

GRIAL 2.0

Una propuesta de integración de servicios y aplicaciones web en un portal académico personalizable

Proyecto de Fin de Carrera

Ingeniería Informática



**VNiVERSiDAD
D SALAMANCA**

Enero de 2011

Alicia García Holgado

Francisco José García Peñalvo

Agradecimientos:

- A mi tutor, Francisco J. García Peñalvo, y a todos los miembros del Grupo GRIAL por haber confiado en mí.
- A mi familia por haberme apoyado siempre y haber sufrido conmigo los momentos de desesperación.
- A mis pequeños que consiguen hacerme sonreír siempre.
- A mis amigos por estar ahí, tanto en los buenos como en los malos momentos.

Tabla de contenido

1. INTRODUCCIÓN	1
2. OBJETIVOS	7
2.1. OBJETIVOS <i>SOFTWARE</i>	7
2.2. OBJETIVOS TÉCNICOS	8
2.3. OBJETIVOS PERSONALES	9
3. CONTEXTO TEÓRICO.....	10
3.1. PORTALES WEB 2.0	10
3.2. COMPONENTES DE UN PORTAL ACADÉMICO 2.0.....	11
3.2.1. <i>Gestor de contenidos</i>	11
3.2.2. <i>Plataforma de eLearning</i>	12
3.2.3. <i>Gestor de blogs</i>	13
3.2.4. <i>Repositorio</i>	13
3.2.5. <i>Redes Sociales</i>	14
3.2.6. <i>Otros</i>	15
3.3. TECNOLOGÍA DE DESARROLLO PARA LA WEB 2.0	15
3.3.1. <i>Servicios web</i>	15
3.3.2. <i>Mashup</i>	16
3.3.3. <i>Integración de servicios</i>	17
4. TÉCNICAS Y HERRAMIENTAS.....	18
4.1. COMPONENTES DEL SISTEMA	18
4.1.1. <i>Drupal</i>	18
4.1.2. <i>Moodle</i>	18
4.1.3. <i>WordPress</i>	18
4.1.4. <i>DSpace</i>	20
4.1.5. <i>Multimedia</i>	21
4.2. NUEVAS VERSIONES EN LAS HERRAMIENTAS UTILIZADAS	29
4.2.1. <i>WordPress MU 2.9.2 vs. WordPress 3.0</i>	29
4.2.2. <i>Moodle 1.9 vs. Moodle 2.0</i>	30
4.2.3. <i>Drupal 6.20 vs. Drupal 7</i>	30
4.3. DESARROLLO	30
4.3.1. <i>Metodología</i>	30
4.3.1.1. Herramientas CASE	31
4.3.1.1.1. Gestor de proyectos.....	31
4.3.1.1.2. Herramientas de modelado	32
4.3.1.1.3. Gestor de versiones	33

4.3.1.2.	<i>Framework</i>	33
4.3.1.3.	PHP	34
4.3.2.	<i>El entorno de documentación</i>	34
4.3.2.1.	Entorno <i>online</i> – el <i>blog</i> del proyecto	34
4.3.2.2.	Entorno ofimático	35
5.	DESCRIPCIÓN DEL SISTEMA	36
6.	ASPECTOS RELEVANTES	39
6.1.	EL CICLO DE VIDA: ITERACIONES E INCREMENTOS	39
6.2.	ARQUITECTURA DEL SISTEMA	41
6.2.1.	<i>Arquitectura lógica</i>	41
6.2.1.1.	Servicios web	41
6.2.1.1.1.	Drupal web services	43
6.2.1.1.2.	Moodle web services	44
6.2.1.1.3.	WordPress web services	44
6.2.1.1.4.	SOAP vs. XML-RPC	45
6.2.1.2.	Conexión de componentes	46
6.2.1.2.1.	Conectar Drupal con WordPress: <i>Drupal to WP</i> y <i>Drupal to WP XML-RPC</i>	46
6.2.1.2.2.	Conectar Drupal con Moodle: <i>Drupal to Moodle</i> y <i>Drupal to Moodle xmlrpc</i>	49
6.2.2.	<i>Arquitectura física</i>	50
6.3.	<i>SINGLE SIGN ON</i>	52
6.3.1.	<i>OpenID provider de GRIAL</i>	52
6.4.	DESARROLLO EN DRUPAL	55
6.4.1.	<i>Research Support</i>	55
6.4.2.	<i>Drupal theme. La imagen corporativa de GRIAL</i>	60
6.5.	AGORÁ. EL GESTOR DE <i>BLOGS</i> DE GRIAL	62
6.5.1.	<i>Portal social</i>	62
6.5.2.	<i>Funcionalidad: selección de plugins</i>	62
6.5.3.	<i>Multisite Global Search</i>	65
6.5.4.	<i>BuddyPress Activity Tags</i>	73
6.5.5.	<i>WordPress theme. La imagen corporativa de GRIAL</i>	75
6.6.	PÓLIS. EL CAMPUS <i>ONLINE</i> DE GRIAL	79
6.6.1.	<i>User categories</i>	79
6.6.2.	<i>Estadísticas del curso</i>	80
6.6.3.	<i>Informe y bloque de estadísticas globales</i>	81
6.6.4.	<i>Informe de actividad</i>	82
6.7.	MISCELÁNEA	83
6.7.1.	<i>Varias instancias de DSpace en un mismo servidor</i>	83
6.7.2.	<i>Diseminación de la información</i>	85
6.7.3.	<i>Gestión dinámica de enlaces recomendados</i>	91

7. PROYECTOS RELACIONADOS.....	93
7.1. <i>DRUPAL Y MOODLE</i>	93
7.2. <i>DRUPAL Y WORDPRESS</i>	93
8. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	94
9. REFERENCIAS.....	98

Tabla de figuras

Figura 1. Antiguo portal de GRIAL	1
Figura 2. Propuesta para GRIAL 2.0	4
Figura 3. Evolución de la Web.....	10
Figura 4. Mapa Meme de la Web 2.0 (http://nww.nerdwideweb.com/web20/es.png) ...	11
Figura 5. Esquema básico de un LMS (García, 2008).	12
Figura 6. Logotipos de Blogger y WordPress.....	13
Figura 7. Ejemplo de estructura social.....	14
Figura 8. Funcionamiento de los servicios web. Fuente: W3C.....	16
Figura 9. Sitios que utilizan WordPress.....	19
Figura 10. <i>Microsite</i> en WordPress.....	20
Figura 11. Gredos. Repositorio de la USAL	20
Figura 12. RSS de la colección "Conferencias" del repositorio de GRIAL.....	21
Figura 13. Nube de servicios de GRIAL.....	22
Figura 14. Facebook de GRIAL.....	23
Figura 15. Twitter de GRIAL.....	23
Figura 17. Vimeo de GRIAL	24
Figura 16. Flickr de GRIAL.....	24
Figura 18. YouTube de GRIAL	25
Figura 19. SlideShare de GRIAL	25
Figura 20. Delicious de GRIAL	26
Figura 21. Gravatar de GRIAL	26
Figura 22. WordPress.com de GRIAL	27
Figura 25. Bit.ly de GRIAL	28
Figura 23. Estadísticas de AddThis.....	28
Figura 24. Botón de AddThis utilizado en GRIAL.....	28
Figura 26. WordPress MU a WordPress 3.0	29
Figura 27. Comentarios de <i>Multisite Global Search</i>	35
Figura 28. Componentes de GRIAL 2.0	38
Figura 29. Diagrama de Gantt correspondiente al primer incremento	40
Figura 30. Diagrama de Gantt del incremento de Agorá finalizado	40
Figura 31. Diagrama de Gantt del incremento de Pólis finalizado	41
Figura 32. Diagrama de Gantt de los dos últimos incrementos	41
Figura 33. Esquema del protocolo XML-RPC.....	42

Figura 34. Protocolo SOAP.....	43
Figura 35. Protocolo REST	43
Figura 36. Opción de activación de XML-RPC en WordPress	47
Figura 37. Configuración del módulo <i>Drupal to WP</i>	48
Figura 38. Activar el uso de XML-RPC en Moodle	49
Figura 39. Configuración del módulo <i>Drupal to Moodle</i>	50
Figura 40. Diagrama de despliegue del sistema.....	51
Figura 42. Formulario de <i>login</i> de Moodle	53
Figura 41. Formulario de <i>login</i> de WordPress	53
Figura 43 Formulario de <i>login</i> de Drupal	54
Figura 44. Community-ID de GRIAL.....	54
Figura 45. Ejemplo de miga de pan o <i>breadcrumb</i>	55
Figura 46. Primer paso para crear un contenido de tipo <i>Research Support</i>	57
Figura 47. Campos para crear un proyecto de investigación	58
Figura 48. Parte del formulario de proyecto para crear un <i>microsite</i>	58
Figura 49. Parte del formulario de proyecto para crear cursos	59
Figura 50. Módulo <i>Twitter</i>	59
Figura 51. Vista previa del proyecto antes de ser publicado.....	60
Figura 52. Portal de GRIAL con el tema Zen activado.....	61
Figura 53. Portal GRIAL con el tema Grial activado	62
Figura 54. Uso de <i>WPMU Global Search</i> en Diarium.....	66
Figura 55. Estadísticas de descargas de <i>WPMU Global Search</i> (http://wordpress.org/extend/plugins/wpmu-global-search/stats).....	67
Figura 56. Formulario horizontal	68
Figura 57. Formulario vertical	68
Figura 58. Configuración del <i>widget Multisite Global Search</i>	69
Figura 59. Estadísticas de descargas de <i>Multisite Global Search</i> (http://wordpress.org/extend/plugins/multisite-global-search/stats).....	70
Figura 60. <i>Multisite Global Search</i> en Terrapinn blogs.....	71
Figura 61. Multisite Global Search en MVN - Most Valuable Network	71
Figura 63. Multisite Global Search en el blog central de Centros Educativos de la Comunidad de Canarias	72
Figura 62. <i>Multisite Global Search</i> en los Blogs Comunitat Valenciana	72
Figura 64. Nube de etiquetas generada por <i>BuddyPress Activity Tags</i>	73
Figura 65. Estadísticas de descargas de <i>BuddyPress Activity Tags</i> (http://wordpress.org/extend/plugins/buddypress-activity-tags/stats)	74

Figura 66. <i>BuddyPress Activity Tags</i> en Planet Aveyron	75
Figura 67. <i>BuddyPress Activity Tags</i> en Millenium Project Nodes	75
Figura 68. Ejemplo de menús en el <i>theme</i> de GRIAL	76
Figura 69. <i>Sidebars</i> posibles en el <i>theme</i> GRIAL para WordPress	77
Figura 70. Funciones de los comentarios en el <i>theme</i> de GRIAL.....	78
Figura 71. Ejemplo de uso de Google Font API	78
Figura 72. Ejemplo de utilización del módulo <i>User Categories</i> para Moodle	79
Figura 73. Ejemplos del bloque de estadísticas de un curso de Moodle.....	80
Figura 74. Ejemplo del bloque de estadísticas globales de Moodle	81
Figura 75. Informe estadísticas globales de Moodle.....	82
Figura 76. Comienzo y fechas con cursos activos de la tabla de actividad	82
Figura 77. Imagen corporativa de GRIAL para DSpace.....	85
Figura 78. Diagrama de componentes de la diseminación de la información	86
Figura 79. Diseminación de la información en GRIAL	87
Figura 80. Publicar en el blog del proyecto	88
Figura 81. Se envía la publicación a Twitter.....	89
Figura 82. Se publica en el blog de GRIAL.....	89
Figura 83. Se envía la publicación al Twitter de GRIAL	90
Figura 84. Publicar en mi Facebook	90
Figura 85. Se publica en el Facebook de GRIAL	91
Figura 86. Diagrama de Actividad. Publicar enlaces de interés en el portal	92
Figura 87. GRIAL	96

1. Introducción

El **GR**upo de Investigación en Inter**Acción** y e**Learning** (GRIAL) es un grupo de investigación reconocido en la Universidad de Salamanca y Grupo de Excelencia de la Junta de Castilla y León, compuesto por un nutrido grupo de investigadores de diferentes ámbitos de conocimiento, desde perfiles puramente técnicos a perfiles de humanidades. Esta heterogeneidad de conocimientos entre sus miembros permite que las iniciativas realizadas sean tratadas desde diferentes puntos de vista, lo que aporta un enriquecimiento de las mismas.

Muchas veces el conocimiento que es generado por cualquier persona o institución se pierde en gran parte porque no llega al público de interés. Crear conocimiento, llevar a cabo estudios en temas innovadores o desarrollar métodos y modelos didácticos, son tareas de investigación que se pierden si no se les da utilidad y se les intenta sacar el máximo partido. Para que no se den ese tipo de situaciones se le debe dar visibilidad a ese conocimiento.

Este problema puede verse dentro de GRIAL, se genera gran cantidad de conocimiento, pero al no darle la visibilidad adecuada se pierde. Esto provoca que el grupo de cara al exterior sea una caja negra, dentro de GRIAL hay gran cantidad de información pero no es sencillo acceder a ella. La causa principal de esta situación es que no se había cuidado una imagen pública del grupo combinada con una adecuada estrategia de diseminación del conocimiento. El portal de GRIAL, realizado en la plataforma Joomla! (<http://www.joomla.org>), se encontraba desactualizado, con información antigua y escasa información de interés debido al proceso necesario para actualizar los contenidos, completamente ligado a personas concretas, lo que se acaba convirtiendo en un embudo. Aunque se realicen proyectos de investigación, se lleven a cabo actividades formativas de tutoría *online* o se publiquen artículos, al no estar dicha información contenida en el portal es como si no existiera. Incluso secciones tan importantes como la mostrada en la Figura 1 están vacías.



Figura 1. Antiguo portal de GRIAL

Se daban varias circunstancias para que esto ocurra. En primer lugar la herramienta utilizada no era intuitiva para un usuario que desee añadir información, lo cual hace que los miembros del grupo de investigación eviten realizar dicha tarea. En segundo lugar dentro de GRIAL se crea y gestiona gran cantidad de información, esto provoca que la actualización del portal recaiga en una o dos personas, las que manejan la herramienta con más soltura, que no tienen capacidad para manejar el continuo flujo de trabajo que reciben de todos los miembros.

Bajo este escenario se plantea la necesidad de buscar una solución que permita dar visibilidad a toda esa información, y que permitiera que todos los miembros, o la mayoría, sean los encargados de gestionar el conocimiento que generan, de tal forma que, combinando el trabajo colaborativo y los flujos de trabajo adecuados, se consiga alcanzar una solución satisfactoria.

Lo que mejor se adapta a esta situación es proporcionar a GRIAL un portal académico construido bajo la filosofía de la llamada Web 2.0, un portal en constante construcción que involucre a todos los miembros y que tenga unos flujos de trabajo correctamente definidos que faciliten la gestión de la información.

En GRIAL se apuesta por el conocimiento abierto, es decir, poner a disposición de todo aquel que desee utilizarlo el conocimiento que se gesta en el grupo de investigación, así como permitir su reutilización. De esta forma se facilita una especie de realimentación, que redundará en una mayor producción de conocimiento, y además de mayor calidad.

Una de las condiciones para que una obra se considere conocimiento abierto es la ausencia de restricciones tecnológicas para acceder a ella, es decir, se debe disponer de la obra en un formato abierto, disponible públicamente y de manera totalmente gratuita. Los usuarios deben disponer de herramientas que tengan en cuenta estos aspectos, lo que facilite llevar a cabo la gestión de la gran variedad de información que se genera dentro de un grupo de investigación, desde noticias a proyectos y publicaciones de investigación o desarrollos *software*. Toda esa información, además de accesible, debe ser fácilmente localizable, si un usuario busca publicaciones realizadas por algunos de los miembros de GRIAL debe resultarles fácil localizarlas y, sobre todo, debe ser posible que conozca de su existencia. Este es otro de los problemas detectados en el grupo GRIAL, la información existía, y aunque se publique en el portal, no existía una red en la que las personas interesadas en alguna de las líneas de trabajo del grupo pudiera recibir información sobre la existencia de una nueva publicación o un nuevo curso. Se hace, por tanto, necesario explotar las posibilidades que brinda la Web 2.0, con todo el entramado social, para dar a conocer a GRIAL y lo que es más importante, llevar el conocimiento a mayor cantidad de personas.

En el grupo de investigación se llevan a cabo diferentes iniciativas formativas, desde un Máster en *eLearning* hasta cursos específicos de tutoría *online*, que muchas veces requieren el uso de herramientas experimentales que no pueden ser instaladas en una plataforma institucional o en un *software* propietario. Hasta el momento las iniciativas se encuentran desperdigadas en diferentes plataformas de *eLearning*, lo que obliga a los miembros encargados de tutorizarlas a tener que sufrir bruscos cambios de entorno y periodos de adaptación cada vez que se comienza una nueva iniciativa.

Otro aspecto que se debe cubrir es la necesidad de disponer de un espacio donde aplicar los diferentes conocimientos en *eLearning*, una de las principales líneas de trabajo de GRIAL, poniendo en práctica métodos y modelos didácticos de referencia en la formación *online* y proporcionando un entorno donde experimentar desarrollos *software* orientados a mejorar la calidad del *eLearning*.

En el portal académico se debe proporcionar también un espacio donde poder unificar toda la actividad formativa así como mantener un entorno capaz de soportar nuevos desarrollos que mejoren la experiencia del usuario, tanto de tutores o docentes como de los propios estudiantes. Dicho espacio deberá ser la plataforma de *eLearning* de GRIAL, que deberá ser un elemento más del portal académico.

Era necesario que GRIAL se actualizara, evolucionara su cara pública hasta el nivel que se encuentra todo el trabajo realizado por sus miembros, de esta manera debería ser mejor valorado, a pesar de que su calidad siga siendo excelente.

Cada individuo del grupo de investigación tiene mucho que comunicar, tanto formal como informalmente, ayudándose de todos los medios que tienen a su alcance para lograrlo. Mantener un *blog* o utilizar diferentes perfiles sociales para difundir sus ideas son algunos de estos medios. Ese conocimiento más social, más informal, que se puede compartir en un *blog* o enviar en un *tweet* tiene un gran valor para GRIAL y se estaba desaprovechando. Si se consigue que los miembros aporten también al grupo esos conocimientos, luego eso revertirá en ellos. Se trata de una relación de realimentación que hay que fomentar para que las dos partes se beneficien, el grupo como imagen de todos sus miembros, y cada uno de los miembros logrará una mayor repercusión.

Si en el portal académico se incluye un espacio de comunidad, donde permitir a los miembros y a otros usuarios con intereses comunes al grupo y donde se expresen libremente, el conocimiento permanecerá en GRIAL a la vez que obtendrá una mayor difusión, cumpliendo la máxima de darle visibilidad.

GRIAL necesita un portal académico que permita la gestión colaborativa de conocimiento abierto, así como un lugar que desempeñe la función de plataforma de *eLearning* y un espacio que sirva de contenedor para la comunicación y difusión de la información desde un punto de vista más social e informal. Todo ello con unos flujos de trabajo bien definidos que permitan que los usuarios tengan una visión del portal sencilla e intuitiva, que no les suponga esfuerzo utilizarlo y que les aporte ventajas respecto a la situación inicial.

El portal académico GRIAL 2.0, mediante la combinación de diferentes herramientas ya existentes, pero con una integración entre las mismas, será el cambio que GRIAL necesita. No se trata de poner a disposición de sus miembros una serie de herramientas para que trabajen con cada una de ellas como elementos independientes, el usuario debe percibir el portal como un conjunto de diferentes funcionalidades unidas para mejorar su experiencia como investigador. Todas los elementos son importantes para cubrir las necesidades existentes pero se debe proporcionar una infraestructura que aporte más, que permita que si GRIAL tiene que dar difusión a una conferencia de uno de sus miembros, todo el portal se haga eco de la noticia, de tal forma que el usuario, independientemente de por donde llegue, reciba la información.

En la Figura 2 se muestra una propuesta de portal 2.0, donde todo gira en torno al intercambio y difusión de la información utilizando diferentes vías para conseguirlo. El portal se compone de un repositorio de datos siguiendo la política del conocimiento abierto, un lugar de comunidad, que a partir de ahora se denominará Agorá, donde los usuarios puedan compartir sus conocimientos de una forma más informal, una plataforma para la formación *online*, que recibirá el nombre de Pólis, y como centro del sistema un portal donde dar visibilidad a todo aquello que se realiza dentro del grupo de investigación.

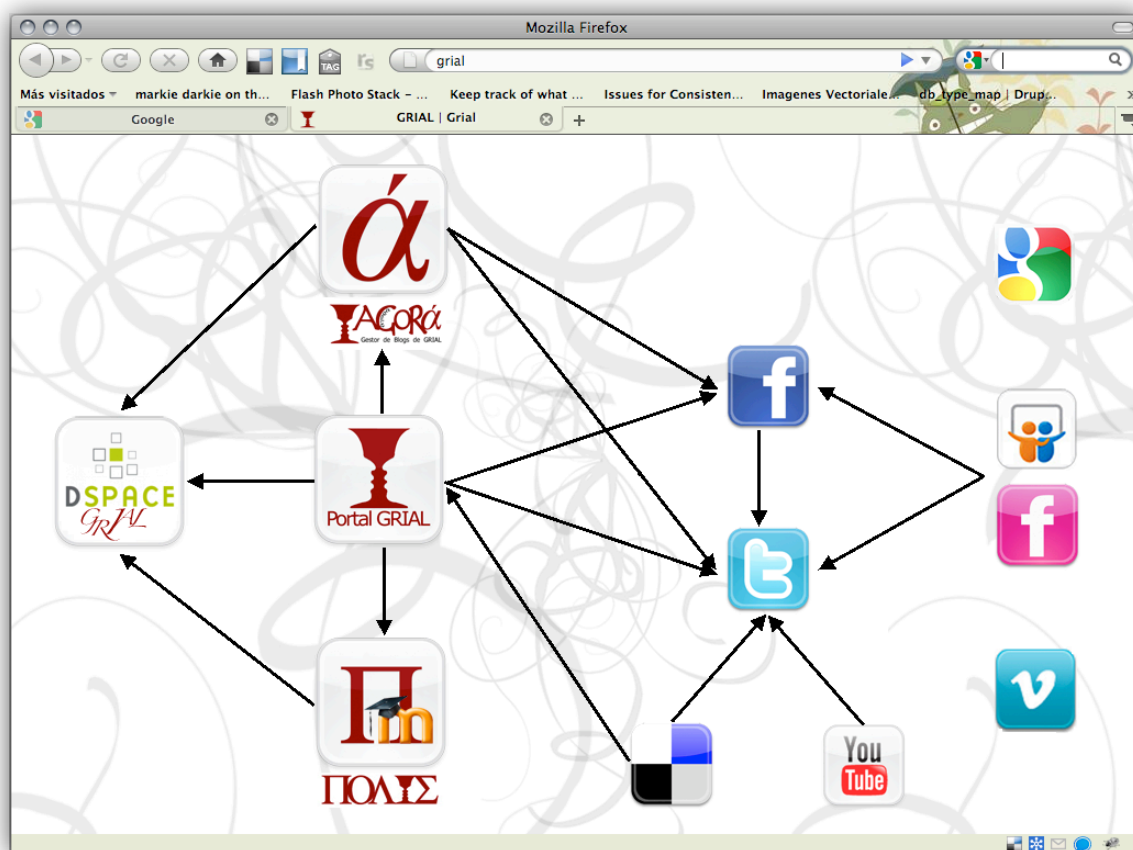


Figura 2. Propuesta para GRIAL 2.0

Para implementar el portal 2.0 de GRIAL se dispone de un conjunto de servidores nuevos que hay que configurar adecuadamente para que suplan las necesidades del grupo. Se debe construir el portal aprovechando la infraestructura existente y adecuándola a sus necesidades. Al disponer de más de un servidor no solo se deben tomar decisiones de cómo configurarlo y qué herramientas se necesitan para un correcto funcionamiento, sino que también hay que decidir cómo se distribuirá todo el sistema atendiendo a criterios de rendimiento, seguridad, estabilidad.

Para poder gestionar adecuadamente los elementos *hardware* es necesario tener claro los elementos *software* que van a ser necesarios para montar el portal. Se debe decidir cuidadosamente el conjunto de herramientas que van a utilizarse teniendo en cuenta factores técnicos y humanos. Para las aplicaciones, que van a utilizarse tanto por los miembros del grupo como por otros usuarios, debe tenerse en cuenta variables de usabilidad, eficiencia, robustez, consistencia, donde el usuario no siempre va a conocer cómo funciona un sistema en su totalidad, por lo que pueden darse situaciones en las que intente utilizar la aplicación de forma inadecuada. Es importante tener presentes esos criterios, ya que son claves a la hora de marcar la diferencia entre el viejo portal y el nuevo. Si al usuario no le resulta agradable y sencillo utilizar GRIAL 2.0 entonces no lo utilizará.

Tanto durante el desarrollo como durante la posterior utilización del portal pueden surgir diferentes problemas y hay que intentar ser conscientes de esa posibilidad. Al ser un portal que se va a basar en una gran cantidad de servicios ajenos al control del grupo puede ocurrir que algún servicio desaparezca o evolucione, lo que influye en la forma en la que el portal hace uso de él. El portal debe poder evolucionar, adaptarse a los

cambios tanto de las aplicaciones y servicios que lo componen como de los propios miembros del grupo de investigación. La Web 2.0 evoluciona, cambia, y un portal 2.0 debe poder hacer lo mismo. Hoy están de moda Facebook y Twitter, pero mañana surgirá una nueva herramienta y hay que tener capacidad de adaptación. El portal no son las herramientas sino la información.

La integración de todos los elementos del sistema requiere llevar a cabo desarrollos específicos que cubran necesidades propias del grupo de investigación. Los sistemas deben comunicarse entre sí, permitiendo el flujo de información entre los elementos pero a la vez deben ser lo suficientemente independientes para poder evolucionar y adaptarse por separado. Hay que dotar al portal de personalidad propia, dándole características que no se puede conseguir únicamente instalando y configurando herramientas. Se debe poner especial interés en que todos los desarrollos realizados sigan la filosofía del *software* libre, de tal forma que no sólo GRIAL se beneficie de la funcionalidad implementada, sino que también pueda hacerlo el resto del mundo.

Por último, no se puede dejar de lado la seguridad. El portal debe ser totalmente accesible para cualquier usuario, pero a la vez se debe controlar minuciosamente que la información y el conocimiento del grupo de investigación permanezcan íntegros. Se deben poner los medios necesarios para que el acceso a la gestión y creación de la información esté restringido a los miembros de GRIAL. En este punto entra en juego la gestión del acceso al sistema, al estar formado por un conjunto de herramientas, cada una de ellas con un método de acceso independiente, se obliga al usuario a acceder por separado a los diferentes elementos del portal. Es importante buscar una solución que unifique el acceso al sistema, utilizar un sistema de *Single Sign On* que proporcione mayor seguridad y trabajar con métodos de encriptación que protejan los datos privados de los usuarios.

Esta memoria refleja todo el proceso necesario para poner en marcha el portal académico de GRIAL, explicando los problemas surgidos, las decisiones que se han tomado, las soluciones que finalmente se han aplicado y las diferentes técnicas y herramientas utilizadas para ello. Se ha estructurado en varios apartados:

- *Objetivos*, se explican los objetivos perseguidos en la realización del proyecto, diferenciando entre tres tipos de objetivos: *software*, técnicos y personales.
- *Contexto teórico*, se realiza una introducción a los distintos conceptos relacionados con el dominio del problema de este proyecto para una mejor comprensión del mismo.
- *Técnicas y herramientas*, se explican las distintas técnicas utilizadas a lo largo del desarrollo del proyecto así como las herramientas que se han utilizado en cada etapa de desarrollo indicando el porqué de su elección.
- *Descripción del sistema*, donde se presentan de una forma visual los aspectos más destacados del portal, siempre desde una perspectiva funcional y de usuario.
- *Aspectos relevantes*, se realiza una descripción de los aspectos más interesantes y complejos (desde una perspectiva principalmente técnica) surgidos durante el proceso de desarrollo del proyecto.
- *Conclusiones y líneas de trabajo futuro*, contiene las conclusiones obtenidas al final de la realización del proyecto así como posibles ampliaciones futuras del mismo.

Este documento se complementa con un conjunto de documentación técnica que se entrega en formato digital. Los anexos que se han realizado son los siguientes:

- *Documento de Planificación Temporal*, se recoge de forma gráfica todas las actividades necesarias para producir el portal GRIAL 2.0 y el tiempo invertido en cada una de ellas.
- *Documento de Requisitos del Sistema*, se explican todos los requisitos que el sistema a desarrollar tiene que cumplir.
- *Documento de Análisis de Requisitos del Sistema*, donde se recogen un conjunto de diagramas para tener una idea clara de cómo solucionar los problemas y cubrir los objetivos planteados en el documento de requisitos.
- *Documento de Diseño del Sistema*, se muestra una visión de la arquitectura del sistema y del conjunto de información con el que se va a trabajar en el sistema.
- *Manual de instalación*, se proporciona una guía de instalación para los diferentes plugins y módulos desarrollados a lo largo del proyecto.
- *Manual del programador*, generado con la herramienta Doxygen (<http://www.doxygen.org>) a partir de los comentarios realizados en el código PHP.
- *Manual del usuario*, se recoge el conjunto de tutoriales desarrollados para explicar cómo se utilizan los diferentes sistemas que forman GRIAL 2.0.

2. Objetivos

Este apartado explica cuáles son los objetivos que se deben alcanzar al realizar el proyecto. Se hace una distinción entre tres tipos de objetivos.

2.1. Objetivos *software*

Los objetivos principales que deben ser cubiertos a la hora de desarrollar el portal 2.0 son los siguientes:

1. **Poner en explotación un Portal Académico 2.0 del Grupo GRIAL** que contenga los siguientes elementos:
 - El sistema debe tener un CMS, *Content Management System*, que permita mantener toda la información del grupo de investigación de manera ordenada y disponible para cualquier usuario.
 - El sistema debe proporcionar un método para gestionar *blogs*.
 - El sistema debe proporcionar un repositorio de datos donde almacenar y mantener la información digital del grupo.
 - El sistema debe tener un LMS en el que poder llevar a cabo las diferentes iniciativas de aprendizaje que impulsa el grupo de investigación.
 - El sistema debe poder manejar elementos multimedia tales como vídeos, presentaciones, fotos, etc.
 - A veces los proyectos de investigación requieren disponer de su propia página web o *microsite* dentro de GRIAL 2.0. El sistema debe disponer de un espacio para la creación de estos *microsites*.
2. **El sistema debe presentar un alto grado de integración y cohesión entre sus componentes**, es decir, debe definirse y desarrollarse una integración entre los diferentes elementos que conforman el Portal Académico.
 - Imagen corporativa de GRIAL. El sistema debe mantener el mismo *look & feel* en todos los elementos que lo conforman.
 - Autenticación *Single Sign On*. El sistema debe proporcionar un método de autenticación que permita al usuario acceder a los diferentes sistemas que conforman el portal con una sola instancia de autenticación.
 - Interoperabilidad. El sistema debe poder intercambiar datos y realizar operaciones entre los diferentes elementos que lo componen.
3. **El Portal debe tener una estructura centralizada y posibilitar un mantenimiento distribuido**. El sistema debe facilitar a los usuarios el mantenimiento de los contenidos del portal, de tal forma que sea una tarea rápida y sencilla.
4. **El Portal debe contar con un flujo de información automatizado entre los diferentes componentes del sistema**, de forma que se facilite el *workflow* de publicación de contenidos y su diseminación a través de los componentes del portal, como por ejemplo a través de las redes sociales
5. **Se debe asegurar una evolución sostenible de los diferentes elementos que conforman el portal**, sin que afecte al sistema en producción.

2.2. Objetivos técnicos

Existen también una serie de objetivos técnicos:

- Investigar, conocer y seleccionar sistemas/servicios 2.0 para la explotación de un portal académico.
- Utilizar Drupal (<http://drupal.org>) como gestor de contenidos.
- Utilizar Moodle (<http://moodle.org>) como plataforma de *eLearning*.
- Utilizar DSpace (<http://www.dspace.org>) como repositorio de datos.
- Utilizar WordPress (<http://wordpress.org>) como componente de gestión de *blogs*.
- Extender el CMS de GRIAL con desarrollos *ad-hoc*.
- Extender el LMS GRIAL con desarrollos *ad-hoc*.
- Extender el gestor de blogs con desarrollos *ad-hoc*.
- Desarrollar una API para facilitar la integración entre los diferentes sistemas que conforman el portal.
- Desarrollar un *wizard* para el mantenimiento sistemático de los contenidos del portal.
- Garantizar la neutralidad tecnológica del Portal, de forma que el sistema debe funcionar correctamente en las últimas versiones de los diferentes navegadores web actuales: Firefox, Opera, Chrome, Safari y IE8.
- Garantizar la diseminación de la información generada por el Grupo por la Red, mediante perfiles sociales en diferentes servicios web o redes sociales tales como Facebook, Twitter, SlideShare, YouTube (<http://www.youtube.com>), Vimeo (<http://vimeo.com>), etc.
- El sistema debe permitir que todos los miembros del grupo de investigación colaboren aportando contenidos al portal.
- Restringir la administración, así como la gestión de contenidos, a los miembros del grupo de investigación. Los usuarios ajenos al mismo solo obtendrán acceso al LMS para realizar los cursos pertinentes o a su *blog* personal, si previamente se les ha creado una cuenta en el gestor de *blogs*.
- Soportar un entorno multilingüe que permita, como mínimo, inglés y español.
- Desarrollar una interfaz intuitiva, amigable y consistente que facilite su uso a los diferentes perfiles de usuario del sistema, es decir, cuidar la usabilidad de todo el Portal
- Facilitar a los diferentes miembros del grupo de investigación la publicación en el *blog* principal de GRIAL, independientemente de que cada uno mantenga un *blog* individual.
- Proporcionar a los usuarios las herramientas necesarias para cubrir sus principales necesidades a la hora de administrar y mantener su *blog*.
- Organizar la presentación de los *blogs* de forma que se presenta el *blog* principal del grupo de investigación así como métodos para dar visibilidad a los *blogs* del sistema.

