

UNIVERSIDAD DE ALCALÁ

Escuela Técnica Superior de Informática

Ingeniería Técnica en Informática de Sistemas



Proyecto Fin de Carrera

**GESTIÓN DINÁMICA DE GRUPOS EN UNA
PLATAFORMA DE APRENDIZAJE VIRTUAL
BASADA EN TIKIWIKI**

Autor: Aldo Borrero González

Director: Roberto Javier López Sastre

TRIBUNAL:

Presidente: D. Hilario Gómez Moreno

Vocal 1º: D. Ricardo Jiménez Martínez

Vocal 2º: D. Roberto Javier López Sastre

CALIFICACIÓN:..... **FECHA:**.....

GESTIÓN DINÁMICA DE GRUPOS

en una plataforma virtual

BASADA EN TIKIWIKI



Aldo Borrero González

11 DE MAYO DE 2012

*Proyecto Fin de Carrera redactado
para la obtención del título de
Ingeniería Técnica en Informática de Sistemas*

Versión 1.2

AGRADECIMIENTOS

Porque la felicidad no es la suma de las experiencias individuales, va más allá de éstas y tiene mucho que ver con la percepción y memoria interna de nuestra vida en su conjunto. Por esta razón la gratitud es la clave. Gratitud por lo bueno y lo menos bueno de la vida. El ver la vida como una aventura en la que, por supuesto, no todo es fácil, pero en la que el centro se encuentra en el proceso mismo, no en el objetivo final. Vivir con el sentimiento de que cada día es nuevo. Disfrutar del camino y sacurdirse el polvo después de cada caída, porque, como Edward Diener demostró, pasado cierto tiempo de cualquier tragedia, se suelen recuperar los niveles normales de felicidad de cada persona, siendo lo que más nos cuesta superar la pérdida de un ser querido y la del puesto de trabajo. Experimentar dolor en la vida da hondura al ser. Quedarse apegado al dolor es un sinsentido. En definitiva, cada uno es responsable de ver la botella medio vacía o medio llena de la belleza de la vida.

*Excusas para no pensar,
Eduardo Punset*

ÍNDICE GENERAL

Agradecimientos	I
Índice general	II
Índice de figuras	v
Resumen	x
Abstract	xI
Resumen extendido	xII
1 Introducción	1
1.1	Objetivos del proyecto 2
1.2	Organización del documento 3
 Entendiendo ALMA y TikiWiki	
2 Estado de la cuestión	5
2.1	¿Qué es el proyecto ALMA? 5
3 Elección de un software wiki	14
3.1	¿Qué necesidades tiene ALMA? 14
3.2	¿Qué herramientas existen? 16
3.3	¿Por qué se escogió TikiWiki? 18
4 Conceptos fundamentales de TikiWiki	21

ÍNDICE GENERAL

- 4.1 ¿De qué manera podemos implementar una comunidad de práctica con Tiki-Wiki? 22
- 4.2 ¿Qué hemos aprendido de todo lo comentado en el capítulo? 42

Automatización de recursos con Profiles

- 5 Profiles 44
 - 5.1 ¿Por qué surgen los Profiles? 45
 - 5.2 ¿Qué son los Profiles? 46
 - 5.3 Flujo de trabajo con los Profiles 48
 - 5.4 Explicación de la interfaz de los Profiles 54
 - 5.5 Un ejemplo con Profiles 55
 - 5.6 Implementación interna de los Profiles 59
- 6 Gestión dinámica de la plataforma ALMA 66
 - 6.1 ¿Qué ha cambiado en la gestión de TikiWiki utilizando Profiles? 67
 - 6.2 Implementación interna de los *Profiles* desarrollados 69

Sumario

- 7 Conclusiones 78
 - 7.1 Conclusiones de la implementación 78
 - 7.2 Lecciones aprendidas 81
 - 7.3 Futuras líneas de trabajo 81

Apéndices

- A YAML 85
 - A.1 ¿Qué es YAML? 85
 - A.2 Especificación de YAML 87
- B Código Fuente 90
 - B.1 Organización del código de TikiWiki 90

ÍNDICE GENERAL

B.2	Listado de Profiles	91
C	Manual de Usuario	99
C.1	Obtención del código fuente	99
C.2	Requisitos previos	100
C.3	Instalación	100
	Acrónimos	103
	Bibliografía	104

ÍNDICE DE FIGURAS

- 1.1 Fases principales del proyecto. 3
- 2.1 Flujo de trabajo en la filosofía *wiki*. Como podemos observar, éste se repite indefinidamente. 10
- 2.2 Distintas comunidades de práctica trabajando conjuntamente en su propio dominio. El administrador de la plataforma es el encargado de que todo funcione correctamente. 11
- 3.1 Logotipo de Tikiwiki. 18
- 4.1 La comunidad de práctica de la asignatura de Derecho Romano. 22
- 4.2 Los recursos son aquellas herramientas que TikiWiki nos pone a nuestra disposición: *Wikis*, Foros, Galerías de Imágenes ... 24
- 4.3 Panel de habilitación de recursos en TikiWiki. 24
- 4.4 Panel de administración principal. Nos indica con transparencias si un recurso está habilitado o no (en este caso en concreto el recurso “galería de archivos” está desactivado). Si pinchamos en el icono del recurso nos lleva a su página de configuración. 25
- 4.5 Los grupos nos permiten aunar usuarios según preferencias. Pueden ser tan granulares como deseemos y un usuario puede pertenecer a uno o varios grupos. 26
- 4.6 Panel de administración de grupos. 26
- 4.7 Conjunto de recursos agrupados en una categoría y que pertenecen a la comunidad de práctica de Derecho Romano. 27
- 4.8 Listado de todas las categorías que se han creado en TikiWiki. 28

ÍNDICE DE FIGURAS

- 4.9 Vista detallada de la categoría Circuitos Eléctrico y sus recursos asociados. 29
- 4.10 Concepto de lo que son los permisos en TikiWiki. 29
- 4.11 Orden en el que se aplican los permisos en TikiWiki. Imagen inspirada en la documentación oficial http://twbasics.keycontent.org/img/wiki_up/tiki_permissions.png. 32
- 4.12 Listado de todas las categorías disponibles. Para editar los permisos de una categoría hay que pinchar en el icono de la llave verde. 33
- 4.13 Listado de los posibles permisos que se pueden asignar a la categoría (en la imagen no se muestra toda la información). Como se puede apreciar, se pueden asignar permisos de manera independiente para cada grupo. 33
- 4.14 Algunos valores almacenados en la tabla tiki_preferences. 35
- 4.15 Estructura de la tabla tiki_preferences. 36
- 4.16 Un ejemplo de lo que contiene el array \$prefs (que son todos los datos de la tabla tiki_preferences). 36
- 4.17 Ejemplo de dos posibles alumnos en el cual el Alumno 1 desconoce que existe la comunidad de práctica de Circuitos. Las perspectivas en TikiWiki nos permiten aislar de forma transparente comunidades de práctica dentro de una misma instalación. 37

ÍNDICE DE FIGURAS

- 4.18 Supongamos que existen dos usuarios y que pertenecen a dos comunidades de práctica diferentes. En el primer ejemplo (denotado en el gráfico con el número uno) los dos ven el color del logotipo de TikiWiki azul, ya que, así lo indica sus preferencias genéricas. En un instante de tiempo $t + 1$ el usuario amarillo decide entrar en una comunidad de práctica. Dicha comunidad de práctica posee una perspectiva que cambia la preferencia de que el logotipo sea de color azul a que sea verde (se puede observar en el gráfico número dos como se ejecuta el cambio internamente). El usuario amarillo acaba entrando en la comunidad de práctica (gráfico número tres) y ya lo que él observa es que el logotipo ha cambiado de color, sin embargo, para el usuario de color rojo al no pertenecer a la misma comunidad de práctica (y, por consiguiente, al no haber ejecutado la perspectiva asociada) observa que el color del logotipo sigue siendo azul. En el momento en el que el usuario amarillo abandone la comunidad de práctica, todos los cambios que ha efectuado la perspectiva se revierten y se volvería a estar en el gráfico número uno. 39
- 4.19 TikiWiki permite crear, editar y borrar perspectivas. Hay que reconocer que la interfaz web no es la forma más cómoda de administrar las perspectivas, aunque para ediciones sencillas sirve perfectamente. 40
- 4.20 Aquí se listan las preferencias que se activarán en la comunidad de práctica de la asignatura de Circuitos Eléctricos. Han sido generadas de forma automática con un *Profile* (añadir preferencias de forma manual es un proceso tedioso). 40
- 4.21 Ejemplo de menú que se puede utilizar para cambiar las perspectivas y cambiar de comunidad de práctica de manera dinámica. 41
- 4.22 Panel de administración de menús. Se pueden crear, editar y borrar. Hay una cantidad inmensa de opciones, más información en <http://doc.tiki.org/Menu+HOWTO>. 41
- 5.1 Un *Profile* modifica las preferencias de TikiWiki de la manera en la que esté especificado. El rectángulo con el interrogante es el traductor que se encarga de “entender” lo que el *Profile* desea alterar (veremos más detalles internos en la § 5.3 (pág. 48)). 46

ÍNDICE DE FIGURAS

- 5.2 Para cada *Profile* hay que definir exactamente que deseamos ejecutar. Una forma de crear un *Profile* con éxito es dividir lo que queremos crear en etapas. 47
- 5.3 49
- 5.4 Ejemplo de una posible página *wiki* con dos bloques de *Profiles*. Cada bloque puede realizar una tarea distinta o bien ser la misma tarea dividida en secciones y explicada por partes. Ejemplo extraído de: <http://profiles.tiki.org/Article+Handler>. 51
- 5.5 Estas son las opciones que se pueden poner en un *Profile* si queremos crear un usuario. 53
- 5.6 Panel principal de administración de los *Profiles*. 55
- 5.7 Panel donde podemos listar los *Profiles* que tiene un repositorio y ejecutarlos. 56
- 5.8 Panel donde podemos exportar las preferencias que hayamos modificado de TikiWiki. Se puede utilizar como método de copia de seguridad para restaurar posteriormente la misma configuración. 57
- 5.9 Panel que nos permite añadir nuevos repositorios, agregar nuevos *Data Channels* y ejecutar *Profiles* de prueba. 62
- 5.10 Categoría creada con una página *wiki* tal y como el *Profile* había especificado. 62
- 5.11 En la sección de preferencias se pueden poner las preferencias directamente que encontramos en la tabla `tiki_preferences` como ya comentamos. Hay que añadir que al crear múltiples recursos en este *Profile* hacemos uso de las referencias (aquellas que comienzan con `ref`, lo usaremos más adelante [8]). 63
- 5.12 Este código copiado tal cual en la página *wiki* genera la interfaz gráfica e invoca al *Profile* que hemos creado antes. Hay que observar que en donde pone `channel` se ha escrito el nombre del *identificador que dimos en el Data Channel*. En la documentación oficial tenemos más información acerca del Plugin `Data Channel` que es como se denomina esta herramienta que genera interfaces para los *Profiles* [9]. 63
- 5.13 Escribimos el contenido del *Profile* en una página *wiki*. 64

ÍNDICE DE FIGURAS

- 5.14 Interfaz gráfica generada por el *plug-in Data Channel*, podemos ejecutar el *Profile* cómodamente con un clic de ratón. 64
- 5.15 Implementación del Users Handler. 65
- 6.1 Dado que el *Profile* es una página *wiki* hay que buscar donde se encuentra éste. 67
- 6.2 Es tan sencillo como poner el nombre de la asignatura a crear y darle a Ir. El *Profile* se encargará del resto. 68
- 6.3 En el panel de usuarios se pueden agregar qué usuarios pertenecen a qué grupos. Por defecto, por cada comunidad de práctica que se crea, el usuario *admin* pertenece a cada grupo de dicha comunidad. 68
- 6.4 Etapas empleadas para elaborar el *Profile* que configura TikiWiki a las necesidades de Aula Libre Multidisciplinar Abierta (ALMA). 69
- 6.5 Supongamos que un administrador o docente desea crear una asignatura. Para ello se dirige a la página Crear_Comunidad_de_Practica_GUI. Ésta es la encargada de dibujar la interfaz gráfica que es lo que el usuario ve como un campo para introducir un texto (aunque puede ser tan compleja como deseemos). Una vez que el administrador ha decidido el nombre de la asignatura y pulsa el botón enviar, la petición es enviada a la página *wiki* Crear_Comunidad_de_Practica_Template que es donde se encuentran las instrucciones precisas para generar una comunidad de práctica. A continuación, se rellenan las variables oportunas para personalizar los nombres de los grupos, las páginas *wikis*, etc y se ejecuta uno a uno las instrucciones que hay dentro de dicho *Profile* (tal y como explicamos en el capítulo anterior). 73
- 6.6 Elementos que genera el *Profile* Crear_Comunidad_de_Practica_Template. 74
- 7.1 Implementación real de ALMA con TikiWiki. 80
- 7.2 Ejemplo del programa Automator. 83
- A.1 Ejemplo de configuración del programa Jekyll utilizando YAML Ain't Another Markup Language o YAML No Es Otro Lenguaje de Marcado (YAML). 86

RESUMEN

El presente trabajo consiste en la aceleración de la gestión que tiene que realizar un administrador o docente en una plataforma de aprendizaje, en nuestro caso TikiWiki [36]. Con motivo de esta aceleración se han desarrollado unos *Profiles* [5] (una herramienta de automatización que existe dentro de la plataforma y en la que nos basamos para mejorar la gestión), que permiten generar grupos de comunidades de práctica con tan solo pulsar un botón en lugar de tener que configurar cada elemento uno a uno con el consiguiente ahorro de tiempo que ello conlleva. Los resultados son claros, el administrador o docente de la plataforma puede desarrollar mejor su tarea y administrar de manera más eficiente sus comunidades de práctica.

Palabras clave: *ALMA, TikiWiki, Comunidades de Práctica, Gestión, Profiles*

ABSTRACT

This work involves the acceleration of the management that an administrator or teacher has to perform in a learning platform, in our case TikiWiki [36]. As a result of this acceleration, we have developed some *Profiles* [5] (an automation tool that exists within the platform on which we rely to improve the management), which can generate groups of communities of practice with few interactions instead of having to configure each element one by one. The facts are clear, the administrator or teacher of the platform can manage more efficiently their communities of practice.

Keywords: *ALMA, TikiWiki, Communities of Practice, Management, Profiles*

RESUMEN EXTENDIDO

ACTUALMENTE EL MUNDO de las nuevas tecnologías ha supuesto una revolución en la manera que tenemos las personas de consumir la información. Podemos afirmar que múltiples sectores de la sociedad ven cada día como muchos de los principios sobre los que han estado asentados de manera tradicional dejan de tener sentido y acaban adaptándose, de una manera u otra, a las nuevas formas de expresión: el sector educativo es uno de ellos. Durante los últimos veinte años, y de manera más acentuada con la aparición y posterior expansión de Internet, dicho sector está buscando maneras de adaptarse al nuevo mundo tecnológico (que es el que impera ahora). Si echamos un vistazo a las posibilidades que nos ofrece Internet como herramienta, nos daremos cuenta de que hay muchas propuestas que tratan, con un mayor o menor grado de éxito, de intentar sustituir por completo el concepto que tenemos de enseñanza *in situ*, donde ésta se produce en un único lugar y momento determinado, por algo más genérico y flexible como es la educación a distancia donde el tiempo y el espacio ya no juegan un papel fundamental.

Otras propuestas, como es el proyecto ALMA, tratan de no ser tan radicales en su visión y buscan combinar las teorías del aprendizaje social (que basa sus principios en las interacciones que se producen entre las personas) con otras más modernas que implican el uso de plataformas *wikis*. De la unión de estos dos conceptos nace lo que Wenger, Snyder y McDermott denominaron en su libro *Communities of Practice: Learning, Meaning, and Identity* [46] como *comunidades de práctica*.

El objetivo principal de las comunidades de práctica es pretender que distintas personas, de diferente índole y origen, trabajen de manera aunada y consensuada en una misma temática. Las plataformas *wikis* se prestan como las herramientas que hacen posible que la información fluya de manera equilibrada entre los diferentes participantes.

RESUMEN EXTENDIDO

Por lo tanto, y en este caso tan concreto, es muy importante averiguar cuales son las necesidades específicas de ALMA y las implicaciones que tienen el uso de las comunidades de práctica. De esta manera obtenemos una lista de requisitos que se puede analizar para encontrar, de las diversas herramientas que existen en el mercado, una plataforma que se ajuste a las características particulares de nuestro proyecto. En nuestro caso, y tras examinar las ventajas y desventajas de cada una, el resultado de dicha elección fue TikiWiki.

Pero, utilizando dicha herramienta, el principal problema que nos encontramos a la hora de administrar grandes comunidades de práctica es la propia gestión de éstas. Por ejemplo, cuando un administrador necesita crear una comunidad de práctica dentro de TikiWiki, conlleva utilizar diferentes partes de la herramienta que, a simple vista, no son triviales de manejar. Además, podemos añadir, que es un proceso que puede llegar a ser tedioso y con grandes probabilidades de que el administrador cometa errores. Este hecho, en concreto, impide que la plataforma sea accesible a un docente (que es una persona que usualmente no está versada en conocimientos técnicos pero que, dada la naturaleza de ALMA, puede poseer un rol de administrador en una comunidad de práctica) y eso es algo que debemos solventar.

¿Qué soluciones existen para paliar este problema en concreto? La respuesta son los *Profiles*¹. Éstos forman parte del repertorio de herramientas de TikiWiki y se sitúan como los útiles que automatizan y solucionan el problema enunciado arriba así como otros totalmente distintos. Son múltiples las ventajas que aportan los *Profiles* a TikiWiki en general y a ALMA en particular. Listamos, a continuación, algunas de ellas:

- Permiten personalizar cualquier aspecto de TikiWiki (de esta manera adaptamos la plataforma a nuestras necesidades).
- Simplifican la vida del administrador ya que le evita tener que manejar la plataforma para tareas repetitivas. La filosofía de un *Profile* es: “Configúralo una vez, ¡ejecútalo tantas veces como quieras!”.

¹ Aunque la RAE sugiere que las palabras de origen extranjero deben de ser traducidas a sus homónimas en castellano si no son extranjerismos crudos, se ha decidido mantener la palabra original del inglés por poseer una mayor connotación tecnológica en el contexto en el que se mueve esta obra. La traducción correcta sería *perfiles*.

RESUMEN EXTENDIDO

- Facilitan que TikiWiki sea una plataforma más accesible para ese grupo de personas que no tienen tantos conocimientos avanzados. (Las tecnologías deben de ser integradoras y no excluyentes.)
- Favorecen la reutilización de los propios *Profiles* ya que éstos no son más que un conjunto de instrucciones que se encuentran escritas en un fichero de *texto plano*.

Por lo tanto, saber manejar esta herramienta para implementar las comunidades de práctica de una forma cómoda, sencilla y fiable, se convierte en una necesidad más que en una afición; de la misma manera que cuando queremos viajar hasta París en coche hemos tenido que aprender el funcionamiento del acelerador y del freno (así como otra serie de nociones) para poder llegar al destino de manera segura.

A lo largo de este Proyecto Fin de Carrera (PFC) se describen las claves fundamentales para que un lector interesado en aprender las dos cuestiones principales que tratamos aquí (comunidades de práctica y la automatización de éstas) pueda manejarse con soltura en la plataforma de TikiWiki y más concretamente con los *Profiles*. La organización de este libro sigue una estructura lógica que permite entender los contenidos de manera sencilla y, sobre todo, didáctica (en la § 1.2 (pág. 3) se puede leer como se ha estructurado la obra).

El resultado de todo este esfuerzo: tanto del análisis de las necesidades del proyecto, como de la implementación de las comunidades de práctica con TikiWiki, como de la automatización en las tareas de gestión utilizando para ello los *Profiles* y la posterior documentación de todo lo que se ha realizado, ha dado como resultado una implementación real de ALMA en la que los profesores y los alumnos pueden utilizar la herramienta para crear comunidades de práctica paralelas a las enseñanzas impartidas en un aula.

1

INTRODUCCIÓN



DESDE LA REVOLUCIÓN INDUSTRIAL hasta ahora, la *Era de la Información* [51], la humanidad ha cambiado por completo su manera de vivir, de pensar y de actuar. Actualmente la tecnología se ha convertido en un pilar fundamental en el progreso humano. Utilizándola como herramienta hemos sido capaces de salvar vidas, de construir edificios imposibles o incluso de conectar en el momento a personas separadas por miles de kilómetros de distancia. Y no sólo eso, en la actualidad, la manera que tenemos de consumir la información ha cambiado notablemente en los últimos 20 años. Que estamos inmersos en un mundo lleno de datos, de posibilidades, de constantes decisiones... no es un hecho, es una realidad.

Se podría decir que el punto de inflexión lo marca la creación de Internet y su posterior expansión al público en general. Hoy en día con un móvil en nuestra mano accedemos a Google [12], consultamos el correo electrónico y estamos en contacto con nuestros seres queridos. Hemos sido testigos de como cada sector de la sociedad ha ido sucumbiendo, en mayor o menor medida, a las “maravillas” del progreso tecnológico. Algunos, en cambio,

1. INTRODUCCIÓN

como es el sector educativo todavía se encuentran en fases tempranas de adaptación. Pero, lo que no es discutible es que existen diversos esfuerzos que tratan de crear lo que se podría denominar como *Educación 2.0* [37].

En el mercado actual existen aplicaciones o plataformas que se dedican a la educación a distancia (conocido también como *e-learning* [50]) para intentar llevar los procesos de enseñanza y aprendizaje a un nuevo nivel. Las ventajas que presentan esta forma de instruir son múltiples ya que eliminan ciertos factores intrínsecos a la educación tradicional como, por ejemplo, las barreras que suponen el espacio o el tiempo. Así mismo añaden mejoras en la forma de asimilar nuevos contenidos puesto que la información fluye de manera asíncrona, es decir, no hace falta tener un diálogo en directo entre la persona que imparte una materia y la que la recibe.

Otros proyectos, como puede ser Aula Libre Multidisciplinar Abierta (ALMA) y que es el que tratamos en este escrito, surgen más bien como una necesidad de *complementar* a la educación tradicional sin pretender, de ninguna manera, sustituirla por completo.

Por lo tanto, a lo largo de este proyecto se buscarán las fórmulas adecuadas para poder llevar a cabo con éxito los ideales de ALMA que no son, ni más ni menos, que facilitar las tareas de gestión y administración utilizando para ello los *Profiles* [5] (que es una herramienta incluida dentro de la plataforma que hemos elegido y adaptado a nuestras necesidades: TikiWiki [36]).

1.1 OBJETIVOS DEL PROYECTO

El objetivo fundamental de este proyecto titulado “*Gestión dinámica de grupos en una plataforma virtual basada en TikiWiki*” es doble: por una parte trata de *explicar* de manera clara y concisa el funcionamiento de los *Profiles*. Al finalizar la lectura de este Proyecto Fin de Carrera (PFC) el lector debería ser capaz de poder crear nuevos *Profiles* que le permitan administrar la plataforma ALMA (o cualquier otra basada en TikiWiki), así como la comprensión del funcionamiento interno de dicha herramienta de automatización. Por otra parte, se desarrollan (y se explican) unos *Profiles* preparados por el autor del escrito que atienden a las necesidades concretas de ALMA.

En la Figura 1.1 se muestra el flujo de trabajo utilizado en el desarrollo de este PFC:

1. INTRODUCCIÓN

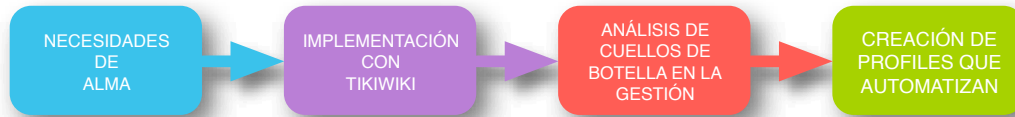


Figura 1.1: Fases principales del proyecto.

1.2 ORGANIZACIÓN DEL DOCUMENTO

Este PFC está dividido en tres partes:

1. La primera titulada *Entendiendo ALMA y TikiWiki* tratará de explicar de manera detallada qué es el proyecto ALMA, sus motivaciones, sus implicaciones, los casos de uso... (Capítulo 2 (pág. 5)), posteriormente se analizarán las diversas opciones para implementar de manera tangible y real el proyecto utilizando herramientas de *software* libre: TikiWiki en nuestro caso (Capítulo 3 (pág. 14)). En el Capítulo 4 (pág. 21) se profundizará con mayor detalle en los conceptos fundamentales de TikiWiki necesarios para entender el funcionamiento de los *Profiles* (se darán explicaciones concretas a características tan importantes como la gestión de grupos y de permisos, administración de recursos y el concepto de perspectivas).
2. La segunda parte titulada *Automatización de recursos con Profiles* se sumerge en la búsqueda soluciones concretas para atacar diversos problemas que aparecen con la gestión de una gran comunidad de usuarios. Se explicará qué son y cómo funcionan los *Profiles* (Capítulo 5 (pág. 44)). Una vez comprendido su funcionamiento interno podremos entender con detalle los *Profiles* que se han desarrollado para la gestión de ALMA (Capítulo 6 (pág. 66)).

Se concluye el proyecto con la última parte, *Sumario*, en el Capítulo 7 (pág. 78) con las lecciones que se han aprendido a lo largo de todo el desarrollo, una autocrítica de las partes del proyecto susceptibles de mejora y se mostrarán unas ideas de lo que podrían ser las futuras líneas de trabajo. Además, al final del libro, se encuentran disponibles una serie de apéndices que sirven como complemento a la información presentada en el resto del documento.

