon a un objecte Blog.
    private $b;
    //Atribut privat que correspon a un objecte Post.
    private $p;

    /**
     * Mètode constructor, crea un nou controlador.
     */
    public function __construct(){
    }

    /**
     * Mètode destructor, destrueix el controlador i torna
     * el blog al seu estat inicial.
     */
    public function __destruct(){
        restore_current_blog();
    }

    /**
     * Mètode de recollida d'un nou blog a partir del identificador.
     */
    public function Recull_blog($id){
        $this->b = new BlogH($id);
        switch_to_blog($this->b->getId());
    }

    /**
     * Mètode per afegir el pare a un blog.
     */
    public function Afegir_pare($id){
        $pare = new BlogH($id);
        $this->b->Afegir_pare($pare);
    }

    /**
     * Mètode per afegir l'ordre d'un blog al menú.
     */
    public function Afegir_ordre($ord){
        $this->b->Afegir_ordre($ord);
    }

    /**
     * Mètode per recollir un article.
     */
    public function Selecciona_article($idArt){
        $this->p = new Post($idArt);
    }
}
```

```

/**
    Mètode per introduir una rellevància a un article i
    preparar l'article per emmagatzemar.
*/
public function Selecciona_rellevancia($rel){
    $this->p->Selecciona_rellevancia($rel);
    $this->b->Introduir_article($this->p);
}

/**
    Mètode per finalitzar la selecció de posts rellevants.
*/
public function Fi_seleccio (){
    $bl = new BlogH(1);
    $this->b->Fi_seleccio($bl);
}

public function Esborra_article($idArt){
    $this->Recull_blog(1);
    $this->Selecciona_article($idArt);
    $this->b->Esborra_article($this->p);
}
}
?>

```

12.2.3 Classe blog (classBlog.php)

Classe corresponent al component Blog. És el mateix component que al anterior plugin però amb els mètodes que pertanyen al corresponent plugin. També s'ha omès a aquest anexe els mètodes getters i setters.

L'arxiu *classBlog.php* realment conté tots els mètodes del component.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Blog. Conté les dades del blog.
 * Dades del blog:
 *   @id: identificador del blog a transformar.
 *   @path: ruta URL d'accés al blog.
 *   @nom: títol del blog.
 *   @pare: blog del que depèn.
 *   @fills: array amb els blogs que depenen d'ell.
 *   @vist: flag que indica si el blog ha sigut pintat o
 *   no al menú (0 = no pintat).
 */

class BlogH{
    private $id;
    private $nom;
    private $path;
    private $pare;
    private $ordre;
    private $fills;
    private $vist;
    private $post_rellevants = array ();

    /**
     * Mètode constructor, crea un nou Blog.
     * @id: id del blog a crear.
     * @pa: path del blog (opcional).
     */
    public function __construct($id){
        $this->id = $id;
        $this->nom = BlogH::name();
        $this->path = BlogH::path();
        $this->pare = BlogH::pare();
        $this->ordre = BlogH::ordre();
        $this->vist = 0;
    }

    /**
     * Mètode que introdueix un article a
     * l'array de posts rellevants.
     */
    public function Introduir_article ($art){
        //Array_push: funció php que introdueix un objecte
        //dintre d'un array.
        array_push($this->post_rellevants, $art);
    }

    /**
     * Mètode per cridar al blog que ha d'emmagatzemar els
     * articles rellevants.
     */
    public function Fi_seleccio ($bl){
        $bl->Finalitza($this->post_rellevants);
    }
}
```

```

/**
    Mètode que introdueix els articles rellevants al blog
    principal.
*/
public function Finalitza ($posts){
    //Per cada article contingut a l'array de post rellevants,
    //l'introdueix al blog.
    foreach ($posts as $post){
        $cont = $post->getContingut();
        $tit = $post->getTitol();
        $autor = $post->getAutor()->getId();
        $rele = $post->getRellevancia();
        $da = $post->getData();
        switch_to_blog($this->id);
        //Creació de l'article al estil WordPress.
        $pnuevo = array('post_title' => $tit,
            'post_content' => $cont,
            'post_status' => 'publish',
            'post_author' => $autor,
            'post_date' => $da);
        $b = new BlogH($post->getBlog());
        //Insercció de l'article amb funció de WordPress.
        $idnou = wp_insert_post($pnuevo);
        //Afegit d'informació adicional: rellevància, id i nom de
        //la comissió a la que pertany l'article.
        add_post_meta($idnou, 'relevancia', $rele);
        add_post_meta($idnou, 'comissio', addslashes($b->nom));
        add_post_meta($idnou, 'idcomissio', $post->getBlog());
        $imat = $post->getImatge();
        if(count($imat)!= 0){
            $idthumb = rand(300,500);
            add_post_meta($idnou, '_thumbnail_id', $idthumb);
            add_post_meta($idthumb, '_wp_attached_file', $imat[0]);
            add_post_meta($idthumb, '_wp_attachment_metadata', $imat[1]);
        }
        restore_current_blog();
    }
}

public function Esborra_article($art){
    //Esborrem la informació de la relevancia.
    delete_post_meta($art->getId(), 'relevancia');
    //Esborrem la informació del nom de la comissió a la que pertany.
    delete_post_meta($art->getId(), 'comissio');
    //Esborrem la informació de l'identificador de la comissió a la
    //que correspon.
    delete_post_meta($art->getId(), 'idcomissio');
    //Esborrem la imatge destacada
    delete_post_meta($art->getId(), '_thumbnail_id');
    //Eliminem l'article, forçant l'eliminació.
    wp_delete_post($art->getId(), true);
    global $wpdb;
    $sql = "SELECT COUNT(id) as n
        FROM ".$wpdb->prefix."posts
        WHERE post_status = 'publish' and post_type='post'";
    $n = $wpdb->get_var($sql);
    if($n == 0) $this->Insereix_art_no_rellevants();
}

```

```

public function Insereix_art_no_rellevants(){
    global $wpdb;
    $sql_bl = "SELECT blog_id
                FROM ".$wpdb->prefix."blogs
                WHERE blog_id != 1;";
    //Array amb els identificadors de cadascun dels blogs existents.
    $ar_blogs = $wpdb->get_results($sql_bl);
    $cnt = 1;
    //Per cada comissió, recuperem els articles.
    foreach ($ar_blogs as $b){
        //Fem el canvi de blog per recollir els articles.
        switch_to_blog($b->blog_id);
        //Cas en que tenim més d'una comissió, fem un UNION
        //per ercuperar tots els articles.
        if ($cnt != 1) $sql_t .= " UNION (";
        //Consulta de recuperació d'articles.
        $sql_t .= "SELECT id, post_date, '".$b->blog_id.'" as nblog
                    FROM ".$wpdb->prefix."posts
                    WHERE post_status = 'publish'
                    and post_type = 'post' ";
        restore_current_blog();
        if ($cnt != 1) $sql_t .= " ) ";
        $cnt++;
    }
    //Ordenem la consulta per data, de més actuals a més antics
    //i recuperem només els cinc primers.
    $sql_t .= " ORDER BY post_date DESC
                LIMIT 0, 5;";

    //Recuperem els resultats de la consulta.
    $cinc = $wpdb->get_results($sql_t);
    //Creem una variable que es decrementarà per guardar les rellevàncies.
    $cnnt = 5;
    //Guardem els articles amb les rellevàncies que corresponen segons la
data.
    foreach ($cinc as $c){
        $site = new Site();
        $site->Recull_blog($c->nblog);
        $site->Selecciona_article($c->id);
        $site->Selecciona_rellevancia($cnnt);
        $site->Fi_seleccio();
        $cnnt--;
    }
}
?>

```

12.2.4 Classe post (classPost.php)

Classe corresponent al component Post.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Post. Conté les dades de l'article.
 * Dades de l'article:
 *   @id: identificador de l'article a transformar.
 *   @Data: data en la que es va crear l'article.
 *   @Titol: títol de l'article.
 *   @Rellevancia: importància que té l'article dintre de la pàgina
principal.
 *   @Autor: persona que ha escrit l'article.
 *   @Contingut: cos de l'article.
 *   @Blog: comissió a la que pertany l'article.
 */

class Post {

    private $id;
    private $Data;
    private $Titol;
    private $Rellevancia;
    private $Autor;
    private $Contingut;
    private $Blog;
    private $Imatge;

    /**
     * Mètode constructor, crea un nou Post.
     * @id: id de l'article a crear.
     */
    public function __construct($id){

        $this->id = $id;
        $this->Data = Post::data();
        $this->Titol = Post::tit();
        $this->Autor = Post::aut();
        $this->Contingut = Post::cont();
        global $blog_id;
        $this->Blog = $blog_id;
        $this->Imatge = Post::imatge();
    }

    /**
     * Mètode que introdueix la rellevància al post.
     */
    public function Selecciona_rellevancia ($rel){
        $this->Rellevancia = $rel;
    }

    /**
     * Mètode que cerca a base de dades la data de publicació
de l'article.
     */
    private function data (){
        global $wpdb;
        $sql_data = "SELECT post_date
                    FROM ".$wpdb->prefix."posts
```

```

        WHERE id = '". $this->id. "'";
    return $wpdb->get_var($sql_data);
}

/**
 * Mètode que cerca el títol de l'article a base de dades.
 */
private function tit (){
    global $wpdb;
    $sql_nom = "SELECT post_title
                FROM " . $wpdb->prefix. "posts
                WHERE id = '". $this->id. "'";
    return $wpdb->get_var($sql_nom);
}

/**
 * Mètode que cerca el contingut a base de dades.
 */
private function cont(){
    global $wpdb;
    $sql_cont = "SELECT post_content
                FROM " . $wpdb->prefix. "posts
                WHERE id = '". $this->id. "'";
    return $wpdb->get_var($sql_cont);
}

/**
 * Mètode que cerca les imatges destacades a la base de dades i les
 * introdueix al post de la home.
 */
private function imatge(){
    global $wpdb;
    $metas = array();
    $sql_idimg = "SELECT meta_value
                FROM " . $wpdb->prefix. "postmeta
                WHERE meta_key = '_thumbnail_id' and post_id = " . $this->id. "'";
    $mv = $wpdb->get_var($sql_idimg);
    $sql_img = "SELECT meta_key, meta_value
                FROM " . $wpdb->prefix. "postmeta
                WHERE post_id = " . $mv. "'";
    $atr = $wpdb->get_results($sql_img);
    foreach ($atr as $a){
        if($a->meta_key == "_wp_attached_file"){
            $at = $a->meta_value;
            array_push($metas, $at);
        }
        else{
            $am = $a->meta_value;
            array_push($metas, $am);
        }
    }
    return $metas;
}

public function getImatge(){
    return $this->imatge;
}
}
?>

```


12.2.5 Pàgina principal del Theme, modificada per al plugin (index.php)