2.3. Objetivos personales

A parte de los objetivos sobre la aplicación en concreto, existen otra serie de objetivos que se pretenden alcanzar gracias al desarrollo del proyecto en sí:

- Colaborar activamente en el desarrollo de *software* libre poniendo a disposición de la comunidad los diferentes módulos y *plugins* que resulten del proyecto.
- Trabajar con nuevas herramientas y tecnologías desconocidas hasta el momento.
- Adquirir soltura a la hora de plantear soluciones frente a los problemas que pueden surgir a diario en el puesto de trabajo.
- Poner en práctica los conocimientos de ingeniería adquiridos acerca de planificación temporal, gestión de riesgos, etc. a lo largo de la carrera.
- Aprender a tratar con usuarios reales en un entorno en explotación.

3. Contexto teórico

3.1. Portales web 2.0

Un portal web se trata de un *website* que contiene gran cantidad de información, enlaces y servicios cuya función es facilitar al usuario el acceso a ese conjunto de recursos y herramientas. Hay varios tipos de portales dependiendo del tipo de información que contengan. Si los contenidos que conforman el portal están relacionados con un tema en particular, en este caso el grupo de investigación GRIAL, entonces se tratará de un portal vertical. En el caso de que los contenidos sean de ámbito general o abarquen un gran número de temas entonces se estará hablando de un portal horizontal.

El término Web 2.0, acuñado por Dale Dougherty y Tim O'Reilly en 2004 (O'Reilly, 2005), hace referencia a una nueva generación de webs basadas en la creación de páginas donde los contenidos son compartidos y producidos directa o indirectamente por los propios usuarios. Los usuarios han pasado de ser consumidores pasivos de información a producir contenidos para otros usuarios. A día de hoy cualquiera dispone de las herramientas necesarias para poder crear un *blog*, colaborar en un *wiki*, impartir un curso *online*, organizar su propia red social o compartir fotos, vídeos, música con otros usuarios. En la Figura 3 se recoge una de las diferentes infografías que presentan la evolución de la Web.

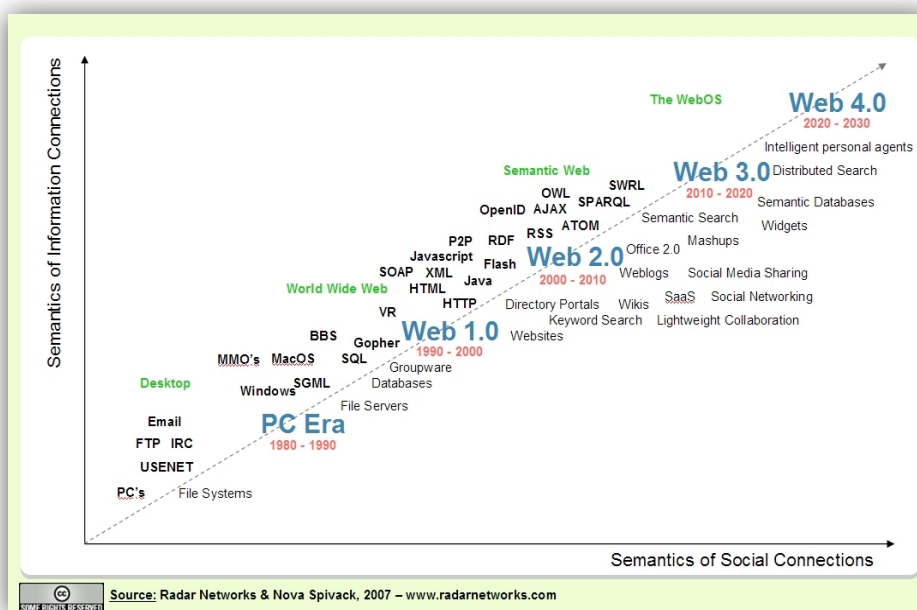


Figura 3. Evolución de la Web

Red social, *tag*, sindicación, *feed*, postear, folcsonomía, *blog*, son algunos de los términos que la Web 2.0 trae consigo y que ya forman parte de nuestro vocabulario. En este sentido en la Figura 4 se recoge un mapa meme¹ de la Web 2.0.

¹ Un meme es, según las modernas teorías sobre la difusión cultural, la unidad teórica de información cultural transmisible de un individuo a otro o de una mente a otra (o de una generación a la siguiente). Es un neologismo acuñado por Richard Dawkins en *El gen egoísta*, por la semejanza fonética con gen y para señalar la similitud radical con la memoria y la mimesis.



Figura 4. Mapa Meme de la Web 2.0 (<http://nww.nerdwideweb.com/web20/es.png>)

La difusión de la información se ha convertido en una tarea sencilla, ya no se necesita un periódico para publicar noticias, tan sólo debemos disponer de acceso a Internet para publicar en un *blog* y compartir en algún servicio como Menéame (<http://www.meneame.net>), la noticia probablemente será leída por más personas que si hubiera aparecido en un periódico de tirada nacional.

Los medios tradicionales han tenido que adaptarse a este cambio, buscando otras formas de negocio y utilizando las nuevas vías y posibilidades que ofrece la Web 2.0. Ahora podemos encontrar que una revista como Cinemanía se promociona colgando una miniserie, Descarga Completa, en su canal de YouTube o que un periódico como El País lanza una red social, Eskup (<http://eskup.elpais.com>), para compartir, aprender y dialogar sobre diferentes temas de actualidad creando así un medio de comunicación vivo.

Los portales web también han sufrido esta evolución, pasando a ser portales web 2.0 que permiten a sus usuarios interactuar con otros usuarios, compartir contenidos con otros sitios web, utilizar aplicaciones tales como YouTube, SlideShare, Digg (<http://digg.com>), dejando de ser un mero contenedor de información y servicios.

3.2. Componentes de un portal académico 2.0

3.2.1. Gestor de contenidos

Un gestor de contenidos o CMS es una plataforma que permite crear y administrar contenidos mediante una interfaz más o menos amigable que se conecta con una base de datos, de tal forma que cada vez que se carga una página dentro del CMS esta es leída de la base de datos y generada dinámicamente.

Se caracteriza por su enorme flexibilidad, permitiendo que cualquier usuario con ciertas nociones de administración web pueda crear su propia página sin necesidad de escribir ni una línea de código. Y en el caso de trabajar con un CMS de código libre se podrá extender su funcionalidad de forma sencilla hasta donde sea necesario siempre que se disponga de los conocimientos adecuados.

Existen diferentes tipos de gestores de contenido, dependiendo de su uso y funcionalidad podemos encontrar *blogs*, foros, *wikis*, plataformas para el comercio electrónico, etc. También se pueden clasificar, como todas las aplicaciones *software*, según su licencia, es decir, si se trata de *software* libre, como Drupal, Joomla! o OsCommerce (<http://www.oscommerce.com>), o privativo como OpenText (<http://www.opentext.com>).

3.2.2. Plataforma de eLearning

Una plataforma de eLearning, en inglés *Learning Management System* o LMS, es una herramienta que permite administrar, distribuir y controlar las actividades de formación *online*.

Los LMS son mucho más que la simple y tradicional página web estática asociada a un contenido informativo, pues se trata de aplicaciones que facilitan la creación de entornos de enseñanza/aprendizaje, mediante la integración de materiales didácticos, herramientas de comunicación, colaboración y gestión educativa. Es decir, combina aplicaciones como el correo electrónico, los foros de discusión, los cuestionarios, con los cuales el usuario interactúa para comunicarse con el profesor, realizar las actividades programadas y en definitiva seguir el curso en el que está matriculado.

Desempeñan un papel fundamental en el *eLearning* o aprendizaje *online*, y son utilizadas en un gran número de instituciones y organismos para impartir enseñanza no presencial, ya que los LMS ofrecen ambientes de aprendizaje ya diseñados e integrados donde los diferentes actores acceden a él mediante una clave personal, lo que permite configurar un espacio privado, dotado de las herramientas necesarias para aprender (comunicación, documentación, contenidos, interacción, etc.), en el que se facilita hacer un mejor seguimiento del progreso de los estudiantes.

En la Figura 5 se presenta el esquema básico de un LMS.

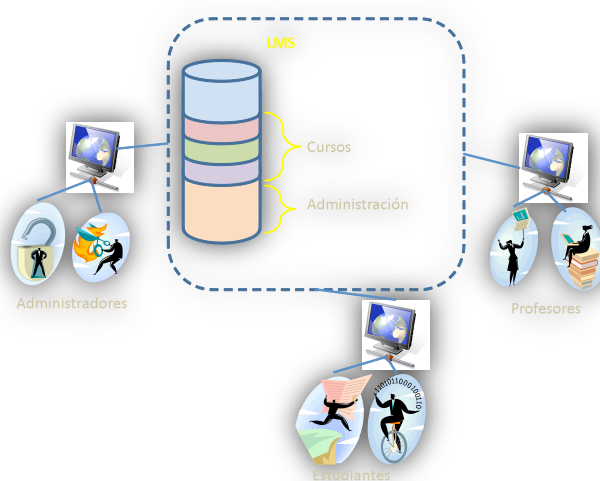


Figura 5. Esquema básico de un LMS (García, 2008).

Existen un gran número de plataformas de *eLearning*, tanto comerciales como de *software* libre. Entre las primeras cabe destacar WebCT, actualmente fusionada con Blackboard. Como sistema libre predomina cada vez más el uso de Moodle, tanto es así que muchas Universidades confían en esta solución para crear su plataforma de *eLearning* institucional, como es el caso de la Universidad de Salamanca en su plataforma Studium (<http://studium.usal.es>).

3.2.3. Gestor de blogs

El término *blog* es una abreviación de la palabra *weblog*. Un *blog* es un sitio web que se actualiza de manera más o menos periódica publicando entradas que pueden ser comentadas por otros usuarios. Existen en la actualidad una gran variedad de tipos de blogs dependiendo de su temática.

Se denomina gestor de *blogs* a un gestor de contenido cuya finalidad es permitir crear y administrar un número ilimitado de *blogs*.

Normalmente existe un *blog* principal o portal a través del cual los usuarios se registran en el sistema, bien para obtener un *blog* o para participar en otros *blogs* con comentarios o en algunos casos como colaboradores, dependiendo de la configuración de cada *blog* en particular.

Existen dos gestores de *blogs* muy conocidos y que son utilizados en todo el mundo por la calidad del servicio que ofrecen, Blogger y WordPress.com (Ver Figura 6). El primero forma parte de la gran familia de Google y el segundo es un servicio de gestión de *blogs* basado en un *software* con el mismo nombre.

Además de estos dos grandes servicios, existen herramientas para montar un gestor de *blogs* en cualquier servidor. Una de las herramientas más populares en este ámbito es el ya mencionado WordPress.



Figura 6. Logotipos de Blogger y WordPress

3.2.4. Repositorio

Un repositorio es un sistema que permite administrar de forma centralizada diferentes tipos de información digital. Dependiendo del tipo de contenidos que se almacenen en el repositorio se pueden distinguir diferentes tipos de repositorios, científicos, *software*, bibliográficos, institucionales, de objetos de aprendizaje, etc.

Mantener la información organizada en un repositorio facilita su localización y recuperación a partir de criterios de búsqueda como autor, fecha de creación, tema,

palabras clave, etc. Se puede definir el tipo de acceso al repositorio completo o a ciertos elementos que lo componen, de tal forma que algunas contenidos pueden estar protegidos mediante unas credenciales.

Utilizar un repositorio de datos para gestionar el conocimiento proporciona mayor seguridad que un simple disco duro accesible a través de Internet. Los repositorios suelen contar con sistemas de copia de seguridad y mantenimiento preventivo y correctivo, lo que hace que la información se pueda recuperar en el caso de que la máquina donde se encuentra quede inutilizable. Esta es una de sus características más importantes, ya que los repositorios de datos pueden alcanzar un gran tamaño, suponiendo una gran catástrofe si tras un fallo no hubiera la garantía de poder recuperar la información, suponiendo la pérdida del trabajo de muchas personas.

Su uso está muy extendido entre la comunidad universitaria, y es frecuente que cada universidad disponga de un repositorio institucional en el que ofrece una gran base de conocimientos multidisciplinares en formato digital.

3.2.5. Redes Sociales

Las Redes son formas de interacción social, definida como un intercambio dinámico entre personas, grupos e instituciones en contextos de complejidad. Un sistema abierto y en construcción permanente que involucra a conjuntos que se identifican en las mismas necesidades y problemáticas y que se organizan para potenciar sus recursos (Aruguete, 2001).

Según Wikipedia, las redes sociales son estructuras sociales compuestas de grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos.

La estructura social se puede representar en forma de uno o varios grafos, en los cuales los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos. (Ver Figura 7).

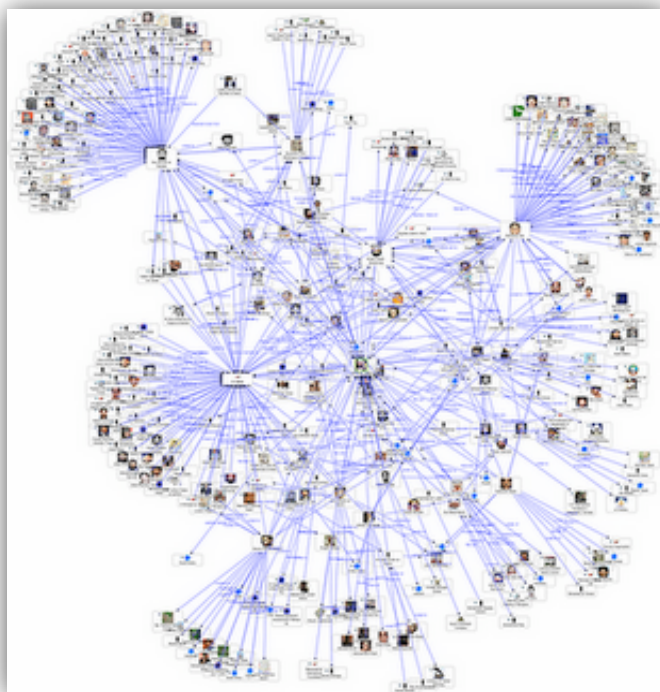


Figura 7. Ejemplo de estructura social

Las redes sociales en Internet son aplicaciones web en las que el individuo se conecta con otros individuos mediante relaciones de amistad, parentesco o intereses comunes. Para formar parte de una red social tan solo hay que crearse un perfil de usuario con algunos datos de carácter más o menos personal.

Las redes sociales tienen dos aspectos fundamentales:

- Hay conexión. Tiene que ver con quién está conectado con quién.
- Hay contagio. Que concierne a lo que fluye por los vínculos y lo que fluye se comporta de acuerdo con unas reglas.

Hay una serie de reglas relativas a las conexiones y a la forma en que se contagian las redes sociales, es decir, en cuanto a su estructura y función (Christakis y Fowler, 2010). Estos principios explican por qué los vínculos puedan hacer que el todo sea mayor que la suma de sus partes.

1ª regla: somos nosotros quienes damos forma a nuestra red.

2ª regla: nuestra red nos da forma a nosotros.

3ª regla: nuestros amigos nos influyen.

4ª regla: los amigos de nuestros amigos también nos influyen.

3.2.6. Otros

Existen otra serie de elementos que forman parte de un portal 2.0, una batería de servicios y herramientas *online* cuya principal función es compartir contenidos en diferentes formatos.

La inserción de vídeos, presentaciones, revistas digitales, fotografías, permiten sacar el máximo partido a la información, dotando al portal de elementos con los que el usuario pueda interactuar.

Este conjunto de herramientas, en concreto las utilizadas en GRIAL 2.0 se describen en el apartado 4.1.5.

3.3. Tecnología de desarrollo para la Web 2.0

3.3.1. Servicios web

El W3C (*World Wide Web Consortium*) define como servicios web un “conjunto de aplicaciones o tecnologías con capacidad para interoperar en la Web” (W3C). Dichas aplicaciones están diseñadas para intercambiar datos entre sí mediante APIs que pueden ser accedidas a través de la Red y ejecutarse en un sistema remoto.

Se pueden diferenciar dos tipos de aplicaciones, proveedores y consumidores o usuarios. Los primeros son aquellas aplicaciones que ofrecen servicios a través de una API, y los clientes o consumidores, como su propio nombre indica, son aquellos que hacen uso de dichos servicios.

Mediante el uso de una serie de estándares los servicios web permiten la comunicación entre aplicaciones independientemente de la plataforma donde se ejecuten o del lenguaje de programación en el que estén desarrolladas, permitiendo que las aplicaciones evolucionen por separado, siempre y cuando se mantenga la misma interfaz de servicio. Por ejemplo, un programa escrito en PHP puede conversar con otro escrito en Ruby, y aplicaciones corriendo bajo Windows pueden comunicarse con aplicaciones que se ejecutan bajo Unix.

El intercambio de datos se realiza utilizando un lenguaje estandarizado, XML, siguiendo uno de los protocolos existentes para ello como pueden ser SOAP o XML-RPC. Para que los consumidores puedan utilizar los servicios se debe definir una interfaz pública mediante WSDL, *Web Services Description Language*, un lenguaje especificado en XML que se ocupa de definir los servicios web como colecciones de puntos de comunicación capaces de intercambiar mensajes. En la Figura 8 se puede ver un ejemplo del funcionamiento de un servicio web.

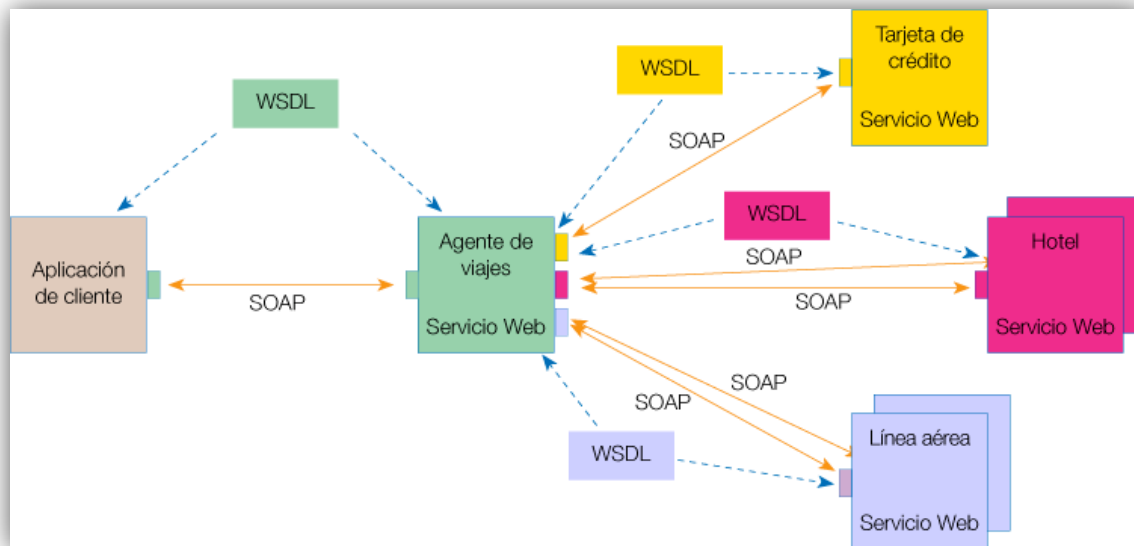


Figura 8. Funcionamiento de los servicios web. Fuente: W3C

Cada vez son más las aplicaciones que ofrecen servicios web, algunos ejemplos de ello son Twitter, Moodle, Amazon Web Services, Google Maps Web Services API, etc.

3.3.2. Mashup

Un *mashup* consiste en una aplicación web que combina elementos de más de una fuente para crear un servicio completo. Este concepto forma uno de los pilares de la ya mencionada Web 2.0 y se sustenta en el uso de diferentes protocolos y servicios web para integrar un conjunto de aplicaciones en una única.

Se compone de tres partes fundamentales, el proveedor de contenidos, el sitio *mashup* en cuestión y el navegador de usuario por el que se accede a la aplicación.

Hay que hacer hincapié en que un *mashup* no es un contenedor de datos de otros sitios, sino que debe darle un valor añadido a los datos que obtiene de los proveedores. Por ejemplo, una aplicación que tan solo muestre las publicaciones de diferentes *blogs* recogidas mediante RSS no sería un *mashup*, en cambio si la aplicación permite clasificar las publicaciones por categorías, valorarlas, mostrar las más leídas entonces sí podría ser considerada un *mashup*.

La facilidad de utilización de los recursos de los proveedores de contenido a través de APIs o fuentes RSS ha permitido que proliferen los *mashups*, aportando usos y funcionalidades que los servicios originales no contemplaron. Un claro ejemplo es Google Maps (<http://maps.google.com>), uno de los proveedores de contenido más utilizados a la hora de realizar *mashups*, permitiendo crear aplicaciones tales como Tagzania (<http://www.tagzania.com>), una herramienta que permite crear y compartir

mapas geográficos, WebcamVue (<http://webcamvue.com>), que muestra webcams colocadas por todo el mundo, o Trendsmap (<http://trendsmap.com>), que combinando Twitter con Google Maps muestra los *trending topics* locales.

3.3.3. Integración de servicios

Cuando hablamos de servicios web y de *mashups* estamos hablando de integración, donde los servicios web son el vehículo para conseguir aplicaciones híbridas.

La finalidad de integrar servicios debe ser siempre obtener una mejor funcionalidad, sacarle mayor partido a las herramientas que se están integrando. Actualmente, con el boom de las redes sociales, ha surgido una tendencia a integrarlas dentro de las aplicaciones web, pudiendo ver *timelines* de Twitter dentro de los *blogs* o aplicaciones como Tumblr (<http://www.tumblr.com>) que permiten actualizar tu *microblog* desde otros sistemas como WordPress (<http://wordpress.org>), YouTube, Digg.

Existen diferentes niveles de integración:

- Nivel de presentación. Consiste en integrar las interfaces de las aplicaciones, de tal forma que el usuario las percibirá como una única aplicación.
- Nivel de lógica de negocio. Se trata de integrar las aplicaciones a través de sus APIs o de servicios web, accediendo así a la mayoría de la lógica de negocio de una aplicación.
- Nivel de datos. Las aplicaciones se conectan a una única base de datos.

La integración en el nivel de datos genera muchas dependencias entre las aplicaciones, y pone en peligro la consistencia de los datos a menos que ambas aplicaciones sigan rigurosamente las reglas de negocio. Por tanto suele ser más común la integración en los niveles de presentación y lógica de negocio, siendo habitual una combinación de ambas.

4. Técnicas y herramientas

A lo largo de este apartado se presentarán las técnicas metodológicas y las diferentes herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto, explicando los motivos por los que se ha seleccionado una herramienta y no otra.

Se ha dividido en cinco subapartados que abarcan desde la elección de las aplicaciones en las que se apoya el portal 2.0, pasando por la metodología y el entorno de desarrollo, para concluir con la elección de la tecnología utilizada para integrar los diferentes sistemas.

4.1. Componentes del sistema

El sistema se sustentará sobre estos un CMS, un LMS y un gestor de *blogs* por lo que es muy importante seleccionarlas adecuadamente.

Para la elección final se han tenido en cuenta varios factores comunes. En primer lugar se tenía claro que la experiencia de usuario final era un punto muy importante ya que, tanto durante el desarrollo como una vez en explotación, es importante sentirse cómodo usando las diferentes herramientas. En segundo lugar todas las herramientas deben ser *open source* ya que se va a tocar partes del código para realizar una mejor integración.

4.1.1. Drupal

Se sigue la política de uso de *software* libre que se promovió en la Universidad de Salamanca a través de la Oficina de Conocimiento Abierto y el Vicerrectorado de Innovación Tecnológica (2007-2009). En este sentido se ha continuado la línea que el director del grupo y tutor de este proyecto, el Dr. Francisco J. García Peñalvo, trazó para el portal institucional de la USAL en su etapa como Vicerrector de Innovación Tecnológica, el CMS elegido ha sido Drupal.

Se trata de una herramienta muy flexible, desarrollada en PHP, que permite extender su funcionalidad mediante módulos desarrollados por una activa comunidad de usuarios.

4.1.2. Moodle

La elección del LMS tampoco ha supuesto ningún problema ya que, tanto desde el grupo, como por mi parte, se confía en Moodle para desarrollar los cursos que se imparten dentro de GRIAL. La gran experiencia de los docentes del grupo de investigación en el uso de dicha plataforma, la escalabilidad que ha demostrado tener como campus de la Universidad, Studium, y el gran abanico de posibilidad que ofrece para realizar futuros desarrollos de interés para el grupo, han sido algunos de los motivos por los que se ha elegido.

La elección de esta herramienta desde dentro de GRIAL se basa también en las experiencias previas de los miembros del grupo, que a lo largo de los años han trabajado con diferentes LMS disponibles en el mercado, Dokeos, Blackboard, E-ducative, BSCW, entre otros.

4.1.3. WordPress

La elección del gestor de *blogs* se ha basado en la grata experiencia adquirida en la implantación del gestor de *blogs* de la USAL, Diarium (<http://diarium.usal.es>). En aquel momento se contempló diferentes formas de crear una estructura estable para albergar un gran número de *blogs* y la mejor solución fue WPMU (<http://mu.wordpress.org>),

WordPress MultiUser, ya que dicha herramienta surgió a partir de la experiencia que los desarrolladores de WordPress tuvieron con su propio gestor de *blogs*, WordPress.com, además de la existencia de una gran comunidad de desarrolladores que dan soporte a la misma. Entre los usuarios que han confiado en WordPress se pueden encontrar los nombres de grandes universidades, empresas y medios de comunicación tales como *The New York Times* (<http://www.nytimes.com/interactive/blogs/directory.html>), *The University of British Columbia* (<http://blogs.ubc.ca>), AMD (<http://blogs.amd.com>), *The Times* (<http://blogs.timeslive.co.za>), *University of Berlin* (<http://blogs.fu-berlin.de>), y otras muchas que pueden verse en la Figura 9.

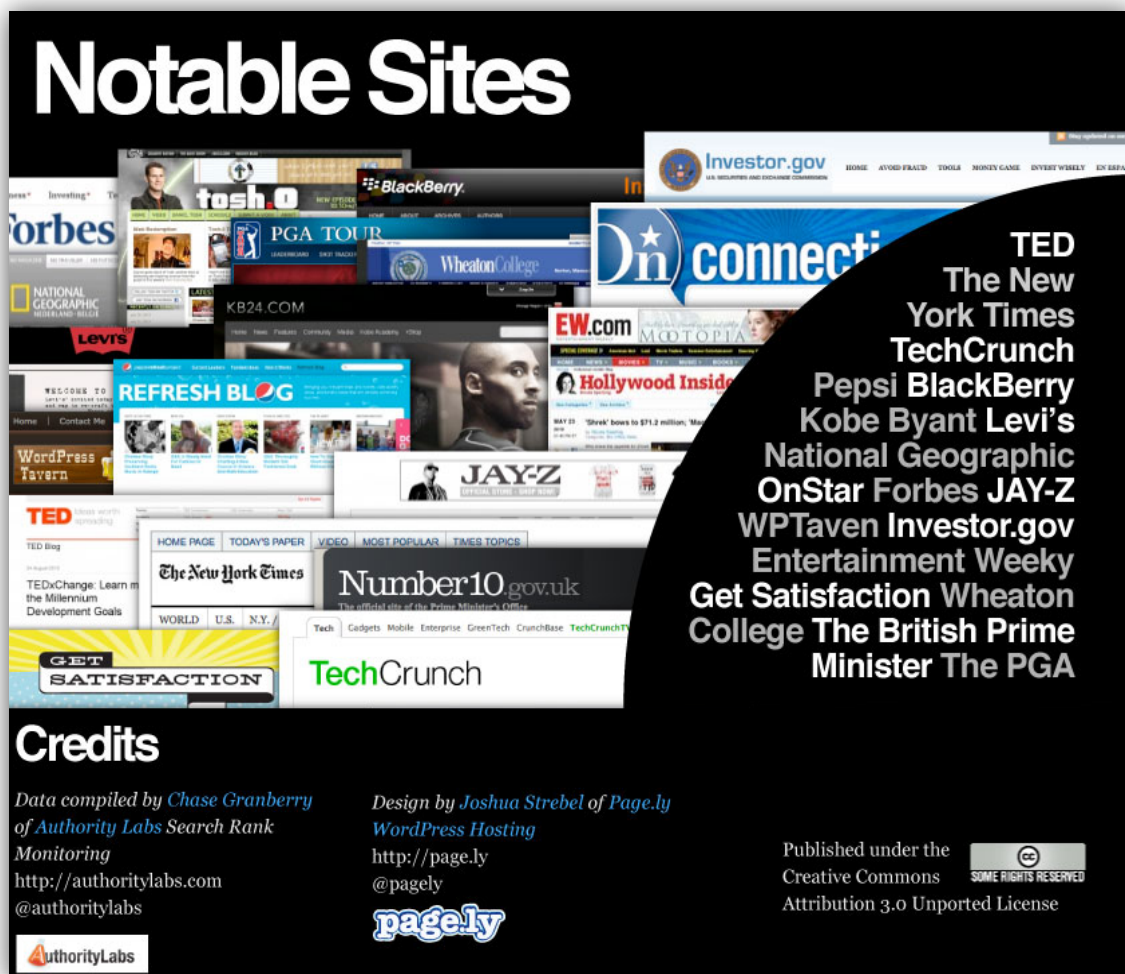


Figura 9. Sitios que utilizan WordPress

Es cierto que Drupal y Moodle integran herramientas de *blogs*, pero estas no satisfacen las necesidades existentes, ya que la idea es crear una comunidad de usuarios, sean miembros o no de GRIAL, que compartan su conocimiento a través de una plataforma. Mantener un sistema independiente para los *blogs* permite separar la capa social, de tal forma que si un día se desea cambiar el CMS o el LMS no sea necesario realizar una migración de todos los *blogs* existentes en el sistema.

WordPress también ofrece la posibilidad de crear *microsite* que alberguen proyectos del grupo de investigación, de esta forma la web del proyecto se gestionará a través del gestor de *blogs* pero mantendrá la apariencia de una web independiente tal y como se muestra en la Figura 10.



Figura 10. Microsite en WordPress

4.1.4. DSpace

La elección de DSpace como repositorio de datos viene impuesta por el grupo de investigación porque en su momento, cuando se puso en explotación Gredos (<http://gredos.usal.es>), el repositorio institucional de la USAL, se barajaron diversas opciones y fue la que mejores características ofrecía. (Ver Figura 11).



Figura 11. Gredos. Repositorio de la USAL

DSpace es una herramienta *open source* que permite administrar colecciones digitales de datos. Soporta una gran variedad de tipos de datos tales como libros, fotografías, tesis, informes técnicos, revistas, etc.

Casi todos los elementos de DSpace disponen de sindicación, de tal manera que cualquier usuario puede suscribirse a la comunidad o a la colección de contenidos centrada en un tema de su interés, de esta forma, cada vez que se añade un nuevo elemento a dicha colección o comunidad el usuario tendrá conocimiento de ello. (Ver Figura 12). Es de especial interés para los miembros del grupo de investigación el canal de sindicación general, que les permitirá conocer los contenidos aportados por los otros miembros.



Figura 12. RSS de la colección "Conferencias" del repositorio de GRIAL

4.1.5. Multimedia

Un punto a tener en cuenta a la hora de promover una web es el uso de las redes sociales. Para ello se ofrece a los visitantes facilidades para que puedan compartir un determinado recurso de nuestro sistema en diferentes sitios como Facebook, Twitter, Delicious, etc.

También se hará uso de una serie de herramientas web para permitir la difusión de la información en diferentes formatos tales como vídeos, fotos, enlaces, etc. De esta forma GRIAL tendrá perfiles dentro de herramientas para publicación de vídeos, presentación, enlaces, fotografías, creando una pequeña nube de contenidos dentro de Internet.

También se hará uso de una serie de herramientas web para permitir la difusión de la información en diferentes formatos tales como vídeos, fotos, enlaces, etc. De esta forma GRIAL tendrá perfiles dentro de herramientas para publicación de vídeos, presentación, enlaces, fotografías, creando una pequeña nube de contenidos dentro de Internet (Ver Figura 13).