I

ENTENDIENDO ALMA Y TIKIWIKI

ESTADO DE LA CUESTIÓN



ESTE CAPÍTULO tratará de explicar qué es el proyecto ALMA y cuáles son sus motivaciones principales. De esta manera podemos hacernos una idea aproximada de las necesidades a cumplimentar y en qué áreas dirigir los esfuerzos. Sobre los pilares de conocimiento que se construyan aquí surgirán los demás capítulos que tratarán de dar forma a las ideas expuestas.

2.1 ¿QUÉ ES EL PROYECTO ALMA?

En la introducción de este PFC en el capítulo anterior § 1 (pág. 1) se ha comentado que uno de los sectores que todavía está en plena fase de adaptación a las nuevas tecnologías es el sector educativo. Eduardo Punset en su libro *Excusas para no pensar* señala que en la actualidad hay una crisis educativa y que hay que cambiar, de alguna forma, el modelo vigente [38, No todos los niños son iguales, pág. 50]:

2. ESTADO DE LA CUESTIÓN

Curtis W. Johnson, presidente de Citistates Group y socio director de Education Evolving [...], me contó que uno de los grandes errores de este sistema educativo imperante es que «tratamos a los niños como si fueran iguales. Los educamos igual a todos, les presentamos el material del mismo modo, y esperamos que todos aprendan las mismas cosas de la misma manera, durante el mismo día y al mismo tiempo. Esta postura no es realista, puesto que no contempla lo diferente que son los niños en muchísimas cosas, por ejemplo en el estilo de aprendizaje, pero también en el ritmo de adquisición de los conocimientos.» Johnson reivindica, además, la necesidad de enseñar nuevas competencias y habilidades para que los niños y las niñas de hoy en día dominen estas técnicas para conseguir trabajo en el mundo actual. «La mayoría de los niños, igual que la mayoría de los jóvenes, deberán adquirir destrezas que las generaciones anteriores no tenían. Me refiero a que no solamente tendrán que aprender asignaturas básicas, sino que deberán saber cómo encontrar las cosas que necesitan saber, y luego tendrán que aprender a trabajar tal como trabaja el mundo hoy, que es principalmente en equipo [...] deberán practicar el arte de la colaboración, que me parece que es el reto de colaborar con desconocidos. A todos nos encanta la idea de colaborar con nuestros amigos, y lo hacemos de muchas maneras todo el rato, pero el mundo laboral nos fuerza a llevarnos bien productivamente y crear algo de valor con gente que quizá ni siquiera nos gusta, lo cual requiere un tipo completamente distinto de educación.»

Del fragmento anterior podemos extraer las siguientes conclusiones: por un lado necesitamos hacer que la educación sea completamente personalizada, donde se adapte a las necesidades de cada persona; por otro, necesitamos centrar los esfuerzos en buscar un modelo cooperativo en los que puedan surgir las bases del aprendizaje social que es la clave que sustenta la reivindicación propuesta por Eduardo Punset y Curtis W. Johnson. Pero... ¿qué entendemos nosotros por aprendizaje social?, ¿cómo y cuándo se produce? Esto queda explicado de manera desarrollada en [39, pág. 1]:

Las aulas no son los únicos entornos donde los seres humanos aprendemos, somos capaces de hacerlo en otros muchos contextos. Pensemos en cualquier escena de la vida cotidiana. Si lo hacemos descubriremos entornos no formales de desarrollo, de aprendizaje, que a menudo resultan desapercibidos. Una vez recuperada esta perspectiva de los procesos de enseñanza y aprendizaje, podemos plantearnos qué significa esta dimensión.

2. ESTADO DE LA CUESTIÓN

Por aprendizaje social entendemos aquél aprendizaje que tiene en cuenta las interacciones sociales, es, por tanto, la forma en que los individuos adquieren conocimientos a través de la socialización e interacción con el medio.

Las teorías del aprendizaje social tienen en cuenta las interacciones sociales, pero adoptan una perspectiva de naturaleza psicológica. Destacan las relaciones entre personas que intervienen en la imitación y el modelado, y que, en consecuencia, se centran en el estudio de procesos cognitivos por los que la observación se puede convertir en fuente de aprendizaje.

En el fragmento de arriba se menciona que el aprendizaje social *requiere* de las interacciones sociales *a través* de un medio en el cual los participantes pueden aprender *observando*. Es innegable que una buena parte del aprendizaje, del desarrollo y de la creación de nuevas conductas en una persona tiene que ver con los factores sociales de su entorno y, no sólo eso, sino de las consecuencias positivas que le conlleva el realizar un acto que está bien visto —bien sea por la sociedad o por los semejantes— y que le es provechoso. Es probable que una persona que esté aprendiendo decida imitar y tomar como modelo las conductas que sean buenas —las que se “premian”— y que tienda a rechazar las que no lo son —las que se “castigan”— [53].

Las interacciones sociales se manifiestan de múltiples maneras y formas, puesto que muchas veces su frontera no está claramente diferenciada. Podemos decir de manera análoga lo mismo para el medio. Por ejemplo, en una conversación donde intervienen dos personas, un interlocutor y un oyente (y donde es frecuente el intercambio de roles), puede ser un buen ejemplo de interacción social. El medio en este caso podría ser la voz de los participantes. Otro ejemplo podría ser las comunicaciones indirectas en los que los individuos no interactúan en persona (*chat*, teléfono, correo electrónico...) y en los que el medio cobra mayor importancia, pues, condiciona la manera de recibir el mensaje. Lo que es común a ambos ejemplos es que, independientemente del medio, puede suceder que se den los mecanismos adecuados para que los individuos se enriquezcan con las aportaciones realizadas por cada participante en la conversación. Y es ahí, por lo tanto, donde se gestan las bases del aprendizaje social mencionado en el fragmento de arriba.

En la actualidad las nuevas tecnologías han creado y/o expandido nuevas formas de comunicación añadiendo o mejorando los medios existentes para expresar un mensaje. Esta última década ha supuesto un gran avance en la manera que tenemos de consumir y

2. ESTADO DE LA CUESTIÓN

procesar la información. La revolución de la llamada *Web 2.0* [40] ha traído consigo una nueva forma de entender las relaciones humanas. Actualmente propuestas tan conocidas como pueden ser *Flickr* [11] —donde los usuarios comparten sus fotos y votan las de los demás—, *Twitter* [44] —donde los usuarios escriben en 140 caracteres lo que piensan— o *Facebook* [10] —donde el usuario tiene la posibilidad de crear un espacio virtual y tener a sus amigos cercanos en contacto— no hacen más que reforzar el hecho de que lo social está en auge. Todos estos servicios web tienen en común la pretensión de que el usuario participe con sus semejantes de una manera cooperativa para la construcción de un conocimiento más amplio y enriquecedor.

Cabe entonces preguntarnos: ¿podemos de alguna manera aprovecharnos de las nuevas formas de interacción social y tratar de aplicarlas a un entorno de enseñanza y aprendizaje que pueda ser cooperativo? La respuesta es que sí y es en esta línea en la que trabaja el proyecto ALMA.

La misión principal de éste es que trata de crear espacios alternativos de enseñanza y aprendizaje a los que tenemos en la actualidad sin tratar de sustituirlos —por ejemplo, un aula tradicional donde se imparte una asignatura— si no más bien de complementarlos. Para ello se apoya en herramientas sociales cooperativas como son las *wikis*. De esta manera se logra personalizar la educación, adaptarla al ritmo de cada individuo (como señalaban Punset y Johnson anteriormente) y hacerla más social e interactiva. En el siguiente fragmento se explica de manera detallada las ventajas que aportan usar un *software wiki* en un aula [40, pág. 4]:

¿Qué es una *wiki*? Imaginemos un sitio web cooperativo donde los usuarios pudieran crear, editar, borrar o modificar el contenido. ¡Eso es una *wiki*! La *wiki* más conocida es la Wikipedia [48], una enciclopedia creada a partir de las aportaciones de todos los usuarios que la visitan.

¿Cómo podemos utilizar una *wiki* en nuestras clases? Una *wiki* nos permite crear un espacio de trabajo cooperativo con nuestros alumnos. Podemos montar una *wiki* para editar, entre todos, los apuntes de una asignatura (*wiki-apuntes*), o podemos utilizarla para crear un glosario con los términos conceptuales que surjan durante las clases. Incluso podemos proponer a los alumnos que mejoren algún artículo ya existente en la Wikipedia y que guarde relación con su asignatura, como por ejemplo este: http://es.wikipedia.org/wiki/Derecho_romano.

2. ESTADO DE LA CUESTIÓN

Son muchas las posibilidades de trabajo que acerca un espacio *wiki* al proceso de enseñanza y aprendizaje de los alumnos, y como docentes debemos indagar en ellas.

Todas las ediciones y cambios que se hacen en una *wiki* quedan registradas, de forma que el profesor sabe qué ha editado quien, y cuando lo ha hecho. Una *wiki* permite a los alumnos y al profesor editar de una forma muy sencilla los contenidos de un sitio web. Todos pueden trabajar editando un glosario con los conceptos de la asignatura, y además establecer discusiones sobre cada concepto en el que trabaja la clase. Es por lo tanto una herramienta sencilla y que facilita enormemente la tarea de editar contenidos en grupo.

Sobre la utilización de las *wikis* en la docencia podemos decir muchas más cosas. Rompen la asimetría y la unidireccionalidad de la enseñanza, de modo que los propios alumnos son los que enseñan a sus compañeros, pues el conocimiento se construye entre todos. El profesor adquiere nuevos roles: facilitador, moderador, orientador, y evaluador (continuo). También lo hacen los alumnos: colaboradores, cooperadores y capaces de enseñar a sus compañeros. Las barreras de la clase se superan de tal forma que podemos compartir y generar conocimiento con el resto del mundo.

Como queda demostrado arriba las *wikis*, por su naturaleza, se asientan en los principios de las *interacciones sociales*, es decir, se nutren de las participaciones de todas las personas. Además, su máximo interés reside en el hecho de que se dispone de una total libertad para poder modificar tantas veces se quiera el *mensaje* (pudiendo crear nuevos contenidos, editarlos, borrarlos y recuperarlos cuando y como deseemos). A esto es a lo que se conoce como *filosofía wiki* (Figura 2.1) [54]. Lo maravilloso de esta forma de trabajar es que, de manera inconsciente, se acaba creando lo que Wenger en su libro *Communities of Practice: Learning, Meaning, and Identity* [46] denominó por primera vez, en 1988, como *comunidades de práctica*. Y... ¿qué son exactamente?

Una definición, más o menos formal, podría decir que: “las comunidades de práctica se originan cuando existe un grupo de individuos que comparten un interés, o un conjunto de problemas o una pasión sobre un tema determinado y, que profundizan tanto su conocimiento y experiencia en el área, que lo que produce es que las relaciones entre los distintos miembros se acaban fortaleciendo” [41].

2. ESTADO DE LA CUESTIÓN

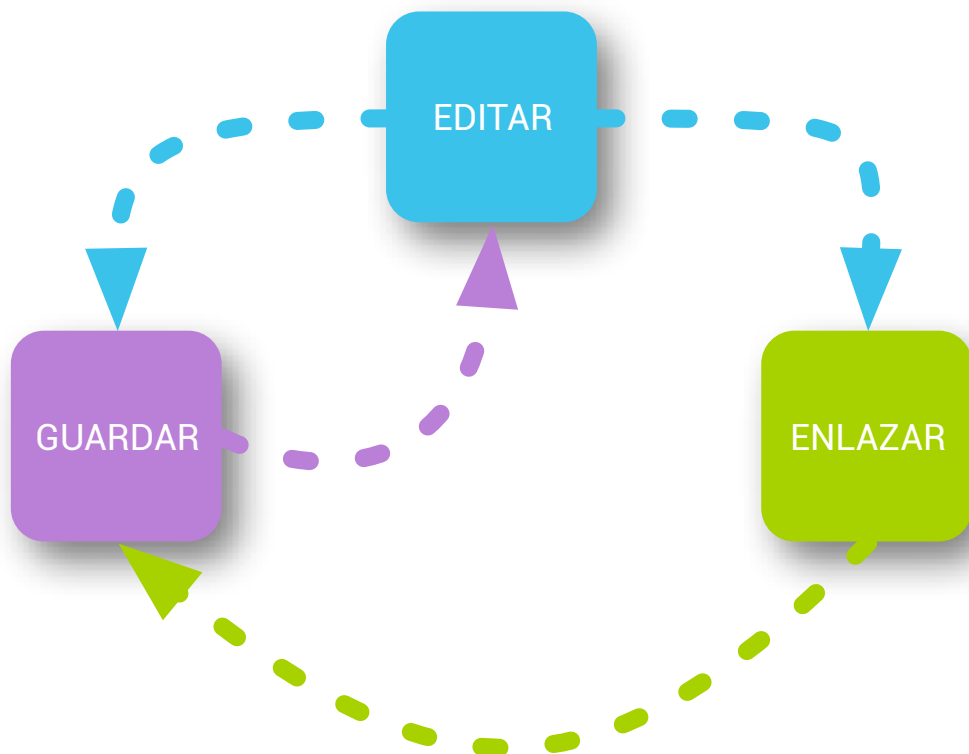


Figura 2.1: Flujo de trabajo en la filosofía *wiki*. Como podemos observar, éste se repite indefinidamente.

Las comunidades de práctica se pueden dar en múltiples ámbitos (o en todos a la vez), es más, es bastante probable que cualquier persona que tenga una afición, sea cual sea, por el mero hecho de compartir sus gustos con otras personas ésta pertenezca ya, sin saberlo, a una comunidad de práctica. Aunque hay que discernir que no todas las comunidades a las que se puede pertenecer son comunidades de práctica. Para que lo sean se deben dar tres características fundamentales:

- *Dominio*: es el campo de interés compartido por la comunidad y es el que crea una identidad común al grupo.

2. ESTADO DE LA CUESTIÓN

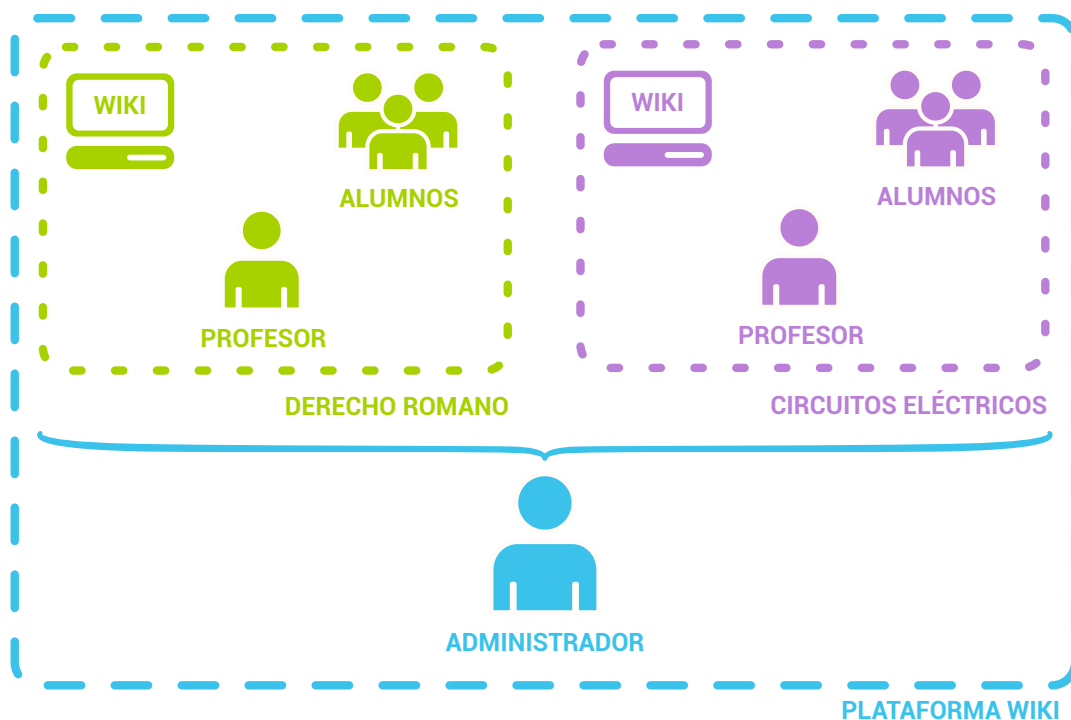


Figura 2.2: Distintas comunidades de práctica trabajando conjuntamente en su propio dominio. El administrador de la plataforma es el encargado de que todo funcione correctamente.

- *Comunidad*: es aquí donde los miembros establecen las relaciones que les permiten aprender y, a su vez, exponer su experiencia y su conocimiento en un tema determinado.
- *Práctica*: los miembros forman parte del proceso mismo, participando en actividades de manera conjunta y donde se comparten los intereses de la comunidad.

Como las comunidades de práctica se pueden dar en cualquier lugar es por eso que se mencionaba anteriormente que el *software wiki* las genera de manera inconsciente. Si lo pensamos bien, podemos llegar a la siguiente conclusión: el *software wiki* es utilizado de tal manera que es un *medio* que sirve como causa para lograr un *fin* que no es otro que el *dominio*, la *comunidad* y la *práctica*. (En la Figura 2.2 podemos ver a diferentes comunidades de práctica trabajando en grupos y que están implementadas sobre una plataforma *wiki* cualquiera.)

2. ESTADO DE LA CUESTIÓN

Otro aspecto importante que hay que tener en cuenta y que se puede ver analizando la Figura 2.2, son los roles y las interacciones que se producen en una comunidad de práctica. Wenger, Snyder y McDermott en el libro *Cultivating Communities of Practice: A Guide to Managing Knowledge* [47] sostienen que no sólo es significativo pensar en las tres características que conforman las comunidades de práctica, sino que hay que dar un paso adelante y analizar los diferentes perfiles que hay dentro de ellas. Esto se debe a que no todos los miembros de una comunidad de práctica participan en ésta de forma equitativa; cada persona tiene un nivel distinto de interés en el papel que le toca desempeñar. Estos son los diferentes roles o niveles de participación que podemos encontrar (clasificados de menor a mayor grado de implicación):

- *Agentes externos*: No pertenecen directamente a la comunidad pero muestran una inclinación en ésta, puede ser que como clientes o porque comparten algún punto de interés.
- *Miembros periféricos*: Este grupo pertenece a la comunidad y su característica principal es que participan en escasas ocasiones y son el grupo mayoritario. Normalmente suelen observar las interacciones de los miembros activos, del núcleo y de los coordinadores. Las motivaciones del porqué no participan suelen ser porque consideran que sus aportaciones no son importantes para la comunidad o que no poseen con la autoridad suficiente para que sean tenidos en cuenta. Otros, en cambio, consideran simplemente que no quieren participar porque no disponen de tiempo suficiente para ser más activos. Aún así, sus actividades periféricas, son sumamente importantes para el resto ya que gracias a su observación de todo lo que sucede dentro, consiguen obtener una visión amplia de todo lo que se expone.
- *Miembros activos*: Este grupo son aquellos que, según Wenger y col., suelen participar ocasionalmente en los foros u otras actividades de la comunidad pero sin llegar al nivel de intensidad de los miembros del núcleo. No suele ser un grupo muy amplio, ya que quizá no llegue a más del 15 ó 20 % de la comunidad.
- *Núcleo*: Es, por lo general, el grupo más compacto y pequeño que participa activamente en todas las actividades de la comunidad de una forma activa. Son los que guían a la comunidad a través de una agenda de actividades. Conforme la comunidad va madurando, este grupo suele tomar una buena parte del liderazgo

2. ESTADO DE LA CUESTIÓN

junto con el coordinador. Es un grupo minoritario que no comprende más del 10 % de los participantes.

- *Coordinador*: Es el encargado de organizar eventos y conectar a los diferentes miembros de la comunidad. Su contribución se basa en que la comunidad esté enfocada en su dominio, que mantenga relaciones entre sus miembros y otras comunidades, y que desarrolle su práctica. La dedicación de esta persona como coordinador es bastante alta, típicamente los valores oscilan entre 20 y el 50 % de su tiempo. Además, sus otras funciones principales son:
 - Vela por que la comunidad esté siempre activa y que se produzcan contribuciones de ésta a los demás miembros.
 - Relaciona informalmente miembros de la comunidad.
 - Planea y facilita eventos en la comunidad.
 - Contribuye en la construcción de la práctica. Esto incluye trabajar en la administración del conocimiento en la comunidad, lecciones aprendidas, mejores prácticas y métodos para el aprendizaje.
 - Identifica cuestiones o temas importantes en el dominio de la comunidad.
 - Posee una buena relación con el Núcleo y distribuye ciertas tareas con dicho grupo.

Saber identificar estos roles nos ayuda a entender como funcionan las comunidades de práctica internamente. También nos da una idea general de que características necesita una plataforma *wiki* para que la implementación sea lo más afín a la definición dada.

Por lo tanto, en el siguiente capítulo nos veremos en la necesidad de buscar un *software* que pueda implementar de manera fiable todo lo que aquí se ha explicado ya que la clave que sustenta al proyecto ALMA, son al fin y al cabo, las comunidades de práctica.

3

ELECCIÓN DE UN SOFTWARE WIKI



EN ESTE CAPÍTULO se analizarán las necesidades que tiene el proyecto ALMA y se explicarán las causas del porqué de la elección de TikiWiki como herramienta principal frente a otras alternativas similares. Para finalizar se dará una visión genérica a las características más importantes de dicho *software* y se informará al lector de los enlaces de interés para expandir su conocimiento sobre la plataforma.

3.1 ¿QUÉ NECESIDADES TIENE ALMA?

A lo largo de todo el capítulo anterior se han ido razonando las motivaciones que tiene ALMA siguiendo el modelo de las comunidades de práctica. Intuitivamente sabemos

3. ELECCIÓN DE UN SOFTWARE WIKI

que la índole de este proyecto queda englobada en un entorno educativo que involucra grupos de usuarios trabajando alrededor de una temática concreta. Por lo tanto, es lógico pensar qué tipo de herramienta vamos a utilizar para implementar el concepto y para ello nos debemos preguntar por las necesidades concretas que debe responder el *software*. A continuación, se listan los requisitos más importantes que consideramos a tener en cuenta:

- *Wikis*: Dado que este es el pilar fundamental en el que se basan las comunidades de práctica, necesitamos que el *software* que elijamos posea una implementación robusta de la filosofía *wiki*.
- *Gestión excelente de grupos y permisos asociados a ellos*: Una implementación correcta en la gestión de los grupos y sus permisos nos permite dividir las comunidades de práctica en distintos ámbitos (quizá no interese que los alumnos que pertenecen a una asignatura de “Circuitos Eléctricos” se mezclen con los alumnos que pertenecen a la asignatura de “Estadística”, ni que tampoco puedan modificar los elementos sobre los que estén trabajando estos últimos).
- *Facilidad en la instalación, mantenimiento y manejo*: Buscamos que nuestro *software* no posea ninguna complejidad a la hora de instalarlo, que use componentes robustos y que, en general, el manejo sea sencillo (que no tiene porqué significar necesariamente que sea simple).
- *Automatización*: Es probable que si la plataforma es usada por mucha gente dicha gestión, por parte de un administrador, deje de ser algo trivial (asignación de permisos, creación de nuevas comunidades de práctica, establecer usuarios a diferentes grupos...).
- *Robustez*: ¿Podemos estar seguros de que la plataforma soportará una gran cantidad de usuarios?
- *Que sea adaptable*: El *software* debe permitir crear cualquier estructura posible. En nuestro caso el concepto de las comunidades de práctica como se mostró en la Figura 2.2 en el Capítulo 2 (pág. 5).

3. ELECCIÓN DE UN SOFTWARE WIKI

3.2 ¿QUÉ HERRAMIENTAS EXISTEN?

En el mercado actualmente podemos encontrar muchas propuestas que nos permiten implementar de múltiples maneras y formas un sistema de aprendizaje virtual (*e-learning*). La denominación que reciben este tipo de *software* es: Learning Management System o Sistema de Gestión de Aprendizaje (SGA). Y es un *software* que se instala en un servidor web y que se emplea para administrar, distribuir y controlar las actividades de formación no presencial de una institución o organización [52].

Las principales funciones del SGA son: gestionar usuarios, recursos, así como materiales y actividades de formación. También se encargan de administrar el acceso al mismo, controlar y hacer seguimiento de los procesos de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión... En cambio, generalmente, no incluyen posibilidades de autoría, es decir, que se pueda crear contenidos propios, sino que se centra en gestionar contenidos creados por fuentes diferentes.

ALMA, *per se*, no utiliza todas las características a las que se adscriben los SGA. Por ejemplo, el hecho de que no suelen incluir contenidos generados por los propios usuarios es algo que contradice expresamente la naturaleza de este proyecto. Este pequeño detalle, como veremos más adelante, nos hizo descartar algunas propuestas por no cumplir con los requisitos necesarios.

Por lo tanto, si ALMA no se puede categorizar dentro de un SGA propiamente dicho (aún compartiendo muchas de sus ideas), ¿qué es entonces? Podríamos responder que es un híbrido entre un SGA y un Content Management System o Sistema de Gestión de Contenidos (SGC) [49].

Un SGC es un programa que, al igual que un SGA, se puede instalar en un servidor web y permite, de manera sencilla, la creación y la administración de contenidos (principalmente para ser consumidos en páginas web) por parte de los administradores, editores, participantes y demás roles.