```
<!-- Cos. -->
<div id="body-bg" class="clear">
  <div id="body" class="content">
    <?php
      //Menú lateral de l'esquerra.
      get_sidebar();
      global $blog_id;
    //Cas que es tracti del blog principal.
    if($blog_id == 1){ ?>
      <!-- Cos de la pàgina. -->
      <div id="content">
        <?php $slider = ddp('slider'); if ($slider == 'on') { ?>
          <div id="slider-bg" class="box">
            <?php //Slider. ?>
            <div id="slider">
              <ul>
                <?php
                  $a = array('meta_key' => 'relevancia',
                            'orderby' => 'meta_value',
                            'order' => 'DESC');
                  query_posts($a);
                  if (have_posts ()) {
                    while (have_posts()) : the_post();
                      ?>
                      <li>
                        <?php
                          global $post;
                          $tmp_post = $post;
                          $data = $post->post_date;
                          $autor = $post->post_author;
                          //Recuperació de la comissió de l'article.
                          $value = get_post_custom_values('idcomissio', $post->ID);
                          $Thumbnailid = get_post_meta($post->ID, '_thumbnail_id',
                                                         $single = true);
                          $thumbnail = get_post_meta($Thumbnailid, '_wp_attached_file',
                                                         $single = true);
                          $com = get_post_custom_values('idcomissio', $post->ID);
                          $url = "/wp-content/blogs.dir/" . $com[0] . "/files/" . $thumbnail;
                          global $blog_id;
                          switch_to_blog($value[0]);
                          //Consulta que ens recuperarà l'identificador de l'article original.
                          $sql_post = "SELECT id FROM ".$wpdb->prefix."posts
                                      WHERE post_status = 'publish' and
                                      post_date = '" . $data . "' and post_author =
                                      '" . $autor . "'";
                          $idp = $wpdb->get_var($sql_post);
                          $incl = array( 'include' => $idp);
                          $myp = get_posts($incl);
                          foreach($myp as $post){
                            setup_postdata($post);
                            $hometempl = get_theme_root_uri();?>
                            <div>
                              
                            </div>
                            <!-- Títol amb l'enllaç a l'article. -->
                            <h2>
                              <a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
                            </h2>
                        <?php
                    }
                  }
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <!-- Extracte de l'article. -->
        <p><?php echo substr(get_the_excerpt(), 0, 50).' [...]'; ?></p>
        <?php}
        restore_current_blog();
    ?>
</li>
<?php
    endwhile;
}?>
</ul> </div> <!-- Fi de l'slider. -->
</div>
<?php }
//Cos de la pàgina principal.
//Recuperació dels articles del plugin.
//Consulta amb els articles amb relevància ordenats de major a menor.
wp_reset_query();
$a = array('meta_key' => 'relevancia',
           'orderby' => 'meta_value',
           'order' => 'DESC');
query_posts($a);
if (have_posts ()) {
    while (have_posts()) : the_post();
        global $post;
        $tmp_post = $post;
        $nonnull = 0;
        $data = $post->post_date;
        $autor = $post->post_author;
        $Thumbnailid = get_post_meta($post->ID, '_thumbnail_id',
                                     $single = true);
        if($Thumbnailid != null){
            $thumbnail = get_post_meta($Thumbnailid, '_wp_attached_file',
                                       $single = true);
            $com = get_post_custom_values('idcomissio', $post->ID);
            $url = "/wp-content/blogs.dir/" . $com[0] . "/files/" . $thumbnail;
            $nonnull = 1;
        }
        //Recuperació de la comissió de l'article.
        $value = get_post_custom_values('idcomissio', $post->ID);
        global $blog_id;
        switch_to_blog($value[0]);
        //Consulta que ens recuperarà l'identificador de l'article original.
        $sql_post = "SELECT id FROM ".$wpdb->prefix."posts
                    WHERE post_status = 'publish' and
                    post_date = '". $data.'" and post_author = '". $autor.'";";
        $idp = $wpdb->get_var($sql_post);
        $incl = array( 'include' => $idp);
        $myp = get_posts($incl);
        foreach($myp as $post){
            setup_postdata($post);
            ?>
            <!-- Article -->
            <div class="post box">
            <!-- Títol -->
                <h3 class="post-title">
<a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h3>
            <!-- Metainformació -->
            <!-- Imatge destacada -->
            <div class="post-thumbnail">
                <?php if($nonnull == 1){ ?>
                

```

```

        <?php } ?>
    </div>
    <!-- Contingut -->
    <?php $excerpt = ' <p>' . get_the_excerpt() . ' ...</p>';
    echo $excerpt; ?>
    <a class="more-link" href="<?php the_permalink(); ?>">Llegeix Més</a>
    </div> <!-- // Fi article. -->
    <?php
    }
    restore_current_blog();
    $post = $tmp_post;
endwhile;
} else {
    //Cas en que no hi hagin articles a mostrar.
    //Mostrem un missatge conforme no hi ha articles.
?>
    <div class="post box"> <h3>No hi ha articles disponibles.</h3> </div>
<?php } ?>
</div>
<!-- // Fi del cos de la pàgina. -->

<?php
}
//Cas dels blogs normals.
else{ ?>
    <!-- Cos de la pàgina. -->
    <div id="content">
        <?php $slider = ddp('slider'); if ($slider == 'on') { ?>
        <div id="slider-bg" class="box">
            <!-- Slider. -->
            <div id="slider">
                <ul>
                    <?php
                    $slider_posts = new WP_Query();
                    $slider_posts->query('showposts=' . ddp('slider_count') . '&cat=' .
                        ddp('categories'));
                    while ($slider_posts->have_posts()) {
                        $slider_posts->the_post();
                        ?>
                        <li><?php the_post_thumbnail(array(470, 263)); ?>
                        <h2><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
                        <p><?php echo substr(get_the_excerpt(), 0, 50). ' [...]'; ?></p>
                        </li>
                    <?php } ?>
                </ul>
            </div>
        <!-- Fi slider. -->
    </div>
<?php }
//Cos de la pàgina principal.
//Mostra els articles en ordre cronològic invers.
$slider_exclude = ddp('slider_exclude'); if ($slider_exclude == 'on')
    query_posts($query_string . '&cat=-' . ddp('categories'));
//Loop de WordPress.
if (have_posts ()) {
    while (have_posts ()) {
        (the_post());?>
        <!-- Article -->
        <div class="post box">
            <!-- Títol -->
            <h3 class="post-title"><a href="<?php the_permalink(); ?>"

```

```

        <?php the_title(); ?></a></h3>
        <!-- Metainformació -->
        <div class="post-meta">
            <?php the_author_posts_link(); ?>
            <span class="post-category"><?php the_category(', ');
?></span>

            <span class="post-date"><?php the_time('F d, Y');
?></span>

            <span class="post-comments"><a href="<?php
comments_link(); ?> "><?php comments_number('Sense comentaris', 'Un comentari',
'% comentaris'); ?></a></span>
        </div>
        <!-- Imatge destacada -->
        <div class="post-thumbnail">
            <?php the_post_thumbnail(array(141, 141)); ?>
        </div>
        <!-- Contingut -->
        <?php $excerpt = '<p>' . get_the_excerpt() . ' ...</p>';
        echo $excerpt; ?>
        <a class="more-link" href="<?php the_permalink(); ?>">Llegeix Més</a>
    </div> <!-- // Fi article. -->
<?php
    } ?>
    <div class="post-nav">
        <?php next_posts_link('&laquo; Older Entries');
        previous_posts_link('Newer Entries &raquo;'); ?>
    </div>
<?php
    } else { ?>
        <div class="post box">
            <h3>No hi ha articles disponibles.</h3>
        </div>
    </div>
    <?php
    } ?>
    </div>
    <!-- // Fi del cos. -->
<?php
}
get_sidebar('secondary'); ?>
    </div>
    <!-- // Fi de la barra lateral dreta. -->
</div>
<?php
//Peu de la pàgina.
get_footer();
?>
<script type="text/javascript">

    var _gaq = _gaq || [];
    _gaq.push(['_setAccount', 'UA-6724405-14']);
    _gaq.push(['_trackPageview']);

    (function() {
        var ga = document.createElement('script'); ga.type = 'text/javascript';
ga.async = true;
        ga.src = ('https:' == document.location.protocol ? 'https://ssl' :
'http://www') + '.google-analytics.com/ga.js';
        var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
    })();
</script>
</body>
</html>

```

12.3 Plugin Quickr To *WordPress*

12.3.1 Classe principal del plugin (quickrtowp.php)

```
<?php
/*
Plugin Name: Quickr to WordPress
Plugin URI: URL
Description: Xuclador de blogs de Quickr
Author: David G&oacute;mez
Version: 2.0
Author URI: URL
*/

class xuclablog{

    var $opt;
    var $table_name;

    // Mètode constructor del plugin.
    function xuclablog() {
        //Variable global de WordPress per permetre l'accés a base de dades.
        global $wpdb;
        //Nom de la taula on aniran les dades del plugin Quickr.
        $this->table_name = $wpdb->prefix.'xucla';
        //Accés a la instal·lació del plugin.
        register_activation_hook(__FILE__, array(&$this, 'install'));
        //Accés a la desinstal·lació del plugin.
        register_deactivation_hook(__FILE__, array(&$this, 'uninstall'));
        //Posició del plugin al menú.
        add_action('admin_menu', array(&$this, 'config_page'));
    }

    // Instal·lació del plugin.
    function install() {
        //Creem una taula amb els identificadors dels feeds de Quickr.

        global $wpdb;
        /**Consulta de creació de taula. Camps necessaris:
            idLoc: autonuméric. Identifica el blog.
            loc: text. Localització del feed.
            nom: text. Nom del feed corresponent.
        */
        $sql_create = "CREATE TABLE wp_xucla
            (idLoc bigint(20) auto_increment ,
            loc varchar(150),
            nom varchar(200),
            PRIMARY KEY (idLoc));";
        $wpdb->query($sql_create);
        //Array inicial de valors predefinitos de la taula.
        $array_loc[0]['clave'] = "37f8fd004dd902038bcfcf36e2c7a147";
        $array_loc[0]['valor'] = "Prova0";
        //Consulta d'insercció a la taula per cada element de l'array.
        foreach($array_loc as $ar){
            $sql_afegir = "INSERT
                INTO ".$wpdb->prefix."xucla (loc, nom)
                VALUES ('".$ar['clave']."', '".$ar['valor']."');";
            $wpdb->query($sql_afegir);
        }
    }
}
```

```

    }
}

// Desinstal·lació del plugin.
function uninstall() {
    //Esborrem la taula que hem creat anteriorment.
    global $wpdb;
    $sql_elim = "DROP TABLE ".$wpdb->prefix."xucla;";
    $wpdb->query($sql_elim);
}

// Pàgina de configuració.
function config_page() {
    /**Definició de les opcions de configuració:
    - Títol de la pàgina.
    - Títol del menú.
    - Nivell d'usuari que pot fer ús. 7 = editor.
    - Nom del menú.
    - Acció del menú. En el nostre cas crida a la funció options_xucla.
    */
    add_options_page('Quickr to WordPress',
        'Quickr to WordPress',
        7,
        'Quickoptions',
        array(&$this, 'options_xucla'));
}

//Cos del plugin
function options_xucla() {
    //Definim la variable global d'accés a base de dades per quan ens
    //faci falta.
    global $wpdb;
    //Iniciem variables sessió per mantenir actiu el login i password
    //que usarem a Quickr.
    $_SESSION['id'] = 'wpsadmin';
    $_SESSION['pass'] = 'MANIPULACION78';

    //En el cas que ja haguem seleccionat l'origen del blog Quickr, en
    // guardem l'identificador.
    if(isset($_POST['com'])) $com = $_POST['com'];
    if(isset($_GET['com'])) $com = $_GET['com'];

    ?>
    <h3>Selecciona l'origen dels blogs</h3>
    <!-- Formulari de selecció de blog Quickr -->
    <form method="POST">
    <table border="0">
        <tr><td><select name="com" onChange="this.form.submit()">
        <?php
            if(!isset($_POST['com']) and !isset($com))
                echo "<option value=0 selected>&nbsp;</option>";
            //Recuperació del blogs guardats.
            switch_to_blog(1);
            $sql_recid = "SELECT * FROM ".$wpdb->prefix."xucla;";
            $resul = $wpdb->get_results($sql_recid);
            restore_current_blog();
            foreach ($resul as $res){
                echo "<option value=".$res->loc;
                if($com == $res->loc) echo " selected ";
            }
        }
    </td></tr>
    </table>
    </form>
}

```

```

        echo ">". $res->nom. "&nbsp;"</option>";
        //Guardem en un array (id) el id i localitzador del blog.
        //Clau = id.
        //Valor = localitzador.
        $id[$res->loc] = $res->loc;
    }
?>
</select>
</td>
</table>
</form>
<?php
//Fi del formulari de selecció de blogs.