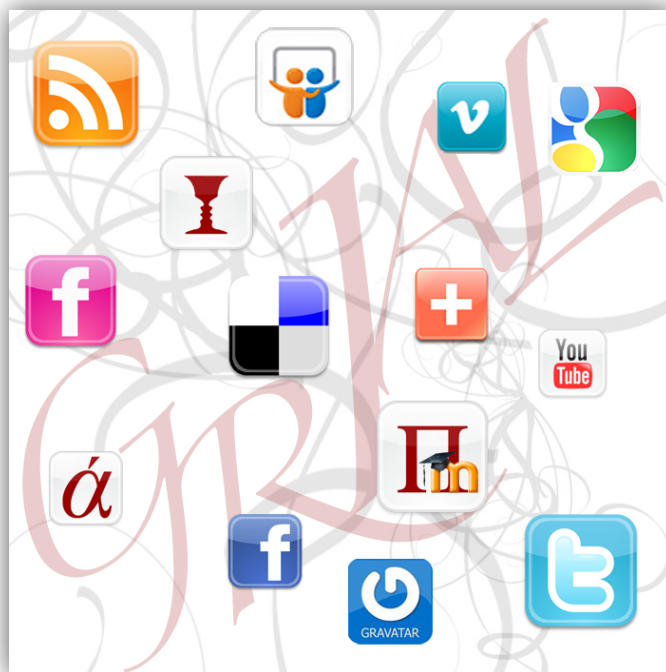


Figura 13. Nube de servicios de GRIAL

Las aplicaciones o servicios web seleccionados son los siguientes:

- Facebook. <http://facebook.com/grialusal> (Figura 14). Se ha elegido esta red social mundialmente conocida para introducir a GRIAL en un entorno puramente social aunque en el futuro se pretende dar presencia a GRIAL en una red más especializada como LinkedIn (<http://www.linkedin.com>). Facebook permite, además de los perfiles de usuario, crear páginas para empresas, servicios, entidades, etc. Para un grupo de investigación no tiene sentido mantener un perfil personal y hacer amigos, sino conseguir seguidores, fans, que les guste lo que el grupo ofrece.



Figura 14. Facebook de GRIAL

- Twitter. http://twitter.com/grial_usal. Servicio de *microblogging* que permite publicar entradas de un máximo de 140 caracteres denominados *tweets*. Actualmente es uno de los servicios de Internet más utilizados, pudiendo ser conectado con un gran número de aplicaciones.



Figura 15. Twitter de GRIAL

- Flickr. <http://flickr.com/grialusal> (Figura 16). Es un servicio que te permite subir, organizar y compartir fotos y vídeos en línea así como insertarlos en cualquier sitio web. Permite proteger tus contenidos asociándoles una licencia Creative Commons, algo fundamental para mantener la política del grupo de investigación de generar contenidos bajo dichas licencias. Existen otras opciones para compartir y almacenar imágenes pero se ha optado por utilizar Flickr por ser la que permitía mayor integración con WordPress, Facebook y Twitter.

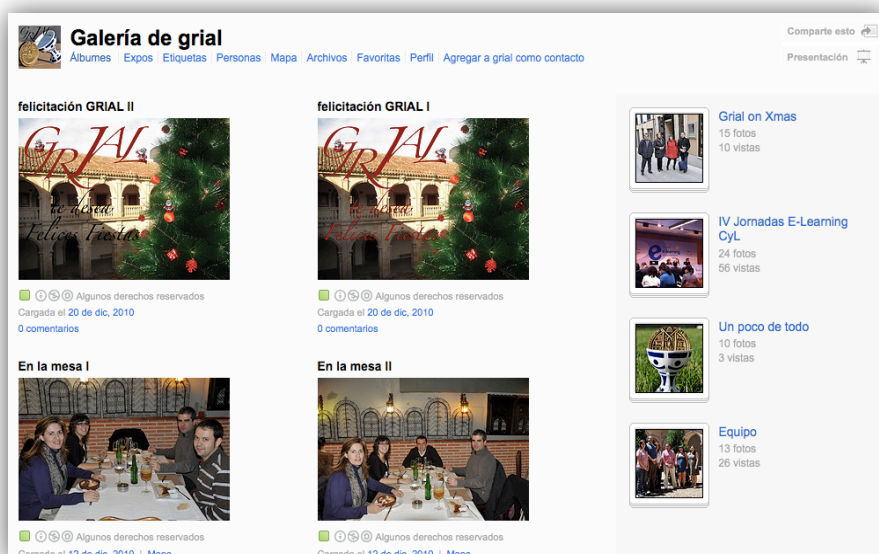


Figura 16. Flickr de GRIAL

- Vimeo y YouTube. <http://vimeo.com/grial> (Figura 17) y <http://youtube.com/grialusal> (Figura 18). A pesar de que Flickr permita compartir vídeos, dicha aplicación está más orientada a la fotografía por lo que se ha considerado adecuado mantener los vídeos de GRIAL en un servicio diseñado explícitamente para ese tipo de contenido. Se han elegido dos aplicaciones, Vimeo, que será el contenedor principal de vídeos por tener un aspecto mucho más elegante y por ser un servicio conocido por la calidad de los vídeos de sus usuarios. También se mantiene un perfil en YouTube para poder mantener un registro de todos aquellos vídeos visualizados relacionados con las líneas de trabajo del grupo de investigación.

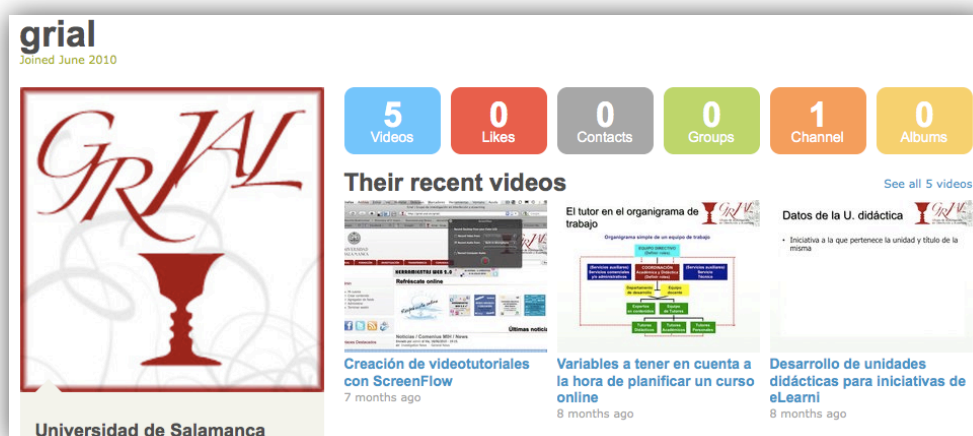


Figura 17. Vimeo de GRIAL



Figura 18. YouTube de GRIAL

- SlideShare. <http://slideshare.net/grialusal> (Figura 19). Se trata de un servicio *online* para compartir documentos e insertarlos en cualquier sitio web. Permite la carga de documentos de hasta 100MB en diferentes formatos, destacando las presentaciones y los PDFs. Se pueden configurar diferentes aspectos como la licencia del contenido o si se permite descargar el fichero fuente.

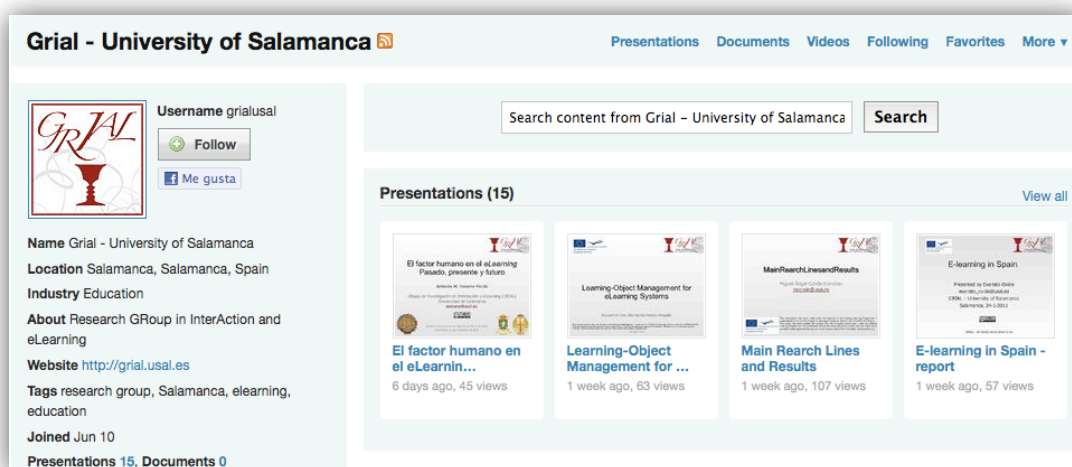


Figura 19. SlideShare de GRIAL

- Delicious. <http://delicious.com/grialusal> (Figura 20). Servicio de gestión de marcadores sociales que permite organizarlos mediante etiquetas. A pesar de los rumores de cierre, actualmente sigue funcionando y en caso de baja del servicio la migración será sencilla ya que muchas otras herramientas permiten importar los enlaces de Delicious. Este servicio incorpora RSS en todas sus páginas, facilitando su utilización como fuente para los enlaces recomendados del portal de GRIAL.

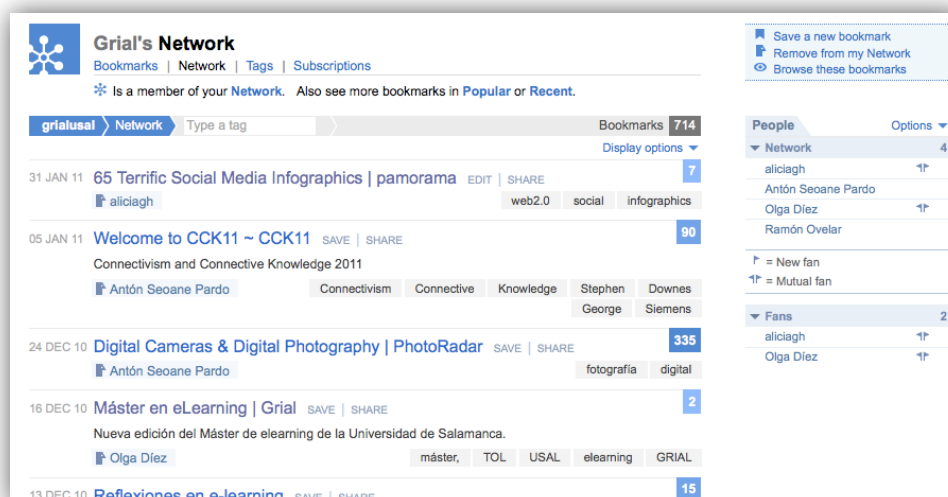


Figura 20. Delicious de GRIAL

- Gravatar. <http://gravatar.com/grialusal> (Figura 21). Servicio que permite mantener un avatar, imagen de perfil, asociada a uno o más cuentas de correo electrónico de tal manera que cuando uses una de esas cuentas en alguna de las aplicaciones que soportan Gravatar tu imagen se mostrará de forma automática. Se trata de una herramienta muy útil a la hora de mantener tu imagen en la Red ya que tan solo es necesario cambiar el avatar en un sitio para que se cambie en la mayoría de las aplicaciones que utilizas. Tanto WordPress como Drupal tienen soporte para Gravatar y se pretende desarrollar un módulo para integrar este servicio en Moodle.

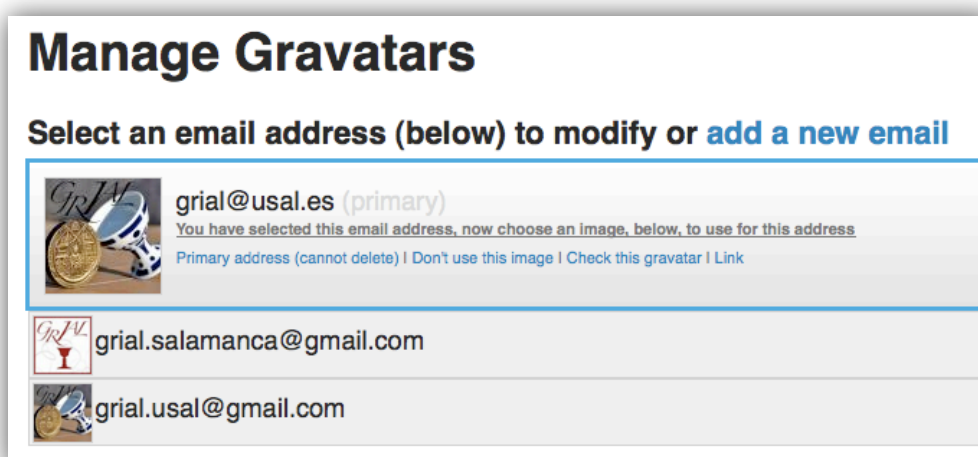


Figura 21. Gravatar de GRIAL

- WordPress.com (Figura 22). Existen una serie de *plugins* para WordPress que necesitan de una clave API para poder ser utilizados dentro de tu propia instalación. Para conseguir dicha clave es necesario tener una cuenta gratuita dentro de WordPress.com. Akismet, *plugin* para controlar el *spam* dentro de los *blogs*, y WordPress.com *stats*, el *plugin* por excelencia de estadísticas, ambos utilizados en GRIAL, requieren esta clave.

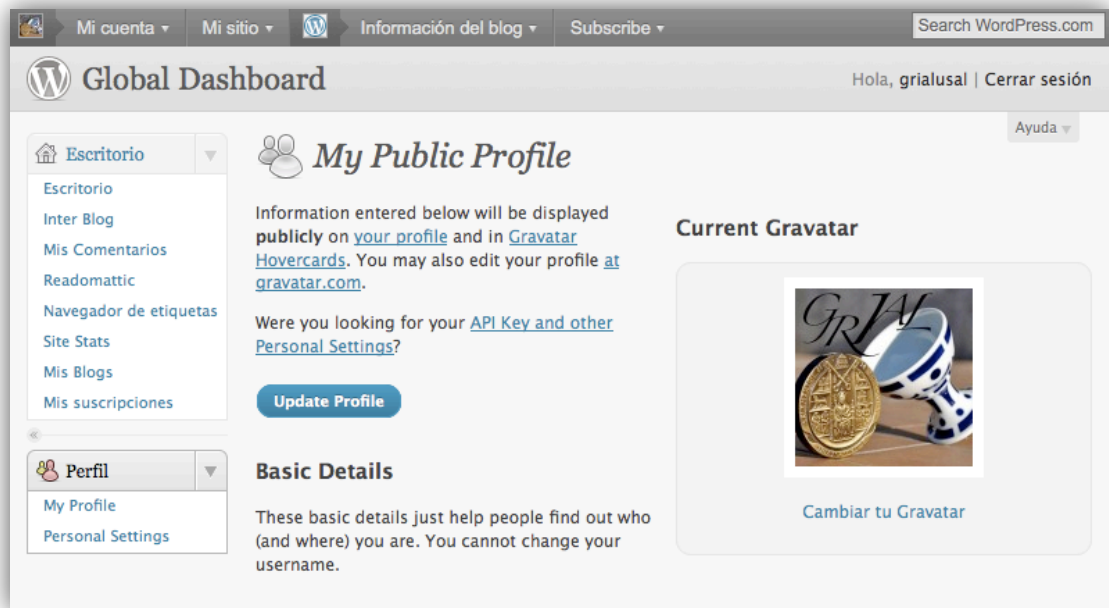


Figura 22. WordPress.com de GRIAL

- AddThis. Se trata de un servicio que te permite gestionar *bookmarks* y compartir contenidos en más de 300 servicios como Delicious, Tumblr, Digg, FriendFeed o Twitter. (Ver Figura 24).

No es necesario registrarse para poder utilizarlo, pero el registro te permite acceder a estadísticas acerca de qué contenidos de tu sistema se están compartiendo, donde se están publicando, etc. como las que se muestran en la Figura 23.

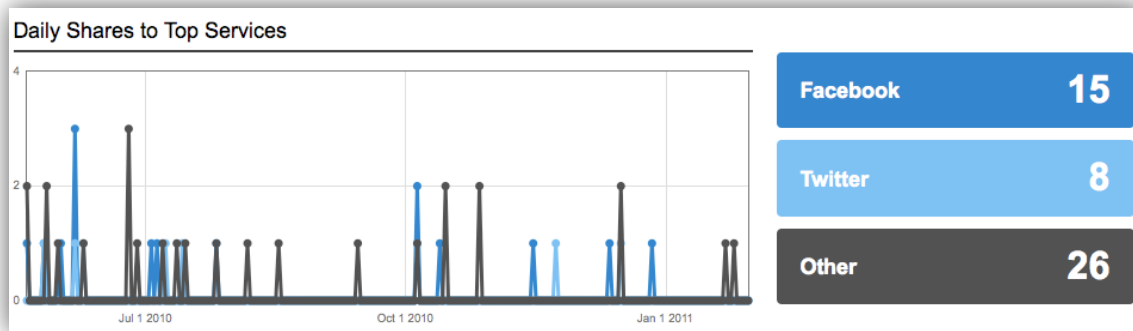


Figura 23. Estadísticas de AddThis



Figura 24. Botón de AddThis utilizado en GRIAL

- Bit.ly. (<http://bit.ly/u/grial>) Figura 25. Servicio para acortar URLs. Cualquier URL acortada tiene su propia página con todas las métricas posibles asociadas a ella: *clicks* en el tiempo, *referrers*, países, *tweets* en los que aparece referenciada, etc. Además, existen diferentes herramientas para utilizar toda esa información y una API para poder utilizar el servicio desde tu propia aplicación. En GRIAL se utiliza como complemento a Twitter para aprovechar al máximo los 140 caracteres que permite un *tweet*.

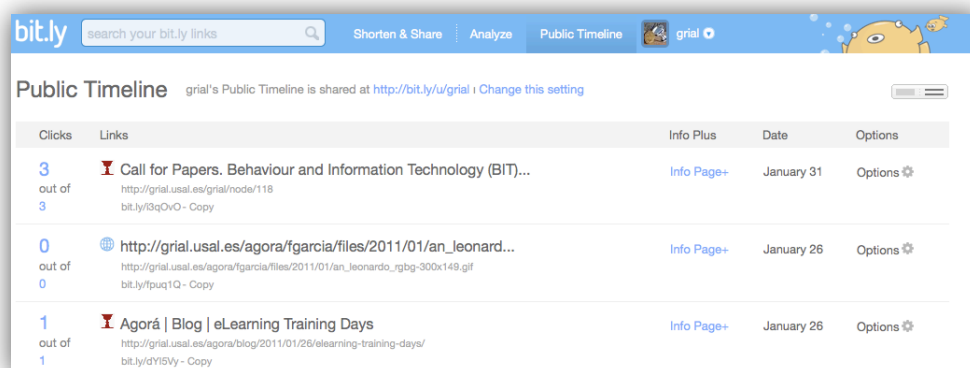


Figura 25. Bit.ly de GRIAL

4.2. Nuevas versiones en las herramientas utilizadas

Durante el desarrollo del proyecto las herramientas que forman los pilares del portal han lanzado nuevas versiones con importantes cambios. A lo largo del verano y el otoño de 2010 han aparecido Drupal 7, WordPress 3.0 y Moodle 2.0, marcando todas ellas un antes y un después dentro de sus líneas de desarrollo. Drupal 7 ha incorporado características de orientación a objetos pertenecientes a PHP5, WordPress 3.0 ha integrado la funcionalidad de WordPress MU y Moodle 2.0 ha sido prácticamente construido desde cero incorporando una capa de servicios web.

A pesar de todo el trabajo desarrollado hasta la fecha en la que han aparecido cada una de las versiones, fue necesario plantearse si merecía la pena migrar el sistema a la nueva versión o mantener la elegida en un primer momento.

4.2.1. WordPress MU 2.9.2 vs. WordPress 3.0

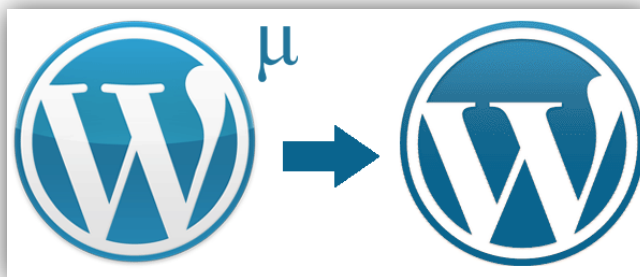


Figura 26. WordPress MU a WordPress 3.0

El gestor de *blogs* ha sido una de las primeras herramientas instaladas y configuradas para funcionar en GRIAL 2.0. En un primer momento la instalación se realizó con un WordPress MU, por lo que cuando se lanzó WordPress 3.0 todo el sistema estaba ya en producción.

Para tomar una decisión sobre la migración de WPMU a WordPress 3.0 con soporte *multisite*, Figura 26, se ha realizado un trabajo de campo buscando casos de éxito en la migración de un sistema a otro así como opiniones de los usuarios desde dos puntos de vista, como desarrolladores y como usuarios finales. De esta fase de investigación se ha concluido que el paso de un sistema a otro había estado contemplado por los desarrolladores de WordPress así que la transición sería lo más transparente posible. Además WordPress dispone de una comunidad de usuarios mayor y existe mucha más documentación que para WPMU, a esto hay que sumarle que WPMU queda sin soporte por lo que tarde o temprano habrá que migrar el sistema si no se lleva a cabo inicialmente.

El siguiente paso ha sido realizar una serie de comprobaciones y pruebas para asegurarse que todos los *plugins* seleccionados para que formen parte del *core* del gestor de *blogs* de GRIAL existen en la nueva versión de WordPress. Para ello se ha instalado un WordPress 3.0 en el servidor de preproducción y se han instalado y configurado uno a uno los diferentes *plugins*.

Todos los *plugins* han superado las pruebas de funcionamiento, incluido BuddyPress por lo que se hará la migración. Esto es debido, en gran medida, a que los *plugins* seleccionados para el gestor de *blogs* se encuentran entre los más utilizados y valores

por los usuarios de WordPress lo que siempre impulsa que su actualización para soportar nuevas versiones sea rápida.

Respecto a la experiencia de usuario, tanto administrador como propietario de un *blog*, han realizado una serie de cambios que la mejoran en comparación a WPMU. La parte de administración es algo más sencilla y han introducido funciones nuevas como soporte para menús personalizados.

Una vez tomada la decisión la migración se ha realizado instalando desde cero un nuevo WordPress 3.0 y los pocos *blogs* que existían en la instalación de WPMU se han importado dentro del nuevo. De esta manera se ha evitado posibles problemas de migración entre sistemas.

4.2.2. Moodle 1.9 vs. Moodle 2.0

La versión 2.0 de Moodle se suponía que iba a estar disponible para producción en julio de 2010 pero en el momento de comenzar la fase de Moodle, septiembre de 2010, la versión estable 2.0 todavía no se encontraba disponible y utilizar una versión inestable para montar un sitio en producción es poco recomendable. Retrasar esta fase hasta poder utilizar Moodle 2.0 no ha sido considerado como una opción ya que, a pesar de que se considerará una versión estable, durante los primeros meses en producción hay muchas más posibilidades de encontrar *bugs* que interfirieran con la integración que vamos a llevar a cabo entre Drupal y Moodle.

Esto no quiere decir que Moodle 2.0 se vaya a dejar de lado, todo lo contrario, se pretende mantener una instalación en el servidor de preproducción para hacer pruebas y llevar a cabo posibles desarrollos.

Se mantendrá la versión 1.9 que se instaló en un primer momento, actualizándola a la 1.9.9 para que incorpore todas las mejoras posibles antes de que dejen de dar soporte a esta versión y pasen a hacerlo de la 2.0.

4.2.3. Drupal 6.20 vs. Drupal 7

La versión 7 de Drupal ha sido lanzada de manera oficial a principios de enero de 2011 por lo que realizar una migración del sistema es inviable. Tal y como se ha configurado GRIAL 2.0, el salto de versión no será una tarea complicada, pero requiere la adaptación de todos los módulos de Drupal desarrollados durante este proceso, así como esperar a que los módulos de terceros instalados tengan versiones compatibles con la nueva *release*.

La necesidad de adaptar los módulos a la nueva versión es un proceso común dentro de los desarrollos de Drupal, existiendo guías que te permiten hacer el salto de versión de tus módulos de la forma más transparente posible.

4.3. Desarrollo

4.3.1. Metodología

La parte central del proyecto gira en torno a Drupal por lo que a la hora de elegir una metodología se ha observado cómo se iba a tener que trabajar con el CMS.

Si únicamente se considera que un desarrollo es orientado a objetos por estar construido mediante la definición de clases y la interacción entre objetos entonces Drupal 6 no es orientado a objetos ya que no utiliza las características que PHP proporciona para ese

tipo de desarrollos. Aun así, si se analiza la estructura y el diseño de Drupal se pueden encontrar en él muchos de los principios del paradigma orientado a objetos:

- Objetos.
- Abstracción.
- Encapsulación.
- Polimorfismo.
- Herencia.
- Patrones.

Las razones por las que el diseño actual de este CMS no hace uso de clases aparecen redactadas en la documentación oficial (<http://drupal.org/node/547518>):

“First, PHP's support for object-oriented constructs was much less mature at the time of Drupal's design. Drupal was built on PHP 4, and most of the improvements in PHP 5 relate to its object-oriented features.

Second, Drupal code is highly compartmentalized into modules, each of which defines its own set of functions. The inclusion of files is handled inside functions as well; PHP's performance suffers if unneeded code is included, so Drupal attempts to load as little code as possible per request. This is a critical consideration, especially in the absence of a PHP accelerator: the act of compiling the code accounts for more than half of the time to complete a Drupal page request. Functions are therefore defined inside other functions in Drupal, with respect to the runtime scope. While this is perfectly legal for bare functions, PHP does not allow the same kind of nesting with class declarations. This means that the inclusion of files defining classes must be "top-level," and not inside any function, which leads either to slower code (always including the files defining classes) or a large amount of logic in the main index.php file.

Finally, Drupal attempts to adopt some OOP design aspects that PHP's OOP system does not yet support, or did not support at the time of adoption.”

Basándose en los mismos argumentos que dan los desarrolladores de Drupal se ha decidido que la elección de una metodología que no abarcara el paradigma orientado a objetos no tiene sentido, por ello se ha elegido el Proceso Unificado (Jacobson et al., 1999) y se ha combinado con OOWS (*Object Oriented Web Solution Modeling*) (Pastor et al., 2001) para cubrir el modelado conceptual desde un punto de vista orientado a las aplicaciones web.

Una de los objetivos principales del OOWS es ampliar los conceptos de modelado orientado a objetos para introducir una semántica de navegación imprescindible en contextos web. Para ello se introduce una nueva etapa en el modelo conceptual en la cual se definen los mapas de navegación de cada agente.

De la metodología OOWS se ha tomado el modelo navegacional, que utiliza UML igual que el Proceso Unificado, así como toda la semántica de navegación.

4.3.1.1. Herramientas CASE

4.3.1.1.1. Gestor de proyectos

Para la planificación y gestión del proyecto existen diversas opciones. Se han descartado las aplicaciones de escritorio y se ha optado por instalar una herramienta *online* para que, además de servir para gestionar este proyecto, quede a disposición del

grupo de investigación para construir una completa y accesible base de datos de conocimientos, compartición de recursos y documentos, gestión de equipos de trabajo en los diversos proyectos de investigación en los que participa el grupo a lo largo del año, tareas, calendarios y otras muchas funciones útiles para la buena gestión de un proyecto.

La herramienta elegida debe ser fácil de manejar y con una curva rápida de aprendizaje ya que será utilizada por todos los miembros de GRIAL implicados en proyectos, así como por evaluadores externos y *partners*. Se debe tratar de una herramienta de código abierto que se encuentre en una versión actual, es decir, que no se trate de un proyecto abandonado ya que ante cualquier problema será mucho más complicado encontrar su correspondiente solución. Tampoco debe tener restricciones de número de proyectos o usuarios, como ocurre en algunas aplicaciones *online* en su versión gratuita, por ejemplo, Doolphy.com.

Anteriormente se ha trabajado con Egroupware, una herramienta muy completa que quedó en desuso por su complejidad. Disponía de gran cantidad de funciones que no eran necesarias para el perfil de gestión que se realiza dentro del grupo y que lo único que aportaban era dificultad a la hora de realizar tareas sencillas.

Para elegir una buena herramienta de gestión de proyecto, además de buscar información sobre las diferentes opciones, es muy importante realizar un periodo de prueba con aquellas que cumplan los requisitos mencionados anteriormente. Tras utilizar las demos de DotProject y Redmine se optó por la segunda.

Redmine es una aplicación web destinada a soportar la gestión de proyectos software con posibilidad de personalización para adaptarla a proyectos de otra índole. Se trata de un proyecto vivo, realizado en Ruby on Rails bajo los términos de la licencia “*GNU General Public License v2 (GPL)*”. Posee una interfaz clara e intuitiva que permite que el usuario comience casi de inmediato a realizar la gestión del proyecto sin tener que dedicar demasiado tiempo a conocer el manejo de la misma.

Entre sus características principales cabe destacar el control de versiones, *time tracking*, seguimiento de *bugs*, integración con gestores de versiones como Subversion o Git, diagramas de Gantt, exportación a PDF, creación de documentos colaborativos, tareas, calendario, personalización de los diferentes perfiles que un usuario pueda poseer dentro de un proyecto así como la configuración de los permisos disponibles en cada perfil.

4.3.1.1.2. Herramientas de modelado

Para la realización del conjunto de diagramas UML que conforman la documentación técnica del proyecto se ha utilizado Visual Paradigm for UML, una herramienta CASE multiplataforma de pago que dispone de una versión gratuita, Community Edition, para su utilización en proyectos no comerciales o de índole académica.

Visual Paradigm for UML se trata de un producto de calidad, con un gran número de versiones a sus espaldas, se ha trabajado con la versión 8.1, lo cual influye en la calidad de los diagramas que se obtienen con ella. Su soporte para aplicaciones web, algo de lo que no todas las herramientas CASE disponen, la hace totalmente aplicable al proyecto que se ha elaborado sin necesidad de tener que forzar su adaptación.

Entre sus muchas características cabe destacar la facilidad para mantener una relación entre los diferentes tipos de diagramas, pudiendo así realizar un seguimiento del sistema a través de las diferentes fases del ciclo de vida del desarrollo. Todos los diagramas

pueden ser exportados a diferentes formatos, entre ellos imágenes, que en el caso de la versión gratuita aparecerán con una marca de agua. A pesar de haberse utilizado dicha versión, los diagramas mostrados en la documentación técnica no muestran la marca de agua porque han sido obtenidos mediante capturas de pantalla.

Además de las características mencionadas, hay dos motivos principales por los que se ha elegido esta herramienta. En primer lugar la recomendación del tutor como una de las herramientas más completas con soporte para una de las versiones más actuales de UML, 2.1. En segundo lugar se contaba con experiencia previa en el uso de la misma por lo que la curva de aprendizaje ha sido inexistente.

4.3.1.1.3. Gestor de versiones

El uso de un gestor de versiones se vuelve imprescindible para mantener un registro de los diferentes cambios que sufren los desarrollos software, permitiendo deshacer cambios y mantener diferentes versiones según se va avanzando el proceso de implementación.

Entre los gestores de versiones más conocidos podemos ver Git, Subversion y CVS.

Como el desarrollo se centra en extensiones para los diferentes gestores de contenido utilizados, se ha optado por no instalar un gestor de versiones propio y utilizar los que cada uno de los sistemas proporciona en su repositorio oficial.

Drupal utilizar CVS para mantener un control de versiones sobre los módulos aportados por los usuarios del sistema. WordPress ha optado por Subversion como repositorio para toda la batería de *plugins* disponibles. Tan sólo Moodle no proporciona un gestor de versiones para los módulos desarrollados por la comunidad por lo cual se ha instalado Git en modo local para mantener un control sobre los desarrollos relativos al LMS.

4.3.1.2. *Framework*

Es habitual plantearse la utilización de un *framework* que facilite las tareas de desarrollo. Existen varios *frameworks* interesantes para el desarrollo de aplicaciones en PHP, Symfony, CakePHP, Zend, todos ellos proporcionan una arquitectura, un conjunto de componentes y de herramientas para el desarrollo que permiten construir aplicaciones web más rápidamente.

En el caso que nos ocupa la funcionalidad que se necesita implementar deberá ser complementaria a las herramientas anteriormente mencionadas, por lo tanto la utilización de un *framework* externo complicaría el desarrollo innecesariamente, y es más correcto seguir el mismo estilo y esquema de programación utilizado en cada una de las herramientas para las que se realiza el desarrollo. Estos gestores de contenidos proporcionan un marco donde desarrollar nuevas funcionalidades utilizando toda una batería de librerías que permiten obtener pequeñas aplicaciones que se añaden al sistema de forma transparente. Desarrollar un módulo para Drupal o añadir un nuevo *plugin* a WordPress requiere conocer un conjunto de herramientas que cada uno de los gestores proporciona y utilizarlas correctamente para que el sistema reconozca la nueva funcionalidad. Visto de esta manera un gestor de contenido es un *framework* ya que proporciona una base para que otros programadores web extiendan su funcionalidad siguiendo una manera estandarizada de trabajar.

Utilizar un CMS a la hora de realizar una aplicación web, siempre y cuando se adapte a tus necesidades, tiene muchas de las ventajas que un *framework* como tal proporcionaría. Algunas de ellas son escalabilidad y mantenimiento, mayor rapidez a la

hora de desarrollar aplicaciones de gran envergadura, extensa documentación con una gran comunidad de usuarios que comparten tus mismos problemas.

4.3.1.3. PHP

El lenguaje de programación utilizado para llevar a cabo todos los desarrollos necesarios es PHP. Esto se debe a que tanto Drupal como Moodle y WordPress están desarrolladas en este lenguaje.

Es cierto que algunas de las herramientas utilizadas, como DSpace o Redmine, están desarrolladas en otros lenguajes, JSP, Ruby on Rails, pero el trabajo con estas herramientas se centrará en su correcta puesta en marcha y configuración, y en caso de ser necesario en la adaptación visual a la imagen corporativa de GRIAL, teniendo únicamente que tratar con HTML y CSS.

Además, PHP posee una serie de ventajas que hacen que sea uno de los lenguajes de programación web más utilizados:

- Se trata de un lenguaje sencillo de aprender que soporta orientación a objetos.
- Reusabilidad en distintas plataformas.
- Gran cantidad de extensiones que ofrecen una amplia gama de funcionalidades.
- Posee una amplia documentación en su página oficial, destacando la explicación de todas las funciones que forman parte de su núcleo así como de sus extensiones, incluyendo ejemplos y notas enviadas por usuarios.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, Oracle, PostgreSQL, y otros muchos.
- Se puede ejecutar bajo diferentes servidores pudiendo ser configurado como módulo CGI.
- Lenguaje *open source*.