En ALMA los contenidos son creados por los propios usuarios, otras veces, por alguien externo. Por lo tanto, ¿qué propuestas encontramos en el mercado que más o menos se ajusten a un híbrido de un SGA y un SGC? Se listan a continuación las opciones más relevantes:

3. ELECCIÓN DE UN SOFTWARE WIKI

- WebCT o BlackBoard
- Moodle
- MediaWiki
- TikiWiki

WebCT, conocido también como BlackBoard [26], es, por definición expresa, el sistema que ha determinado, tal y como lo conocemos, las bases de los SGA de manera concreta y del aprendizaje virtual (*e-learning*) en general. Nuestra universidad lo usa en múltiples cursos y asignaturas de las diferentes carreras que se imparten [45].

No negamos que se trata de un *software* que es bastante competente en su terreno, pero la mayor desventaja que tiene es que su licencia es privativa y por ello el usuario no tiene ningún derecho a poder copiar, estudiar, modificar y redistribuir el código de la plataforma [40, pág 2]. Es más, tendríamos que pagar una licencia por el uso de la misma. Por lo tanto desde el principio consideramos que no era una opción plausible.

La alternativa más conocida a WebCT es Moodle [31]. Posee muchas de las características del primero más algunas suyas propias. Nuestra universidad también lo usa en determinadas asignaturas [3]. Sin embargo, tampoco es una opción a tener en cuenta porque principalmente está centrado en distribuir cursos (o lecciones) provenientes de una fuente central (el profesor por ejemplo) y unos alumnos trabajando sobre esos materiales. Realmente se pueden ver como una ayuda a la clase, o como una forma de sustituirla, pero no para complementarla de manera creativa a como lo propone el concepto de las comunidades de práctica. Por lo que también quedó descartada.

MediaWiki [30] es quizá la más conocida de las expuestas, puede que no de manera directa, pero es el *software* que corre la Wikipedia [48]. Como su propio nombre indica se centra en las *wikis* de manera exclusiva (lo cual es bastante positivo porque es uno de los requisitos que pedimos para ALMA). Otro aspecto es que se puede instalar muy fácilmente en cualquier servidor que cumpla unos requisitos mínimos. Además posee una comunidad de usuarios bastante amplia y hay muchos manuales que explican como configurar y modificar el propio *software*. Sin embargo, como características negativas es que la gestión y la administración de los usuarios no es tan trivial, tampoco se pueden crear grupos aislados de usuarios ni controlar los accesos a las diferentes páginas *wikis*

3. ELECCIÓN DE UN SOFTWARE WIKI



Figura 3.1: Logotipo de Tikiwiki.

de una manera selectiva, y no hay posibilidad de automatizar ciertas tareas de gestión rutinarias. Por lo tanto es otra opción que descartamos.

3.3 ¿POR QUÉ SE ESCOGIÓ TIKIWIKI?

De las cuatro opciones que se han presentado, la última que queda es TikiWiki. Fue escogida por los siguientes motivos (responde a muchas de las peticiones que pusimos antes y añade más características que no habíamos pensado pero que, sin embargo, nos son útiles):

- *Versátil*: El diseño de TikiWiki es lo que se conoce como un “todo-en-uno”, es decir, aglutina muchas características en un sólo *software*. Esto es positivo porque actualmente ALMA utiliza las *wikis* como medio para crear las comunidades de práctica, pero, en algún futuro se podrían utilizar otras herramientas que sirviesen para el mismo propósito, por ejemplo, foros de discusión. Al tener TikiWiki integradas las *wikis*, los foros, los *blogs*, galerías de imágenes, mapas... dentro del propio *software*, nos olvidamos de tener que instalar complementos (*plug-ins*) u otras herramientas externas para conseguir el mismo efecto.
- *Fiable*: Muchos proyectos de código abierto (y otros tantos comerciales) utilizan TikiWiki como motor principal para gestionar sus *wikis* (y otras necesidades). Como ejemplo importante que utilice TikiWiki encontramos al motor de 3D Ogre [19, 15].
- *Flexible*: TikiWiki se puede adaptar a cualquier necesidad ya que, al incorporar una gran cantidad de opciones, puede ser utilizado como una *wiki*, como un *blog* para

3. ELECCIÓN DE UN SOFTWARE WIKI

una persona, un gestor de errores en programación (*bug tracker*), una comunidad de usuarios, un portal de noticias, una web corporativa [2]... La imaginación es nuestro límite.

- *Profiles*: Como apoyo al punto anterior, y debido a que TikiWiki posee más de 1500 opciones de configuración distintas, surgieron los *Profiles*. Éstos permiten modificar cualquier opción posible de la plataforma de TikiWiki de una manera sencilla y eficaz utilizando ficheros de “texto plano”. Esta fue la característica principal de la elección de TikiWiki como *software* y con la que se consiguió implementar ALMA con éxito.
- *Perspectivas*: Otra de las habilidades de este *software* que destaca por si sola es que permite, por ejemplo, que para un mismo usuario la experiencia usando el *software* de TikiWiki sea completamente diferente dependiendo de las acciones que realice.
- *Excelente gestión de usuarios y de permisos*: Con TikiWiki se obtiene una jerarquía de permisos muy fina y detallada, se pueden aplicar de manera global a todos los usuarios, o pueden ser concretos tanto a un usuario como a un grupo o bien a un recurso (una *wiki*) permitiendo dar acceso o denegarlo acorde a las necesidades que se tengan.
- *Facilidad de instalación*: Para funcionar TikiWiki sólo necesita un servidor capaz de ejecutar código en PHP Hypertext Preprocessor (PHP) 5 [27] y una base de datos MySQL [32] para guardar la información.

Información de interés sobre TikiWiki

A continuación dejamos al lector interesado en obtener más información una serie de enlaces importantes sobre TikiWiki y que varían desde la descarga de el propio *software* hasta casos complejos de configuración, a saber:

- Información básica de la plataforma (<http://tiki.org>): Aquí podemos descargar el *software* y obtener información básica de sus características principales. Es útil para averiguar si TikiWiki se ajusta a nuestras necesidades o no.

3. ELECCIÓN DE UN SOFTWARE WIKI

- Documentación genérica (<http://doc.tiki.org>): Esta página está orientada a la obtención de información por parte del administrador de la plataforma. Cualquier necesidad que surja a la hora de configurar el *software*, aquí es probable que se encuentre la respuesta.
- Desarrollo (<http://dev.tiki.org>): Esta *wiki* está preparada para los desarrolladores del código ya que se muestran las últimas novedades que se producen en el código fuente de TikiWiki, así como también nuevas características que se añadirán a versiones futuras, implementaciones que se han realizado y en general cualquier tipo de documentación útil para la gente que quiera programar.
- *Profiles* (<http://profiles.tiki.org>): Aquí se puede encontrar toda la información importante sobre la información relacionada con la configuración, desarrollo y creación de los *Profiles*. También es un punto de intercambio de nuevos *Profiles* que hayan creado otros usuarios. Instamos al lector interesado a que visite esta página web para obtener una información más detallada sobre esta herramienta.
- Canal de IRC (<http://webchat.freenode.net/?channels=tikiwiki>): En este canal de IRC se alojan los principales “sabios de TikiWiki” que nos pueden ayudar siempre que lo necesitemos.

4

CONCEPTOS FUNDAMENTALES DE TIKIWIKI



EN LOS CAPÍTULOS ANTERIORES se han descrito conceptos tan importantes como: qué es ALMA, qué son las comunidades de práctica y, sobre todo, qué tipo de *software* nos puede ayudar a conseguir nuestro objetivo. En este capítulo nos centraremos en dar una visión detallada de cómo se puede implementar una comunidad de práctica con las herramientas que tiene disponible por defecto TikiWiki (sin utilizar, todavía, ningún tipo de automatización con *Profiles*). Esto nos ayudará a, por una parte, tener una visión genérica de las capacidades que nos brinda el *software* (y hasta que punto podemos “adaptarlo” a nuestras necesidades) y por otra, a darnos cuenta de que crear comunidades de práctica es una tarea repetitiva y tediosa para un administrador.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI



Figura 4.1: La comunidad de práctica de la asignatura de Derecho Romano.

4.1 ¿DE QUÉ MANERA PODEMOS IMPLEMENTAR

UNA COMUNIDAD DE PRÁCTICA CON TIKIWIKI?

Si recordamos, en la Figura 2.2 del Capítulo 2 (pág. 5) aparecían múltiples comunidades de práctica (en recuadros de color verde y lila, las asignaturas de “Derecho Romano” y “Circuitos Eléctricos”, respectivamente) dentro de una plataforma wiki que era dirigida por un administrador (el recuadro azul). A continuación, en la Figura 4.1, se ha cogido del ejemplo anterior la comunidad de práctica de “Derecho Romano” y será sobre ella donde se ilustrarán los conceptos más importantes que debemos de conocer de TikiWiki para implementar las comunidades de práctica. Hay que añadir que toda información que se presente aquí no pretende ser exhaustiva en cuanto a demostraciones de como administrar la plataforma, ya que, no es el objetivo principal de este PFC hacerlo. Si bien cuando se considere oportuno, se instará al lector a buscar más información dentro de los enlaces presentados en el capítulo anterior y de los que se presenten en éste. ¡Comenzamos!

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Concepto de recursos

Un *recurso* no es, ni más ni menos, que cualquiera de las herramientas que nos pone TikiWiki a nuestra disposición (*wikis*, foros de discusión, galerías de imágenes...) y que puede ser utilizado por las comunidades de práctica para implementar sus actividades (Figura 4.2). La creación de recursos en TikiWiki es bastante sencillo, ya que, basta con que un administrador acuda a la página de administración de la plataforma (por defecto: <http://localhost/tiki-admin.php?page=features>¹) y habilite las características que desea. (En la Figura 4.3 vemos como se pueden activar y desactivar los diferentes recursos que posee la plataforma.) Una vez que el administrador ha habilitado los recursos que desea, puede visitar la página: <http://localhost/tiki-admin.php> y configurar las opciones pertinentes a dicho recurso (Figura 4.4).

Cabe añadir que dichos recursos son globales, es decir, sin haber cambiado ninguna opción más de la plataforma, se crearía una sola *wiki* o una sola galería de imágenes o un foro. Estos elementos serían comunes a todos los usuarios (no habría todavía ningún concepto de comunidades de práctica).

Concepto de grupos

Los *grupos* (como su propio nombre indica) se utilizan en TikiWiki como un contenedor donde se pueden agrupar, utilizando algún criterio de clasificación, a distintos usuarios de la plataforma. Se utilizan para múltiples propósitos pero, entre ellos, el más importante es que se les puede otorgar (o eliminar) permisos de acceso a los diferentes recursos (por ejemplo: que un determinado grupo pueda editar una página *wiki* pero que a su vez no pueda subir imágenes a ésta). Esta característica nos es útil, en nuestro caso, porque podemos crear múltiples grupos que engloben a los diferentes roles que existen (profesores y alumnos) y asignarles a éstos diferentes permisos de acceso a los recursos (Figura 4.5 y Figura 4.6).

Por defecto, TikiWiki genera tres grupos importantes que son tratados de manera especial por el *software*: Anonymous, Registered y Admins. Éstos engloban a cualquier usuario de la plataforma: al primero, pertenecen todos aquellos usuarios que están

¹Asumimos que tenemos TikiWiki correctamente instalado, funcionando con una base de datos habilitada para tal efecto y en nuestro caso se está ejecutando en un ordenador local, de ahí que se use <http://localhost> como dominio principal de la dirección de ejemplo [13].

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

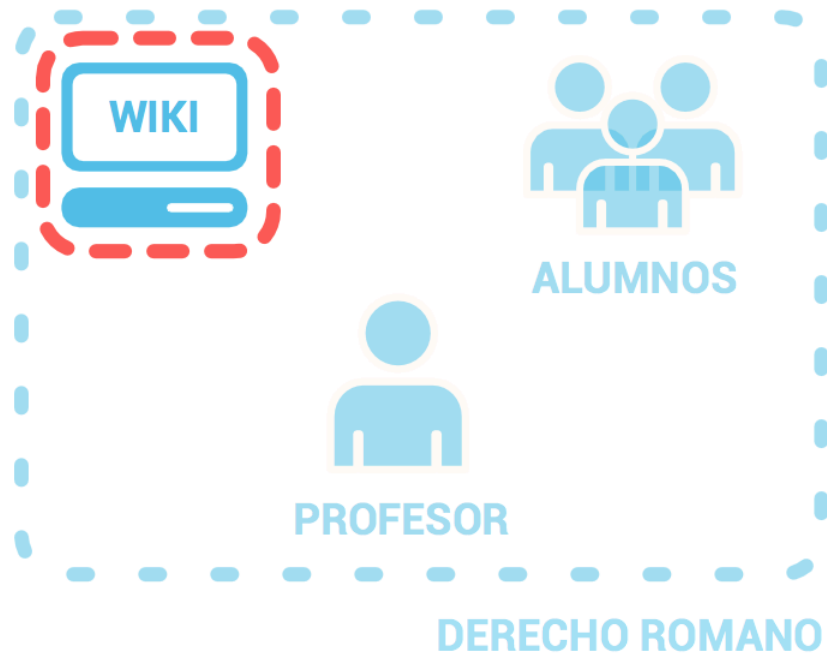


Figura 4.2: Los recursos son aquellas herramientas que TikiWiki nos pone a nuestra disposición: Wikis, Foros, Galerías de Imágenes ...



Figura 4.3: Panel de habilitación de recursos en TikiWiki.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Administration

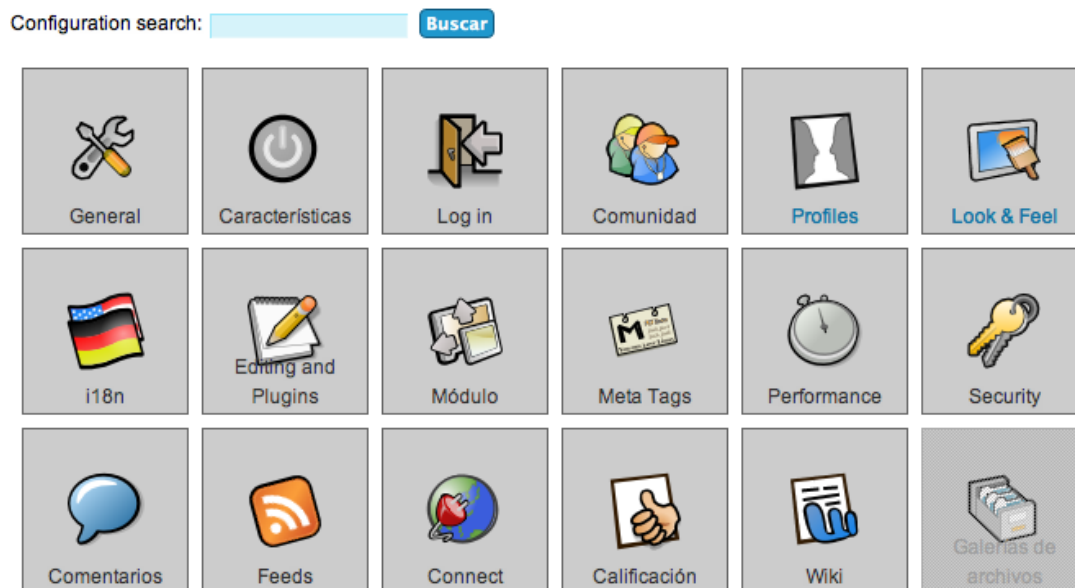


Figura 4.4: Panel de administración principal. Nos indica con transparencias si un recurso está habilitado o no (en este caso en concreto el recurso “galería de archivos” está desactivado). Si pinchamos en el icono del recurso nos lleva a su página de configuración.

visitando TikiWiki pero que no se han registrado o que no han iniciado una sesión en la plataforma; al segundo, si hacemos una traducción directa del inglés, pertenecen todos aquellos que si han iniciado una sesión en el *software*; al tercero, pertenecen todos aquellos usuarios que tienen roles de administradores, es decir, los que poseen el máximo poder administrativo de la propia plataforma.

También hay que destacar que otra de las características que poseen los grupos en TikiWiki es que se pueden crear unos dentro de otros (y montar así una jerarquía de grupos). Esto permite el que se pueda concebir un grupo cualquiera con unos permisos genéricos y luego que otros grupos hereden dichos permisos e ir restringiéndolos según se necesite. Se evita duplicar esfuerzos aunque, en nuestro caso, no utilizamos esta característica para implementar las comunidades de práctica.

Para más información relativa a los grupos podemos visitar: <http://doc.tiki.org/Groups>.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

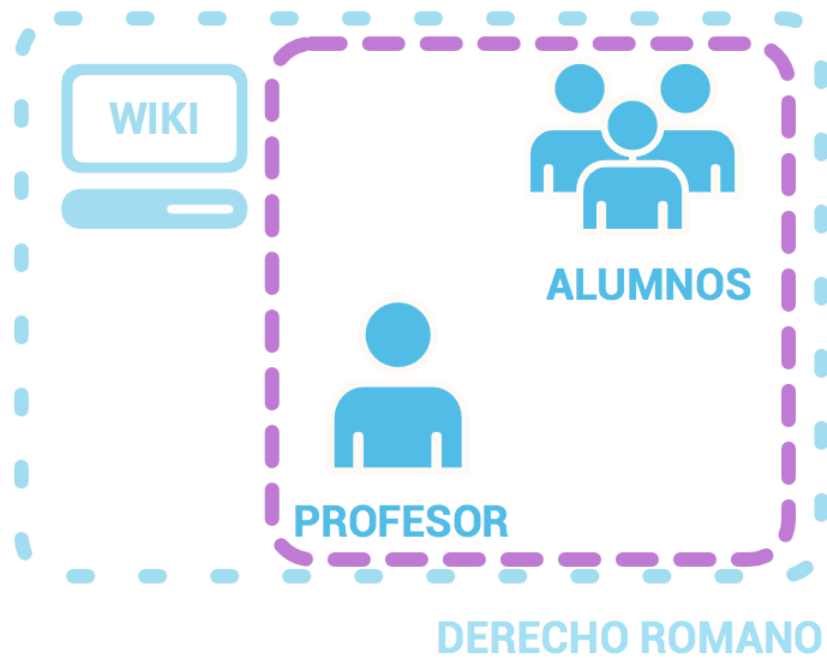


Figura 4.5: Los grupos nos permiten aunar usuarios según preferencias. Pueden ser tan granulares como deseemos y un usuario puede pertenecer a uno o varios grupos.















ID	Nombre	Descripción	Inherits Permissions from	Inicio	User Choice	Permisos
 3	Admins			n	 1	
 1	Anonymous			n	 3	
 4	Circuitos Electricos Alumnos			n	 14	✗
 5	Circuitos Electricos Profesores			n	 27	✗
 6	Estadística Alumnos			n	 14	✗
 7	Estadística Profesores			n	 27	✗
 2	Registered		Anonymous	n	 0	

Figura 4.6: Panel de administración de grupos.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI



Figura 4.7: Conjunto de recursos agrupados en una categoría y que pertenecen a la comunidad de práctica de Derecho Romano.

Concepto de categorías

Al igual que con los grupos podíamos asociar a usuarios en distintos conjuntos, las *categorías* es un concepto similar que funcionan de la misma manera, sólo que en lugar de agrupar usuarios permiten clasificar los recursos de TikiWiki. Cualquier recurso que existe en la plataforma es susceptible de ser categorizado incluyendo: *blogs*, galerías de imágenes, artículos, encuestas, foros, páginas *wiki*, galerías de archivos, etcétera. Y al igual que con los grupos, un recurso puede pertenecer a una o varias categorías y que cada categoría tenga sus propios permisos individuales (Figura 4.7).

El principal uso que se le da a las categorías es para controlar el acceso a un conjunto específico de recursos. Al establecer permisos concretos en una categoría, lo que hace es que todos los permisos globales que tenían los recursos por si solos desaparecen y son sobrescritos por los que hay en la categoría (en la siguiente sección se verá de manera más detallada como se aplican los permisos).

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

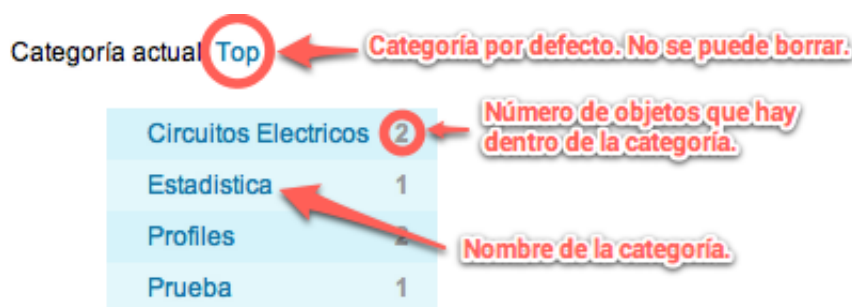


Figura 4.8: Listado de todas las categorías que se han creado en TikiWiki.

Las categorías también se pueden utilizar para ayudar en la navegación del sitio o para crear un conjunto de recursos relacionados por algún criterio. Por ejemplo: en la Figura 4.8 observamos que hay una categoría llamada “Circuitos Eléctricos”, dentro de ella estarían los recursos que pertenecen a dicha comunidad de práctica.

Como sucedía con los grupos, TikiWiki crea una categoría especial denominada Top que es la categoría madre donde todas las demás dependerán de ella, es decir, heredarán de esa categoría. También es posible crear categorías dentro de categorías y obtener así un árbol jerárquico (aunque nosotros no exploraremos esa opción ya que no nos es necesario para poder crear las comunidades de práctica).

Las categorías en TikiWiki no vienen habilitadas por defecto, así que es tarea del administrador decidir si desea tenerlas activas o no. Para administrirlas se debe de ir a la siguiente dirección: http://localhost/tiki-browse_categories.php y ahí nos encontraremos con el panel de la Figura 4.8. Si se pincha en una categoría se podrá obtener información más detallada sobre ésta, los recursos que tiene asociados y los permisos que posee... (Figura 4.9)

Para averiguar más sobre las categorías podemos visitar: <http://doc.tiki.org/Category>.

Concepto de permisos

Los *permisos* son una de las herramientas más poderosas que posee TikiWiki y es con ellos con los que podemos crear, junto con las perspectivas § 4.1 (pág. 38), el concepto de las comunidades de práctica. Un administrador de la plataforma puede decidir quién y bajo qué circunstancias un usuario o un grupo puede acceder a un recurso y modificarlo. El esquema de permisos de TikiWiki, sobre todo al principio, puede parecer un poco

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI



Figura 4.9: Vista detallada de la categoría Circuitos Eléctrico y sus recursos asociados.

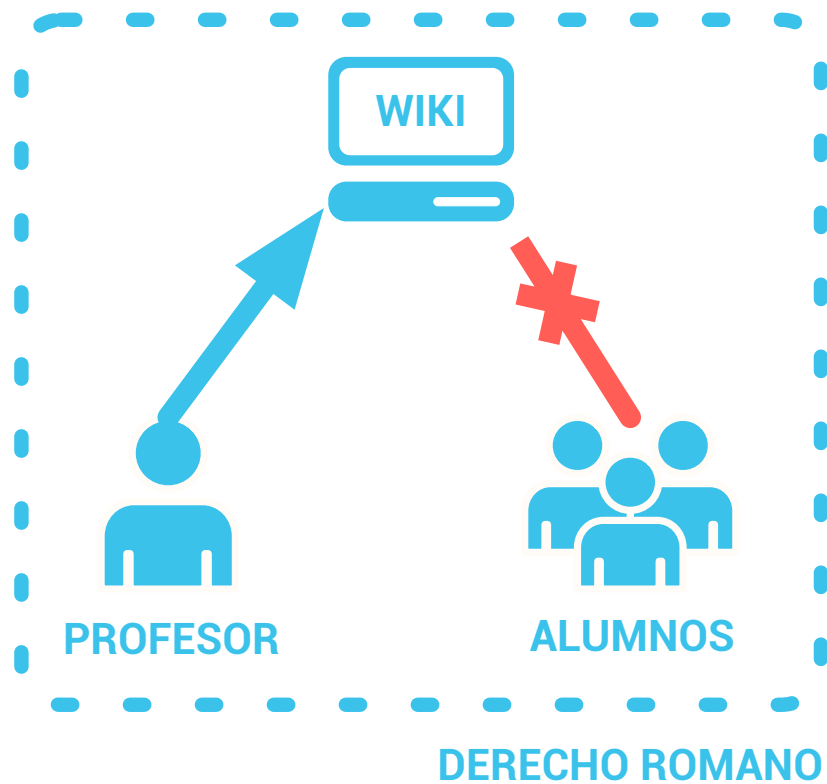


Figura 4.10: Concepto de lo que son los permisos en TikiWiki.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

caótico (en parte debido a como se ha implementado el panel de administración), pero, una vez acostumbrados podemos personalizar la plataforma a nuestro gusto de maneras que no habíamos imaginado. Se intentará dar la mejor explicación posible aunque, no obstante, la mejor forma de aprender sobre esta herramienta es realizando pruebas sobre una instalación de TikiWiki destinada para tal efecto.