//Cas en el que ja s'hagi seleccionat un blog.
if(isset($com)){
    //Creem un nou site per iniciar el cas d'ús.
    $site = new SiteQ();
    //Selecció de la comissió
    $site->Selecciona_quickkr($com);
    /** Taula amb tots els articles que conté el blog recuperat.
        El mostrem com un formulari per poder seleccionar els que
        volguem afegir a WordPress.
    */
?>
<h3><u>Transferencia de posts de Quickkr</u></h3>
Selecciona els articles que vulguis que s'agefeixin al blog<p>
Els articles actuals a Quickkr son els seg&uuml;ents<br>
<form name="inspost" method="POST">
<table border="0">
    <tr><td><b>T&iacute;tol</b></td>
        <td><b>Autor</b></td>
        <td><b>Data</b></td>
        <td>&nbsp;</td></tr>
    <?php
        //Comptador dels articles existents.
        $i = 0;
        //Recuperem la comissió seleccionada.
        $q = $site->getQ();
        //Recuperem el repositori d'articles de la comissió.
        $postsq = $q->getPostsq();
        foreach ($postsq as $p){
            echo "<tr><td><a href='". $p->getLink(). "'
target=_blank>". $p->getTitol(). "</a>&nbsp;</td>";
            echo "<td>". $p->getAutor(). "&nbsp;</td>";
            echo "<td>". $p->getData(). "&nbsp;</td>";
            echo "<td align='right'><input type='checkbox'
name='sel_". $i. "' value='0'></td></tr>";
            $i++;
        }

        //Camp ocult amb el número d'articles que hi ha.
        echo "<input type='hidden' name='contapost' value='". $i. "'>";
        echo "</table>";

    /** Selecció de la comissió de destí.
        Mostrem el llistat de comissions disponibles.
    */
    //Identificadors de les comissions
    switch_to_blog(1);

```

```

$sql_ids = "SELECT blog_id FROM ".$wpdb->prefix."blogs;";
restore_current_blog();
$idblogs = $wpdb->get_results($sql_ids);
$si = current_user_can('moderate_comments');

?>
<table border="0">
  <tr><td><h4>Selecciona el destí del post</h4></td></tr>
  <tr><td colspan="2">
    <select name="blog_desti">
<?php
    foreach ($idblogs as $blog){
        //Recuperació del nom de la comissió per cada una.
        $b = new BlogQ($blog->blog_id);
        $nomb = $b->getNom();
        global $blog_id;
        //En el cas que es tracti d'un usuari editor només mostrarà
        //el blog actual al que es troba
        if(!is_super_admin()){
            //Al blog principal (id = 1) no es poden afegir nous
            //articles per aquí. Per tant el saltem.
            if($blog->blog_id != 1 and ($blog->blog_id == $blog_id))
                echo "<option value=".$blog->
                blog_id.">".$nomb."&nbsp;</option>";
            }else{
                if($blog->blog_id != 1)
                    echo "<option value=".$blog->
                    blog_id.">".$nomb."&nbsp;</option>";
            }
        }
    }
    echo "<input type='hidden' name='com' value=".$com.">";
?>
</select>
</td>

<?php
//Fi selecció del blog destí.
//Camp ocult per identificar que l'acció seleccionada és la de
//afegir l'article a la comissió.
echo "<input type='hidden' name='accio' value='1'>";
echo "<td colspan='4' align='right'>
    <input type='submit' class='button-primary' value='Inserta'>
    </td></tr>";

?>
</table>
</form>
<?php
//Acció d'importació de Quickr a WordPress desde WordPress.
if(isset($_POST['accio'])){
    //Nombre d'articles disponibles.
    $i = $_POST['contapost'];
    //Comissió seleccionada com destí dels articles.
    $bl = $_POST['blog_desti'];
    //Recorregut per tots els articles per cercar els que han sigut
    //seleccionats.
    for($r = 0; $r < $i; $r++){
        //Cas en el que es trobi amb un article seleccionat.
        if(isset($_POST['sel_'. $r])) $site->Sel_article_quickr($r);
    }
    //Per últim cridem als dos últims esdeveniments. Seleccionem el
    //blog destí i finalitzem l'importació.
}

```



```

        $site->Recull_blog($bl);
        $site->Fi_importacio();
    }
    //Acció d'exportació desde Quickr
    else if(isset($_GET['quick'])){
        $in = $_GET['art'];
        $b = $_GET['b'];
        $site->Sel_article_quickr($in);
        $site->Recull_blog($b);
        $site->Fi_importacio();
    }

    } //Fi del cas de selecció de blog.

} //Fi del mètode d'opcions.

}
//Carreguem una nova instància del plugin.
$quick = new xuclablog();
include('classSite.php');
?>

```

12.3.2 Classe Site (classSite.php)

Classe controlador de l'aplicació. Es reaprofitja l'anterior classe *Site* creada a plugins anteriors.

S'han omès els mètodes getters i setters i els mètodes corresponents a altres plugins que no es fan servir en aquest. A l'arxiu original es troba tot el codi.

```
<?php
/**
 * Classe controlador.
 */
class SiteQ{

    //Atribut privat que correspon a un objecte Blog.
    private $b;
    //Atribut privat que correspon a un objecte Post.
    private $p;
    //Atribut privat que correspon a un objecte Quickr.
    private $q;
    //Atribut privat que correspon a un objecte PostQuickr.
    private $pq;

    /**
     * Mètode constructor, crea un nou controlador.
     */
    public function __construct(){
    }

    /**
     * Mètode destructor, destrueix el controlador i torna
     * el blog al seu estat inicial.
     */
    public function __destruct(){
        restore_current_blog();
    }

    /**
     * Mètode de recollida d'un nou blog a partir del identificador.
     * @id: localitzador del blog a escollir.
     */
    public function Recull_blog($id){
        $this->b = new BlogQ($id);
        switch_to_blog($this->b->getId());
    }

    /**
     * Mètode que selecciona una comissió de Quickr.
     * @idCom: identificador de la comissió.
     */
    public function Selecciona_comquickr ($idCom){
        $this->q = new Quickr($idCom);
    }

    /**
     * Mètode que selecciona un article de Quickr.
     * @idArt: identificador de l'article a seleccionar.
     */
    public function Sel_article_quickr ($idArt){
        $this->q->Sel_article_quickr($idArt);
    }
}
```

```
}

/**
 * Mètode que finalitza la importació d'articles.
 */
public function Fi_importacio (){
    $this->q->Fi_importacio($this->b);
    restore_current_blog();
}

}
include('classBlog.php');
include('classPostGen.php');
include('classPost.php');
include('classPostQ.php');
include('classQuickr.php');
include('classUsuari.php');
//Recuperació de la classe snoopy per fer la recuperació del contingut de webs.
require('class-snoopy.php');

?>
```

12.3.3 Classe Post (classPostGen.php)

Aquesta classe fa referència a una classe genèrica que reproduïx un article. Està formada únicament per els mètodes getters i setters, donat que no té mètodes propis comuns.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Post. Conté les dades de l'article.
 * Dades de l'article:
 *   @id: identificador de l'article a transformar.
 *   @Data: data en la que es va crear l'article.
 *   @Titol: títol de l'article.
 *   @Autor: persona que ha escrit l'article.
 *   @Contingut: cos de l'article.
 */

class PostQ{

    private $id;
    private $Data;
    private $Titol;
    private $Autor;
    private $Contingut;

    /**
     * Mètode constructor, crea un nou Post.
     * @id: id de l'article a crear.
     */
    public function __construct($id){

        $this->id = $id;
    }

    /**
     * Mètode que retorna l'identificador de l'article.
     * @Retorna: identificador del post.
     */
    public function getId(){
        return $this->id;
    }

    /**
     * Mètode que retorna la data de creació de l'article.
     * @Retorna: data de creació de l'article.
     */
    public function getData(){
        return $this->Data;
    }

    /**
     * Mètode que canvia la data de l'article.
     * @dat: data de l'article.
     */
    public function setData($dat){
        $this->Data = $dat;
    }

    /**
     * Mètode que retorna el títol de l'article.
     * @Retorna: títol de l'article.
     */
}
```

```

public function getTitol(){
    return $this->Titol;
}

/**
 *Mètode que canvia el títol de l'article.
 *@t: títol del post.
 */
public function setTitol($t){
    $this->Titol = $t;
}

/**
 *Mètode que retorna l'autor de l'article.
 *@Retorna: autor del post.
 */
public function getAutor(){
    return $this->Autor;
}

/**
 *Mètode que modifica l'autor de l'article.
 *@au: autor del post.
 */
public function setAutor($au){
    $this->Autor = $au;
}

/**
 *Mètode que retorna el cos de l'article.
 *@Retorna: contingut del post.
 */
public function getContingut(){
    return $this->Contingut;
}

/**
 *Mètode que modifica el cos de l'article.
 *@c: contingut del post.
 */
public function setContingut($c){
    $this->Contingut = $c;
}
}
?>

```

12.3.4 Classe PostQ (classPostQ.php)

Classe que representa un article de *Quickr*. Extén de la classe *Post*, per tant pot fer ús dels seus mètodes i té els seus atributs.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Post Quickr. Conté les dades de
 * l'article.
 * Dades de l'article:
 *   @Quickr: comissió a la que pertany l'article.
 *   @link: enllaç a l'article original.
 *   @datawp: data de l'article en format WordPress.
 */

class PostQQ extends PostQ{

    private $Quickr;
    private $link;
    private $dataWP;

    /**
     * Mètode constructor, crea un nou Post.
     * @id: id de l'article a crear.
     */
    public function __construct($id){
        $this->id = $id;
    }

    /**
     * Mètode que selecciona l'identificador de l'article.
     * @i: id de l'article.
     */
    public function setId ($i){
        $this->id = $i;
    }

    /**
     * Mètode que retorna l'identificador de l'article
     * @Retorna: id de l'article.
     */
    public function getId(){
        return $this->id;
    }