Aunque también presenta algunas desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- Dificulta la organización por capas de la aplicación si no se utiliza un *framework* adecuado.

4.3.2. El entorno de documentación

4.3.2.1. Entorno *online* – el *blog* del proyecto

Poco después de comenzar la realización del proyecto se tomó la decisión de mantener un *blog* para comentar problemas, decisiones, desarrollos, etc. Dicho *blog* se encuentra desde un primer momento en el gestor de *blogs* de GRIAL, primero en un WPMU y después en WordPress 3.0. La dirección del *blog* es <http://grial.usal.es/agora/pfcgrial>.

Una de las partes más interesantes del *blog* es su función como página “oficial” para los *plugins* desarrollados para WordPress. Cuando un nuevo *plugin* es agregado al repositorio de WordPress es recomendable indicar una página web donde se da soporte al *plugin* y se aporta información adicional a la que aparece en la WordPress.org. Cada

una de las páginas dedicadas a *plugins*, MS Global Search, WPMU Global Search y BP Activity Tags, contienen un conjunto de comentarios que representan parte del soporte que se ha ofrecido desde que el *plugin* se puso a disposición de la comunidad. Algunos de los comentarios han sido los causantes de desarrollar nuevas versiones que solucionaran *bugs* o que implementaran peticiones de los usuarios. (Ver Figura 27).

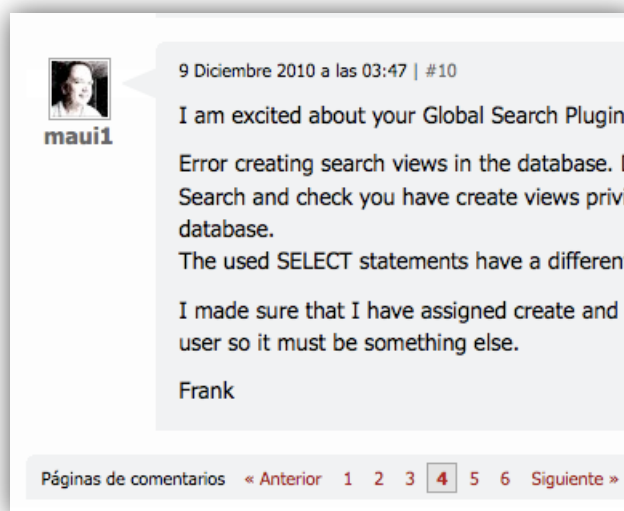


Figura 27. Comentarios de *Multisite Global Search*

Por último destacar la utilización del *blog* como medio para ofrecer soluciones, en forma de pequeños tutoriales en español, a problemas encontrados durante la instalación y configuración de algunas herramientas y para los cuáles no se encontraron soluciones válidas en Internet.

4.3.2.2. Entorno ofimático

La principal herramienta que se ha utilizado a lo largo de todo el desarrollo ha sido Google Docs. De esta manera el tutor ha podido realizar un seguimiento más cercano.

Desde el primer día que se inició la realización de este proyecto se han mantenido vivos varios documentos. Toda la parte de investigación se iba recogiendo en un documento, indicando enlaces, imágenes y todo tipo de información recopilada para tomar las diferentes decisiones.

Los aspectos relevantes, los desarrollos llevados a cabo y otros puntos relativos a la memoria del proyecto se han ido escribiendo, unas veces de forma esquemática otras redactando adecuadamente, a medida que se realizaban las fases de documentación y posterior desarrollo.

Informes de estado, borrador de requisitos o líneas futuras de desarrollo son otros de los documentos que se han elaborado.

Finalmente todos estos documentos se han utilizado para elaborar la presente memoria en Microsoft Word para Mac.

5. Descripción del sistema

GRIAL 2.0 está formado por tres sistemas principales. El primero es un portal de información cuya finalidad es poner a disposición de todo el mundo los conocimientos que se generan dentro del grupo de investigación. Toda la información relacionada con el grupo será accesible a través del portal, pudiendo consultar los cursos que se están impartiendo, los proyectos de investigación en los que se ha trabajado o las publicaciones realizadas por los diferentes miembros del grupo. (<http://grial.usal.es>).

El segundo pilar es una plataforma de *eLearning* (<http://grial.usal.es/polis>), utilizada para impartir todos los cursos que se realizan dentro de GRIAL así como el Máster en *eLearning*. Al tratarse de una herramienta en la que los usuarios pueden interactuar por medio de foros, mensajería, *chat*, se utiliza también como medio de comunicación entre los *partners* de los proyectos de investigación. Disponer de una plataforma propia permite trabajar en nuevos desarrollos mejorándola y adaptándola a las necesidades del momento. Dicha plataforma recibe el nombre de ΠΟΛΙΣ (“Pólis”, en griego antiguo, “ciudad”). La *pólis* para los griegos representa no sólo un lugar para vivir sino un espacio de convivencia e interacción, a la vez que se erige como el lugar genuino para la toma de decisiones que afectan a la vida *política* del ciudadano. Consideramos Pólis como el núcleo del ecosistema de formación de nuestro grupo, compuesto por un conjunto de aplicaciones, todas ellas relacionadas e incluso integradas con nuestra ciudad. Pólis, pues, es mucho más que un espacio de formación. Es toda una ciudad en la que habitan los estudiantes que se forman en las diferentes iniciativas del grupo y desde la que pueden acceder a todos los servicios y personas implicadas en los procesos de formación *online*.

En cuanto a la imagen del logotipo, se ha optado por mantener el nombre con la grafía original en griego y en letras mayúsculas (ΠΟΛΙΣ), porque su lectura no ofrece mayor dificultad incluso para quienes no lean griego clásico. El uso del alfabeto griego y de un nombre con referencias clásicas evoca el carácter interdisciplinar y no sólo tecnológico del grupo, conecta con un modelo y una metodología de formación que hunde sus raíces en la *paideía* de la Antigua Grecia (que es parte de sus señas de identidad) y refuerza el carácter científico mediante el uso de caracteres griegos que poseen gran simbolismo en diferentes ciencias tanto teóricas como prácticas y experimentales.

El tercer sistema principal es un gestor de *blogs* (<http://grial.usal.es/agora>), la parte más social del sistema, permite que cada miembro del grupo de investigación, así como todo aquel que lo solicite, disponga de un *blog* de manera totalmente gratuita y con el soporte de GRIAL. También sirve como herramienta para crear los *microsites* para los proyectos de investigación que lo requieran. Recibe el nombre de Ἀγορά (“Agorá”, en griego antiguo “Ágora” o “Plaza pública”). Como en el ágora de las ciudades griegas, el gestor de *blogs* es el lugar donde cada miembro del grupo y cada estudiante que se forma con GRIAL puede expresarse públicamente, hacerse escuchar y escuchar las opiniones de los otros ciudadanos. Por eso se ha concedido tanta importancia a la integración de estos servicios con las redes sociales, porque sólo así puede decirse que una ciudad es un espacio de “auténtica” interacción entre sus habitantes, y sólo así un *blog* pasa de ser un mero transmisor de las opiniones particulares de sus autores a convertirse en un punto de producción de espacios de interacción social, es decir, un “Agorá” que, como en el caso de las ciudades griegas, sirve a cualquier propósito natural en la vida civil, como en este caso sirve a cualquier propósito relacionado con la formación, la investigación y la transferencia de conocimiento.

En cuanto a la imagen del logotipo, se ha optado por mantener el nombre con la grafía original en griego por coherencia con Pólis, porque su lectura no ofrece ninguna dificultad incluso a quienes no leen griego clásico y porque manifiesta las señas de identidad del Grupo, como se ha expuesto anteriormente a hablar del campus virtual.

Para acceder a las anteriores aplicaciones los miembros de GRIAL disponen de un identificador único que permite que tras acceder a una de las aplicaciones no tengan que volver a ingresar su contraseña mientras el navegador permanezca abierto.

A este gran portal formado por las tres herramientas se unen otra serie de aplicaciones y servicios. GRIAL dispone de un repositorio de datos en el que ir construyendo una base de conocimiento sobre todas las líneas de trabajo del grupo de investigación (<http://grialdspace.usal.es:443>). La mayoría de los contenidos están bajo una licencia *Creative Commons*.

Como complemento al repositorio, GRIAL dispone de un disco virtual compartido accesible a través de cualquier ordenador que tenga conexión a Internet y siempre que el usuario disponga de los datos de acceso, necesarios para visualizar la parte privada del disco compartido y para poder escribir en él.

Para mantener un control sobre los proyectos de investigación vivos se dispone de un gestor de proyectos que permite realizar planificación temporal así como hacer las veces de repositorio de documentos relacionados con el proyecto y que no tienen cabida en el repositorio GRIAL por ser documentos temporales sin ningún valor más allá de gestionar el proyecto al que pertenezcan. (<http://grial4.usal.es/redmine>).

Por último pero no menos importante, GRIAL tiene presencia en un gran número de herramientas de la web 2.0. Así pueden encontrarse colecciones de fotografías de eventos relacionados con el grupo de investigación, presentaciones y vídeos con posibilidad de inserción en cualquier sitio web que lo desee, perfiles en redes sociales, selección de enlaces sobre educación, eLearning, web semántica y muchos otros temas relacionados con GRIAL.

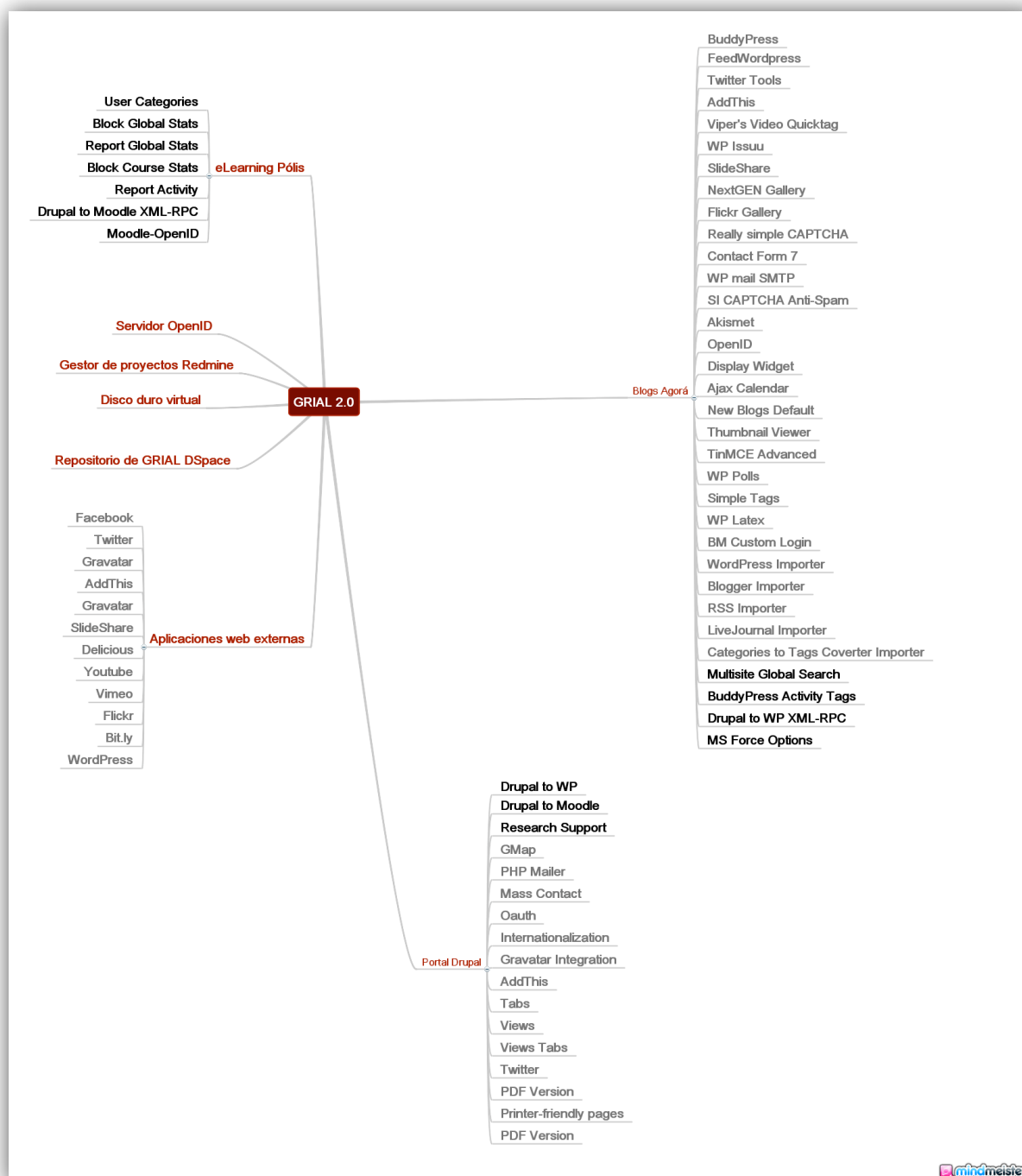


Figura 28. Componentes de GRIAL 2.0

6. Aspectos relevantes

Este apartado pretende recoger los aspectos más interesantes del desarrollo del sistema software centro del proyecto, comentados por los autores del mismo.

Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación.

Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales.

6.1. El ciclo de vida: iteraciones e incrementos

La evolución de los diferentes incrementos se muestra en diagramas de Gantt de tal forma que se han realizado capturas de los mismos en diferentes momentos del desarrollo para poder mostrar la evolución del proceso en el anexo de planificación temporal.

El proceso se ha dividido en cinco fases o incrementos. Cada incremento tiene sus propias fases de documentación y realización de diagramas, excepto el primer y último incremento que se trata del incremento asociado a Drupal dividido en dos partes como se explica más adelante. Sólo se han realizado los diagramas necesarios, acercándose un poco más a la filosofía de las metodologías ágiles. En cada incremento se han marcado una serie de hitos cuyo fin es indicar la finalización de un conjunto de tareas orientadas a alcanzar los objetivos definidos en el hito.

La primera fase abarca la documentación de la parte central del proyecto, es decir, el módulo desarrollado para Drupal que actúa como interfaz de publicación de los diferentes tipos de contenido dentro del portal. El objetivo de este incremento es sentar las bases del proyecto, realizar la obtención de los principales requisitos, así como el desarrollo de toda la documentación técnica asociada a Drupal. La mayor parte de la investigación y búsqueda de soluciones se ha llevado a cabo durante esta fase, lo que ha permitido que los productos resultantes sean la base para el desarrollo de los incrementos posteriores. Tanto el diseño de la imagen corporativa de GRIAL, como la instalación de las herramientas iniciales forman parte de las tareas de esta fase. La planificación temporal realizada para este incremento engloba las definidas para el resto de incrementos, por lo que en la Figura 29 solo se muestra las tareas relativas al primer incremento, aunque el diagrama de Gantt completo abarque desde el inicio hasta la fecha de finalización de todo el proyecto.

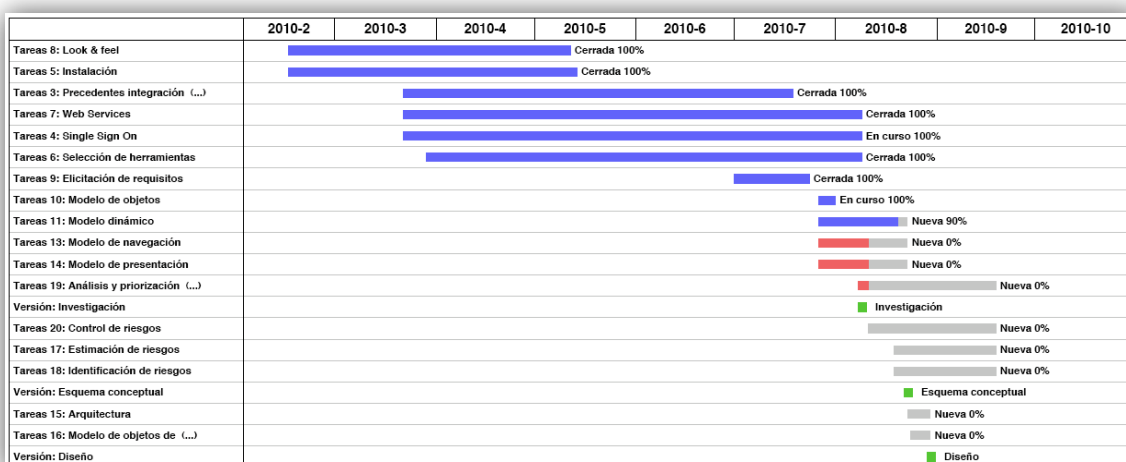


Figura 29. Diagrama de Gantt correspondiente al primer incremento

La segunda fase o incremento se ha centrado en el gestor de *blogs*. Se ha realizado la documentación técnica pertinente y todas las tareas de desarrollo y configuración del mismo, como se puede ver en la planificación temporal mostrada en la Figura 30. También se ha migrado de WPMU a WordPress 3.0 *Multisite*, tal y como se explica en el apartado 4.2.1, se han revisado traducciones, se han configurado los *plugins*, se han desarrollado los temas visuales (tanto como para el portal del gestor como para los *blogs* individuales), se ha estudiado y aplicado formas de optimizar el rendimiento del sistema y se ha realizado un análisis exhaustivo de los diferentes *plugins* utilizados.

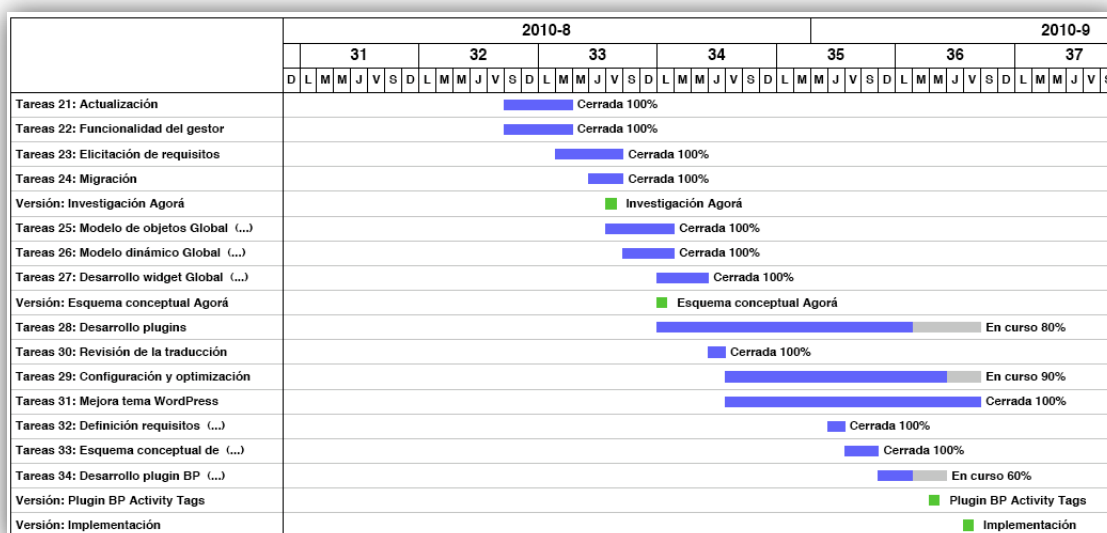


Figura 30. Diagrama de Gantt del incremento de Agorá finalizado

La tercera fase, Figura 31, se ha dedicado al campus *online* basado en Moodle, Polis. El objetivo de esta fase es mejorar los desarrollos que en su día se realizaron para Studium, de tal forma que el producto final sea en forma de módulos que puedan ponerse a disposición de la comunidad en el repositorio de Moodle.

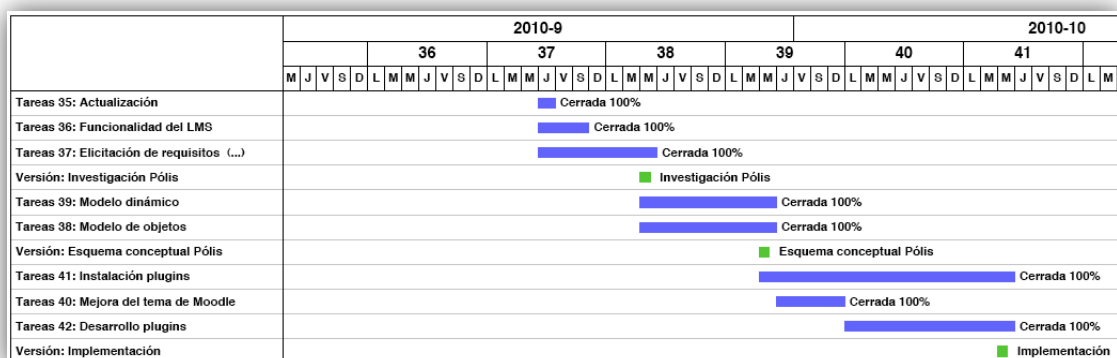


Figura 31. Diagrama de Gantt del incremento de Pólis finalizado

La penúltima fase, Figura 32, ha tenido como objetivo integrar todos los servicios conectando Drupal con Moodle y WordPress mediante SSO (*Single Sign On*). Esta fase no comenzó inmediatamente después de la fase anterior, sino que se realizó un paréntesis de un mes debido a temas urgentes dentro del grupo de investigación.

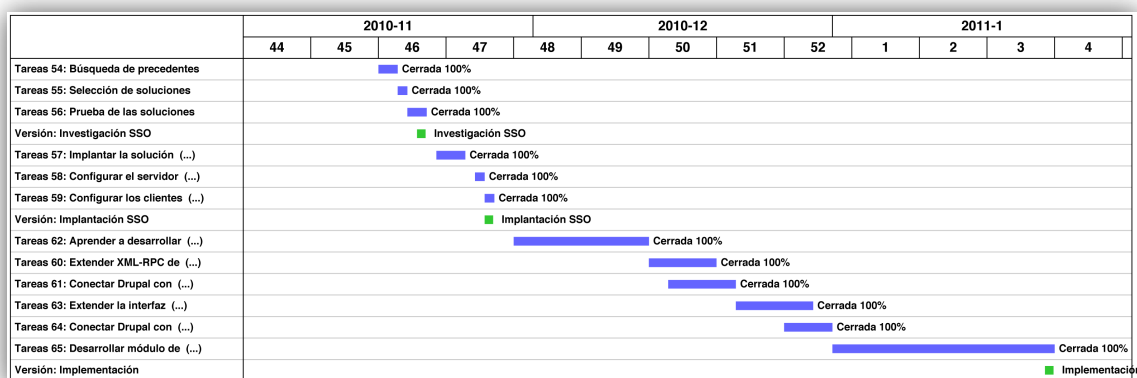


Figura 32. Diagrama de Gantt de los dos últimos incrementos

Por último, se han desarrollados los módulos de Drupal, para lo que se ha utilizado la documentación realizada en la primera fase del proyecto. Este ha sido el periodo de desarrollo más largo, debido a que se trataba del módulo principal que da soporte a gran parte de la información generada dentro del grupo de investigación.

Todos los sistemas han permanecido en funcionamiento a lo largo de todas las fases de tal forma que las mejoras y modificaciones han sido puestas a prueba por los diferentes usuarios del sistema, y siempre habían sido previamente probadas en un servidor habilitado especialmente para ello (diferente, obviamente al servidor de explotación).

6.2. Arquitectura del sistema

6.2.1. Arquitectura lógica

6.2.1.1. Servicios web

La integración entre los diferentes sistemas se ha realizado utilizando servicios web para que las dependencias entre sistemas sean mínimas, permitiendo que evolucionen de manera independiente. La opción de realizar la integración mediante el uso de la misma

base de datos generaría problemas a la hora de actualizar una de las aplicaciones, pudiendo provocar que el sistema quede obsoleto al no poder adaptarse a los cambios.

La capa de servicios mediante la cual van se comunican las diferentes aplicaciones maneja un único protocolo estándar, de tal forma que el protocolo elegido para tal fin ha sido aquel soportado por el mayor número de sistemas posibles.

Se barajaron tres opciones, XML-RPC (Figura 33), SOAP (Figura 34) y REST (Figura 35), mediante el estudio de las posibilidades que Drupal, Moodle y WordPress dan para cada una de ellas.

XML-RPC es un protocolo de llamada a procedimiento remoto, RPC, que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Se trata de un protocolo muy simple que mediante la definición de unos cuantos tipos de datos y comandos permite la comunicación entre un cliente y un servidor, de tal forma que el cliente solicita al servidor que ejecute un procedimiento o función y éste envía de vuelta el resultado al cliente.

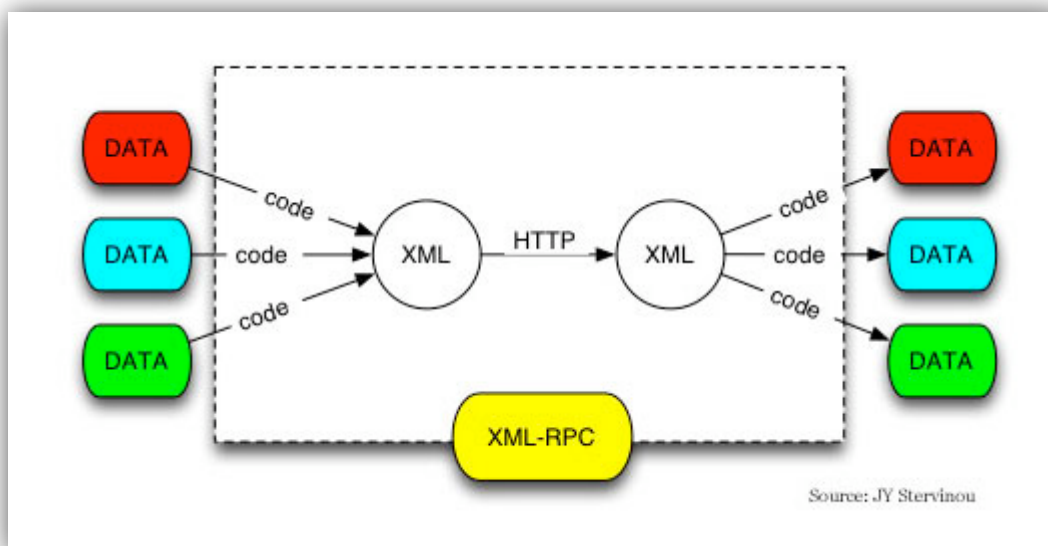


Figura 33. Esquema del protocolo XML-RPC

SOAP se trata de un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de las XML-RPC, siendo más complicado que su antecesor.

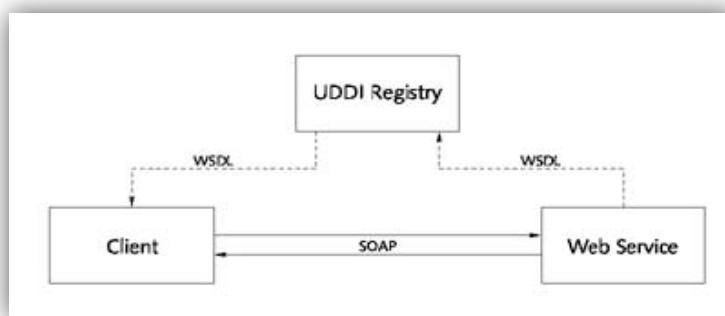


Figura 34. Protocolo SOAP

REST se trata de un protocolo orientado a recursos en vez de hacer énfasis en las operaciones del protocolo, de tal forma que con lo que se trata son componentes, objetos, sobre los que se realizan las operaciones necesarias. Los clientes trabajan con estos recursos a través de las operaciones estándar de HTTP, incluso para realizar la búsqueda de los propios recursos se utiliza otro recurso del cuál sí se debe conocer su localización.

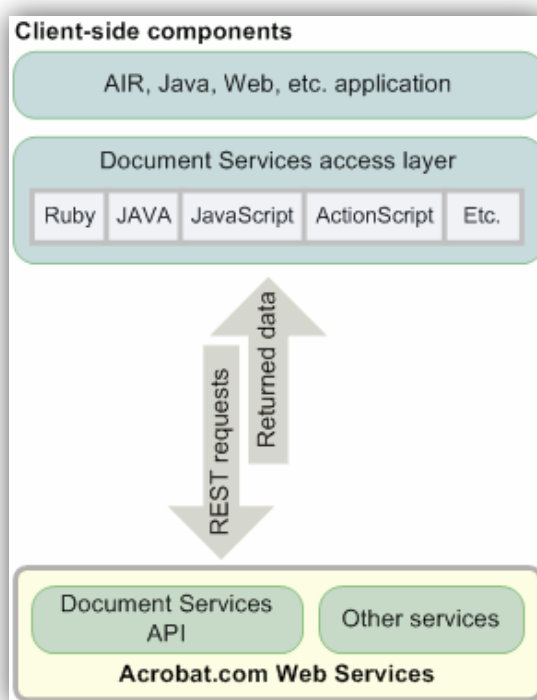


Figura 35. Protocolo REST

6.2.1.1.1. Drupal web services

En el caso que nos concierne, Drupal trabaja como un cliente de servicios web, por lo que serán las otras aplicaciones las que deberán proporcionar una interfaz de servicios adecuada.

Drupal proporciona una API para trabajar con XML-RPC, permitiendo definir clientes que se comuniquen con otros sistemas o crear tu propio servidor XML-RPC. Es importante tener en cuenta las medidas de seguridad que deben tomarse a la hora de definir el cliente XML-RPC ya que este se ejecutará como un módulo de Drupal

pudiendo recibir peticiones de usuarios anónimos si no se configura correctamente el acceso al cliente.

Existe un módulo denominado *Services* que se utiliza para integrar aplicaciones externas con Drupal. Este módulo permite crear servicios o colecciones de métodos que puedan ser utilizados por aplicaciones remotas. Los clientes podrán llamar a estos métodos mediante cualquiera de los protocolos que soporta el módulo, SOAP, REST, JSON entre otros, y esto es posible gracias a la definición de servicios web a través de múltiples interfaces que utilizan el mismo código.

Para utilizar SOAP como cliente existe un módulo, *SOAP client*, que proporciona una API que puede ser usada por otros módulos para acceder a servicios web basados en SOAP.

En el caso de REST existe un módulo, disponible para Drupal 5, que implementa una API para acceder a servicios web basados en recursos. Sería posible colaborar en la adaptación del módulo para versiones más actuales de Drupal, lo que permitiría la opción de utilizar REST como protocolo para conectar los sistemas.

6.2.1.1.2. Moodle web services

La versión 2.0 de Moodle, cuyo lanzamiento se ha ido retrasando hasta el invierno de 2010, proporciona una capa consistente de servicios web. En el momento de comenzar el desarrollo de GRIAL 2.0 la versión estable de Moodle no disponía de soporte para SOAP y REST, tan sólo disponía de Moodle NETwork API, un conjunto de llamadas remotas cuyas peticiones y respuestas se codifican según el protocolo XML-RPC, siendo la interfaz mediante la cual se pueden conectar varios Moodles creando una red.

Para crear un servidor SOAP hay una API estable que permita definir una colección de métodos sin tener que llevar a cabo la implementación desde cero. Lo único que existe es un módulo en desarrollo, totalmente experimental, denominado OKTech Web Services que proporciona servicios web para la gestión de usuarios, cursos y calificaciones.

Respecto a REST también existe una pequeña API que implementa dicho protocolo. Tan solo tiene definido el recurso *Categories* ya documentación del es prácticamente inexistente. Además la última versión del data del 2005, siendo totalmente incompatible con la versión actual de Moodle.

6.2.1.1.3. WordPress web services

WordPress dispone de una interfaz XML-RPC que permite realizar, de forma remota, diferentes tareas de gestión dentro del gestor de *blogs*. Muchas aplicaciones de publicación remota hacen uso de dicha interfaz para publicar en *blogs* basados en WordPress.

A la hora de querer definir servicios SOAP o REST hay que recurrir a la instalación de *plugins*.

Existen dos *plugins* que permiten utilizar clientes SOAP contra WordPress. El primero es WordPress Web Service, un *plugin* que, mediante una engorrosa configuración, permite seleccionar las operaciones de WordPress que podrán ser utilizadas por el cliente. El segundo *plugin*, WordPress-on-a-Rope proporciona una interfaz SOAP basada en la interfaz XML-RPC que ofrece WordPress.

En cuanto a REST actualmente no hay ningún desarrollo que permita definir servicios en WordPress que usen este protocolo.

6.2.1.1.4. **SOAP vs. XML-RPC**

Cerniéndose a los desarrollos ya existentes en los tres sistemas, el único protocolo que se soporta de forma nativa por todos ellos es XML-RPC. WordPress y Moodle proporcionan una interfaz remota a la cual se pueden realizar peticiones desde un cliente definido en Drupal mediante la API XML-RPC.

Respecto a los otros dos protocolos la situación es la siguiente.

Drupal dispone de APIs para desarrollar clientes SOAP y REST. El módulo que implementa el cliente para REST requiere de desarrollo ya que está obsoleto.

Para WordPress existen *plugins* que implementan un cliente en REST, pero no existe un servidor REST para que pueda utilizarse WordPress mediante dicho protocolo. En cambio sí que existe un *plugin*, poco probado, que permite utilizar SOAP.