Para entender como funcionan los permisos antes hay que asumir una serie de ideas (muchas de ellas ya han sido expuestas en las anteriores secciones, pero, se volverán a repetir con el fin de clarificar al máximo los conceptos):

- Referente a los recursos, es decir, las *wikis*, los *blogs*, los foros... pueden tener permisos de manera individual a los que tengan los grupos y las categorías.
- Referente a las categorías:
 - Los recursos pueden ser categorizados.
 - Una categoría se puede asignar a un grupo específico.
 - Los permisos que se le otorguen a una categoría garantiza a los miembros del grupo el derecho a poder ver y a editar el contenido de dicha categoría.
 - Una categoría puede pertenecer a otra categoría.
 - Si a una categoría no se le especifican permisos concretos, se aplican los genéricos.
- Referente a los grupos:
 - Cada grupo puede tener un acceso personalizado a todas las características de TikiWiki.
 - Los usuarios pueden estar en uno o en múltiples grupos.
 - Los grupos pueden tener “grupos dentro de otros grupos”.
 - Los permisos se añaden a los grupos en general, NO a los usuarios.
 - Si a un grupo no se le especifican permisos concretos, se aplican los genéricos.
 - Por defecto, TikiWiki genera tres grupos de manera predeterminada:
 - Anonymous: Los usuarios que no tienen una cuenta en la plataforma o que no han iniciado una sesión con su usuario pertenece a este grupo.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

- Registered: Cualquier usuario que ha iniciado una sesión en el sistema, independientemente de que pertenezca a otro grupo, por defecto pertenecen a este grupo.
- Admins: Grupo encargado de la administración, como ya comentamos anteriormente.



En la documentación oficial podemos encontrar un listado de todos los permisos que posee TikiWiki [16] (conviene visitar ese enlace para tener una idea aproximada de la cantidad de permisos que existen). El formato que utilizan para definir un permiso es de la siguiente manera: `tiki_p_ + nombre_del_permiso`. Ponemos tres ejemplos:

- *tiki_p_edit*: Permiso que permite editar una página *wiki*.
- *tiki_p_admin*: Permiso que garantiza a un usuario derechos de administración.
- *tiki_p_upload_picture*: Permiso que permite subir una imagen a una página *wiki*.

Si queremos tener éxito a la hora de manejar los permisos, es importante conocer el orden en el que se deben aplicar ya que, si no, no obtendremos los resultados deseados (y sucede con bastante frecuencia). A continuación, se muestran éstos ordenados de mayor a menor importancia (Figura 4.11):

- *Recurso*: Los permisos de un recurso definen que acciones puede realizar un usuario para un recurso en concreto. Por ejemplo, podríamos tener una página *wiki* que NO permitiese subir ficheros, o que NO permitiese la edición de su contenido, y dichos permisos serían los que imperasen, independientemente de que dicha *wiki* estuviese categorizada en una categoría que permite editar contenidos *wiki* o subir ficheros o de que el grupo, en general, pueda hacerlo.
- *Categoría*: Los siguientes permisos en tenerse en cuenta son los relativos a que el recurso pertenezca a una categoría. Si, por ejemplo, en la categoría se describe que NO se pueden subir imágenes a las páginas *wikis* de esa categoría, y en cualquier página *wiki* no existe un permiso que contradiga dicha regla, entonces se podrán

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI



Figura 4.11: Orden en el que se aplican los permisos en TikiWiki. Imagen inspirada en la documentación oficial http://twbasics.keycontent.org/img/wiki_up/tiki_permissions.png.

subir imágenes. Recordamos que no puede existir un recurso que no esté asignado a ninguna categoría ya que, por defecto, pertenece a la categoría Top y dicha categoría también tiene unos permisos por omisión (que podemos modificar si así deseamos).

- *Grupo (y los permisos globales)*: Los permisos de grupo son los últimos en aplicarse y son los que menos preferencia tienen en caso de que existan permisos contradictorios en los de categoría y los de recurso. Por otra parte, como cualquier usuario pertenece a un grupo, bien sea Registered o Anonymous sin contar con que esté en otros grupos se garantiza que, como mínimo, haya unos permisos globales básicos.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Admin categorías



Figura 4.12: Listado de todas las categorías disponibles. Para editar los permisos de una categoría hay que pinchar en el icono de la llave verde.

Permisos				
Filtrar: <input type="text"/> <input type="button" value="Toggle sections"/> <input type="checkbox"/> Show only selected				
<input type="checkbox"/> Anonymous	<input type="checkbox"/> Registered	<input type="checkbox"/> Circuitos Electricos Alumnos	<input type="checkbox"/> Circuitos Electricos Profesores	Permiso
▼ category				
You might want to also set the tiki_p_modify_object_categories permission under the tiki section				
Anonymous	Registered	Circuitos Electricos Alumnos	Circuitos Electricos Profesores	Permiso
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Can admin categories (<i>tiki_p_admin_categories</i>)
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Can remove objects from the category (needs <i>tiki_p_modify_object_categories</i>) (<i>tiki_p_remove_object</i>)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Can see the category in a listing (<i>tiki_p_view_category</i>)
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Can create new categories (<i>tiki_p_create_category</i>)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Can add objects to the category (needs <i>tiki_p_modify_object_categories</i>) (<i>tiki_p_add_object</i>)

Figura 4.13: Listado de los posibles permisos que se pueden asignar a la categoría (en la imagen no se muestra toda la información). Como se puede apreciar, se pueden asignar permisos de manera independiente para cada grupo.

A modo de resumen de lo expuesto anteriormente podemos decir que, si no hay una contradicción en los permisos, estos se heredan de arriba a abajo, es decir, Grupo ▷ Categoría ▷ Recurso. En cambio, si hay algún permiso que contradiga alguno de los anteriores, éstos sobrescriben de abajo a arriba, es decir, Grupo ◁ Categoría ◁ Recurso.

Si queremos obtener más información para administrar los permisos de las categorías, debemos acudir a la siguiente dirección: http://localhost/tiki-admin_categories.php.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Concepto de preferencias

Como ya se ha comentado anteriormente, TikiWiki posee más de 1500 configuraciones y opciones posibles divididas en varios paneles de administración. Las *preferencias* (como su propio nombre indica) nos permiten configurar la plataforma a nuestro gusto. Será una tarea del administrador decidir qué “características” desee que estén activas, pasando por los recursos, como por la configuración de los envíos de correo a los usuarios, que apariencia externa tiene la plataforma, etcétera.

Uno de los problemas que plantea TikiWiki respecto a la administración es que tiene demasiadas posibilidades de configuración. Hay demasiados paneles, demasiadas casillas para activar y desactivar y múltiples opciones. Normalmente la tarea de configurar la plataforma se suele realizar la primera vez que se instala en el servidor. Este proceso puede incluso hasta llevar varias horas en completarse dependiendo, claro está, de las necesidades concretas de cada instalación y de la pericia que tenga manejando el *software* el administrador.

Las principales desventajas que encontramos, aparte de la pérdida de tiempo que supone configurar, es que un usuario tiene que conocerse bien la plataforma (o al menos haber leído mucha información, o haber probado a base de ensayo y error). Además, las configuraciones no se pueden reutilizar, es decir, si disponemos de dos plataformas y queremos compartir la configuración que tenemos en una y llevarla a la otra de forma directa, no vamos a ser capaz de poder hacerlo.

Por fortuna los *Profiles* solventan de manera elegante todos estos inconvenientes y en el Capítulo 6 (pág. 66) veremos como se ha creado un *Profile* que configura ALMA en muy poco tiempo. Pero para ello, antes debemos conocer un poco el funcionamiento básico de las preferencias en TikiWiki.

De forma interna las preferencias se guardan en tres tablas importantes de la base de datos que hay que conocer:

- tiki_preferences
- tiki_perspective_preferences
- tiki_user_preferences

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

			name	value
<input type="checkbox"/>			browsertitle	ALMA
<input type="checkbox"/>			https_login	disabled
<input type="checkbox"/>			https_port	
<input type="checkbox"/>			error_reporting_level	0
<input type="checkbox"/>			error_reporting_adminonly	n
<input type="checkbox"/>			smarty_notice_reporting	n
<input type="checkbox"/>			log_tpl	n
<input type="checkbox"/>			feature_switch_ssl_mode	n
<input type="checkbox"/>			feature_show_stay_in_ssl_mode	n
<input type="checkbox"/>			language	es
<input type="checkbox"/>			lastUpdatePrefs	122
<input type="checkbox"/>			case_patched	y
<input type="checkbox"/>			tiki_version_last_check	1333045313
<input type="checkbox"/>			tiki_release	7.2
<input type="checkbox"/>			tiki_needs_upgrade	n
<input type="checkbox"/>			feature_wysiwyg	y
<input type="checkbox"/>			wysiwyg_default	y
<input type="checkbox"/>			wysiwyg_optional	y

Figura 4.14: Algunos valores almacenados en la tabla tiki_preferences.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

La principal y la más importante es la tabla `tiki_preferences`, ya que, es la encargada de almacenar los valores principales de todas las posibles configuraciones que pueden existir en TikiWiki. Dicha tabla almacena por ejemplo: el idioma que debe mostrar la aplicación, las “características” que hay habilitadas, la configuración para el envío de notificaciones de correo, el nombre de la plataforma... Es decir, cualquier opción que modifiquemos en el panel de administración queda almacenado ahí. La estructura relacional de esta tabla es sencilla: almacena el nombre de la preferencia y su valor asociado.

```
1 | CREATE TABLE 'tiki_preferences' (  
2 |     'name' varchar(40) NOT NULL default '',  
3 |     'value' text,  
4 |     PRIMARY KEY ('name')  
5 | ) ENGINE=MyISAM;
```

Figura 4.15: Estructura de la tabla `tiki_preferences`.

La librería de código que se encarga de almacenar y recoger los datos guardados en dicha tabla se llama `Prefslib.php` y dicho PHP se encuentra en la carpeta `lib/` al descargar el código fuente².

De forma genérica, y sin entrar en detalles muy concretos, cuando un usuario realiza una petición a TikiWiki esa librería extrae los datos de la tabla `tiki_preferences` y los almacena en un array asociativo [4]. De esta manera podemos obtener los valores que deseemos utilizando una simple sentencia de programación de PHP:

```
1 | <?php  
2 | print($prefs[ 'tiki_release' ]); // Y obtendríamos el número de versión de TikiWiki.
```

Figura 4.16: Un ejemplo de lo que contiene el array `$prefs` (que son todos los datos de la tabla `tiki_preferences`).

La tabla `tiki_user_preferences`, como su nombre indica, guarda, de manera análoga a como lo hace `tiki_preferences`, las preferencias que tienen los usuarios. Esto les permite configurar a cada uno que opciones desean de manera individual, es decir, cada

²Para más información sobre como está organizado el código fuente de TikiWiki recomendamos al lector la lectura del Apéndice B.1 (pág. 90) al final del documento.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

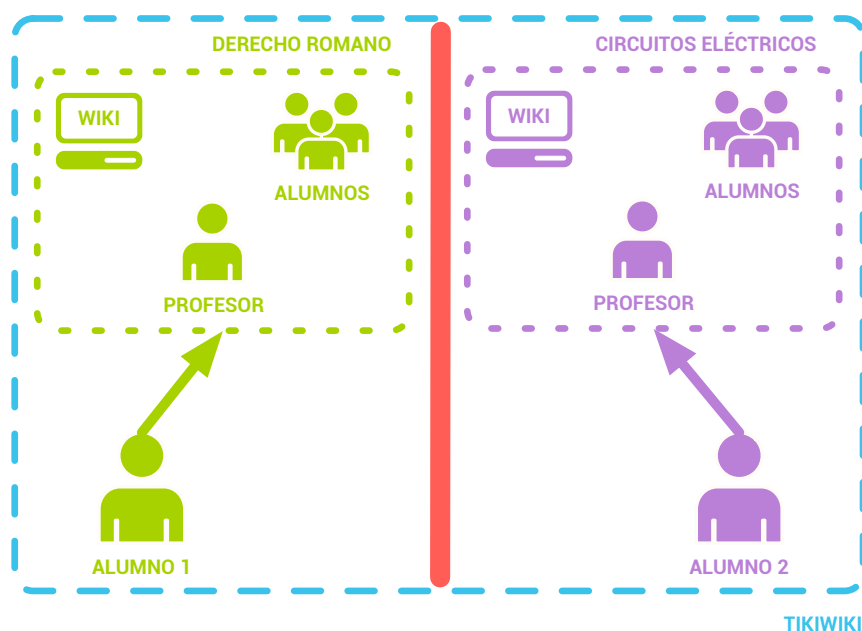


Figura 4.17: Ejemplo de dos posibles alumnos en el cual el Alumno 1 desconoce que existe la comunidad de práctica de Circuitos. Las perspectivas en TikiWiki nos permiten aislar de forma transparente comunidades de práctica dentro de una misma instalación.

usuario tiene sus propias preferencias. Por ejemplo: uno se podría llamar Pepito Pérez y su alias pperez y otro John Doe y su alias jdoe y esos datos se guardan en esa tabla.

La tabla `tiki_perspective_preferences` funciona de manera análoga a las anteriores, pero, su uso es diferente puesto que está ligado al concepto de las perspectivas. En la siguiente sección hablaremos más detalladamente de esta última tabla.

Tener un conocimiento de como se almacenan las preferencias y como se emplean nos ayuda posteriormente a poder desarrollar mejor los *Profiles* que configuren una instalación de TikiWiki.

Hay que decir que este tipo de información no se encuentra en ninguno de los canales oficiales presentados, por lo que recomendamos al lector interesado que explore de manera profunda lo que se ha presentado aquí si quiere obtener un conocimiento avanzado de como funciona internamente TikiWiki.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Concepto de perspectivas

Las *perspectivas* son un concepto que apareció en las últimas versiones de TikiWiki (concretamente en la versión 4) y con las cuales somos capaces de implementar la separación entre diferentes comunidades de práctica utilizando la misma instalación de *software*. Esto nos interesa mucho a la hora de implementar las comunidades de práctica, ya que, de esta manera podemos hacer que TikiWiki muestre solo cosas relevantes a un determinado grupo y no a otro (a un alumno concreto no le importa ver otras asignaturas que no sean a las que él está matriculado, Figura 4.17).

Además, aparte de separar diferentes comunidades de práctica, las perspectivas nos permiten personalizar TikiWiki y adaptarlo a las necesidades concretas de cada asignatura. Por ejemplo: en la vida real cuando un alumno entra en un aula puede encontrar un determinado color de pared, la distribución de las sillas tienen una forma determinada, la iluminación se manifiesta de una forma concreta... y todas estas características son completamente diferentes en otro aula. Podemos replicar de manera “psicológica” este mismo comportamiento cambiando por ejemplo el logotipo principal de TikiWiki por el de la asignatura Circuitos cuando el alumno entra dentro de la comunidad de práctica, o cambiar el color del tema visual que se está utilizando por ejemplo de azul a verde. Lo positivo es que estos “efectos” duran mientras el usuario esté dentro de una comunidad de práctica, cuando la abandona, los cambios se revierten a los que había previamente (al igual que cuando un alumno abandona un aula el entorno previo permanece igual, Figura 4.18).

El funcionamiento de las perspectivas es sencillo, simplemente sobrescriben temporalmente cualquiera de las 1500 preferencias que existen en TikiWiki. Recordamos al lector que las preferencias se guardaban en la tabla `tiki_preferences`. Éstas en el momento en el que se activan, cargan y ejecutan las preferencias modificadas de la tabla `tiki_perspective_preferences`. Por lo tanto es importante saber que cualquier preferencia que exista en la tabla `tiki_preferences` es susceptible de ser modificada por una perspectiva³.

Las perspectivas pueden ser administradas desde la interfaz de TikiWiki, si el administrador visita el siguiente enlace: http://localhost/tiki-edit_perspective.php se encontrará

³Esto es importante saberlo ya que a la hora de implementar *Profiles* en la documentación oficial de TikiWiki no se listan todas las opciones posibles. La mejor solución es acudir a la tabla de las preferencias y ver que valores hay ahí.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

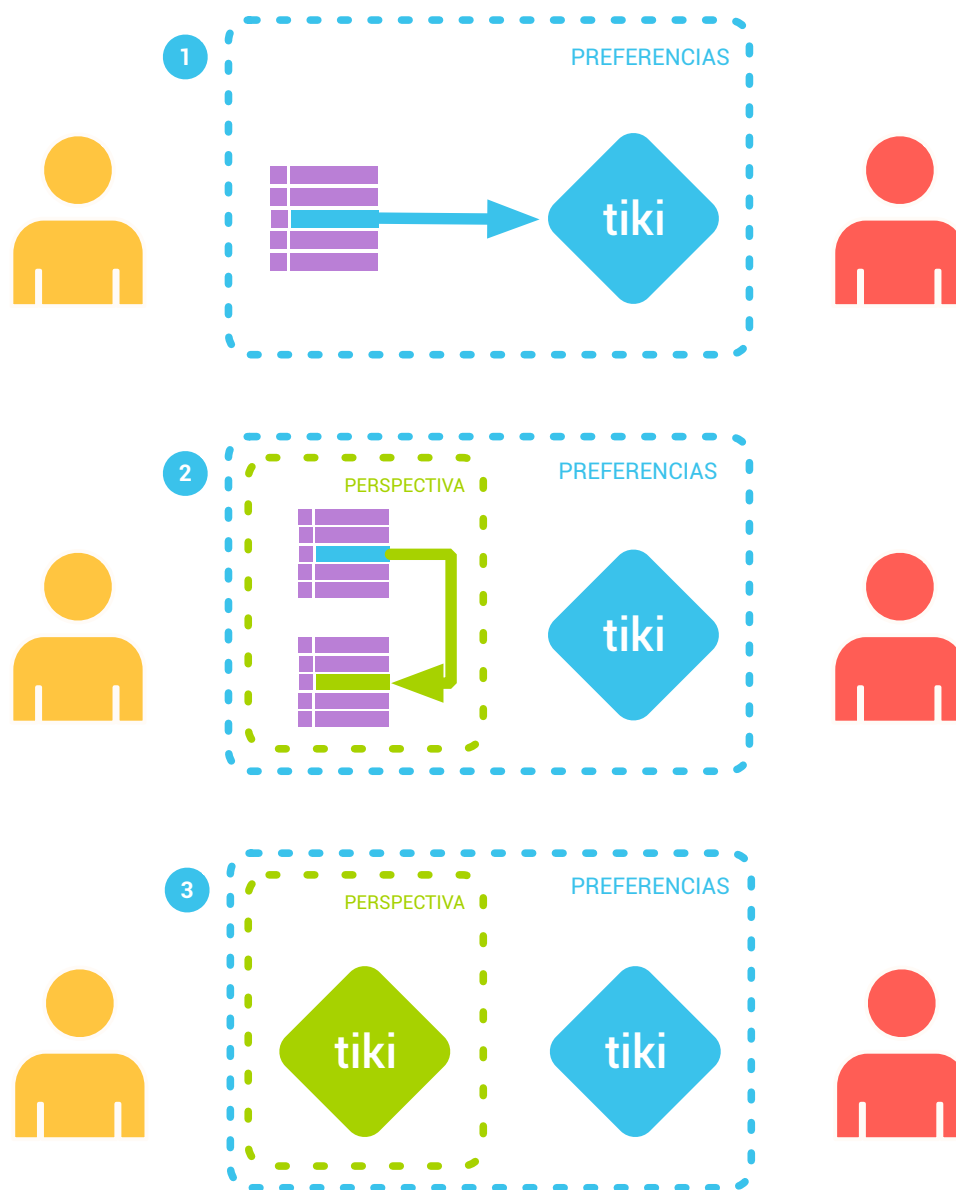


Figura 4.18: Supongamos que existen dos usuarios y que pertenecen a dos comunidades de práctica diferentes. En el primer ejemplo (denotado en el gráfico con el número uno) los dos ven el color del logotipo de TikiWiki azul, ya que, así lo indica sus preferencias genéricas. En un instante de tiempo $t + 1$ el usuario amarillo decide entrar en una comunidad de práctica. Dicha comunidad de práctica posee una perspectiva que cambia la preferencia de que el logotipo sea de color azul a que sea verde (se puede observar en el gráfico número dos como se ejecuta el cambio internamente). El usuario amarillo acaba entrando en la comunidad de práctica (gráfico número tres) y ya lo que él observa es que el logotipo ha cambiado de color, sin embargo, para el usuario de color rojo al no pertenecer a la misma comunidad de práctica (y, por consiguiente, al no haber ejecutado la perspectiva asociada) observa que el color del logotipo sigue siendo azul. En el momento en el que el usuario amarillo abandone la comunidad de práctica, todos los cambios que ha efectuado la perspectiva se revierten y se volvería a estar en el gráfico número uno.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI



Figura 4.19: TikiWiki permite crear, editar y borrar perspectivas. Hay que reconocer que la interfaz web no es la forma más cómoda de administrar las perspectivas, aunque para ediciones sencillas sirve perfectamente.

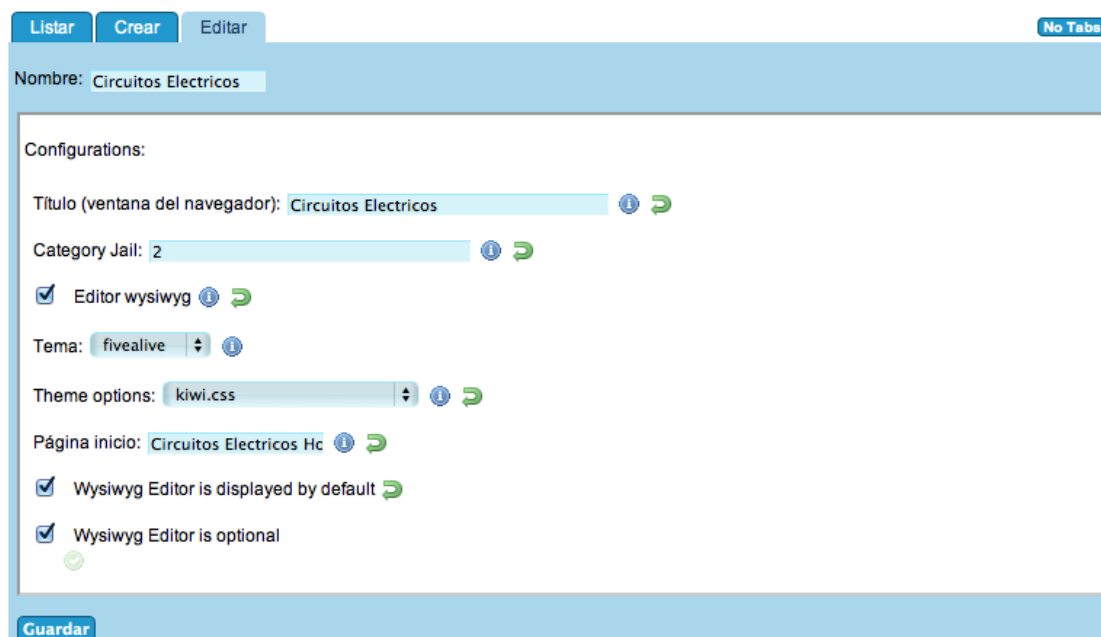


Figura 4.20: Aquí se listan las preferencias que se activarán en la comunidad de práctica de la asignatura de Circuitos Eléctricos. Han sido generadas de forma automática con un *Profile* (añadir preferencias de forma manual es un proceso tedioso).

con lo que muestra la Figura 4.19; si decide editar las preferencias que modificará una perspectiva entonces verá lo que muestra la Figura 4.20.

Para encontrar más información relativa a las perspectivas nada mejor que acudir a la documentación oficial <http://doc.tiki.org/Perspectives>.

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

Asignaturas

☐ Stay on this page

Default

Edit perspectives

Figura 4.21: Ejemplo de menú que se puede utilizar para cambiar las perspectivas y cambiar de comunidad de práctica de manera dinámica.

Menús

Create/Edit Menu

No Tabs

Buscar

Ir

ID	Nombre	Tipo	Opciones	Acción
42	Application menu Main extensive navigation menu	d	185	<div>RESET</div> <div></div>

Figura 4.22: Panel de administración de menús. Se pueden crear, editar y borrar. Hay una cantidad inmensa de opciones, más información en <http://doc.tiki.org/Menu+HOWTO>.

Concepto de menús

TikiWiki tiene un gestor de creación de menús de manera que podemos generar menús que en nuestro caso nos permiten cambiar de manera dinámica las perspectivas y listar a las comunidades de práctica a las que pertenece el usuario. Esto nos permite que el usuario cambie de comunidad de práctica de manera rápida y sencilla, y al activarse la perspectiva asociada a dicha comunidad, cambiará la experiencia en TikiWiki de manera acorde. Es como si un alumno cambiase de aula (Figura 4.21).

Por otra parte los menús tienen mucho más usos que para listar simplemente perspectivas. Si queremos administrar y crear menús, debemos ir a la siguiente dirección: http://localhost/tiki-admin_menus.php (Figura 4.22).

4. CONCEPTOS FUNDAMENTALES DE TIKIWIKI

4.2 ¿QUÉ HEMOS APRENDIDO DE TODO LO COMENTADO EN EL CAPÍTULO?

A lo largo de este capítulo hemos hecho un recorrido a las características más importantes que tiene TikiWiki que nos sirven para implementar una comunidad de práctica⁴. Si bien, en cada sección no se ha profundizado con mucho detalle, en parte, porque si no se podrían rellenar muchas hojas presentando conceptos que el lector será capaz de descubrir fácilmente en el momento que decida utilizar la plataforma. Aún así, podemos extraer una idea importante: crear comunidades de práctica conlleva mucho tiempo y es una tarea en la que se pueden cometer errores fácilmente, puesto que, la complejidad crece de manera lineal conforme haya más comunidades de práctica a crear. Y eso, sin mencionar de tener que administrar dichas comunidades. Por lo tanto, los *Profiles* se postulan como una herramienta útil que nos ahorra tener que crear todas estas opciones de manera manual. En la siguiente parte descubriremos todo lo necesario para poder conseguir nuestro objetivo principal: liberar al administrador de las tareas repetitivas y aburridas.

⁴De hecho en la documentación oficial de TikiWiki encontramos el concepto de Workspaces (<http://doc.tiki.org/Workspace>) que es básicamente lo que nosotros hemos denominado como comunidades de práctica. Y uno de los ejemplos es la creación manual de un Workspace similar a lo que hemos realizado aquí.