    /**
     * Mètode que selecciona l'enllaç de l'article
     * @lk: enllaç de l'article.
     */
    public function setLink($lk){
        $this->link = $lk;
    }

    /**
     * Mètode que retorna l'enllaç de l'article.
     * @Retorna: enllaç de l'article.
     */
    public function getLink(){
        return $this->link;
    }
}
```

```
/**
    Mètode que modifica la data de l'article.
    @data: data de l'article en format WordPress.
*/
public function setDatawp($data){
    $this->dataWP = $data;
}

/**
    Mètode que retorna la data de l'article.
    @Retorna: data de l'article en format WordPress.
*/
public function getDatawp(){
    return $this->dataWP;
}

}
?>
```

12.3.5 Classe PostWordPress (classPost.php)

Classe es correspon a un article de *WordPress*. Aquesta classe és la que habíem fet servir fins ara. Ara canvia i es transforma en una classe heretada de *Post*.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Post. Conté les dades de l'article.
 * Dades de l'article:
 *   @Rellevancia: importancia que té l'article dintre de la pàgina
 *   principal.
 *   @Blog: comissió a la que pertany l'article.
 */

class PostWordpressQ extends PostQ{

    private $Rellevancia;
    private $Blog;

    /**
     * Mètode constructor, crea un nou Post.
     * @id: id de l'article a crear.
     */
    public function __construct($id){

        $this->id = $id;
        $this->Data = Post::data();
        $this->Titol = Post::tit();
        $this->Autor = Post::aut();
        $this->Contingut = Post::cont();
        global $blog_id;
        $this->Blog = $blog_id;
    }

    /**
     * Mètode que introdueix la rellevància al post.
     * @rel: rellevància a introduir.
     */
    public function Selecciona_rellevancia ($rel){
        $this->Rellevancia = $rel;
    }

    /**
     * Mètode que retorna la rellevància de l'article.
     * @Retorna: rellevància de l'article.
     */
    public function getRellevancia(){
        return $this->Rellevancia;
    }

    /**
     * Mètode que retorna el blog al que pertany l'article.
     * @Retorna: blog al que correspon 'article.
     */
    public function getBlog(){
        return $this->Blog;
    }

    /**
```



```

    Mètode que modifica el blog al que pertany l'article.
    @b: blog a seleccionar per l'article.
*/
public function setBlog($b){
    $this->Blog = $b;
}

/**
    Mètode que cerca a base de dades la data de publicació
    de l'article.
    @Retorna: data de l'article.
*/
private function data (){
    global $wpdb;
    $sql_data = "SELECT post_date
                FROM ".$wpdb->prefix."posts
                WHERE id = '".$this->id."'";
    return $wpdb->get_var($sql_data);
}

/**
    Mètode que cerca el títol de l'article a base de dades.
    @Retorna: títol de l'article.
*/
private function tit (){
    global $wpdb;
    $sql_nom = "SELECT post_title
              FROM ".$wpdb->prefix."posts
              WHERE id = '".$this->id."'";
    return $wpdb->get_var($sql_nom);
}

/**
    Mètode que cerca l'autor a base de dades i el converteix
    en un objecte de tipus Usuari.
    @Retorna: usuari de l'article.
*/
private function aut (){
    global $wpdb;
    $sql_aut = "SELECT post_author
              FROM ".$wpdb->prefix."posts
              WHERE id = '".$this->id."'";
    $idAut = $wpdb->get_var($sql_aut);
    return new Usuari($idAut);
}

/**
    Mètode que cerca el contingut a base de dades.
    @Retorna: contingut de l'article.
*/
private function cont(){
    global $wpdb;
    $sql_cont = "SELECT post_content
                FROM ".$wpdb->prefix."posts
                WHERE id = '".$this->id."'";
    return $wpdb->get_var($sql_cont);
}
}
?>

```

12.3.6 Classe Blog (classBlog.php)

Classe que representa un blog de *WordPress*. S'han omés els mètodes que corresponen a altres plugins i els mètodes getters i setters, a l'arxiu original es troben tots els mètodes implementats.

```
<?php
/**
 * Classe que crea un nou objecte de tipus Blog. Conté les dades del blog.
 * Dades del blog:
 *   @id: identificador del blog a transformar.
 *   @path: ruta URL d'accés al blog.
 *   @nom: títol del blog.
 *   @pare: blog del que depèn.
 *   @fills: array amb els blogs que depenen d'ell.
 *   @vist: flag que indica si el blog ha sigut pintat o
 *   no al menú (0 = no pintat).
 */

class BlogQ{
    private $id;
    private $nom;
    private $path;
    private $pare;
    private $ordre;
    private $fills;
    private $vist;

    private $post_relevants = array ();

    /**
     * Mètode constructor, crea un nou Blog.
     * @id: id del blog a crear.
     */
    public function __construct($id){

        $this->id = $id;
        $this->nom = BlogQ::name();
        $this->path = BlogQ::path();
        $this->pare = BlogQ::pare();
        $this->ordre = BlogQ::ordre();
        $this->vist = 0;
    }

    /**
     * Mètode privat que cerca el nom del blog i el retorna.
     * En cas de que es tracti del blog principal el titula "Home".
     * @Retorna: nom del blog.
     */
    private function name(){
        global $wpdb;
        $id = $this->id;
        if($id == 1) return "Home";
        switch_to_blog($id);
        $optname = "'blogname'";
        $sql = "SELECT option_value
                FROM ".$wpdb->prefix."options
                WHERE option_name = ";
        $sql_name = $sql.$optname;
        $nom = $wpdb->get_var($sql_name);
        restore_current_blog();
        return $nom;
    }
}
```

```

}

/**
 *Mètode privat que cerca i retorna el pare del que depèn el blog.
 *@Retorna: pare actual del blog.
 */
private function pare(){
    global $wpdb;
    $id = $this->id;
    switch_to_blog($id);
    $optname = "'pare'";
    $sql = "SELECT option_value
          FROM ".$wpdb->prefix."options
          WHERE option_name = ";
    $sql_pare = $sql.$optname;
    $pare = $wpdb->get_var($sql_pare);
    restore_current_blog();
    return $pare;
}

/**
 * Cerca i retorna l'ordre que ocupa al menú
 * @Retorna: ordre del blog.
 */
private function ordre(){
    global $wpdb;
    $or = get_blog_option($this->id, 'ordre');
    return $or;
}

/**
 * Cerca i retorna la URL del blog.
 * @Retorna: URL del blog.
 */
private function path(){
    global $wpdb;
    $id = $this->id;
    $sql_path = "SELECT path
                FROM ".$wpdb->prefix."blogs
                WHERE blog_id = ".$id.";";
    $path = $wpdb->get_var($sql_path);
    return $path;
}

/**
 * Mètode que insereix un array d'articles.
 * @artImp: articles a importar.
 */
public function Insereix ($artImp){
    foreach ($artImp as $article){
        //Insercció al blog
        /** Nou article
         * - post_status: estat del article. En el nostre cas, publicat.
         * - post_title: títol de l'article. El que habíem recollit
         * anteriorment.
         * - post_date: data en format GMT.
         * - post_content: contingut de l'article.
        */
    }
}

```

```

        - post_author: autor de l'article.
    */
    $post = array(
        'post_status' => 'publish',
        'post_title' => $article->getTitol(),
        'post_date' => $article->getDatawp(),
        'post_content' => $article->getContingut(),
        'post_date_gmt' => $article->getDatawp(),
        'post_author' => $article->getAutor()->getId();
    //Consulta de selecció per veure si ja existeix l'article dintre de
la comissió.
    global $wpdb;
    $sql = "SELECT id
        FROM ".$wpdb->prefix."posts
        WHERE post_title = '".$article->getTitol()."' and
post_date = '".$article->getData()."'
        and post_type='post' and post_status = 'publish'";
    $res = $wpdb->get_var($sql);
    //Cas en que no existeixi l'article.
    if($res == null or $res == 0){
        //Inserta l'article a la comissió.
        wp_insert_post($post, $err);
    }
    //Cas en que ja existeixi.
    else{
        //Actualitzem l'article per si ha hagut algun canvi al contingut.
        $post['ID'] = $res;
        wp_update_post($post);
    }
}
}
}
}
}
?>

```

12.3.7 Classe Quickr (classQuickr.php)

Classe que representa un blog de *Quickr*. Aquesta classe fa ús de les classes *cURL* i *Snoopy*. S'han omés els mètodes getter i setter.

```
<?php
/**
    Classe que crea un nou objecte de tipus Quickr. Conté les dades de la
    comissió Quickr.
    Dades de la comissió:
        @id: identificador del comissió a transformar.
        @nom: títol de la comissió.
        @url: direcció del feed Quickr.
        @blog: blog amb el que es correspon de WordPress.
*/

class Quickr{
    private $id;
    private $nom;
    private $url;
    private $blog;
    private $articles_importar = array ();
    private $usr; //Usuari i pass de quickr
    private $pass;
    private $xml; //Feed amb totes les entrades
    private $postsq = array ();

    /**
        Mètode constructor, crea un nou Quickr.
        @id: id del blog a crear.
    */
    public function __construct($id){

        $this->id = $id;
        $this->nom = Quickr::name();
        $this->url = Quickr::url($id);
        $this->usr = "wpsadmin";
        $this->pass = "MANIPULACION78";
        $this->xml = Quickr::Feed();
        $_SESSION['id'] = $this->usr;
        $_SESSION['pass'] = $this->pass;
        Quickr::EntradesFeed();
    }

    /**
        Mètode que selecciona un article Quickr
        @idart: identificador de l'article.
    */
    public function Sel_article_quickr ($idart){
        //Cerca l'article dintre del repositori d'articles de quickr.
        foreach($this->postsq as $p){
            if($p->getId() == $idart){
                $a = $p;
                break;
            }
        }
        //Quan troba l'article el modificap er adaptar-lo a WordPress.
        Quickr::ModPost($a);
        //Finalment l'afegeix al repositori d'articles per importar.
        array_push($this->articles_importar, $a);
    }
}
```

```

/**
    Mètode que finalitza el procés d'importació.
    @b: objecte blog destí.
*/
public function Fi_importacio ($b){
    $this->blog = $b;
    //Demana al blog corresponent que afegeixi els articles a importar.
    $this->blog->Insereix($this->articles_importar);
}

/**
    Mètode que retorna la URL del feed del blog quickr.
    @loc: identificador de la comissió a Quickr.
    @Retorna: URL on es troben els articles.
*/
private function url($loc){
    $path = "http://espais.eic.cat/webcontent/rest/library/" . $loc . "/feed";
    return $path;
}

/**
    Mètode que retorna el feed de la comissió.
    @Retorna: variable xml amb el resultat del feed
*/
private function Feed (){
    /** Creació d'una nova instància a snoopy.
        $snoopy: Objecte de la classe snoopy amb el que treballarem.
        $snoopy->user: Usuari de Quickr.
        $snoopy->pass: Contrasenya del usuari de Quickr.
        $snoopy->fetch($url): Recuperació de tot el contingut de la direcció
        que li indiquem.
    */
    $snoopy = new Snoopy();
    $snoopy->user = $_SESSION['id'];
    $snoopy->pass = $_SESSION['pass'];
    $snoopy->fetch($this->url);
    //Com el contingut recuperat està en format xml, creem un objecte
    //SimpleXMLElement
    //que ens permetrà tractar-lo.
    //La variable xml guardarà tot el contingut de la pàgina formatejat a
    //xml.
    //$snoopy->results: Mostra el resultat de la direcció que hem recollit.
    return new SimpleXMLElement($snoopy->results);
}

/**
    Mètode que tracta l'arxiu XML amb les entrades.
*/
private function EntradesFeed (){
    //Comptador dels articles existents.
    $i = 0;
    //Identifiquem els articles al xml per l'etiqueta <entry>.
    //Cada article en el guardem un valor de l'array entrada.
    foreach ($this->xml->entry as $entrada[$i]){

        //Recollida dels atributs de l'etiqueta <content>. Aquí es trobaran
        //els enllaços.
        $link = $entrada[$i]->content->attributes();
        //L'atribut src conté la ruta de l'article, per tant ho guardem en
        //un array de links (lk).
    }
}