En Moodle se está desarrollando una capa de servicios web que soportará todos estos protocolos, pero la versión que integrará esta nueva funcionalidad no había sido lanzada cuando se comenzó este proyecto y no se podía depender de su lanzamiento para desarrollar la aplicación. La implementación que existe de REST se ha descartado por incompatibilidad con la versión actual de Moodle, así como por su falta de documentación y por estar incompleta. Si se opta por usar SOAP se puede utilizar un módulo que actualmente está en desarrollo y que no pertenece al *core* de Moodle.

Por tanto, a pesar de su sencillez y claridad, REST queda descartado ya que la elección de este obligaría a llevar a cabo la implementación del protocolo para las tres aplicaciones, con la consecuencia de que la evolución de los sistemas a integrar sería más rígida.

En cambio SOAP podría ser una opción viable ya que existen implementaciones del protocolo más o menos estables en las tres aplicaciones.

Llegados a este punto es necesario realizar una comparación entre SOAP y XML-RPC para elegir aquel que se adapte mejor a las necesidades:

- SOAP es un protocolo descentralizado, en un entorno distribuido, para el intercambio de información. A diferencia de XML-RPC, incluye una infraestructura a su alrededor. No es un mero protocolo de comunicación entre ordenadores, sino que además se rodea de términos como WSDL y UDDI.
- Los métodos remotos en XML-RPC solo pueden contener caracteres alfanuméricos, guión bajo, punto, dos puntos y *slash*, lo cual dificulta, por ejemplo, pasar como argumento objetos.
- En XML-RPC las estructuras y los *arrays* siempre son anónimos por lo que para identificarlos habrá que recurrir al orden de los parámetros.
- XML-RPC está diseñado para ser lo más sencillo posible, lo que permite realizar la mayoría de las operaciones con un esfuerzo mínimo. En cambio SOAP está diseñado para transferir estructuras de datos más complejas, pero requiere una especificación de datos exhaustiva para describir exactamente que se está enviando.
- La curva de aprendizaje de XML-RPC es más rápida.
- SOAP deriva de XML-RPC, intentando cubrir las limitaciones que tiene XML-RPC.

Tras observar las ventajas e inconvenientes de ambos protocolos se concluye que es recomendable el uso de SOAP si se requiere utilizar complejos tipos de datos definidos por el usuario indicando como deben ser procesados. En cambio, si no hay problema en utilizar *arrays* y estructuras anónimas, y no existe la necesidad de realizar cosas complicadas que requieran definición de tipos, entonces se debe elegir XML-RPC.

En este caso la mejor opción es por tanto XML-RPC, ya que las operaciones que deben realizarse a través de llamadas a procedimiento remoto son sencillas y no tienen una gran complejidad en cuanto a definición de tipos de datos se refiere. A esto hay que unir la ventaja de que los tres sistemas incorporen este protocolo en su núcleo, lo que facilita la evolución de los sistemas por no tener dependencias a la hora de cambiar de versión con módulos o *plugins* externos.

6.2.1.2. Conexión de componentes

La parte central de GRIAL 2.0 gira en torno a la conexión entre los tres gestores de contenidos, Drupal, Moodle y WordPress.

Los elementos no se conectarán todos con todos, sino que se realizará una conexión unidireccional, es decir, se mantendrá uno de los sistemas como centro de mando conectándolo con los otros sistemas para llevar a cabo las tareas pertinentes. El usuario solo tendrá que manejar la herramienta central para poder introducir información en el sistema. Si la conexión entre herramientas fuera bidireccional, es decir, que desde cualquiera de ellas pudieras añadir información en las otras, el usuario dispondría de varias formas de hacer una misma tarea lo cual podría llegar a confundirle, al tener que aprender diferentes maneras de hacer una misma cosa. A veces un sistema demasiado flexible puede convertirse en un inconveniente para el usuario final.

Drupal desempeñará el papel de centro de operaciones, conectándose con Moodle y WordPress para evitar que el usuario tenga que trabajar con los tres sistemas para llevar a cabo una tarea completa. Por ejemplo, si se desea crear un nuevo proyecto de investigación, habrá que publicar la información relativa al proyecto en el portal, en algunos casos será necesario crear un *microsite* en el gestor de *blogs* y añadir una serie de cursos en la plataforma de *eLearning* para proporcionar un punto de reunión *online* entre los diferentes *partners*. El usuario tendría que realizar una tarea en cada uno de los sistemas de GRIAL 2.0, en cambio, conectando Drupal con WordPress y Moodle, el usuario solo tendrá que acceder a Drupal para realizar todas las tareas de una vez.

6.2.1.2.1. Conectar Drupal con WordPress: *Drupal to WP* y *Drupal to WP XML-RPC*

La conexión de estos dos sistemas se realiza mediante XML-RPC, de tal manera que WordPress proporciona una interfaz de servicios y Drupal hace uso de ellos.

La funcionalidad que Drupal necesita utilizar a través de XML-RPC es la siguiente:

- Crear un nuevo *blog*.
- Obtener los usuarios del sistema.
- Obtener los *blogs* de un usuario.
- Obtener el identificador de un *blog*.
- Realizar una nueva publicación en un *blog*.
- Obtener las categorías existentes en un *blog*.
- Crear un nuevo usuario.

- Eliminar un usuario.

Solo algunas de estas funciones están disponibles a través de la interfaz XML-RPC que proporciona WordPress, por lo que es necesario extender dicha interfaz para añadir las funciones que faltan. Para ello se ha creado *Drupal to WordPress XML-RPC*, un *plugin* para WordPress que permite utilizar algunas de las características *multisite* que no eran accesibles a través de este protocolo.

Para llevar a cabo cualquier operación a través de XML-RPC es necesario proporcionar un usuario y una contraseña válidos para controlar que solo los usuarios con los privilegios adecuados utilicen las funciones disponibles a través de este protocolo. Por defecto, la interfaz XML-RPC de WordPress requiere el usuario y la contraseña en texto plano, es decir, que cualquier usuario que esté escuchando la comunicación entre Drupal y WordPress podrá obtener los datos de acceso a nuestro gestor de *blogs*. Para solucionar este agujero de seguridad se ha utilizado un sistema de cifrado, de tal forma que el usuario y la contraseña son enviados en base 64 cifrados mediante el algoritmo de Rijndael de 256 bits. Cuando llegan al servidor este los descifra utilizando la misma clave que se definió en el cliente, dicha clave se conoce de antemano, no se envía de un servidor a otro ya que podría ser escuchada y cualquiera podría descifrar los datos de acceso.

Aunque algunas de las funciones necesarias ya existen en la implementación XML-RPC de WordPress, ha sido necesario recubrirlas para que soporten el método de acceso cifrado y mantener una uniformidad en el modo de utilización de la extensión XML-RPC que proporciona el *plugin*.

Hay un último aspecto que se debe tener en cuenta para utilizar la interfaz XML-RPC de WordPress, tanto la nativa como la extensión que se ha desarrollado. Para manejar un *blog* a través de XML-RPC es necesario que tenga activado este protocolo de publicación tal y como se muestra en la Figura 36. Aunque todos los *blogs* se puedan crear con esta opción activada por defecto eso no impide que un usuario la cambie. Hacer que esta opción desaparezca de la configuración del *blog* implica modificar ficheros que forman parte del núcleo de WordPress. Realizar ese tipo de modificaciones supone una práctica peligrosa ya que establecería dependencias con la actual versión de la herramienta y dificultaría la actualización de la misma. Por este mismo motivo WordPress proporciona una serie de acciones o *hooks* que permiten modificar la mayor parte de la funcionalidad desde *plugins* que se desactivan en caso de que la nueva actualización de WordPress no soporte esas modificaciones. Para evitar este tipo de problemas se ha creado un pequeño *plugin*, denominado *MS Force Options*, cuya finalidad es forzar el valor de esta opción, de tal forma que las modificaciones por parte del usuario serán ignoradas.

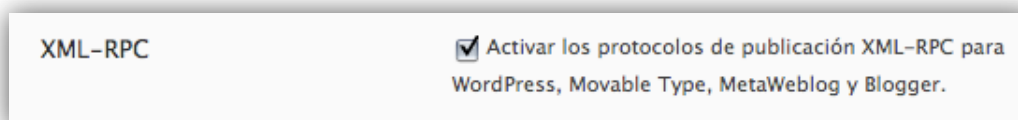


Figura 36. Opción de activación de XML-RPC en WordPress

En el lado de Drupal ha sido necesario desarrollar un módulo que encapsule la conexión mediante XML-RPC con WordPress, de tal manera que esta funcionalidad pueda ser utilizada por otros módulos de Drupal sin tener que conocer cómo se está realizando. El

módulo, denominado *Drupal to WP*, incluye una biblioteca de conexión con WordPress y una página de configuración, como la que se muestra en la Figura 37, en la que se deben indicar:

- El dominio del gestor de *blogs*.
- En caso de encontrarse en un subdirectorío se debe indicar.
- La clave para cifrar que comparten el cliente y el servidor.
- El nombre de un usuario con permisos de administrador.
- La contraseña correspondiente al usuario anterior. Se almacenará cifrada en la base de datos.

En la Figura 37 también se puede ver una pestaña denominada “Check”, sirve para, una vez configurado el módulo, comprobar si funciona correctamente. También hace las funciones de guía para otros programadores que deseen ver un ejemplo de utilización de las funciones de la biblioteca.

Figura 37. Configuración del módulo *Drupal to WP*

Se ha intentado minimizar, en la medida de lo posible, las dependencias entre el *plugin* y el módulo, de tal forma que el *plugin* puede ser utilizado por cualquier otra aplicación que lo requiera siempre que las credenciales se envíen con el cifrado adecuado. En cambio el módulo sí depende del *plugin* de WordPress para su funcionamiento ya que realiza llamadas a través de XML-RPC que sólo están disponibles si se encuentra dicho *plugin* instalado.

6.2.1.2.2. **Conectar Drupal con Moodle: *Drupal to Moodle* y *Drupal to Moodle xmlrpc***

La conexión de estos dos sistemas se realiza también mediante XML-RPC. En este caso Moodle proporciona una interfaz de servicios y Drupal hace uso de ellos.

La funcionalidad que Drupal necesita utilizar a través de XML-RPC es la siguiente:

- Obtener las categorías de cursos existentes en el sistema.
- Crear una nueva categoría.
- Crear un nuevo curso.
- Obtener los usuarios del sistema.
- Asignar docentes a un curso.
- Crear una noticia en el foro principal.

En Moodle, en vez de extender la interfaz XML-RPC que viene por defecto, se ha optado por implementar una interfaz nueva, sencilla y potente, que permite realizar operaciones básicas de gestión de cursos. Esta nueva interfaz está construida utilizando la misma filosofía que la nativa de Moodle, es decir, hace uso de la biblioteca *xmlrpc* de PHP.

Se ha creado un directorio con el nombre “xmlrpc” y dentro de él se encuentra la interfaz XML-RPC desarrollada. El nombre del directorio no es vinculante, es decir, se puede incluir en otro directorio dentro de la instalación de Moodle, lo importante es que a la hora de utilizar la interfaz se indique correctamente la ruta donde se encuentra.

Para manejar Moodle a través de XML-RPC, ya sea con la implementación añadida o con la nativa, es necesario que se active el permiso desde la administración (Ver Figura 38). De esta forma se puede desactivar el servicio cuando sea necesario.

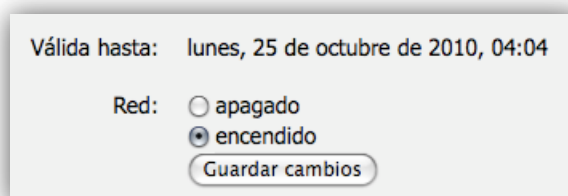


Figura 38. Activar el uso de XML-RPC en Moodle

Al igual que en el *plugin* de WordPress, para llevar a cabo cualquier operación a través de la nueva interfaz XML-RPC es necesario proporcionar un usuario y una contraseña válidos para controlar que sólo los usuarios con los privilegios adecuados utilicen las funciones disponibles a través de este protocolo. El usuario y la contraseña son enviados en base 64 cifrados mediante el algoritmo de Rijndael de 256 bits y cuando llegan al servidor este los descifra utilizando la misma clave que se definió en el cliente.

En el lado de Drupal se ha desarrollado otro módulo, denominado *Drupal to Moodle*, que en este caso encapsula la conexión mediante XML-RPC con Moodle, de tal manera que esta funcionalidad pueda ser utilizada por otros módulos de Drupal sin tener que conocer cómo se está realizando, lo que permite que esta cambie sin afectar al resto de módulos. El módulo incluye una biblioteca de conexión con Moodle, una página de

configuración como la mostrada en la Figura 39 y una página de comprobación al igual que en *Drupal to WP*.

Figura 39. Configuración del módulo *Drupal to Moodle*

En este caso también se ha intentado minimizar las dependencias entre Moodle y el módulo, de tal forma que la nueva interfaz XML-RPC para Moodle puede utilizarse por cualquier otra aplicación que la requiera siempre que las credenciales se envíen con el cifrado adecuado. En cambio el módulo sí depende del de dicha interfaz para su funcionamiento.

6.2.2. Arquitectura física

El sistema se compone de cuatro servidores, dos físicamente ubicados en el CPD de la Universidad de Salamanca, grial.usal.es y grial4.usal.es, y otros dos en la Facultad de Ciencias de dicha Universidad, antia.fis.usal.es y grial3.usal.es. Puede observarse la distribución de componentes en el diagrama de despliegue de la Figura 40.

Cada uno de los servidores desempeña un papel dentro de todo el sistema. El servidor principal es grial.usal.es, debido a que es donde se encuentran las aplicaciones principales del sistema, Agorá, Pólis y el portal Drupal. Posee un procesador de 64 bits Intel(R) Xeon(R) E5530 a 2.40GHz con 7GB de memoria. Utiliza como sistema operativo un Ubuntu Server 9.10 con PHP 5.2.10, MySQL 5.1.37 y Apache/2.2.12.

Grial3.usal.es se trata del servidor de identidad OpenID y a su vez es utilizado para el desarrollo y la fase de preproducción y testeo de las aplicaciones antes de pasar a funcionar al servidor principal. Tiene instaladas réplicas de Agorá, Pólis y el portal para poder realizar las pruebas en un entorno lo más parecido al de producción. Además mantiene una instalación de Subversion para utilizar el servidor como repositorio en futuros proyectos. Posee un procesador de 64 bits Intel(R) Xeon(TM) a 3.20GHz con 2GB de memoria. Utiliza como sistema operativo un Ubuntu Server 10.04 con PHP 5.3.2, MySQL 5.1.41 y Apache/2.2.14.

El servidor grial4.usales está destinado a todas aquellas aplicaciones que utilizan Tomcat y/o PostgreSQL, así como aquellas que estén desarrolladas en Ruby o Rails. Tanto la aplicación de gestión de proyectos como el repositorio de datos de GRIAL se encuentran en este servidor. Posee un procesador de 64 bits Intel(R) Xeon(R) E5530 a 2.40GHz con 7GB de memoria. Utiliza como sistema operativo Ubuntu Server 9.10 con PHP 5.2.10, MySQL 5.1.37, Ruby 1.8.8, Rails 2.3.5, PostgreSQL 8.4.1, Apache/2.2.14 y Tomcat 6.

Por último antia.fis.usales desempeña la labor de disco duro virtual dividido en dos partes, una totalmente privada y otra de solo lectura. Ambas son accesibles mediante WebDAV desde cualquier cliente que soporte este protocolo y están protegidas contra escritura utilizando el módulo HTTP Digest Authentication de Apache. En un futuro funcionará como servidor de la herramienta de *webconference* de GRIAL, una de las líneas futuras de desarrollo que surgen a partir de este proyecto. Actualmente existe una instalación de OpenMeetings, una herramienta *open source* que permite realizar conferencias a través de Internet y que funciona únicamente desde dentro de la red de la USAL debido a conflictos de puertos con el *firewall*. Posee un procesador de 64 bits Intel(R) Xeon(R) 3040 a 1.86GHz. Utiliza como sistema operativo Debian 2.6.26 con PHP 5.2.6, MySQL 5.0.51, Apache/2.2.9 y Red5 0.9.

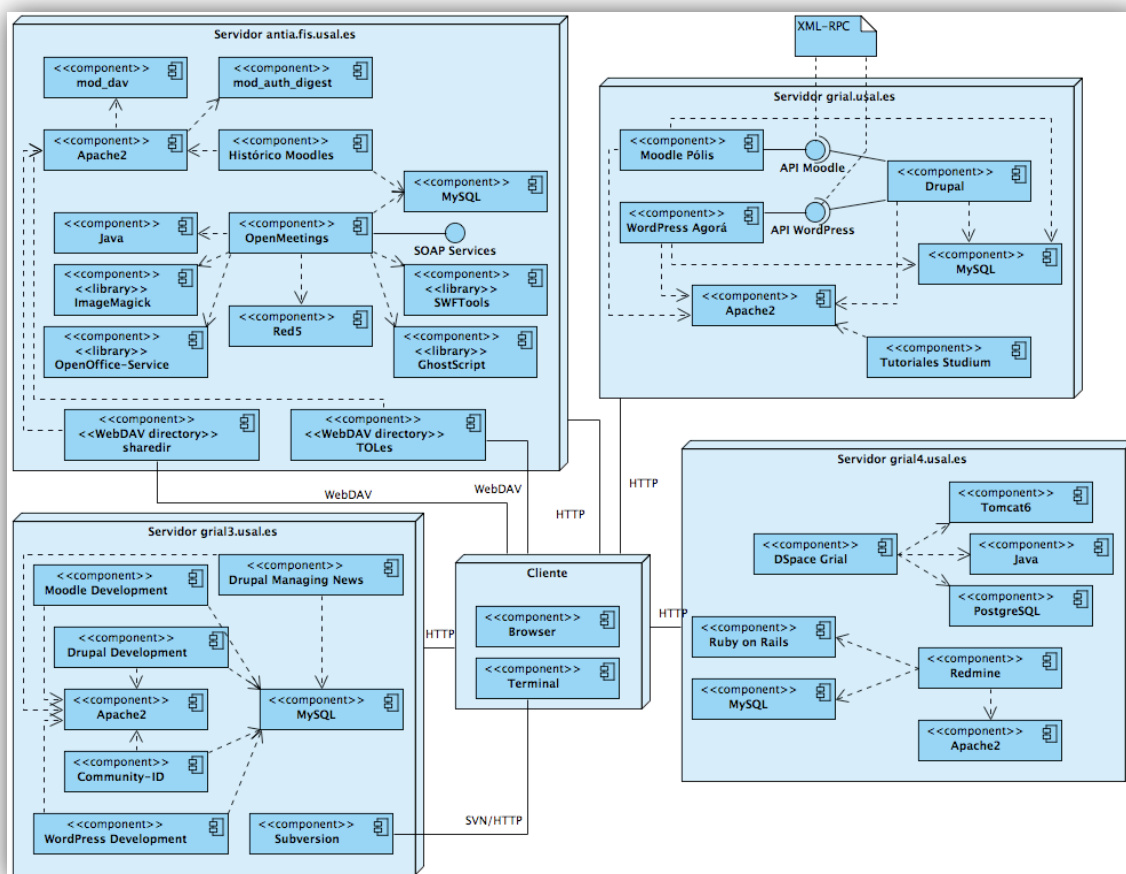


Figura 40. Diagrama de despliegue del sistema

6.3. *Single Sign On*

Uno de los objetivos que debe cubrir este proyecto es unificar la autenticación en los diferentes servicios que conforman GRIAL 2.0, de tal forma que el usuario solo tendrá que introducir su contraseña la primera vez que acceda a uno de las herramientas.

Algunas de las ventajas de utilizar SSO son las siguientes:

- Una única pantalla donde autenticar a los usuarios.
- Las aplicaciones que requieren autenticación no tienen acceso a las claves.
- Los usuarios no necesitan autenticarse en cada acceso a una nueva aplicación.
- Centralización de la información del usuario en vez de tenerla duplicada en los diferentes sistemas.
- Envío de atributos junto con la autenticación.
- Reduce los riesgos de sufrir *phishing* ya que el usuario sólo introduce la contraseña en el *identity provider* o servidor de identidad.

Se han barajado dos opciones para implementar SSO, OpenID y OAuth.

OpenID es un sistema abierto que permite compartir una única autenticación entre todas las webs que lo implementan. Este sistema protege los datos de los usuarios estableciendo identidades únicas, aunque descentralizadas, por toda la red.

OAuth es un estándar abierto y simple para la autenticación segura de APIs. Permite a los usuarios que compartan sus datos privados almacenados en el proveedor de servicio con los consumidores, aplicaciones que desean acceder a los recursos privados del usuario. Es el protocolo elegido por Twitter que, además de permitir compartir autenticación entre distintas web, igual que hace OpenID, permite acceso a la API, es decir, poder interactuar con los servicios de una página web desde otra.

Existen una serie de ideas equivocadas acerca de la relación entre OAuth y OpenID. OAuth no sustituye a OpenID, sino que pueden complementarse. OAuth no es una extensión de OpenID, sino que tienen dos enfoques distintos, OAuth se centra en la autorización, permite así acceder a información privada sin compartir por completo la identidad, mientras que OpenID se centra en la autenticación mediante una única identidad para acceder en varios sitios. Es cierto que se puede utilizar OAuth para implementar el acceso a diferentes sistemas, pero lo interesante es utilizar sus características para el acceso a APIs.

En el caso que concierne el objetivo es mantener una única autenticación en los diferentes servicios. Es cierto que para realizar la integración de funcionalidad entre los diferentes sistemas será necesario proporcionar APIs para utilizar los servicios de Moodle y WordPress, pero el uso de OAuth obligaría a mantener un proveedor de servicios por cada sistema, lo que no cumpliría con el requisito de mantener una única interfaz de autenticación.

Al no poder explotar la característica principal de OAuth se ha optado por realizar el SSO mediante el protocolo OpenID.

6.3.1. *OpenID provider de GRIAL*

En un primer momento el objetivo era que Drupal desempeñara la labor de OpenID *provider*, mientras que Moodle y WordPress funcionarían como clientes. Tras un análisis más exhaustivo, se decidió mantener la capa de autenticación separada de los

componentes del sistema, de tal forma que no se dependiera de uno de ellos para que todo el proceso de autenticación funcionara.

Al configurar un OpenID *provider* independiente de las aplicaciones del sistema se reducen las dependencias entre elementos, de tal forma que si en un futuro se decide cambiar de gestor de contenidos o se desean añadir más aplicaciones tan solo habrá que instalar o desarrollar el cliente OpenID correspondiente.

Otra de las ventajas de mantener esta separación entre aplicaciones es que el método de acceso en todas ellas será igual, el usuario introducirá su identificador OpenID y al pulsar el botón de *login* será enviado a la pantalla de autenticación del OpenID *provider*, una pantalla parecida a la que se muestra en la Figura 44. En la **Figura 41, la Figura 42 y la Figura 43** se pueden ver los formularios de *login* de correspondientes a WordPress, Moodle y Drupal.

The image shows a web form titled "OpenID" in red text. Below the title is a text input field with a small OpenID icon on the left. Below the input field is the text "¿Qué es esto?" and a button labeled "Entrar".

Figura 41. Formulario de *login* de WordPress

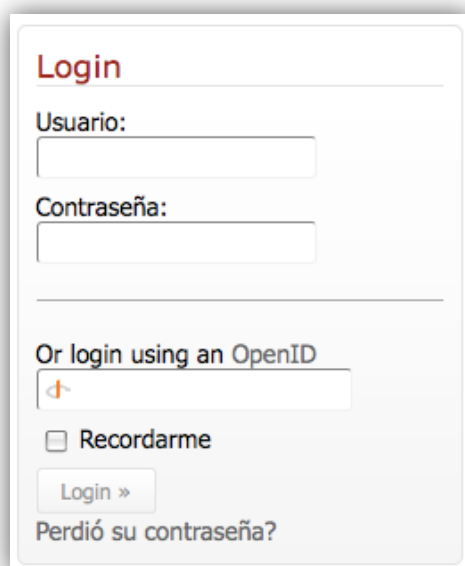
The image shows a web form titled "Login" in red text. Below the title are two text input fields labeled "Usuario:" and "Contraseña:". Below these fields is a horizontal line. Below the line is the text "Or login using an OpenID" and a text input field with a small OpenID icon on the left. Below the input field is a checkbox labeled "Recordarme". Below the checkbox is a button labeled "Login »". Below the button is the text "Perdió su contraseña?".

Figura 42. Formulario de *login* de Moodle



The image shows a Drupal login form titled 'Inicio de sesión'. It features a section for 'Entrar usando OpenID:' with a small OpenID icon and a link '¿Qué es OpenID?'. Below this is a button labeled 'Iniciar sesión'. At the bottom, there are two links: 'Cancelar entrada vía OpenID' and 'Solicitar una nueva contraseña'.

Figura 43 Formulario de *login* de Drupal

El proveedor de identidad, o OpenID *provider*, ha sido instalado en un servidor distinto al que se encuentran el resto de herramientas, manteniéndolo físicamente separado de los consumidores.

Se ha elegido como servidor OpenID de GRIAL la herramienta Community-ID (Ver Figura 44). Se trata de un proyecto *open source* desarrollado en PHP que utiliza Zend Framework. Entre sus características principales cabe destacar la obtención de estadísticas del tráfico recibido y la posibilidad de que cada usuario mantenga diferentes perfiles datos.



The image shows the GRIAL Community-ID login form. At the top is the logo 'GRIALCommunityid'. Below it, the 'OpenID URL:' is set to 'http://grial3.usal.es/openid/identity/aliciagh'. There is a text input field for 'Contraseña:'. At the bottom is a button labeled 'Ingresar'.

Figura 44. Community-ID de GRIAL

Por defecto la herramienta viene configurada para que usuarios externos puedan registrarse. Dicha funcionalidad ha sido desactivada ya que tan solo se quiere dar servicio a los miembros del grupo de investigación, a los que se les crean las cuentas de manualmente.

6.4. Desarrollo en Drupal

6.4.1. Research Support

La mayor parte de desarrollo necesario para construir GRIAL 2.0 se concentra en Drupal. A parte de los dos módulos mencionados en el apartado 6.2.1.2 *Drupal to WP* y *Drupal to Moodle*, ha sido necesario desarrollar un módulo que proporcione las herramientas necesarias para mantener un portal de investigación de tal forma que las tareas de publicación y edición de nuevos contenidos sean rápidas y sencillas para el usuario final.

Al tratarse de un portal con una gran cantidad de tipos de información estructurarlo y añadir nuevas páginas resulta una tarea costosa y un pequeño aprendizaje previo, lo cual hace que el peso de aportar información al portal recaída en un número mínimo de personas. Para que el portal tenga vida y se aproveche como medio de difusión de la actividad dentro del grupo de investigación, es necesario que todos los miembros de GRIAL puedan aportar contenidos al sistema.

Drupal permite definir módulos que agreguen nuevos tipos de contenido al portal y que se administran como los contenidos nativos, los artículos y las páginas. GRIAL 2.0, además de definir nuevos tipos de contenidos, necesita que se les dote de funcionalidad extra. No se trata únicamente de definir formularios con los campos apropiados, sino que los nuevos tipos de contenido deben cumplir cinco objetivos principales:

1. Disponer de una interfaz intuitiva y amigable que facilite al usuario la tarea de publicación de información.
2. Automatizar la definición de la miga de pan para que se muestren correctamente todas las secciones hasta llegar a la información en cuestión tal y como se muestra en la Figura 45).

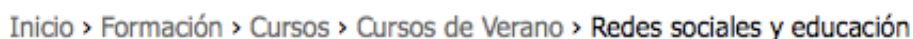


Figura 45. Ejemplo de miga de pan o *breadcrumb*

3. Gestionar automáticamente la disposición de la información dentro del portal, es decir, el usuario no se encargará de que el nuevo proyecto de investigación aparezca en la sección correspondiente o que un nuevo curso se muestre en el listado de cursos. Será el módulo el encargado de realizar dicha tarea y se basa en las opciones de configuración que indiquen la sección que corresponde a cada tipo de contenido.
4. Funcionar como interfaz para trabajar con WordPress y Moodle de forma transparente, es decir, el usuario no necesita saber cómo se crea un *microsite* o como se añade un curso a la plataforma de enseñanza, tan solo tiene saber que debe rellenar una serie de campos con datos válidos.
5. Diseminar la información de forma automática sin que sea necesaria la intervención del usuario, es decir, el módulo deberá encargarse de diseminar la información nueva siguiendo la política de diseminación indicada en el apartado 6.7.2.

El número de tipos de información que debe gestionar el portal de GRIAL incluye diferentes tipos de proyectos, cursos, noticias, resultados de transferencia,

publicaciones, etc. Definir un tipo de contenido para cada uno de estos elementos obligaría a duplicar gran parte del código ya que la definición de la miga de pan, las tareas asociadas a mantener la estructura del portal y la diseminación automática de la información deberán existir en todos los tipos. Además el usuario percibirá mayor complejidad a la hora de añadir un nuevo contenido si dispone de un número elevado de opciones, aunque realmente se trate de lo mismo que si dispone de un único contenido en el que seleccionar el tipo de información que va a publicar. En cuanto a la flexibilidad para añadir nuevos contenidos será más sencillo si todos los tipos de información se encuentran en el mismo módulo ya que gran parte de la funcionalidad estará ya implementada, teniendo únicamente que añadir aquellas partes que difieran del resto. Por tanto se ha optado por desarrollar un único módulo que englobe los diferentes tipos de información.

El nuevo módulo, orientado al soporte de la investigación, de ahí el nombre, permitirá que el usuario publique la información deseada siguiendo un pequeño *wizard* que le guiará por el proceso. No será necesario que disponga de conocimientos acerca de otras partes de la gestión de Drupal, tan sólo deberá conocer el uso del módulo para construir un portal de investigación rico en contenidos.

Para cubrir los objetivos principales del módulo ha sido necesario utilizar los dos módulos previamente desarrollados, *Drupal to WP* y *Drupal to Moodle*, para trabajar con WordPress y Moodle respectivamente, el módulo *Twitter* para diseminar la información a la cuenta GRIAL en Twitter y el módulo *menu*, que forma parte del núcleo de Drupal, para gestionar la estructura del portal. Los módulos desarrollados en este mismo proyecto podrían haberse incluido en uno solo pero, a pesar de suponer un mayor tiempo de desarrollo, se ha querido realizar una clara separación de funcionalidad para permitir que los módulos de conexión entre sistemas puedan ser utilizados por otros usuarios de la comunidad Drupal.

La mejor forma para ver de manera clara cómo funciona el módulo *Research Support* es mediante la realización de un ejemplo con uno de los tipos de información que abarca las dos conexiones con sistemas externos, ya que un proyecto, además de la información técnica, puede necesitar un *microsite* o un curso en la plataforma de enseñanza.

- Figura 46. En primer lugar el usuario tiene que seleccionar el tipo de información que desea añadir al sistema, si no lo hace el *wizard* no continuará.

Inicio > Crear contenido > Crear Research support

Crear Research support

Content type: *

- ☐ Award
- ☐ Course
- ☐ Event
- ☐ Journal
- ☐ News
- ☐ Partner
- ☒ Project
- ☐ Publication
- ☐ Regulated Formation
- ☐ Resource
- ☐ Transfer Result
- ☐ Tutorial
- ☐ University-specific Degree

Select a content type

Next

Figura 46. Primer paso para crear un contenido de tipo *Research Support*

- Figura 47. El sistema mostrará el formulario correspondiente al tipo de data seleccionado.
- El usuario completará todos los campos de carácter obligatorio relativos a la información técnica del proyecto.

Inicio > Crear contenido > Crear Research support

Crear Research support

Título: *

▼ Opciones del menú

Section projects: *

Investigation project

Título del enlace del menú.:

Peso:

0

Project type: *

Figura 47. Campos para crear un proyecto de investigación

- Figura 48. Si el proyecto requiere un *microsite* y GRIAL es el responsable de esa parte técnica entonces el usuario debe rellenar los campos mostrados bajo la etiqueta “Microsite settings”. Solamente deberá indicar la ruta relativa a la URL mostrada, un título y el usuario del gestor de *blogs* que será el administrador del *microsite*. Si el usuario no existe puede añadirse, para lo que hay que aportar un nombre de usuario y un *email*.

▼ Microsite settings

Microsite path http://grial.usal.es/agora/:

Microsite title:

Admin:

▼ Add a new user

New user login:

New user email:

Figura 48. Parte del formulario de proyecto para crear un *microsite*

- Figura 49. Si el proyecto requiere mantener cursos en la plataforma de enseñanza, bien para impartir cursos o para servir de herramienta de

comunicación entre los diferentes *partners* del proyecto, el usuario debe rellenar los campos mostrados bajo la etiqueta “Course settings”. Será necesario indicar la categoría en la que se va a insertar el nuevo curso o crear una nueva, el nombre completo, el nombre corto, los usuarios del sistema que van a ser docentes en el curso y el número de cursos que se desea crear, de tal forma que si se indica un número mayor de uno se crearán el número de cursos indicado, para lo que se añade al nombre largo y al corto un número detrás.

▼ Course settings

Course category:

▼

— ▶ Add a new course category —

Fullname:

Shortname:

Course teachers:

Admin Grial dev (grial@usal.es)
 Adriana Berlanga (adrianaberlanga@tutoron-line.com)
 Alicia García Holgado (alicia_86gh@hotmail.com)
 Ángeles Bosom (abosom@tutoron-line.com)
 Antón Seoane (aseoane@usal.es)

How many courses do you need?:

Figura 49. Parte del formulario de proyecto para crear cursos

- Figura 50. Aunque no sea necesario, el usuario tiene la opción de decidir si el contenido no será publicado en Twitter. Los contenidos creados con *Research Support* se publicarán en Twitter a menos que se indique lo contrario.