II

AUTOMATIZACIÓN DE RECURSOS CON PROFILES

PROFILES



A LO LARGO DE TODO este PFC se han mencionado los *Profiles* en múltiples ocasiones y lo útiles que son o lo bien que encajan con el concepto de crear comunidades de práctica de una forma rápida y cómoda para un administrador o docente. Pero es aquí, en este capítulo, en el que se tratará de explicar de manera profunda qué son y cómo funcionan. Para lograrlo, se ha hecho una división del mismo en dos partes fundamentales: en la primera, veremos la teoría básica que sustenta los *Profiles*, primero a vista de pájaro, es decir, entenderemos el concepto y aprenderemos a manejar la herramienta en TikiWiki utilizando para ello un ejemplo real; en la segunda, se darán breves detalles técnicos de la implementación interna, es decir, del código fuente responsable del funcionamiento por si el lector desea modificar la herramienta a sus necesidades concretas.

5. PROFILES

5.1 ¿POR QUÉ SURGEN LOS PROFILES?

TikiWiki, al ser un *software* muy completo, posee muchas características implementadas en su interior. Como usuarios de la plataforma podemos tomar ventaja de dicha arquitectura para construir casi cualquier cosa que deseemos. Pero, por norma general, resulta que muchas veces no queremos utilizar todas las herramientas que tenemos disponibles para implementar todos los posibles casos de uso que existen, si no que, dependiendo de nuestras necesidades específicas, modificamos la plataforma para realizar algo en concreto (utilizamos un subconjunto de ella). Por ejemplo: podríamos querer usar TikiWiki como un sistema para administrar nuestro *blog*, o para hacer una *wiki* corporativa para uso interno de la empresa en la que trabajamos, o para crear un foro de una comunidad de amantes del motor... Como ya se ha comentado anteriormente, somos nosotros quienes ponemos límites a lo que queremos realizar.

Pero, uno de los problemas que ha arrastrado TikiWiki desde sus primeras versiones y hasta hace bien poco es que, con todas las características que posee la plataforma, ¿cuál es la mejor forma de estructurar los paneles de administración de una manera eficiente y que promueva la facilidad de uso entre sus usuarios? La respuesta a esta pregunta es que los desarrolladores de TikiWiki todavía no lo han conseguido (si, incluso hasta en las versiones más modernas siguen adoleciendo de este problema). A lo largo de la vida del proyecto han modificado, en varias ocasiones, los paneles de administración para hacerlos más intuitivos, pero, 1500 opciones de configuración posibles son demasiadas opciones a tener en cuenta. Esto hace que cualquier usuario que quiera manejar la plataforma tenga que dedicar una considerable cantidad de tiempo en aprender las entrañas del programa. Además otro problema, análogo al anterior, es que no sólo existen los paneles de administración principales (como los que vimos en el Capítulo 4 (pág. 21)), si no que también hay otros paneles secundarios (también dedicados para propósitos administrativos) que están distribuidos por diferentes partes de la plataforma.

Debido a los inconvenientes expuestos, a partir de la versión 4 de TikiWiki, se decidió hacer una reescritura masiva de todas las librerías principales y se comenzó a trabajar con el proyecto de los *Profiles*. A día de hoy, y en la versión 8, todavía continúan integrando y mejorando esta herramienta. Lo que está claro es que es una parte importante de la

5. PROFILES

plataforma y así lo seguirá siendo en los próximos años, ya que, los *Profiles* solucionan el problema conocido como: *la paradoja de la elección*¹.

5.2 ¿QUÉ SON LOS PROFILES?

Un *Profile* no es más que un archivo de texto (que se encuentra almacenado en una página *wiki* normal y corriente) con una serie de instrucciones que explican, paso a paso, que deseamos modificar de cualquiera de las 1500 preferencias de TikiWiki, permitiéndonos también, el poder crear, cambiar y borrar cualquier recurso que queramos. Para ello, hay una herramienta dentro de la librería de los *Profiles* encargada de traducir el texto contenido en una página *wiki* en acciones concretas de la plataforma (Figura 5.1).

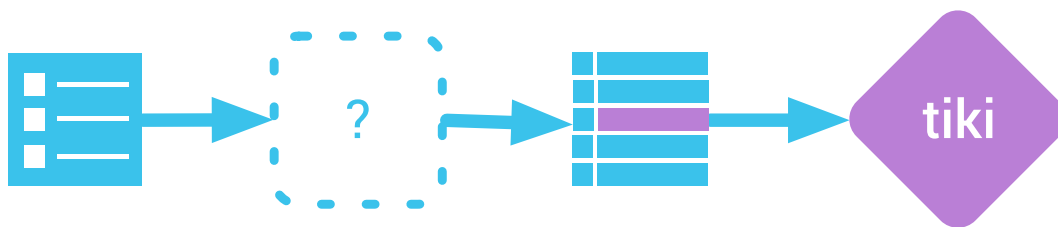


Figura 5.1: Un *Profile* modifica las preferencias de TikiWiki de la manera en la que esté especificado. El rectángulo con el interrogante es el traductor que se encarga de “entender” lo que el *Profile* desea alterar (veremos más detalles internos en la § 5.3 (pág. 48)).

Si recordamos nuestro periplo por el Capítulo 4 (pág. 21), buena parte de éste se dedicó a explicar muchos conceptos *críticos* de TikiWiki a la vez que se demostraban cuales eran los pasos que tenía que seguir un administrador para poder crear *una* comunidad de práctica. En lugar de tener que lidiar con los paneles de administración, una opción es que podemos concebir un *Profile* que explique, como si de una receta de cocina se tratase, como se debe crear una comunidad de práctica (Figura 5.2). (Y De esta manera podemos automatizar un proceso que es de índole repetitiva.)

La parte buena del asunto es que, con los *Profiles*, al ser realmente páginas *wiki* podemos hacer que la comunidad (o cualquier persona) participe en compartirlos o en mejorar

¹Esta teoría fue propuesta por Barry Schwartz en su libro *The Paradox of Choice: Why More Is Less* donde plantea que el exceso de opciones nos lleva a la parálisis y a la inacción. Y el problema posterior que surge: ansiedad, cuando hemos elegido [42].

5. PROFILES

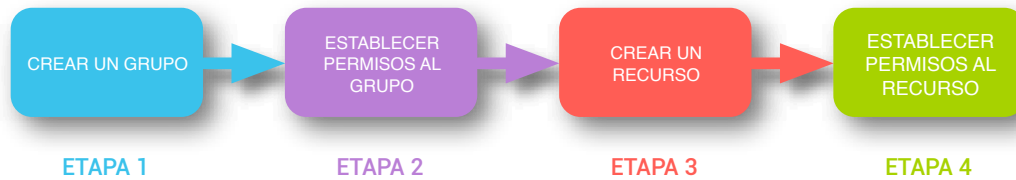


Figura 5.2: Para cada *Profile* hay que definir exactamente que deseamos ejecutar. Una forma de crear un *Profile* con éxito es dividir lo que queremos crear en etapas.

los ya existentes (y ya no sólo son útiles para crear comunidades de práctica si no para cualquier otra tarea). De esta forma se aumenta el número de casos de uso a los que podemos configurar TikiWiki y se facilita la reutilización de código ya que algo que deseemos realizar, con toda certeza, habrá sido pensado por alguien antes que nosotros.

Beneficios que aportan los Profiles a TikiWiki

En la documentación oficial de los *Profiles* se listan algunas de las características y ventajas que llevan consigo la utilización de dicha herramienta en TikiWiki (se han adaptado y expandido ciertas ideas, la mayoría se han expuesto anteriormente) [5]:

- Son sencillos de almacenar. Un *Profile* es un archivo de texto plano que puede ser almacenado en las propias páginas *wiki*.
- Facilitan la colaboración, ya que, al ser guardados en forma de páginas *wiki* podemos utilizar todas las ventajas que nos proporcionan estas herramientas, como por ejemplo: múltiples versiones de un *Profile* gracias al historial de ediciones que incorpora la *wiki*.
- Pueden modificar cualquier preferencia que exista en TikiWiki como todas las que se almacenan en la tabla `tiki_preferences` (§ 4.1 (pág. 34) en el Capítulo 4 (pág. 21)); también pueden crear nuevos recursos (§ 4.1 (pág. 23) del mismo capítulo) y asignar o modificar cualquier tipo de permiso de los existentes en TikiWiki en cualquier ámbito (ya sea de recurso, de categoría o de grupo).
- Permiten la ejecución simultánea de varios *Profiles* y se pueden ejecutar en cualquier momento que deseemos. Si alguno de los *Profiles* que ejecutemos en paralelo genera

5. PROFILES

conflictos con otro, la configuración del último ejecutado es prioritaria, es decir, evitan dejar a TikiWiki mal configurado.

- Composición. Los *Profiles* admiten adjuntar otros *Profiles* dentro de un *Profile*, es decir, se puede construir uno gigante con la unión de varios pequeños. También un mismo *Profile* puede ser dividido en múltiples fragmentos y entre ellos insertar sintaxis *wiki*.

Beneficios que aportan los Profiles a ALMA

Nosotros hemos decidido utilizar los *Profiles* por dos motivos principalmente:

1. Crear comunidades de práctica. Dado que este es el interés principal de este PFC, los *Profiles* encajan muy bien con lo que queremos hacer, puesto que, permiten, en un sólo segundo, tener preparado una comunidad de práctica para cualquier asignatura.
2. Configurar la plataforma de ALMA. Dado que el proyecto tiene unas necesidades muy concretas, la propia configuración de éste es una tarea que se realiza muy bien con un *Profile*. De esta forma podemos replicar múltiples “ALMAS” en poco tiempo.

5.3 FLUJO DE TRABAJO CON LOS PROFILES

El flujo de trabajo con *Profiles* es bastante intuitivo una vez que se tienen claro algunos conceptos importantes. Para ello, vamos a ampliar el contenido mostrado en la Figura 5.1 y también a dividirlo en pequeñas etapas que expliquen cada apartado de manera clara y concisa. Recomendamos al lector observar detenidamente la Figura 5.3, pues, contiene una visión general de cada etapa:

Recolección

Una vez que hemos decidido trabajar con los *Profiles*, y sobre todo para esta primera etapa, debemos efectuarnos la siguiente pregunta: “¿Obtenemos un *Profile* que ya esté hecho o lo creamos desde cero?” Dependiendo de la respuesta, el flujo de trabajo puede variar

5. PROFILES

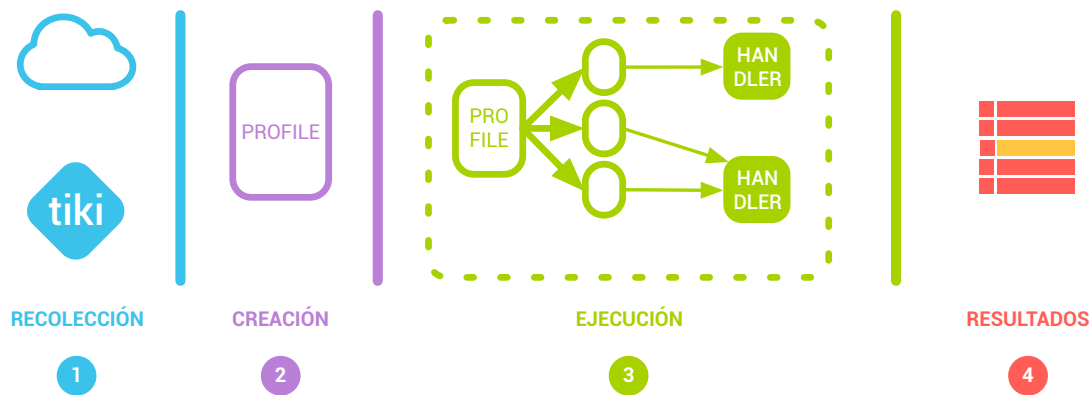


Figura 5.3

ligeramente: si descargamos un *Profile* prefabricado hay que conseguirlo de algún lado mientras que, si lo creamos desde cero, no habría que hacer nada más (nos tendríamos que poner a escribir lo que deseamos crear, pero, sin depender de nada ni de nadie).

Para obtener un *Profile* que ya está diseñado de antemano, TikiWiki utiliza el concepto de *repositorios*². ¿Y qué son? Entendemos por repositorio aquel sitio centralizado donde se almacena y se mantiene un conjunto de *Profiles*. Se pueden clasificar en dos tipos:

- **Externos:** Son en los que en nuestra instalación de TikiWiki, ésta necesita conectarse a un servidor externo para obtener un listado de todos los *Profiles* existentes. Por defecto, cuando instalamos TikiWiki por primera vez, nos añade el repositorio principal de *Profiles*: <http://profiles.tiki.org>. En dicho repositorio podemos encontrar todo tipo de *Profiles*, y éstos se pueden encontrar clasificados por funcionalidad, por número de versión, por la cantidad de votos que han recibido de otros usuarios que los han usado... Pero también podemos optar por crear nuestro propio repositorio de *Profiles*, ya que, en el fondo son simplemente páginas *wikis*.
- **Locales:** Son aquellos que se encuentran de manera local en nuestra plataforma y, por lo tanto, nuestra instalación de TikiWiki no necesita realizar ninguna petición a Internet. A su vez, se clasifican en dos formas: los que almacenan los *Profiles* en

²El concepto de repositorio en TikiWiki se asemeja bastante, en idea, a los gestores de paquetes que se utilizan en varias distribuciones de Linux como: apt [22], pacman [24] o yum [20].

5. PROFILES

la misma instalación de TikiWiki, por ejemplo: ALMA utiliza esta configuración³; o replicar a su vez, una instalación de TikiWiki dedicada a *Profiles* (similar al repositorio oficial) pero en el que no hay un acceso a Internet. Por ejemplo: podría usarlo una empresa en una *intranet* y separar así la instalación de trabajo de la del repositorio.

Creación

En esta segunda etapa, según la elección que hayamos tomado en la anterior, puede variar en la forma de llevarla a cabo: si decidimos obtener un *Profile* de algún repositorio es bastante probable que queramos ejecutarlo sin modificar nada de éste (aunque tenemos la opción de poder hacerlo); en cambio, si, por el contrario, hemos decidido crear un *Profile* desde el principio es, en esta etapa, en la que tenemos que pensar de manera creativa que queremos realizar con el mismo.

Cuando hablamos de un *Profile*, en concreto, nos estamos refiriendo al texto que encontramos dentro de una página *wiki* y que es el que posee la información para que pueda ser evaluado por el intérprete de *Profiles* y que ejecute una acción (o varias) dentro de TikiWiki. Para poder crear uno hay que seguir unas reglas definidas:

- Utilizan un formato de marcado ligero llamado YAML Ain't Another Markup Language o YAML No Es Otro Lenguaje de Marcado (YAML)⁴ y, por lo tanto, el documento debe de estar conforme a esa especificación. Si no es así, el intérprete de *Profiles* lanzará un error informando al usuario de que no se ha podido leer correctamente el contenido del documento.
- Además, también tienen que ser escritos conforme a las reglas que dictaminen los *Handlers* (los encargados de traducir la sintaxis YAML en acciones concretas de TikiWiki, veremos más información en la § 5.3 (pág. 52)).
- Ya que los *Profiles* se almacenan en páginas *wikis*, éstos pueden utilizar la potencia de la sintaxis *wiki* y, por ejemplo: se puede dividir un *Profile* en múltiples partes y

³En la § 5.5 (pág. 57) veremos que la ejecución de los *Profiles* de esta manera requieren un tratamiento especial, utilizan un componente añadido llamado *Data Channel*.

⁴Instamos al lector que si desconoce que es YAML visite el Apéndice A (pág. 85) para obtener unas nociones básicas, puesto que, los *Profiles* se basan en dicho formato y es necesario conocerlo para poder trabajar con ellos.

5. PROFILES

```
1  Esto es un documento Wiki en el que ahora mismo utilizamos
2  la sintaxis wiki. A continuación en la siguiente línea
3  tenemos un bloque que marca el comienzo de un Profile.
4
5  {CODE(caption=>YAML)}
6  preferences:
7    feature_articles: y
8  objects:
9    -
10     type: article_type
11     ref: type
12     data:
13       name: New Type
14       allow: [ ratings, comments ]
15       show: [ author, reads, language ]
16     -
17     type: topic
18     ref: topic
19     data:
20       name: New Topic
21 {CODE}
22
23 Podemos continuar editando nuestro documento con formato wiki
24 y cuando queramos, volvemos con un Profile que haga otra cosa
25 (o que continúe con el anterior Profile).
26
27 {CODE(caption=>YAML)}
28 preferences:
29   feature_galleries: y
30   feature_wiki: y
31 {CODE}
```

Figura 5.4: Ejemplo de una posible página *wiki* con dos bloques de *Profiles*. Cada bloque puede realizar una tarea distinta o bien ser la misma tarea dividida en secciones y explicada por partes. Ejemplo extraído de: <http://profiles.tiki.org/Article+Handler>.

en cada una de ellas ir explicando su funcionamiento. O también, cada división puede ser un *Profile* que realice una tarea determinada. Lo importante es que cada bloque que pertenezca a un *Profile* debe de comenzar con `{CODE(caption=>YAML)}` y finalizar con `{CODE}`. A continuación, un ejemplo real que ilustra mejor los conceptos explicados:

5. PROFILES

Ejecución

En las anteriores etapas se requería tomar ciertas decisiones que condicionaban, un poco, la manera de hacer las cosas frente a la creación de un *Profile*. En esta etapa, en cambio, no requiere ninguna interacción por parte del usuario ya que se realiza de forma automática cuando se ejecuta un *Profile*. Más bien nos sirve como excusa para presentar un concepto muy importante denominado: *Handler*.

Un *Handler* es un componente programado en la librería de *Profiles* que se encarga de leer y traducir a acciones reales (por ejemplo: crear una categoría, activar un recurso, asignar permisos...) los bloques de texto que contengan sentencias de *Profiles* en una página *wiki*. Actualmente hay dieciocho *Handlers* cada uno encargado de realizar una tarea determinada, a saber:

- *Article Handler*
- *Blog Handler*
- *Category Handler*
- *External Wiki Handler*
- *File Gallery Handler*
- *Forum Handler*
- *Menu Handler*
- *Module Handler*
- *Perspective Handler*
- *Plugin Alias Handler*
- *RSS Handler*
- *Template Handler*
- *Tracker Handler*
- *Transition Handler*

Options

Field Name	Mandatory	Value
name	y	Username
pass		If no pass is provided, then the pass will be the username
email		User's email (no validation message yet)
change		User must change password on first login (y/n default: y)
groups		User will be assigned to the group or groups listed. If the group doesn't exist previously, the handler will not currently create it (use the Group Handler to create groups).

Figura 5.5: Estas son las opciones que se pueden poner en un *Profile* si queremos crear un usuario.

- *Webmail Handler*
- *Webservice Handler*
- *Wiki Handler*
- *Users Handler*

Cada uno tiene su propia sintaxis que se debemos respetar para que el intérprete de *Profiles* sea capaz de dar por bueno un *Profile* que hayamos escrito. Podemos encontrar toda la documentación referente a todas las posibles opciones que podemos poner visitando la documentación oficial [17]. En la Figura 5.5, se muestra las opciones que posee el *Handler* de Usuarios (*Users Handler*) y que podríamos utilizar para crear un *Profile* que implique la creación, de manera automatizada, de usuarios. Cabe añadir que la sintaxis no hace falta aprenderla de memoria, cuando tengamos alguna necesidad, simplemente hay que visitar la web y consultar lo que deseemos.

Resultados

En esta etapa, la última, lo que obtenemos son los resultados de manera visible de lo que hayamos realizado con nuestro *Profile*. Ya sean cambios muy drásticos o, por contra, muy leves. A partir de aquí podemos repetir el flujo de trabajo con los *Profiles* desde el

5. PROFILES

principio y tantas veces como deseemos, solo que en las siguientes veces, ya no tendremos que diseñarlo, pues, éste fue creado anteriormente.

5.4 EXPLICACIÓN DE LA INTERFAZ DE LOS PROFILES

La administración de los *Profiles* por parte de un administrador es bastante simple y, prácticamente, no requiere de aprendizaje alguno. Si un usuario visita la siguiente dirección: <http://localhost/tiki-admin.php?page=profiles> lo que se encontrará es lo que se muestra en la Figura 5.6 (un panel principal dividido en tres pestañas):

- *Ejecutar Profiles (Apply Profiles)*: Mostrada en la figura anterior, esa pestaña es la encargada de listar todos los *Profiles* que ha encontrado TikiWiki en un determinado repositorio (sea el oficial o cualquier otro). Desde aquí el usuario puede elegir y filtrar, según preferencias, que tipos de *Profiles* desea ver. Tiene la opción de obtener más información acerca de lo que hace un *Profile* en concreto (que cosas crea y que cambia) y, por supuesto, de ejecutarlo en su instalación de TikiWiki (Figura 5.7).
- *Exportar*: Esta pestaña permite al administrador volcar todos los datos de configuración que haya realizado sobre TikiWiki. Pero, ojo, ¡sólo las preferencias! Si se ha ejecutado anteriormente un *Profile* que genera grupos, por poner un ejemplo, no se guarda ni los grupos creados ni la información que contengan éstos (Figura 5.8).
- *Avanzado*: Aquí se pueden añadir nuevos repositorios, crear múltiples *Data Channels* (como se dijo antes, entenderemos el concepto con un ejemplo práctico en la siguiente sección) y, en última instancia, existe un apartado para ejecutar los *Profiles* directamente. Esto nos sirve para probar configuraciones útiles o para realizar ejecuciones de prueba. Hay que tener especial cuidado porque como se indica en la imagen, se pueden hacer cambios irreversibles en la configuración de la base de datos (Figura 5.9).

5. PROFILES

Apply Profiles Exportar Avanzado No Tabs

Configure Tiki in 3 easy steps using Profiles

Profiles

Step 1: Use the Quick or Manual Filter option to see a list of Profiles you can apply

Option 1: Quick Filter

Option 2: Manual Filter

Filter the list of profiles:

Perfil:

Repository: Todos

Featured Profiles
Featured Profiles is a list of applications that are maintained by the Tiki community and are a great way to get started.

Full Profiles
Full Profiles are full featured out of the box solutions. ← Nos muestra los Profiles clasificados por tipo.

Mini Profiles
Mini Profiles will configure specific features and are a great way to add more functionality to an existing configuration.

Learning Profiles
Learning Profiles will allow you to quickly evaluate specific features in Tiki.

Estado

Profile repository	Estado	Última actualización
http://profiles.tiki.org		2012-03-29 20:27:24

Listado de repositorios. Por defecto, se lista el oficial.

Podemos refrescar el listado de Profiles que posee TikiWiki en su cache.

Figura 5.6: Panel principal de administración de los *Profiles*.

5.5 UN EJEMPLO CON PROFILES

Vamos a mostrar dos ejemplos: uno sencillo y otro más complejo para entender mejor todos los conceptos que han sido presentados hasta ahora. En ambos casos, se pretende explicar al lector los pasos necesarios que tiene que seguir un hipotético administrador que desea crear, todo de manera automatizada, una página *wiki* denominada “Prueba”, una categoría (también con el mismo nombre) en cuyo interior se encuentra la página anterior y, por último, cambiar el tema visual que utiliza TikiWiki (por defecto en azul) por otro en amarillo. La diferencia principal de los dos ejemplos estriba en la forma de llevarlo a cabo, en el primero, se asume que hay un *Profile* en el repositorio oficial. En el segundo, en cambio, se crea el *Profile* desde cero y se ejecuta en la instalación local utilizando para ello los *Data Channels*.

5. PROFILES

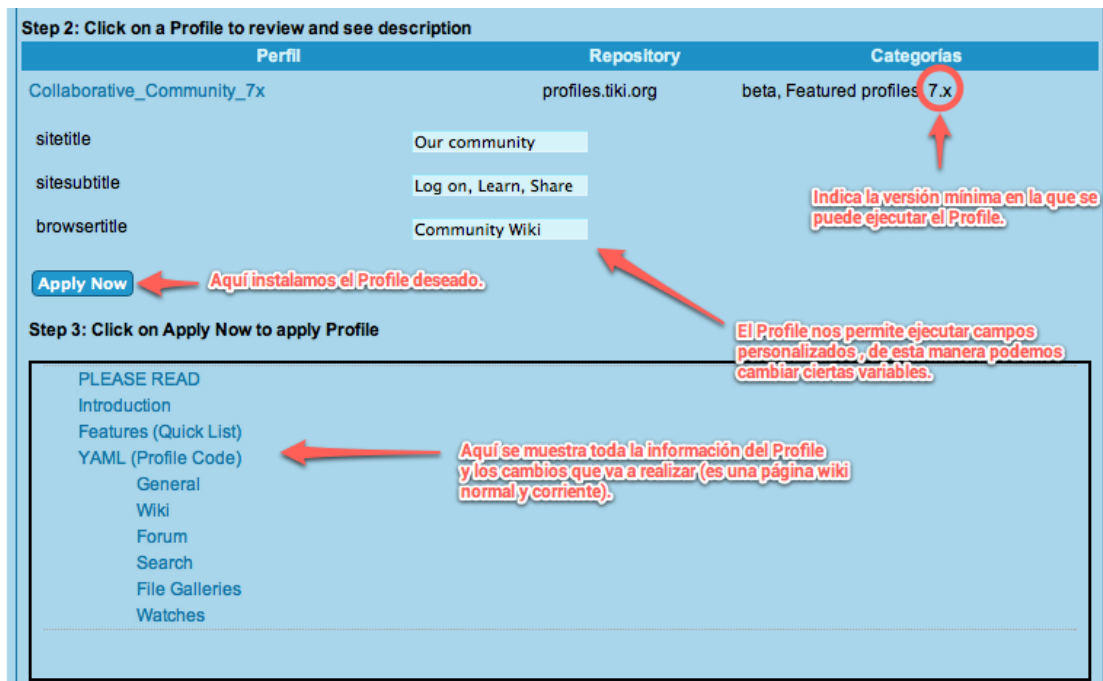


Figura 5.7: Panel donde podemos listar los *Profiles* que tiene un repositorio y ejecutarlos.