```

```

$linkart = $link['src'];
//Mostrem el títol de l'article i l'opció de veure'l accedint a ell
//directament.
$titolart = $entrada[$i]->title;
//Recollida del nom de l'autor. Aquest nom l'identifiquem per el seu
//login de Quickr, que es correspondrà amb el de WordPress.
$autorart = $entrada[$i]->author->uri;
//Retallem el nom perquè ens quedi només nom.cognom només.
$autorart = substr($autorart, 6);
$mida = strlen($autorart);
$autorart = substr($autorart, 0, $mida-35);

//Data a la que es va publicar l'article en format dd/mm/aaaa.
//Es recupera per l'etiqueta <published> i el tractem al mètode
//dataesp.
$dataart = Quickr::Dataesp(substr($entrada[$i]->published, 0, 19),
'EUR');

$contart = Quickr::CapturaHTML($linkart);
//Creem un nou objecte Post Quickr i introduïm tota la informació de
//l'article.
$post = new PostQQ($i);
$post->setData($dataart);
$post->setDatawp(Quickr::Dataesp(substr($entrada[$i]->published, 0,
19), 'WP'));
$post->setTitol($titolart);
$post->setAutor($autorart);
$post->setLink($linkart);
$post->setContingut($contart);
//Finalment afegim l'article al repositori d'articles de Quickr.
array_push($this->postsq, $post);

    $i++;
}
}

/**
Canvia el format de la data a format europeu o WP.
@data: Data que es vol formatjar.
@mode: Mode de format.
    EUR -> estil europeu (dd-mm-aaaa).
    WP -> estil base de dades de WordPress (dd-mm-aaaa hh:mm:ss).
@Retorna: data en el format especificat.
*/
private function Dataesp ($data, $mode){
    $dia = substr($data, 8, 2);
    $mes = substr($data, 5, 2);
    $any = substr($data, 0, 4);
    $hora = substr($data, 11, 8);
    if($mode == 'EUR') $nova = $dia."-".$mes."-".$any;
    if($mode == 'WP') $nova = $any."-".$mes."-".$dia." ".$hora;
    return $nova;
}

/**
Captura el codi HTML d'una entrada.
@url: direcció web que es vol capturar.
@Retorna: el contingut HTML de la pàgina sol·licitada.
*/
private function CapturaHTML ($url){

```

```

//Generem un nou objecte de la classe snoopy per capturar el codi HTML.
$snoopy = new Snoopy();
//Usuari i contrasenya de les variables sessió que tenim iniciades.
$snoopy->user = $_SESSION['id'];
$snoopy->pass = $_SESSION['pass'];
$snoopy->fetch($url);
//Retornem el resultat.
return ($snoopy->results);
}

/**
Mètode que modifica un article per recuperar les imatges.
@post: objecte post a modificar (passat per referència).
*/
public function ModPost(&$post){
    global $wpdb;
    //Cerca les imatges contingudes a l'article.
    $img = Quickr::searchimg($post->getLink());
    //Cas en el que trobi imatges.
    if($img != ""){
        foreach ($img as $imagen){
            srand(Quickr::make_seed());
            //Generem un valor aleatori per el nom de la imatge.
            $randval = rand();
            //Recuperem la URL absoluta de l'imatge.
            $url = $imagen['abs'];
            //Nom de l'imatge a guardar.
            $file = $randval.'_img.jpg';
            //Guardem l'imatge al servidor mitjançant el mètode
            //Salvaimatge.
            $imguardada = Quickr::Salvaimatge($url, $file);
            //Canviem la ruta de la imatge antiga per la nova dintre del
            //contingut de l'article.
            $cont = Quickr::Replaceimg($imagen['rel']['rutaentera'],
            $imagen['rel']['numi'], $imguardada, $post->getContingut());
            $post->setContingut($cont);
        }
        //Formatejem el títol a HTML mitjançant el mètode formataccent.
        $titol = $post->getTitoll();
        Quickr::Formataccent ($titol);
        $post->setTitoll($titol);
        //Formatejem el contingut a HTML mitjançant el mètode formataccent.
        $cont = $post->getContingut();
        Quickr::Formataccent ($cont);
        $post->setContingut($cont);
        //Recuperem el id de l'autor de la base de dades de WordPress
        //indicant l'autor de l'article.
        $idauth = Quickr::getAuthorWP($post->getAutor());
        $usu = new UsuariQ($idauth);
        $post->setAutor($usu); //Canviem l'autor per un objecte de tipus
        //usuari.
    }
}

/**
Mètode que cerca les etiquetes IMG que es troben a una URL.
@url: pàgina a tractar.
@Retorna: array amb les URL de les imatges.
*/

```



```

*/
private function Searching ($url){
    //Recollida del contingut de l'article mitjançant la classe snoopy.
    $snoopy = new Snoopy();
    $snoopy->user = $_SESSION['id'];
    $snoopy->pass = $_SESSION['pass'];
    $snoopy->fetch($url);
    //Longitud total de l'article.
    $long = strlen($snoopy->results);
    $cadena = $snoopy->results;
    unset($comienzo);
    //Comptador d'imatges. Per el cas que hi hagi més d'una imatge a
    //l'article.
    $numimg = 0;
    //Recorregut per tot l'article caràcter a caràcter.
    for($snt = 0; $snt < $long; $snt++){
        //Cas en el que trobem l'etiqueta img.
        if($cadena[$snt] == 'i' && $cadena[$snt+1] == 'm' && $cadena[$snt+2]
== 'g'){
            //Marquem l'inici de la ruta.
            $comienzo = $cadena[$snt+9];
            unset($final);
            $snt = $snt+10;
            //Posició on comença la ruta.
            $numi = $snt;
        }
        //Cas en el que trobem " que ens indiquin el final de la ruta
        if($cadena[$snt] == '"' && isset($comienzo) && !isset($final)){
            //Últim caràcter de la ruta.
            $final = $cadena[$snt-1];
            //Posició final de la ruta.
            $numfi = $snt;
            //Longitud de la ruta.
            $longi = $numfi - $numi;
            //Tallem la cadena per obtenir la ruta.
            /** L'array ruta conté les següents dimensions:
                - Numimg: número d'imatge corresponent.
                - Rel: fa referència a la ruta relativa de l'imatge.
                - Abs: fa referència a la ruta absoluta de l'imatge.
                    Ruta correcta que ens interessa.
                - Ruta: cadena indicant posició inici i final.
                - Rutaentera: cadena sencera incloent els caràcters
                    d'inici i final correctes.
                - Numi: posició d'inici de la ruta, per saber on
                    trobar-la a l'hora de reemplaçar.
            */
            $ruta[$numimg]['rel']['ruta'] = substr($cadena, $numi, $longi);
            $ruta[$numimg]['rel']['rutaentera'] = substr($cadena, $numi-1,
$longi+1);
            $ruta[$numimg]['rel']['numi'] = $numi;
            $ruta[$numimg]['abs'] =
"http://espais.eic.cat/lotus/wcm/my".$ruta[$numimg]['rel']['ruta'];
            //Augmentem el comptador per el cas en que hi hagi més imatges.
            $numimg++;
        }
    }
    //Retornem l'array sencer amb totes les imatges i totes les seves rutes.
    return $ruta;
}
}

```

```

/**
    Funció que genera una semilla per el rand
*/
private function make_seed(){
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}

/**
    Baixa la imatge que conté la url, i la puja al servidor.
    Inicia sessió a quickr i accedeix a la ruta de l'imatge per
    descarregar-la
    @url: imatge origen.
    @path: nom de la imatge al servidor.
    @Retorna: ruta URL de l'arxiu.
*/

private function Salvaimatge ($urlimg, $path){
    //Creem un nou objecte de la classe snoopy.
    $snoopy = new Snoopy();
    $sem = Quickr::make_seed();
    srand($sem);
    $randval = rand();
    //Nom de la cookie on guardarem les dades de la sessió.
    $nomcookie = $randval."_cookie.txt";
    //Usuari i contrasenya de quickr.
    $login_email = $_SESSION['id'];
    $login_pass = $_SESSION['pass'];
    //Ruta de la cookie.
    // getcwd() recupera la ruta del servidor on s'està treballant.
    $cookies = getcwd() . "/" . $nomcookie;
    //Direcció de login de quickr.
    $url="http://espais.eic.cat/";
    //Direcció de la web a la que s'accedeix per fer el login al omplir el
    //formulari.
    $urlaction =
"lotus/quickr/!ut/p/cl/04_SB8K8xLLM9MSSzPy8xBz9CP0os_hQJ29XlzBnI0N3YyNzAyMv51Cng
MBAIxNXE_lwKA6zeD9_olA3E09DQwszV0MDIzMPeyefME8DdxdjLwBDuBooO_nkZ-bq1-
QnZ3m6KioCADKJMie/dl2/d1/L0lDUWtpQ1NZSkNncFJBISEvb0VvZ0FFQ1FRREdJUXBTR0djRndUT0E
hL1lBeEpKNDUwLTVGNGTzdXl0dyEhLzdfVUJLRURWQzIxRzMyNzAySkNVQlBRUTI0VTQvS19fX18yL3d
wcy5wb3J0bGV0cy5sb2dpcg!!/";
    //Camps que s'envien amb el formulari de login de Quickr.
    $postdata = "wps.portlets.userid=". $login_email."&password=" .
$login_pass."&PC_7_UBKEDVC21G32702JCUBPQQ24U4__login=Iniciar%20Sesi3n";
    $z = 0;
    //Bucle que es repeteix mentres que no es guardi l'imatge.
    while($z == 0){
        //Iniciem sessió de curl. Ens permetrà fer el login a Quickr.
        $curl = curl_init();
        /** curl_setopt: opcions que donem a la sessió curl iniciada.
        - CURLOPT_URL: indiquem la ruta on va a parar el login.
        - CURLOPT_POSTFIELDS: camps que es passen al formulari.
        - CURLOPT_POST: mètode del formulario en format post.
        - CURLOPT_FOLLOWLOCATION: anar a la ruta de després de fer login.
        - CURLOPT_SSL_VERIFYPEER: verificació amb seguretat SSL del
            login.
        - CURLOPT_COOKIEJAR: guarda la sessió en una cookie.
        - CURLOPT_COOKIEFILE: guarda la cookie en un arxiu.
        - CURLOPT_RETURNTRANSFER: retorna el resultat a un string.
        - CURLOPT_USERAGENT: indica el navegador web predeterminat.