▼ Publicar en Twitter

☒ Compartir este artículo en Twitter

!title !tinyurl

The given text will be posted to twitter.com. You can use !url, !url-alias, !tinyurl, !title and !user as replacement text.

Figura 50. Módulo Twitter

- Figura 51. Por último el nuevo contenido es previsualizado, lo que permite al usuario realizar las modificaciones oportunas antes de publicarlo.


Inicio > Investigación > Proyectos > Vista previa

Vista previa

Multicultural Interdisciplinary Handbook: Tools for Learning History and Geography in a Multicultural Perspective (MIH)

Enviado por admin el Jue, 03/02/2011 - 05:43.

Financial institution: Unión Europea. Lifelong Learning Programme. Sub-Programme COMENIUS Multilateral Project.

Project type:  eu.

Beneficiary: Universidad de Salamanca (Spain).

Main researcher: Francisco J. García Peñalvo.

Reference: 502461-2009-LLP-ES-COMENIUS-CM.

Duration: 30/09/2009 - 29/09/2011.

Total amount: 427.670€

Number of researchers: 12.

Summary:

Este proyecto tiene por objetivo la construcción de una serie de objetos digitales de aprendizaje y una manual de uso de los mismos para las asignaturas de Geografía e Historia en Enseñanza Secundaria y su uso en los distintos países de los miembros del proyecto. Asimismo, se impartirá formación a profesores (en formación y en servicio) sobre cómo usarlos. El objetivo final, más allá de ofrecer objetos de aprendizaje con más efectividad en el proceso educativo y con más atractivo para el estudiante, es fomentar la identidad europea de los estudiantes mediante materiales educativos que permitan ver y comprender el punto de vista del Otro, algo que pocas veces pasa con las políticas educativas nacionales.

URL: <http://mihproject.eu>


state: 

Figura 51. Vista previa del proyecto antes de ser publicado

6.4.2. Drupal *theme*. La imagen corporativa de GRIAL

Drupal, al igual que la gran mayoría de CMSs, separan la funcionalidad de la interfaz mediante el uso de temas (*themes* en inglés). Cada página mostrada se genera a partir de una plantilla que se rellena con el contenido de la base de datos, esto permite que la interfaz sea completamente personalizable si se tienen los conocimientos necesarios para desarrollar un tema propio.

Un tema de Drupal se compone de un conjunto de plantillas en PHP que combinan HTML con el uso de variables en PHP, a través de las cuales el sistema rellena la plantilla para posteriormente mostrarla en el navegador.



Figura 52. Portal de GRIAL con el tema Zen activado

Existen miles de temas *open source*, pero para mantener el *look & feel* de GRIAL ha sido necesario el desarrollo de un tema propio. Se ha partido de la base del tema base de Drupal, Zen Figura 52, se ha definido una colección de plantillas que le dan al portal el aspecto mostrado en la Figura 53.



Figura 53. Portal GRIAL con el tema Grial activado

6.5. AGORÁ. El gestor de *blogs* de GRIAL

6.5.1. Portal social

El gestor de *blogs*, además de albergar los diferentes *blogs* de los usuarios, debe tener un portal de presentación con características sociales en el que esté contenido el *blog* del grupo de investigación. Para ello será necesario instalar *BuddyPress*, un *plugin* que inicialmente fue desarrollado para WPMU y que se ha adaptado a la característica de *multisite* de WordPress 3.0. Dicho *plugin* es la base para otra serie de *plugins* que añaden funcionalidad social a WordPress, desde perfiles personalizados, pasando por mensajería interna entre usuarios o una herramienta de conexión entre usuarios. Para que *BuddyPress* funcione correctamente necesita un tema de WordPress adaptado al *plugin*, dicho tema será utilizado únicamente en el *blog* principal y restringido en el resto de *blogs*, de esta forma la parte social del gestor se concentrará en el *blog* principal y el resto de *blogs* utilizarán temas no compatibles con *BuddyPress* para evitar recargar todo el gestor con la funcionalidad social y mantener así el aspecto clásico de un *blog*.

6.5.2. Funcionalidad: selección de *plugins*

La forma de añadir funcionalidad a WordPress es mediante el uso de *plugins*. Los usuarios no podrán instalar nuevos *plugins* por lo que es importante que se les proporcione una buena colección que cubra sus principales necesidades. En el caso de que un usuario necesite alguna funcionalidad que no se proporcione en los *plugins* ofrecidos por defecto, si el usuario lo solicita y el *plugin* pasa las pruebas de estabilidad y seguridad pertinentes, será instalado y puesto a disposición de todos los usuarios del sistema.

Algunos *plugins* estarán activados por defecto en todos los *blogs*, mientras que otros podrán ser activados a elección del usuario. De esta forma se evita al usuario tener que preocuparse, por ejemplo, de cómo embeber recursos multimedia de otros servicios web o cómo poner un formulario de contacto. Al no tener que buscar o activar dichos *plugins* la visión que el usuario tendrá del sistema será de una herramienta muy completa que le ofrece ventajas frente a tener una instalación propia o un *blog* en Wordpress.com. Al contrario de lo que pueda parecer, esta activación obligatoria no quita flexibilidad al gestor, ya que los *plugins* que formen parte del núcleo serán aquellos que proporcionen funcionalidades básicas que se consideran imprescindible a día de hoy en una herramienta web.

Para realizar la integración de las redes sociales en el gestor de *blogs* es necesario instalar varios *plugins*. En primer lugar se necesita conectar los *blogs* con Twitter de tal forma que al realizar una publicación se envíe automáticamente un *tweet* a la cuenta asociada, y también es importante que los últimos *tweets* se puedan mostrar en el *blog*. *Twitter Tools* cubre estas necesidades y además integra Bit.ly.

También se requiere un método sencillo para que los lectores del *blog* puedan compartir las publicaciones con otras personas a través de diferentes webs sociales tales como Facebook, Delicious o Digg. Entre todos los *plugins* que implementan esta funcionalidad cabe destacar tres: *AddToAny*, *ShareThis* y *AddThis*. Todos ellos proporcionan un listado muy completo de servicios con los que compartir el contenido del *blog* y su funcionamiento es bastante parecido. Esta funcionalidad también debe estar presente en el portal del grupo de investigación por lo que es conveniente que el servicio elegido funcione correctamente también en Drupal. Después de probarlos tanto en Drupal como en WordPress se ha decidido utilizar *AddThis*.

Al querer un gestor de blogs 2.0 se hace imprescindible posibilitar la inserción de elementos multimedia provenientes de otros servicios web. Es por ello que serán necesarios *Viper's Video Quicktag* para permitir la inserción de vídeos, *SlideShare* y *WP Issuu* para insertar elementos de los correspondientes servicios, y *Flickr Gallery* para mostrar las galerías de Flickr dentro del *blog*. Como complemento a las herramientas externas también se instalará un potente *plugin* para gestión de imágenes dentro del propio *blog*, *NextGEN Gallery*.

Algunos usuarios pueden desear añadir una página de contacto dentro de su *blog*, de tal forma que a través de un formulario cualquier usuario pueda ponerse en contacto con ellos. Al entrar en juego el uso de un formulario se debe poner especial cuidado en evitar *spam* y usos indebidos del formulario en general. La mejor opción en este caso es utilizar *Contact Form 7*, que permite crear formularios de contacto personalizados, y complementarlo con *Really simple CAPTCHA*, que permitirá añadir un campo CAPTCHA al formulario. Además es necesario instalar el *plugin WP mail SMTP* para poder configurar adecuadamente el envío de correo a través del servidor de correo de la USAL y de esta forma evitar que se le tengan que dar permisos especiales a la máquina donde vaya a estar en explotación el gestor de *blogs*.

Al igual que el formulario de contacto se protege con un CAPTCHA, también debe hacerse lo mismo con el formulario de envío de comentarios. Para este propósito se utilizará el *plugin SI CAPTCHA Anti-Spam* en combinación con *Akismet*, *plugin* encargado de supervisar todos los comentarios escritos para detectar aquellos que sean *spam*.

Otro de los requisitos que debe cumplir el sistema es SSO, *Single Sign On*. Tal y como se indica en el apartado correspondiente a autenticación, Drupal, WordPress y Moodle

deberán permitir utilizar OpenID para acceder al sistema. El *plugin* denominado *OpenID* proporciona exactamente dicha funcionalidad.

Sidebar Login permite integrar el formulario de acceso como un *widget* más dentro del *sidebar*, de esta forma el usuario no tendrá que cambiar de página cuando inicie o cierre sesión.

En WordPress, utilizando un tema adecuado, se permite añadir bloques o *widgets* que son mostrados en todas las páginas del *blog*. Mediante *Display Widget* se consigue obtener mayor flexibilidad en la configuración de la estructura del blog ya que añade una opción en cada uno de los *widgets* para indicar en qué páginas se desea que se vea el *widget* y en cuáles no.

Ajax Calendar y *Calendar* incrementan la funcionalidad del calendario que WordPress trae por defecto. El primero evita que la página web se recargue cada vez que el usuario navega por el calendario. El segundo *plugin* permite realizar gestión de eventos pudiéndolos mostrar dentro de un calendario por categorías con colores asociados.

No se pueden dejar de lado los *plugins* que utilizan servicios de Google. *Google Analytics for WordPress* permite centralizar la recolección y análisis de estadísticas del gestor de *blogs* en la herramienta Analytics. Y *Google XML sitemap* que permite generar un *sitemap* XML o mapa del sitio para favorecer la tarea de los robots de Google y mejorar la indexación de las páginas. Durante la fase de instalación de *plugins* ninguno de estos soportaban WordPress 3.0 con la opción de *multisite* activada por lo que su instalación pasa a formar parte de líneas futuras.

Cuando se crea un *blog* todas sus opciones contienen los valores por defecto de WordPress y será el administrador del sistema o del *blog* el que tendrá que configurarlo con los valores deseados. *New Blogs Default* permite que el administrador del gestor de blogs defina la configuración que tendrán los nuevos *blogs*, de tal forma que el propietario del blog no necesitará preocuparse de la configuración del mismo aunque siempre podrá modificarla si lo cree necesario.

Además de los *plugins* ya mencionados, se han seleccionado otra serie de *plugins* que añaden herramientas útiles al gestor de *blogs*.

- *Thumbnail Viewer* inserta un JavaScript para mostrar de una forma más amigable las imágenes ampliadas.
- *TinMCE Advanced* soluciona un problema del editor visual de WordPress, el cual no permite insertar más de un salto de línea entre párrafos.
- *WP-Polls* permite crear encuestas y mostrarlas en forma de *widget* o de página.
- *Simple Tags* mejora la gestión de etiquetas.
- *WP Latex* permite escribir ecuaciones tan lo complicadas que se desee utilizando el lenguaje LaTeX.
- *BM Custom Login* permite personalizar el aspecto del formulario de acceso genérico de WordPress, así se puede adaptar a la imagen corporativa de GRIAL.

Por último existen una serie de *plugins* que permiten importar en WordPress contenidos de otras herramientas. Hasta la versión 2.9 esta funcionalidad formaba parte del núcleo pero en la versión 3.0 se requiere la instalación de cada uno de los *plugins* de importación, *WordPress Importer*, *Blogger Importer*, *RSS Importer*, *LiveJournal Importer*, *Categories to Tags Converter Importer*, *OPML Importer* y *Movable Type or TypePad Importer*.

De los *plugins* anteriormente mencionados formarán parte del núcleo del gestor los siguientes:

- *AddThis.*
- *Ajax Calendar.*
- *Akismet.*
- *Blogger Importer.*
- *BuddyPress.*
- *Categories to Tags Converter Importer.*
- *Display Widget.*
- *LiveJournal Importer.*
- *Movable Type or TypePad Importer.*
- *New Blogs Default.*
- *OpenID.*
- *OPML Importer.*
- *RSS Importer.*
- *SI CAPTCHA Anti-Spam.*
- *Sidebar Login.*
- *Simple tags.*
- *SlideShare.*
- *Thumbnail viewer.*
- *TinMCE Advanced.*
- *Viper's Video Quicktag.*
- *WordPress importer.*
- *WP-Issuu.*

6.5.3. Multisite Global Search

El origen de este *plugin* para WordPress se remonta al desarrollo de Diarium, el gestor de blogs de la USAL. En aquel momento surgió la necesidad de proporcionar un sistema de búsqueda a través de todos los *blogs* de la instalación de WPMU. Se realizó una exhaustiva búsqueda de *plugins* que implementaran esta funcionalidad. Tan solo un par de *plugins* desactualizados decían cubrir ese agujero funcional de WPMU, pero ninguno de ellos funcionaba correctamente. Por ello se tomó la decisión de desarrollar uno propio y ponerlo a disposición de la comunidad, cubriendo así este aspecto tan importante.

Surge así la primera versión de *WPMU Global Search*, un *plugin* para WPMU que permite buscar términos a través de las publicaciones de todos los blogs de una instalación. (Ver Figura 54).



Figura 54. Uso de WPMU Global Search en Diarium

Para explicar su funcionamiento primero es necesario explicar cómo está estructurada en la base de datos las diferentes publicaciones de los diferentes *blogs*. En WPMU por cada *blog* se crean un conjunto de tablas en la base de datos que almacenan las publicaciones, los enlaces, las opciones de configuración, etc. de tal forma que las publicaciones de blogs distintos se encuentran en tablas distintas. Esto implica que cada vez que se desee buscar un término en todos los blogs haya que recorrer todas las tablas buscando ese término.

Se plantearon dos soluciones a ese problema, crear una única tabla en la que mantener una copia de cada uno de las publicaciones con los problemas de consistencia de los datos que eso acarrea. O definir una vista que contenga todas las publicaciones, de tal forma que cada vez que se realice una búsqueda se trabajará con la publicación original, evitando tener que mantener copias.

Se optó por la segunda opción, a pesar de que obligara a utilizar una versión de MySQL superior a la que se indica en los requisitos mínimos de WPMU.

De esta forma, cada vez que un usuario introduce un término en el formulario de búsqueda, se comprueba dicho término contra la vista definida durante la instalación del *plugin*, mostrando los resultados obtenidos en una página definida especialmente para ello en el blog donde se encuentra insertado el formulario.

A día de hoy el *plugin* sigue instalado en Diarium pero por causas ajenas a nosotros no funciona, así que la mejor forma de probarlo es descargarlo del repositorio de WordPress donde se encuentra a disposición de la comunidad la versión más actual, con un total de mil seiscientas descargas como se ve en la Figura 55, y la posibilidad de descargar las versiones anteriores. (<http://wordpress.org/extend/plugins/wpmu-global-search>).

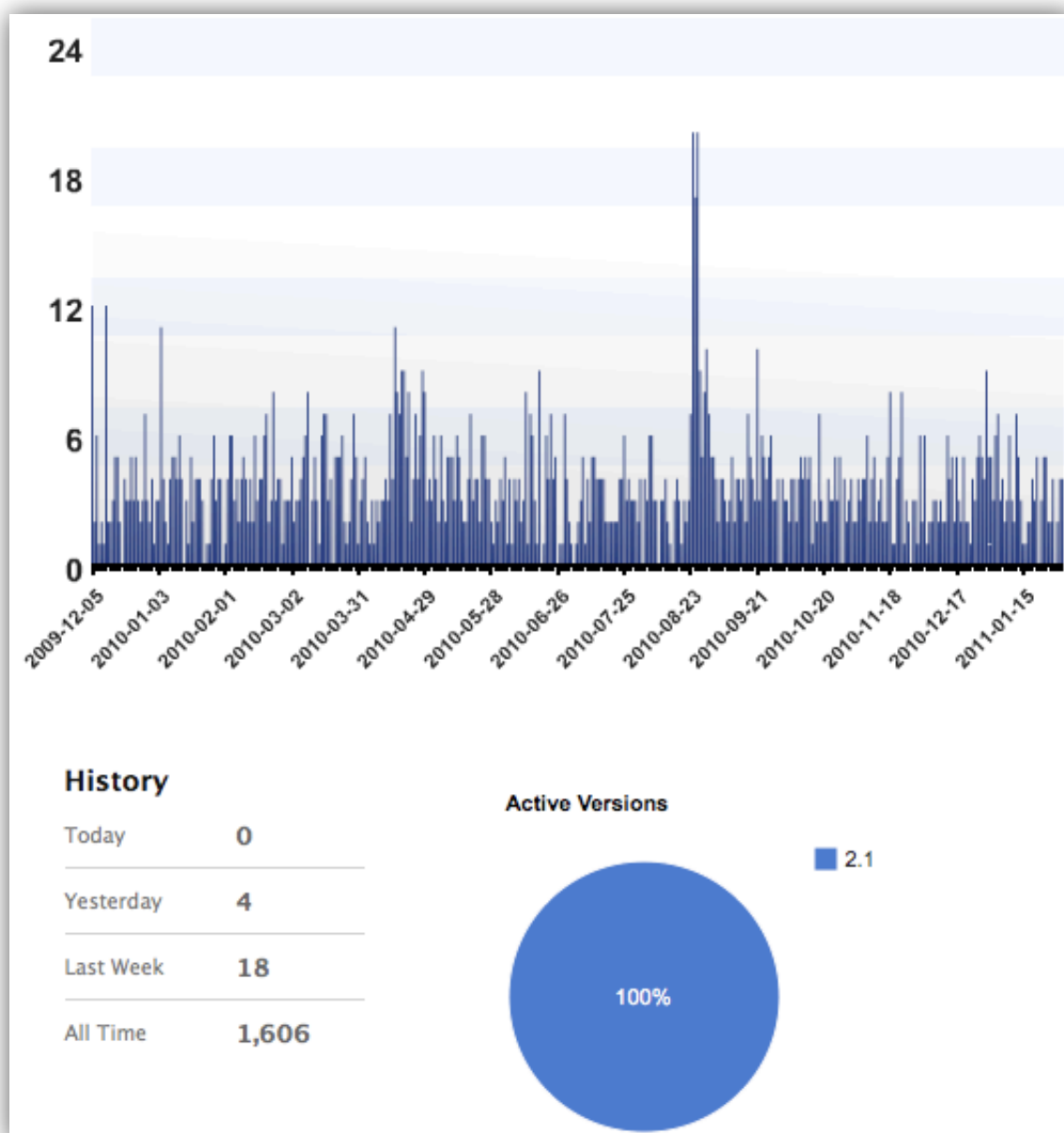


Figura 55. Estadísticas de descargas de WPMU Global Search
(<http://wordpress.org/extend/plugins/wpmu-global-search/stats>)

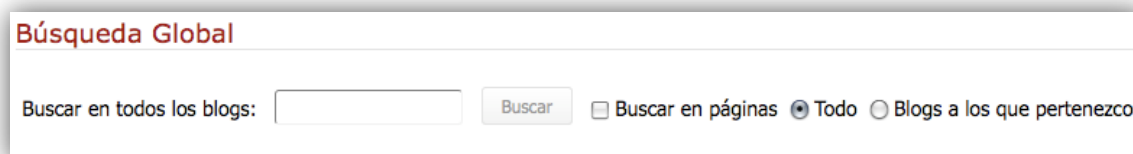
Cuando se montó el gestor de *blogs* para GRIAL se utilizó este *plugin* para proporcionar búsquedas globales. El problema vino a la hora de migrar de WPMU a WordPress 3.0, el *plugin* no funcionaba correctamente y necesitaba actualizarse. Se decidió realizar un desarrollo paralelo al *plugin* original, de tal forma que aquellos usuarios que todavía utilicen WPMU puedan seguir disponiendo de un *plugin* de estas características.

El nuevo *plugin* se denominó *Multisite Global Search* y está implementado para poder utilizarse únicamente sobre WordPress con la opción de *multisite* activada. No tiene sentido proporcionar un método de búsqueda para *blogs* individuales cuando ya existe la potente herramienta de búsqueda dentro de un *blog* individual que proporciona WordPress o la que proporciona Google mediante otro *plugin*.

Multisite Global Search continúa realizando las búsquedas mediante una vista creada al ser activado el *plugin*, pero se han añadido ciertas mejoras, algunas de ellas sugeridas por los usuarios.

El nuevo *plugin* posee las siguientes características:

- Disponible inicialmente en inglés y en español con posibilidad de que los usuarios generen traducciones a su idioma y las envíen para añadirlas en futuras versiones. Gracias a Sam Savinov el *plugin* dispone del fichero de idioma en Ruso.
- Búsqueda en las publicaciones y también en las páginas de todos los *blogs* del sistema.
- Limpieza de la base de datos cuando el *plugin* es desactivado.
- Dos tipos de formulario, uno para espacios horizontales y otro para espacios verticales tal y como se muestran en la Figura 56 y en la Figura 57.



Búsqueda Global

Buscar en todos los blogs: ☐ Buscar en páginas ☒ Todo ☐ Blogs a los que pertenezco

Figura 56. Formulario horizontal



Búsqueda Global

Buscar en todos los blogs:

☐ Buscar en páginas

☒ Todo ☐ Blogs a los que pertenezco

Figura 57. Formulario vertical

- El formulario y la página de resultados son personalizables a través de una hoja de estilo, CSS, propia.
- Posibilidad de insertar el formulario de búsqueda directamente en el código de un *theme* de WordPress.

```
Multisite_Global_Search::ms_global_search_vertical_form
    (your_results_page)

Multisite_Global_Search::ms_global_search_horizontal_form
    (your_results_page)
```

- Posibilidad de insertar el formulario de búsqueda en publicaciones o páginas mediante el uso de un *shortcode*, una cadena predefinida escrita entre corchetes

que WordPress identifica y sustituye por código PHP previamente definido. Mediante atributos se puede configurar el tipo de formulario y el URI de la página de búsqueda.

```
[multisite_search_form]
[multisite_search_form type="vertical"]
[multisite_search_form type="horizontal"]
[multisite_search_form page="multisite-search"]
```

- *Multisite Global Search widget* que muestra el formulario de búsqueda en uno de los *sidebars* del blog. Permite personalizar el URI para la página de resultados. Ver Figura 58.

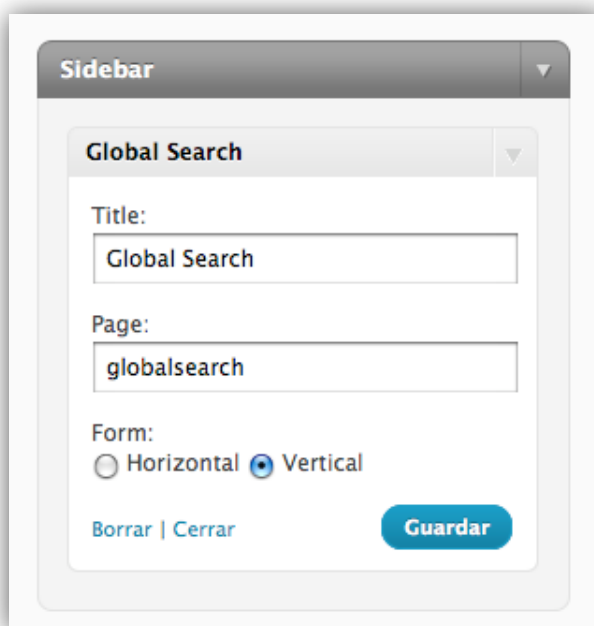


Figura 58. Configuración del *widget Multisite Global Search*

- Configuración de una página donde mostrar los resultados insertando únicamente el siguiente *shortcode*. Permite seleccionar si mostrar las publicaciones enteras o solo un extracto de las mismas.

```
[multisite_search_result excerpt="yes"]
```

- Si el usuario ha iniciado sesión en el sistema se muestra la posibilidad de buscar únicamente en aquellos blogs en los que participe de alguna manera.

Multisite Global Search ha tenido cierto éxito entre la comunidad de WordPress, alcanzando un total de dos mil descargas en el mes de enero, cinco meses después de publicar la versión 1.0 (Ver Figura 59). Actualmente el *plugin* se encuentra en la versión 1.2.4 gracias al feedback recibido tanto en su página oficial como en la de WordPress.

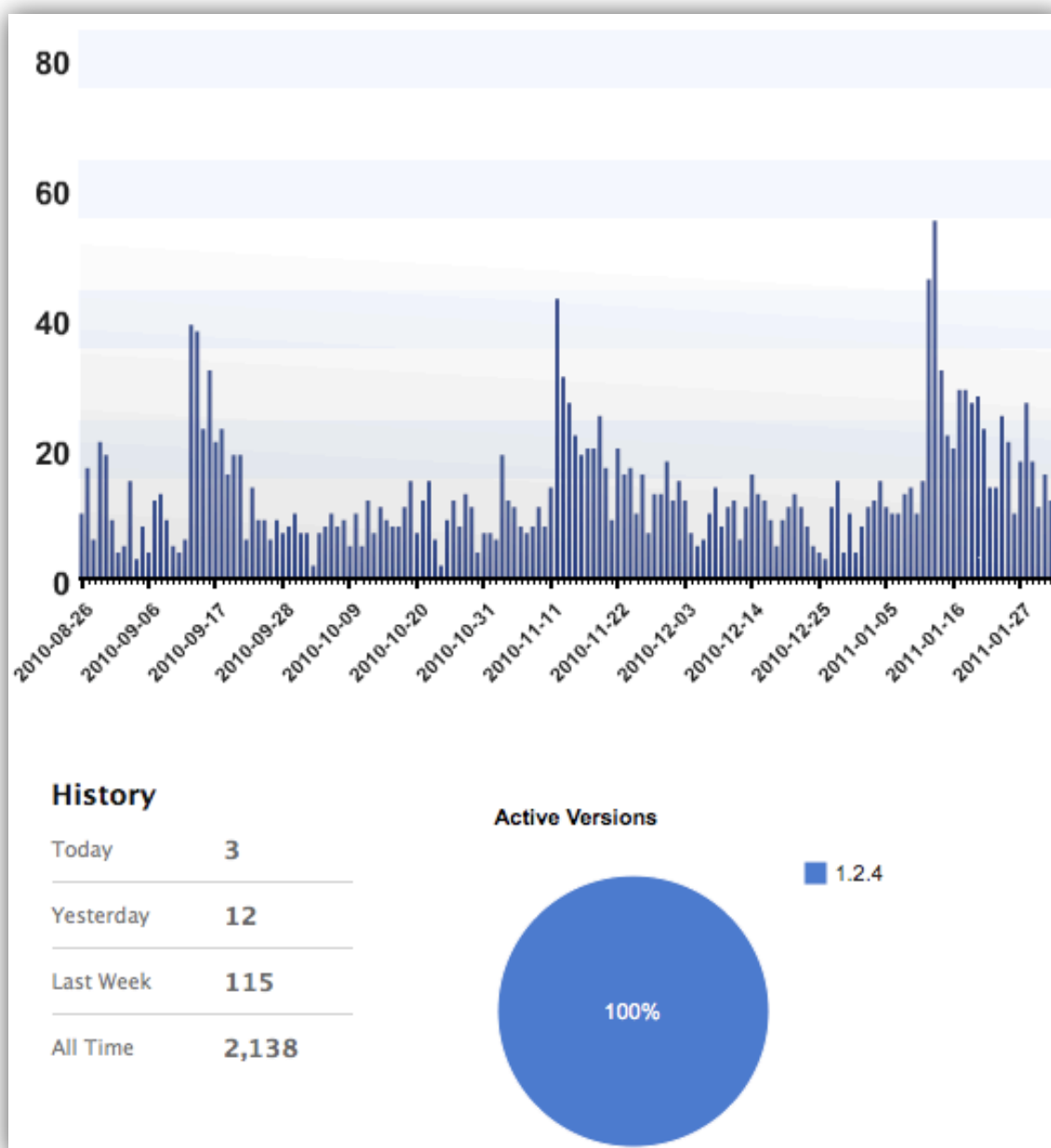


Figura 59. Estadísticas de descargas de *Multisite Global Search*
 (<http://wordpress.org/extend/plugins/multisite-global-search/stats>)

Algunos ejemplos de cómo se utiliza el *plugin* así como diferentes maneras de personalizar el aspecto de los resultados y el formulario pueden verse en las siguientes webs:

- BlogsMX. <http://blogsmx.com/>.
- OBN. Outdoor Blog Network. <http://www.outdoorblog.net>.
- Terrapinn blogs. <http://blogs.terrapinn.com/> (Figura 60).

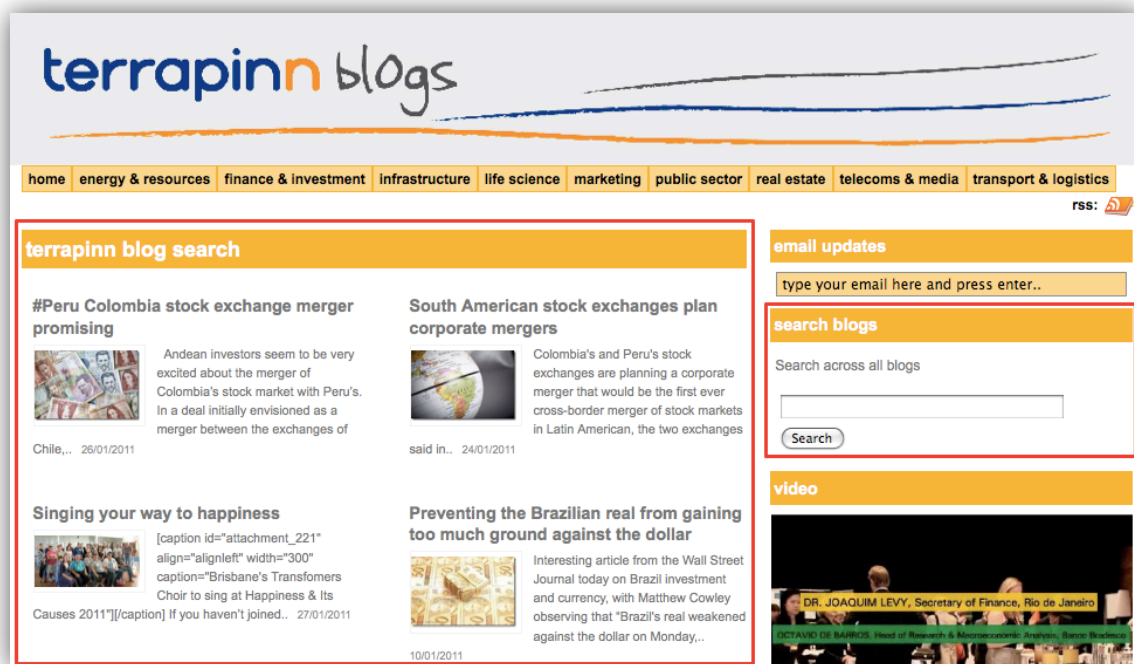


Figura 60. Multisite Global Search en Terrapinn blogs

- Trip Advisor Diary. <http://www.tripadvisoridary.com/>.
- Otis Blogosphere Community. <https://blogs.otis.edu/>.
- MVN – Most Valuable Network. <http://mvn.com/>. (Figura 61).

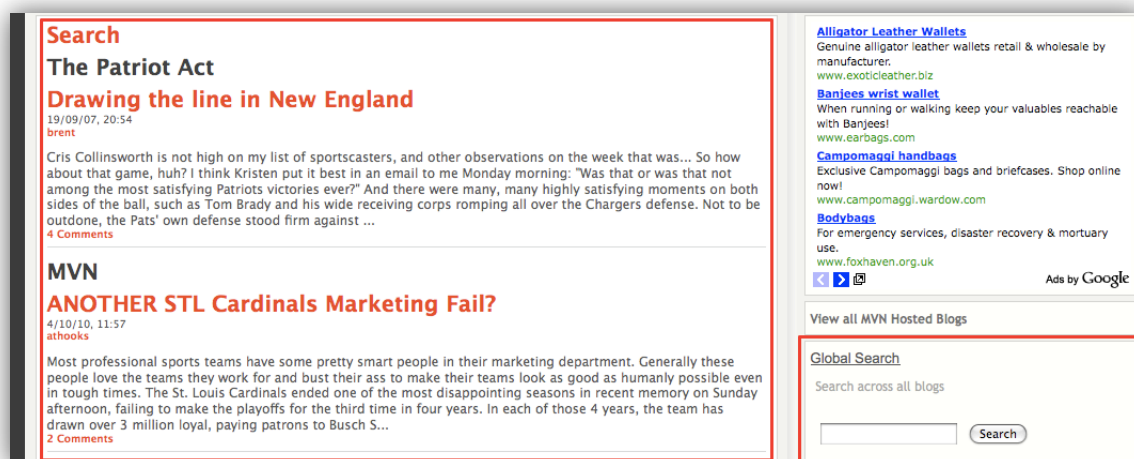


Figura 61. Multisite Global Search en MVN - Most Valuable Network

- Proyecto Lunar – Andalucía. <http://www.smaland.es/mu/>
- Blogs Comunitat Valenciana. <http://blogs.comunitatvalenciana.com/>. (Figura 62).



Figura 62. Multisite Global Search en los Blogs Comunitat Valenciana

- Blog central de Centros Educativos - Gobierno de Canarias. <http://www2.gobiernodecanarias.org/educacion/clicescuela20/edublogs/>. (Figura 63).



Figura 63. Multisite Global Search en el blog central de Centros Educativos de la Comunidad de Canarias

6.5.4. *BuddyPress Activity Tags*

Se trata de un *plugin* que permite mostrar un *widget* o bloque con las etiquetas de las nuevas publicaciones mostradas en la pestaña *Actividad* de *BuddyPress* (<http://grial.usal.es/agora/activity/>). Es una forma de proporcionar una nube de etiquetas de las publicaciones de todo el sistema.

Al disponer de un portal de acceso al gestor de blogs es importante darle que en él se de visibilidad a los *blogs* que conforman el sistema. Permitir búsqueda global o en este caso crear una nube con las etiquetas más usadas en todos los blogs, cumple ese objetivo.