Profile ejecutado desde un repositorio

Este es el caso más sencillo, los pasos son los siguientes:

1. Como administradores nos dirigimos a la página de administración de *Profiles* (recordemos: <http://localhost/tiki-admin.php?page=profiles>).
2. Listamos los *Profiles* existentes en el repositorio oficial y buscamos uno que se llama *Test_Wiki_Group* (podemos basarnos en las imágenes de la anterior sección puesto que este paso no cambia prácticamente nada, sólo el nombre del *Profile* a listar, Figura 5.7).
3. Rellenamos las casillas con los nombres apropiados, el nombre de la página *wiki*, el de la categoría y el del grupo y pulsamos ejecutar.
4. ¡Et voilà! Acción ejecutada, podemos comprobar en la Figura 5.10 que se cumple lo que el *Profile* prometía.

5. PROFILES

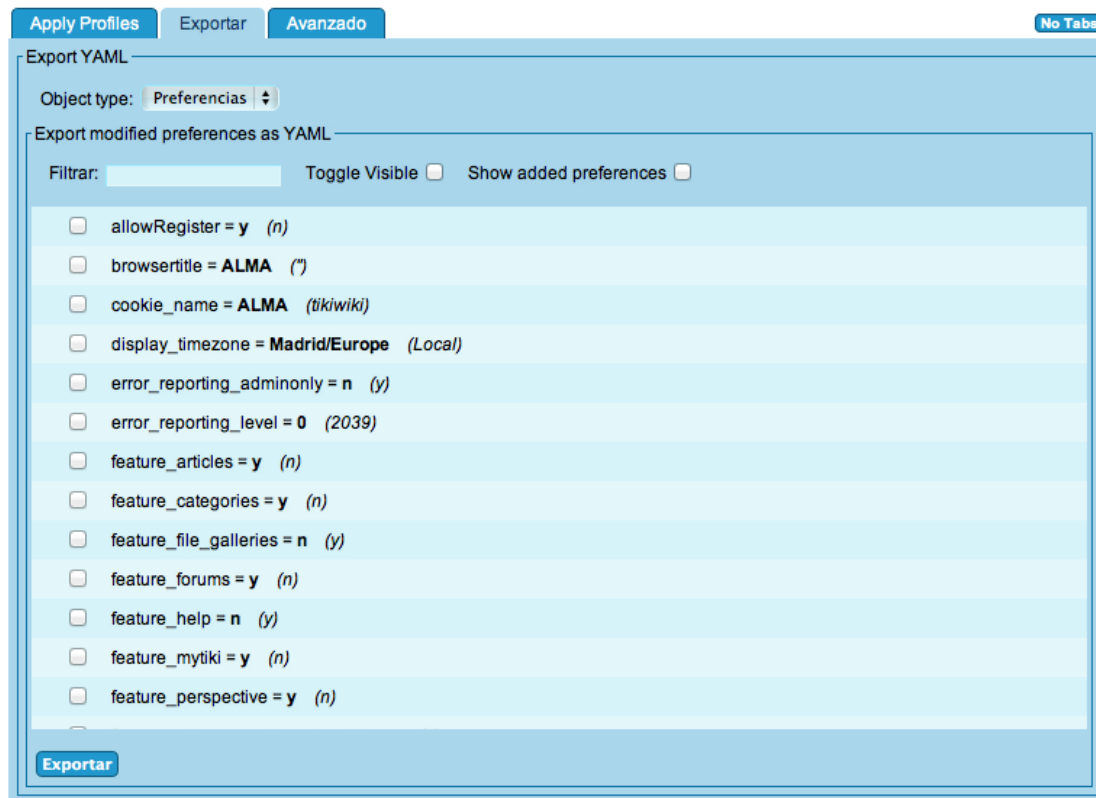


Figura 5.8: Panel donde podemos exportar las preferencias que hayamos modificado de TikiWiki. Se puede utilizar como método de copia de seguridad para restaurar posteriormente la misma configuración.

Profile ejecutado desde una instalación local

En este caso, un poco más complejo, el *Profile* se ejecuta en una instalación local por lo que los pasos varían en varios puntos respecto al anterior ejemplo:

1. Dado que en esta situación tenemos que crear el *Profile* desde cero, conviene pensar (como ya hicimos anteriormente) en términos absolutos: ¿qué queremos conseguir? y ¿cómo lo vamos a realizar? La respuesta a la primera pregunta está clara, lo definimos anteriormente: crear una página *wiki* con un nombre determinado, una categoría de idéntico nombre que agrupe al anterior recurso y cambiar el color del tema visual. La respuesta para la segunda pregunta nos obliga a pensar en términos de *Handlers*, en este caso, ¿qué *Handlers* intervienen? Para crear la página *wiki* es casi seguro que necesitaremos el *Wiki Handler*, para la categoría utilizaremos el *Category Handler* y para cambiar el color al ser una preferencia se puede poner

5. PROFILES

directamente en el *Profile* (en la Figura 5.11 tenemos de manera completa el *Profile* que utilizaremos, dejamos al lector la tarea de investigar y comprender las opciones que se listan utilizando para ello la bibliografía existente).

2. Una vez que tenemos claro como va a ser el *Profile*, hay que crear una página *wiki* para almacenarlo, denominaremos a la página *wiki* `primer_profile` (no hay que olvidar este nombre ya que lo utilizaremos después y lo debemos respetar tal y como está escrito), copiamos el *Profile* (Figura 5.11), añadimos los fragmentos al principio (`{CODE(caption=>YAML)}`) y al final (`{CODE}`) del *Profile* y guardamos la página (Figura 5.13).
3. Ya hemos creado la página *wiki* con nombre `primer_profile` y el *Profile* dentro. A partir de ahora podemos modificar la página tantas veces como queramos y recuperar su contenido con el historial de versiones.
4. Debido a que el *Profile* se va a ejecutar de manera local tenemos que utilizar los *Data Channels* que es el encargado de avisar a TikiWiki para que no busque en otro repositorio externo, sino dentro de si mismo⁵. Para ello vamos a la pestaña de Avanzado en el panel de administración de *Profiles* (Figura 5.9) y tenemos que introducir lo siguiente en la casilla de *Data Channels*:

```
1 | mi_prueba, tiki://local, primer_profile, Admins
```

La documentación de los *Data Channels* explican el significado de cada elemento, un resumen:

- *mi_prueba*: Tenemos que denominar al *Data Channel* con un identificador único, puede ser cualquier nombre que sea un carácter alfanumérico incluyendo también al símbolo `_`.
- *tiki://local*: Indica a TikiWiki que utilice la página *wiki* como *Profile* ya que está en su propia base de datos y no en un repositorio externo.
- *primer_profile*: Indicamos a TikiWiki cual es la página en concreto que se usará como *Profile*. Se corresponde con el nombre que dimos a la página *wiki* anteriormente.

⁵ Aunque los *Data Channels* tienen más usos que el que se lista en este ejemplo. Si queremos ver todos y aprender más sobre ellos conviene visitar su página *wiki* [6].

5. PROFILES

- *Admins*: Especificamos que grupo tiene derecho a ejecutar el *Profile*. Se recomienda poner grupos que sean de confianza (es una imprudencia asignar el *Profile* a un grupo como Anonymous donde cualquiera puede ejecutarlo). Si queremos poner más grupos, se pueden poner separados por coma, por ejemplo: Admins, Grupo1, Grupo2.
5. Guardamos la información que hemos escrito en el *Data Channel*. ¡La parte más difícil ya está superada! Ahora tenemos que buscar la manera de poder ejecutar el *Profile*, y para ello vamos a crear una interfaz gráfica con un botón al que siempre podamos acudir cuando queramos utilizar el *Profile* (en la Figura 5.14 vemos un ejemplo de como quedaría al final del proceso). Para ello vamos a generar una nueva página *wiki* denominada *mi_profile_interfaz* y copiaremos el contenido de la Figura 5.12 (en la descripción del mismo se explica el significado de todo el código).
 6. ¡Y ya está! Cada vez que queramos utilizar dicho *Profile* acudimos a la página *mi_profile_interfaz* pulsamos el botón y este ejecuta el *Profile* guardado en *primer_profile*. Hay que avisar que si un usuario que no pertenezca al grupo Admins (que así fue como lo especificamos en la línea de los *Data Channel* en el paso 4) intenta ejecutar el *Profile*, éste no hará nada. No obstante un usuario que no esté autorizado a ejecutar *Profiles* no debería ver tampoco esta página así como donde está el *Profile* en sí (nos evitaremos muchos problemas a la larga).

5.6 IMPLEMENTACIÓN INTERNA DE LOS PROFILES

A continuación, para todas aquellas personas que estén interesadas en desarrollar y/o mejorar un *Handler* o simplemente desean saber más sobre como funciona internamente la plataforma, se dará una explicación de las partes más interesantes que hacen posible que los *Profiles* existan tal y como los conocemos. Para ello, se explicará como se desarrolló el *Users Handler* (el que se encarga de crear usuarios) por ser éste bastante sencillo de entender y que fue creado por el autor del PFC. Aconsejamos al lector leer (si no se ha realizado anteriormente) el Apéndice C (pág. 99) para obtener primero una visión

5. PROFILES

complementaria de como está estructurado el código de TikiWiki. Sin más dilaciones, comenzamos.

¿Cómo funciona por dentro un Handler?

El código fuente que da vida a los *Profiles* se halla en la carpeta `lib/profilelib`. Dentro de dicho directorio nos encontramos con cuatro archivos importantes:

- `profilelib.php`: Esta librería es la que da el funcionamiento principal de todo el código y la encargada de interpretar los documentos en formato `YAML`, de buscar referencias de objetos en el propio *Profile*, de devolver un mensaje de error al usuario si algo ha ido mal, y de realizar las peticiones a otras plataformas *wiki* si el *Profile* está alojado en otros repositorios. Normalmente el código que hay aquí no hace falta tocarlo para poder crear un *Handler* a no ser que queramos añadir una nueva forma de interpretar los documentos (por ejemplo: que en lugar de leerlos en formato `YAML` los hiciese en formato JavaScript Object Notation (`JSON`)).
- `channellib.php`: Esta librería es la que lleva el manejo de los *Data Channels*, algunas tareas que realiza es la comprobación de que un *Data Channel* puede ser ejecutado por un usuario, y otra es cargar en la interfaz los *Channels* que hay guardados en la base de datos. De manera idéntica a lo que se dijo en el punto anterior, no debemos modificar nada de este fichero.
- `listlib.php`: Esta librería es la encargada de realizar peticiones a otros servidores y generar una base de datos con los *Profiles* que hay existentes en dicho repositorio.
- `installlib.php`: Esta librería es la encargada de crear los dieciocho *Handlers* que hay actualmente. Hay una clase principal denominada `Tiki_Profile_InstallHandler` y es sobre la que deberemos heredar si queremos crear un nuevo *Handler*. Dicha clase determina los siguientes métodos: `__construct()`, `canInstall()`, `install()`, `replaceReferences()` y `_install()` y son importantes puesto que las clases que implementemos tendrán que sobre-escribirlos (dependiendo de las necesidades del *Handler*). El nombre de la clase utilizado para el *User Handler* es `Tiki_Profile_InstallHandler_User` y hereda de `Tiki_Profile_InstallHandler`, los métodos que implementa son los siguientes:

5. PROFILES

- `getData()`: Este método lo que realiza es obtener la información del *Profile* (todo lo que le corresponde al *Handler* de usuarios) y reemplazar las referencias que haya.
- `canInstall()`: Este método se encarga de preguntar si se puede instalar el *Profile* o no y para ello comprueba que hay algo escrito dentro del *Profile*.
- `_install()`: Este es el método más importante ya que es el que realiza el grueso de la operación de añadir usuarios. Lo que hace es comprobar si están los campos que necesita el *Handler* para funcionar y si es así llama a la librería `userlib.php` que es la encargada de crear nuevos usuarios.

En la Figura 5.15 se muestra la implementación completa (por ser bastante breve) del *User Handler*.

5. PROFILES

Apply Profiles Exportar Avanzado No Tabs

Repositories

☐ Developer mode ⓘ ⚠

Repository URLs: ⓘ

http://profiles.tiki.org/profiles

Podemos añadir nuevos repositorios (tantos como queramos).

Enter multiple repository URLs, one per line.

Data Channels: ⓘ ⚠ ↻

crear_asignatura, tiki://local, Crear_Comunidad_de_PracticaTemplate, Admins
mi_prueba, tiki://local, primer_profile, Admins

Cuando queremos crear un Data Channel, se hacen aquí (uno por línea).

Data channels create a named pipe to run profiles from user space. One channel per line. Each line is comma delimited and contain channel name, domain, profile, allowed groups.

Guardar

Profile tester

⚠ Cuidado

Paste or type wiki markup and YAML (with or without the {CODE} tags) into the text area below
This will run the profile and make potentially unrecoverable changes in your database!

Test Profile Name: Test

Nos permite ejecutar un Profile directamente sin necesidad de tenerlo guardado (hay que tener cuidado).

Test

Figura 5.9: Panel que nos permite añadir nuevos repositorios, agregar nuevos *Data Channels* y ejecutar *Profiles* de prueba.

Categoría actual: Top » Prueba

..

Nombre Tipo
Prueba wiki

Figura 5.10: Categoría creada con una página wiki tal y como el Profile había especificado.

5. PROFILES

```
1 | {CODE(caption=>YAML)}
2 | preferences:
3 |   style_option: kiwi.css
4 |   enable: [ feature_wiki ]
5 | objects:
6 |   -
7 |     type: wiki_page
8 |     data:
9 |       name: Prueba
10 |      content: Creando una wiki con profiles
11 |      ref: pagina_prueba
12 |   -
13 |     type: category
14 |     data:
15 |       name: Prueba
16 |       items:
17 |         - [ wiki_page, pagina_prueba ]
18 | {CODE}
```

Figura 5.11: En la sección de preferences se pueden poner las preferencias directamente que encontramos en la tabla `tiki_preferences` como ya comentamos. Hay que añadir que al crear múltiples recursos en este *Profile* hacemos uso de las referencias (aquellas que comienzan con `ref`, lo usaremos más adelante [8]).

```
1 | {DATACHANNEL(channel=mi_prueba)}{DATACHANNEL}
```

Figura 5.12: Este código copiado tal cual en la página *wiki* genera la interfaz gráfica e invoca al *Profile* que hemos creado antes. Hay que observar que en donde pone `channel` se ha escrito el nombre del *identificador que dimos en el Data Channel*. En la documentación oficial tenemos más información acerca del Plugin *Data Channel* que es como se denomina esta herramienta que genera interfaces para los *Profiles* [9].

Editar: primer_profile

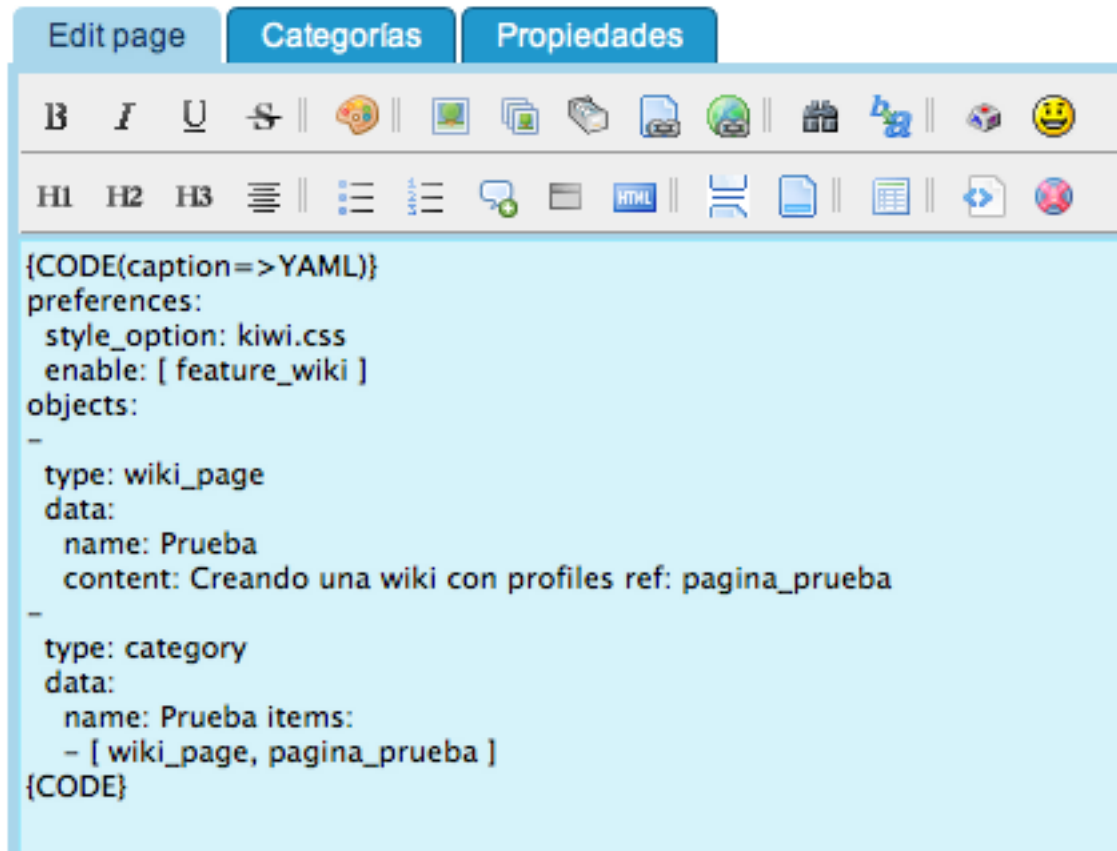


Figura 5.13: Escribimos el contenido del *Profile* en una página wiki.

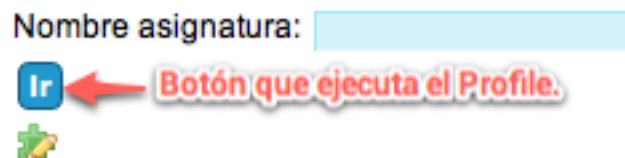


Figura 5.14: Interfaz gráfica generada por el *plug-in Data Channel*, podemos ejecutar el *Profile* cómodamente con un clic de ratón.

5. PROFILES

```
1  <?php
2  class Tiki_Profile_InstallHandler_User extends Tiki_Profile_InstallHandler
3  {
4      function getData()
5      {
6          if( $this->data )
7              return $this->data;
8          $data = $this->obj->getData();
9          $this->replaceReferences($data);
10
11         return $this->data = $data;
12     }
13
14     function canInstall()
15     {
16         $data = $this->getData();
17
18         if (isset($data)) return true;
19         else return false;
20     }
21
22     function _install()
23     {
24         if ($this->canInstall())
25         {
26             global $userlib; if (!$userlib) require_once 'lib/userslib.php';
27
28             $user = $this->getData();
29
30             if (!$userlib->user_exists($user[ 'name' ])) {
31                 $pass = isset($user[ 'pass' ]) ? $user[ 'pass' ] : $user[ 'name' ];
32                 $email = isset($user[ 'email' ]) ? $user[ 'email' ] : '';
33                 if (isset($user[ 'change' ]) && $user[ 'change' ] === false) {
34                     $userlib->add_user($user[ 'name' ], $pass, $email);
35                 } else {
36                     $userlib->add_user($user[ 'name' ], $pass, $email, $pass, true);
37                 }
38             }
39
40             if (isset($user[ 'groups' ])) {
41                 foreach ($user[ 'groups' ] as $group) {
42                     $userlib->assign_user_to_group($user[ 'name' ], $group);
43                 }
44             }
45
46             return $userlib->get_user_id($user[ 'name' ]);
47         }
48     }
49 }
```

6

GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA



EN ESTE CAPÍTULO, dividido en dos partes fundamentales, se mostrará en la primera que mejoras se han producido utilizando los *Profiles* (y desde el punto de vista de un administrador) en la creación de nuevas comunidades de práctica. En la segunda parte se mostrarán detalles internos de los *Profiles* realizados (no obstante, para analizar las implementaciones completas de dichos *Profiles* se deberá visitar el Apéndice B (pág. 90), ya que, en este capítulo sólo se enseñarán los fragmentos que se consideran relevantes).



Figura 6.1: Dado que el *Profile* es una página *wiki* hay que buscar donde se encuentra éste.

6.1 ¿QUÉ HA CAMBIADO EN LA GESTIÓN DE TIKIWIKI UTILIZANDO PROFILES?

¿Cómo se crea una nueva asignatura?

En esta nueva situación, crear asignaturas (comunidades de práctica) se ha convertido en una tarea muy simple (cuando antes no lo era). Los pasos que tiene que seguir el administrador o docente para crear una nueva asignatura son los siguientes (se asume que ya están instalados los *Profiles* diseñados para ALMA correctamente, en el Apéndice C (pág. 99) se encuentra el manual de usuario que explica como hacerlo de manera muy detallada):

1. El administrador o docente ha de listar todas las páginas *wiki* que están disponibles en TikiWiki (Figura 6.1). Ha de buscar una llamada: *Crear_Comunidad_de_Practica_GUI*, en ella se encuentra la interfaz por la cuál se puede ejecutar el *Profile* que se encuentra definido en otra página *wiki* denominada: *Crear_Comunidad_de_Practica_Template*.
2. Una vez en la página *Crear_Comunidad_de_Practica_GUI* se escribe el nombre de la asignatura que se desee crear (Figura 6.2).

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

Menú

Inicio
Categorías
MITiki
Wiki
Artículos
Foros
Admin

Crear_Comunidad_de_Practica_GUI

Nombre asignatura:



Asignaturas

☐ Stay on this page

☒ Default
Circuitos Electricos
Estadística

Figura 6.2: Es tan sencillo como poner el nombre de la asignatura a crear y darle a Ir. El *Profile* se encargará del resto.

Usuario	Correo	Última conexión	Grupos	Acción
admin	a@a.com	Lunes 07 de Mayo, 2012 18:56:51 CEST	<input checked="" type="checkbox"/> Admins <input checked="" type="checkbox"/> Circuitos Electricos Alumnos <input checked="" type="checkbox"/> Circuitos Electricos Profesores <input checked="" type="checkbox"/> Estadística Alumnos <input checked="" type="checkbox"/> Estadística Profesores	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="Historial"/>
<input type="checkbox"/> rober		Miércoles 29 de Febrero, 2012 16:58:25 CET	<input checked="" type="checkbox"/> Circuitos Electricos Profesores <input type="checkbox"/> Registered	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="Historial"/> <input checked="" type="button" value="X"/>

Figura 6.3: En el panel de usuarios se pueden agregar qué usuarios pertenecen a qué grupos. Por defecto, por cada comunidad de práctica que se crea, el usuario *admin* pertenece a cada grupo de dicha comunidad.

3. ¡Y listo, comunidad de práctica creada! A partir de ahora el administrador o docente deberá ir a la sección de usuarios y decidir quién quiere que sea miembro de dicha comunidad (Figura 6.3).

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

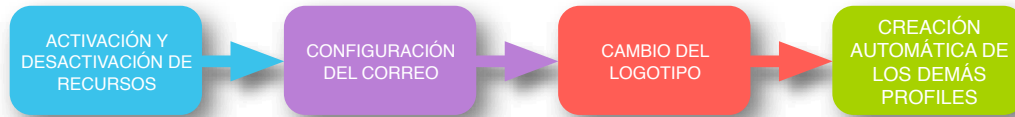


Figura 6.4: Etapas empleadas para elaborar el *Profile* que configura TikiWiki a las necesidades de ALMA.

6.2 IMPLEMENTACIÓN INTERNA DE LOS *PROFILES* DESARROLLADOS

Profile que configura TikiWiki a las necesidades de ALMA

Objetivo

El objetivo principal de este *Profile* es configurar una nueva instalación de TikiWiki y modificarla a las necesidades del proyecto ALMA. Esto presenta múltiples ventajas, entre ellas:

- Nos permite replicar ALMA de una manera muy rápida y cómoda para un administrador. Si, por ejemplo, la instalación que existe actualmente sobre TikiWiki desapareciese (por el motivo que fuere), se podría descargar de nuevo el programa, ejecutar dicho *Profile* y tener una réplica exacta de la anterior ALMA en ajustes y preferencias. De esta manera el administrador de la plataforma sólo se tiene que preocupar por recuperar los datos que se hayan guardado en la anterior instalación e importarlos en la nueva.
- Como punto que parte del anterior, nos sirve también para crear una réplica de ALMA para hacer nuestras pruebas internas sin afectar al servidor principal, es decir, como plataforma de desarrollo. Este es el método que ha usado el autor del PFC para crear los demás *Profiles*.

¿Cómo se ha implementado?