```

```

*/
curl_setopt ($curl, CURLOPT_URL, $url . $urlaction);
curl_setopt ($curl, CURLOPT_POSTFIELDS, $postdata);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($curl, CURLOPT_COOKIEJAR, $cookies);
curl_setopt($curl, CURLOPT_COOKIEFILE, $cookies);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_USERAGENT, "Mozilla/5.0 (Windows; U;
Windows NT 6.0; en-US; rv:1.9.0.12) Gecko/2009070611 Firefox/3.0.12");
//Recollim a snoop el resultat de l'autenticació a Quickr.
$snoopy->fetch(curl_exec($curl));
curl_close($curl);

//Fem una nova connexió curl, amb la sessió autenticada que es troba
//a les cookies creades.
//Aquesta nova connexió recollirà l'imatge indicada per la ruta
//urlimg.
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, $urlimg);
curl_setopt($curl, CURLOPT_COOKIEJAR, $cookies);
curl_setopt($curl, CURLOPT_COOKIEFILE, $cookies);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_USERAGENT, "Mozilla/5.0 (Windows; U;
Windows NT 6.0; en-US; rv:1.9.0.12) Gecko/2009070611 Firefox/3.0.12");
curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
$s = curl_exec($curl);
//Directori de pujades d'arxius de WordPress.
$uploads = wp_upload_dir();
//Ruta final on es guardarà l'imatge.
$pathf = $uploads['path']. "/" . $path;

//Generem l'arxiu nou en mode escriptura i amb el nom de la imatge.
$f = fopen($pathf, 'wb');
//Escribim al nou arxiu el que hem recollit anteriorment, la imatge.
$z = fwrite($f, $s);
}
$pathurl = $uploads['url']. "/" . $path;
return $pathurl;
}

/**
Canvia la ruta de la imatge actual per la que hem pujat al servidor
@imgor: imatge d'origen
@locor: localització de l'imatge d'origen (posició en el string cont)
@imgdest: imatge destí
@cont: contingut
@Retorna: cadena amb la imatge modificada.
*/
private function Replaceimg ($imgor, $locor, $imgdest, $cont){
//Mida del contingut.
$long = strlen($cont);
//Mida de l'imatge d'origen.
$longimgor = strlen($imgor);
//Posició de l'imatge d'origen.
/** strpos retorna l'inici de la cadena cadena que se li indiqui, dintre
del contingu que es vulgui.
@cont: contingut.
@imgor: cadena a cercar.
*/
*/

```

```

$posori = strpos($cont, $imgor);

//Comparació de l'imatge per comprovar que és l'imatge correcte.
/** substr_compare compara dues cadenes per veure si són iguals.
    @cont: contingut.
    @imgor: cadena origen de comparació.
    @posori: posició de la cadena destí de la comparació.
    @longimor: mida de la cadena destí.
    Compararà la cadena continguda desde @posori fins @longimor.
    Retorna 0 si són iguals.
*/
$si = substr_compare($cont, $imgor, $posori, $longimor);
$st2 = $cont;
//Cas en el que siguin iguals les cadenes.
if($si == 0){
    //Canviem la cadena de l'imatge d'origen per la de la imatge destí.
    /** str_replace reemplaça una cadena origen per una altre destí.
        @imgor: cadena origen.
        @imgdest: cadena destí.
        @cont: contingut en el que es trova la cadena.
    */
    $st2 = str_replace($imgor, $imgdest, $cont);
}
return $st2;
}

/**
Formateja la cadena de caràcters d'entrada cercant els accents i els
transforma a format HTML.
@cadena: conté el string que es vol formatejar.
El paràmetre cadena es passa per referència, amb lo qual queda modificat
i no s'ha de retornar.
*/
private function Formataaccent (&$cadena){
    $nueva_cadena = $cadena;
    $nueva_cadena = str_replace('á', '&acute;', $nueva_cadena);
    $nueva_cadena = str_replace('Á', '&Aacute;', $nueva_cadena);
    $nueva_cadena = str_replace('à', '&agrave;', $nueva_cadena);
    $nueva_cadena = str_replace('À', '&Agrave;', $nueva_cadena);
    $nueva_cadena = str_replace('é', '&eacute;', $nueva_cadena);
    $nueva_cadena = str_replace('É', '&Eacute;', $nueva_cadena);
    $nueva_cadena = str_replace('è', '&egrave;', $nueva_cadena);
    $nueva_cadena = str_replace('È', '&Egrave;', $nueva_cadena);
    $nueva_cadena = str_replace('í', '&iacute;', $nueva_cadena);
    $nueva_cadena = str_replace('Í', '&Iacute;', $nueva_cadena);
    $nueva_cadena = str_replace('ì', '&igrave;', $nueva_cadena);
    $nueva_cadena = str_replace('Ì', '&Igrave;', $nueva_cadena);
    $nueva_cadena = str_replace('ó', '&oacute;', $nueva_cadena);
    $nueva_cadena = str_replace('Ó', '&Oacute;', $nueva_cadena);
    $nueva_cadena = str_replace('ò', '&ograve;', $nueva_cadena);
    $nueva_cadena = str_replace('Ò', '&Ograve;', $nueva_cadena);
    $nueva_cadena = str_replace('ú', '&uacute;', $nueva_cadena);
    $nueva_cadena = str_replace('Ú', '&Uacute;', $nueva_cadena);
    $nueva_cadena = str_replace('ù', '&ugrave;', $nueva_cadena);
    $nueva_cadena = str_replace('Û', '&Ugrave;', $nueva_cadena);
    $nueva_cadena = str_replace('ä', '&auml;', $nueva_cadena);
    $nueva_cadena = str_replace('Ä', '&Auml;', $nueva_cadena);
    $nueva_cadena = str_replace('ë', '&euml;', $nueva_cadena);
    $nueva_cadena = str_replace('Ë', '&Euml;', $nueva_cadena);
    $nueva_cadena = str_replace('ï', '&iuml;', $nueva_cadena);
}

```

```

    $nueva_cadena = str_replace('ï', '&Iuml;', $nueva_cadena);
    $nueva_cadena = str_replace('ö', '&ouml;', $nueva_cadena);
    $nueva_cadena = str_replace('Ö', '&Ouml;', $nueva_cadena);
    $nueva_cadena = str_replace('ü', '&uuml;', $nueva_cadena);
    $nueva_cadena = str_replace('Ü', '&Uuml;', $nueva_cadena);
    $cadena = $nueva_cadena;
}

/**
 * Captura l'autor de la base de dades de WordPress indicant l'autor d'un
 * article de Quickr.
 * @login: nom d'usuari de Quickr.
 * @Retorna: id d'usuari de WordPress.
 */

function getAuthorWP ($login){
    global $wpdb;
    $sql = "SELECT id
            FROM ".$wpdb->prefix."users
            WHERE user_login = '".$login."'";
    $user = $wpdb->get_var($sql);
    return $user;
}
}
?>

```

12.4 Quickr To WordPress. Versió Exportació d'articles des de Quickr.

12.4.1 Interfície gràfica del portlet (articles.jsp)

Aquest arxiu implementa la interfície gràfica del portlet de *Quickr*. S'han omés els imports.

```
<html>
<head>

<title>Articles per importar a WordPress</title>
<link href="http://www.eic.es/include/css/estils.css" rel="stylesheet"
type="text/css" />
</head>
<body bgcolor="#ffffff" text="#000000" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0" link="#003177" vlink="#003177" alink="#003177">
<br/>
<%
    //Recuperació de l'objecte Quickr amb la informació.
    Quickr quick = (Quickr)request.getAttribute("quickr");
    //Recuperació de la llista de blogs destí.
    String listab[] = (String [])request.getAttribute("listab");
    //Llista d'articles a mostrar.
    List articles = quick.getPosts();
    //Iterador per recórrer la llista d'articles.
    Iterator it = articles.iterator();
    //Recuperació de la url de sortida cap a WordPress.
    String url = (String)request.getAttribute("u");
    //Cas en que s'hagi omplert el formulari. Acció d'enviar cap a WordPress.
    if(url != null){
        //Nom d'usuari i contrasenya de WordPress.
        String l = "admin";
        String p = "administracion";
        //Codificació del nom i la contrasenya en base64.
        String logencode;
        logencode = new String(Base64.encodeBase64(l.getBytes()));
        String passencode;
        passencode = new String(Base64.encodeBase64(p.getBytes()));
        //Url amb la crida a la funció de WordPress per inserir
        //articles de forma externa. Inicia sessió.
        String url2 =
"http://localhost/wpcoeic/signinremot.php?in=true&u="+logencode+"&p="+passencode
+"&"+url;
        //Url amb la crida per finalitzar la sessió abans iniciada.
        String url3 = "http://localhost/wpcoeic/signinremot.php?in=false";
    %>
    <!-- Frame ocult que s'encarrega de fer les crides corresponents -->
    <iframe name="trastero2" style="width:0px;height:0px;visibility:hidden">
    </iframe>
    <script>window.frames['trastero2'].location.href = '<%=url2%>';</script>
    <%
    }
    %>
    <!-- Taula amb els articles del feed -->
    <table border = "0">
        <form action = "<portlet:renderURL/>" method="post">
            <tr><td colspan="4"><h3>Articles a exportar a WordPress</h3></td></tr>

            <tr><td><b>Títol</b></td><td><b>Autor</b></td><td><b>Data</b></td><td>
&nbsp;</td></tr>
        <%
            int i=0;
```

```

<!-- Recorregut per els articles -->
while(it.hasNext()){
    PostQuickr p = (PostQuickr)it.next();

    %>
    <tr>
        <td><a href="<%=p.getLink()%>"><%=p.getTitol() %></a></td>
        <td><%=p.getAutor() %></td>
        <td><%=p.getData() %></td>
        <td><input type="radio" name="sel" value="<%=i %>"></td>
    </tr>
    <%
    i++;
    }
    %>
    <input type="hidden" name="i" value="<%=i%>">
    <tr><td colspan="4" align="right">
        <select name="blog">
            <%
                //Llista de blogs destí.
                int l = listab.length;
                System.out.println(l);
                for(int cnt=0; cnt<l; cnt++){
                    if(listab[cnt] != null){
                        %>
                        <option value="<%=cnt %>"><%=listab[cnt] %></option>
                        <%
                    }
                }
            %>
        </select>
    </td></tr>
    <tr><td colspan="4" align="right"><input name="evento" type="submit"
value="Exportar"></td></tr>
    </form>
</table>
</body>
</html>