Esta funcionalidad no había sido implementada hasta el momento, y mucho menos de la forma en que se ha llevado a cabo.

Contar las veces que se han utilizado todas etiquetas del sistema cada vez que se carga el *widget* supone alto consumo de tiempo en cada consulta a la base de datos lo que ralentizaría el correcto funcionamiento de todo el sistema. Para poder buscar las etiquetas es necesario obtener cada una de las publicaciones existentes en los diferentes blogs del sistema y luego obtener sus las etiquetas para posteriormente realizar un conteo. Frente a este problema se barajó la posibilidad de utilizar vistas como en el *plugin* de búsqueda, pero al tener que incluir las etiquetas la cosa se complicaba. Entonces se decidió cambiar el enfoque del *plugin*, de tal forma que seguiría mostrando una nube de etiquetas pero de las publicaciones que *BuddyPress* almacena como actividad reciente. Ese conjunto de publicaciones, independientemente del blog al que pertenezcan, están en la base de datos todas juntas en una tabla que utiliza *BuddyPress* por lo que para generar la nube de etiquetas tan solo hay que recorrer esa tabla obteniendo las etiquetas utilizadas en cada publicación. Luego se realiza el conteo de cada etiqueta y se muestran en la nube las más utilizadas, variando el tamaño y el color dependiendo de si aparece en más publicaciones o en menos. Puede verse un ejemplo en la Figura 64.



Figura 64. Nube de etiquetas generada por *BuddyPress Activity Tags*

Al tratarse de un *plugin* mucho más específico por requerir la instalación de *BuddyPress*, su uso es más puntual. Aun así en los cinco meses que lleva disponible ha recibido un total de alrededor setecientas descargas como se puede ver en la Figura 65.

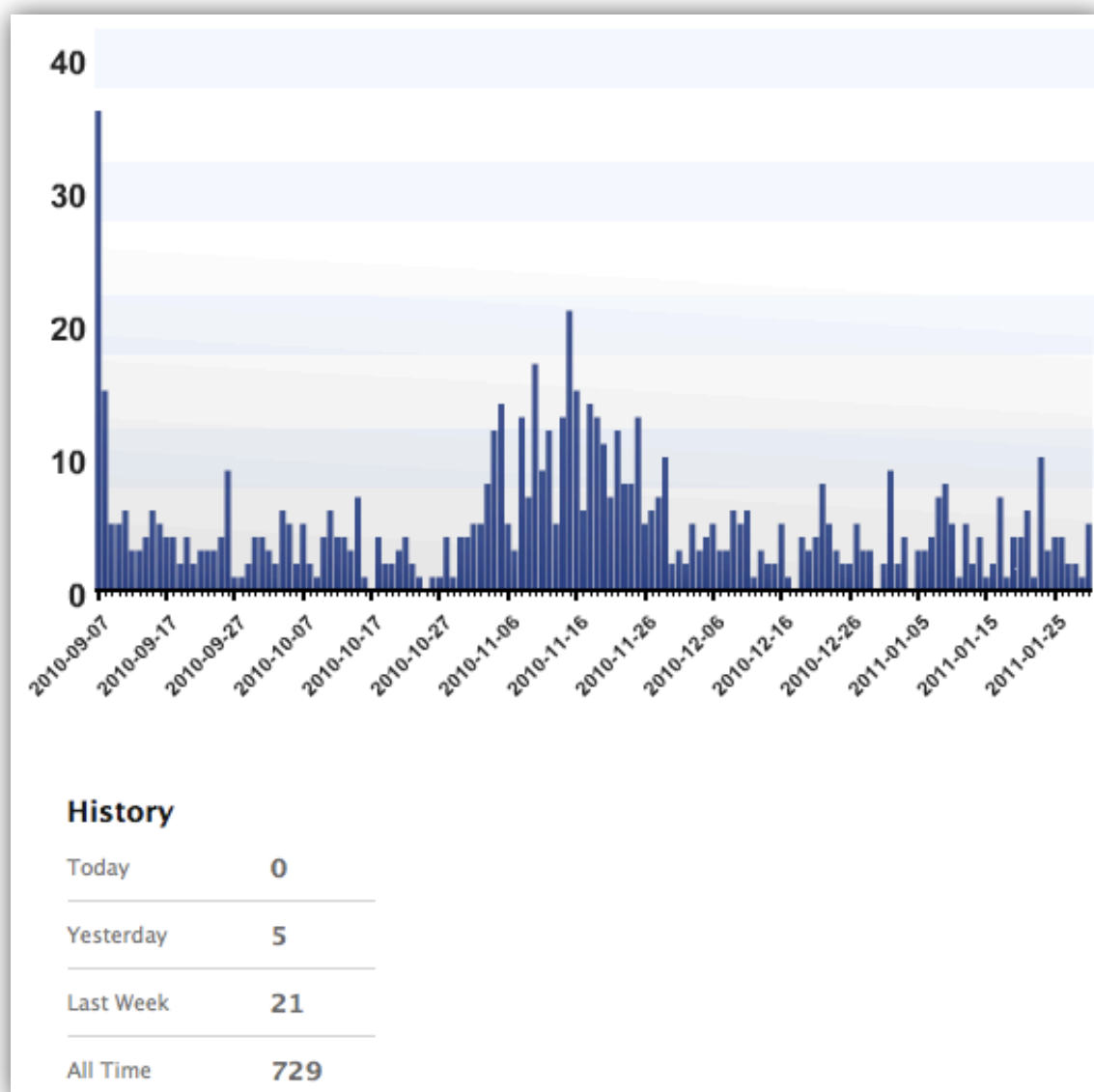


Figura 65. Estadísticas de descargas de *BuddyPress Activity Tags*
(<http://wordpress.org/extend/plugins/buddypress-activity-tags/stats>)

Algunas de las webs que hacen uso de este *plugin*:

- Planet Aveyron. <http://www.fortetvert.com/>. (Figura 66)

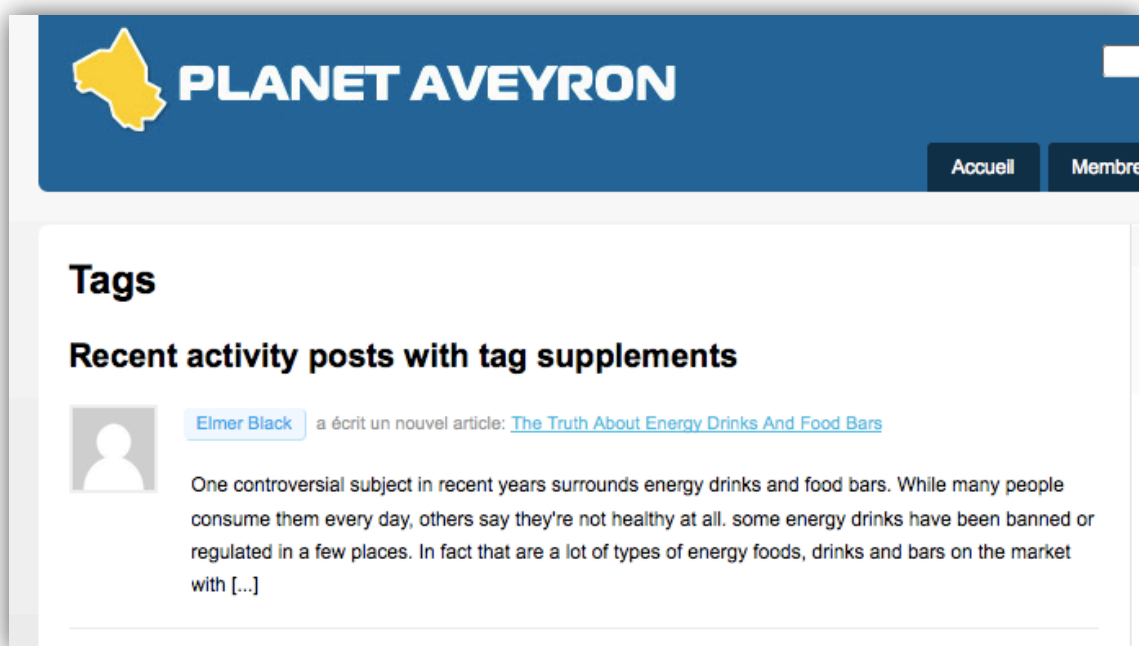


Figura 66. *BuddyPress Activity Tags* en Planet Aveyron

- Millenium Project Nodes. <http://mpnodes.info/>. (Figura 67).



Figura 67. *BuddyPress Activity Tags* en Millenium Project Nodes

6.5.5. WordPress theme. La imagen corporativa de GRIAL

WordPress tiene un sistema de temas que permiten separar la parte visual del *blog* de la parte funcional. El contenido de los *blogs* se genera de forma dinámica, es decir, no

existe una página por cada *post*, y para poder mostrar dicho contenido se utilizan un conjunto de plantillas que es lo que se denomina tema o *theme* de WordPress.

Las plantillas que conforman un tema son ficheros PHP que combinan funciones propias del lenguaje con funciones especiales de WordPress para conseguir generar dinámicamente las diferentes secciones de un *blog*. Además de las plantillas, un tema está compuesto por ficheros CSS u hojas de estilo que permiten definir el diseño de las plantillas, imágenes en diversos formatos y ficheros JavaScript.

Existen miles de temas *open source* pero para mantener el *look & feel* de GRIAL ha sido necesario el desarrollo de un tema propio que cubriera todas las necesidades del gestor de blogs. El tema de GRIAL tiene las siguientes características:

- Multilingüe.
- Soporte para menús. Ver ejemplo en la Figura 68.

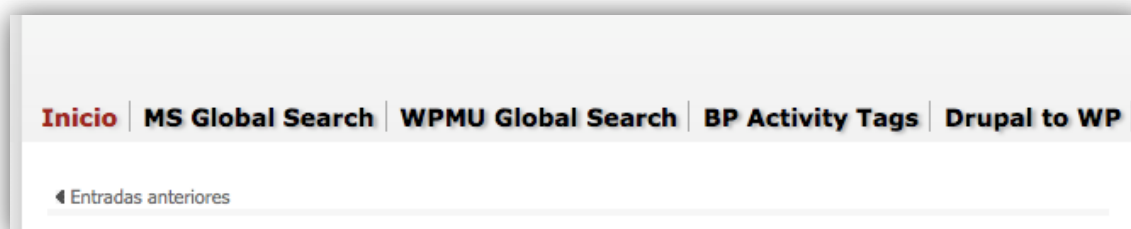


Figura 68. Ejemplo de menús en el *theme* de GRIAL

- Soporte para *widgets* mediante la definición de tres *sidebars* que se muestran en la Figura 69.

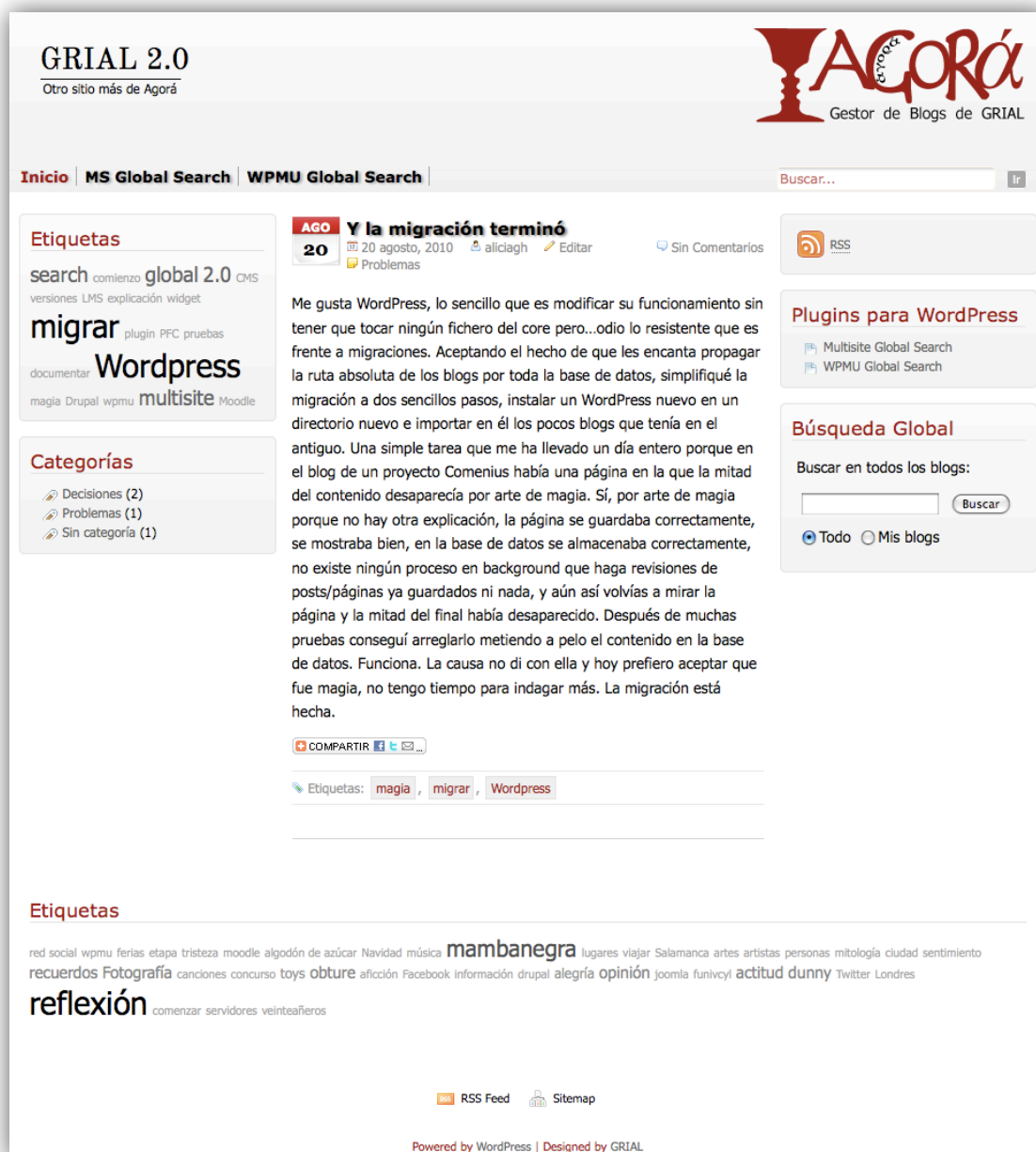


Figura 69. Sidebars posibles en el *theme* GRIAL para WordPress

- JavaScript para responder y citar comentarios. Figura 70.

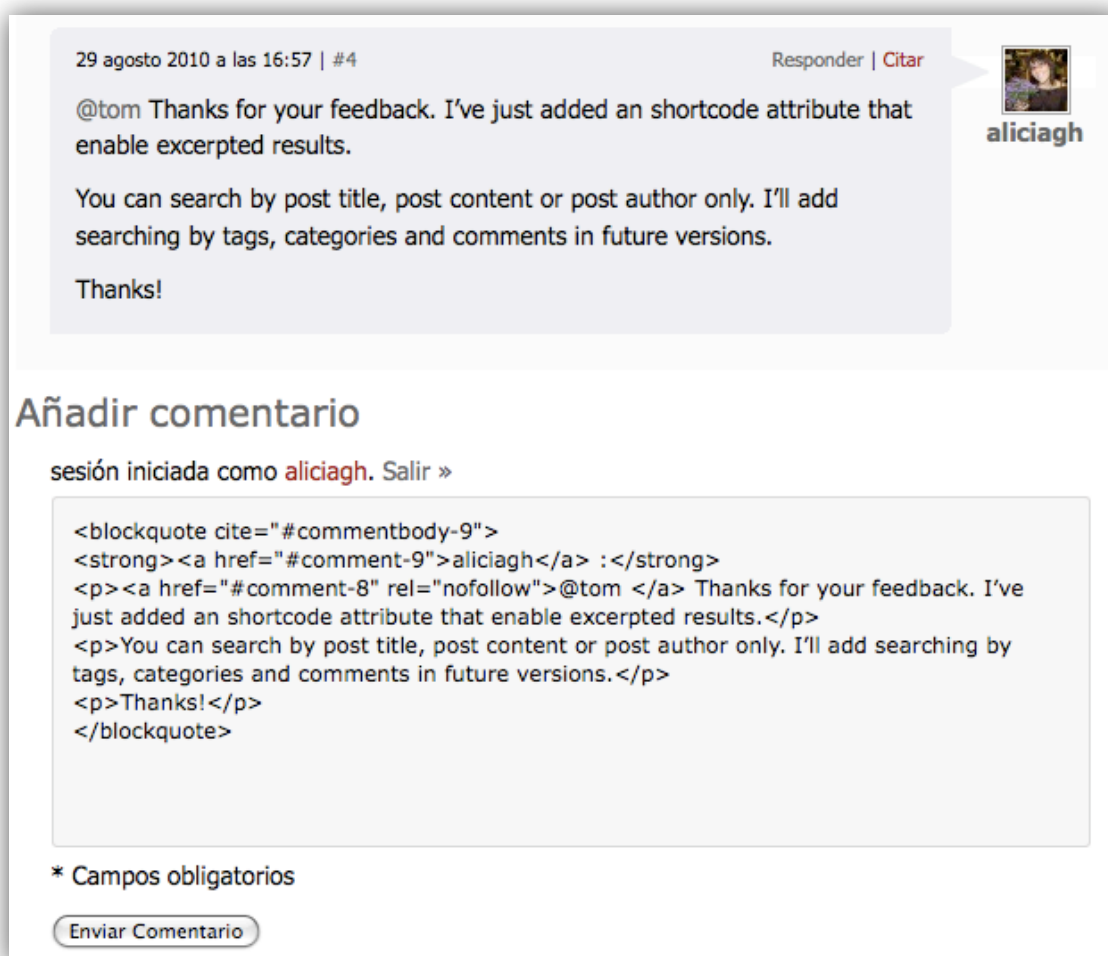


Figura 70. Funciones de los comentarios en el *theme* de GRIAL

- Compatible con Gravatar.
- Plantillas de página.
- Página que muestra las publicaciones de una categoría concreta.
- Página sin metadatos ni comentarios.
- Página de opciones.
- Uso de Google Font API. Old Standard TT en la Figura 71.

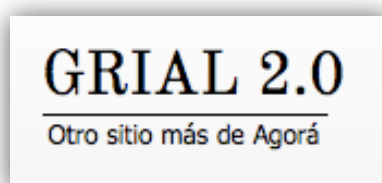


Figura 71. Ejemplo de uso de Google Font API

- Posibilidad de añadir código de *web analytics*.
- Posibilidad de utilizar el motor de búsqueda de Google dentro del *blog*.

6.6. PÓLIS. El campus *online* de GRIAL

6.6.1. *User categories*

Poco a poco el número de cursos a los que se tiene acceso en el Moodle que se utiliza como plataforma de enseñanza, ya sea como creadores, profesores o estudiantes, se va incrementando, siendo cada vez más difícil encontrar el curso deseado. Por este motivo se ha desarrollado un módulo denominado *User Categories* que introduce un sistema de categorías personales cuya finalidad es permitir que cada usuario organice los cursos que le aparecen en “Mi Moodle” de la forma que mejor le convenga (Ver Figura 72). La herramienta permite crear, modificar y eliminar categorías, manteniendo siempre una, denominada “Sin categoría” o “*No category*”, en la que se agruparán los cursos que no se asignen a ninguna otra creada por el usuario.

Para mejorar aún más la visibilidad dentro de Moodle, se ha incorporado un botón cuya función es ocultar o mostrar la actividad reciente correspondiente a cada curso. El usuario podrá elegir si desea un listado sencillo de los cursos o ver una información más detallada de lo que ha ocurrido en cada uno de ellos durante su ausencia.

El otro botón que se añade en la parte superior es el botón que permite gestionar las categorías personales.

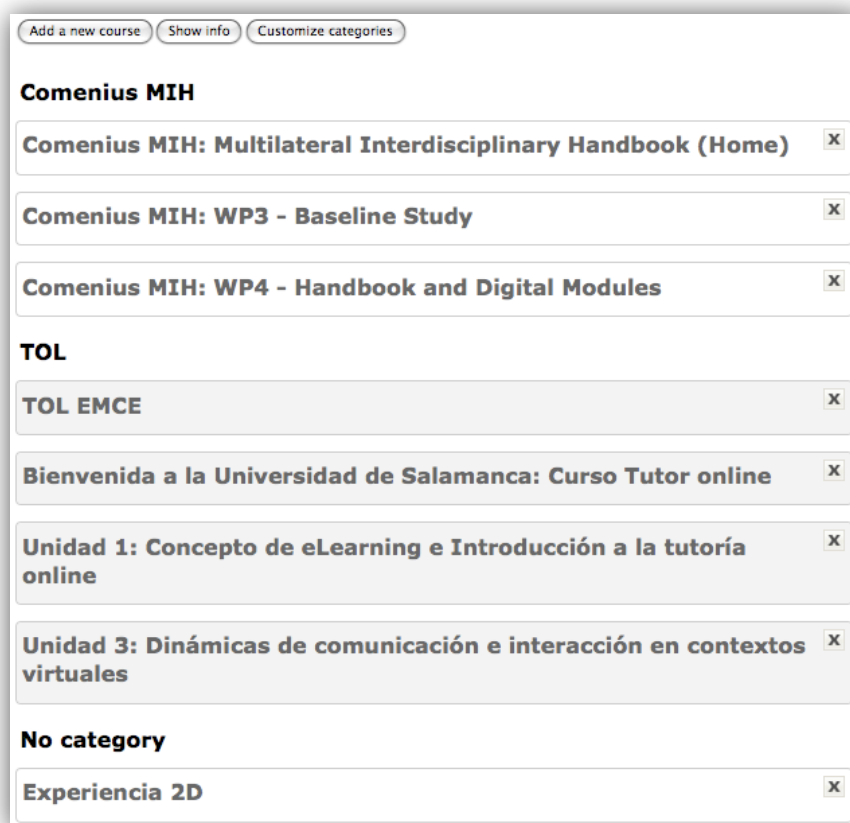


Figura 72. Ejemplo de utilización del módulo *User Categories* para Moodle

6.6.2. Estadísticas del curso

Se trata de un bloque que únicamente se visualiza dentro de los cursos y que pone a disposición del usuario, tanto si es alumno como si es profesor, información relevante del curso en el que se encuentra. Los datos que se muestran son:

- Número de entradas existentes en cada uno de los foros.
- Gráfica que muestra la actividad diaria del curso en el último mes.
- Para cada tarea creada se muestra el número de alumnos que la han enviado y el número de estas que han recibido una calificación por parte del profesor.
- Número total de usuarios con acceso al curso.
- Número total de recursos creados.
- Número total de entradas en cada uno de los glosarios del curso.
- Número total de actividades creadas.



Figura 73. Ejemplos del bloque de estadísticas de un curso de Moodle

La generación de la gráfica mediante la *API de Google Chart*, Figura 73, se realiza a partir de datos recopilados cada noche en una tabla de la base de datos que se crea al instalar el bloque. Dicha tabla contiene el número de interacciones que han ocurrido

cada día en un curso. Si cada vez que se carga el bloque calculáramos la actividad del curso de manera dinámica la base de datos sufriría bastante y ralentizaría el sistema ya que el número de interacciones se obtiene de la tabla de *log* cuyo tamaño crece diariamente en cientos de entradas. Realizando el cálculo en las horas que la plataforma recibe menos tráfico el número de usuarios afectados es mínimo y los datos obtenidos son fiables ya que una vez terminado el día el número de interacciones realizadas en el curso no varía con el tiempo.

6.6.3. Informe y bloque de estadísticas globales



Figura 74. Ejemplo del bloque de estadísticas globales de Moodle

A veces es interesante que los usuarios conozcan algunos datos estadísticos acerca del uso que se le está dando a la plataforma de enseñanza. Existen un gran número de variables que pueden ser calculadas y mostradas pero se ha optado por mostrar datos muy generales que sean comprensibles por los usuarios sin recargar demasiado la interfaz. Los datos que se proporcionan se pueden ver en la Figura 74 y son:

- El número total de cursos que hay en el sistema.
- La cantidad de usuarios registrados.
- El número de usuarios que ejercen como docentes en alguno de los cursos.
- El número de usuarios que ejercen como estudiantes en alguno de los cursos.
- Cuántos recursos existen en el sistema. Se consideran recursos las etiquetas, las páginas de texto, las páginas web compuestas dentro de un curso, los enlaces a ficheros o web externas, los contenidos IMS y los directorios.

Este bloque permite presentar dichos datos de forma bastante visual, con gráficas generadas mediante la *API de Google Chart*. Al contrario que en el bloque de estadísticas del curso, las estadísticas globales se calculan dinámicamente cuando se carga el bloque ya que las consultas que se realizan a la base de datos son rápidas y tienen muy poca carga para el sistema.

Además del conjunto de datos existen otros datos de interés para los administradores de la plataforma que se muestran en un informe de estadísticas globales añadido al sistema como un módulo de Moodle. En dicho informe se irán incluyendo gráficos y datos que sean relevantes a la hora de gestionar la plataforma aunque de momento se muestren

únicamente los mismos datos que en el bloque como puede verse en la Figura 75. Por ejemplo, en Studium este módulo incorpora una serie de tablas que permiten analizar el impacto de la plataforma en los diferentes centros que conforman la Universidad.

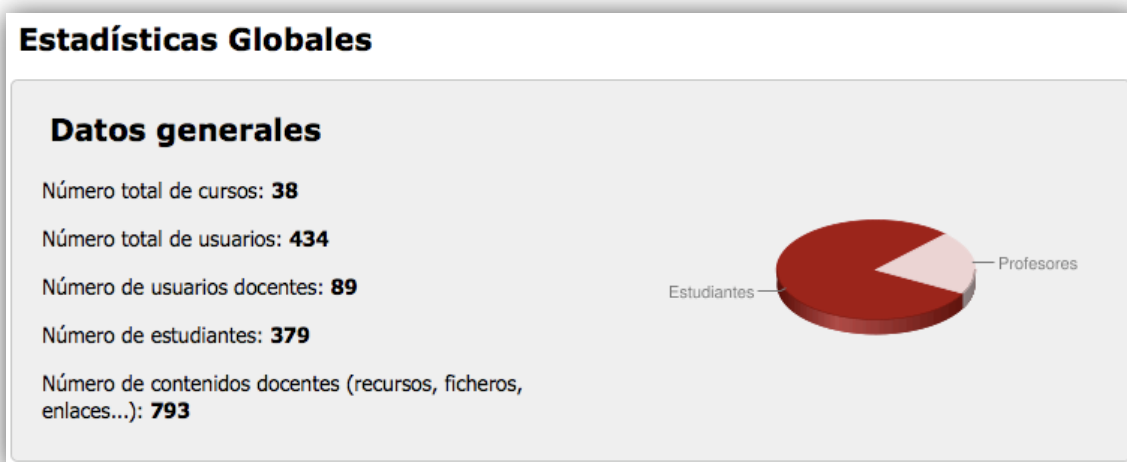


Figura 75. Informe estadísticas globales de Moodle

6.6.4. Informe de actividad

Por último, se ha añadido otra página a la sección de informes cuya labor es mostrar la actividad diaria en la plataforma. Cada vez que en Moodle se lleva a cabo una acción, por ejemplo acceder a un curso, participar en un foro, dar de alta un usuario, queda registrado en la base de datos en una tabla de *log*.

Para medir la actividad diaria se utilizan los datos recopilados en la tabla de *log*, de tal forma que cada vez que se carga la página del informe se genera dinámicamente una tabla con la fecha y cantidad de actividad que se generó ese día (Ver Figura 76). Como ocurre en el bloque de estadísticas del curso, obtener datos de dicha tabla ralentiza el sistema ya que procesar una tabla con un número tan elevado de registros supone una gran carga. En este caso las cifras totales diarias no se guardan estáticamente en una nueva tabla, sino que se calculan cada vez que uno de los administradores accede al informe ya que se trata de una acción puntual, al contrario que las estadísticas de cursos que se muestran cada vez que un usuario acceda a un curso que tenga el bloque de actividad.

Fecha	Interacciones		
04.02.2010	9	18.11.2010	2746
06.02.2010	106	19.11.2010	2497
08.02.2010	5	20.11.2010	944
10.02.2010	10	21.11.2010	1331
11.02.2010	8	22.11.2010	3803
15.02.2010	3	23.11.2010	3473
17.02.2010	16	24.11.2010	3288
05.03.2010	2944	25.11.2010	2772
15.03.2010	100	26.11.2010	1507
17.03.2010	55	27.11.2010	822
18.03.2010	22		
19.03.2010	238		

Figura 76. Comienzo y fechas con cursos activos de la tabla de actividad

6.7. Miscelánea

6.7.1. Varias instancias de DSpace en un mismo servidor

La instalación de DSpace se ha llevado a cabo en un servidor en el que ya existía otra instancia de la aplicación. Al tratarse de una herramienta que requiere Tomcat en vez de Apache y la utilización de PostgreSQL en vez de MySQL, se tomó la decisión de mantener un único servidor configurado con dichas aplicaciones, por ser estas menos utilizadas en el entorno de desarrollo de GRIAL.

Existen un gran número de tutoriales que explican cómo realizar la instalación de DSpace, tras seguir unos sencillos pasos y después de tener instaladas las aplicaciones necesarias, DSpace funciona correctamente. Cuando se habla de instalar varias instancias la cosa cambia, no existe demasiada información de cómo realizar el proceso de instalación para que todas las instancias funcionen correctamente.

Tras varios intentos desafortunados se encontró un tutorial (<http://kmtk.cs.ait.ac.th/knowledge-center/how-to/install-many-dspace-instances-on-one-tomcat-server>) que adaptándolo permitía finalizar la instalación satisfactoriamente.

Los pasos seguidos, utilizando DSpace 1.6.2 y Tomcat 6.0.20, han sido los siguientes. [dspace-source] se refiere a /opt/dspace-1.6.2-src-release.

1. Crear el usuario 'dspace' si no existe ya.

```
sudo useradd -m dspace
sudo passwd dspace
```

2. Crear un directorio por cada instancia de DSpace que se vaya a instalar. No es necesario que las instancias se encuentren en /opt, vale cualquier otro directorio donde se tengan permisos de escritura.

```
sudo mkdir /opt/dspace-2
sudo chown dspace /opt/dspace-2
sudo chmod 777 /opt/dspace-2
```

3. Se crea el usuario 'dspace' en 'PostgreSQL' y una base de datos por cada instancia que se vaya a instalar.

```
sudo su postgres
createuser -U postgres -d -A -P dspace
createdb -U dspace -E UNICODE dspace-2
```

4. Para cada instancia se debe definir un fichero de configuración en [dspace-source]/dspace/config. Para ello se duplica el fichero por defecto dspace.cfg.

```
sudo cp [dspace-source]/config/dspace.cfg [dspace-source]/config/dspace-2.cfg
```

5. Ahora se edita cada fichero de configuración y se cambian las siguientes líneas. Debe indicarse el puerto en el que esté configurado Tomcat.

```
dspace.dir = /opt/dspace-2
dspace.hostname = http://your-dspace-url
dspace.baseUrl = http://your-dspace-url:443
dspace.url = http://your-dspace-url:443/dspace-2
db.url = jdbc:postgresql://localhost:5432/dspace-2
```

```
db.username = dspace
```

```
db.password = contraseña-usuario-postgres-dspace
```

6. Si se desea configurar el envío de correo hay que editar las siguientes líneas del fichero de configuración. En el ejemplo se muestran los parámetros que deben utilizarse para el envío a través de Gmail.

```
mail.server=smtp.gmail.com
```

```
mail.server.username = usuario-de-gmail
```

```
mail.server.password = contraseña-de-gmail
```

```
mail.from.address = dirección-de-email
```

```
feedback.recipient = dirección-de-email
```

```
mail.admin = dirección-de-email
```

```
# Descomentar las siguientes líneas
```

```
mail.extraproperties = mail.smtp.socketFactory.port=465, \
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory, \
mail.smtp.socketFactory.fallback=false
```

7. Se establece el idioma por defecto y se habilitan el inglés y el español para poder ofrecer una interfaz multilingüe.

```
default.locale = es
```

```
webui.supported.locales = es, en
```

8. Para terminar la configuración se desactiva el registro automático de usuarios hasta que el sistema sea estable y se defina una política de mantenimiento del repositorio.

```
authentication.shib.autoregister = false
```

```
# La opción anterior no cumplía con su cometido así que se
habilitó el siguiente parámetro indicando un dominio que no
existe
```

```
# Este parámetro restringe el registro únicamente a aquellos
emails que terminan en los dominios indicados.
```

```
authentication.password.domain.valid = xxx.xxx
```

9. Acceder al sistema como el usuario 'dspace' y generar los paquetes de la instalación con maven.

```
cd [dspace-source]/dspace/
```

```
sudo su dspace
```

```
mvn package
```

10. Se instalan cada una de las instancias utilizando los ficheros de configuración definidos anteriormente.

```
cd [dspace-source]/dspace/target/dspace-[version].dir/
```

```
sudo su dspace ant -Dconfig=[dspace-source]/config/dspace-
2.cfg
```

```
fresh_install
```

11. Se añade una cuenta de administrador.

```
sudo -u dspace [dspace-instances]/dspace-2/bin/create-
administrator
```

12. Se configura Tomcat.

```
sudo vim [TOMCAT_HOME]/conf/server.xml

<!-- DEFINE A CONTEXT PATH FOR DSpace JSP User Interface -->
<Context path="/dspace-2" docBase="[dspace-instances]/dspace-2/webapps/jspui/" debug="0" reloadable="true"
cachingAllowed="false" allowLinking="true"/>
```

13. Por último, se reinicia Tomcat. Para comprobar que funciona correctamente basta acceder a la URL que se indicó en el fichero de configuración: `http://your-dspace-url:443/dspace-2`.

```
sudo /etc/init.d/tomcat start
```

Una vez instalada y configurada la aplicación correctamente se pasó a adaptar la interfaz a la imagen corporativa de GRIAL obteniendo como resultado el que se muestra en la Figura 77.