La lógica que se ha utilizado para crear este *Profile* es la que sigue en la Figura 6.4. A continuación, se explican las etapas una a una:

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

- *Activación y desactivación de recursos:* Dado que ALMA tiene unas necesidades muy concretas se han habilitado todos los elementos que hacen posible que funcionen las comunidades de práctica (entre ellos se activan los *Profiles*, las categorías y las perspectivas):

```
1 | feature_categories: y
2 | feature_myтики: y
3 | feature_userPreferences: y
4 | feature_profiles: y
5 | feature_perspective: y
```

Así mismo se desactivan múltiples características de TikiWiki que, en principio, no se van a utilizar (al menos por ahora). Si analizamos el *Profile*, encontraremos que se anulan los siguientes elementos:

```
1 | feature_blogs: n
2 | feature_file_galleries: n
3 | feature_forums: n
4 | feature_freetags: n
5 | feature_help: n
6 | feature_trackers: n
```

- *Configuración del correo electrónico:* Se dan los datos necesarios para que se puedan enviar correos electrónicos desde la propia plataforma, ya sea para notificar a los usuarios de que se han registrado en una comunidad de práctica así como para recibir la contraseña en caso de extravío.

```
1 | sender_email: alma@uah.es
2 | zend_mail_handler: smtp
3 | default_mail_charset: utf-8
4 | zend_mail_smtp_user: alma.disciplinar@gmail.com
5 | zend_mail_smtp_pass: password
6 | zend_mail_smtp_port: 465
7 | zend_mail_smtp_security: ssl
8 | zend_mail_smtp_auth: plain
9 | zend_mail_smtp_server: smtp.gmail.com
```

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

- *Cambio del logotipo:* Se personaliza la imagen de TikiWiki sustituyendo el logotipo que trae por defecto a uno diseñado específicamente para ALMA.

```
1 | sitelogo_title: ALMA (Aula Libre Multidisciplinar Abierta)
2 | sitelogo_src: img/tiki/tikisitelogo.png
3 | sitelogo_alt: Alma
```

- *Creación automática de los demás Profiles:* Por último se encarga de crear un grupo denominado Profiles donde se almacenarán los dos Profiles que se utilizan para crear comunidades de práctica (el Profile en si y su Data Channel asociado).

```
1 | objects:
2 |   -
3 |     type: wiki_page
4 |     ref: wiki1
5 |     data:
6 |       name: Crear_Comunidad_de_Practica_GUI
7 |       content: "{DATACHANNEL(channel=crear_comunidad)}"
8 |       nombre_asignatura, Nombre de la comunidad de practica
9 |       {DATACHANNEL}"
10 |   -
11 |     type: wiki_page
12 |     ref: wiki2
13 |     data:
14 |       name: Crear_Comunidad_de_Practica_Template
15 |   -
16 |     type: category
17 |     ref: profilecateg
18 |     data:
19 |       name: Profiles
20 |       items:
21 |         - [ wiki_page, $wiki1 ]
22 |         - [ wiki_page, $wiki2 ]
```

Y se le asignan permisos específicos para que sólo los administradores puedan ver, editar y borrar todos los elementos que contenga dicha categoría.

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

```
1 | permissions:
2 |   Admins:
3 |     objects:
4 |       -
5 |         type: category
6 |         id: $profilecateg
7 |         allow:
8 |           - admin
```

También se realizan los ajustes pertinentes (que no puedan ver, ni editar, ni borrar) para los demás grupos (Registered, Anonymous).

```
1 |   Registered:
2 |     objects:
3 |       -
4 |         type: category
5 |         id: $profilecateg
6 |         deny: view, edit, view_category, edit_category
```

- *Miscelánea:* Otros cambios que se realizan son las mejoras en los editores What You See Is What You Get o Lo Que Ves Es Lo Que Obtienes (WYSIWYG) (se añade la posibilidad de que los usuarios puedan crear fórmulas matemáticas) o se pone el lenguaje de TikiWiki en castellano.

```
1 |   language: es
2 |   feature_wysiwyg: y
3 |   wysiwyg_default: y
4 |   wysiwyg_optional: y
5 |   wikiplugin_equation: y
6 |   wikiplugininline_equation: y
```

Si queremos analizar la implementación completa hay que visitar la § B.2 del Apéndice B (pág. 90).

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

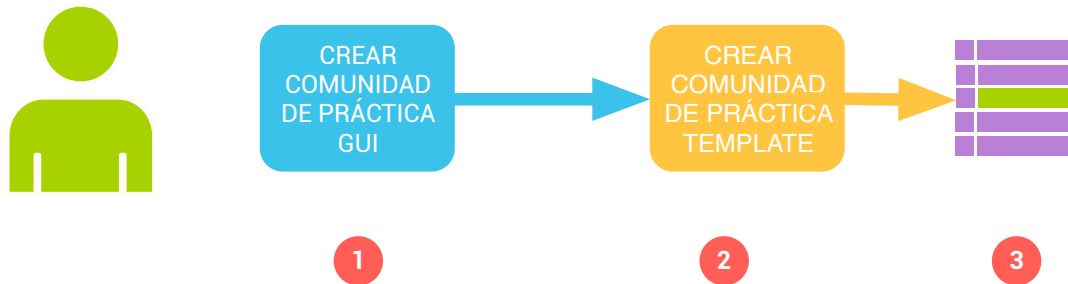


Figura 6.5: Supongamos que un administrador o docente desea crear una asignatura. Para ello se dirige a la página *Crear_Comunidad_de_Practica_GUI*. Ésta es la encargada de dibujar la interfaz gráfica que es lo que el usuario ve como un campo para introducir un texto (aunque puede ser tan compleja como deseemos). Una vez que el administrador ha decidido el nombre de la asignatura y pulsa el botón enviar, la petición es enviada a la página *wiki Crear_Comunidad_de_Practica_Template* que es donde se encuentran las instrucciones precisas para generar una comunidad de práctica. A continuación, se rellenan las variables oportunas para personalizar los nombres de los grupos, las páginas *wikis*, etc y se ejecuta uno a uno las instrucciones que hay dentro de dicho *Profile* (tal y como explicamos en el capítulo anterior).

Profile que crea de manera automática comunidades de práctica

Objetivo

El objetivo de este conjunto de *Profiles* (se necesitan dos para llevar a cabo la tarea, aunque, el grueso de la operación realmente se produce en uno y el otro sirve como apoyo) está bastante claro. Por definición expresa, es el que simplifica la gestión de TikiWiki y son los que dan sentido completo a este PFC.

¿Cómo se ha implementado?

Anteriormente se explicó como un administrador o docente podía crear una asignatura visitando la página *wiki: Crear_Comunidad_de_Practica_GUI*. En la Figura 6.5 podemos ver como es realmente el flujo por detrás (la implementación que existe).

Por defecto, los elementos que el *Profile Crear_Comunidad_de_Practica_Template* generan son los siguientes (podemos verlo de manera gráfica en la Figura 6.6):

- Se crea, por defecto, una página *wiki* de bienvenida a la asignatura. Sirve como un punto de paso para cualquier miembro de la comunidad de práctica. Los posibles

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

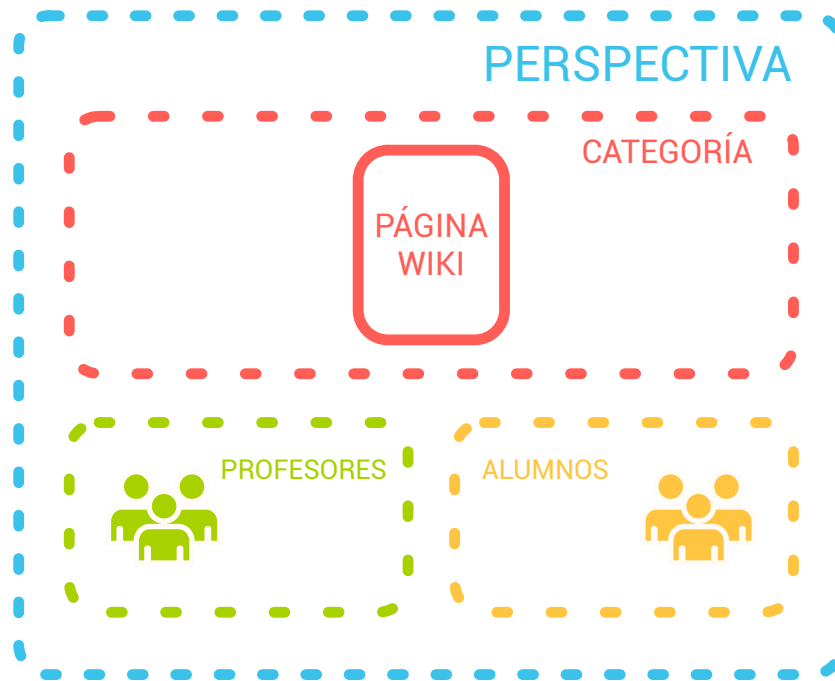


Figura 6.6: Elementos que genera el *Profile Crear_Comunidad_de_Practica_Template*.

usos que se le pueden dar son variados, por ejemplo: se podría crear un listado de enlaces con las últimas novedades que han sucedido en la comunidad.

```
1 | type: wiki_page
2 |   ref: dashboard
3 |   data:
4 |     name: $profilerequest:nombre_asignatura$Asignatura sin nombre$ Homepage
5 |     content: "Bienvenido a tu nueva asignatura"
```

- Se crea una categoría que sirve para agrupar todos los elementos que se generen durante la actividad de la propia comunidad. Se utiliza por dos razones principalmente: la primera que así controlamos los permisos de dicha comunidad y sabemos, invariablemente, que sólo los miembros pueden actuar sobre los recursos que contiene; la segunda, los recursos quedan categorizados y facilitan la navegación y el orden dentro de TikiWiki.

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

```
1  objects: -
2      type: category
3      id: $project_root
4      allow:
5          - view
6          - edit
7          - wiki_view_history
8          - wiki_view_source
9          - minor
10         - upload_picture
11         - rollback
12         - view_category
13         - search
14         - delete_account
15         - group_view
16         - group_view_members
17         - add_object
18         - modify_object_categories
```

- Se generan dos grupos: *Profesores* y *Alumnos* de dicha comunidad. Cada uno tienen unos permisos detallados, por ejemplo: los alumnos no pueden borrar las ediciones que hayan realizado en una página *wiki* mientras que los profesores sí. Si un usuario no pertenece a ninguno de estos dos grupos (sea la comunidad de práctica que sea) se considera que no es miembro de ésta y por lo tanto no aparece la asignatura en el menú *Asignaturas*.
- Se crea una perspectiva que es la encargada de realizar el “cambio de contexto” de una comunidad de práctica a otra.

```
1  type: perspective
2      ref: perspective
3      data:
4          name: $profilerequest:nombre_asignatura$sin nombre$
5          preferences:
6              category_jail: $project_root
7              wikiHomePage: $dashboard
8              browsertitle: $profilerequest:nombre_asignatura$Asignatura sin nombre$
```

6. GESTIÓN DINÁMICA DE LA PLATAFORMA ALMA

```
9      style: fivealive.css
10     style_option: kiwi.css
11     feature_wysiwyg: y
12     wysiwyg_default: y
13     wysiwyg_optional: y
```

- Por último se ajustan todos los permisos de manera adecuada por cada elemento creado anteriormente.

```
1      permissions:
2      Member:
3      autojoin: y
4      allow:
5          - view
6          - edit
7          - wiki_view_history
8          - wiki_view_source
9          - minor
10         - upload_picture
11         - rollback
12         - view_category
13         - search
```

La implementación completa, tanto del *Profile* Crear_Comunidad_de_Practica_Template como del Crear_Comunidad_de_Practica_GUI, la podemos encontrar en la § B.2 (pág. 94) y § B.2 (pág. 98), respectivamente, del Apéndice B (pág. 90).

III

SUMARIO

7

CONCLUSIONES



7.1 CONCLUSIONES DE LA IMPLEMENTACIÓN

Implementar las comunidades de práctica con TikiWiki fue una decisión muy importante que tuvimos que sopesar bastante. Sobre todo, cuando estábamos trabajando con un conjunto grande de usuarios y su gestión se tornaba de vital importancia para el éxito del proyecto. Los *Profiles* han resultado ser una herramienta muy útil que ayuda a gestionar la plataforma de una manera bastante cómoda. Entre las alternativas que podemos encontrar en el mercado, no existe un concepto similar y que ofrezca tantas posibilidades como lo hacen éstos en TikiWiki.

Si hay algo que es importante a tener en cuenta es, que las nuevas tecnologías deben ir siempre un paso por delante y tienen que promover, por encima de todas las cosas, la facilidad de uso. Hay muchos colectivos que no están tan implicados, al menos no de una manera tan profunda, con las nuevas tecnologías y son a ellos a los que más hay que tener presente cuando se piensa en crear una nueva interfaz de usuario. Es evidente que TikiWiki tiene que mejorar en múltiples aspectos, sobre todo en la facilidad de uso.

7. CONCLUSIONES

También lo tiene que hacer en la propia herramienta de los *Profiles*, puesto que, aún se tiene que aprender a utilizar YAML y eso es algo que no está al alcance de todo el mundo.

Las conclusiones positivas que se extraen de este trabajo realizado son varias:

- Se ha conseguido implementar el proyecto ALMA con bastante éxito y tal cual se pedía en los requisitos. Prueba de ello es que se está utilizando de manera real con un grupo de usuarios considerable en algunas asignaturas impartidas en la UAH (en la Figura 7.1 se puede ver una captura).
- Se han mejorado considerablemente las tareas administrativas por parte de un administrador respecto a la que ofrecía originalmente el flujo de trabajo sin *Profiles*. Y esa mejora repercute en una cosa: *productividad*.
- Estamos contentos de haberlo implementado en TikiWiki (pese a que no es el *software* perfecto, ahora veremos porqué), ya que, esta herramienta, aún a la vez simplicidad y complejidad. Simplicidad porque es un programa que nada más instalarlo, se puede usar sin mayor complicación. Complejidad porque como hemos visto, se ha comportado de manera excelente a la hora de buscar “casos de uso” más complejos.

Por contra, en los aspectos negativos nos encontramos que:

- Uno de los requisitos que pedíamos cuando estábamos buscando un *software* para implementar las comunidades de práctica era: *la instalación y el manejo de la plataforma debe de ser sencillo*. Este hecho, nos limitó un poco el campo de búsqueda de aplicaciones que pudieran crear, hipotéticamente, un gran colectivo de usuarios trabajando con las comunidades de práctica. Todas las aplicaciones que analizamos están basadas en PHP por ser, precisamente, uno de los lenguajes más fáciles de aprender y de instalar en un servidor. Pero, hay otras plataformas programadas en otros lenguajes de programación como JAVA y que son bastante más potentes que las que hay en PHP. Entre muchas propuestas nos encontramos a *Liferay* [29] o *Alfresco* [28] y que se utilizan, sobre todo, en entornos empresariales. Éstos SGC tienen una muy buena implementación del concepto de grupos y recursos y una interfaz de usuario bastante más amigable que la que ofrece TikiWiki. Quizá, implementar las comunidades de práctica en esas plataformas hubiera sido más sencillo

7. CONCLUSIONES



PáginaInicio

Felicidades

Esta es la página inicial por omisión para tu Tiki. Si ves esta página, tu instalación se ha realizado correctamente.

Puedes cambiar esta página después de conectarte. Por favor, revisa la [sintaxis wiki](#) para los detalles de edición.

★ Ponte en marcha.

Para empezar la configuración de tu sitio:

1. Inicia la sesión como usuario **admin** con contraseña **admin**.
2. Cambia la contraseña de administrador.
3. Habilita funcionalidades específicas de Tiki.

Figura 7.1: Implementación real de ALMA con TikiWiki.

que hacerlo en TikiWiki (aún sin contar con los *Profiles*), pero, eso si, aumentando considerablemente la dificultad de modificación del código fuente en dichas aplicaciones (requieren tener un buen conocimiento interno de su estructura, que es infinitamente más compleja que la que utiliza TikiWiki).

- La herramienta de los *Profiles* aún está en construcción y, debido a esto, las especificaciones de los *Handlers* cambian con bastante frecuencia (aunque cada vez menos). Actualmente se están centrando en mejorar la experiencia de usuario para permitir que cualquier persona pueda usarlos y lo haga sin mucho esfuerzo. Quizá, si hubiéramos esperado más tiempo a la hora de ejecutar este proyecto, es bastante posible que la creación de los *Profiles* se hubiera realizado a *golpe de ratón* en lugar de a *golpe de YAML*.

7. CONCLUSIONES

7.2 LECCIONES APRENDIDAS

Este PFC ha tenido una duración aproximada de unos dos años (desde que se comenzó a plantear la idea hasta que se ha definido el objetivo concreto), y durante ese tiempo han sucedido múltiples eventos que han servido al autor para aprender en muchos ámbitos. Una de las mayores noticias fue la aceptación por parte de Google para participar en su programa *Summer of Code* de 2009 en la mejora de los *Workspaces* (que no es un concepto bastante parecido al de las comunidades de práctica) de TikiWiki. Entre otras cosas se aprendió:

- Como se organiza una plataforma de código abierto (su arquitectura interna) y como los recursos humanos son coordinados para la consecución de un objetivo común.
- A mejorar las habilidades codificando con PHP. Al comenzar el proyecto desconocía cómo se programaba en este lenguaje y, al acabarlo, puedo decir orgulloso que es uno de los que más conozco.
- A mejorar mi nivel de inglés. Estar en un ambiente de índole internacional me permitió conocer nueva gente y a expresar mis ideas dando charlas en público.
- A aprender el maravilloso mundo de la tipografía y la maquetación. Maquetar un libro es una tarea que no es trivial y que requiere un esfuerzo y unas ganas por aprender bastante importantes. El resultado que has podido ver (y leer), es fruto de todo este esfuerzo.

7.3 FUTURAS LINEAS DE TRABAJO

Hay que reconocer que, aunque durante todo este PFC se han estado ensalzando las virtudes de TikiWiki, éste posee una serie de desventajas considerables en cuanto a varios aspectos de la plataforma. La verdad sea dicha, hay bastantes cosas que se pueden enriquecer aún más y que son susceptibles de que alguien coja el testigo y las convierta en realidad, entre ellas destacamos (sin ningún orden concreto de prioridad):

1. Mejorar las interfaces gráficas de administración de *Profiles* de TikiWiki. Actualmente los *Profiles* requieren que una persona tenga ciertos conocimientos técnicos

7. CONCLUSIONES

a la hora de poder implementarlos. Eso en un entorno docente puede ser que no sea factible, quizá un profesor está al cargo de alguna asignatura y por el mero hecho de que ni si quiera sabe qué es YAML no debería ser discriminado. ¿De qué manera podríamos solucionar este problema? Actualmente Apple [21] en su sistema operativo OS X [23] tiene una herramienta llamada *Automator* [25] que enmascara a un usuario que no sabe de programación la tarea de crear pequeños programas que hagan ciertas tareas (conocidos como *flujos de trabajo*). Para crear un nuevo programa el usuario simplemente tiene que ir arrastrando pequeños componentes que realizan una acción concreta (por ejemplo, “listar todos los documentos en un directorio”) e ir uniendo más componentes al flujo (podría añadir: “seleccionar aquellos ficheros que tengan formato PDF”) hasta que obtiene un programa que realiza una tarea concreta (podría finalizar el flujo añadiendo: “y mover dichos archivos a la carpeta de mis documentos”). En la Figura 7.2 podemos ver el flujo real que se ha comentado antes. En TikiWiki se podría realizar algo similar con los *Profiles* y eso es algo que una persona que no tiene muchos conocimientos técnicos, puede asumirlo y aprender a usarlo.

2. Los *Profiles* que se han creado son muy “rígidos”, y asumen muchas cosas por defecto (por ejemplo sólo crea una *wiki*, pero no crea un foro). Se podrían crear más *Profiles* que abarquen más casos de uso.
3. El sistema de los *Data Channels* no tiene una implementación muy clara. Sería útil crear un apartado en TikiWiki donde se pudieran guardar los *Data Channels* creados al igual como se listan los recursos en las categorías. Esto permitiría mejorar la facilidad de uso de la plataforma.
4. Enriquecer y adaptar la documentación que se encuentra en: <http://doc.tiki.org>. Muchas veces cuando un usuario necesita saber cosas concretas de la plataforma tiene que acudir a la documentación oficial para aclarar sus dudas. El problema es que en muchas áreas ésta no está de forma clara o incluso contiene información anticuada de anteriores versiones que ya no es vigente y que no sirve. Mejorar la documentación ayudaría a que los usuarios tuviesen un mejor conocimiento de la plataforma.

7. CONCLUSIONES

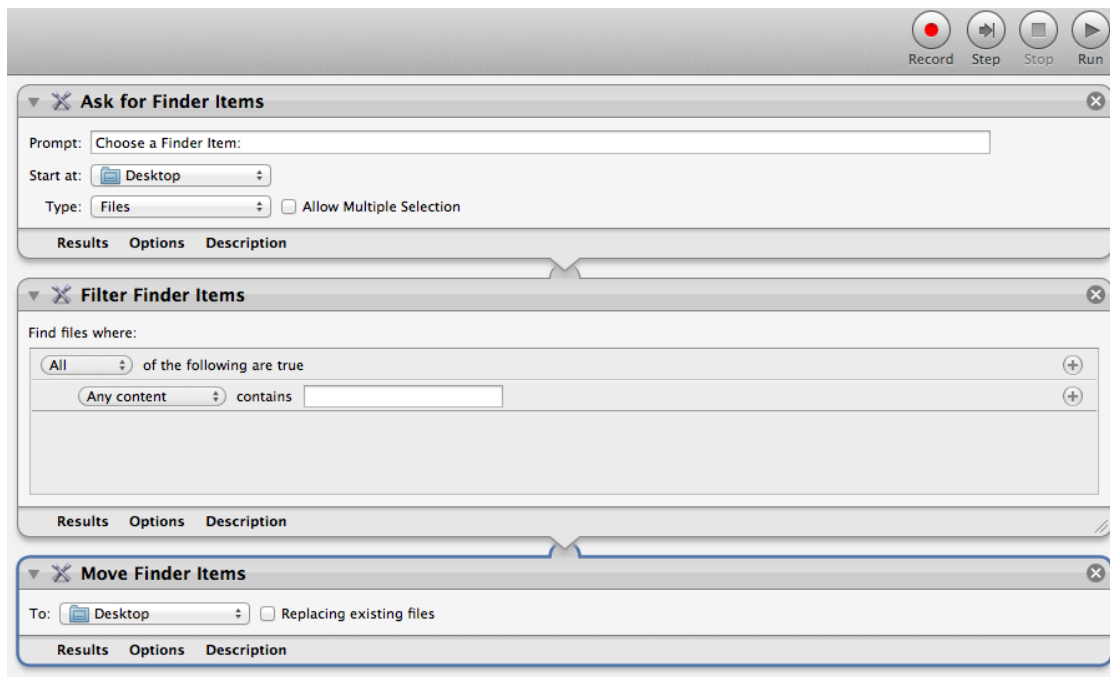


Figura 7.2: Ejemplo del programa Automator.

5. La traducción al castellano del propio TikiWiki es irregular y, en ocasiones, de mala calidad: se entremezclan oraciones en inglés con algunas en castellano. De cara a mostrar una buena imagen, se debería traducir de forma correcta.

APÉNDICES

A

YAML

A.1 ¿QUÉ ES YAML?

YAML Ain't Another Markup Language o YAML No Es Otro Lenguaje de Marcado (YAML) es un formato de serialización de datos creado por Clark Evans, Ingy döt Net y Oren Ben-Kiki en 2001 y que está fuertemente inspirado en lenguajes como Extensible Markup Language o Lenguaje de Marcado Extendido (XML), C, Python y Perl. A diferencia de XML, la sintaxis de YAML es muy liviana y es muy fácil de interpretar por las personas, de hecho, surgió como una alternativa a XML por considerar a éste un poco *engorroso* de crear documentos y, sobre todo, de leerlos.

Actualmente tiene una amplia aceptación en el mundo de la programación, donde, lenguajes tan populares y conocidos como son: C, Python, JAVA, Ruby, Haskell, JavaScript (y otros tantos más) tienen implementaciones, en forma de librerías externas en algunos casos o incluidas en el propio lenguaje en otros, de un intérprete de YAML. Los usos que se le pueden dar son tan variados que dependen del propio programador y lo que desee hacer con el formato aunque, generalmente, se suele utilizar para crear archivos de configuración que luego los programas pueden leer y modificar las preferencias del mismo; un ejemplo claro lo encontramos en el programa *Jekyll* [14]. Éste es un generador de sitios web estáticos (muy popular entre la comunidad de programadores) y que utiliza YAML para pasar datos básicos de configuración al propio programa (Figura A.1). Encontramos la misma situación en varios programas de renombre como: *Ruby on Rails*

Default Configuration

Note — You cannot use tabs in configuration files.

```
safe:      false
auto:      false
server:    false
server_port: 4000
baseurl:   /
url: http://localhost:4000

source:    .
destination: ./_site
plugins:   ./_plugins

future:    true
lsi:       false
pygments:  false
markdown:  maruku
permalink: date
```

Figura A.1: Ejemplo de configuración del programa Jekyll utilizando YAML.

[33] o *Symfony* [35] que son dos aplicaciones que sirven para crear páginas web dinámicas y que utilizan a YAML con el mismo objetivo: configurar las propias aplicaciones.

El hecho de que los *Profiles* utilicen YAML no es otro que por la claridad de lectura que se obtiene con ellos. Además, en el caso de PHP, hay una implementación del formato incluida en el propio lenguaje que está bastante optimizada.

En la web podemos encontrar multitud de recursos para aprender sobre la especificación de YAML. En este apéndice, en concreto, sólo se mostrarán los elementos más básicos del lenguaje. Si deseamos obtener más información no hay nada mejor que ir a la página oficial [7] donde siempre encontraremos la última versión de la especificación de YAML.