```

12.4.2 Controlador del portlet (QuickrToWordPressPortlet.java)

Classe controladora del portlet implementada en java. S'han omés els imports i altre informació irrellevant. El contingut sencer es troba al codi de la implementació.

```
package com.ibm.quickrtowordpress;
/**
 * Portlet QuickrToWordPress.
 */
public class QuickrToWordPressPortlet extends javax.portlet.GenericPortlet {
    public void init() throws PortletException{
        super.init();
    }
    public void doView(RenderRequest request, RenderResponse response) throws
PortletException, IOException {
        // Set the MIME type for the render response
        response.setContentType(request.getResponseContentType());
        PortletRequestDispatcher rd = getPortletContext().getRequestDispatcher(
"/WEB-INF/articles.jsp");
        //Recuperació dels paràmetres del portlet.
        PortletPreferences pref = request.getPreferences();
        //Recuperació de l'identificador de la comissió.
        String idquick = pref.getValue("IdComissio", "");
        Quickr quick;
        try {
            //Creem un nou objecte Quickr amb l'identificador de la comissió.
            quick = new Quickr(idquick);
            //Recuperació de la cadena de blogs.
            String blogs = pref.getValue("WordPressBlogs", "");
            //Recuperació de la llista de blogs a partir de la cadena.
            String listab [] = quick.recuperablogs(blogs);
            //Assignació dels paràmetres de retorn amb l'objecte Quickr i
            //la llista de blogs destí.
            request.setAttribute("quickr", quick);
            request.setAttribute("listab", listab);
        } catch (SAXException e) {
            e.printStackTrace();
        } catch (ParserConfigurationException e) {
            e.printStackTrace();
        }
        //Acció per quan s'executa el formulari.
        String actionEventName = request.getParameter("evento");
        if("Exportar".equalsIgnoreCase(actionEventName)){
            //Recuperació de l'article seleccionat.
            String sel = request.getParameter("i");
            sel = sel.trim();
            int s = Integer.parseInt(sel);
            String selec = request.getParameter("sel");
            if(selec != null){
                //Recuperació del blog destí.
                String blog = request.getParameter("blog");
                //Montatge de la url de sortida cap a WordPress.
                String u = "com="+idquick+"&art="+selec+"&b="+blog;
                //Retorn de la url de sortida.
                request.setAttribute("u", u);
            }
        }
        rd.include(request,response);
    }
}
```


12.4.3 Classe Quickr (Quickr.java)

Classe que conté tota la informació rellevant d'una comissió *Quickr*. S'han omés els imports i altre informació irrellevant, com mètodes *getters* i *setters*.

```
/**
    Classe Quickr.
    Representa un objecte d'una comissió Quickr.
    @id: identificador de la comissió.
    @Nom: nom de la comissió.
    @url: direcció del rss.
    @usr: usuari administrador Quickr.
    @pass: contrasenya admin Quickr.
*/
public class Quickr {
    private String id;
    private String Nom;
    private String url;
    private static String usr = "wpsadmin";
    private static String pass = "MANIPULACION78";
    private String xml;
    private Document xmlreal;
    public List posts = new ArrayList();

    /**
        Constructor.
        Crea un objecte Quickr.
    */
    public Quickr (String id) throws SAXException, ParserConfigurationException,
    IOException{
        //Assignem l'identificador a l'objecte.
        this.id = id;
        //Assignem la url del feed.
        setUrl(id);
        //Identificació per http Basic Authentication i recuperació del cos del
    feed.
        this.xml = conecta(this.url);
        //Creació de l'arxiu xml amb el contingut del feed.
        this.xmlreal = stringToDom(this.xml);
        posts.clear();
        //Recuperació de les entrades del feed.
        entradasfeed(this.xmlreal);
    }

    /**
        Recuperació dels blogs destí de la comissió Quickr.
        @cadena: string amb els blogs destí. Forma: idBlog, nomBlog;
        @return Array amb cadascun dels blogs.
    */
    public String [] recuperablogs(String cadena){
        //Indicador de l'inici del registre.
        int inici = 0;
        //Vector de blogs. Mida màxima: 50.
        String blogs [] = new String[50];
        //String identificador del blog.
        String ident = "";
        int id = 0;
        //Recorregut per la cadena d'entrada.
        for(int i=0; i<cadena.length(); i++){
            //Tractament caràcter a caràcter.
```

```

Character c = new Character(cadena.charAt(i));
//Caràcters de comparació "," i ";".
String coma = ",";
Character com = new Character(coma.charAt(0));
String puntocom = ";";
Character puntocomma = new Character(puntocom.charAt(0));
//Cas que trobi una ,. Troba un identificador del blog.
if(c.compareTo(com) == 0){
    ident = cadena.substring(inici, i);
    //Guardem l'identificador e indiquem que com inici el
    //següent caràcter.
    id = Integer.parseInt(ident);
    inici = i+1;
}
//Cas en que trobi un ; o finalitzi la cadena. Recupera el nom
//del blog.
else if((c.compareTo(puntocomma) == 0) ||
        (i == (cadena.length()-1))){
    //Recuperació del nom.
    String nom = cadena.substring(inici, i);
    //Guardem al vector de blogs. ID = posició del vector.
    blogs[id] = nom;
    inici = i+1;
}
}
//Retorna el vector de blogs.
return blogs;
}
/**
Mètode que tracta les entrades d'un document xml.
@Xml Document xml a tractar
*/
public void entradasfeed (Document xml){
//Recuperació de les entrades del xml. Les guardem com un node de
llista.
NodeList llista = xml.getElementsByTagName("entry");
//Recorregut per cadascuna de les entrades.
for(int i=0; i < llista.getLength(); i++){
//Creem un nou objecte post.
PostQuickr post = new PostQuickr();
//Creem un nou element per la llista.
Node node = llista.item(i);
//Guardem els atributs de l'entrada per tractarlos.
NodeList fillNode = node.getChildNodes();
//Recorregut per els atributs de l'entrada.
for(int a = 0; a < fillNode.getLength(); a++){
Node n = fillNode.item(a);
//Títol de l'article.
if(n.getNodeName() == "title"){
Node tit = n.getFirstChild();
post.setTitoll(tit.getNodeValue());
}
//Data de l'article.
if(n.getNodeName() == "published"){
Node up = n.getFirstChild();
String data = up.getNodeValue();
data = data.substring(0, 19);
data = Dataesp(data, "EUR");
post.setData(data);
}
}
}

```

```

    }
    //Autor de l'article.
    if(n.getNodeName() == "author"){
        NodeList auth = n.getChildNodes();
        for(int cont=0; cont<auth.getLength(); cont++){
            Node childauth = auth.item(cont);
            if(childauth.getNodeName() == "uri"){
                Node uri = childauth.getFirstChild();
                String nom = uri.getNodeValue();
                nom = nom.substring(6);
                int longnom = nom.length();
                nom = nom.substring(0, longnom-35);
                post.setAutor(nom);
            }
        }
    }
    //Contingut de l'article.
    if(n.getNodeName() == "content"){
        NamedNodeMap att = n.getAttributes();
        for(int c = 0; c<att.getLength(); c++){
            Node childcont = att.item(c);
            if(childcont.getNodeName() == "src"){
                post.setLink(childcont.getNodeValue());
            }
        }
    }
}
//Afegim l'article a la llista de posts.
try{
    this.posts.add(post);
}catch (NullPointerException e){
    System.out.println("Error al añadir");
}
}
}

/**
 * Configuració de la data.
 * @data Data a modificar.
 * @mode Estil de la data.
 * @return Data modificada.
 */
private String Dataesp (String data, String mode){
    String dia = data.substring(8, 10);
    String mes = data.substring(5, 7);
    String any = data.substring(0, 4);
    String hora = data.substring(11, 19);
    String nova = "";
    if(mode == "EUR") nova = dia+"-"+mes+"-"+any;
    if(mode == "WP") nova = any+"-"+mes+"-"+dia+" "+hora;
    return nova;
}

/**
 * Assignació de la Url del feed.
 * @loc Localitzador de la comissió.
 */
public void setUrl (String loc){
    String path =
"http://espais.eic.cat/webcontent/rest/library/"+loc+"/feed";
    this.url = path;
}

```

```

}

/**
 * Autenticació al feed i recuperació del resultat.
 * @url Direcció del feed.
 * @return String amb el contingut del feed.
 */
public String conecta (String url){
    GetMethod post = null;
    String contingut = "";
    String target = "espais.eic.cat";

    try {
        post = new GetMethod(url);
        HttpClient client = new HttpClient();
        Credentials defaultcreds = new UsernamePasswordCredentials(this.usr,
this.pass);
        client.getState().setCredentials(new AuthScope(target, 80,
AuthScope.ANY_REALM), defaultcreds);
        client.executeMethod(post);
        contingut = post.getResponseBodyAsString();

    } catch(IOException ioe) {
        ioe.printStackTrace();
    } finally {
        post.releaseConnection();
    }
    return contingut;
}

/**
 * Mètode que converteix un string a un document Xml.
 * @xmlSource String a convertir.
 * @return Document Xml.
 */
public static Document stringToDom(String xmlSource)
    throws SAXException, ParserConfigurationException, IOException {

    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder builder = factory.newDocumentBuilder();
    return builder.parse(new InputSource(new StringReader(xmlSource)));
}
}

```

12.4.4 Connexió entre *Quickr* i *WordPress* (signinremot.php)

Arxiu que recupera la petició de *Quickr* d'exportar articles, inicia sessió a *WordPress* i posteriorment la tanca.

```
<?php
require( dirname(__FILE__) . '/wp-load.php' );
require_once( ABSPATH . WPINC . '/user.php' );
require_once( ABSPATH . WPINC . '/pluggable.php' );

$creds = array();
$in = $_GET['in'];
//Cas en que es tingui que fer login.
if($in == "true"){
    //Recupera l'usuari, el password, la comissió origen, l'article i el blog
    destí.
    $creds['user_login'] = $_GET['u'];
    $creds['user_login'] = base64_decode($creds['user_login']);
    $creds['user_password'] = $_GET['p'];
    $creds['user_password'] = base64_decode($creds['user_password']);
    $creds['remember'] = false;
    $com = $_GET['com'];
    $art = $_GET['art'];
    $b = $_GET['b'];
    //Inicia sessió.
    $user = wp_signon( $creds, false );
    //Redirecciona al plugin QuickrToWordPress de WordPress per insertar
    //les dades.
    header( 'Location:wp-admin/options-
general.php?page=Quickoptions&quick=s&com='.$com.'&art='.$art.'&b='.$b );
    if ( is_wp_error($user) )
        echo $user->get_error_message();
}
//Cas de desconnexió. Fa un logout.
elseif ($in == "false"){
    wp_logout();
}

?>
```