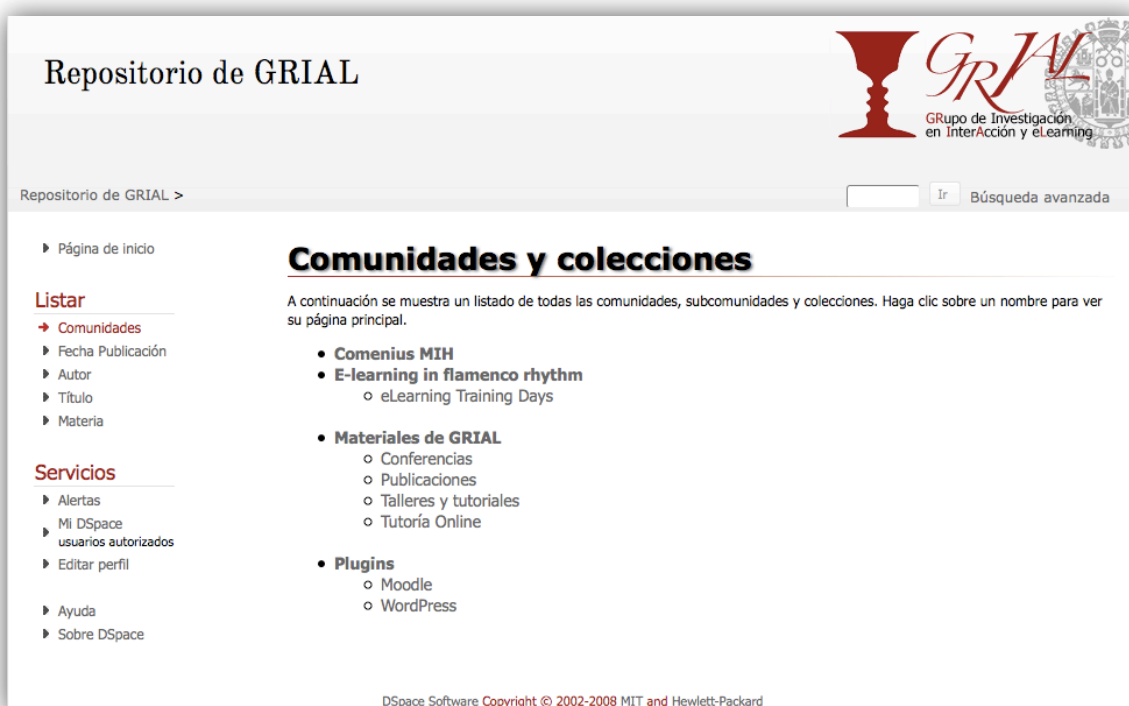


Figura 77. Imagen corporativa de GRIAL para DSpace

6.7.2. Diseminación de la información

Uno de los aspectos más importantes de GRIAL 2.0 es definir una política de diseminación de la información que permita difundir de manera sencilla y eficaz todo aquello que se realiza dentro del grupo de investigación.

Es fundamental que los miembros tengan que realizar el mínimo esfuerzo a la hora de llevar a cabo este cometido, ya que si se trata de un proceso complicado que requiera algo de tiempo al final casi nadie se encargará de realizarlo, perdiéndose así gran parte de su valor. Puedes argumentar, investigar, desarrollar todo lo que quieras pero de qué sirve si nadie sabe de su existencia.

En primer lugar se ha definido un esquema indicando qué conexiones iban a realizarse y entre qué elementos, puede verse en la Figura 78. El centro de las conexiones reside en la página de GRIAL en Facebook y en la cuenta de Twitter.

Todos los contenidos subidos a las diferentes aplicaciones web se publican automáticamente en Twitter, pero no todos lo que se publica en Twitter va a Facebook. Esta decisión se debe a que en muchos casos los mensajes o contenidos publicados en Twitter no tienen sentido dentro de Facebook, pudiendo llegar a ser molesto inundar a tus seguidores con mensajes de 140 caracteres que fuera del contexto de la herramienta de *microblogging* no son útiles.

Para conectar YouTube, Delicious, Flickr, SlideShare y Facebook con Twitter se han utilizado las diferentes herramientas de conexión que proporciona cada uno de estos servicios. Además para Flickr y SlideShare se ha configurado sus respectivas aplicaciones en Facebook para que se muestre su contenido en pestañas de la página de GRIAL. Estas aplicaciones no generan publicaciones en el muro pero si permiten que estos contenidos estén accesibles a través de Facebook de forma automática.

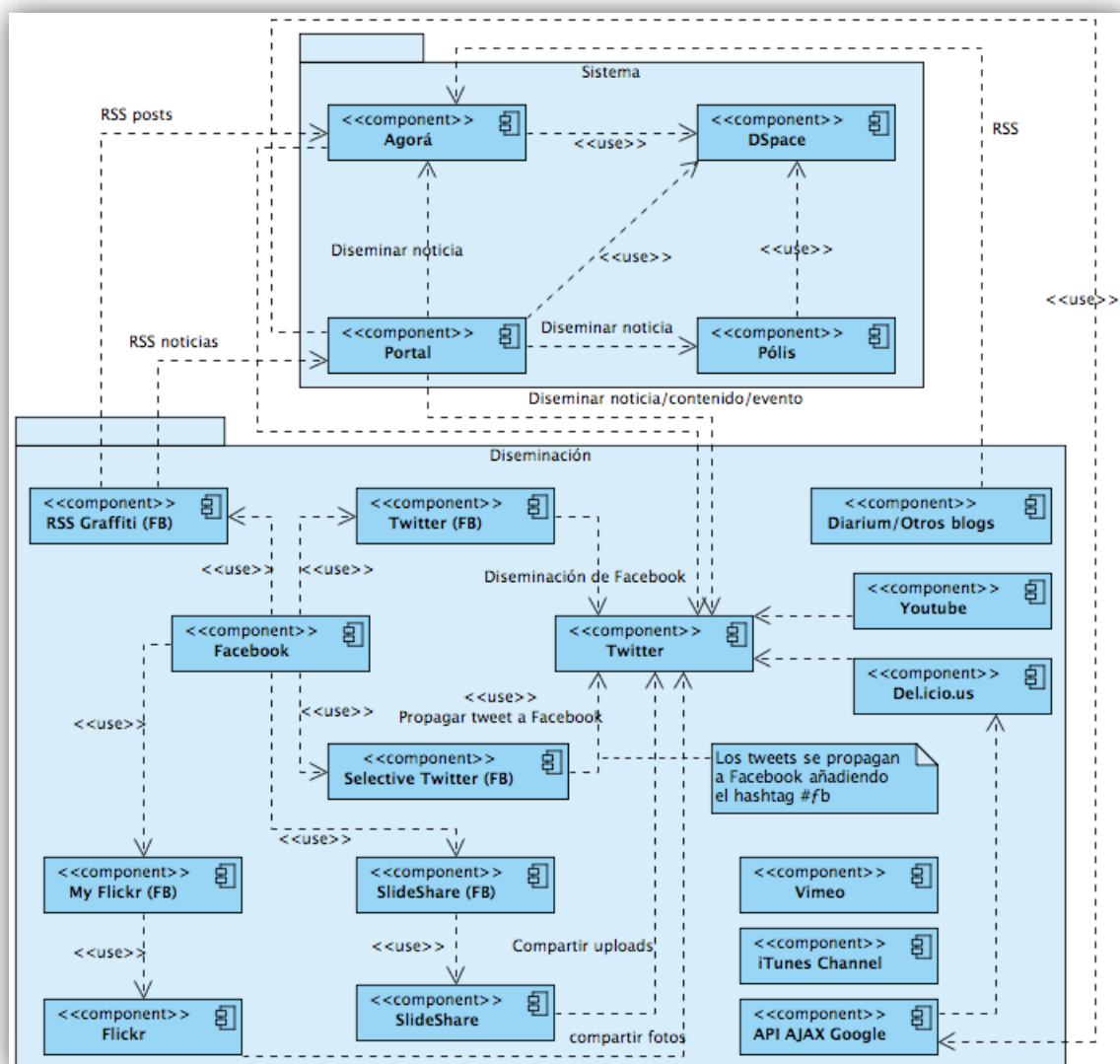


Figura 78. Diagrama de componentes de la diseminación de la información

También se ha configurado una conexión selectiva de Twitter a Facebook de tal forma que cuando se desee que un *tweet* se publique también en el muro de Facebook tan solo hay que añadirle el *hashtag* #fb al final.

Respecto al portal de GRIAL se han definido dos caminos. Todos los contenidos publicados en el portal, proyectos de investigación, cursos, noticias, eventos, etc. se envían automáticamente a Twitter en el momento de ser creados. Y por otro lado las noticias se publican en el muro de Facebook a través de la aplicación *RSS Graffiti* que chequea periódicamente el RSS del portal para ver si hay nuevos contenidos y en ese caso publicarlos automáticamente. Además, cuando se publica una noticia se da la posibilidad al usuario de crear una nueva publicación en el *blog* del grupo de investigación y/o una noticia en el muro de Moodle, con el mismo texto de la noticia o modificado.

El *blog* de GRIAL también está conectado automáticamente con Twitter y con Facebook de tal forma que cuando realizas una nueva publicación se envía un *tweet* a la cuenta GRIAL de Twitter y mediante la aplicación *RSS Graffiti* se publica en el muro de Facebook. El resto de *blogs* también pueden configurarse de la misma forma, automatizando así la difusión de los *posts*.

Se puede ver un esquema más claro de las conexiones entre aplicaciones en la Figura 79.

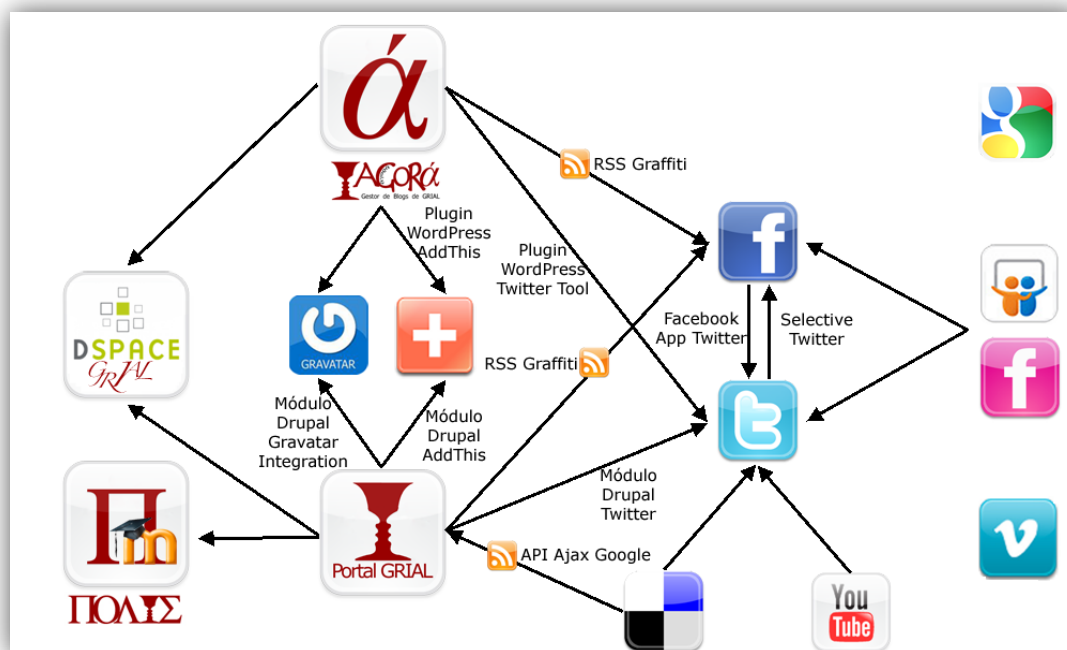


Figura 79. Diseminación de la información en GRIAL

Por último, pero no menos importante, se ha buscado una solución al problema de mantener el *blog* del grupo de investigación. Además de permitir que las noticias del portal se publiquen automáticamente en dicho *blog*, se ha configurado una red que lo abastece sin tener que escribir explícitamente en él. Esto se ha logrado utilizando el *plugin* de WordPress denominado *FeedWordPress*, un agregador de noticias Atom/RSS para WordPress que permite syndicar en un *blog* de WordPress noticias de las fuentes que se elijan. Dicho *plugin* se ha configurado en el *blog* principal de tal manera que cada 60 minutos comprueba si en los *blogs* de los miembros del grupo de investigación se ha realizado alguna publicación que contenga la etiqueta “Grial” o cualquiera de sus combinaciones de mayúsculas y minúsculas, y en caso afirmativo lo publica en el *blog*

de GRIAL. La nueva publicación estará en ambos *blogs* y en el caso de ser modificada, siempre en el *blog* de origen, se actualizará en el *blog* destino.

En resumen, si un miembro de GRIAL realiza una publicación en su *blog* tan solo tendrá que poner la etiqueta correspondiente y la publicación irá al blog de GRIAL, y por consiguiente se publicará en la cuenta de Twitter y en el muro de Facebook del grupo de investigación. En las imágenes mostradas a continuación se ve el proceso descrito anteriormente:

- Figura 80. Se realiza una publicación en el blog del proyecto GRIAL 2.0 con la etiqueta “Grial”.



Figura 80. Publicar en el blog del proyecto

- Figura 81. Se envía automáticamente la nueva publicación a mi cuenta Twitter.

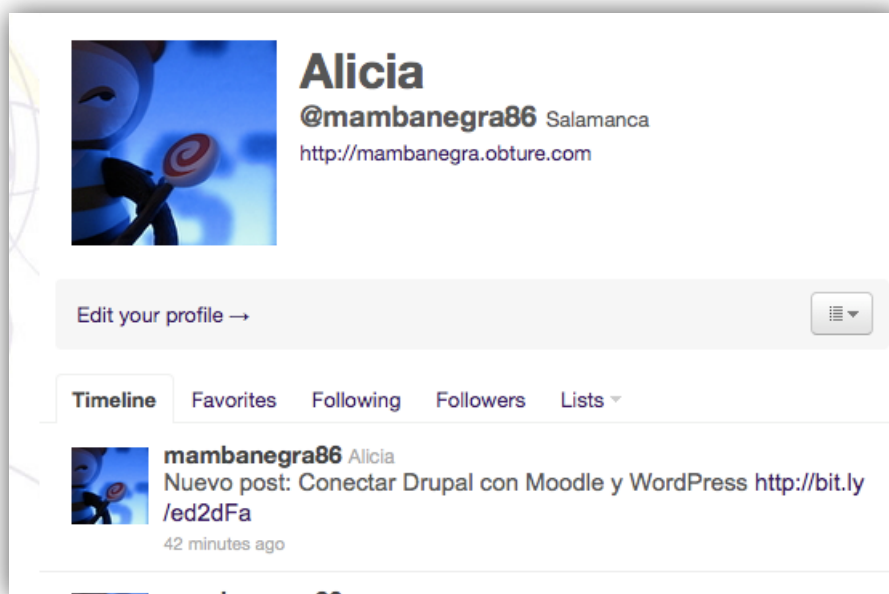


Figura 81. Se envía la publicación a Twitter

- Figura 82. La aplicación *FeedWordPress* localiza la nueva publicación con la etiqueta “Grial” y la publica en el *blog* del grupo de investigación.



Figura 82. Se publica en el blog de GRIAL

- Figura 83. Se envía automáticamente la nueva publicación a la cuenta de GRIAL en Twitter.



Figura 83. Se envía la publicación al Twitter de GRIAL

- Figura 84. La aplicación *RSS Graffiti* de mi perfil de Facebook detecta una nueva publicación en el *blog* y la publica en mi muro.



Figura 84. Publicar en mi Facebook

- Figura 85. Por último la aplicación *RSS Graffiti* de la página de GRIAL en Facebook detecta una nueva publicación en el *blog* y la publica en el muro.



Figura 85. Se publica en el Facebook de GRIAL

6.7.3. Gestión dinámica de enlaces recomendados

Como se ha ido viendo en apartados anteriores, en GRIAL 2.0 prima la automatización y la sencillez para que con el mínimo esfuerzo se consigan grandes resultados. Mantener un listado de enlaces dentro del portal puede ser también una tarea engorrosa, y en muchos casos puede suponer que el listado quede desactualizado.

Para conseguir que los enlaces recomendados en el portal sean dinámicos y su mantenimiento sea una tarea trivial, se ha optado por utilizar Delicious como fuente de dicha sección. Cada uno de los miembros del grupo de investigación que desee aportar enlaces deberá disponer de una cuenta en Delicious y pertenecer a la red de personas de la cuenta de GRIAL. Cuando se añada un enlace que sea interesante para mostrar en el portal, tanto desde la propia cuenta de GRIAL como desde la de uno de sus miembros, se deberá indicar la etiqueta “grial”.

El siguiente paso es procesar los enlaces dentro del portal. Para ello se hace uso de la *API Ajax de Google para feeds*. En primer lugar se obtienen por RSS los enlaces de la cuenta de GRIAL en Delicious que poseen la etiqueta “grial” y se muestran. Se lleva a cabo el mismo proceso con los enlaces de los usuarios que pertenecen a la red de GRIAL. Cada vez que la sección de enlaces es cargada se obtienen los correspondientes enlaces por RSS de tal forma que el listado se mantendrá en todo momento actualizado.

En la Figura 86 se puede ver, mediante un diagrama de actividad, el proceso de gestión dinámica de los enlaces recomendados.

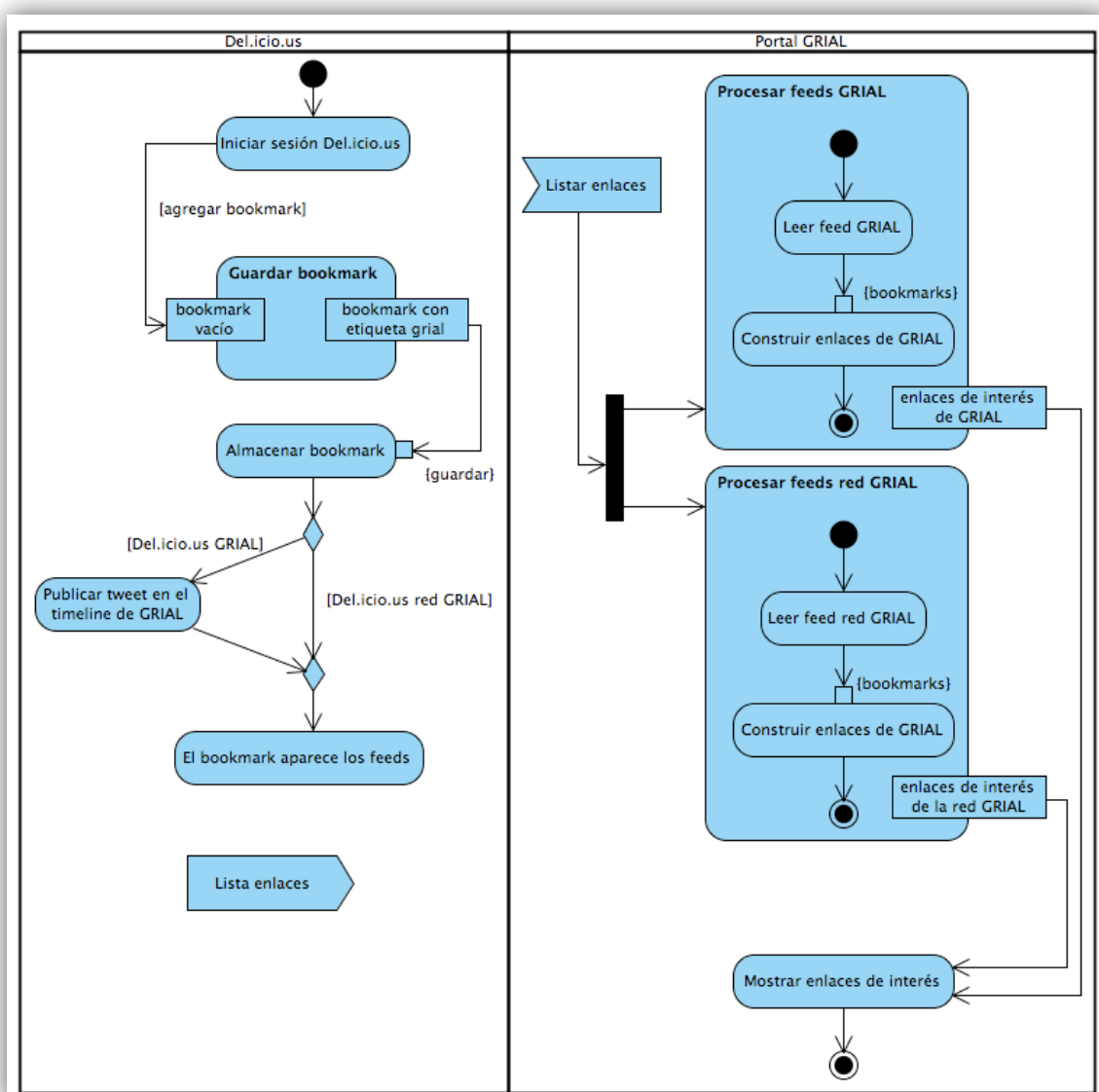


Figura 86. Diagrama de Actividad. Publicar enlaces de interés en el portal

7. Proyectos relacionados

Antes de llevar a cabo la integración de sistemas se ha buscado información acerca de desarrollos existentes. No se ha encontrado ningún precedente de integración entre una instalación de Drupal, una de Moodle y una de WordPress, o al menos no existe información técnica acerca de ello. Sí que existen casos de integración entre pares de estas herramientas, siendo la combinación de Drupal con Moodle y de Drupal con WordPress la que concierne a este proyecto, ya que desde un primer momento se ha definido Drupal como la parte central del sistema.

Cuando se utiliza la palabra integración no se está hablando de fusionar las herramientas, es decir, se quieren mantener independencia entre los tres sistemas pero que las tareas básicas que se realizan dentro del grupo de investigación, y que implican el uso de los tres, se realicen desde una única interfaz. La forma de realizar esto es mediante la utilización de servicios web que permitan realizar de forma transparente las diferentes operaciones sobre cada uno de los sistemas.

Buscando precedentes lo único que se ha podido encontrar son pequeños desarrollos que integran algunas partes de WordPress o Moodle en Drupal.

7.1. *Drupal y Moodle*

El módulo *Moodle Single Sign On* desarrollado para la versión 5 de Drupal permite que los usuarios de Drupal se añadan de forma automática a Moodle. Para ello es necesario que Moodle tenga acceso de lectura a la base de datos de Drupal y que tenga activada la matriculación externa.

Existe otro módulo denominado *Moodle Integration* que necesita la configuración de *Drupal* como servidor OpenID y Moodle como cliente para que el módulo funcione correctamente. Este módulo permite mostrar dentro de Drupal la información de los cursos existentes en Moodle, así como enlaces a dichos cursos.

7.2. *Drupal y WordPress*

Existen varios módulos de Drupal cuyo objetivo es conectar Drupal con WordPress.

Infowordpress es un módulo que obtiene las publicaciones y los *blogs* de una instalación de WordPress MU y los visualiza en bloques utilizando para ello el servicio web XML-RPC que proporciona WordPress.

Post2blogs se trata de un módulo cuyo fin es que las publicaciones realizadas en Drupal sean publicadas en los blogs de WordPress, BlogSpot y MovableType previamente configurados. Dicho proyecto no posee ningún tipo de código y está parado desde 2007.

Por último *WordPress publish*, que proporcionaría parte de la integración que se necesita en GRIAL 2.0. Este módulo permite publicar de forma automática en un *blog* de WordPress cuando un nodo es publicado en Drupal. El inconveniente es que se trata de un módulo actualmente en desarrollo y por tanto no está todavía disponible para su uso.

8. Conclusiones y líneas de trabajo futuras

Desarrollar un portal académico 2.0 requiere tener en cuenta un gran número de variables para que el sistema final cubra los requisitos planteados inicialmente. El volumen de trabajo necesario para su consecución ha hecho necesario definir una buena planificación que permitiera un desarrollo evolutivo. GRIAL necesitaba poder disponer del portal lo antes posible, por lo que había que montar la infraestructura y luego se irían introduciendo mejoras e integrando servicios.

Para poder preparar y configurar toda la infraestructura *hardware* primero ha sido necesario realizar una cuidadosa selección de las herramientas utilizadas como pilares del sistema. Esto ha supuesto un largo proceso de investigación, había que elegir aquellas que permitieran obtener los mejores resultados. WordPress como el gestor de *blogs* Agorá, Moodle como la plataforma de *eLearning* Pólis, Drupal como el portal de contenidos y DSpace como repositorio, ha sido el principal *software* utilizado, pero no el único. Cada una de estas herramientas permite extender su funcionalidad, y para seleccionar las extensiones que había que instalar se han estudiado las necesidades de los miembros a la hora de utilizar cada una de las herramientas.

Una vez conocidas estas aplicaciones se ha podido estructurar el portal, con la distribución de los diferentes elementos en los servidores disponibles, de tal forma que el sistema no estuviera ligado a una máquina, sino que explotara la posibilidad de disponer de varias, con el objeto de aprovechar las características de cada una. GRIAL ha podido disponer de las herramientas a pesar de que todavía no se pudiera considerar el conjunto como un portal.

Con las herramientas en producción, ha sido necesario mantener un servidor con una copia de las mismas para poder llevar a cabo el desarrollo del portal. Las aplicaciones desarrolladas para suplir los requisitos del sistema han sido primero probadas en este entorno de preproducción para luego pasar directamente al entorno de explotación. A esta fase de pruebas hay que añadir el *feedback* recibido por las comunidades de usuarios de las diferentes aplicaciones, ya que, cada vez que se ha finalizado el desarrollo de una extensión, esta se ha puesto a disposición de los usuarios, lo que ha permitido obtener información de la utilización de la aplicación en entornos diferentes al de GRIAL.

A la hora de suplir los requisitos funcionales se ha intentado no reinventar la rueda, es decir, si algo ya está hecho, reutilízalo. Se han desarrollado *plugins* para WordPress que cubrían necesidades básicas a la hora de dar vida a un gestor de *blogs* pero que hasta el momento no se habían implementado. Proporcionar un método de búsqueda a través de todos los *blogs* del sistema, o visualizar las etiquetas más utilizadas por todos los usuarios, aportan una mayor integración de los *blogs* individuales, lo que fomenta una estructura de comunidad. Lo mismo ocurre con los desarrollos aportados a la comunidad de Moodle, los administradores pueden organizar por categorías todos los cursos del sistema pero los usuarios solo ven un listado de todos los cursos en los que participan. Si el listado es pequeño no hay problema, pero si el número de cursos es elevado resulta necesario poder establecer un orden. De esa necesidad surgió el desarrollo de categorías personales para Moodle, que actualmente se utiliza tanto en GRIAL como en la plataforma institucional de la USAL, Studium, cubriendo otro aspecto que no estaba contemplado en ningún otro desarrollo realizado para la plataforma.

Una vez que se han tenido todos los sistemas completos individualmente, se ha trabajado en su integración. Inicialmente se realizaron los estudios necesarios para definir como se conectarían los sistemas, por lo que a la hora de implementarlo ya se habían tomado todas las decisiones oportunas y se habían buscado soluciones para los posibles problemas. Aunque los tres sistemas a conectar se han instalado en el mismo servidor, la conexión entre ellos se ha realizado de tal forma que funcione si en un futuro cada sistema está en una máquina. Se ha conseguido realizar una integración muy independiente de las plataformas utilizadas, de tal forma que, si un día uno de los elementos varía, solo habrá que desarrollar la parte de integración del nuevo elemento. También se trata de una integración robusta frente a cambios de versión, por lo que el sistema no está atado a la versión actual de las diferentes plataformas, pudiendo satisfacer uno de los principales obtenidos, la posibilidad de evolución del sistema.

Definir la política de disseminación, de tal manera que todos los servicios estuvieran cubiertos, y la información pudiera llegar a los usuarios a través de diferentes vías, ha sido una de las tareas más laboriosas. Qué información y en qué servicios puede aparecer duplicada, si se conecta todo con todo o todo fluye hacia un mismo punto, son variables a tener en cuenta. De nada servía conectar servicios porque sí, ya que el objetivo de mantener una red de GRIAL es que la información llegue adecuadamente a aquellos que puedan estar interesados en recibirla.

Una vez instaladas las aplicaciones, adaptadas a las necesidades de GRIAL mediante desarrollos que cubren los objetivos planteados inicialmente y realizada la integración tanto entre los sistemas que forman el portal como con los servicios externos, había que tratar el acceso a todo ello. Además de tener en cuenta la seguridad a lo largo de todo el recorrido, utilizando técnicas de cifrado a la hora de conectar los sistemas de GRIAL, quedaba por cubrir el objetivo de lograr el acceso *Single Sign On*.

Tras buscar información acerca de la mejor manera de llevar a cabo el acceso único a todos los servicios, y tras observar las diferentes soluciones que podían aplicarse, se ha optado por mantener un servidor OpenID de GRIAL, lo que aumenta aún más la independencia entre los sistemas que forman el portal. Si un sistema deja de funcionar, se cambia o se añade uno nuevo, de lo único que hay que preocuparse es de que tenga instalado un cliente OpenID. Esto es posible porque los datos de acceso de todos los miembros se han almacenado únicamente en el servidor de identidad, da igual si los servicios cambian, el servidor sigue siendo el mismo. Tras conseguir tener un sistema robusto de acceso, se ha podido ver realmente la utilidad del mismo. Es cierto que requiere tener que recordar una URL en vez de un nombre corto de usuario, pero a cambio se tiene la misma URL y la misma contraseña para todos los sistemas, en vez de tener la necesidad de utilizar diferentes usuarios y contraseñas difíciles de recordar e incrementando la posibilidad de robo de identidad.

Se puede concluir que los objetivos *software* y los objetivos técnicos se han alcanzado satisfactoriamente, dotando al grupo de investigación GRIAL de un potente sistema que permitirá ahorrar trabajo a todos sus miembros. En la Figura 87 se puede ver el resultado en conjunto, los servicios utilizados, el portal 2.0, la imagen corporativa del grupo.



Figura 87. GRIAL

Desde el ámbito personal se ha tratado de una dura, pero satisfactoria, experiencia que ha servido para adquirir muchísimos conocimientos y mucha más soltura a la hora de afrontar problemas en un entorno laboral.

Los meses transcurridos durante la elaboración GRIAL 2.0 han supuesto de un aprendizaje continuo. Al tratarse de un sistema compuesto por diferentes herramientas, cada una con sus particularidades, se han planteado problemas a lo largo de todo el proyecto, no solo durante las primeras fases. Se han tenido que tomar decisiones complicadas cuya repercusión en el sistema podía acarrear consecuencias fatídicas si no se sopesaban adecuadamente los pros y los contras de una solución frente a otra.

Cada nueva herramienta, cada servicio web, ha requerido de un periodo de aprendizaje, bien para familiarizarse con la utilización de la misma o para conocer su funcionamiento interno y poder llevar a cabo desarrollos para extender su funcionalidad. Esto ha hecho que se haya adquirido gran experiencia en el uso y desarrollo de un amplio abanico de herramientas de *software* libre, alcanzado así una parte de los objetivos personales.

A pesar de que todos los requisitos han sido cubiertos, a lo largo del desarrollo han surgido nuevas ideas y necesidades para mejorar el sistema que serán llevadas a cabo en un futuro:

- Implementar un servicio de *webconference* de GRIAL.
- Añadir más datos estadísticos a los informes de Moodle, incluyendo gráficas para mostrarlos de forma visual.
- Dar soporte de Gravatar en Moodle para que los tres sistemas que forman GRIAL 2.0 mantengan sincronizadas sus imágenes de perfil.

- Integrar en el gestor de blogs herramientas de Google como *Google Analytics for WordPress* y *XML sitemap*.
- Añadir LinkedIn al conjunto de herramientas sociales que utiliza GRIAL.
- Proporcionar mayor variedad de temas de WordPress que soporten *blogs* de diferentes tipos.

9. Referencias

O'Reilly, T. *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software.* 2005.
(<http://www.oreillynnet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>)

García Peñalvo, F.J., Laviña Orueta, J. y Mengual Pavón, L. *Libro Blanco de la Universidad Digital 2010.* Pp. 29-61. Barcelona: Ariel, 2008.
(<http://www.universidaddigital2010.es>)

Aruguete, G. *Redes sociales. Una propuesta organizacional alternativa.* 2001.
(http://practicasgrupales.com.ar//index.php?option=com_content&task=view&id=76)

W3C. *Guía Breve de Servicios Web.*
(<http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>)

Christakis, N. A. y Fowler, J. H. *Conectados.* Madrid: Taurus, 2010.

Jacobson, I., Booch, G., Rumbaugh, J. *The Unified Software Development Process.* Object Technology Series. Addison-Wesley, 1999.

Pastor, O., Abrahão, S. M., Fons, J. J., Ramón, S. *An Object-Oriented Approach to Automate Web Applications Development.* En Proceedings of the 2nd International Conference on Electronic Commerce and Web Technologies (EC-Web'01). Series: Lecture Notes in Computer Science, LNCS 2115, pp. 16-28. Springer Verlag, 2001.