A. YAML

A.2 ESPECIFICACIÓN DE YAML

YAML tiene dos consideraciones básicas previas que se deben cumplir si se quiere producir un documento que sea validado por un intérprete:

- El documento tiene que estar estructurado utilizando la *indentación* con espacios en blanco, por contra no se permite el uso del tabulador para *indentar*.
- El documento debe de estar codificado en formato *Unicode*, bien sea en UTF-8 o UTF-16.

A continuación, vamos a ver los componentes básicos que conforman un documento en YAML (y realmente, los *Profiles* utilizan la combinación de estos elementos sencillos):

Comentarios

```
1 | # El símbolo almohadilla se utiliza como comentario.  
2 | edad: 20 # Los comentarios se pueden situar donde queramos
```

Comienzo de documento

```
1 | --- # Tres guiones indican al intérprete que comienza un nuevo documento de YAML  
2 | --- # Podemos tener tantos nuevos documentos como queramos en un mismo fichero  
3 | --- # Los dos primeros documentos están vacíos, este en cambio, no  
4 | edad: 24
```

Valores básicos

```
1 | ---  
2 | nombre: Aldo # Un valor básico es como una variable  
3 | edad: 20 # Se compone de una etiqueta : valor
```


A. YAML

Listas

```
1 --- # Frutas (Formato largo)
2 - Manzana # Conjunto de elementos que comparten algo
3 - Pera # Entre el guión - y el nombre hay un espacio
4 - Melón # En la línea siguiente comenzamos un nuevo documento de YAML
5
6 --- # Frutas (Formato corto)
7 [Manzana, Pera, Melón] # Ambas notaciones son equivalentes
```

Arrays Asociativos

```
1 # (Formato Largo) Se indentan, se muestran los espacios utilizados
2 ---
3     nombre: Aldo Borrero
4     edad: 24
5
6 --- # (Formato Corto) Se engloban en corchetes, no hay que indentar
7 {nombre: Aldo Borrero, edad: 24}
```

Cadenas de caracteres

```
1 --- # No hay que poner comillas a las cadenas
2 string: Puedo poner lo que quiera sin comillas
```

Texto de múltiples líneas

```
1 # Podemos tener textos que queramos preservar los retornos de línea
2
3 --- |
4     Esto es un ejemplo de un texto
5     que tiene múltiples líneas.
6     YAML preserva el texto
7     de manera inteligente.
8     ¡Como cabría esperar!
```

A. YAML

Texto de múltiples líneas convertido a una

```
1 | --- >
2 |     Este texto aunque
3 |     tiene múltiples
4 |     líneas
5 |     será convertida a una sola
```

Listas de arrays asociativos

```
1 | --- # Podemos crear estructuras más complejas utilizando las simples
2 | - {nombre: Aldo Borrero, edad: 24}
3 | - nombre: Michael Jackson
4 |   edad: 50
```

Arrays asociativos de listas

```
1 | ---
2 | alumnos: [Pepito Pérez, Jose María Gonzalo]
3 | mujeres: [] # No hay ninguna mujer, array vacío
4 | perros:
5 |   - Lassie
6 |   - Milú
```

B

CÓDIGO FUENTE

B.1 ORGANIZACIÓN DEL CÓDIGO DE TIKIWIKI

TikiWiki es un proyecto de código abierto que ha sido desarrollado completamente en PHP. La organización del código sigue una distribución precisa y lógica. Aún así, para una persona neófito que se interna por primera vez en el desarrollo de alguna librería o modificación, puede pensar que la estructura del propio proyecto es algo caótica. A continuación, se muestra un listado de los directorios y archivos más importantes de TikiWiki así como una breve descripción de ellos:

- *lib/*: Este directorio guarda todas las librerías principales de TikiWiki, si queremos realizar una modificación en el núcleo del programa este es nuestro sitio.
- *db/*: En este directorio se guardan las definiciones del esquema *relacional* de TikiWiki. Aquí es donde se aloja el archivo `local.php`.
- *db/local.php*: Este archivo guarda la configuración a la base de datos que hayamos instalado en TikiWiki. Si queremos reinstalar todo desde cero, simplemente hay que limpiar la base de datos y borrar este archivo.
- *templates/*: Este directorio guarda las plantillas programadas en Smarty [34].
- *templates_c/*: Este directorio sirve para guardar la *caché* de TikiWiki. Normalmente no es útil cambiar nada de aquí (aunque se puede borrar el directorio para obligar a que TikiWiki regenere los datos de nuevo).

B. CÓDIGO FUENTE

- *img*: Este directorio guarda la identidad de TikiWiki. Es útil por si se quiere cambiar el logotipo que muestra por defecto por uno propio.

B.2 LISTADO DE PROFILES

Se listan todos los *Profiles* que han sido diseñados para llevar a cabo este proyecto:

Profile: Configuración de ALMA

```
1 Este profile configura Tikiwiki para trabajar en ALMA.
2
3 Crea también los \profiles{} necesarios para crear asignaturas de manera automática.
4
5 Se puede ejecutar directamente en Admin > Profiles > Advanced > Profile Tester
6
7 Esta sección se encarga de configurar las opciones generales de Tiki. ¡No olvides
8 cambiar la contraseña del correo por la contraseña original!
9
10 Tampoco olvides copiar el contenido de Crear_Comunidad_de_Practica_Template
11 a la página wiki con el mismo nombre!
12
13 {CODE(caption="YAML" wrap="1")}
14 preferences:
15   language: es
16   feature_wysiwyg: y
17   wysiwyg_default: y
18   wysiwyg_optional: y
19   wikiplugin_equation: y
20   wikiplugin_perspective: n
21   wikiplugininline_perspective: n
22   wikiplugininline_equation: y
23   feature_blogs: n
24   feature_categories: y
25   feature_file_galleries: n
26   feature_forums: n
27   feature_freetags: n
28   feature_help: n
```

B. CÓDIGO FUENTE

```
29 feature_trackers: n
30 feature_myтики: y
31 feature_userPreferences: y
32 feature_profiles: y
33 feature_perspective: y
34 allowRegister: y
35 min_username_length: 1
36 max_username_length: 50
37 min_pass_length: 4
38 forgotPass: y
39 cookie_name: ALMA
40 wikiHomePage: HomePage
41 sender_email: alma@uah.es
42 siteTitle: ALMA
43 browsertitle: ALMA
44 site_title_location: before
45 server_timezone: Madrid/Europe
46 display_timezone: Madrid/Europe
47 zend_mail_handler: smtp
48 default_mail_charset: utf-8
49 zend_mail_smtp_user: alma.disciplinar@gmail.com
50 zend_mail_smtp_pass: password
51 zend_mail_smtp_port: 465
52 zend_mail_smtp_security: ssl
53 zend_mail_smtp_auth: plain
54 zend_mail_smtp_server: smtp.gmail.com
55 log_mail: n
56 wikiplugin_datachannel: y
57 profile_channels: crear_comunidad, tiki://local, Crear_Comunidad_de_Practica_Template, Admins
58 sitelogo_title: ALMA (Aula Libre Multidisciplinar Abierta)
59 sitelogo_src: img/tiki/tikisitelogo.png
60 sitelogo_alt: Alma
61 {CODE}
62
63 Esta sección se encarga de crear las páginas Wiki con el Profile de Crear
64 Comunidad de Paractica y su Data Channel correspondiente. Los asigna al grupo
65 de Profiles y sólo los admins son capaces de ejecutar y ver esas páginas.
66
```

B. CÓDIGO FUENTE

```
67 {CODE(caption="YAML" wrap="1")}
68 objects:
69   -
70     type: wiki_page
71     ref: wiki1
72     data:
73       name: Crear_Comunidad_de_Practica_GUI
74       content: "{DATACHANNEL(channel=crear_comunidad)}nombre_asignatura,
75       Nombre de la comunidad de practica{DATACHANNEL}"
76   -
77     type: wiki_page
78     ref: wiki2
79     data:
80       name: Crear_Comunidad_de_Practica_Template
81       content: Rellenar el contenido de esta pagina con el archivo del mismo nombre
82   -
83     type: category
84     ref: profilecateg
85     data:
86       name: Profiles
87       description: Categoria donde se almacenan los profiles de las comunidades de practica
88       items:
89         - [ wiki_page, $wiki1 ]
90         - [ wiki_page, $wiki2 ]
91 permissions:
92   Admins:
93     objects:
94       -
95         type: category
96         id: $profilecateg
97         allow:
98           - admin
99   Registered:
100     objects:
101       -
102         type: category
103         id: $profilecateg
104         deny: view, edit, view_category, edit_category
```

B. CÓDIGO FUENTE

```
105 Anonymous:
106   objects:
107     -
108       type: category
109       id: $profilecateg
110       deny: view, edit, view_category, edit_category
111 {CODE}
112
113 Creamos el menú que permite cambiar de asignatura a los usuarios que estén
114 registrados en la plataforma.
115
116 {CODE(caption="YAML", wrap="1")}
117 objects:
118   -
119     type: module
120     ref: switcher
121     data:
122       name: perspective
123       position: left
124       order: 10
125       groups: [ Registered ]
126       params:
127         title: Asignaturas
128 {CODE}
```

Profile: Creación Comunidad de Práctica

```
1  !!!!!Creación de una asignatura usando Profiles
2
3  {CODE(caption=>YAML,wrap=1)}
4  mappings:
5    Member: $profilerequest:nombre_asignatura$Asignatura sin nombre$ Alumnos
6    Lead: $profilerequest:nombre_asignatura$Asignatura sin nombre$ Profesores
7  objects:
8    -
9    type: category
10   ref: project_root
```

B. CÓDIGO FUENTE

```
11 data:
12   name: $profilerequest:nombre_asignatura$Asignatura sin nombre$
13   parent: 0
14   items:
15     - [ wiki page, $dashboard ]
16 -
17 type: perspective
18 ref: perspective
19 data:
20   name: $profilerequest:nombre_asignatura$sin nombre$
21   preferences:
22     category_jail: $project_root
23     wikiHomePage: $dashboard
24     browsertitle: $profilerequest:nombre_asignatura$Asignatura sin nombre$
25     style: fivealive.css
26     style_option: kiwi.css
27     feature_wysiwyg: y
28     wysiwyg_default: y
29     wysiwyg_optional: y
30 -
31 type: wiki_page
32 ref: dashboard
33 data:
34   name: $profilerequest:nombre_asignatura$Asignatura sin nombre$ Homepage
35   content: "Bienvenido a tu nueva asignatura"
36 permissions:
37 Member:
38   autojoin: y
39   allow:
40     - view
41     - edit
42     - wiki_view_history
43     - wiki_view_source
44     - minor
45     - upload_picture
46     - rollback
47     - view_category
48     - search
```


B. CÓDIGO FUENTE

```
49     - delete_account
50     - group_view
51     - group_view_members
52     - view_category
53     - add_object
54     - modify_object_categories
55 objects:
56     -
57     type: category
58     id: $project_root
59     allow:
60         - view
61         - edit
62         - wiki_view_history
63         - wiki_view_source
64         - minor
65         - upload_picture
66         - rollback
67         - view_category
68         - search
69         - delete_account
70         - group_view
71         - group_view_members
72         - add_object
73         - modify_object_categories
74     -
75     type: perspective
76     id: $perspective
77     allow: [ perspective_view ]
78 Lead:
79     autojoin: y
80     allow:
81         - view
82         - edit
83         - wiki_view_history
84         - wiki_view_source
85         - minor
86         - upload_picture
```

B. CÓDIGO FUENTE

```
87     - rollback
88     - view_category
89     - search
90     - delete_account
91     - group_view
92     - group_view_members
93     - view_category
94     - add_object
95     - modify_object_categories
96     - remove
97     - rollback
98     - wiki_attach_files
99     - admin_attachments
100    - view_attachments
101    - upload_picture
102    - minor
103    - rename
104    - lock
105    - edit_structures
106    - edit_copyrights
107    - wiki_view_comments
108    - wiki_view_ratings
109    - wiki_vote_ratings
110    - wiki_admin_ratings
111    - wiki_view_history
112    - use_HTML
113  objects:
114    -
115    type: category
116    id: $project_root
117    allow:
118      - view
119      - edit
120      - wiki_view_history
121      - wiki_view_source
122      - minor
123      - upload_picture
124      - rollback
```

B. CÓDIGO FUENTE

```
125     - view_category
126     - search
127     - delete_account
128     - group_view
129     - group_view_members
130     - view_category
131     - add_object
132     - modify_object_categories
133     - remove
134     - rollback
135     - wiki_attach_files
136     - admin_attachments
137     - view_attachments
138     - upload_picture
139     - minor
140     - rename
141     - lock
142     - edit_structures
143     - edit_copyrights
144     - wiki_view_comments
145     - wiki_view_ratings
146     - wiki_vote_ratings
147     - wiki_admin_ratings
148     - wiki_view_history
149     - use_HTML
150     -
151     type: perspective
152     id: $perspective
153     allow: [ perspective_view ]
154 {CODE}
```

Profile: Interfaz para Crear Comunidad de Practica

```
1 Este Profile genera una comunidad de práctica con el nombre que se le especifique:
2 {DATACHANNEL(channel=crear_comunidad)}
3 nombre_asignatura, Nombre asignatura
4 {DATACHANNEL}
```

C

MANUAL DE USUARIO

C.1 OBTENCIÓN DEL CÓDIGO FUENTE

Todo el código que se ha producido para este PFC se ha subido a un repositorio de *BitBucket* [1] (que es un servidor que permite alojar código fuente de manera gratuita y que utiliza el gestor de código distribuido *Mercurial*, conocido también como: hg [18]).

Para descargar el código utilizando hg podemos ejecutar el siguiente comando en la terminal:

```
1 | $ hg clone https://bitbucket.org/aldborrero/alma-profiles
```

También podemos obtener una versión comprimida lista para descargar por *http* si acudimos al siguiente enlace:

```
1 | https://bitbucket.org/aldborrero/alma-profiles/get/v1.2.zip
```

El directorio que obtendremos al descomprimir será el siguiente:

- `config.yml`: Este archivo contiene el *Profile* de configuración de ALMA.
- `Crear_Comunidad_de_Practica_GUI.yml`: Contiene el Data Channel.
- `Crear_Comunidad_de_Practica_Template.yml`: Contiene la implementación del *Profile* que genera las comunidades de práctica.
- `logo`: Este directorio en su interior posee el logotipo que utiliza ALMA.

C. MANUAL DE USUARIO

- **mail:** Este directorio contiene una implementación alternativa de la librería de correo que utiliza TikiWiki y que soluciona ciertos problemas que teníamos con la antigua. Los ficheros que hay dentro son:
 - **TikiMaillib.diff:** La nueva implementación de la librería de correo re-escrita por el autor del PFC. Hay que decir que la implementación no es completa, sólo funciona para los requisitos que tiene ALMA y no contempla otros casos de uso. Se desarrolló sólo con el objetivo de solucionar el problema que nos surgió, ¡no para reemplazar al original de manera completa!
 - **swift:** Código fuente del proyecto SwiftMailer [43] que es una librería escrita en PHP que permite trabajar con el envío de correos electrónicos de manera sencilla.

C.2 REQUISITOS PREVIOS

Para poder ejecutar el código fuente se asume que el usuario ha instalado correctamente TikiWiki y que no ha configurado más que las opciones básicas de la instalación. También se asume que el usuario posee una versión de UNIX, da igual si es Linux, OS X o BSD (aunque los comandos se pueden adaptar fácilmente a un entorno Windows).

C.3 INSTALACIÓN

El primer paso que se va a realizar es proceder a la sustitución de la librería de correo de TikiWiki, así pues, hay que seguir los siguientes pasos:

1. Copiar el contenido del directorio `mail` (no la carpeta).
2. Dirigirnos a la carpeta donde tengamos instalado TikiWiki.
3. Navegar hasta `lib/webmail`.
4. Pegar lo que copiamos en dicho directorio.
5. A continuación, vamos a usar la herramienta de UNIX, `patch` (si no se ha usado nunca esta herramienta se recomienda visitar el siguiente enlace: <http://www.>

C. MANUAL DE USUARIO

linuxtutorialblog.com/post/introduction-using-diff-and-patch-tutorial que incluye un tutorial que explica el procedimiento), nos situamos en el directorio donde hemos copiado antes y hay que ejecutar lo siguiente en la terminal:

```
1 | patch tikimaillib.php -i TikiMaillib.diff -o tikimaillib.php
```

6. Con esto la librería de TikiWiki queda con el *parche* puesto y funcionando para las necesidades de ALMA.

A continuación, vamos a instalar el *Profile* que configura ALMA. Para ello seguimos los siguientes pasos:

1. Abrimos el archivo `config.yml` y copiamos todo su contenido.
2. Acudimos al panel de administración de *Profiles* a la pestaña Avanzado. En el apartado denominado `Profile Tester` (el que nos permite ejecutar *Profiles* de prueba directamente) pegamos en el recuadro y cambiamos la contraseña de correo por la *correcta*, asignamos un nombre en la casilla `Test Profile Name`, por ejemplo: Configuración y ejecutamos.
3. ¡Ya está ALMA configurado!

¿Por qué en este caso hemos utilizado directamente el `Profile Tester` en lugar de crear una página wiki y crear un *Data Channel* y un botón de ejecución con la herramienta *plug-in Data Channel* como se explicó en el capítulo anterior? La respuesta es bien sencilla, esto es debido a que realmente no vamos a ejecutar dicho *Profile* de manera regular. Simplemente se utiliza una vez por cada instalación nueva en la que queramos replicar las bases del proyecto ALMA. Por lo tanto no tiene sentido que *perdamos* tiempo en realizar todos los pasos anteriores.

Ahora ya sólo queda realizar el último paso que es copiar el contenido del *Profile* que genera comunidades de práctica dentro de la página wiki:

1. Abrimos el documento `Crear_Comunidad_de_Practica_Template.yml` y copiamos todo el contenido.
2. Nos dirigimos como administrador a `Listar Páginas Wiki` y buscamos la página wiki del mismo nombre que el archivo.

C. MANUAL DE USUARIO

3. Pegamos el contenido en la página y guardamos.
4. ¡Ya podemos ejecutar nuestro *Profile* y crear comunidades de práctica visitando la página wiki `Crear_Comunidad_de_Practica_GUI.yml`!

ACRÓNIMOS

ALMA Aula Libre Multidisciplinar Abierta 2, 3, 5, 8, 13, 14, 16–19, 21, 34, 48, 50, 67, 69–71, 79, 99–101

JSON JavaScript Object Notation 60

PFC Proyecto Fin de Carrera 2, 3, 5, 22, 44, 48, 59, 69, 73, 81, 99, 100

PHP PHP Hypertext Preprocessor 19, 36, 79, 81, 86, 90, 100

SGA Learning Management System o Sistema de Gestión de Aprendizaje 16, 17

SGC Content Management System o Sistema de Gestión de Contenidos 16, 79

WYSIWYG What You See Is What You Get o Lo Que Ves Es Lo Que Obtienes 72

XML Extensible Markup Language o Lenguaje de Marcado Extendido 85

YAML YAML Ain't Another Markup Language o YAML No Es Otro Lenguaje de Marcado 50, 60, 79, 80, 82, 85–87

BIBLIOGRAFÍA

- [1] *BitBucket: Servidor de alojamiento de código utilizando los gestores GIT o HG.* URL: <http://bitbucket.org/>.
- [2] *Casos de usos en los que se puede utilizar y adaptar TikiWiki (Inglés).* URL: <http://tiki.org/use+cases>.
- [3] *Departamento de Automática en la UAH utilizando Moodle.* URL: <http://atc1.aut.uah.es/>.
- [4] *Documentación de PHP sobre los arrays.* URL: <http://es2.php.net/manual/es/intro.array.php>.
- [5] *Documentación oficial de los Profiles (Inglés).* URL: <http://profiles.tiki.org/Welcome+to+Tiki+Profiles>.
- [6] *Documentación que explica el uso de los Data Channels (Inglés).* URL: <http://profiles.tiki.org/Data+Channels>.
- [7] *Especificación oficial del lenguaje de marcado YAML (Inglés).* URL: <http://www.yaml.org/spec/1.2/spec.html>.
- [8] *Explicación del concepto de Object References utilizado en los Profiles (Inglés).* URL: <http://profiles.tiki.org/Object+References>.
- [9] *Explicación del Plugin Data Channel (Inglés).* URL: <http://doc.tiki.org/PluginDataChannel>.
- [10] *Facebook.* URL: <http://www.facebook.com>.
- [11] *Flickr.* URL: <http://www.flickr.com>.
- [12] *Google.* URL: <http://www.google.es>.
- [13] *Guía de instalación de TikiWiki (Inglés).* URL: <http://doc.tiki.org/Installation>.

BIBLIOGRAFÍA

- [14] *Jekyll programa escrito en Ruby que genera páginas web estáticas utilizando YAML (Inglés).* URL: <http://jekyllrb.com/>.
- [15] *Listado de organizaciones que utilizan TikiWiki como motor principal de su página web.* URL: <http://info.tiki.org/Featured+Tikis>.
- [16] *Listado de permisos existentes en TikiWiki.* URL: <http://doc.tiki.org/Permissions+List>.
- [17] *Listado de todos los Handlers de TikiWiki (Inglés).* URL: <http://profiles.tiki.org/Profile+Developer+Documentation>.
- [18] *Mercurial.* URL: <http://mercurial.selenic.com/>.
- [19] *Motor Ogre utilizando TikiWiki para su documentación.* URL: <http://www.ogre3d.org/tikiwiki/>.
- [20] *Página de Fedora Yum.* URL: <http://fedoraproject.org/wiki/Yum>.
- [21] *Página de Apple (Inglés).* URL: <http://apple.com>.
- [22] *Página de Debian Apt.* URL: <http://www.debian.org/doc/manuals/debian-reference/cho2.en.html>.
- [23] *Página del sistema operativo OS X de Apple (Inglés).* URL: <http://apple.com/macosx/>.
- [24] *Página de Pacman.* URL: <https://wiki.archlinux.org/index.php/Pacman>.
- [25] *Página no oficial de Apple que enseña a manejar la herramienta Automator incluida en el sistema por defecto (Inglés).* URL: <http://automator.us/leopard/index.html>.
- [26] *Página oficial de la plataforma e-learning de pago BlackBoard.* URL: <http://www.blackboard.com>.
- [27] *Página oficial del lenguaje de programación PHP.* URL: <http://www.php.net>.
- [28] *Página oficial del proyecto Alfresco.* URL: <http://www.alfresco.com/>.
- [29] *Página oficial del proyecto Liferay.* URL: <http://www.liferay.com/>.
- [30] *Página oficial del proyecto MediaWiki.* URL: <http://mediawiki.org>.
- [31] *Página oficial del proyecto Moodle.* URL: <http://moodle.org>.
- [32] *Página oficial del proyecto MySQL.* URL: <http://www.mysql.com/>.
- [33] *Página oficial del proyecto Ruby on Rails.* URL: <http://rubyonrails.org/>.

BIBLIOGRAFÍA

- [34] *Página oficial del proyecto Smarty Templates*. URL: <http://www.smarty.net/>.
- [35] *Página oficial del proyecto Symfony*. URL: <http://www.symfony-project.org/>.
- [36] *Página principal del proyecto de Código Abierto TikiWiki (Inglés)*. URL: <http://tiki.org>.
- [37] *Página que explica conceptos fundamentales relacionados con la Educación 2.0*. URL: <http://jjdeharo.blogspot.com/2010/02/herramientas-para-una-educacion-20.html>.
- [38] Eduardo Punset. *Excusas para no pensar*. Ediciones Destino, 2011. ISBN: 9788423344130.
- [39] Roberto J. López Sastre. “Aprendizaje y software social: Comunidades de práctica y wikis en el EEES”. En: 2007. URL: <http://agamenon.tsc.uah.es/Personales/rlopez/docs/lopez-edu-2007c.pdf>.
- [40] Roberto J. López Sastre y A. Pérez Köhler. “Innovación Docente y Software Libre”. En: 2007. URL: <http://agamenon.tsc.uah.es/Personales/rlopez/docs/lopez-edu-2007a.pdf>.
- [41] Roberto J. López Sastre y col. “FILWIT: Innovando con Wikis en el aula”. En: 2008. URL: <http://agamenon.tsc.uah.es/Personales/rlopez/docs/lopez-edu-2008.pdf>.
- [42] Barry Schwartz. *The Paradox of Choice: Why More Is Less*. HarperCollins, 2005. ISBN: 0060005696.
- [43] *SwiftMailer: Librería de PHP dedicada al correo*. URL: <http://swiftmailer.org/>.
- [44] *Twitter*. URL: <http://www.twitter.com>.
- [45] *WebCT en la UAH*. URL: <https://webcto8.uah.es>.
- [46] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge Univ. Press, 1988. ISBN: 0521663636.
- [47] Etienne Wenger, William M. Snyder y Richard McDermott. *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Harvard Business Press, 2002. ISBN: 1578513308.
- [48] *Wikipedia*. URL: <http://www.wikipedia.com>.
- [49] *Wikipedia: CMS (Inglés)*. URL: https://en.wikipedia.org/wiki/Content_management_system.

BIBLIOGRAFÍA

- [50] *Wikipedia: E-learning (Inglés)*. URL: <http://en.wikipedia.org/wiki/E-learning>.
- [51] *Wikipedia: La Era de la Información (Inglés)*. URL: http://en.wikipedia.org/wiki/Information_Age.
- [52] *Wikipedia: LMS (Inglés)*. URL: https://en.wikipedia.org/wiki/Learning_management_system.
- [53] *Wikipedia: Social Learning (Inglés)*. URL: https://en.wikipedia.org/wiki/Social_learning_theory.
- [54] *Wikis in plain English*. URL: <http://www.youtube.com/watch?v=-dnLooTdmLY>.