12.5 Funcionalitats diverses

12.5.1 Inserció, modificació i eliminació d'usuaris (usuaris.php)

Arxiu corresponen al registre d'usuaris mitjançant un trigger.

```
<?php
require( dirname(__FILE__) . '/wp-load.php' );
require_once( ABSPATH . WPINC . '/registration.php' );
require_once( ABSPATH . WPINC . '/pluggable.php' );

global $wpdb;
if(isset($_GET['accio'])) $accio = $_GET['accio'];
if(isset($_GET['login'])) $login = $_GET['login'];
$login = base64_decode($login);
if(isset($accio) && !(is_null($accio))){
    if((strcmp($accio, "insert") == 0) or (strcmp($accio, "update") == 0)){
        if(isset($_GET['pass'])) $pass = $_GET['pass'];
        $pass = base64_decode($pass); //Codificació en base64.
        $pass = wp_hash_password($pass);
        if(isset($_GET['email'])) $email = $_GET['email'];
    }

    if(strcmp($accio, "delete") != 0){
        if(strcmp($accio, "insert") == 0){
            //Insercció d'usuaris: LOGIN, PASS, EMAIL.
            $dades = array(
                'user_login' => $login,
                'user_pass' => $pass,
                'user_email' => $email);

            wp_insert_user($dades);
            echo "Usuari inserit correctament<br>";
        }
        if(strcmp($accio, "update") == 0){
            //Actualització d'usuari: PASS NOU, EMAIL NOU.
            $us = get_userdata_bylogin($login);
            $dni = $us->ID;
            $dades = array('ID' => $dni);
            $boolpas = ($pass != "" and ($pass != null));
            $boolmail = ($email != "" and ($email != null));
            //Actualització de password.
            if($boolpas and !($boolmail)){
                $dades = array('ID' => $dni,
                    'user_login' => $login,
                    'user_pass' => $pass);
            }
            //Actualització de mail.
            elseif(!($boolpas) and $boolmail){
                $dades = array('ID' => $dni,
                    'user_login' => $login,
                    'user_email' => $email);
            }
            //Actualització de mail i password.
            elseif($boolpas and $boolmail){
                $dades = array('ID' => $dni,
                    'user_login' => $login,
                    'user_pass' => $pass,
                    'user_email' => $email);
            }
        }
    }
}
```

```

        echo "Lo que entra<br><br>";
        $i = wp_insert_user($dades);
        if(is_int($i)) echo "Usuari actualitzat correctament<br>";
        else echo "Error en l'actualització de l'usuari<br>";
    }
}
else{
    //El·liminació d'usuaris.
    $us = get_userdata_by_login($login);
    $dni = $us->ID;
    wp_delete_user($dni, 1);
    echo "Usuari esborrat correctament<br>";
}
}

//Funció de WordPress modificada per fer una reassignació correcte dintre d'un
//multisite i esborri després l'usuari.
function wp_delete_user( $id, $reassign = 'novalue' ) {
    global $wpdb;
    $id = (int) $id;
    do_action('delete_user', $id);
    $reassign = (int) $reassign;
    $sql_ids = "SELECT blog_id FROM ".$wpdb->prefix."blogs;";
    $idblogs = $wpdb->get_results($sql_ids);
    foreach ($idblogs as $blog){
        switch_to_blog($blog->blog_id);
        $wpdb->update( $wpdb->posts, array('post_author' => $reassign),
            array('post_author' => $id) );
        restore_current_blog();
    }

    clean_user_cache($id);
    if ( !is_multisite() ) {
        $wpdb->query( $wpdb->prepare("DELETE FROM $wpdb->usermeta WHERE user_id
= %d", $id) );
        $wpdb->query( $wpdb->prepare("DELETE FROM $wpdb->users WHERE ID = %d",
$id) );
        echo "borrado sin multisite<br>";
    } else {
        $level_key = $wpdb->get_blog_prefix() . 'capabilities';
        $wpdb->query("DELETE FROM $wpdb->usermeta WHERE user_id = $id AND
meta_key = '{$level_key}'");
        $wpdb->query( $wpdb->prepare("DELETE FROM $wpdb->users WHERE ID = %d",
$id) );
    }

    // allow for commit transaction
    do_action('deleted_user', $id);

    return true;
}
?>

```

12.5.2 Fitxa d'usuari (articles.php)

Arxius corresponent a la devolució dels articles que ha escrit un usuari via *Json* per la fitxa d'usuari.

```
<?php
//Resposta a JSON
header('Content-type: application/json');

require( dirname(__FILE__) . '/wp-load.php' );
require_once(ABSPATH.' /wp-admin/includes/user.php');
require_once(ABSPATH.' /wp-includes/pluggable.php');

global $wpdb;

if(isset($_GET['login'])){
    //Recollida del login i decodificació.
    $login = $_GET['login'];
    $login = base64_decode($login);
    //Recuperació de l'identificador de l'usuari.
    $datos = get_userdata_by_login($login);
    $id = $datos->ID;
    //Blogs de l'usuari.
    $user_blogs = get_blogs_of_user($id);
    echo "[";
    $conta = 0;
    //Per cada blog al que pertany l'usuari.
    foreach ($user_blogs as $b){
        $idblog = $b->userblog_id;
        if($idblog != 1){
            if($conta != 0) echo ",";
            $sql_posts = "SELECT * FROM ".$wpdb->prefix.$idblog."_posts
                WHERE post_author = ".$id.";";
            $posts_au = $wpdb->get_results($sql_posts);

            $cnt = 0;
            //Per cada article escrit pe l'usuari.
            foreach ($posts_au as $p){
                if($cnt != 0) echo ",";
                echo "{";
                echo "\"titol\": ";
                echo "\"".$p->post_title.\"\", ";
                echo "\"data_publicacio\": ";
                echo "\"".$p->post_date.\"\", ";
                echo "\"url\": ";
                echo "\"".$p->guid.\"\", ";
                echo "\"blog\": ";
                echo "\"".$b->blogname.\"\"";

                echo "}";
                $cnt++;
            }
            $conta++;
        }
    }
    echo "];";
}
?>
```


12.5.3 Restricció de seguretat a l'accedir a un article (single.php)

```
<!-- MIDDLE (Main page content) =====>-->
<div id="content">
<?php //Usuari amb sessió iniciada.
    if($user_ID){?>
        <?php if (have_posts()) {
            while (have_posts()) {
                (the_post()); ?>
                <!-- Post -->
                <div class="post box">
                    <!-- [title] -->
                    <h3 class="post-title"><a href="<?php the_permalink(); ?>"><?php
the_title(); ?></a></h3>
                    <!-- [meta] -->
                    <div class="post-meta">
                        <a class="post-author" href="<?php the_author_link();
?>"><?php the_author(); ?></a>
                        <span class="post-category"><?php the_category(' ', '); ?></span>
                        <span class="post-date"><?php the_date(); ?></span>
                        <span class="post-comments"><a href="<?php comments_link(); ?>
"><?php comments_number('No Comments', 'One Comment', '% Comments'); ?></a></span>
                    </div>
                    <!-- [post content] -->
                    <?php the_content(); ?>
                </div> <!-- // End of Post -->
                <?php comments_template(); ?>
            <?php } } else { ?>
        <div class="post box"> <h3>El post ni hi &eacute;s dispnible.</h3></div>
        <?php }
    }
//Usuari sense iniciar sessió. Es mostra el formulari de login.
else{
    $ar = array(
        'echo' => true,
        'redirect' => $post->guid,
        'form_id' => 'loginform',
        'label_username' => __( 'Username' ),
        'label_password' => __( 'Password' ),
        'label_remember' => __( 'Remember Me' ),
        'label_log_in' => __( 'Log In' ),
        'id_username' => 'user_login',
        'id_password' => 'user_pass',
        'id_remember' => 'rememberme',
        'id_submit' => 'wp-submit',
    );
    wp_login_form($ar);
?>
    <link media="all" type="text/css" href="http://blogs.eic.cat/wp-
admin/css/login.css" id="login-css" rel="stylesheet">
<link media="all" type="text/css" href="http://blogs.eic.cat/wp-
admin/css/colors-fresh.css" id="colors-fresh-css" rel="stylesheet">
    <?php
}
?>
</div>
```

13. Conclusions

Amb aquest projecte he après sobretot, a més de molts conceptes, com es realitzen les coses dintre del món laboral. Poc a poc he anat descobrint com les necessitats de la gent mai són clares, no es corresponen el que diuen amb el que volen. Aquesta crec que és la millor i la més valiosa lecció que he pogut aprendre.

En relació amb la proposta original s'ha pogut comprovar que hi ha moltes coses que han canviat o que no s'han fet realitat, com és el cas dels *Newsletters* (no implementats) o els *RSS*, que finalment s'han aprofitat els que ofereix *WordPress*, i és que, com deia, quan es treballa segons les necessitats de la gent, la feina es distorsiona i varia molt amb el temps.

S'ha partit d'un idea d'inici, el fet de crear una blogosfera i una proposta de com dur a terme aquesta feina, mitjançant l'eina *IBM Portal*. Fins que no ha anat passant el temps no s'han pogut identificar les necessitats reals que feien falta, de manera que no es podia fer un treball fitat, seguint una sèrie de passos.

És per això que no s'ha pogut dur a terme el procés teòric estudiat, com és el fet d'estudiar un problema sencer, fer un anàlisi, un disseny i finalment una implementació de la solució. En canvi, he estudiat cadascuna de les parts del problema conforme anaven apareixent i fent les pertinents modificacions sobre la marxa. En tot moment he intentar mantenir una estabilitat i una concordança amb tot el que havia fet anteriorment, d'aquí que per cadascun dels mòduls implementats hagi decidit continuar amb el que estava ja implementat fins a l'anterior plugin i fer la ampliació amb les modificacions pertinents. En tot moment he tractat de mantenir un sistema complet i conseqüent en la part pràctica amb els estudis fets en la part teòrica.

Finalment el resultat ha estat una eina totalment diferent a la pensada al començament. S'ha passat d'una eina amb un tipus de tecnologia i amb unes característiques que són totalment diferents al resultat final.

En canvi, s'han fet les implementacions pertinents, per tal que el gran avantatge que oferia *IBM Portal*, total integració en el sistema del COEIC, també el pugui oferir *WordPress*.

D'aquesta manera, s'ha aconseguit el gran objectiu que es cercava per el col·legi d'enginyers, que era el de poder englobar tots els blogs que tenien arreu del món en diferents formats diferents (*Quickr*, portals de *WordPress.com*) en un de sol que serveixi per tothom i on els entesos en la matèria pugin donar el seu punt de vista amb total facilitat i visibilitat per el seu col·lectiu, independentment del tipus de la comissió de la que formi part, i on els quantios grup d'enginyers industrials titulats i apunt de titular-se, puguin tenir un lloc de referència per estar al corrent de totes les notícies que hi ha dintre del seu col·lectiu.

Creo que s'han complert els objectius en que es basa un projecte final de carrera, i és que no hi ha millor forma de finalitzar els estudis i començar a treballar, que aconseguint juntar les dues coses, donat que així, es fusiona el finalitzament d'una amb l'inici de l'altre.

14. Bibliografia

- [1] WordPress.com. *WordPress Codex*. <http://codex.wordpress.org/Main_Page> [Darrera consulta: 17-01-2011]
- [2] PHP. *PHP: Documentation*. <<http://php.net/docs.php>> [Darrera consulta: 28-12-2010]
- [3] Wikimedia Foundation. *Wikipedia*. <<http://es.wikipedia.org/wiki/Wikipedia:Portada>> [Darrera consulta: 10-01-2011]
- [4] Merenciano, Josep M. *Enginyeria del software: Disseny*. Edició UPC, 2009.
- [5] AyudaWordPress. *Foro AyudaWordPress*. <<http://ayudawordpress.com/foro/>> [Darrera consulta: 25-11-2010]
- [6] Enginyers Industrials de Catalunya. *Enginyers - Portal*. <<http://www.eic.cat>> [Darrera consulta: 05-10-2010]
- [7] Tim Hanis. *Portlet Development Workbook. Using the Standard API*. [Online]. <http://www.ieee.org/portal/cms_docs/pubs/transactions/auinfo03.pdf>
- [8] IEEE Periodicals Transactions / Journals Department. *Information for Authors*. The Institute of Electrical and Electronics Engineers, Inc, 2006.